ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΕΣΩΤΕΡΙΚΩΝ

εκδδα
ΕΘΝΙΚΟ ΚΕΝΤΡΟ ΔΗΜΟΣΙΑΣ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΑΥΤΟΔΙΟΙΚΗΣΗΣ

ΥΠΟΕΡΓΟ: ΥΠΟΕΡΓΟ 2 «ΠΡΟΓΡΑΜΜΑΤΑ ΚΑΤΑΡΤΙΣΗΣ, ΑΝΑΠΤΥΞΗΣ ΔΕΞΙΟΤΗΤΩΝ, ΕΝΔΥΝΑΜΩΣΗΣ, ΠΙΣΤΟΠΟΙΗΣΗΣ - ΥΛΟΠΟΙΗΣΗ ΜΕ ΙΔΙΑ ΜΕΣΑ, ΕΠΙΜΟΡΦΩΣΗ ΑΠΟ ΤΟ ΕΚΔΔΑ» του Έργου «SUB4. Αναβάθμιση των δεξιοτήτων του ανθρώπινου δυναμικού του Δημόσιου Τομέα» με κωδικό ΟΠΣ ΤΑ 5150174
της Δράσης 16972 ΤΑΑ

ΤΙΤΛΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ:
ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ (SUPERVISED LEARNING) ΜΕ ΧΡΗΣΗ PYTHON

ΕΚΠΑΙΔΕΥΤΙΚΟ ΥΛΙΚΟ

Κωδικός εκπαιδευτικού υλικού:

Κωδικός Πιστοποίησης προγράμματος: 949

Ελλάδα 2.0
Με τη χρηματοδότηση της Ευρωπαϊκής Ένωσης
NextGenerationEU

Ελλάδα 2.0
ΕΘΝΙΚΟ ΣΧΕΔΙΟ ΑΝΑΚΑΜΨΗΣ ΚΑΙ ΑΝΘΕΚΤΙΚΟΤΗΤΑΣ
Με τη χρηματοδότηση της Ευρωπαϊκής Ένωσης
NextGenerationEU

ΥΠΟΕΡΓΟ: : ΥΠΟΕΡΓΟ 2 «ΠΡΟΓΡΑΜΜΑΤΑ ΚΑΤΑΡΤΙΣΗΣ, ΑΝΑΠΤΥΞΗΣ ΔΕΞΙΟΤΗΤΩΝ, ΕΝΔΥΝΑΜΩΣΗΣ, ΠΙΣΤΟΠΟΙΗΣΗΣ - ΥΛΟΠΟΙΗΣΗ ΜΕ ΙΔΙΑ ΜΕΣΑ, ΕΠΙΜΟΡΦΩΣΗ ΑΠΟ ΤΟ ΕΚΔΔΑ» του Έργου «SUB4. Αναβάθμιση των δεξιοτήτων του ανθρώπινου δυναμικού του Δημόσιου Τομέα» με κωδικό ΟΠΣ ΤΑ 5150174 της Δράσης 16972 ΤΑΑ

ΤΙΤΛΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ:
ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ (SUPERVISED LEARNING) ΜΕ ΧΡΗΣΗ PYTHON

ΟΜΑΔΑ ΕΡΓΑΣΙΑΣ

Μέλη Ομάδας

Συντονιστής/στρια:
Ιωάννης Ματζαβάκης

Συγγραφείς:
Αντώνιος Νείρος
Γεώργιος Ρηγόπουλος
Δημήτριος Καραπιπέρης
Νικόλαος Παπαχρήστου

Αξιολογητές/τριες:
Γεώργιος Παπαμιχαήλ
Λάμπρος Κωσταπαππάς

Ελλάδα 2.0
Με τη χρηματοδότηση της Ευρωπαϊκής Ένωσης
NextGenerationEU

Περιεχόμενα

Λίγα λόγια για τους συγγραφείς

Βιογραφικό Αντωνίου Νείρου

Ο Αντώνιος Νείρος έλαβε το πτυχίο του στην Πληροφορική από το Εθνικό & Καποδιστριακό Πανεπιστήμιο Αθηνών το 1996, το μεταπτυχιακό του δίπλωμα με τίτλο «Ψηφιακά Συστήματα Επικοινωνίας» από το Τμήμα Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών του Πανεπιστημίου Loughborough (Ηνωμένο Βασίλειο) το 1997, το μεταπτυχιακό του δίπλωμα από την σχολή Ανθρωπιστικών Σπουδών του Ελληνικού Ανοικτού Πανεπιστημίου με τίτλο «Σπουδές στην Εκπαίδευση» το 2016 και το Διδακτορικό του Δίπλωμα με τίτλο «Ανάπτυξη Μεθόδων Ασαφούς Συσταδοποίησης για τη Μοντελοποίηση Νευρωνικών Δικτύων Συναρτήσεων Ακτινικής Βάσης» από το Τμήμα Πολιτισμικής Τεχνολογίας & Επικοινωνίας του Πανεπιστημίου Αιγαίου το 2012. Από το 2000 εργάζεται ως μόνιμος καθηγητής Πληροφορικής ΠΕ86 στη Β/θμια Εκπαίδευση (σε Πρότυπο Γενικό Λύκειο και από το Σεπτέμβριο του 2021 είναι Διευθυντής στο ίδιο σχολείο). Είναι Μέλος ΣΕΠ στο Ελληνικό Ανοικτό Πανεπιστήμιο όπου διδάσκει σε μεταπτυχιακά μαθήματα από τον Οκτ. 2021. Έχει διδάξει σε Προπτυχιακά μαθήματα στο τμήμα Μηχανικών Οικονομίας και Διοίκησης της Πολυτεχνικής Σχολής του Πανεπιστημίου Αιγαίου, στο Τμήμα Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιά, στο Τμήμα Επικοινωνίας & Μέσων Μαζικής Ενημέρωσης του Εθνικού & Καποδιστριακού Πανεπιστημίου Αθηνών, στο Τμήμα Πολιτισμικής Τεχνολογίας & Επικοινωνίας του Πανεπιστημίου Αιγαίου, στην Ανώτατη Σχολή Παιδαγωγικής και Τεχνολογικής Εκπαίδευσης Α.Σ.ΠΑΙ.Τ.Ε.. Είναι Πιστοποιημένος Επιμορφωτής Β΄ Επιπέδου εκπαιδευτικών Πληροφορικής ΠΕ86. Είναι εκπαιδευτής ενηλίκων και έχει διδάξει σε πλήθος σεμιναρίων στο Ε.Κ.Δ.Δ.Α. Έχει δημοσιεύσει σε διεθνή επιστημονικά περιοδικά με κριτές και σε πρακτικά διεθνών συνεδρίων σε θέματα τεχνητής νοημοσύνης, τεχνητών νευρωνικών δικτύων, ασαφών συστημάτων και επεξεργασίας εικόνας. Είναι κριτής σε πολλά Διεθνή Επιστημονικά Περιοδικά και Συνέδρια. Συμμετείχε στην επιτροπή εξορθολογισμού της διδακτέας ύλης Πληροφορικής ΓΕ.Λ. του ΙΕΠ κατά το 2016-17 και εμπειρογνώμονας Πληροφορικής του ΙΕΠ για την ανάπτυξη του νέου Προγράμματος Σπουδών Πληροφορικής ΓΕΛ και του νέου Βιβλίου Πληροφορικής της Γ΄ ΓΕΛ κατά τα έτη 2018-19. Συμμετείχε ως εμπειρογνώμονας Πληροφορικής του ΙΕΠ στην επιτροπή εκπόνησης του νέου Προγράμματος Σπουδών Πληροφορικής του Γενικού Λυκείου.

Βιογραφικό Δημητρίου Καραπιπέρη

Ο Δρ. Δημήτριος Καραπιπέρης είναι πτυχιούχος του τμήματος Πληροφορικής του ΑΤΕΙΘ, έλαβε το μεταπτυχιακό του δίπλωμα από το Πανεπιστήμιο του York (2002) και το διδακτορικό του δίπλωμα από το Ελληνικό Ανοικτό Πανεπιστήμιο (2017). Οι μεταπτυχιακές του σπουδές χρηματοδοτήθηκαν από το Ίδρυμα Κρατικών Υποτροφιών (Ι.Κ.Υ.) μετά από εξετάσεις σε πανελλήνιο επίπεδο. Η διδακτορική τουδιατριβή επελέγει να συμπεριληφθεί στο IEEE Intelligent Informatics Bulletin τεύχος Αυγούστου 2017.

Αντικείμενο τη έρευνάς του είναι η αποτελεσματική διασύνδεση μεγάλου όγκου εγγραφών σε κατανεμημένες βάσεις δεδομένων με ταυτόχρονη προστασία της ιδιωτικότητας των δεδομένων. Στο πλαίσιο αυτό αναπτύσσει τυχαιοποιημένους αλγορίθμους έχοντας πάντα ως γνώμονα την παροχή θεωρητικών εγγυήσεων στα αποτελέσματα σε όσο το δυνατόν λιγότερο χρόνο εκτέλεσης. Παράλληλα, αναπτύσσει ασύγχρονο λογισμικό με χρήση πολυνηματικών διατάξεων όπου εφαρμόζει τους τυχαιοποιημένους αλγορίθμους σε κατανεμημένες βάσεις δεδομένων μεγάλου όγκου. Η τεχνολογία της συμπερίληψης (summarization) σε δεδομένα συνεχούς ροής (data streams) και το συνοδευτικό λογισμικό για την εξαγωγή αποτελεσμάτων σε πραγματικό χρόνο αποτελεί την ερευνητική του ενασχόληση τα τελευταία χρόνια με αντίστοιχες δημοσιεύσεις. Το συγγραφικό του έργο περιλαμβμάνει δημοσιεύσεις σε συνέδρια και περιοδικά διεθνούς κύρους όπως τα PVLDB, EDBT, IEEE TKDE, SIGKDD, IEEE BigData, DMKD, IEEE ICDE, IEEE ICDM κτλ.

Υπηρετεί ως συμβασιούχος διδάσκων σε προγράμματα μεταπτυχιακών σπουδών στο Διεθνές Πανεπιστήμιο της Ελλάδος και στο Ελληνικό Ανοικτό Πανεπιστήμιο. Διδάσκει Βάσεις Δεδομένων, Λειτουργικά Συστήματα, Προγραμματισμό στο Διαδίκτυο και Μηχανική Μάθηση.

Βιογραφικό Γεωργίου Ρηγόπουλου

Ο Γεώργιος Ρηγόπουλος είναι Επίκουρος Καθηγητής Πληροφορικής Οικονομικών Επιστημών στο Τμήμα Οικονομικών Επιστημών του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών (ΕΚΠΑ). Έλαβε το πτυχίο του από το Τμήμα Φυσικής του Πανεπιστημίου Πατρών, το MSc από το Οικονομικό Πανεπιστήμιο Αθηνών στην Διοικητική Επιστήμη και το διδακτορικό του στα Πληροφοριακά Συστήματα από το Εθνικό Μετσόβιο Πολυτεχνείο. Έχει επίσης πραγματοποιήσει μεταδιδακτορική διατριβή στην Αναλυτική Δεδομένων στο Πάντειο Πανεπιστήμιο.

Τα κύρια ερευνητικά του ενδιαφέροντα περιλαμβάνουν τα Πληροφορική Οικονομικών Επιστημών, Υπολογιστικά Οικονομικά, Πληροφοριακά Συστήματα, την Επιχειρησιακή

Έρευνα, Μηχανική Μάθηση και Επιστήμη Δεδομένων, καθώς και Δεδομένα μεγάλης κλίμακας και ανοικτά δεδομένα.

Έχει δημοσιεύσει περισσότερα από 30 άρθρα σε διεθνή περιοδικά με κριτές και έχει συμμετάσχει σε περισσότερα από 30 διεθνή και εθνικά συνέδρια. Έχει εργαστεί σε σειρά ερευνητικών προγραμμάτων και διδάσκει για περισσότερα από 15 έτη μαθήματα σε προπτυχιακό και μεταπτυχιακό επίπεδο μαθήματα πληροφορικής και ποσοτικών μεθόδων σε προπτυχιακά και μεταπτυχιακά προγράμματα. Έχει επιβλέψει σημαντικό αριθμό διπλωματικών και πτυχιακών εργασιών, είναι μέλος επιστημονικής επιτροπής, κριτής σε επιστημονικά περιοδικά και μέλος οργανωτικής επιτροπής συνεδρίων.

Βιογραφικό Νικόλαου Παπαχρήστου

Ο Νικόλαος Παπαχρήστου είναι κάτοχος μεταπτυχιακού (2010) και διδακτορικού διπλώματος (2015) από το τμήμα Εφαρμοσμένης Πληροφορικής του Πανεπιστημίου Μακεδονίας. Για αυτές τις μεταπτυχιακές σπουδές τού είχε απονεμηθεί υποτροφία από το Ίδρυμα Κρατικών Υποτροφιών (ΙΚΥ). Η έρευνά του επικεντρώνεται κυρίως στην Τεχνητή Νοημοσύνη με εφαρμογή στα παιχνίδια, στη μηχανική μάθηση, στην ενισχυτική μάθηση και στα νευρωνικά δίκτυα. Συμμετείχε 2 φορές σε διαγωνισμούς "ολυμπιάδων" προγραμμάτων παιχνιδιών (2011, 2015), όπου με το πρόγραμμα τεχνητής νοημοσύνης "Παλαμήδης" που ανέπτυξε κατέκτησε και τις δύο φορές την πρώτη θέση. Για αυτές τις επιτυχίες του έχουν απονεμηθεί α) εύφημος μνεία από τον Πρύτανη του Πανεπιστημίου Μακεδονίας (2011) και β) το βραβείο Tech Pioneer Award από το περιοδικό PC Magazine Greece (2012). Έχει εργαστεί ως καθηγητής πληροφορικής δευτεροβάθμιας εκπαίδευσης, ως προγραμματιστής στον ιδιωτικό και δημόσιο τομέα και έχει εκτεταμένη εμπειρία σε ανάπτυξη και συντήρηση πληροφοριακών συστημάτων.

ΚΕΦΑΛΑΙΟ 1: Εισαγωγή στην Μηχανική Μάθηση. Τύποι προβλημάτων και εφαρμογές.

Η μηχανική μάθηση είναι ένας τομέας της τεχνητής νοημοσύνης (ΤΝ) που επικεντρώνεται στην ανάπτυξη αλγορίθμων και μοντέλων που επιτρέπουν στους υπολογιστές να "μαθαίνουν" από δεδομένα και να κάνουν προβλέψεις ή λήψεις αποφάσεων χωρίς να είναι ρητά προγραμματισμένα για αυτό. Η μηχανική μάθηση χρησιμοποιείται ευρέως σε διάφορους τομείς, όπως η αναγνώριση προτύπων, η επεξεργασία φυσικής γλώσσας, η ανάλυση μεγάλων δεδομένων και η ρομποτική.

1.1. Τύποι Προβλημάτων στη Μηχανική Μάθηση

Τα προβλήματα στη μηχανική μάθηση μπορούν να κατηγοριοποιηθούν σε διάφορους τύπους, με βάση τον τύπο των δεδομένων που χρησιμοποιούνται και τον σκοπό της μάθησης. Οι κύριες κατηγορίες περιλαμβάνουν:

- **Επιβλεπόμενη Μάθηση (Supervised Learning)**: Στην επιβλεπόμενη μάθηση, το μοντέλο εκπαιδεύεται χρησιμοποιώντας ένα σύνολο δεδομένων που περιέχει εισόδους και τις αντίστοιχες εξόδους. Ο στόχος είναι να μάθει ένα μοντέλο που αντιστοιχίζει τις εισόδους στις αντίστοιχες εξόδους. Τα κύρια προβλήματα επιβλεπόμενης μάθησης είναι:
    - **Ταξινόμηση (Classification):** Η κατηγοριοποίηση των εισόδων σε προκαθορισμένες κατηγορίες. Π.χ., η αναγνώριση εικόνων (όπου οι εικόνες ταξινομούνται σε κατηγορίες όπως γάτες, σκύλοι κ.λπ.).
    - **Παλινδρόμηση (Regression):** Η πρόβλεψη συνεχών τιμών. Π.χ., η πρόβλεψη των τιμών των ακινήτων.
- **Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning):** Στη μη επιβλεπόμενη μάθηση, το μοντέλο εκπαιδεύεται χρησιμοποιώντας δεδομένα που δεν έχουν ετικέτες (labels). Ο στόχος είναι να ανακαλυφθούν υποκρυπτόμενες δομές (patterns) στα δεδομένα. Το κύριο πρόβλημα στην μη επιβλεπόμενης μάθησης είναι η **ομαδοποίηση (Clustering),** η οποία επιτυγχάνει την ταξινόμηση των δεδομένων σε ομάδες με βάση τις ομοιότητές τους. Π.χ., η τμηματοποίηση πελατών σε μια εταιρεία.

- **Ημι-Επιβλεπόμενη Μάθηση (Semi-Supervised Learning):** Η ημι-επιβλεπόμενη μάθηση είναι μια μέθοδος όπου χρησιμοποιούνται τόσο επισημασμένα όσο και μη επισημασμένα δεδομένα για την εκπαίδευση του μοντέλου. Αυτή η προσέγγιση είναι χρήσιμη όταν η λήψη επισημασμένων δεδομένων είναι δαπανηρή ή χρονοβόρα.
    - **Ενισχυτική Μάθηση (Reinforcement Learning):** Στην ενισχυτική μάθηση, ένας πράκτορας (agent) μαθαίνει να λαμβάνει αποφάσεις αλληλεπιδρώντας με ένα περιβάλλον. Ο πράκτορας λαμβάνει ανταμοιβές ή ποινές με βάση τις ενέργειές του και ο στόχος είναι να μεγιστοποιηθεί η συνολική ανταμοιβή. Παραδείγματα περιλαμβάνουν τη ρομποτική και τα παιχνίδια στρατηγικής.

1.2. Εφαρμογές της Μηχανικής Μάθησης

Η μηχανική μάθηση έχει εφαρμογές σε πολλούς τομείς, μερικοί από τους οποίους περιλαμβάνουν:

- **Αναγνώριση Προτύπων:**
    - **Αναγνώριση εικόνων και βίντεο:** Αναγνώριση προσώπων, αντικειμένων και σκηνών σε εικόνες και βίντεο.
    - **Αναγνώριση γραφής:** Αναγνώριση χειρόγραφου κειμένου.
- **Επεξεργασία Φυσικής Γλώσσας (Natural Language Processing - NLP):**
    - **Μετάφραση γλωσσών:** Μεταφραστές γλωσσών όπως το Google Translate χρησιμοποιούν μηχανική μάθηση για τη μετάφραση κειμένων από τη μία γλώσσα στην άλλη.
    - **Ανάλυση συναισθήματος:** Πρόκειται για την ανάλυση συναισθημάτων σε κείμενα όπως κριτικές προϊόντων και αναρτήσεις στα κοινωνικά δίκτυα.
- **Υγειονομική Περίθαλψη:**
    - **Διάγνωση ασθενειών:** Πρόβλεψη και τη διάγνωση ασθενειών με βάση τα ιατρικά δεδομένα των ασθενών.
    - **Προσωποποιημένη ιατρική:** Παροχή εξατομικευμένων προτάσεων θεραπείας.
- **Οικονομικά και Επιχειρήσεις:**
    - **Πρόβλεψη αγοράς:** Πρόβλεψη των τιμών των μετοχών και άλλων χρηματοοικονομικών δεικτών.
    - **Ανίχνευση απάτης:** Ανίχνευση απάτης στις συναλλαγές με πιστωτικές κάρτες.

- **Αυτόνομα Οχήματα:**
    - **Ανίχνευση αντικειμένων και πλοήγηση:** Ανίχνευση αντικειμένων κατα την πλοήγηση αυτόνομων οχημάτων στους δρόμους.

Η μηχανική μάθηση συνεχίζει να αναπτύσσεται και να επηρεάζει πολλούς τομείς της επιστήμης και της τεχνολογίας, προσφέροντας νέες δυνατότητες και λύσεις σε σύνθετα προβλήματα. Με τη συνεχή εξέλιξη των υπολογιστικών πόρων και των τεχνικών, οι εφαρμογές της μηχανικής μάθησης αναμένεται να επεκταθούν ακόμη περισσότερο στο μέλλον.

1.3. Εφαρμογές της μηχανικής μάθησης στη Δημόσια Διοίκηση

Η ενσωμάτωση της μηχανικής μάθησης στην ελληνική δημόσια διοίκηση μπορεί να επιφέρει σημαντικές βελτιώσεις στην αποδοτικότητα, την αποτελεσματικότητα και την ποιότητα των παρεχόμενων υπηρεσιών, ενώ παράλληλα να προάγει τη διαφάνεια και την αξιοπιστία των δημοσίων υπηρεσιών προς τους πολίτες.

Ενδεικτικά αναφέρουμε τον ψηφιακό βοηθό (chatbot) του gov.gr, επίσημα ονομαζόμενο ως mAigov, το οποίο αποτελεί μια καινοτόμα εφαρμογή τεχνητής νοημοσύνης/μηχανικής μάθησης που τέθηκε σε δοκιμαστική λειτουργία το 2023. Στόχος του είναι η βελτίωση της εμπειρίας των πολιτών κατά την αλληλεπίδρασή τους με την ψηφιακή διοίκηση. Το mAigov βασίζεται σε τεχνολογία κατανόησης φυσικής γλώσσας, επιτρέποντας στους πολίτες να διατυπώνουν ερωτήματα και αιτήματα με απλό και φυσικό τρόπο, σαν να μιλούσαν με έναν πραγματικό βοηθό. Ο εν λόγω ψηφιακός βοηθός έχει πρόσβαση σε πλήθος δεδομένων και πληροφοριών από διάφορες δημόσιες υπηρεσίες και έτσι δύναται να απαντήσει σε ερωτήματα σχετικά με διαδικασίες, δικαιολογητικά, προϋποθέσεις, ραντεβού, πληρωμές και πολλά άλλα. Συνολικά, ο ψηφιακος βοηθός του gov.gr αποτελεί ένα σημαντικό βήμα προς τον εκσυγχρονισμό της ψηφιακής διοίκησης στην Ελλάδα, προσφέροντας στους πολίτες μια πιο φιλική, άμεση και αποτελεσματική πρόσβαση σε πληροφορίες και υπηρεσίες.

1.4. Νευρωνικά δίκτυα

Τα νευρωνικά δίκτυα είναι ένα υποσύνολο της μηχανικής μάθησης εμπνευσμένο από τη δομή και λειτουργία του ανθρώπινου εγκεφάλου. Αποτελούνται από ένα πλήθος διασυνδεδεμένων

μονάδων, που ονομάζονται νευρώνες ή κόμβοι, οι οποίοι δουλεύουν μαζί για να επεξεργάζονται δεδομένα και να μαθαίνουν να εκτελούν συγκεκριμένα καθήκοντα. Κάθε νευρώνας λαμβάνει μία ή περισσότερες εισόδους, τις επεξεργάζεται και παράγει μία έξοδο. Αυτή η έξοδος στη συνέχεια μεταβιβάζεται στους επόμενους νευρώνες του δικτύου. Οι συνδέσεις μεταξύ των νευρώνων έχουν βάρη, τα οποία προσαρμόζονται κατά τη διάρκεια της διαδικασίας μάθησης ώστε το δίκτυο να μπορεί να πραγματοποιήσει επιθυμητές προβλέψεις ή αποφάσεις. Τα νευρωνικά δίκτυα εντάσσονται στην κατηγορία της επιβλεπόμενης μάθησης, αν και μπορούν να χρησιμοποιηθούν και σε άλλες κατηγορίες μάθησης ανάλογα με τον τρόπο που εκπαιδεύονται.

ΚΕΦΑΛΑΙΟ 2: Ανάλυση και οπτικοποίηση δεδομένων με την γλώσσα προγραμματισμού Python

Η Python είναι μια ευρέως χρησιμοποιούμενη γλώσσα προγραμματισμού υψηλού επιπέδου που δημιουργήθηκε από τον Guido van Rossum και κυκλοφόρησε για πρώτη φορά το 1991. Χαρακτηρίζεται από την ευκολία χρήσης και τη σαφήνεια του κώδικα της, καθιστώντας την ιδανική για αρχάριους, αλλά και για έμπειρους προγραμματιστές.

Η Python είναι γνωστή για τις παρακάτω ιδιότητες:

1. **Απλότητα και Αναγνωσιμότητα**: Η σύνταξη της Python είναι καθαρή και ευανάγνωστη, επιτρέποντας στους προγραμματιστές να γράφουν κώδικα που είναι εύκολο να κατανοηθεί και να διατηρηθεί.
2. **Δυναμική Τυποποίηση**: Η Python είναι μια δυναμικά τυποποιημένη γλώσσα, που σημαίνει ότι δεν απαιτεί ρητό ορισμό των τύπων των μεταβλητών, καθιστώντας την ανάπτυξη ταχύτερη.
3. **Πολυμορφισμός και Ευελιξία**: Υποστηρίζει πολλαπλά προγραμματιστικά παραδείγματα, όπως αντικειμενοστραφή, διαδικαστικό και λειτουργικό προγραμματισμό.
4. **Μεγάλη Κοινότητα και Υποστήριξη**: Η Python έχει μια μεγάλη και ενεργή κοινότητα που παρέχει υποστήριξη και συνεισφέρει σε χιλιάδες βιβλιοθήκες και εργαλεία.
5. **Πολυπλατφορμικότητα**: Είναι μια διαλειτουργική γλώσσα που τρέχει σε πολλαπλά λειτουργικά συστήματα όπως Windows, macOS και Linux.

Η Python χρησιμοποιείται ευρέως σε διάφορους τομείς όπως η ανάπτυξη διαδικτυακών εφαρμογών, η επιστήμη δεδομένων, η μηχανική μάθηση, η αυτοματοποίηση συστημάτων, και πολλές άλλες εφαρμογές.

Με αυτές τις ιδιότητες, η Python είναι μία από τις πιο δημοφιλείς και σε ζήτηση γλώσσες προγραμματισμού στον κόσμο, προσελκύοντας προγραμματιστές όλων των επιπέδων.

2.1. Πλατφόρμες και περιβάλλοντα

Η Python είναι εξαιρετικά ευέλικτη και μπορεί να τρέξει σε πολλές πλατφόρμες και περιβάλλοντα, καθιστώντας την προσιτή σε χρήστες με διαφορετικές ανάγκες και προτιμήσεις. Μια από τις πιο δημοφιλείς πλατφόρμες για την ανάπτυξη Python είναι τα Jupyter Notebooks, τα οποία προσφέρουν ένα διαδραστικό περιβάλλον για προγραμματισμό, ανάλυση δεδομένων και παρουσίαση αποτελεσμάτων. Παρακάτω θα βρείτε πληροφορίες για τις διάφορες πλατφόρμες στις οποίες μπορείτε να τρέξετε Python, με ιδιαίτερη έμφαση στα Jupyter Notebooks:

2.1.1. Τοπικά στον Υπολογιστή σας

Μπορείτε να εγκαταστήσετε Python και να τρέξετε κώδικα τοπικά στον υπολογιστή σας. Υπάρχουν διάφορες επιλογές:

- **Anaconda Distribution**: Μια δημοφιλής διανομή που περιλαμβάνει Python και μια συλλογή από πακέτα και εργαλεία, συμπεριλαμβανομένων των Jupyter Notebooks. Είναι ιδιαίτερα χρήσιμη για την επιστήμη δεδομένων και τη μηχανική μάθηση.
- **Εγκατάσταση Python και Jupyter Notebook ξεχωριστά**: Μπορείτε να εγκαταστήσετε την Python μέσω της επίσημης σελίδας της Python (python.org) και στη συνέχεια να εγκαταστήσετε το Jupyter Notebook χρησιμοποιώντας το `pip`.

2.1.2. Διαδικτυακές Πλατφόρμες

Υπάρχουν πολλές διαδικτυακές πλατφόρμες που σας επιτρέπουν να τρέχετε Python και Jupyter Notebooks χωρίς να χρειάζεται να εγκαταστήσετε τίποτα στον υπολογιστή σας:

- **Google Colab**: Μια δωρεάν πλατφόρμα που προσφέρεται από την Google, η οποία σας επιτρέπει να τρέχετε Jupyter Notebooks στο cloud. Είναι ιδανική για συνεργασία και παρέχει δωρεάν πρόσβαση σε GPU για ταχύτερους υπολογισμούς.
- **Kaggle**: Προσφέρει ένα περιβάλλον με Jupyter Notebooks όπου μπορείτε να εργαστείτε σε διαγωνισμούς δεδομένων, να κάνετε ανάλυση δεδομένων και να συνεργαστείτε με άλλους χρήστες.
- **Binder**: Ένα δωρεάν εργαλείο που σας επιτρέπει να δημιουργήσετε ένα φορητό περιβάλλον Jupyter Notebook που βασίζεται σε ένα αποθετήριο GitHub.

- **Microsoft Azure Notebooks**: Μια πλατφόρμα που σας επιτρέπει να δημιουργήσετε και να μοιραστείτε Jupyter Notebooks στο cloud.

2.1.3. Εταιρικά και Εκπαιδευτικά Περιβάλλοντα

Σε εταιρικά και εκπαιδευτικά περιβάλλοντα, η Python και τα Jupyter Notebooks χρησιμοποιούνται ευρέως για ανάλυση δεδομένων, εκπαίδευση και ερευνητικούς σκοπούς. Οι πλατφόρμες αυτές περιλαμβάνουν:

- **JupyterHub**: Μια πολυχρηστική πλατφόρμα που επιτρέπει σε ομάδες χρηστών να δουλεύουν με Jupyter Notebooks σε κοινόχρηστα υπολογιστικά περιβάλλοντα.
- **Databricks**: Παρέχει ένα συνεργατικό περιβάλλον για ανάλυση μεγάλων δεδομένων και μηχανική μάθηση, βασισμένο σε Apache Spark, με υποστήριξη για Jupyter Notebooks.

Η επιλογή της κατάλληλης πλατφόρμας εξαρτάται από τις ανάγκες σας, τον τύπο του έργου σας και τους διαθέσιμες πόρους. Οι Jupyter Notebooks προσφέρουν μια ισχυρή και ευέλικτη λύση για την ανάπτυξη Python, καθιστώντας τις ιδανικές για εκμάθηση, ανάλυση δεδομένων και πειραματισμό.

2.2. Μεταβλητές

Οι μεταβλητές στην Python είναι ένας από τους βασικούς πυλώνες της γλώσσας και χρησιμοποιούνται για την αποθήκευση δεδομένων τα οποία μπορούν να ανακληθούν και να χρησιμοποιηθούν αργότερα στον κώδικα. Παρακάτω θα δούμε μερικές βασικές έννοιες και ιδιότητες που αφορούν τις μεταβλητές στην Python:

Στην Python, η δήλωση και η ανάθεση μιας μεταβλητής είναι πολύ απλή. Δεν απαιτείται ρητός ορισμός του τύπου της μεταβλητής, καθώς η Python είναι μια δυναμικά τυποποιημένη γλώσσα. Αυτό σημαίνει ότι ο τύπος της μεταβλητής καθορίζεται από την τιμή που της ανατίθεται.

```
x = 10         # Ακέραιος αριθμός
y = 3.14       # Δεκαδικός αριθμός (float)
name = "John" # Συμβολοσειρά (string)
is_active = True  # Λογική τιμή (boolean)
```

Το σύμβολο `#` εισαγάγει ένα σχόλιο.

2.2.1. Τύποι Δεδομένων

Οι μεταβλητές στην Python μπορούν να αποθηκεύουν διάφορους τύπους δεδομένων, όπως:

- **Ακέραιοι** `(int)`: π.χ. `x = 10`
- **Δεκαδικοί** `(float)`: π.χ. `y = 3.14`
- **Συμβολοσειρές** `(str)`: π.χ. `name = "John"`
- **Λογικές τιμές** `(bool)`: π.χ. `is_active = True`

Υπάρχουν μερικοί κανόνες και καλές πρακτικές για την ονομασία των μεταβλητών στην Python:

- Τα ονόματα των μεταβλητών πρέπει να ξεκινούν με γράμμα (a-z, A-Z) ή κάτω παύλα (_), και μπορούν να περιέχουν αριθμούς (0-9).
- Είναι case-sensitive, δηλαδή το `name` και το `Name` είναι δύο διαφορετικές μεταβλητές.
- Καλό είναι να χρησιμοποιείτε περιγραφικά ονόματα που αντικατοπτρίζουν τη χρήση της μεταβλητής, π.χ. `student_name` αντί για `s`.

Μπορούμε να μετατρέψουμε τον τύπο μιας μεταβλητής χρησιμοποιώντας τις ενσωματωμένες συναρτήσεις της Python, όπως `int(), float(), str()`, κ.λπ.

```
x = 10
y = float(x)  # Μετατροπή του x από int σε float
```

```
name = "25"
age = int(name)  # Μετατροπή της name από string σε int
```

Τέλος, μπορούμε να ελέγξουμε τον τύπο μιας μεταβλητής χρησιμοποιώντας τη συνάρτηση `type()`.

```
x = 10
print(type(x))  # Εμφανίζει <class 'int'>
y = 3.14
print(type(y))  # Εμφανίζει <class 'float'>
```

Οι μεταβλητές είναι θεμελιώδεις για την ανάπτυξη προγραμμάτων στην Python. Με την κατανόηση των βασικών αρχών των μεταβλητών, μπορούμε να γράψουμε πιο αποτελεσματικό και κατανοητό κώδικα.

2.3. Η εντολή if

Η εντολή `if...else` στην Python είναι ένας βασικός τρόπος για να πραγματοποιείτε λογικούς ελέγχους και να εκτελείτε διαφορετικά τμήματα κώδικα ανάλογα με τις συνθήκες. Αυτή η δομή ελέγχου επιτρέπει την εκτέλεση συγκεκριμένων εντολών όταν μια συνθήκη είναι αληθής (`True`) και άλλων εντολών όταν η συνθήκη είναι ψευδής (`False`).

Η βασική μορφή της εντολής `if` είναι:

```
if condition:
    # Εντολές που θα εκτελεστούν αν η συνθήκη είναι True
```

Παράδειγμα:

```
x = 10
if x > 5:
    print("x is greater than 5")
```

Η εσοχή (indentation) είναι κρίσιμος κανόνας στην Python και χρησιμοποιείται για να δηλώσει τα μπλοκ του κώδικα. Σε αντίθεση με πολλές άλλες γλώσσες προγραμματισμού που χρησιμοποιούν άγκιστρα `{}`, η Python χρησιμοποιεί την εσοχή για να καθορίσει τη δομή και τη ροή του

προγράμματος. Η σωστή χρήση της εσοχής είναι απαραίτητη για τη σωστή λειτουργία του κώδικα και την αποφυγή συντακτικών σφαλμάτων.

2.3.1. Η εντολή if … else

Η εντολή `if...else` χρησιμοποιείται όταν θέλουμε να εκτελέσουμε διαφορετικές εντολές αν η συνθήκη είναι ψευδής.

```
if condition:
    # Εντολές που θα εκτελεστούν αν η συνθήκη είναι True
else:
    # Εντολές που θα εκτελεστούν αν η συνθήκη είναι False
```

Παράδειγμα:

```
x = 3
if x > 5:
    print("x is greater than 5")
else:
    print("x is not greater than 5")
```

Η εντολή `if...elif...else` χρησιμοποιείται όταν έχουμε πολλές συνθήκες για να ελέγξουμε.

```
if condition1:
    # Εντολές που θα εκτελεστούν αν η condition1 είναι True
elif condition2:
    # Εντολές που θα εκτελεστούν αν η condition2 είναι True
else:
    # Εντολές που θα εκτελεστούν αν καμία από τις παραπάνω συνθήκες
δεν είναι True
```

Παράδειγμα:

```
x = 5
if x > 10:
    print("x is greater than 10")
```

```
elif x == 5:
    print("x is equal to 5")
else:
    print("x is less than 10 and not equal to 5")
```

Μπορείτε να συνδυάσετε πολλαπλές συνθήκες χρησιμοποιώντας λογικούς τελεστές όπως `and`, `or`, και `not`.

```
x = 7
if x > 5 and x < 10:
    print("x is between 5 and 10")
```

2.4. Iterables

Τα iterables στην Python είναι αντικείμενα που μπορούν να επαναληφθούν (iterated) σε έναν βρόχο, επιτρέποντας την επεξεργασία κάθε στοιχείου τους με τη σειρά. Η έννοια των iterables είναι θεμελιώδης για πολλές λειτουργίες της γλώσσας και περιλαμβάνουν μια ποικιλία τύπων δεδομένων. Παρακάτω θα δούμε τους κύριους τύπους iterables και πώς μπορούν να χρησιμοποιηθούν.

Η εντολή `for` στην Python χρησιμοποιείται για να επαναλάβει ένα μπλοκ κώδικα για κάθε στοιχείο ενός iterable. Η βασική σύνταξη της εντολής `for` είναι η εξής:

```
for element in iterable:
    # Εντολές που θα εκτελεστούν για κάθε στοιχείο του iterable
```

Παρακάτω θα δούμε παραδείγματα χρήσης της εντολής `for`.

2.4.1. Λίστες (Lists)

Οι λίστες είναι ένας διατεταγμένος, μεταβαλλόμενος τύπος δεδομένων που μπορεί να περιέχει στοιχεία διαφόρων τύπων.

```
numbers = [1, 2, 3, 4, 5]
for num in numbers:
    print(num)
# Εμφανίζει 1 2 3 4 5
```

Η εντολή `append()` χρησιμοποιείται για να προσθέσουμε ένα νέο στοιχείο στο τέλος μιας λίστας. Το νέο στοιχείο προστίθεται στο τέλος της λίστας, αυξάνοντας το μήκος της κατά ένα.

```
my_list = [1, 2, 3, 4, 5]
my_list.append(6)
print(my_list) # Εμφανίζει [1, 2, 3, 4, 5, 6]
```

2.4.2. Πλειάδες (Tuples)

Οι πλειάδες είναι ένας διατεταγμένος, αμετάβλητος τύπος δεδομένων που μπορεί να περιέχει στοιχεία διαφόρων τύπων.

```
coordinates = (10.0, 20.0)
for coordinate in coordinates:
    print(coordinate)
# Εμφανίζει 10.0 20.0
```

2.4.3. Σύνολα (Sets)

Τα σύνολα είναι ένας μη διατεταγμένος τύπος δεδομένων που δεν επιτρέπει διπλότυπα στοιχεία.

```
unique_numbers = {1, 2, 3, 4, 5}
for num in unique_numbers:
    print(num)
# Εμφανίζει 1 2 3 4 5
```

2.4.4. Λεξικά (Dictionaries)

Τα λεξικά είναι ένας μη διατεταγμένος τύπος δεδομένων που αποθηκεύει ζεύγη κλειδιών-τιμών.

```
student = {"name": "John", "age": 20}
for key, value in student.items():
    print(key, value)
# Εμφανίζει name John age 20
```

2.4.5. Συμβολοσειρές (Strings)

Οι συμβολοσειρές είναι διατεταγμένοι, αμετάβλητοι τύποι δεδομένων που αποθηκεύουν ακολουθίες χαρακτήρων.

```
message = "Hello"
for char in message:
    print(char)
# Εμφανίζει H e l l o
```

2.4.6. Ενσωματωμένες Συναρτήσεις και Τεχνικές για Iterables

Η συνάρτηση `iter()` χρησιμοποιείται για να δημιουργήσει έναν iterator από ένα iterable, και η `next()` για να πάρει το επόμενο στοιχείο από τον εν λόγω iterator.

```
numbers = [1, 2, 3]
iterator = iter(numbers)
print(next(iterator))  # Εμφανίζει 1
print(next(iterator))  # Εμφανίζει 2
print(next(iterator))  # Εμφανίζει 3
```

Τα list comprehensions επιτρέπουν τη δημιουργία λιστών χρησιμοποιώντας εκφράσεις μέσα σε αγκύλες.

```
squares = [x**2 for x in range(4)]
print(squares) # Εμφανίζει [0, 1, 4, 9]
```

Η συνάρτηση `range()` στην Python χρησιμοποιείται για τη δημιουργία ακολουθιών αριθμών και είναι ιδιαίτερα χρήσιμη σε συνδυασμό με βρόχους `for`. Η `range()` μπορεί να δεχθεί έως τρεις

παραμέτρους: αρχική τιμή, τελική τιμή και βήμα. Η συνάρτηση επιστρέφει έναν iterator που παράγει αριθμούς από την αρχική τιμή μέχρι, αλλά όχι συμπεριλαμβανομένης, της τελικής τιμής.

Τέλος, μπορούμε να χρησιμοποιήσουμε και τις κάτωθι συναρτήσεις:

- `len()`: Επιστρέφει το μήκος ενός iterable.
- `sum()`: Επιστρέφει το άθροισμα των στοιχείων ενός iterable.
- `max()` και `min()`: Επιστρέφουν το μέγιστο και ελάχιστο στοιχείο αντίστοιχα.
- `sorted()`: Επιστρέφει ένα νέο διατεταγμένο iterable.
- `enumerate()`: Επιστρέφει ένα iterator που παρέχει δείκτη και τιμή για κάθε στοιχείο.

```
numbers = [1, 2, 3, 4, 5]
print(len(numbers))        # Εμφανίζει 5
print(sum(numbers))        # Εμφανίζει 15
print(max(numbers))        # Εμφανίζει 5
print(min(numbers))        # Εμφανίζει 1
print(sorted(numbers))     # Εμφανίζει [1, 2, 3, 4, 5]
for index, value in enumerate(numbers):
    print(index, value)
# Εμφανίζει 0 1  1 2  2 3  3 4  4 5
```

Τα iterables στην Python είναι πανίσχυρα εργαλεία που επιτρέπουν την επανάληψη και επεξεργασία στοιχείων με ευκολία και ευελιξία. Αυτές οι βασικές γνώσεις για τα iterables θα σας βοηθήσουν να γράψετε πιο αποδοτικό και καθαρό κώδικα.

2.5. Δεικτοδότηση και απόσπαση

Η δεικτοδότηση (indexing) και η απόσπαση (slicing) είναι δύο βασικές τεχνικές που χρησιμοποιούνται για την πρόσβαση και την τροποποίηση των στοιχείων μιας λίστας στην Python. Αυτές οι τεχνικές μπορούν να εφαρμοστούν όχι μόνο στις λίστες, αλλά και σε άλλες ακολουθίες (sequences) όπως συμβολοσειρές και πλειάδες. Οι ακολουθίες είναι ένας ειδικός τύπος iterable που διατηρεί τη σειρά των στοιχείων του. Αυτό σημαίνει ότι κάθε στοιχείο έχει έναν συγκεκριμένο δείκτη (index) που ορίζει τη θέση του στην ακολουθία.

2.5.1. Δεικτοδότηση

Η δεικτοδότηση επιτρέπει την πρόσβαση σε ένα συγκεκριμένο στοιχείο της λίστας χρησιμοποιώντας τον αριθμητικό δείκτη του. Οι δείκτες ξεκινούν από το 0 για το πρώτο στοιχείο, 1 για το δεύτερο, και ούτω καθεξής. Οι αρνητικοί δείκτες ξεκινούν από το -1 για το τελευταίο στοιχείο, -2 για το προτελευταίο, κ.λπ.

Παράδειγμα:

```
my_list = [10, 20, 30, 40, 50]
# Πρόσβαση στο πρώτο στοιχείο
print(my_list[0])  # Εμφανίζει 10
# Πρόσβαση στο τρίτο στοιχείο
print(my_list[2])  # Εμφανίζει 30
# Πρόσβαση στο τελευταίο στοιχείο
print(my_list[-1])  # Εμφανίζει 50
# Πρόσβαση στο προτελευταίο στοιχείο
print(my_list[-2])  # Εμφανίζει 40
```

2.5.2. Απόσπαση

Η απόσπαση επιτρέπει την απόσπαση ενός υποσυνόλου στοιχείων από μια λίστα, δημιουργώντας μια νέα λίστα που περιέχει τα επιλεγμένα στοιχεία. Η σύνταξη της απόσπασης είναι `list[start:stop:step]`, όπου:

- `start` είναι ο δείκτης του πρώτου στοιχείου που θα περιληφθεί (συμπεριλαμβανόμενο).
- `stop` είναι ο δείκτης του πρώτου στοιχείου που δεν θα περιληφθεί (μη συμπεριλαμβανόμενο).
- `step` είναι το βήμα μεταξύ των δεικτών.

Παράδειγμα:

```
my_list = [10, 20, 30, 40, 50]
# Απόσπαση των πρώτων τριών στοιχείων
print(my_list[0:3])  # Εμφανίζει [10, 20, 30]
# Απόσπαση των στοιχείων από το δεύτερο μέχρι και το τέταρτο
print(my_list[1:4])  # Εμφανίζει [20, 30, 40]
# Απόσπαση των στοιχείων από την αρχή μέχρι και το τρίτο στοιχείο
print(my_list[:3])  # Εμφανίζει [10, 20, 30]
# Απόσπαση των στοιχείων από το τρίτο μέχρι το τέλος
```

```
print(my_list[2:])  # Εμφανίζει [30, 40, 50]
# Απόσπαση όλων των στοιχείων
print(my_list[:])  # Εμφανίζει [10, 20, 30, 40, 50]
# Απόσπαση κάθε δεύτερου στοιχείου
print(my_list[::2])  # Εμφανίζει [10, 30, 50]
# Αντίστροφη σειρά των στοιχείων
print(my_list[::-1])  # Εμφανίζει [50, 40, 30, 20, 10]
```

2.6. Μεταβλητότητα αντικειμένων

Η μεταβλητότητα (mutability) είναι μια έννοια που αναφέρεται στη δυνατότητα ενός αντικειμένου να τροποποιηθεί μετά τη δημιουργία του. Στην Python, ορισμένες δομές δεδομένων είναι μεταβλητές (mutable), ενώ άλλες είναι μη μεταβλητές (immutable).

Παραδείγματα μεταβλητών iterables είναι οι λίστες και τα σύνολα. Ας δούμε ένα παράδειγμα με λίστα:

```
my_list = [1, 2, 3]
my_list.append(4)  # Προσθήκη ενός στοιχείου
print(my_list)  # Εμφανίζει [1, 2, 3, 4]
```

Στο παράδειγμα αυτό, μπορούμε να προσθέσουμε ένα νέο στοιχείο στη λίστα `my_list` με τη μέθοδο `append()`, δείχνοντας ότι η λίστα είναι μεταβλητή.

Ένα παράδειγμα μη μεταβλητού iterable είναι η πλειάδα. Δεν μπορούμε να αλλάξουμε τα στοιχεία μιας πλειάδας μετά τη δημιουργία της.

```
my_tuple = (1, 2, 3)
my_tuple[0] = 4  # Προσπάθεια αλλαγής του πρώτου στοιχείου
```

Αυτό το παράδειγμα θα οδηγήσει σε σφάλμα, καθώς οι πλειάδες είναι μη μεταβλητές. Όταν χρειαζόμαστε μη μεταβλητά δεδομένα, οι πλειάδες παρέχουν μια ασφαλή εναλλακτική όταν θέλουμε να δημιουργήσουμε μια δομή δεδομένων που δεν μπορεί να αλλάξει. Αυτό μπορεί να είναι χρήσιμο σε περιπτώσεις όπου θέλουμε να διασφαλίσουμε ότι οι τιμές μας παραμένουν σταθερές και δεν μπορούν να τροποποιηθούν από λάθος ή από αλλοιώσεις κατά την εκτέλεση του προγράμματος.

2.7. Αλφαριθμητικά

Τα αλφαριθμητικά (strings) είναι ένας από τους πιο βασικούς και σημαντικούς τύπους δεδομένων στην Python. Ένα αλφαριθμητικό είναι μια ακολουθία χαρακτήρων και μπορεί να περιέχει γράμματα, αριθμούς, σύμβολα, και διαστήματα. Τα αλφαριθμητικά μπορούν να δημιουργηθούν με μονά, διπλά ή και με τριπλά εισαγωγικά:

```
string1 = 'Γειά σου Κόσμε'
string2 = "Python Programming"
string3 = '''Αυτό είναι ένα πολύ
γραμμο κείμενο.'''
```

Τα αλφαριθμητικά υποστηρίζουν διάφορες λειτουργίες και μεθόδους που διευκολύνουν την επεξεργασία τους, όπως η συνένωση (concatenation) και η επανάληψη (repetition):

```
# Συνένωση
first_name = "John"
last_name = "Doe"
full_name = first_name + " " + last_name
# Αποτέλεσμα: "John Doe"

#Επανάληψη
repeat_str = "Hello " * 3
# Αποτέλεσμα: "Hello Hello Hello "
```

Τα αλφαριθμητικά, ως τύπος δεδομένων, ανήκουν στις ακολουθίες των iterables. Συνεπώς μπορεί να χρησιμοποιηθεί η δεικτοδότηση για να προσπελάσουμε συγκεκριμένους χαρακτήρες ενός αλφαριθμητικού.

```
my_string = "Hello"
first_char = my_string[0]
last_char = my_string[-1]
# Αποτέλεσμα: 'H' και 'o'
```

2.7.1. Ενσωματωμένες μέθοδοι

Η Python παρέχει πολλές ενσωματωμένες μεθόδους (συναρτήσεις) για την επεξεργασία αλφαριθμητικών. Ενδεικτικά αναφέρουμε τις παρακάτω:

- `len()`: Επιστρέφει το μήκος του αλφαριθμητικού.

```
length = len("Hello")
# Αποτέλεσμα: 5
```

- `lower()` και `upper()`: Μετατροπή των χαρακτήρων σε μικρά ή κεφαλαία γράμματα αντίστοιχα.

```
"Hello".lower()
# Αποτέλεσμα: "hello"
"Hello".upper()
# Αποτέλεσμα: "HELLO"
```

- `strip()`: Αφαιρεί τα κενά διαστήματα από την αρχή και το τέλος.

```
"  Hello  ".strip()
# Αποτέλεσμα: "Hello"
```

- `replace()`: Αντικαθιστά ένα υπο-αλφαριθμητικό με ένα άλλο.

```
"Hello World".replace("World", "Python")
# Αποτέλεσμα: "Hello Python"
```

- `split()`: Χωρίζει το αλφαριθμητικό σε λίστα με βάση έναν οριοθέτη (delimiter).

```
"Hello World".split()
# Αποτέλεσμα: ['Hello', 'World']
```

- `join()`: Ενώνει τα στοιχεία μιας λίστας σε ένα αλφαριθμητικό.

```
" ".join(['Hello', 'World'])
# Αποτέλεσμα: "Hello World"
```

2.7.2. Μορφοποίηση Αλφαριθμητικών

Την μορφοποίηση των αλφαριθμητικών η οποία περιλαμβάνει και την συνένωση επιμέρους αλφαριθμητικών την υλοποιούμε με τεχνικές όπως το f-string (interpolation) ή την συνάρτηση `format()`.

```
name = "John"
age = 30
formatted_string = f"My name is {name} and I am {age} years old."
# Αποτέλεσμα: "My name is John and I am 30 years old."

formatted_string = "My name is {} and I am {} years old.".format(name,
age)
# Αποτέλεσμα: "My name is John and I am 30 years old."
```

2.8. Εξαιρέσεις

Οι εξαιρέσεις (exceptions) στην Python είναι μηχανισμοί που χρησιμοποιούνται για να διαχειριστούν τα σφάλματα που μπορεί να προκύψουν κατά την εκτέλεση ενός προγράμματος. Όταν προκύψει ένα σφάλμα, η Python εγείρει (raise) μια εξαίρεση, η οποία μπορεί να ανακτηθεί και να διαχειριστεί με ειδικό τρόπο.

Η βασική δομή για τον χειρισμό των εξαιρέσεων στην Python περιλαμβάνει τα παρακάτω μπλοκ:

1. `try` : Περιέχει τον κώδικα που μπορεί να προκαλέσει εξαίρεση.
2. `except` : Περιέχει τον κώδικα που εκτελείται εάν προκληθεί εξαίρεση στο `try` block.
3. `else` : Περιέχει τον κώδικα που εκτελείται εάν δεν προκληθεί εξαίρεση στο `try` block.
4. `finally` : Περιέχει τον κώδικα που εκτελείται πάντοτε, ανεξάρτητα από το αν προκλήθηκε εξαίρεση ή όχι.

Παράδειγμα χρήσης:

```
try:
    # Κώδικας που μπορεί να προκαλέσει εξαίρεση
    result = 10 / 0
except ZeroDivisionError:
    # Κώδικας που εκτελείται εάν προκληθεί εξαίρεση ZeroDivisionError
    print("Δεν μπορείς να διαιρέσεις με το μηδέν!")
except Exception as e:
```

```
    # Κώδικας που εκτελείται για άλλες εξαιρέσεις
    print(f"Προέκυψε ένα σφάλμα: {e}")
else:
    # Κώδικας που εκτελείται εάν δεν προκληθεί εξαίρεση
    print("Η πράξη ολοκληρώθηκε επιτυχώς!")
finally:
    # Κώδικας που εκτελείται πάντοτε
    print("Η απόπειρα ολοκληρώθηκε.")
```

Συνήθεις Εξαιρέσεις:

1. `ValueError`: Προκύπτει όταν η τιμή δεν είναι κατάλληλη για τη δεδομένη λειτουργία.
2. `TypeError`: Προκύπτει όταν η λειτουργία εκτελείται σε ακατάλληλο τύπο δεδομένων.
3. `IndexError`: Προκύπτει όταν επιχειρείται πρόσβαση σε μη έγκυρη θέση λίστας.
4. `KeyError`: Προκύπτει όταν επιχειρείται πρόσβαση σε μη έγκυρο κλειδί λεξικού.

Οι εξαιρέσεις είναι ζωτικής σημασίας για τη δημιουργία αξιόπιστων και ανθεκτικών προγραμμάτων, επιτρέποντας τη διαχείριση απροσδόκητων καταστάσεων με οργανωμένο και καθαρό τρόπο.

2.9. Συναρτήσεις που ορίζονται από τον χρήστη

Οι συναρτήσεις που ορίζονται από τον χρήστη (user-defined functions) στην Python είναι εργαλεία που επιτρέπουν στους προγραμματιστές να οργανώνουν και να επαναχρησιμοποιούν τον κώδικα τους. Μια συνάρτηση μπορεί να δέχεται δεδομένα εισόδου, να εκτελεί μια σειρά εντολών και να επιστρέφει ένα αποτέλεσμα. Αυτό διευκολύνει τη διαχείριση πολύπλοκων προγραμμάτων και βελτιώνει την αναγνωσιμότητα και τη συντήρηση του κώδικα.

Για να ορίσουμε μια συνάρτηση στην Python, χρησιμοποιούμε τη λέξη-κλειδί `def`, ακολουθούμενη από το όνομα της συνάρτησης και παρενθέσεις που μπορεί να περιέχουν παραμέτρους.

```
def function_name(parameters):
    # Κώδικας της συνάρτησης
    return result
```

Οι συναρτήσεις μπορούν να δέχονται παραμέτρους, οι οποίες είναι μεταβλητές που χρησιμοποιούνται για να μεταφέρουν δεδομένα στη συνάρτηση. Μπορούμε να ορίσουμε προεπιλεγμένες τιμές για τις παραμέτρους:

```
def greet(name="Guest"):
    return f"Hello, {name}!"

print(greet())
# Αποτέλεσμα: Hello, Guest!
```

Τα ορίσματα μπορούν να περαστούν με βάση τη θέση τους ή με τη χρήση κλειδιών:

```
def describe_person(name, age):
    return f"{name} is {age} years old."

print(describe_person("John", 30))  # Θέση
# Αποτέλεσμα: John is 30 years old.

print(describe_person(age=30, name="John"))  # Κλειδιά
# Αποτέλεσμα: John is 30 years old.
```

Οι συναρτήσεις μπορούν να επιστρέφουν αποτελέσματα χρησιμοποιώντας τη λέξη-κλειδί `return`. Εάν δεν υπάρχει `return`, η συνάρτηση επιστρέφει `None` από προεπιλογή. Οι μεταβλητές που ορίζονται μέσα σε μια συνάρτηση έχουν τοπική εμβέλεια και δεν είναι ορατές εκτός της συνάρτησης.

2.9.1. Χρήση *args και **kwargs

Μπορούμε να χρησιμοποιήσουμε `*args` και `**kwargs` για να περάσουμε μεταβλητό αριθμό ορισμάτων σε μια συνάρτηση:

```
def multiply(*args):
    result = 1
    for num in args:
        result *= num
    return result

print(multiply(2, 3, 4))
# Αποτέλεσμα: 24
```

```python
def print_info(**kwargs):
    for key, value in kwargs.items():
        print(f"{key}: {value}")
print_info(name="John", age=30, city="New York")
# Αποτέλεσμα:
# name: John
# age: 30
# city: New York
```

2.10. Βιβλιοθήκη NumPy (https://numpy.org/)

Η Βιβλιοθήκη NumPy αποτελεί τον βασικό πυλώνα της επιστημονικής υπολογιστικής με την Python. Το NumPy (Numerical Python) είναι μια ισχυρή βιβλιοθήκη του Python που παρέχει εργαλεία για την αποτελεσματική επεξεργασία πινάκων πολλών διαστάσεων (arrays). Αυτοί οι πίνακες είναι παρόμοιοι με τους πίνακες σε άλλες γλώσσες προγραμματισμού, αλλά η NumPy τους καθιστά ιδιαίτερα γρήγορους και εύχρηστους, χάρη στην υλοποίησή τους σε γλώσσα C.

Οι πράξεις σε πίνακες NumPy είναι πολύ πιο γρήγορες από τις αντίστοιχες με απλές λίστες του Python. Επίσης παρέχεται μια μεγάλη ποικιλία από συναρτήσεις για μαθηματικές πράξεις, στατιστική ανάλυση, γραμμική άλγεβρα και άλλες επιστημονικές υπολογιστικές εργασίες.

Η NumPy αποτελεί τη βάση για πολλές άλλες βιβλιοθήκες επιστημονικής υπολογιστικής, όπως το SciPy, το Pandas και το Matplotlib.

Βασικές Έννοιες της NumPy:

- Πίνακες (Arrays): Οι βασικές δομές δεδομένων του NumPy.
- Άξονες (διαστάσεις): Οι πίνακες μπορούν να έχουν πολλές διαστάσεις (π.χ., μονοδιάστατοι, δισδιάστατοι, τρισδιάστατοι πίνακες).
- Δεικτοδότηση: Η πρόσβαση σε συγκεκριμένα στοιχεία ενός πίνακα.
- Εξαγωγή: Η εξαγωγή υπο-πίνακων.
- Broadcasting: Η εκτέλεση πράξεων σε πίνακες διαφορετικών μεγεθών.

Παράδειγμα:

```
import numpy as np
# Δημιουργία ενός πίνακα
a = np.array([1, 2, 3])

# Πρόσθεση μιας σταθεράς σε όλα τα στοιχεία του πίνακα και ανάθεση του
# αποτελέσματος στον πίνακα b
b = a + 1
# Αποτέλεσμα: [2 3 4]

# Γινόμενο πινάκων
c = a * a
# Αποτέλεσμα: [1 4 9]

# Υπολογισμός του μέσου όρου
```

```
mean_value = np.mean(a)
# Αποτέλεσμα: 2.0

#Δημιουργία δισδιάστατου πίνακα
a = np.array([ [1, 2, 3], [4, 5, 6] ])
```

Ένας άξονας (axis) σε έναν πίνακα NumPy αντιπροσωπεύει μια διάσταση του πίνακα. Ο συνολικός αριθμός των αξόνων σε έναν πίνακα αναφέρεται στην κατάταξή του (rank). Για παράδειγμα, ένας μονοδιάστατος πίνακα έχει έναν άξονα, ένας δισδιάστατος πίνακας έχει δύο άξονες (γραμμές και στήλες), και ένας τρισδιάστατος πίνακας έχει τρεις άξονες.

Η κατανόηση των αξόνων είναι σημαντική για διάφορους λόγους στο NumPy:

- **Πρόσβαση σε στοιχεία:** Μπορούμε να χρησιμοποιήσουμε τους δείκτες για να αποκτήσουμε πρόσβαση σε συγκεκριμένα στοιχεία ενός πίνακα, αναφέροντας τον δείκτη για κάθε άξονα.
- **Απόσπαση:** Η απόσπαση μας επιτρέπει να επιλέξουμε υποσύνολα ενός πίνακα κατά μήκος συγκεκριμένων αξόνων.
- **Broadcasting:** Το broadcasting είναι μια ισχυρή λειτουργία στο NumPy που επιτρέπει την εκτέλεση πράξεων σε πίνακες διαφορετικού μεγέθους. Οι άξονες διαδραματίζουν σημαντικό ρόλο στον τρόπο που γίνεται το broadcasting.
- **Επαναδιαμόρφωση πινάκων:** Μπορούμε να αλλάξουμε την μορφή των πινάκων αλλάζοντας τους άξονες τους.

Για παράδειγμα, στον παρακάτω κώδικα, δημιουργούμε έναν δισδιάστατο πίνακα `arr` και στη συνέχεια αποκτούμε πρόσβαση σε στοιχεία κατά μήκος των αξόνων του:

```
import numpy as np
# Δημιουργία ενός δισδιάστατου πίνακα
arr = np.array([[1, 2, 3], [4, 5, 6]])

# Άξονες (dimensions)
print(arr.ndim)
# Output: 2 (αριθμός διαστάσεων)

# Πρόσβαση σε στοιχεία κατά μήκος των αξόνων

print(arr[0, 1])
# Πρόσβαση στο στοιχείο στη γραμμή 0, στήλη 1
# Αποτέλεσμα: 2
```

```
print(arr[1])
# Πρόσβαση σε ολόκληρη τη δεύτερη γραμμή
# Αποτέλεσμα: [4 5 6]
```

Οι συναρτήσεις NumPy μπορούν να εφαρμοστούν κατά μήκος συγκεκριμένων αξόνων ενός πίνακα. Αυτή είναι μια χρήσιμη δυνατότητα που μας επιτρέπει να εκτελέσουμε πράξεις σε στοιχεία κατά μήκος των γραμμών, των στηλών ή άλλων διαστάσεων του πίνακα.

Μια κοινή συνάρτηση που χρησιμοποιείται με αυτόν τον τρόπο είναι η `np.sum`. Για παράδειγμα, μπορούμε να υπολογίσουμε το άθροισμα των στοιχείων κάθε γραμμής ή στήλης ενός πίνακα χρησιμοποιώντας τον κατάλληλο άξονα.

```
arr = np.array([1, 2, 3])
print(np.sum(arr, axis=0))
# Αποτέλεσμα: 6
```

Σε έναν δισδιάστατο πίνακα η συνάρτηση θα εφαρμοστεί στα στοιχεία του πρώτου άξονα (`axis=0`) που είναι ουσιαστικά οι στήλες:

```
arr = np.array([[1, 2, 3], [4, 5, 6]])
print(np.sum(arr, axis=0))
# Αποτέλεσμα: [5 7 9]
```

Ενώ η ίδια συνάρτηση στον ίδιο πίνακα αλλά στον δεύτερο άξονα (`axis=1`) θα επιφέρει διαφορετικό αποτέλεσμα αφου θα εφαρμοστεί στις γραμμές του πίνακα:

```
arr = np.array([[1, 2, 3], [4, 5, 6]])
print(np.sum(arr, axis=1))
# Αποτέλεσμα: [6 15]
```

2.10.1. Αναμόρφωση πινάκων

Η αναμόρφωση (reshaping) στη NumPy είναι η αλλαγή του σχήματος ενός πίνακα χωρίς να αλλαχτούν τα δεδομένα που περιέχει. Πιθανοί λόγοι για να προβούμε σε αναμόρφωση είναι οι κάτωθι:

- **Προσαρμογή σε άλλες λειτουργίες:** Πολλές λειτουργίες της NumPy απαιτούν συγκεκριμένα σχήματα πινάκων.
- **Οπτικοποίηση δεδομένων:** Μπορεί να είναι πιο εύκολο να κατανοήσουμε τα δεδομένα όταν είναι διατεταγμένα σε ένα συγκεκριμένο σχήμα.
- **Υπολογισμοί σε υποπίνακες:** Μπορούμε να εκτελέσουμε υπολογισμούς σε συγκεκριμένα τμήματα του πίνακα.

Η συνάρτηση `reshape()` είναι το βασικό εργαλείο για την αναμόρφωση. Παίρνει ως είσοδο τον πίνακα που θέλουμε να αλλάξουμε και το νέο επιθυμητό σχήμα.

Παράδειγμα:

```
import numpy as np

# Δημιουργία ενός πίνακα
arr = np.array([1, 2, 3, 4, 5, 6])

# Reshaping σε πίνακα 2x3
new_arr = arr.reshape(2, 3)
print(new_arr)
#Αποτέλεσμα:   [[1 2 3]
                [4 5 6]]
```

Το νέο σχήμα πρέπει να έχει τον ίδιο συνολικό αριθμό στοιχείων με τον αρχικό πίνακα. Η τάξη με την οποία τα στοιχεία τοποθετούνται στο νέο πίνακα εξαρτάται από τον τρόπο που γίνεται η αναμόρφωση. Μπορούμε να χρησιμοποιήσουμε το -1 σε μία από τις διαστάσεις για να υπολογιστεί αυτόματα από τη NumPy.

Παραδείγματα:

Από 1D σε 2D:

```
arr = np.arange(12)
```

```
new_arr = arr.reshape(3, 4)
# Αποτέλεσμα: [[0  1 2 3]
               [4 5 6 7]
               [8 9 10 11]]
```

Από 2D σε 1D:

```
arr = np.array([[1, 2, 3], [4, 5, 6]])
new_arr = arr.reshape(6)
# Αποτέλεσμα: [1 2 3 4 5 6]
```

Από 2D σε 3D:

```
arr = np.arange(24).reshape(3, 4, 2)
# Αποτέλεσμα:
[[[ 0  1]
  [ 2  3]
  [ 4  5]
  [ 6  7]]

 [[ 8  9]
  [10 11]
  [12 13]
  [14 15]]

 [[16 17]
  [18 19]
  [20 21]
  [22 23]]]
```

2.10.2. Δημιουργία ακολουθιών

Η NumPy παρέχει δύο βασικές συναρτήσεις για τη δημιουργία ακολουθιών αριθμών: την `np.arange` και την `np.linspace`. Αν και και οι δύο δημιουργούν ακολουθίες, έχουν κάποιες σημαντικές διαφορές στον τρόπο που ορίζουν αυτές τις ακολουθίες.

Η `np.arange` Δημιουργεί μια ακολουθία αριθμών με ένα συγκεκριμένο βήμα.

Σύνταξη: `np.arange(start, stop, step)`

- `start`: Η αρχική τιμή της ακολουθίας (συμπεριλαμβάνεται).
- `stop`: Η τελική τιμή της ακολουθίας (δεν συμπεριλαμβάνεται).
- `start`: Το βήμα μεταξύ κάθε στοιχείου.

Παράδειγμα:

```
import numpy as np

# Ακολουθία από 0 έως 10 με βήμα 2
a = np.arange(0, 11, 2)
print(a)
# Αποτέλεσμα: [ 0  2  4  6  8 10]
```

Η `np.linspace` Δημιουργεί μια ακολουθία αριθμών με έναν συγκεκριμένο αριθμό σημείων.

Σύνταξη: `np.linspace(start, stop, num)`

- `start`: Η αρχική τιμή της ακολουθίας.
- `stop`: Η τελική τιμή της ακολουθίας.
- `num`: Ο συνολικός αριθμός σημείων στην ακολουθία.

Παράδειγμα:

```
import numpy as np

# 5 ισαπέχοντα σημεία από 0 έως 1
b = np.linspace(0, 1, 5)
print(b)
# Αποτέλεσμα: [0.  0.25 0.5  0.75 1. ]
```

2.10.3. Δεικτοδότηση και απόσπαση

Η δεικτοδότηση (indexing) και η απόσπαση είναι δύο βασικές έννοιες στη NumPy που μας επιτρέπουν να έχουμε πρόσβαση σε συγκεκριμένα στοιχεία ή υποσύνολα ενός πίνακα.

Η δεικτοδότηση χρησιμοποιείται για να προσπελάσουμε ένα συγκεκριμένο στοιχείο ενός πίνακα. Η διαδικασία είναι παρόμοια με τις απλές λίστες στην Python, αλλά η NumPy επιπρόσθετα υποστηρίζει πολυδιάστατους πίνακες.

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5])
print(arr[2])  # Εκτυπώνει το στοιχείο στην τρίτη θέση (3)
```

και με την χρήση δισδιάστατου πίνακα:

```
arr = np.array([[1, 2, 3], [4, 5, 6]])
print(arr[1, 2])  # Εκτυπώνει το στοιχείο στη δεύτερη γραμμή και τρίτη στήλη (6)
```

Η απόσπαση (slicing) χρησιμοποιείται για να εξάγουμε ένα υποσύνολο ενός πίνακα. Η σύνταξη είναι παρόμοια με τις λίστες, αλλά μπορεί να εφαρμοστεί σε πολλές διαστάσεις.

Σύνταξη: `array[start:stop:step]`

- `start`: Ο δείκτης του πρώτου στοιχείου που θα συμπεριληφθεί (προαιρετικό, προεπιλογή 0).
- `stop`: Ο δείκτης του πρώτου στοιχείου που θα αποκλειστεί (προαιρετικό, προεπιλογή το τέλος του πίνακα).
- `step`: Το βήμα μεταξύ των στοιχείων (προαιρετικό, προεπιλογή 1).

```
arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
print(arr[2:5])  # Εκτυπώνει [2 3 4]
print(arr[::2])  # Εκτυπώνει κάθε δεύτερο στοιχείο [0 2 4 6 8]
```

Απόσπαση σε πολυδιάστατους πίνακες:

```
arr = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
print(arr[1:, :2])  # Εκτυπώνει τις δύο πρώτες στήλες από τη δεύτερη και τρίτη γραμμή
```

2.10.4. Broadcasting

Το broadcasting είναι μια ισχυρή λειτουργία στη NumPy που επιτρέπει την εκτέλεση πράξεων σε πίνακες διαφορετικού μεγέθους. Αντί να απαιτείται οι πίνακες να έχουν ακριβώς ίδιο σχήμα, το NumPy επεκτείνει αυτόματα τους μικρότερους πίνακες για να ταιριάξουν με τους μεγαλύτερους, ακολουθώντας συγκεκριμένους κανόνες.

Συγκεκριμένα, οι μικρότεροι πίνακες επεκτείνονται κατά μήκος των άξονων τους για να ταιριάξουν με το σχήμα των μεγαλύτερων πινάκων. Τα στοιχεία του μικρότερου πίνακα επαναλαμβάνονται κατά μήκος των άξονων που επεκτείνονται, ώστε να δημιουργηθεί ένας πίνακας με το ίδιο σχήμα με τον μεγαλύτερο.

Οι διαστάσεις των πινάκων πρέπει να είναι συμβατές. Αυτό σημαίνει ότι οι διαστάσεις πρέπει να είναι είτε ίσες είτε μία από αυτές να είναι 1. Οι διαστάσεις με μέγεθος 1 επεκτείνονται για να ταιριάξουν με τις αντίστοιχες διαστάσεις του άλλου πίνακα.

Παράδειγμα:

```
import numpy as np

a = np.array([1, 2, 3])
b = 2

c = a + b

print(c)
# Αποτέλεσμα: [3 4 5]
```

2.10.5. Τύποι δεδομένων

Το NumPy υποστηρίζει μια ποικιλία από τύπους δεδομένων, που ονομάζονται `dtypes`, οι οποίοι καθορίζουν τον τρόπο με τον οποίο αποθηκεύονται τα στοιχεία ενός πίνακα στη μνήμη. Η επιλογή του σωστού dtype είναι σημαντική για την αποδοτικότητα της μνήμης και των υπολογισμών.

Βασικοί Τύποι Δεδομένων:

- `int8, int16, int32, int64`**:** Ακέραιοι αριθμοί με διαφορετικό αριθμό bits.
- `uint8, uint16, uint32, uint64`**:** Ακέραιοι αριθμοί χωρίς πρόσημο.

- `float16, float32, float64:` Αριθμοί κινητής υποδιαστολής με διαφορετική ακρίβεια.
- `bool:` Boolean τιμές (`True` ή `False`).
- `complex64, complex128`: Μιγαδικοί αριθμοί.
- `string_`: Σειρές χαρακτήρων.

Μπορούμε να ορίσουμε τον τύπο δεδομένων κατα τον ορισμό δημιουργίας του πίνακα:

```
import numpy as np
# Πίνακας με ακέραιους αριθμούς 64-bit
arr = np.array([1, 2, 3], dtype=np.int64)

# Πίνακας με αριθμούς κινητής υποδιαστολής 32-bit
arr_float = np.array([1.5, 2.5, 3.5], dtype=np.float32)
```

Μπορούμε επίσης να αλλάξουμε τον τύπο δεδομένων ενός πίνακα χρησιμοποιώντας τη μέθοδο `astype():`

```
arr_int = np.array([1, 2, 3])
arr_float = arr_int.astype(np.float32)
```

Η επιλογή του κατάλληλου τύπου δεδομένων μπορεί να μειώσει σημαντικά τη μνήμη που χρησιμοποιείται για την αποθήκευση ενός πίνακα. Για αριθμητικούς υπολογισμούς, η επιλογή του κατάλληλου τύπου είναι κρίσιμη για την ακρίβεια των αποτελεσμάτων. Ορισμένες συναρτήσεις και αλγόριθμοι μπορεί να απαιτούν συγκεκριμένους τύπους δεδομένων.

2.10.6. Συνένωση πινάκων

οι συναρτήσεις `concatenate(), vstack()` και `hstack()` χρησιμοποιούνται για να συνδυάσουν πολλούς πίνακες σε έναν μεγαλύτερο. Ωστόσο, έχουν κάποιες διαφορές στον τρόπο λειτουργίας τους:

- `concatenate()`: Όταν χρειαζόμαστε πλήρη έλεγχο πάνω στον άξονα συνένωσης και όταν θέλουμε να συνενώσουμε πίνακες με διαφορετικό αριθμό διαστάσεων.

- `vstack()`: Όταν θέλουμε να συνενώσουμε πίνακες τοποθετώντας τους κάθετα, χωρίς να χρειάζεται να καθορίσουμε τον άξονα.
- `hstack()`: Όταν θέλουμε να συνενώσουμε πίνακες τοποθετώντας τους οριζόντια, χωρίς να χρειάζεται να καθορίσουμε τον άξονα.

Παραδείγματα:

```
import numpy as np
arr1 = np.array([[1, 2], [3, 4]])
arr2 = np.array([[5, 6], [7, 8]])
# Σύνδεση με concatenate
result_concat = np.concatenate((arr1, arr2), axis=0)
# Αποτέλεσμα: [[1 2]
               [3 4]
               [5 6]
               [7 8]]


result_concat = np.concatenate((arr1, arr2), axis=1)
# Αποτέλεσμα: [[1 2 5 6]
               [3 4 7 8]]



# Σύνδεση με vstack
result_vstack = np.vstack((arr1, arr2))
# Αποτέλεσμα: [[1 2]
               [3 4]
               [5 6]
               [7 8]]


# Σύνδεση με hstack
result_hstack = np.hstack((arr1, arr2))
# Αποτέλεσμα: [[1 2 5 6]
              [3 4 7 8]]
```

2.11. Βιβλιοθήκη Pandas (https://pandas.pydata.org/)

Η Pandas είναι μια βιβλιοθήκη ανοικτού κώδικα που έχει σχεδιαστεί για να χειρίζεται και να αναλύει δεδομένα αποτελεσματικά. Παρέχει δομές δεδομένων υψηλού επιπέδου που διευκολύνουν την εργασία με αριθμητικούς πίνακες και χρονοσειρές, καθιστώντας το ένα από τα πιο δημοφιλή εργαλεία για την ανάλυση δεδομένων στον κόσμο της επιστήμης των δεδομένων.

Το Pandas παρέχει δομές δεδομένων όπως το πλαίσιο δεδομένων `DataFrame`, το οποίο είναι παρόμοιο με ένα φύλλο εργασίας Excel, καθιστώντας την εισαγωγή, επεξεργασία και την ανάλυση δεδομένων πολύ πιο απλή. Χρησιμοποιεί βελτιστοποιημένους αλγορίθμους για να χειριστεί μεγάλους όγκους δεδομένων γρήγορα και αποτελεσματικά. Επίσης, παρέχει μια μεγάλη γκάμα από ενσωματωμένες λειτουργίες για την καθαρισμό, την προετοιμασία και την ανάλυση δεδομένων, όπως:

- Υπολογισμός στατιστικών μεγεθών.
- Ομαδοποίηση και συγκέντρωση δεδομένων.
- Συγχώνευση και συνένωση διαφορετικών πηγών δεδομένων.
- Δημιουργία γραφημάτων και οπτικοποιήσεων.

Τέλος, ενσωματώνεται εύκολα με άλλες βιβλιοθήκες Python όπως το NumPy, το Matplotlib και το Scikit-learn, επιτρέποντας την δημιουργία ολοκληρωμένων μελετών ανάλυσης δεδομένων.

Το Pandas παρέχει μια εύχρηστη συνάρτηση, τη `read_csv(),` για να διαβάσει δεδομένα από ένα CSV αρχείο και να τα φορτώσει σε ένα `DataFrame`,

```
import pandas as pd
# Διαδρομή προς το CSV αρχείο
file_path = "data.csv"
 # Διαβάζουμε το CSV αρχείο, όπου τα δεδομένα είναι  διαχωρισμένα με
 το ερωτηματικό (;) και το αποθηκεύουμε σε ένα DataFrame
 df = pd.read_csv(file_path, sep=”;”)
 # Εμφανίζουμε τις πρώτες 5 γραμμές του DataFrame
 print(df.head())
```

Μπορούμε να δημιουργήσουμε ένα πλαίσιο δεδομένων, το οποίο είναι ένας δισδιάστατος και δυνητικά ετερογενής πίνακας δεδομένων, από ένα λεξικό ως εξής:

```
import pandas as pd
# Δημιουργία ενός DataFrame με ετικέτες
```

```
data = {'Αθήνα': [25, 30, 28], 'Θεσσαλονίκη': [22, 25, 27]}
index = ['Δευτέρα', 'Τρίτη', 'Τετάρτη']
df = pd.DataFrame(data, index=index)
```

Σε ένα πλαίσιο δεδομένων, οι ετικέτες, τις οποίες τις καθορίζουμε με την ιδιότητα `index`, είναι μοναδικά αναγνωριστικά που αντιστοιχούν σε κάθε γραμμή. Πρόκειται ουσιαστικά περι ονομάτων που δίνουμε σε κάθε γραμμή για να μπορούμε να τις αναζητήσουμε και να τις χειριστούμε εύκολα, όπως θα δούμε παρακάτω.

2.11.1. Μέθοδος `loc()`

Η μέθοδος `loc()` στα πλαίσια δεδομένων είναι ένα εξαιρετικά ισχυρό εργαλείο για την επιλογή συγκεκριμένων στοιχείων, γραμμών ή στηλών με βάση τις ετικέτες τους. Είναι ιδιαίτερα χρήσιμη όταν θέλουμε να ανακτήσουμε δεδομένα με βάση την θέση τους μέσα στο πλαίσιο.

Παρακάτω συνοψίζουμε τις βασικές λειτουργίες επιλογής, όπου η μεταβλητή df είναι ένα πλαίσιο δεδομένων:

- **Επιλογή γραμμών:**
  - **Επιλογή γραμμών με βάση τις ετικέτες:**
    - `df.loc['index_label']`: Επιλέγει τη γραμμή με την ετικέτα `'index_label'`.
    - `df.loc[['index_label1', 'index_label2']]`: Επιλέγει πολλές γραμμές.
  - **Επιλογή γραμμών με βάση ένα εύρος ετικετών:**
    - `df.loc['start_label':'end_label']`: Επιλέγει όλες τις γραμμές από την `'start_label'` μέχρι την `'end_label'`.
- **Επιλογή στηλών:**
  - **Βάσει ονομάτων στηλών:**
    - `df.loc[:, 'column_name']`: Επιλέγει όλες τις γραμμές της στήλης `'column_name'`.
    - `df.loc[:, ['column1', 'column2']]`: Επιλέγει πολλές στήλες.
- **Επιλογή συγκεκριμένων στοιχείων:**
  - `df.loc['index_label', 'column_name']`: Επιλέγει το στοιχείο στη διασταύρωση της γραμμής 'index_label' και της στήλης 'column_name'.
- **Επιλογές με κριτήρια τιμών στις στήλες:**

- **Boolean indexing:**
  - `df.loc[df['column_name'] > 5]`: Επιλέγει όλες τις γραμμές όπου η τιμή στη στήλη `'column_name'` είναι μεγαλύτερη από `5`.
- **Επιλογή με βάση πολλαπλά κριτήρια:**
  - `df.loc[(df['column1'] > 5) & (df['column2'] < 10)]`: Επιλέγει γραμμές που ικανοποιούν και τα δύο κριτήρια.

Επιπλέον παραδείγματα:

```
import pandas as pd

# Δημιουργία ενός DataFrame
data = {'column1': [1, 2, 3], 'column2': [4, 5, 6]}
df = pd.DataFrame(data, index=['index1', 'index2', 'index3'])

# Επιλογή της γραμμής με ετικέτα 'index2'
row = df.loc['index2']

# Επιλογή των στηλών 'column1' και 'column2'
selected_columns = df.loc[:, ['column1', 'column2']]

# Επιλογή του στοιχείου στη διασταύρωση της γραμμής 'index1' και της στήλης 'column1'
value = df.loc['index1', 'column1']

# Επιλογή όλων των γραμμών όπου η τιμή στη στήλη 'column1' είναι μεγαλύτερη από 2
filtered_df = df.loc[df['column1'] > 2]
```

2.12. Βιβλιοθήκη Matplotlib (https://matplotlib.org/)

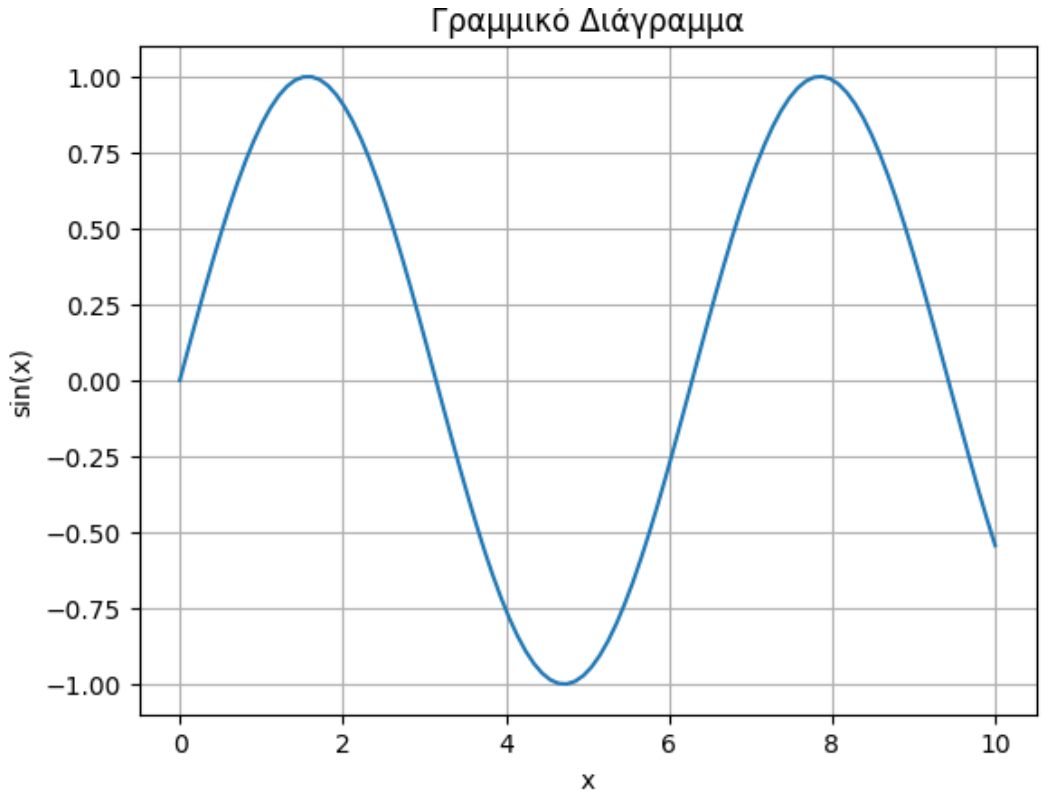
Η Matplotlib είναι μια βιβλιοθήκη που χρησιμοποιείται ευρέως για την δημιουργία οπτικοποιήσεων. Προσφέρει μια ευρεία γκάμα εργαλείων για την δημιουργία διαγραμμάτων, γραφημάτων και άλλων τύπων οπτικοποιήσεων.

- Τύποι διαγραμμάτων:
  - **Γραμμικά διαγράμματα (line plots)**: Για την απεικόνιση αλλαγών σε συνεχή δεδομένα.
  - **Διαγράμματα διασποράς (scatter plots):** Για την απεικόνιση της σχέσης μεταξύ δύο μεταβλητών.
  - **Ιστογράμματα (histograms):** Για την απεικόνιση της κατανομής των δεδομένων.
  - **Πίτες (pie charts):** Για την απεικόνιση της αναλογίας των μερών ενός συνόλου.
  - **Ραβδογράμματα (bar charts):** Για την σύγκριση κατηγοριών.
  - Και πολλά άλλα: Box plots, heatmaps, contour plots, κ.λπ.
- Προσαρμογή:
  - Τίτλοι: Προσθήκη τίτλων στα διαγράμματα και στους άξονες.
  - Ετικέτες: Προσθήκη ετικετών στους άξονες για να περιγράψουν τα δεδομένα.
  - Χρώματα: Επιλογή χρωμάτων για τις γραμμές, τα σημεία και τα στοιχεία του διαγράμματος.
  - Στυλ γραμμών: Επιλογή διαφορετικών τύπων γραμμών (συνεχής, διακεκομμένη, κ.λπ.).
  - Σήματα: Επιλογή διαφορετικών συμβόλων για τα σημεία στα διαγράμματα διασποράς.
- Αποθήκευση:
  - Αποθήκευση των διαγραμμάτων σε διάφορες μορφές αρχείων (PNG, PDF, SVG, κ.λπ.).

2.12.1. Παράδειγμα δημιουργίας γραμμικού διαγράμματος

```
import matplotlib.pyplot as plt
import numpy as np

# Δημιουργία δεδομένων
x = np.linspace(0, 10, 100)
y = np.sin(x)
```

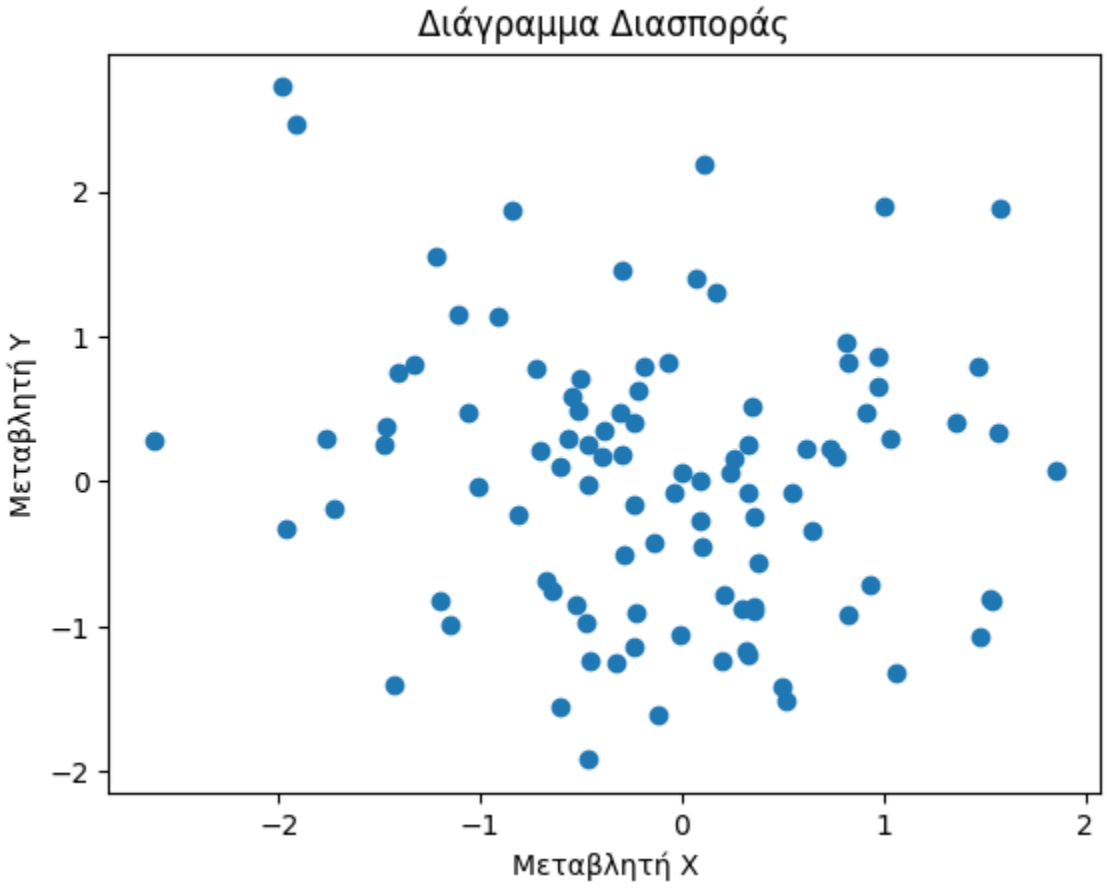
```
# Δημιουργία Figure και Axes
plt.figure()
plt.plot(x, y)
plt.xlabel('x')
plt.ylabel('sin(x)')
plt.title('Γραμμικό Διάγραμμα')
plt.grid(True)

# Εμφάνιση του διαγράμματος
plt.show()
```



Διάγραμμα 2.1: Γραμμικό διάγραμμα με Matplotlib

2.12.2. Παράδειγμα δημιουργίας διαγράμματος διασποράς

```
import matplotlib.pyplot as plt
import numpy as np
# Δημιουργία τυχαίων δεδομένων
np.random.seed(42)
x = np.random.randn(100)
y = np.random.randn(100)

# Δημιουργία του scatter plot
plt.scatter(x, y)
plt.xlabel('Μεταβλητή X')
plt.ylabel('Μεταβλητή Y')
plt.title('Διάγραμμα Διασποράς')
plt.show()
```



Διάγραμμα 2.2: Διάγραμμα διασποράς με Matplotlib

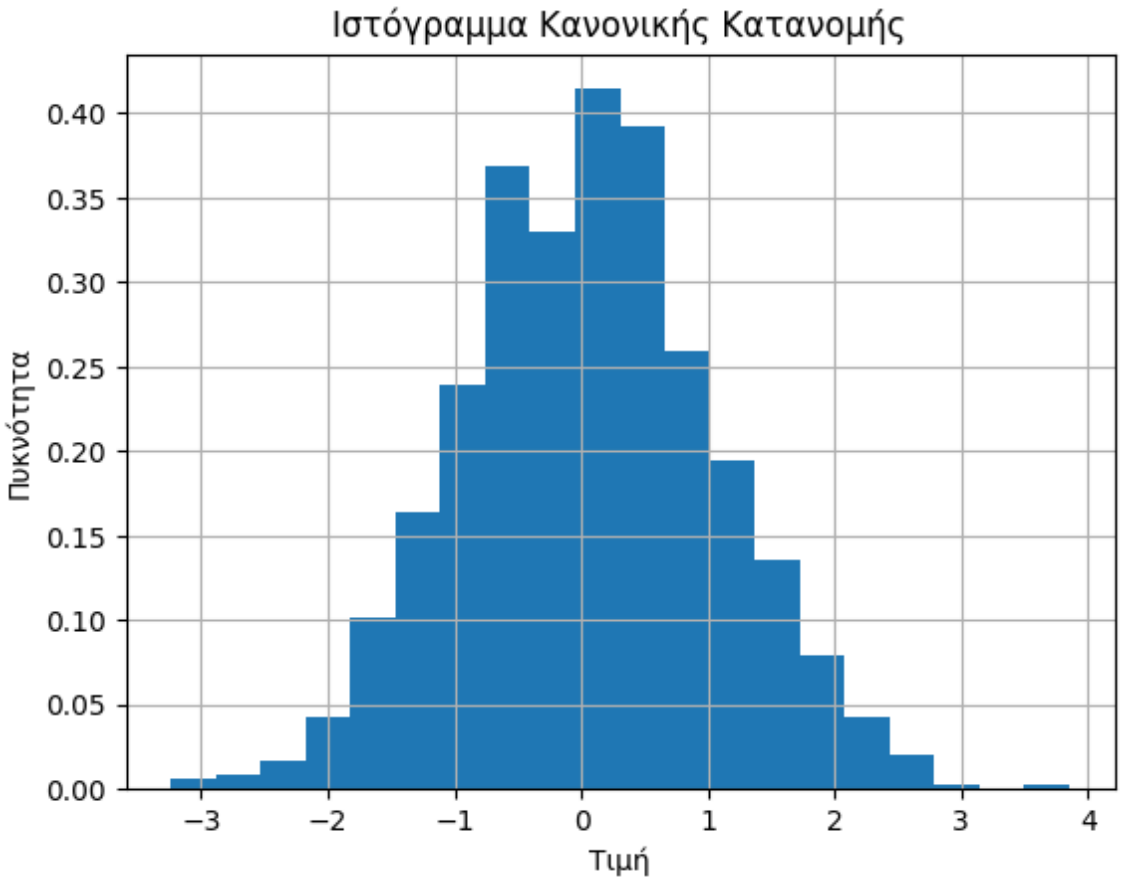
2.12.3. Παράδειγμα δημιουργίας πίτας

```
import matplotlib.pyplot as plt
# Δεδομένα
labels = 'Apples', 'Bananas', 'Cherries', 'Dates'
sizes = [15, 30, 45, 10]
colors = ['yellowgreen', 'gold', 'lightskyblue', 'lightcoral']
 explode = (0, 0.1, 0, 0)
# Απομακρύνει το δεύτερο κομμάτι από το κέντρο

# Δημιουργία του πίτα γραφήματος
plt.pie(sizes, explode=explode, labels=labels, colors=colors,
autopct='%1.1f%%', shadow=True, startangle=90)

plt.axis('equal')
# Ισόπλευροι άξονες για να φαίνεται κυκλικό

plt.title('Κατανάλωση Φρούτων')
plt.show()
```

Διάγραμμα 2.3: Διάγραμμα πίτας με Matplotlib

2.12.4. Παράδειγμα δημιουργίας ιστογράμματος

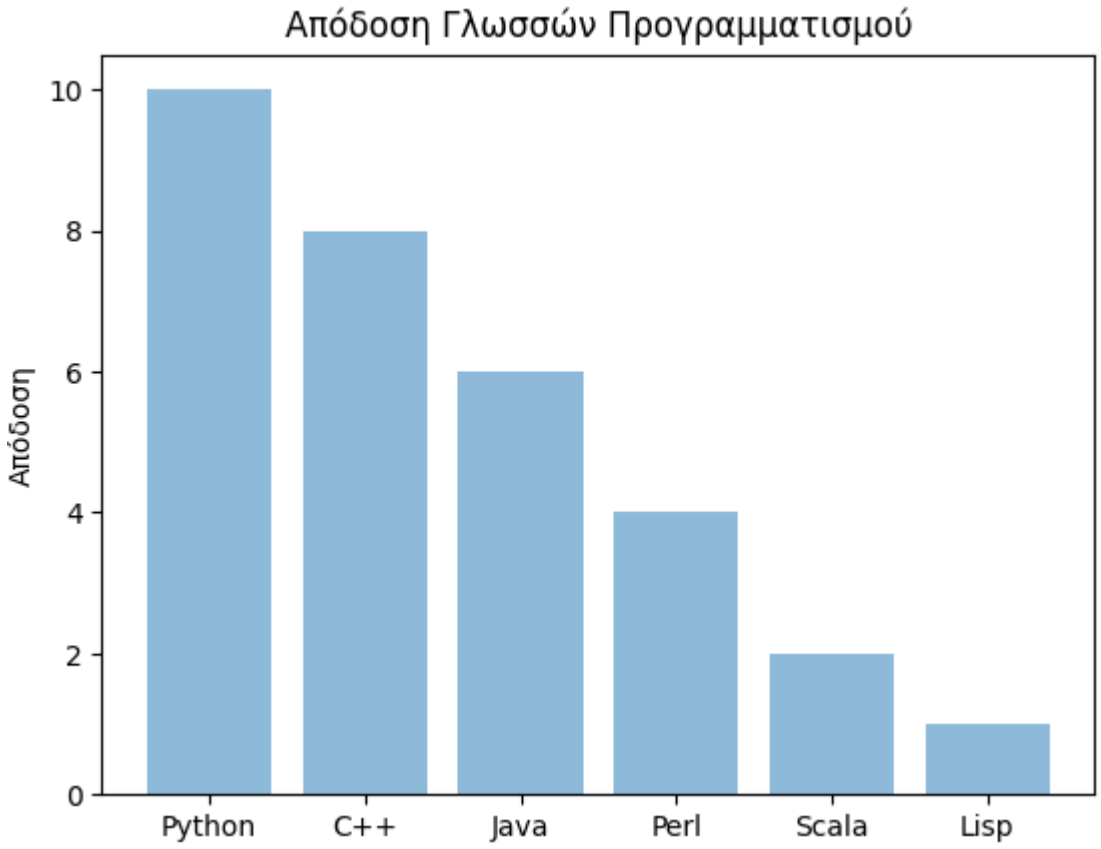
```
import matplotlib.pyplot as plt
import numpy as np
# Δημιουργία τυχαίων δεδομένων
data = np.random.randn(1000)
# Δημιουργία του ιστογράμματος
plt.hist(data, bins=20, density=True)
plt.xlabel('Τιμή') plt.ylabel('Πυκνότητα')
plt.title('Ιστόγραμμα Κανονικής Κατανομής')
plt.grid(True)
plt.show()
```



Διάγραμμα 2.4: Ιστόγραμμα με Matplotlib

2.12.5. Παράδειγμα δημιουργίας ραβδογράμματος

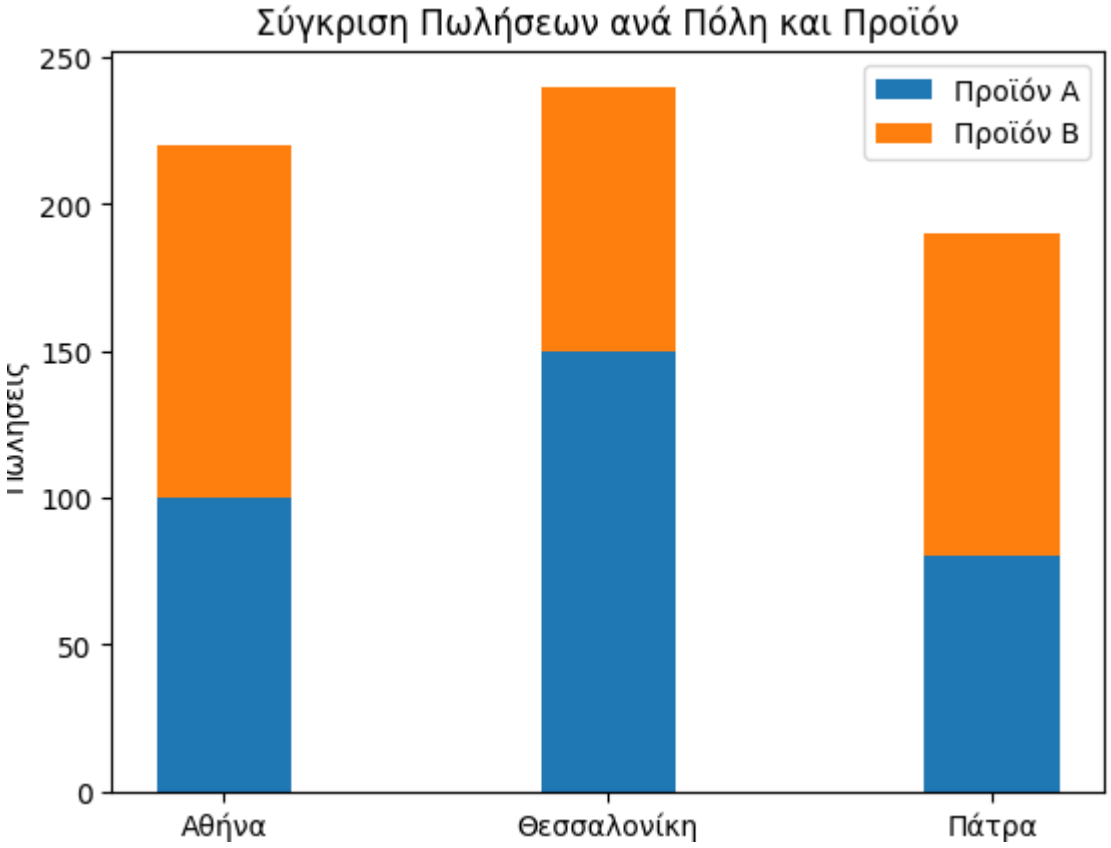
```
import matplotlib.pyplot as plt
import numpy as np
# Δεδομένα
objects = ('Python', 'C++', 'Java', 'Perl', 'Scala', 'Lisp')
y_pos = np.arange(len(objects))
performance = [10, 8, 6, 4, 2, 1]
# Δημιουργία του ραβδογράμματος
plt.bar(y_pos, performance, align='center', alpha=0.5)
plt.xticks(y_pos, objects)
plt.ylabel('Απόδοση')
plt.title('Απόδοση Γλωσσών Προγραμματισμού')
plt.show()
```



Διάγραμμα 2.5: Ραβδόγραμμα με Matplotlib

Μπορούμε να δημιουργήσουμε και ένα σύνθετο ραβδόγραμμα (stacked barchart) όπου σε ένα διάγραμμα περιλαμβάνονται παραπάνω από μία μεταβλητές.

```
import matplotlib.pyplot as plt
import numpy as np

# Δημιουργία δεδομένων
cities = ['Αθήνα', 'Θεσσαλονίκη', 'Πάτρα']
product_A = [100, 150, 80]
product_B = [120, 90, 110]

# Πλάτος των ράβδων
width = 0.35

# Δημιουργία του ραβδογράμματος
plt.bar(cities, product_A, width, label='Προϊόν Α' )
plt.bar(cities, product_B, width, label='Προϊόν Β', bottom=product_A)

# Ετικέτες και τίτλος
plt.ylabel('Πωλήσεις')
plt.title('Σύγκριση Πωλήσεων ανά Πόλη και Προϊόν')
plt.xticks(x, cities)
plt.legend()

# Εμφάνιση του διαγράμματος
plt.show()
```



Διάγραμμα 2.6: Σύνθετο ραβδόγραμμα με Matplotlib

2.13. Βιβλιοθήκη Seaborn (https://seaborn.pydata.org/)

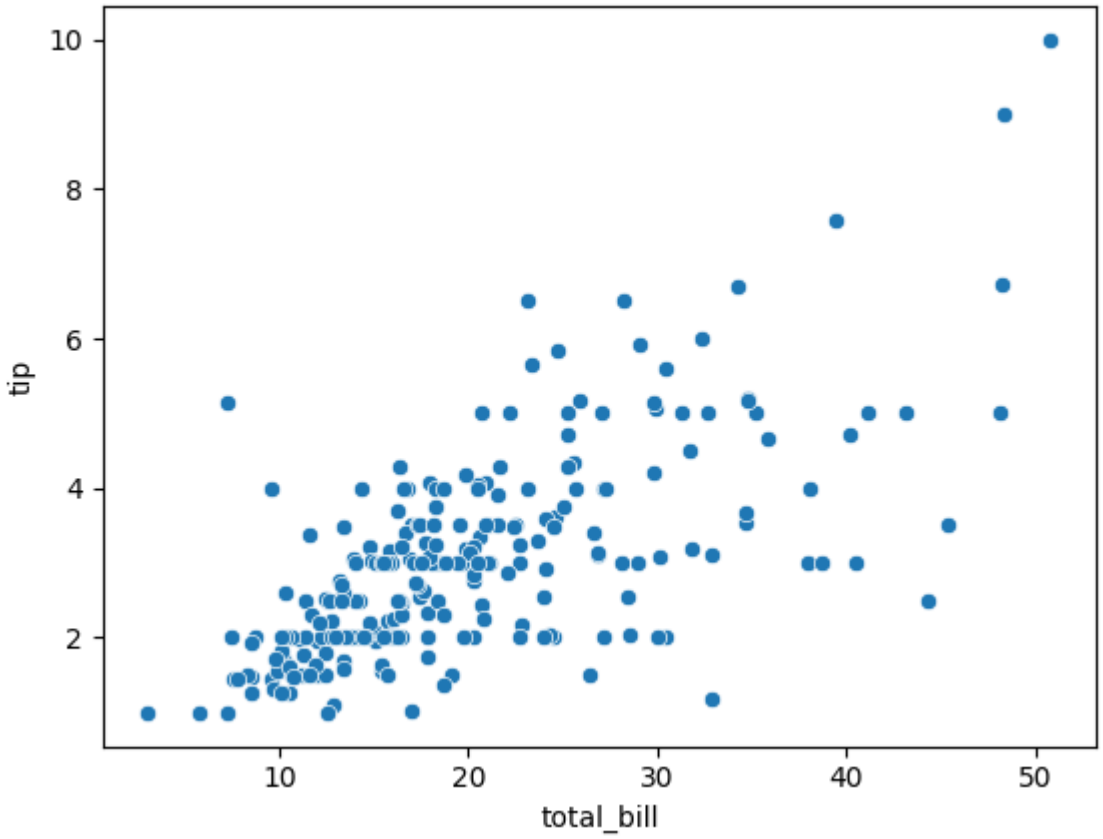
Η Seaborn είναι μια βιβλιοθήκη που εδράζεται στη Matplotlib και προσφέρει ένα υψηλότερου επιπέδου API για τη δημιουργία ελκυστικών και ενημερωτικών στατιστικών γραφικών. Είναι ιδανική για όσους/ες θέλουν να δημιουργήσουν πιο σύνθετες και αισθητικά ευχάριστες οπτικοποιήσεις χωρίς να χρειάζεται να γράφουν πολύ κώδικα.

Παρέχει μια πιο διαισθητική και συνοπτική σύνταξη σε σχέση με το Matplotlib, επιτρέποντάς να δημιουργήσουμε σύνθετα διαγράμματα με λίγες γραμμές κώδικα. Τα γραφήματα που δημιουργούνται με το Seaborn είναι γενικά πιο ελκυστικά και σύγχρονα, χάρη σε προκαθορισμένες παλέτες χρωμάτων και στυλ.

Η Seaborn ενσωματώνει την λειτουργικότητα του Pandas, επιτρέποντάς να δημιουργήσουμε διαγράμματα απευθείας από πλαίσια δεδομένων. Η βιβλιοθήκη είναι εξειδικευμένη στην οπτικοποίηση στατιστικών σχέσεων μεταξύ μεταβλητών.

Παράδειγμα:

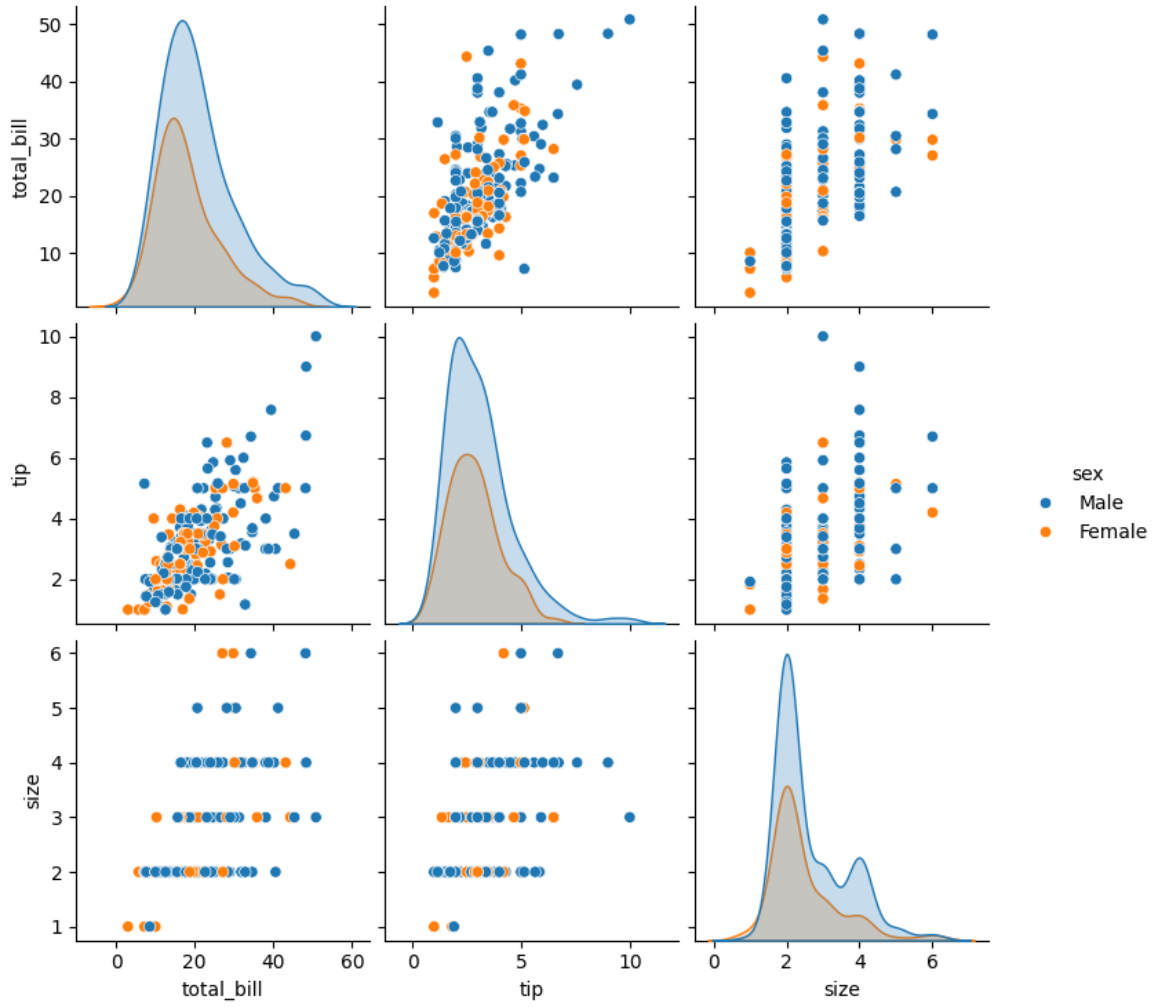
```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# Χρήση του ενσωματωμένου πλαισίου δεδομένων tips
tips = sns.load_dataset("tips")

# Διάγραμμα διασποράς
sns.scatterplot(x="total_bill", y="tip", data=tips)
plt.show()
```



Διάγραμμα 2.7: Διάγραμμα διασποράς με Seaborn

Εάν θέλουμε να δημιουργήσουμε ένα διάγραμμα συσχέτισης που θα μας δείχνει πως συσχετίζονται μεταξύ τους οι διαφορετικές στήλες σε ένα πλαίσιο δεδομένων τότε μπορούμε να χρησιμοποιήσουμε την μέθοδο `pairplot()`.

```
import seaborn as sns
import matplotlib.pyplot as plt
# Χρήση του πλαισίου δεδομένων tips
tips = sns.load_dataset("tips")
# Δημιουργία του pair plot
sns.pairplot(tips, hue="sex")
plt.show()
```



Διάγραμμα 2.8: Διάγραμμα συσχέτισης με Seaborn

Χρησιμοποιώντας το όρισμα `hue` στην μέθοδο `pairplot()` μπορούμε να χρωματίσουμε τα σημεία με βάση μια κατηγορική στήλη. Στο εν λόγω παράδειγμα χρησιμοποιείται η στήλη `sex` όπου χρωματίζονται με μπλε τα σημεία που αντιπροσωπεύουν τις γραμμές με την τιμή `Male` και με πορτοκαλί χρώμα τα σημεία που αντιπροσωπεύουν τις γραμμές με την τιμή `Female.`

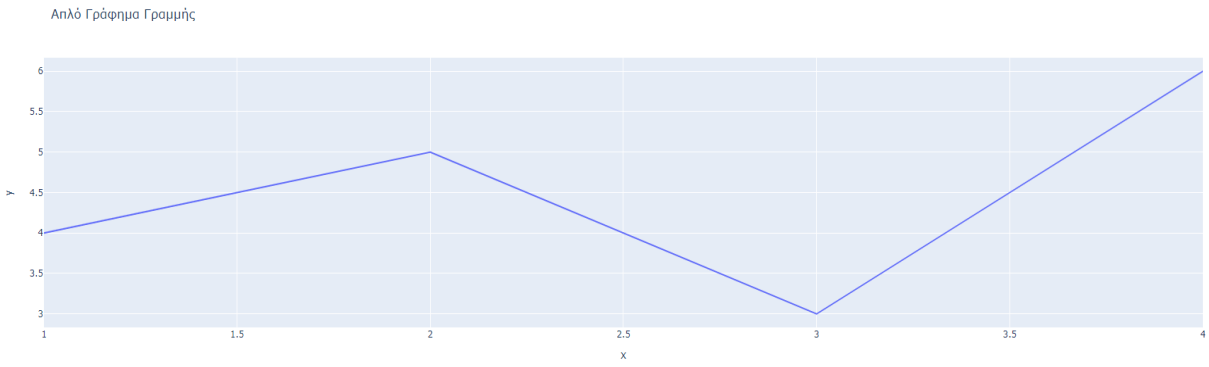
2.14. Βιβλιοθήκη Plotly (https://plotly.com/)

Η Plotly είναι μια δημοφιλής βιβλιοθήκη και πλατφόρμα που χρησιμοποιείται για τη δημιουργία διαδραστικών και υψηλής ποιότητας γραφημάτων και οπτικοποιήσεων δεδομένων. Είναι ένα εξαιρετικό εργαλείο για επιστήμονες δεδομένων, αναλυτές αλλά και για οποιονδήποτε επιθυμεί να παρουσιάσει τα δεδομένα του με έναν ελκυστικό και κατανοητό τρόπο.

Μπορούμε να δημιουργήσουμε απλά γραμμικά γραφήματα μέχρι σύνθετα τρισδιάστατα διαγράμματα. Τα γραφήματα που δημιουργούνται με το Plotly είναι διαδραστικά, επιτρέποντας στους χρήστες να τα μεγεθύνουν, να μετακινούνται και να εξερευνούν τα δεδομένα με λεπτομέρεια.

Παράδειγμα:

```
import plotly.express as px
import pandas as pd

# Δημιουργία ενός πλαισίου δεδομένων
df = pd.DataFrame({'x': [1, 2, 3, 4], 'y': [4, 5, 3, 6]})

# Δημιουργία του γραφήματος
fig = px.line(df, x='x', y='y', title='Απλό Γράφημα Γραμμής')

# Εμφάνιση του γραφήματος
fig.show()
```

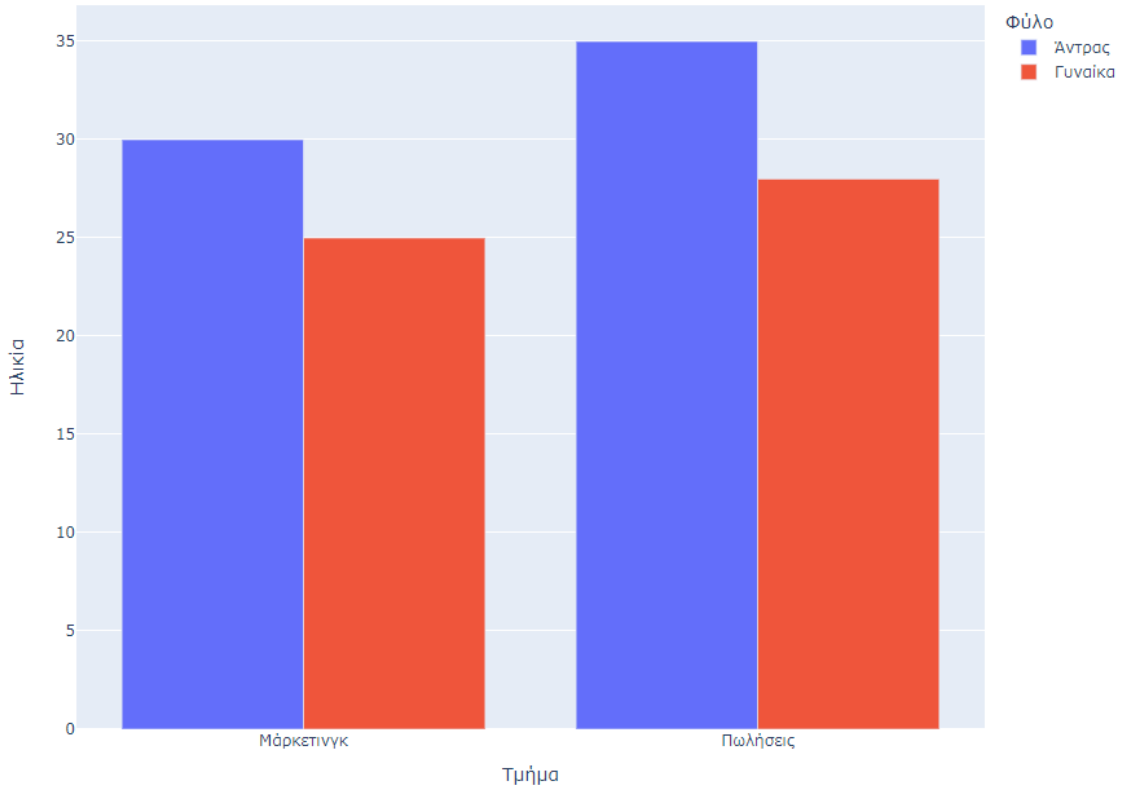


Διάγραμμα 2.9: Γραμμικό διάγραμμα με Plotly

Μπορούμε να δημιουργήσουμε και ένα σύνθετο ραβδόγραμμα (stacked ή grouped barchart) ως εξής:

```
import plotly.express as px
import pandas as pd
import plotly.graph_objects as go

# Δημιουργία ενός πλαισίου δεδομένων
```

```
df = pd.DataFrame({
    'Φύλο': ['Άντρας', 'Γυναίκα', 'Άντρας', 'Γυναίκα'],
    'Τμήμα': ['Μάρκετινγκ', 'Μάρκετινγκ', 'Πωλήσεις', 'Πωλήσεις'],
    'Ηλικία': [30, 25, 35, 28]
})

# Δημιουργία του grouped bar plot
fig = px.bar(df, x="Τμήμα", y="Ηλικία", color="Φύλο")


# Χρήση του ενός αντικειμένου go.Layout για την διαμόρφωση του
διαγράμματος

layout = go.Layout(autosize=False,
   barmode='group',
   width=900,
   height=700,
)


fig.update_layout(layout).show()
```



Διάγραμμα 2.10: Σύνθετο ραβδόγραμμα με Plotly

2.15. *Ερωτήσεις αυτοαξιολόγησης*

2.1 Ποια βιβλιοθήκη στην Python χρησιμοποιείται συνήθως για διαχείριση και ανάλυση αριθμητικών δεδομένων;
α) NumPy
β) Matplotlib
γ) Scikit-learn
δ) TensorFlow

2.2 Ποια βιβλιοθήκη στην Python χρησιμοποιείται συνήθως για οπτικοποίηση δεδομένων;
α) Pandas
β) Matplotlib
γ) Scikit-learn
δ) Keras

2.3 Ποια βιβλιοθήκη στην Python παρέχει λειτουργικότητα για εργασίες διερευνητικής ανάλυσης δεδομένων (EDA);
α) NumPy
β) Matplotlib
γ) Pandas
δ) Scikit-learn

2.4 Ποια από τις παρακάτω συναρτήσεις χρησιμοποιείται για την ανάγνωση ενός αρχείου CSV σε ένα Pandas DataFrame στην Python;
α) pd.load_csv()
β) pd.read_csv()
γ) pdf.load_file()
δ) pd.read_file()

2.5 Ποιο από τα παρακάτω διαγράμματα είναι κατάλληλο για την απεικόνιση της σχέσης μεταξύ δύο συνεχών μεταβλητών;
α) Διάγραμμα πίτας (Pie chart)
β) Ραβδογράφημα (Bar chart)
γ) Γραμμικό διάγραμμα (Line chart)
δ) Διάγραμμα διασποράς (Scatter plot)

2.6 Ποια από τις παρακάτω μεθόδους της NumPy χρησιμοποιείται για την εύρεση του μέγιστου στοιχείου σε έναν πίνακα;
α) np.min()
β) np.argmax()
γ) np.max()
δ) np.maxindex()

ΚΕΦΑΛΑΙΟ 3: Επισκόπηση βασικών στοιχείων θεωρίας πιθανοτήτων

3.1. Εισαγωγή και βασικά σημεία

Στον πραγματικό κόσμο παρατηρούνται φαινόμενα τα οποία σε πολλές περιπτώσεις δεν είναι εφικτό να προβλεφθούν εκ των προτέρων (όπως για παράδειγμα το αποτέλεσμα της ρίψης ενός ζαριού). Τα φαινόμενα μπορεί να οφείλονται σε ανθρώπινες ενέργειες ή στο φυσικό περιβάλλον και προκύπτουν ως αποτελέσματα διεργασιών, ή αιτιών, οι οποίες δεν είναι επαρκώς γνωστές. Η ελλιπής γνώση των διεργασιών οι οποίες οδηγούν στην εμφάνιση ενός συγκεκριμένου φαινομένου (για παράδειγμα η εμφάνιση της πλευράς με τον αριθμό έξι έπειτα από την απλή ρίψη ενός ζαριού με έξι πλευρές), έχει ως αποτέλεσμα την αδυναμία πρόβλεψης και κατά συνέπεια την αβεβαιότητα ως προς την εμφάνιση ενός συγκεκριμένου φαινομένου. Η πιθανότητα, ως έννοια, αποτελεί και χρησιμοποιείται ως το μέτρο του βαθμού αβεβαιότητας ενός αβέβαιου φαινομένου. Πέρα από το διαισθητικό χαρακτήρα της, γύρω από την έννοια της πιθανότητας έχει αναπτυχθεί είναι αρκετά σημαντικός κλάδος των μαθηματικών με σημαντική θεωρητική συνεισφορά και αρκετές περιοχές εφαρμογών, ειδικά στον χώρο της ανάλυσης δεδομένων και μηχανικής μάθησης. Η κατανόηση των αρχών της θεωρίας πιθανοτήτων αποτελεί αναγκαία γνώση για την μετέπειτα κατανόηση και ενασχόληση με μεθόδους μηχανικής μάθησης, αρκετές από τις οποίες βασίζονται ή χρησιμοποιούν μοντέλα πιθανοτήτων.

Στην τρέχουσα ενότητα θα παρουσιαστούν βασικές αρχές της θεωρίας πιθανοτήτων σε εισαγωγικό επίπεδο με τη βοήθεια παραδειγμάτων και ασκήσεων. Ο σκοπός της ενότητας είναι η κατανόηση των βασικών αρχών και θεωρημάτων και η εξοικείωση με επίλυση προβλημάτων πιθανοτήτων.

3.2. Πειράματα τύχης, δειγματικός χώρος και ενδεχόμενα.

Η πιθανότητα χρησιμοποιείται για την έκφραση με ποσοτικό τρόπο της εκτίμησης του βαθμού αβεβαιότητας ενός συγκεκριμένου αποτελέσματος. Αποτελεί, στην ουσία, μια αριθμητική τιμή η οποία εκτιμάται για κάθε δυνατό αποτέλεσμα, μεταξύ ενός συνόλου αποτελεσμάτων, εκφράζοντας τον βαθμό αβεβαιότητας εμφάνισής του κάθε αποτελέσματος. Αν και η πιθανότητα παρέχει μια εκτίμηση του βαθμού αβεβαιότητας, εν τούτοις, παραμένει εκτίμηση, η οποία μπορεί να μην επαληθευτεί στην πράξη. Δεν παρέχει με άλλα λόγια βεβαιότητα, αλλά ένδειξη.

Για παράδειγμα, αν ρίξουμε ένα τυπικό ζάρι με έξι πλευρές (όπου σε κάθε πλευρά εμφανίζεται ένας αριθμός από ένα έως και έξι), είναι αδύνατο να προβλέψουμε εκ των προτέρων το αποτέλεσμα (δηλαδή τον αριθμό που θα εμφανιστεί στην επάνω πλευρά). Διαισθητικά, αν δεν έχουμε κάποιο λόγο να αμφιβάλλουμε για την κατασκευή του ζαριού (ότι δηλαδή δεν υπάρχει κάποια ατέλεια στο ζάρι η οποία να οδηγεί στην εμφάνιση κάποιας συγκεκριμένης πλευράς, οπότε το ζάρι είναι αμερόληπτο), μπορούμε να θεωρήσουμε ότι και οι έξι πλευρές είναι εξίσου πιθανές να εμφανιστούν ως αποτέλεσμα της ρίψης. Οπότε, λαμβάνοντας υπόψη το γεγονός ότι έχουμε έξι πλευρές με ίσο βαθμό αβεβαιότητας, μπορούμε σχετικά εύλογα να ορίσουμε ως μέτρο της αβεβαιότητας, ή ως την πιθανότητα να εμφανιστεί μια συγκεκριμένη πλευρά (για παράδειγμα την πλευρά με τον αριθμό τρία), τον λόγο 1/6. Δηλαδή, ότι το επιθυμητό αποτέλεσμα (πλευρά με τον αριθμό τρία) αποτελεί το κλάσμα 1/6 του συνόλου των αποτελεσμάτων που είναι δυνατό να εμφανιστούν. Την ίδια πιθανότητα (1/6) θα έχει επίσης κάθε πλευρά του ζαριού (καθώς το ζάρι είναι αμερόληπτο). Η πιθανότητα 1/6 είναι επομένως μια αριθμητική τιμή, η οποία ανατίθεται σε ένα δυνατό αποτέλεσμα και εκφράζει τον βαθμό αβεβαιότητας εμφάνισής του κατά τη διεξαγωγή μιας διαδικασίας πειραματισμού.

Για την αυστηρότερη θεμελίωση των πιθανοτήτων, απαιτείται να οριστούν οι έννοιες του πειράματος τύχης, του δειγματικού χώρου και των ενδεχομένων. Οι έννοιες του δειγματικού χώρου και των ενδεχομένων εκφράζονται με μαθηματικό συμβολισμό συνόλων (καθώς αποτελούν σύνολα), αλλά μπορούν να παρασταθούν (σε απλές περιπτώσεις και για καλύτερη εποπτεία) και με τη χρήση διαγραμμάτων Venn.

3.2.1. Πείραμα τύχης

Ως πείραμα τύχης, ή τυχαίο πείραμα, μπορεί να οριστεί οποιαδήποτε διαδικασία, η εκτέλεση της οποίας οδηγεί σε κάποιο συγκεκριμένο αποτέλεσμα, όπου το αποτέλεσμα δεν είναι δυνατό να προβλεφθεί εκ των προτέρων. Θα πρέπει να σημειωθεί ότι σε ένα πείραμα τύχης θα πρέπει να υπάρχουν περισσότερα από ένα δυνατά αποτελέσματα, τα οποία θα πρέπει να είναι γνωστά και καθορισμένα εκ των προτέρων με ακρίβεια. Τα πειράματα τύχης δεν σχετίζονται αποκλειστικά με ρίψεις ζαριών ή κερμάτων (τα οποία αποτελούν συνηθισμένα εποπτικά παραδείγματα), αλλά εκτείνονται σε ολόκληρο το φάσμα της πραγματικού κόσμου. Επίσης, το πείραμα τύχης δεν απαιτεί την εκτέλεση κάποιας πειραματικής διεργασίας με τη στενή επιστημονική έννοια. Ακόμη και η μέτρηση ή καταγραφή μέσω παρατήρησης μπορεί να θεωρηθεί πείραμα τύχης καθώς οδηγεί σε εναλλακτικά αποτελέσματα.

Ενδεικτικά παραδείγματα πειραμάτων τύχης αποτελούν τα παρακάτω:

- Ρίψη ζαριών

- Το χρώμα του αυτοκινήτου το οποίο διέρχεται από ένα οδικό σημείο
- Η απάντηση σε μια ερώτηση δημοσκόπησης
- Το αποτέλεσμα του ελέγχου για το αν είναι ελαττωματικό ένα προϊόν από τη γραμμή παραγωγής

3.2.2. Δειγματικός χώρος

Σε ένα πείραμα τύχης το σύνολο των δυνατών αποτελεσμάτων ορίζεται ως ο δειγματικός χώρος του. Κάθε πείραμα τύχης έχει τον δικό του δειγματικό χώρο, ανάλογα με τις απαιτήσεις του προβλήματος, αλλά σε κάθε περίπτωση ο δειγματικός χώρος πρέπει να είναι σαφώς ορισμένος. Για παράδειγμα στο πείραμα τύχης 'ρίψη ζαριού μια φορά και καταγραφή της επάνω πλευράς' ο δειγματικός χώρος αποτελείται από σύνολο έξι στοιχείων τα οποία μπορούν να παρασταθούν είτε με αριθμητικές τιμές, ή με κουκίδες ή με άλλο τρόπο, αλλά σε κάθε περίπτωση θα αντιστοιχούν στο σύνολο των δυνατών εναλλακτικών αποτελεσμάτων.

3.2.2.1 Παράδειγμα

Θεωρούμε το πείραμα τύχης (ΠΤ1)

ΠΤ1: 'μια ρίψη ενός αμερόληπτου ζαριού με έξι πλευρές'

Ο δειγματικός χώρος (ΔΧ1) του ΠΤ1 είναι το σύνολο των δυνατών αποτελεσμάτων κατά την εκτέλεση του ΠΤ1 (δηλαδή οι έξι πλευρές).

ΔΧ1={1,2,3,4,5,6} ή {I,II,III,IV,V,VI} ή άλλη απεικόνιση

3.2.3. Ενδεχόμενο

Το κάθε δυνατό εναλλακτικό αποτέλεσμα ενός πειράματος τύχης (κάθε στοιχείο του δειγματικού χώρου), ονομάζεται απλό (ή στοιχειώδες) ενδεχόμενο. Επομένως, το σύνολο των απλών ενδεχομένων ενός πειράματος τύχης αποτελεί τον δειγματικό του χώρο. Ωστόσο, εφόσον ο δειγματικός χώρος αποτελεί σύνολο, υπάρχει η δυνατότητα να θεωρήσουμε και να εργαστούμε με υποσύνολά του (με περισσότερα του ενός ενδεχόμενα δηλαδή), τα οποία είναι επίσης ενδεχόμενα. Κάθε σύνολο (υποσύνολο του δειγματικού χώρου) το οποίο αποτελείται από απλά ενδεχόμενα ονομάζεται σύνθετο ενδεχόμενο. Οπότε, στη γενική περίπτωση ενδεχόμενο ονομάζεται κάθε υποσύνολο του δειγματικού χώρου ενός πειράματος τύχης.

3.2.3.1 Παράδειγμα

Στο ΠΤ1 με τον συγκεκριμένο ΔΧ1, ενδεικτικά εναλλακτικά ενδεχόμενα αποτελούν τα

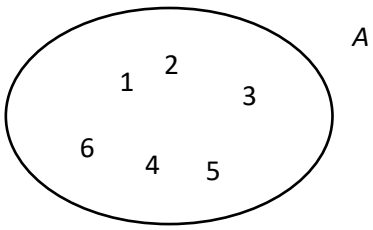
- Α1={Εμφάνιση οποιουδήποτε αριθμού στην επάνω πλευρά έπειτα από μια ρίψη του ζαριού}={1,2,3,4,5,6}

- Α2={Εμφάνιση μονού αριθμού στην επάνω πλευρά έπειτα από μια ρίψη του ζαριού}={1,3,5}
- Α3={Εμφάνιση ζυγού αριθμού στην επάνω πλευρά έπειτα από μια ρίψη του ζαριού}={2,4,6}
- Α4={Εμφάνιση αριθμού > 4 στην επάνω πλευρά έπειτα από μια ρίψη του ζαριού}={5,6}
- Α5={Εμφάνιση αριθμού <2 στην επάνω πλευρά έπειτα από μια ρίψη του ζαριού}={1}

Ένα ενδεχόμενο, όπως είναι εμφανές από τα παραδείγματα, αποτελεί υποσύνολο του δειγματικού χώρου, το οποίο με τη σειρά του αντιστοιχεί σε μια συνθήκη η οποία μπορεί να εκφραστεί με κάποια πρόταση. Σε ένα πείραμα τύχης μπορούμε να μελετήσουμε σε συγκεκριμένα ενδεχόμενα τα οποία μας ενδιαφέρουν ή στο σύνολό τους. Ένα ενδεχόμενο Α, θεωρούμε ότι πραγματοποιείται κατά την εκτέλεση ενός πειράματος τύχης, αν και μόνο αν το αποτέλεσμα του πειράματος αποτελεί στοιχείο του Α. Δηλαδή, το ενδεχόμενο Α3 πραγματοποιείται στο πείραμα ΠΤ1, αν το αποτέλεσμα ανήκει στο σύνολο Α3, δηλαδή ο αριθμός κατά τη ρίψη είναι ένας από τους 2,4,6.

Καθώς τα ενδεχόμενα αποτελούν σύνολα απλών ενδεχομένων, για τη μελέτη τους χρησιμοποιείται ο μαθηματικός συμβολισμός και η θεωρία συνόλων. Ένα σύνολο μπορεί να παρασταθεί είτε ως παράθεση των στοιχείων ή με τη βοήθεια ενός διαγράμματος Venn. Επίσης οι βασικές σχέσεις και πράξεις μεταξύ συνόλων ισχύουν και για τα ενδεχόμενα.

3.2.3.2 Παράδειγμα

Σύνολο	**Παράσταση ως παράθεση στοιχείων**	**Παράσταση με χρήση διαγράμματος Venn**
Τα δυνατά αποτελέσματα της ρίψης ενός ζαριού, δηλαδή οι αριθμοί 1,2,3,4,5,6	$A = \{1,2,3,4,5,6\}$	

3.3. Ορισμοί και αξιωματική θεμελίωση της Θεωρίας Πιθανοτήτων.

Αν και η έννοια της πιθανότητας δείχνει διαισθητικά εύλογη, δεν υπάρχει καθολικά αποδεκτός τρόπος υπολογισμού της για συγκεκριμένα ενδεχόμενα με βάση συγκεκριμένες διαδικασίες. Αυτό οφείλεται στις ερμηνευτικές προσεγγίσεις της πιθανότητας, είτε ως στατιστικό μέγεθος (όριο συχνότητας εμφάνισης), ή ως αξιωματικό μέγεθος (απαρίθμηση ενδεχομένων), ή ακόμη ως

υποκειμενικό μέγεθος (ανάλογο του πονταρίσματος σε λοταρία). Ως προς τον ορισμό της πιθανότητας, η προσέγγιση που προτάθηκε από τον Kolmogorov το 1933 είναι η επικρατούσα αξιωματική θεμελίωση και έχει οδηγήσει στην ανάπτυξη λογισμού πιθανοτήτων με τη χρήση της θεωρίας συνόλων, ωστόσο δεν παρέχει μέθοδο υπολογισμού της.

3.3.1. Αξιωματική θεμελίωση πιθανοτήτων κατά Kolmogorov

Για ένα δειγματικό χώρο Ω και κάθε ενδεχόμενο A του Ω, ορίζεται η συνάρτηση πιθανότητας $P(\cdot)$, ως μια συνάρτηση η οποία σε κάθε ενδεχόμενο A αντιστοιχεί έναν πραγματικό αριθμό $P(A)$ τέτοιον ώστε να ικανοποιούνται τα παρακάτω τρία αξιώματα

- $P(A) \geq 0$ για κάθε ενδεχόμενο του S
- $P(\Omega) = 1$
- Για πεπερασμένο δειγματικό χώρο, $P(A1 \cup A2 \cup \dots) = P(A1) + P(A2) + \cdots$εάν τα ενδεχόμενα $A1$, $A2$,…. είναι ξένα ανά δύο ενδεχόμενα (δύο ενδεχόμενα ονομάζονται ξένα μεταξύ τους αν δεν έχουν κοινά στοιχεία).

Με βάση τον παραπάνω ορισμό, η τιμή της συνάρτησης $P(A)$ για το ενδεχόμενο A, ονομάζεται πιθανότητα του ενδεχομένου A.

3.3.2. Κλασσική προσέγγιση του ορισμού της πιθανότητας (Laplace, 1812)

Σε έναν πεπερασμένο δειγματικό χώρο Ω όπου όλα τα απλά ενδεχόμενά του έχουν την ίδια πιθανότητα επιλογής (ισοπίθανα), ή δεν υπάρχει λόγος για να θεωρηθεί το αντίθετο, τότε η πιθανότητα να συμβεί το ενδεχόμενο A ορίζεται ως ο λόγος του αριθμού των ευνοϊκών περιπτώσεων προς τον συνολικό αριθμό περιπτώσεων, δηλαδή

$$P(A) = \frac{N(A)}{N(\Omega)} = \frac{\pi\lambda\acute{\eta}\theta o\varsigma\ \sigma\tau o\iota\chi\epsilon\acute{\iota}\omega\nu\ \tau o\upsilon\ A}{\pi\lambda\acute{\eta}\theta o\varsigma\ \sigma\tau o\iota\chi\epsilon\acute{\iota}\omega\nu\ \tau o\upsilon\ \Omega} = \frac{\pi\lambda\acute{\eta}\theta o\varsigma\ \epsilon\upsilon\nu o\ddot{\iota}\kappa\acute{\omega}\nu\ \pi\epsilon\rho\iota\pi\tau\acute{\omega}\sigma\epsilon\omega\nu}{\pi\lambda\acute{\eta}\theta o\varsigma\ \delta\upsilon\nu\alpha\tau\acute{\omega}\nu\ \pi\epsilon\rho\iota\pi\tau\acute{\omega}\sigma\epsilon\omega\nu}$$

3.3.3. Στατιστικός ορισμός πιθανότητας (von Misses, 1919)

Αν σε N επαναλήψεις ενός πειράματος τύχης ένα ενδεχόμενο A εμφανίσθηκε $N(A)$ φορές, τότε το πηλίκο

$$\frac{N(A)}{N}$$

ονομάζεται σχετική συχνότητα του ενδεχομένου A.

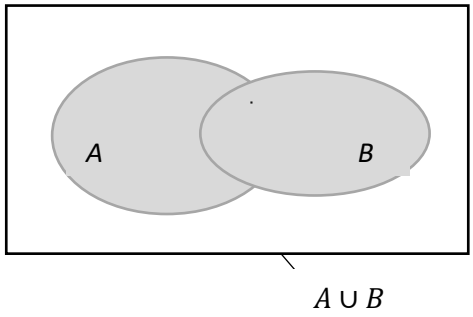
Με βάση τη στατιστική προσέγγιση, όσο ο αριθμός των επαναλήψεων του πειράματος (N) αυξάνεται, η σχετική συχνότητα του ενδεχομένου A σταθεροποιείται γύρω από έναν αριθμό ο οποίος προσεγγίζεται από το όριο της σχετικής συχνότητας για πολύ υψηλές τιμές του N. Η πιθανότητα του ενδεχομένου A ισούται με το όριο της σχετικής συχνότητας του ενδεχομένου A.

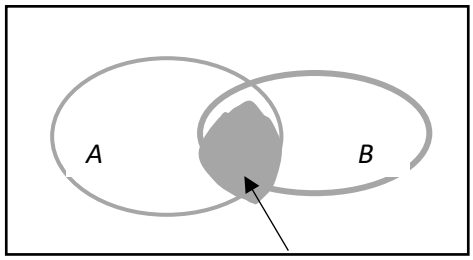
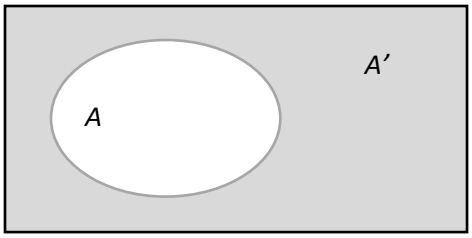
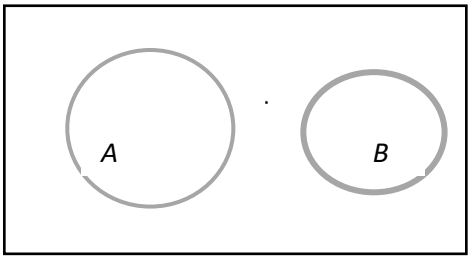
3.3.4. Υποκειμενικός ορισμός

Αν και δεν υπάρχει συγκεκριμένος ορισμός, οι υποστηρικτές της υποκειμενικής προσέγγισης θεωρούν ότι δεν μπορεί να οριστεί η πιθανότητα ενός ενδεχομένου με αντικειμενικό τρόπο, παρά μόνο ως άποψη του ατόμου το οποίο την εκτιμά. Μια προσέγγιση για τον υπολογισμό της αποτελούν υποθετικές λοταρίες όπου η υποκειμενική πιθανότητα για ένα ενδεχόμενο εκφράζεται με το πόσο θα ήταν διατεθειμένο το άτομο να στοιχηματίσει υπέρ του ενδεχομένου.

3.4. Βασικές σχέσεις και πράξεις συνόλων και ενδεχομένων.

Με βάση τον αξιωματικό ορισμό της πιθανότητας είναι εφικτή η έκφραση σχέσεων και η πραγματοποίηση λογικών πράξεων μεταξύ ενδεχομένων κατά αντιστοιχία με τις πράξεις μεταξύ συνόλων. Οι κυριότερες σχέσεις και πράξεις παρουσιάζονται παρακάτω. Θεωρείται ότι ο δειγματικός χώρος είναι το σύνολο Ω, και τα ενδεχόμενα αποτελούν υποσύνολά του.

Σχέση ενδεχομένων	**Ορολογία συνόλων**	**Παράσταση με χρήση διαγράμματος Venn**
Ένωση ενδεχομένων A και B : $A \cup B$	Το σύνολο $A \cup B$ είναι η ένωση των συνόλων A και B, δηλαδή αποτελείται από τα στοιχεία τα οποία ανήκουν ή στο σύνολο A ή στο σύνολο B ή και στα δύο.	

Τομή ενδεχομένων A και B : $A \cap B$	Το σύνολο $A \cap B$ είναι η τομή των συνόλων A και B, δηλαδή αποτελείται από τα στοιχεία τα οποία ανήκουν και στο σύνολο A και στο σύνολο B.	 $A \cap B$
Συμπλήρωμα του ενδεχομένου A: A' (ως προς ένα δειγματικό χώρο Ω)	Το σύνολο A' είναι το συμπλήρωμα του συνόλου A, δηλαδή αποτελείται από τα στοιχεία του Ω τα οποία δεν ανήκουν στο A.	
Ασυμβίβαστα ενδεχόμενα A και B (δεν έχουν κοινά στοιχεία) αν $A \cap B = \emptyset$	Αν $A \cap B = \emptyset$ τα σύνολα A και B είναι αμοιβαία αποκλειόμενα , δηλαδή δεν έχουν κοινά στοιχεία.	 $A \cap B = \emptyset$

3.5. Ιδιότητες πιθανοτήτων

Με βάση την αξιωματική θεμελίωση των πιθανοτήτων αποδεικνύεται μια σειρά θεωρημάτων από τα οποία απορρέουν αρκετές χρήσιμες ιδιότητες των πιθανοτήτων οι οποίες χρησιμοποιούνται για την εκτέλεση υπολογισμών σε πιθανότητες ενδεχομένων. Οι βασικές ιδιότητες των πιθανοτήτων, οι οποίες απορρέουν από τα θεωρήματα, είναι οι εξής.

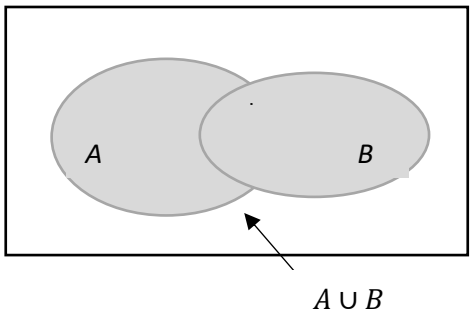
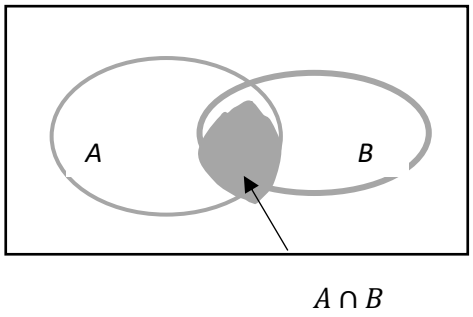
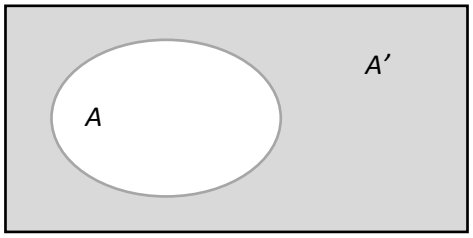
I. Για κάθε ενδεχόμενο A η πιθανότητα ορίζεται στο διάστημα $[0,1]$, δηλαδή $0 \leq P(A) \leq 1$
II. $P(\emptyset) = 0$, (το κενό ενδεχόμενο).
III. $P(\Omega)$=1, (ο δειγματικός χώρος).
IV. $P(A') = 1 - P(A)$, (συμπληρωματικά ενδεχόμενα).

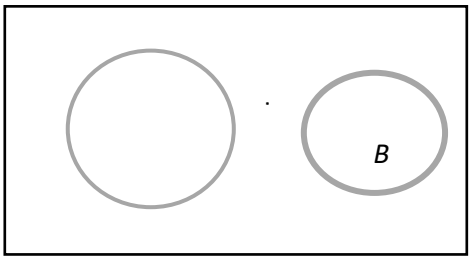
V. $Αν\ A \subseteq B\ τότε\ P(A) \leq P(B)$, (ενδεχόμενο υποσύνολο άλλου ενδεχόμενου)

VI. $P(A \cup B) = P(A) + P(B) - P(A \cap B)$, (νόμος της πρόσθεσης)

VII. $P(A \cup B \cup Γ) = P(A) + P(B) + P(Γ) - P(A \cap B) - P(A \cap Γ) - P(B \cap Γ) + P(A \cap B \cap Γ)$, (νόμος της πρόσθεσης γενίκευση)

Συνδυάζοντας την απεικόνιση με διαγράμματα Venn, η ερμηνεία των σχέσεων ενδεχομένων και η σύνδεση με τις πιθανότητες γίνεται ευκρινέστερη.

Σχέση ενδεχομένων	**Παράσταση με χρήση διαγράμματος Venn**	**Πιθανότητα**
Ένωση ενδεχομένων A και B : $A \cup B$		Η πιθανότητα της ένωσης των ενδεχομένων A και B, $P(A \cup B)$ είναι η πιθανότητα εμφάνισης είτε του ενδεχομένου A, ή του B, ή και των δύο. Ισχύει ότι $P(A \cup B) = P(A) + P(B) - P(A \cap B)$, (νόμος της πρόσθεσης)
Τομή ενδεχομένων A και B : $A \cap B$		Η πιθανότητα της τομής των ενδεχομένων A και B, $P(A \cap B)$ είναι η πιθανότητα εμφάνισης και του ενδεχομένου A και του B ταυτόχρονα.
Συμπλήρωμα του ενδεχομένου A: A' (ως προς ένα δειγματικό χώρο $Ω$)		Η πιθανότητα του A' είναι η πιθανότητα της μη εμφάνισης του ενδεχομένου A, ως συμπληρωματικά ενδεχόμενα $P(A') = 1 - P(A)$

Ασυμβίβαστα ενδεχόμενα A και B (δεν έχουν κοινά στοιχεία) αν $A \cap B = \emptyset$	 $A \cap B = \emptyset$	Αν τα A και B είναι ασυμβίβαστα ενδεχόμενα, τότε τα ενδεχόμενα A και B, δεν μπορούν να εμφανιστούν ταυτόχρονα. Ισχύει ότι $P(A \cap B) = 0$ $P(A \cup B) = P(A) + P(B)$

Διάγραμμα 3.1: Διαγράμματα Venn

Με βάση τις ιδιότητες είναι εφικτός ο υπολογισμός πιθανοτήτων για ενδεχόμενα σε πειράματα τύχης, ωστόσο σε πολλές περιπτώσεις για τον υπολογισμό των πιθανοτήτων απλών ενδεχομένων γίνεται χρήση του κλασσικού ορισμού, οπότε είναι πολύ σημαντική η δυνατότητα απαρίθμησης των δυνατών και ευνοϊκών περιπτώσεων.

3.5.1. Παράδειγμα 1

Εκτελείται ρίψη ενός νομίσματος τρεις διαδοχικές φορές και το αποτέλεσμα θα είναι είτε Κορώνα (Κ), ή Γράμματα (Γ).

- Να οριστεί ο δειγματικός χώρος Ω του πειράματος.
- Να οριστούν τα ενδεχόμενα που προσδιορίζονται από την αντίστοιχη ιδιότητα:
  - Α1: "Ο αριθμός των Κ υπερβαίνει τον αριθμό των Γ"
  - Α2: "Ο αριθμός των Κ είναι ακριβώς 2"
  - Α3: "Ο αριθμός των Κ είναι τουλάχιστον 2"
  - Α4: "Εμφανίζεται ίδια όψη και στις τρεις ρίψεις"
  - Α5: "Στην πρώτη ρίψη εμφανίζεται Κ".
- Να οριστούν τα ενδεχόμενα $A_3^{'}, A_5 \cap A_2, A_5 \cup A_4$.

Επίλυση

Ο δειγματικός χώρος του πειράματος αποτελείται από διατεταγμένες τριάδες με στοιχεία το *Κ* και το *Γ* και είναι

$$\Omega = \{ΚΚΚ, ΚΚΓ, ΚΓΚ, ΚΓΓ, ΓΚΚ, ΓΚΓ, ΓΓΚ, ΓΓΓ\}.$$

Με βάση τον δειγματικό χώρο *Ω* τα ενδεχόμενα ορίζονται ως:

- $A_1 = \{ΚΚΚ, ΚΚΓ, ΚΓΚ, ΓΚΚ\}$
- $A_2 = \{ΚΚΓ, ΚΓΚ, ΓΚΚ\}$
- $A_3 = \{ΚΚΚ, ΚΚΓ, ΚΓΚ, ΓΚΚ\}$

- $A_4 = \{ΚΚΚ, ΓΓΓ\}$
- $A_5 = \{ΚΚΚ, ΚΓΓ, ΚΓΚ, ΚΓΓ\}$.
- $A_3^{'} = \{ΚΓΓ, ΓΚΓ, ΓΓΚ, ΓΓΓ\}$. Το A_3' περιέχει τα στοιχεία του δειγματικού χώρου στα οποία ο αριθμός των *Κ* είναι μικρότερος από 2.
- $A_5 \cap A_2 = \{ΚΚΓ, ΚΓΚ\}$. Τα κοινά στοιχεία A_5 *και* A_2.
- $A_5 \cup A_4 = \{ΚΚΓ, ΚΓΚ, ΚΚΓ, ΚΚΚ, ΓΓΓ\}$.

3.5.2. Παράδειγμα 2

Εκτελείται το πείραμα 'Ρίψη ενός ζαριού δύο φορές'. Να υπολογιστεί η πιθανότητα το αποτέλεσμα να περιλαμβάνει δύο διαδοχικούς αριθμούς, με τον πρώτο αριθμό να είναι μικρότερος του δεύτερου.

Επίλυση

Ο δειγματικός χώρος του πειράματος αποτελείται από το σύνολο των συνδυασμών των αποτελεσμάτων των δύο ρίψεων.

$2^η$ ρίψη $1^η$ ρίψη	1	2	3	4	5	6
1	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)
2	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)
3	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)
4	(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(4,6)
5	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)
6	(6,1)	(6,2)	(6,3)	(6,4)	(6,5)	(6,6)

Ο δειγματικός χώρος *Ω* αποτελείται από 36 ισοπίθανα δυνατά αποτελέσματα, $N(\Omega) = 36$.

Το ενδεχόμενο *Α*: "το αποτέλεσμα να περιλαμβάνει δύο διαδοχικούς αριθμούς με τον πρώτο μικρότερο του δεύτερου", είναι το υποσύνολο του δειγματικού χώρου με τους διαδοχικούς αριθμούς

$$A = \{(1,2), (2,3), (3,4), (4,5), (5,6)\}$$

με $N(A) = 5$

Επομένως, $P(A) = \frac{N(A)}{N(\Omega)} = \frac{5}{36}$.

Άρα, η πιθανότητα εμφάνισης του ενδεχομένου "το αποτέλεσμα να περιλαμβάνει δύο διαδοχικούς αριθμούς με τον πρώτο μικρότερο του δεύτερου " είναι ίση με$\frac{5}{36}$.

3.6. Βασικά στοιχεία απαρίθμησης και συνδυαστικής

Για τον υπολογισμό των πιθανοτήτων είναι αναγκαία πολλές φορές η απαρίθμηση των ενδεχομένων. Σε απλές περιπτώσεις η απαρίθμηση είναι εύκολη, ωστόσο σε σύνθετα προβλήματα απαιτεί τεχνικές οι οποίες προέρχονται από τον κλάδο της συνδυαστικής ανάλυσης και επιτρέπουν τον υπολογισμό του πλήθους ενδεχομένων σε σύντομο χρόνο.

Οι βασικές τεχνικές είναι οι εξής.

3.6.1. Πολλαπλασιαστική αρχή.

Αν μια σύνθετη ενέργεια A απαρτίζεται από την εκτέλεση επιμέρους ενεργειών $A1, A2, A3, \dots An$, τότε το πλήθος των δυνατών αποτελεσμάτων της A υπολογίζεται συνδυάζοντας κάθε αποτέλεσμα των επιμέρους ενεργειών με τις υπόλοιπες. Οπότε το πλήθος των δυνατών αποτελεσμάτων της A υπολογίζεται ως το γινόμενο των δυνατών αποτελεσμάτων επιμέρους ενεργειών $A1, A2, A3, \dots An$. Αναλυτικότερα, αν το στοιχείο $A1$ μπορεί να επιλεγεί με $\nu 1$ διαφορετικούς τρόπους και για κάθε επιλογή του $A1$, το στοιχείο $A2$ μπορεί να επιλεγεί με $\nu 2$ διαφορετικούς τρόπους, …. και για κάθε επιλογή των στοιχείων $A1, A2, \dots, A(\kappa - 1)$, το στοιχείο $A\kappa$ μπορεί να επιλεγεί με $\nu\kappa$ διαφορετικούς τρόπους, τότε όλα τα στοιχεία $A1, A2, \dots, A(\kappa - 1), A\kappa$, μπορούν να επιλεγούν διαδοχικά και με αυτή τη συγκεκριμένη σειρά κατά $\nu 1 \cdot \nu 2 \cdot \dots \cdot \nu\kappa$ τρόπους.

3.6.2. Διατάξεις και μεταθέσεις.

Σε ορισμένα προβλήματα το ενδιαφέρον εστιάζει στην απαρίθμηση τμήματος του συνόλου. Σε αυτή την περίπτωση χρησιμοποιούνται οι διατάξεις, δηλαδή τρόποι με τους οποίους μπορεί να γίνει επιλογή ν στοιχείων από ένα σύνολο κ στοιχείων. Σε μια διάταξη η σειρά εμφάνισης των στοιχείων έχει σημασία, οπότε δύο επιλογές διαφέρουν όταν τα ίδια στοιχεία βρίσκονται σε διαφορετική σειρά εμφάνισης. Ο τύπος για τον υπολογισμό των διατάξεων είναι ο εξής.

Όταν υπάρχουν ν διαφορετικά στοιχεία και τοποθετούνται στη σειρά κ από αυτά, δημιουργείται μια διάταξη των ν στοιχείων ανά κ. Το πλήθος όλων των διαφορετικών διατάξεων των ν στοιχείων ανά κ συμβολίζεται με $\nu(\kappa)$ ή ${}_{\nu}P_{\kappa}$ ή $P(\nu, \kappa)$ και είναι ίσο με

$$\nu(\kappa) = {}_{\nu}P_{\kappa} \text{ ή } P(\nu, \kappa) = \frac{\nu!}{(\nu - \kappa)!}$$

όπου $\nu! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot \nu,\ 0! = 1,$

Όταν $\kappa = \nu$ έχουμε τις μεταθέσεις των ν στοιχείων, των οποίων το πλήθος είναι ίσο με

$$\nu(\nu) = {}_{\nu}P_{\nu} \text{ ή } P(\nu,\nu) = \frac{\nu!}{(\nu-\nu)!} = \nu!$$

3.6.3. Επαναληπτικές διατάξεις

Σε αυτή την περίπτωση από τα ν στοιχεία επιλέγονται κ, αλλά κάθε ένα από αυτά μπορεί να επιλεγεί περισσότερες από μια φορές. Οι επαναληπτικές διατάξεις των ν στοιχείων ανά κ είναι ίσες με

$$\nu \cdot \nu \cdot \ldots \cdot \nu = \nu^{\kappa}$$

3.6.4. Μεταθέσεις με όμοια στοιχεία.

Σε αυτή την περίπτωση τα ν στοιχεία δεν είναι όλα διαφορετικά μεταξύ τους, αλλά υπάρχουν κάποια όμοια στοιχεία. Εάν τα $\nu 1$ είναι ενός είδους $A1$, τα $\nu 2$ είναι ενός άλλου είδους $A2$, ….. και τα $\nu\kappa$ είναι κάποιου άλλου είδους $A\kappa$, όπου $\nu 1 + \nu 2 + \cdots \nu\kappa = \nu$, τότε οι διαφορετικές μεταθέσεις των ν στοιχείων είναι:

$$\binom{\nu}{\nu 1, \nu 2, \ldots, \nu\kappa} = \frac{\nu!}{\nu 1! \cdot \nu 2! \cdot \ldots \cdot \nu\kappa!}$$

3.6.5. Συνδυασμοί.

Σε ορισμένα προβλήματα το ενδιαφέρον εστιάζει στην απαρίθμηση τμήματος του συνόλου, δηλαδή τρόποι με τους οποίους μπορεί να γίνει επιλογή ν στοιχείων από ένα σύνολο κ στοιχείων, χωρίς ωστόσο να έχει σημασία η σειρά εμφάνισης των στοιχείων. Οπότε δύο επιλογές δεν διαφέρουν όταν τα ίδια στοιχεία βρίσκονται σε διαφορετική σειρά εμφάνισης, αλλά μόνο όταν περιέχουν διαφορετικά στοιχεία. Στην περίπτωση αυτή υπολογίζονται οι δυνατοί συνδυασμοί των ν στοιχείων ανά κ. Ο τύπος για τον υπολογισμό των συνδυασμών είναι ο εξής.

Το πλήθος όλων των διαφορετικών συνδυασμών των ν στοιχείων ανά κ συμβολίζεται με $\binom{\nu}{\kappa}$ ή ${}_{\nu}C_{\kappa}$ ή $C(\nu,\kappa)$ και είναι ίσο με

$$\binom{\nu}{\kappa} = {}_{\nu}C_{\kappa} \text{ ή } C(\nu,\kappa) = \frac{\nu!}{\kappa!\,(\nu-\kappa)!}$$

3.6.6. Δειγματοληψία

Σε περιπτώσεις όπου πραγματοποιείται επιλογή δείγματος από σύνολο υπάρχουν οι επιλογές

- Δειγματοληψία με επανάθεση, όπου γίνεται λήψη ενός στοιχείου, εξέταση και επανατοποθέτηση πριν τη λήψη του επόμενου στοιχείου. Σε σύνολο με ν στοιχεία, για επιλογή κ στοιχείων υπάρχουν ν^{κ} τέτοια διαφορετικά δείγματα.

- Δειγματοληψία χωρίς επανάθεση, όπου γίνεται λήψη ενός στοιχείου, εξέταση και όχι επανατοποθέτηση πριν τη λήψη του επόμενου στοιχείου. Σε σύνολο με ν στοιχεία, για επιλογή κ στοιχείων υπάρχουν $\nu(\kappa)$ τέτοια διαφορετικά δείγματα.
- Ταυτόχρονη λήψη κ στοιχείων από σύνολο ν στοιχείων. Σε αυτή την περίπτωση ενδιαφέρει μόνο ποια στοιχεία επιλέχτηκαν. Σε αυτή την περίπτωση υπάρχουν $\binom{\nu}{\kappa}$ δείγματα.

Ως λήψη τυχαίου δείγματος μεγέθους κ εννοείται ότι η δειγματοληψία γίνεται με τέτοιο τρόπο, ώστε όλα τα δείγματα μεγέθους κ έχουν την ίδια πιθανότητα επιλογής.

3.6.7. Παράδειγμα 1

Αν μια επιτροπή 5 μελών ενός διοικητικού συμβουλίου συνεδριάζει για να εκλέξει πρόεδρο, γραμματέα, και ταμία, ποιο είναι το πλήθος των διαφορετικών τριάδων που θα εκλεγούν για τις τρεις θέσεις από τα 5 μέλη;

Επίλυση

Η διαδικασία εκλογής μπορεί να χωριστεί σε τρεις φάσεις.

- Η εκλογή προέδρου μπορεί να γίνει με 5 τρόπους, όσα είναι και τα μέλη της επιτροπής.
- Η εκλογή γραμματέα φάση μπορεί να γίνει με 4 τρόπους, όσα είναι και τα μέλη της επιτροπής που απέμειναν ύστερα από την εκλογή του προέδρου.
- Η εκλογή ταμία μπορεί να γίνει με 3 τρόπους, όσα είναι και τα μέλη της επιτροπής που απέμειναν ύστερα και από την εκλογή του ταμία.

Επομένως, σύμφωνα με την πολλαπλασιαστική αρχή απαρίθμησης, το πλήθος των διαφορετικών δυνατών τριάδων είναι $5 \cdot 4 \cdot 3 = 60$.

Καθεμιά από τις παραπάνω τριάδες ονομάζεται **διάταξη** των 5 ανά 3.

3.6.8. Παράδειγμα 2

Σε ένα τυχερό παίγνιο (bingo), σε κάθε επανάληψη κληρώνονται, ως νικήτρια στήλη με τυχαίο τρόπο 10 αριθμοί από σύνολο 50. Αν ένας παίκτης για μια επανάληψη του παιγνίου έχει το δικαίωμα να επιλέξει 10 τυχαίους από τους 50, ποια είναι η πιθανότητα να επιτύχει τους 6 από 10 της νικήτριας στήλης;

Επίλυση

Επειδή δεν έχει σημασία η σειρά επιλογής, οι δυνατές περιπτώσεις του πειράματος είναι τόσες όσοι και οι συνδυασμοί των 50 ανά 10, δηλαδή $N(\Omega) = \binom{50}{10}$.

Για το πλήθος των ευνοϊκών περιπτώσεων ισχύει ότι υπάρχουν $\binom{10}{6}$ τρόποι για να επιλεγούν οι 6 από τους 10 αριθμούς της νικήτριας στήλης σωστά. Στη συνέχεια μένουν $\binom{50-10}{10-6} = \binom{40}{4}$ τρόποι για την επιλογή των 4 λάθος αριθμών. Επομένως, το πλήθος των ευνοϊκών περιπτώσεων είναι $N(A) = \binom{10}{6} \cdot \binom{40}{4}$.

Επομένως,

$$P(A) = \frac{\binom{10}{6}\binom{40}{4}}{\binom{50}{10}} = \frac{\frac{10!40!}{6!(10-6)!4!(40-4)!}}{\frac{50!}{10!(50-10)!}} = \frac{\frac{10!40!}{6!(10-6)!4!(40-4)!}}{\frac{50!}{10!(50-10)!}} = 0{,}00585 = 0{,}6\%$$

3.7. Δεσμευμένη (υπό συνθήκη) πιθανότητα.

Σε ορισμένες περιπτώσεις η εκτέλεση ενός πειράματος τύχης ενδέχεται να οδηγήσει σε πληροφορίες οι οποίες μπορούν να αξιοποιηθούν για τον ακριβέστερο υπολογισμό της πιθανότητας εμφάνισης ενός ενδεχομένου με βάση τα νέα δεδομένα. Σε ένα πείραμα τύχης με δειγματικό χώρο Ω η πιθανότητα εμφάνισης ενός ενδεχομένου A με το δεδομένο ότι ένα άλλο ενδεχόμενο B έχει ήδη εμφανιστεί ονομάζεται δεσμευμένη πιθανότητα και ορίζεται ως εξής

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Η πιθανότητα να εμφανιστεί το ενδεχόμενο A δοθέντος του B υπολογίζεται σε υποσύνολο του αρχικού δειγματικού χώρου, και για τον λόγο αυτό θεωρείται ότι η επιπλέον πληροφορία, όταν τα ενδεχόμενα δεν είναι ασυμβίβαστα, επιτρέπει μείωση της αβεβαιότητας.

3.8. Απλή, από κοινού πιθανότητα και περιθώρια πιθανότητα.

Η δεσμευμένη πιθανότητα επιτρέπει τον υπολογισμό της από κοινού πιθανότητας δύο ενδεχομένων. Οπότε για τα ενδεχόμενα A και B ισχύει ότι

$$P(A \cap B) = P(B)P(A|B)$$

και

$$P(A \cap B) = P(A)P(B|A)$$

3.9. Στοχαστική ανεξαρτησία και πολλαπλασιαστικός κανόνας.

Από τον ορισμό της δεσμευμένης πιθανότητας προκύπτει ότι για τα ενδεχόμενα A και B ισχύει ο κανόνας που ονομάζεται πολλαπλασιαστικός και μπορεί να επεκταθεί σε ν ενδεχόμενα.

$$P(A \cap B) = P(B)P(A|B)$$

Ωστόσο, υπάρχουν περιπτώσεις όπου η πληροφορία ότι το ένα ενδεχόμενο συνέβη δεν μεταβάλλει τη δεσμευμένη πιθανότητα να συμβεί το άλλο ($P(B) = P(A|B)$).

Σε αυτή την περίπτωση δύο ενδεχόμενα A και B ονομάζονται ανεξάρτητα αν και μόνο αν ισχύει ότι

$$P(A \cap B) = P(A) \cdot P(B)$$

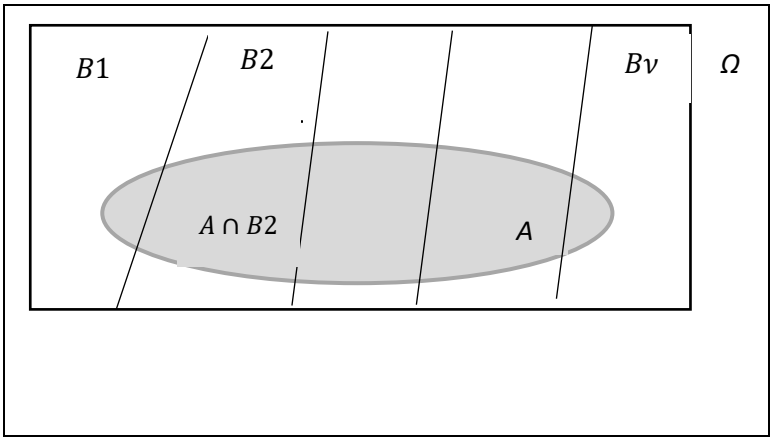
3.10. Θεώρημα της ολικής πιθανότητας

Το θεώρημα της ολικής πιθανότητας επιτρέπει τον υπολογισμό της πιθανότητα ενός ενδεχομένου A ως συνάρτηση των δεσμευμένων πιθανοτήτων του σε σχέση με τα στοιχεία μιας διαμέρισης του δειγματικού χώρου και των πιθανοτήτων των στοιχείων της διαμέρισης.

Ειδικότερα, έστω ένας δειγματικός χώρος Ω και ένα σύνολο ν ασυμβίβαστων μεταξύ τους ενδεχομένων $\{B1, B2, \dots, B\nu\}$ τα οποία καλύπτουν το σύνολο του δειγματικού χώρου (επομένως αποτελούν μια διαμέριση του χώρου, με $B1 \cup B2 \cup \dots \cup B\nu = \Omega \; και \; Bi \cap Bj = \emptyset$), τότε για κάθε ενδεχόμενο A στον Ω έχουμε ότι

$$P(A) = \sum_{i=1}^{\nu} P(B_i)P(A|B_i)$$

Στο παρακάτω διάγραμμα 3.2 Venn απεικονίζεται γραφικά η βασική ιδέα του θεωρήματος.



Διάγραμμα3.2: Venn diagram

3.10.1. Παράδειγμα 1

Σε μια επιχείρηση γίνεται εκλογή αντιπροσώπου για το διοικητικό συμβούλιο. Οι υποψήφιοι είναι 8 άνδρες και 9 γυναίκες. Από τους υποψηφίους, 4 άνδρες και 5 γυναίκες είναι μέσοι διοικητικοί υπάλληλοι, ενώ 4 άνδρες και 4 γυναίκες είναι ανώτεροι διοικητικοί υπάλληλοι. Οι υποψήφιοι μπορούν να ταξινομηθούν στον ακόλουθο πίνακα ως εξής:

Πίνακα 3.2. Πίνακας δεδομένων

	Μέσοι διοικητικοί	Ανώτεροι διοικητικοί	Σύνολο
Άνδρας	4	4	8
Γυναίκα	5	4	9
Σύνολο	9	8	17

Να υπολογιστούν

- η πιθανότητα να εκλεγεί μέσος διοικητικός υπάλληλος,
- η πιθανότητα να εκλεγεί γυναίκα
- η πιθανότητα να εκλεγεί μέσος διοικητικός υπάλληλος με δεδομένο ότι έχει ήδη εκλεγεί γυναίκα

Επίλυση

Ορίζονται τα ενδεχόμενα:

Α: “Μέσος διοικητικός υπάλληλος”

Β: “Γυναίκα”

Γ: “Μέσος διοικητικός υπάλληλος με δεδομένο ότι είναι γυναίκα”

Τα 17 στοιχεία του δειγματικού χώρου είναι ισοπίθανα, οπότε $P(A) = \frac{9}{17}$, $P(B) = \frac{9}{17}$

$$P(\Gamma) = P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{\frac{5}{15}}{\frac{9}{15}} = \frac{5}{9}$$

3.10.2. Παράδειγμα 2

Μια μονάδα παραγωγής διαθέτει τρεις όμοιες μηχανές (Α, Β, Γ). Η κάθε μηχανή παράγει ένα τμήμα της ημερήσιας παραγωγής (παρτίδα προϊόντων). Η πρώτη (Α) παράγει το 20%, η δεύτερη (Β) το 30% και η τρίτη (Γ) το 50% της συνολικής ημερήσιας παραγωγής. Έχει σημειωθεί κατά τον ποιοτικό έλεγχο ότι το 5% της παραγωγής της μηχανής Α, το 4% της Β και το 2% της Γ είναι μη αποδεκτά ποιοτικά.

- Αν επιλεχθεί τυχαία ένα προϊόν σε ένα κατάστημα πωλήσεων ποια είναι η πιθανότητα να είναι μη αποδεκτό ποιοτικά;
- Αν ένα προϊόν που επιλέχθηκε τυχαία είναι μη αποδεκτό ποιοτικά, ποια είναι η πιθανότητα να προέρχεται από τη μηχανή Α;

<u>Επίλυση</u>

Αν A, B, Γ είναι τα ενδεχόμενα το επιλεγμένο προϊόν να προέρχεται από τις μηχανές Α, Β και Γ αντιστοίχως, τότε $P(A) = 0{,}2$, $P(B) = 0{,}3$ και $P(\Gamma) = 0{,}5$ και επιπλέον $A \cup B \cup \Gamma = \Omega$.

Αν Ε είναι το ενδεχόμενο το επιλεγμένο εξάρτημα να είναι ελαττωματικό, τότε

$E = (E \cap A) \cup (E \cap B) \cup (E \cap \Gamma)$

$P(E) = (E \cap A) \cup (E \cap B) \cup (E \cap \Gamma)$.

Από τα δεδομένα του προβλήματος: $P(E|A) = 0{,}05$, $P(E|B) = 0{,}04$ και $P(E|\Gamma) = 0{,}02$.

Επομένως:

Η πιθανότητα να είναι μη αποδεκτό ποιοτικά το προϊόν (από τον πολλαπλασιαστικό νόμο των πιθανοτήτων) είναι

$P(E) = P(A) \cdot P(E|A) + P(B) \cdot P(E|B) + P(\Gamma) \cdot P(E|\Gamma)$

$= 0{,}2 \cdot 0{,}05 + 0{,}3 \cdot 0{,}04 + 0{,}5 \cdot 0{,}02 = 0{,}032.$

Η πιθανότητα ένα μη αποδεκτό ποιοτικά προϊόν να προέρχεται από τη μηχανή Α είναι $P(A|E)$. Έχουμε

$$P(A|E) = \frac{P(A \cap E)}{P(E)} = \frac{P(E|A) \cdot P(A)}{P(E)} = \frac{0{,}2 \cdot 0{,}05}{0{,}032} = \frac{0{,}01}{0{,}032} = 0{,}3125,$$

δηλαδή, $31{,}25\%$.

3.11. Θεώρημα του Bayes.

Το θεώρημα του Bayes χρησιμοποιεί το θεώρημα της ολικής πιθανότητας και παρέχει τη δυνατότητα υπολογισμού της δεσμευμένης πιθανότητας για κάθε στοιχείο της διαμέρισης ενός δειγματικού χώρου σε σχέση με ένα ενδεχόμενο, αντιστρέφοντας τον υπολογισμό των πιθανοτήτων σε σχέση με το θεώρημα της ολικής πιθανότητας.

Ειδικότερα, έστω ένας δειγματικός χώρος Ω και ένα σύνολο ν ασυμβίβαστων μεταξύ τους ενδεχομένων $\{B1, B2, \dots, B\nu\}$ τα οποία καλύπτουν το σύνολο του δειγματικού χώρου (επομένως αποτελούν μια διαμέριση του χώρου, με $B1 \cup B2 \cup \dots \cup B\nu = \Omega \; και \; Bi \cap Bj = \emptyset$), τότε για κάθε ενδεχόμενο A στον Ω έχουμε ότι

$$P(B_i|A) = \frac{P(B_i)P(A|B_i)}{P(A)} = \frac{P(B_i)P(A|B_i)}{\sum_{i=1}^{\nu} P(B_i)P(A|B_i)}$$

Αν και το θεώρημα εμφανίζεται ως μια εναλλακτική μορφή του τύπου για τον υπολογισμό των δεσμευμένων πιθανοτήτων, εν τούτοις έχει λάβει σημαντική προσοχή καθώς η ερμηνεία του οδηγεί σε ένα νέο πλαίσιο υπολογισμού των πιθανοτήτων, όπου νέα πληροφορία οδηγεί στην ανανέωση της τιμής των πιθανοτήτων. Η αρχική επομένως εκτίμηση (εκ των προτέρων πιθανότητα) μπορεί έπειτα από την εκτέλεση πειραμάτων ή παρατηρήσεων να αναθεωρηθεί αξιοποιώντας το θεώρημα του Bayes και να οδηγήσει στην εκ των υστέρων πιθανότητα ενός ενδεχομένου. Αυτή η ερμηνεία του θεωρήματος του Bayes έχει οδηγήσει στην ανάπτυξη του πολύ σημαντικού κλάδου της Bayesian στατιστικής και έχει μεγάλη χρήση και σε εφαρμογές μηχανικής μάθησης.

Ειδικότερα, στον τύπο το θεωρήματος Bayes,

$$P(B_i|A) = \frac{P(B_i)P(A|B_i)}{P(A)}$$

η κατά Bayes ερμηνεία στην Bayesian συμπερασματολογία είναι η ακόλουθη:

- Ο όρος $P(B_i|A)$ αποτελεί την ‘εκ των υστέρων πιθανότητα’ (posterior probability) του ενδεχομένου B_i δοθέντος του A, δηλαδή την πιθανότητα κάτω από το πρίσμα νέας πληροφορίας ότι το A έχει πραγματοποιηθεί.

- Ο όρος $P(B_i)$ αποτελεί την 'εκ των προτέρων πιθανότητα' (prior probability) ενδεχομένου B_i, δηλαδή την αρχική εκτίμηση πιθανότητας με βάση τις αρχικές διαθέσιμες πληροφορίες, ανεξάρτητα από τον αν συμβεί το A.
- Ο όρος $P(A|B_i)$ αποτελεί την 'πιθανοφάνεια', δηλαδή το πόσο πιθανό είναι να συμβεί το ενδεχομένου A δοθέντος του B_i.
- Ο όρος $P(A)$ αποτελεί την πιθανότητα να συμβεί το A και είναι η περιθώρια πιθανότητα του A.

Η παραπάνω ερμηνεία είναι πολύ σημαντική για τον χώρο της στατιστικής και επίσης αποτελεί το θεμέλιο σε αρκετές μεθόδους μηχανικής μάθησης, όπως η ομάδα των ταξινομητών Naïve Bayes.

3.11.1. Παράδειγμα

Βρέθηκε ότι από τα 3.000 μηνύματα που έφθασαν μια χρονική περίοδο σε έναν mail server, τα 2.000 είναι Spam και τα 1.000 δεν είναι Spam. Βρέθηκε επίσης ότι η λέξη «Casino» εμφανίσθηκε σε 250 από τα 2.000 μηνύματα που είναι Spam, και σε 5 από τα 1.000 μηνύματα που δεν είναι Spam.

Ένα μήνυμα φθάνει στον mail server και διαπιστώνουμε ότι περιέχει τη λέξη «Casino». Ποια είναι η πιθανότητα το μήνυμα αυτό να είναι Spam;

Επίλυση

Το ζητούμενο είναι να υπολογιστεί η πιθανότητα P(Spam|Casino).

Εφαρμόζοντας τον τύπο του Bayes, ισχύει ότι

$$P(\text{Spam}|\text{Casino})=\frac{P(\text{Casino}|\text{Spam})}{P(\text{Casino})}P(\text{Spam})$$

Εκτελώντας τους υπολογισμούς, προκύπτει

$$P(\text{Spam})=\frac{2000}{3000}=0,67$$

$$P(\text{OXI Spam})=1-0,67=0,33$$

$$P(\text{Casino}|\text{Spam})=\frac{250}{2000}0,125,\ P(\text{Casino}|\text{OXI Spam})=\frac{5}{1000}0,005$$

$$P(\text{Casino})=P(\text{Casino}|\text{Spam})\,P(\text{Spam})+P(\text{Casino}|\ \text{OXI Spam})\,P(\text{OXI Spam})=0,125*0,67+0,005*0,33=0,085$$

$$P(\text{Spam}|\text{Casino})=\frac{P(\text{Casino}|\text{Spam})\,P(\text{Spam})}{P(\text{Casino})}=\frac{0,125*0,67}{0,085}=0,98$$

Η πιθανότητα είναι 98%.

- Έτσι, αν κατασκευαστεί ένα Bayeasian φίλτρο μηνυμάτων (αλγόριθμος μηχανικής μάθησης για παράδειγμα) και στο οποίο τεθεί ως επίπεδο απόρριψης η πιθανότητα 95% (0.95), τότε το συγκεκριμένο μήνυμα απορρίπτεται ως Spam.
- Ανάλογα υπολογίζουμε τις αντίστοιχες πιθανότητες για φίλτρα που ελέγχουν περισσότερες από μια λέξεις.

3.12. Παραδείγματα

3.12.1. Παράδειγμα 1

Επιλέγεται με τυχαίο τρόπο μια κάρτα από μια τράπουλα 52 φύλλων. Ποια είναι η πιθανότητα η κάρτα να είναι

1. Τέσσερα,
2. Μαύρη κάρτα,
3. Άσσος και κόκκινο,
4. Άσσος δεδομένου ότι είναι κόκκινη κάρτα
5. Κόκκινη κάρτα δεδομένου ότι είναι Τρία
6. Τρία δεδομένου ότι είναι κούπα

3.12.2. Παράδειγμα 2

Σε μια ομάδα 200 καταναλωτών πραγμτοποιήθηκε έρευνα για το αν προτιμούν να καταναλώνουν το προϊόν Α ή το προϊόν Β. Μεταξύ της ομάδας, 120 δήλωσαν ότι προτιμούν μόνο το προϊόν Α, 50 ότι προτιμούν μόνο το προϊόν Β και 20 είπαν ότι προτιμούν και το προϊόν Α και το προϊόν Β.

1. Ποια είναι η πιθανότητα ένας καταναλωτής που επιλέγεται τυχαία να προτιμά να καταναλώνει το προϊόν Α δεδομένου ότι προτιμά και το προϊόν Β;
2. Ποια είναι η πιθανότητα ένας καταναλωτής που επιλέγεται τυχαία να προτιμά να καταναλώνει το προϊόν Β δεδομένου ότι προτιμά και το προϊόν Α;
3. Ποια είναι η πιθανότητα ένας καταναλωτής που επιλέγεται τυχαία να προτιμά να καταναλώνει το προϊόν Α δεδομένου ότι προτιμά ένα προϊόν μόνο.

3.12.3. Παράδειγμα 3

Στον παρακάτω πίνακα περιλαμβάνονται τα αποτελέσματα έρευνας για προτιμήσεις μεταξύ τεσσάρων προϊόντων ανά ηλικιακή κατηγορία σε σύνολο 200 καταναλωτών.

Πίνακας 3.2: Αποτελέσματα έρευνας τεσσάρων προϊόντων

	Προϊόν Α	Προϊόν Β	Προϊόν Γ	Προϊόν Δ	Σύνολο
Έως 25 ετών	35	10	25	15	85
25-50 ετών	10	15	20	5	50
Άνω των 50 ετών	15	15	5	30	65
Σύνολο	60	40	50	50	200

1. Ποια είναι η πιθανότητα ένας τυχαία επιλεγείς καταναλωτής να είναι έως 25 ετών και να προτιμά το προϊόν Γ;
2. Ποια είναι η πιθανότητα ένας καταναλωτής να προτιμά το προϊόν Α γνωρίζοντας ότι είναι 40 ετών;
3. Ποια είναι η πιθανότητα ένας καταναλωτής να είναι άνω των 50 ετών αν γνωρίζουμε ότι προτιμά το προϊόν Δ;

3.12.4. Παράδειγμα 4

Μια εταιρία παραγωγής η οποία κατασκευάζει οθόνες υπολογιστών διαθέτει τρία εργοστάσια. Το εργοστάσιο Α παράγει 20%, εργοστάσιο Β το 50% και εργοστάσιο C το 30% των οθονών. Το 2% των οθονών που παράγονται στο εργοστάσιο Α, το 1% των οθονών που παράγονται στο εργοστάσιο Β και το 3% των οθονών που παράγονται στο εργοστάσιο C είναι ελαττωματικές. Μια οθόνη επιλέγεται τυχαία στην αγορά και διαπιστώνεται ελαττωματική.

Ποια είναι η πιθανότητα αυτός η οθόνη να κατασκευάστηκε από το εργοστάσιο Β;

(οδηγία: απαιτείται το Θεώρημα Bayes)

3.12.5. Παράδειγμα 5

Εκτιμάται ότι ποσοστό 50% των email είναι spam και τοποθετούνται αυτόματα στον φάκελο ανεπιθύμητων. Ένα anti-spam φίλτρο διαφημίζει ότι έχει τη δυνατότητα να αναγνωρίσει το 99% των spam email (δηλαδή να χαρακτηρίσει ένα email ως spam αν αυτό είναι όντως spam), και η πιθανότητα να ανιχνεύσει ένα κανονικό email ως spam (ενώ δεν είναι spam) είναι 5%.

Ποια είναι η πιθανότητα ένα email το οποίο τοποθετείται στο φάκελο ανεπιθύμητων να ΜΗΝ είναι όντως spam;

(οδηγία: απαιτείται το Θεώρημα Bayes)

3.13. Ενδεικτικές λύσεις παραδειγμάτων με χρήση python

3.13.1. Παράδειγμα 1

```
from fractions import Fraction

# Πιθανότητα το φύλλο να είναι τέσσερα
def probability_four():
    total_cards = 52
    four_cards = 4
    probability = Fraction(four_cards, total_cards)
    return probability

# Εκτύπωση του αποτελέσματος με κλάσμα και ποσοστό
def print_probability(name, probability):
    decimal_probability = float(probability)
    print(f"Πιθανότητα {name}: {probability}
({decimal_probability*100:.2f}%)")

# Πιθανότητα το φύλλο να είναι μαύρο
def probability_black():
    black_cards = 26  # Μισές οι μαύρες κάρτες σε μια τράπουλα
    total_cards = 52
    probability = Fraction(black_cards, total_cards)
    return probability

# Πιθανότητα το φύλλο να είναι Άσσος και κόκκινο
def probability_red_ace():
    red_aces = 2  # Δύο Άσσοι είναι κόκκινοι στην τράπουλα
    total_cards = 52
    probability = Fraction(red_aces, total_cards)
    return probability

# Πιθανότητα το φύλλο να είναι Άσσος δεδομένου ότι είναι κόκκινη κάρτα
def probability_ace_given_red():
    red_cards = 26  # Μισές οι κόκκινες κάρτες σε μια τράπουλα
    red_aces = 2  # Δύο Άσσοι είναι κόκκινοι στην τράπουλα
    probability = Fraction(red_aces, red_cards)
    return probability

# Πιθανότητα το φύλλο να είναι κόκκινη κάρτα δεδομένου ότι είναι τρία
def probability_red_given_three():
    red_threes = 2  # Δύο κόκκινα τρία στην τράπουλα
    total_threes = 4  # Συνολικά τέσσερα τρία στην τράπουλα
```

```
    probability = Fraction(red_threes, total_threes)
    return probability

# Πιθανότητα το φύλλο να είναι τρία δεδομένου ότι είναι κούπα
def probability_three_given_cups():
    cups_threes = 1  # Ένα κόκκινο τρία στην τράπουλα
    total_cups = 13  # Συνολικά 13 κούπες στην τράπουλα
    probability = Fraction(cups_threes, total_cups)
    return probability

# Εκτύπωση των αποτελεσμάτων
print_probability("Τέσσερα", probability_four())
print_probability("Μαύρη κάρτα", probability_black())
print_probability("Άσσος και κόκκινο", probability_red_ace())
print_probability("Άσσος δεδομένου ότι είναι κόκκινη κάρτα",
probability_ace_given_red())
print_probability("Κόκκινη κάρτα δεδομένου ότι είναι Τρία",
probability_red_given_three())
print_probability("Τρία δεδομένου ότι είναι κούπα",
probability_three_given_cups())
```

```
Πιθανότητα Τέσσερα: 1/13 (7.69%)
Πιθανότητα Μαύρη κάρτα: 1/2 (50.00%)
Πιθανότητα Άσσος και κόκκινο: 1/26 (3.85%)
Πιθανότητα Άσσος δεδομένου ότι είναι κόκκινη κάρτα: 1/13 (7.69%)
Πιθανότητα Κόκκινη κάρτα δεδομένου ότι είναι Τρία: 1/2 (50.00%)
Πιθανότητα Τρία δεδομένου ότι είναι κούπα: 1/13 (7.69%)
```

3.13.2. Παράδειγμα 2

```
# Υπολογισμός πιθανοτήτων

# Συνολικός αριθμός καταναλωτών
total_consumers = 200

# Πλήθος καταναλωτών που προτιμούν το προϊόν Α μόνο
consumers_A_only = 120

# Πλήθος καταναλωτών που προτιμούν το προϊόν Β μόνο
consumers_B_only = 50

# Πλήθος καταναλωτών που προτιμούν και τα δύο προϊόντα
consumers_both = 20

# α) Πιθανότητα ένας καταναλωτής που προτιμά και το προϊόν Α να προτιμά
και το Β
probability_A_given_B = consumers_both / total_consumers
```

```
print("α) Πιθανότητα ένας καταναλωτής που επιλέγεται τυχαία να προτιμά να καταναλώνει το προϊόν Α δεδομένου ότι προτιμά και το προϊόν Β:", probability_A_given_B)

# β) Πιθανότητα ένας καταναλωτής που προτιμά και το προϊόν Β να προτιμά και το Α
probability_B_given_A = consumers_both / total_consumers
print("β) Πιθανότητα ένας καταναλωτής που επιλέγεται τυχαία να προτιμά να καταναλώνει το προϊόν Β δεδομένου ότι προτιμά και το προϊόν Α:", probability_B_given_A)

# γ) Πιθανότητα ένας καταναλωτής που προτιμά ένα προϊόν μόνο να προτιμά το Α
probability_A_given_single_preference = consumers_A_only / (consumers_A_only + consumers_B_only)
print("γ) Πιθανότητα ένας καταναλωτής που επιλέγεται τυχαία να προτιμά να καταναλώνει το προϊόν Α δεδομένου ότι προτιμά ένα προϊόν μόνο:", probability_A_given_single_preference)
```

```
α) Πιθανότητα ένας καταναλωτής που επιλέγεται τυχαία να προτιμά να καταναλώνει το προϊόν Α δεδομένου ότι προτιμά και το προϊόν Β: 0.1

β) Πιθανότητα ένας καταναλωτής που επιλέγεται τυχαία να προτιμά να καταναλώνει το προϊόν Β δεδομένου ότι προτιμά και το προϊόν Α: 0.1

γ) Πιθανότητα ένας καταναλωτής που επιλέγεται τυχαία να προτιμά να καταναλώνει το προϊόν Α δεδομένου ότι προτιμά ένα προϊόν μόνο: 0.7058823529411765
```

3.13.3. Παράδειγμα 3

```
# Δεδομένα από τον πίνακα
age_groups = ['Έως 25 ετών', '25-50 ετών', 'Άνω των 50 ετών']
products = ['Προϊόν Α', 'Προϊόν Β', 'Προϊόν Γ', 'Προϊόν Δ']

# Αριθμός καταναλωτών για κάθε ηλικιακή κατηγορία και προϊόν
consumer_data = {
    'Έως 25 ετών': {'Προϊόν Α': 35, 'Προϊόν Β': 10, 'Προϊόν Γ': 25, 'Προϊόν Δ': 15},
    '25-50 ετών': {'Προϊόν Α': 10, 'Προϊόν Β': 15, 'Προϊόν Γ': 20, 'Προϊόν Δ': 5},
    'Άνω των 50 ετών': {'Προϊόν Α': 15, 'Προϊόν Β': 15, 'Προϊόν Γ': 5, 'Προϊόν Δ': 30}
}
```

```
# Συνολικός αριθμός καταναλωτών
total_consumers = 200

# α) Πιθανότητα ένας τυχαία επιλεγμένος καταναλωτής να είναι έως 25 ετών και να προτιμά το προϊόν Γ
probability_a = consumer_data['Έως 25 ετών']['Προϊόν Γ'] / total_consumers
print("Απάντηση στο (α): Η πιθανότητα ένας τυχαία επιλεγμένος καταναλωτής να είναι έως 25 ετών και να προτιμά το προϊόν Γ είναι:", probability_a)

# β) Πιθανότητα ένας καταναλωτής να προτιμά το προϊόν Α γνωρίζοντας ότι είναι 40 ετών
# Υποθέτουμε ότι η ηλικία 40 ετών ανήκει στην ηλικιακή κατηγορία 25-50 ετών
probability_b = consumer_data['25-50 ετών']['Προϊόν Α'] / sum(consumer_data['25-50 ετών'].values())
print("Απάντηση στο (β): Η πιθανότητα ένας καταναλωτής να προτιμά το προϊόν Α γνωρίζοντας ότι είναι 40 ετών είναι:", probability_b)


# γ) Πιθανότητα ένας καταναλωτής να είναι άνω των 50 ετών αν γνωρίζουμε ότι προτιμά το προϊόν Δ
# Από τον πίνακα, οι πιθανοί καταναλωτές άνω των 50 ετών που προτιμούν το προϊόν Δ είναι 30
# Συνολικά οι καταναλωτές που προτιμούν το προϊόν Δ είναι 50
probability_c = consumer_data['Άνω των 50 ετών']['Προϊόν Δ'] / sum(consumer_data['Άνω των 50 ετών'].values())
print("Απάντηση στο (γ): Η πιθανότητα ένας καταναλωτής να είναι άνω των 50 ετών αν γνωρίζουμε ότι προτιμά το προϊόν Δ είναι:", probability_c)
```

```
Απάντηση στο (α): Η πιθανότητα ένας τυχαία επιλεγμένος καταναλωτής να είναι έως 25 ετών και να προτιμά το προϊόν Γ είναι: 0.125
Απάντηση στο (β): Η πιθανότητα ένας καταναλωτής να προτιμά το προϊόν Α γνωρίζοντας ότι είναι 40 ετών είναι: 0.2
Απάντηση στο (γ): Η πιθανότητα ένας καταναλωτής να είναι άνω των 50 ετών αν γνωρίζουμε ότι προτιμά το προϊόν Δ είναι: 0.46153846153846156
```

3.13.4. Παράδειγμα 4

```
# Υπολογισμός πιθανοτήτων
p_A = 0.20  # Πιθανότητα εργοστασίου Α
p_B = 0.50  # Πιθανότητα εργοστασίου Β
p_C = 0.30  # Πιθανότητα εργοστασίου C
```

```
p_defect_given_A = 0.02  # Πιθανότητα ελαττωματικής οθόνης δεδομένου ότι προέρχεται από το εργοστάσιο A
p_defect_given_B = 0.01  # Πιθανότητα ελαττωματικής οθόνης δεδομένου ότι προέρχεται από το εργοστάσιο B
p_defect_given_C = 0.03  # Πιθανότητα ελαττωματικής οθόνης δεδομένου ότι προέρχεται από το εργοστάσιο C

# Χρησιμοποιούμε το Θεώρημα Bayes για να υπολογίσουμε την πιθανότητα του εργοστασίου B δεδομένης μιας ελαττωματικής οθόνης
# P(B|D) = (P(D|B) * P(B)) / P(D)

# Υπολογισμός του παρονομαστή P(D)
p_defect = p_defect_given_A * p_A + p_defect_given_B * p_B + p_defect_given_C * p_C

# Υπολογισμός της πιθανότητας ελαττωματικής οθόνης που προήλθε από το εργοστάσιο B
p_B_given_defect = (p_defect_given_B * p_B) / p_defect

print("Η πιθανότητα ότι η ελαττωματική οθόνη προήλθε από το εργοστάσιο B είναι:", p_B_given_defect)
```

```
Η πιθανότητα ότι η ελαττωματική οθόνη προήλθε από το εργοστάσιο B είναι: 0.27777777777777773
```

3.13.5. Παράδειγμα 5

```
def calculate_not_spam_given_in_spam_folder():
    p_notA_given_B = (0.05 * 0.5) / ((0.99 * 0.5) + (0.05 * 0.5))
    return p_notA_given_B

result = calculate_not_spam_given_in_spam_folder()

# Εκτύπωση
print("Η πιθανότητα ένα email που τοποθετείται στο φάκελο ανεπιθύμητων να μην είναι spam είναι:", result, "ή", "{:.2%}".format(result))
```

```
Η πιθανότητα ένα email που τοποθετείται στο φάκελο ανεπιθύμητων να μην είναι spam είναι: 0.04807692307692308 ή 4.81%
```

ΚΕΦΑΛΑΙΟ 4: Επισκόπηση βασικών στοιχείων στατιστικής

4.1. Περιγραφική στατιστική, δειγματοληψία, περιγραφικά μέτρα ποσοτικών δεδομένων

4.1.1. Βασικές έννοιες στατιστικής

Η στατιστική επιστήμη μπορεί να οριστεί ως η επιστήμη η οποία έχει ως σκοπό τη δημιουργία μεθόδων και τεχνικών οι οποίες υποστηρίζουν τη λήψη αποφάσεων κάτω από αβεβαιότητα. Για να το επιτύχει ασχολείται συστηματικά με τα δεδομένα τα οποία προκύπτουν είτε ως αποτελέσματα μετρήσεων ή ως πειραματικές τιμές. Επειδή βασίζεται στα δεδομένα, αποτελεί την κατεξοχήν επιστήμη η οποία έχει ως κύριο αντικείμενο τα δεδομένα, καθώς και την εξαγωγή συμπερασμάτων και λήψη αποφάσεων βασιζόμενη σε αυτά.

Διακρίνεται σε δύο κύριες περιοχές, την περιγραφική στατιστική και την στατιστική συμπερασματολογία. Το αντικείμενο της περιγραφικής στατιστικής είναι η συλλογή, οργάνωση και η συνοπτική παρουσίαση των δεδομένων, καθώς και η παράστασή τους με γραφήματα και ο υπολογισμός κατάλληλων περιγραφικών μέτρων. Η στατιστική συμπερασματολογία, η οποία έχει αναπτυχθεί σε μεγάλο βαθμό τις τελευταίες δεκαετίες, έχει ως αντικείμενο την δημιουργία μεθόδων για τη λήψη αποφάσεων με έγκυρο τρόπο από δεδομένα τα οποία είναι ελλιπή και κατά συνέπεια δημιουργούν αβεβαιότητα. Το εύρος εφαρμογών των μεθόδων της στατιστικής είναι ευρύτατο και εκτείνεται σχεδόν σε κάθε τομέα. Ειδικότερα, σε ότι αφορά στη μηχανική μάθηση, αρκετές μεθοδολογίες της έχουν αφετηρία στατιστικές μεθόδους εμπλουτισμένες με νέα στοιχεία.

Πίνακας 4.1. Κύριες περιοχές στατιστικής

Κύριες περιοχές Στατιστικής	Αντικείμενο
Περιγραφική Στατιστική	Μέθοδοι για οργάνωση, ανάλυση, περιγραφή, σύνοψη, παρουσίαση, δεδομένων
Στατιστική Συμπερασματολογία	Μέθοδοι για εκτίμηση τιμών παραμέτρων, διαστημάτων, έλεγχο υποθέσεων και εξαγωγή συμπερασμάτων για τον πληθυσμό γενικότερα
Μαθηματική Στατιστική	Θεωρητική μελέτη, ανάπτυξη και θεμελίωση μεθόδων

Θεωρία Δειγματοληψίας	Ανάπτυξη θεωρητικής και πρακτικής για λήψη δειγμάτων από πληθυσμό

Βασικές εισαγωγικές έννοιες αποτελούν τα δεδομένα, οι τύποι και οι κλίμακες μέτρησης, ο πληθυσμός και το δείγμα και η διάκριση μεταξύ παραμέτρων πληθυσμού και στατιστικών μεταβλητών, τα οποία παρουσιάζονται συνοπτικά στη συνέχεια.

4.1.2. Τύποι δεδομένων

Η αφετηρία της στατιστικής είναι τα δεδομένα και οι διαδικασίες μέτρησης και συλλογής τους. Ανάλογα με τη διαδικασία μέτρησης έχουν τυποποιηθεί τέσσερις κλίμακες μέτρησης, η ονομαστική, η κλίμακα διάταξης, η κλίμακα διαστήματος και η κλίμακα λόγου. Η επιλογή κλίμακας σχετίζεται με τον τύπο δεδομένων, ο οποίος στη γενική περίπτωση εντάσσεται κάτω από μια γενική κατηγοριοποίηση η οποία διακρίνει τα αριθμητικά από τα μη αριθμητικά δεδομένα.

Ειδικότερα, τα δεδομένα διακρίνονται σε ποσοτικά, όταν πρόκειται για αριθμητικές τιμές (όπως για παράδειγμα η ηλικία, ή το βάρος ενός ανθρώπου), και ποιοτικά, όταν πρόκειται για δεδομένα τα οποία αφορούν κατηγορικές παρατηρήσεις, δηλαδή παρατηρήσεις που αφορούν ένταξη σε κατηγορίες (όπως για παράδειγμα το χρώμα ή το σχήμα ενός αντικειμένου). Τα ποσοτικά δεδομένα διακρίνονται σε διακριτά αν η τιμή είναι ακέραιος αριθμός, και συνεχή αν πρόκειται για πραγματικό αριθμό.

Πίνακας 4.2. Τύποι δεδομένων

Τύπος δεδομένων	Περιγραφή
Ποσοτικά	Αριθμητικές τιμές που προκύπτουν από μετρήσεις οποιουδήποτε είδους Διακρίνονται σε • Συνεχείς • Διακριτές
Ποιοτικά	Κείμενο ή αριθμητικές τιμές οι οποίες εκφράζουν κατηγορίες

4.1.3. Κλίμακες μέτρησης

Με βάση τους δύο τύπους δεδομένων, οι κλίμακες μέτρησης που χρησιμοποιούνται για ποιοτικά δεδομένα είναι η ονομαστική και η κλίμακα διάταξης. Στην ονομαστική κλίμακα οι τιμές αφορούν την κατηγορία στην οποία εντάσσεται μια παρατήρηση, και οι κατηγορίες δεν έχουν διάταξη. Για παράδειγμα το χρώμα ή το σχήμα ενός αντικειμένου. Στην κλίμακα διάταξης, οι τιμές αφορούν την

κατηγορία στην οποία εντάσσεται μια παρατήρηση, ωστόσο οι κατηγορίες είναι διαταγμένες. Για παράδειγμα η αξιολόγηση ενός προϊόντος ή η προτίμηση ενός ερωτώμενου για κάποιο θέμα. Για τα ποσοτικά δεδομένα χρησιμοποιούνται η κλίμακα διαστήματος και η κλίμακα λόγου. Η κλίμακα διαστήματος έχει μια αυθαίρετη αρχή και οι διαφορές τιμών έχουν νόημα, όπως και η διάταξη. Παράδειγμα κλίμακας διάταξης αποτελεί η κλίμακα θερμοκρασίας. Τέλος, η κλίμακα λόγου είναι μια κλίμακα διαστήματος με το επιπλέον χαρακτηριστικό ότι έχει νόημα ο λόγος τιμών και η αρχή της κλίμακας είναι καθορισμένη. Για παράδειγμα το βάρος ενός αντικειμένου.

Πίνακας 4.3. Κλίμακες μέτρησης

Κλίμακα μέτρησης	Τύπος δεδομένων	Περιγραφή
Ονομαστική κλίμακα	Κείμενο ή αριθμητική τιμή η οποία δεν υπονοεί κάποια διάταξη.	Οι τιμές εκφράζουν τις κατηγορίες χωρίς κάποια διάταξη. Είναι εφικτή μόνο απλή καταμέτρηση των στοιχείων κάθε κατηγορίας.
Κλίμακα διάταξης	Κείμενο ή αριθμητική τιμή.	Οι τιμές εκφράζουν τις κατηγορίες αλλά υπάρχει διάταξη των κατηγοριών. Είναι εφικτή καταμέτρηση των στοιχείων κάθε κατηγορίας, αλλά και διάταξή τους.
Κλίμακα διαστήματος	Αριθμητική τιμή.	Η κλίμακα είναι ορισμένη με συγκεκριμένο και ίσο διάστημα μεταξύ των τιμών και οι τιμές είναι διατεταγμένες. Η αρχή της κλίμακας ορίζεται αυθαίρετα. Μπορούν να εκτελεστούν όλοι οι στατιστικοί έλεγχοι εκτός όσων απαιτούν χρήση λόγου.
Κλίμακα λόγου	Αριθμητική τιμή.	Όπως και η κλίμακα διαστήματος, αλλά η αρχή της κλίμακας αντιστοιχεί στο μηδέν, το οποίο έχει νόημα (μη ύπαρξη). Μπορούν να εκτελεστούν όλοι οι στατιστικοί έλεγχοι. Η κλίμακα λόγου παρέχει την πλουσιότερη πληροφόρηση.

4.1.4. Πληθυσμός και δείγμα

Σε κάθε στατιστική μελέτη, είτε αφορά απλή περιγραφική στατιστική ή στατιστική συμπερασματολογία, το πεδίο έρευνας είναι η απάντηση ερωτημάτων για κάποιο ή κάποια χαρακτηριστικά ενός συνόλου οντοτήτων το οποίο βρίσκεται υπό μελέτη. Το σύνολο οντοτήτων το οποίο βρίσκεται υπό μελέτη καλείται πληθυσμός και περιλαμβάνει τις μονάδες που ενδιαφέρουν την εκάστοτε έρευνα. Οι μονάδες μπορεί να είναι άνθρωποι, αυτοκίνητα, επιχειρήσεις, ή οτιδήποτε άλλο υλικό ή άυλο. Οι παρατηρήσεις ή πειράματα που λαμβάνουν χώρα σε μια στατιστική έρευνα εστιάζουν σε συγκεκριμένο ή συγκεκριμένα χαρακτηριστικά του πληθυσμού, τα οποία ονομάζονται παράμετροι. Για παράδειγμα αν μια έρευνα αφορά στην διερεύνηση του μέσου εισοδήματος σε μια χώρα, ο πληθυσμός περιλαμβάνει το σύνολο των εργαζομένων στην χώρα και η παράμετρος του πληθυσμού που διερευνάται είναι το εισόδημα. Για τη μελέτη της παραμέτρου ορίζονται κατάλληλες στατιστικές μεταβλητές για τη συλλογή μετρήσιμων τιμών με την κατάλληλη κλίμακα μέτρησης. Η συλλογή δεδομένων στις περισσότερες περιπτώσεις εκτελείται σε ένα υποσύνολο των μονάδων του πληθυσμού, είτε λόγω αδυναμίας ή λόγω κόστους ή και λόγω μη αναγκαιότητας. Το υποσύνολο του πληθυσμού καλείται δείγμα και η δειγματοληψία, η οποία αποτελεί ειδικό πεδίο της στατιστικής θεωρίας, πραγματοποιείται με κατάλληλη επιλογή τεχνικής με βάση θεωρητικές και πρακτικές απαιτήσεις. Τα δεδομένα τα οποία συλλέγονται από το δείγμα χρησιμοποιούνται για τον υπολογισμό στατιστικών συναρτήσεων για την εκτίμηση στη συνέχεια των τιμών των παραμέτρων του πληθυσμού. Η μετακίνηση από τον πληθυσμό στο δείγμα με τεχνικές δειγματοληψίας, η συλλογή δεδομένων, ο υπολογισμός των στατιστικών συναρτήσεων και η εκτίμηση στη συνέχεια των τιμών των παραμέτρων του πληθυσμού αποτελεί τον πυρήνα της στατιστικής μεθοδολογίας.

Πίνακας 4.4. Πληθυσμός και δείγμα

	Πληθυσμός	**Δείγμα**
Ορισμός	Το πλήρες σύνολο των οντοτήτων υπό μελέτη, το οποίο ενδέχεται να είναι μη εφικτό ή μη επιθυμητό να εξεταστεί στο σύνολό του.	Το υποσύνολο του πληθυσμού, το οποίο επιλέγεται ως ‘αντιπροσωπευτικό’ του πληθυσμού και αποτελεί το αντικείμενο της έρευνας στην πράξη.
Βασική έννοια	Παράμετρος	Στατιστική συνάρτηση

4.1.5. Τεχνικές δειγματοληψίας

Η συλλογή δεδομένων σε μια στατιστική έρευνα έχει ως σκοπό είτε την σύνοψη πληροφοριών και παρουσίαση ή την εκτίμηση και εξαγωγή συμπερασμάτων για τον πληθυσμό και ενδεχομένως λήψη

απόφασης. Στην πρώτη περίπτωση χρησιμοποιούνται τεχνικές της περιγραφικής στατιστικής, ενώ στη δεύτερη αξιοποιούνται μέθοδοι της στατιστικής συμπερασματολογίας.

Οι τεχνικές δειγματοληψίας ομαδοποιούνται σε δύο κατηγορίες με βάση το αν χρησιμοποιούν κάποιο μοντέλο πιθανοτήτων για την επιλογή δείγματος από το σύνολο των δειγματοληπτικών μονάδων, ή αλλιώς πλαίσιο δειγματοληψίας. Στην κατηγορία των μεθόδων βασισμένων σε πιθανότητες κυριότεροι εκπρόσωποι είναι η απλή τυχαία δειγματοληψία, η στρωματοποιημένη δειγματοληψία, η δειγματοληψία κατά ομάδες και η συστηματική δειγματοληψία. Ενώ η δειγματοληψία ευκολίας, η δειγματοληψία σκοπιμότητας και η δειγματοληψία με ορισμένα ποσοστά ανήκουν στις μεθόδους που δεν χρησιμοποιούν πιθανότητες. Η βασική διαφορά των δύο κατηγοριών είναι ότι η χρήση πιθανοτήτων αυξάνει το βαθμό αντιπροσωπευτικότητας του πληθυσμού στο δείγμα και συνεπώς επιτρέπει τη γενίκευση των ευρημάτων από το δείγμα στον πληθυσμό. Για τη στατιστική συμπερασματολογία αποτελεί προϋπόθεση η επιλογή μεθόδου δειγματοληψίας η οποία χρησιμοποιεί πιθανότητες για τη δημιουργία δείγματος.

Πίνακας 4.5. Τεχνικές δειγματολοψίας

Μέθοδος	Χρήση πιθανοτήτων	Περιγραφή
Απλή τυχαία δειγματοληψία	ΝΑΙ	Με χρήση μοντέλου πιθανοτήτων κάθε στοιχείο του συνόλου του πληθυσμού έχει την ίδια πιθανότητα να ενταχθεί στο δείγμα.
Συστηματική δειγματοληψία	ΝΑΙ	Με χρήση μοντέλου πιθανοτήτων και επαναλαμβανόμενου τυχαία ορισμένου μοτίβου κάθε στοιχείο του συνόλου του πληθυσμού έχει την ίδια πιθανότητα να ενταχθεί στο δείγμα.
Στρωματοποιημένη δειγματοληψία	ΝΑΙ	Διαμοιρασμός των στοιχείων του πληθυσμού σε στρώματα με βάση χαρακτηριστικά της έρευνας και εφαρμογή στη συνέχεια απλής τυχαίας δειγματοληψίας.
Δειγματοληψία κατά ομάδες	ΝΑΙ	Διαμοιρασμός των στοιχείων του πληθυσμού σε ομάδες με βάση χαρακτηριστικά της έρευνας, ή με τυχαίο τρόπο και εφαρμογή στη συνέχεια απλής τυχαίας δειγματοληψίας στις ομάδες.

Δειγματοληψία ευκολίας	ΟΧΙ	Επιλογή στοιχείων από το σύνολο του πληθυσμού με βάση την ευκολία του ερευνητή.
Δειγματοληψία σκοπιμότητας	ΟΧΙ	Επιλογή στοιχείων από το σύνολο του πληθυσμού με βάση την ερευνητική στόχευση του ερευνητή.
Δειγματοληψία με ορισμένα ποσοστά	ΟΧΙ	Παρόμοια με τη στρωματοποιημένη, αλλά με βάση υποκειμενικά κριτήρια και όχι τυχαία.

4.1.6. Οργάνωση και παρουσίαση δεδομένων

Ένας από τους στόχους της στατιστικής ανάλυσης δεδομένων είναι η υποβοήθηση στη λήψη αποφάσεων με βάση τα δεδομένα. Για ναι είναι αυτό εφικτό απαιτείται κατάλληλη οργάνωση και παρουσίαση των βασικών σημείων που εμφανίζονται και απορρέουν από ένα σύνολο δεδομένων. Η περιγραφική στατιστική εστιάζει στην οργάνωση και παρουσίαση των βασικών σημείων ενός συνόλου δεδομένων με συνοπτικό τρόπο. Αυτό πραγματοποιείται κατά κύριο λόγο με τη χρήση πινάκων, διαγραμμάτων και τον υπολογισμό περιγραφικών αριθμητικών μέτρων. Σε κάθε περίπτωση ο στόχος είναι η αξιόπιστη σύνοψη των πληροφοριών που μπορούν εξαχθούν από ένα σύνολο δεδομένων, το οποίο σε πολλές περιπτώσεις ενδέχεται να είναι αρκετά υψηλού όγκου.

Τα πρωτογενή δεδομένα, τα οποία έχουν προκύψει από τη διαδικασία συλλογής, είναι συνήθως εκτεταμένα και δεν προσφέρονται για εξαγωγή συμπερασμάτων ή παρουσίαση ως έχουν. Οπότε τα πρωτογενή δεδομένα οργανώνονται και παρουσιάζονται με διαφορετικούς τρόπους ανάλογα με τον τύπο τους και τον όγκο, αλλά και με βάση την εκάστοτε ερευνητική ανάγκη. Το βασικό ενδιαφέρον είναι η παρουσίαση μιας συνοπτικής αποτύπωσης η οποία αντανακλά τις βασικές τάσεις των τιμών. Αυτό επιτυγχάνεται με την παρουσίαση της κατανομής των συχνοτήτων, δηλαδή των κατηγοριών και των συχνοτήτων (του πλήθος των μονάδων του δείγματος που εντάσσονται σε μια κατηγορία ή λαμβάνουν μια τιμή ή εύρος τιμής).

4.1.6.1 Ποιοτικά δεδομένα

Για τα ποιοτικά δεδομένα η οργάνωση πραγματοποιείται κατά κύριο λόγο με συνοπτικούς πίνακες και πίνακες κατανομής σχετικών και απόλυτων συχνοτήτων, ενώ η παρουσίαση με ραβδογράμματα και κυκλικά διαγράμματα. Τα ποσοτικά δεδομένα επιτρέπουν περισσότερες εργασίες, οπότε η οργάνωση πραγματοποιείται κατά κύριο λόγο με συνοπτικούς πίνακες και πίνακες κατανομής σχετικών, απόλυτων και αθροιστικών συχνοτήτων, ενώ χρησιμοποιούνται επιπλέον τύποι

γραφημάτων όπως το ιστόγραμμα, το πολύγωνο συχνοτήτων, το διάγραμμα μίσχου-φύλλου και το διάγραμμα σημείων και το θηκόγραμμα.

Πίνακας 4.6. Ποιοτικά δεδομένα

Τύπος δεδομένων	Οργάνωση
Ποιοτικά (κατηγορικά)	Πίνακας κατανομής συχνοτήτων (απόλυτων, σχετικών, ποσοστιαίων)
	Πίνακας διπλής εισόδου (για δύο μεταβλητές)

Τύπος δεδομένων	Παρουσίαση
Ποιοτικά (κατηγορικά)	Ραβδόγραμμα
	Κυκλικό διάγραμμα
	Σύνθετο ραβδόγραμμα (για δύο μεταβλητές)

4.1.6.2 Ποσοτικά δεδομένα

Όταν στα ποσοτικά δεδομένα οι τιμές είναι συνεχείς, για τη δημιουργία του πίνακα κατανομής συχνοτήτων είναι αναγκαία η ομαδοποίηση των δεδομένων σε κλάσεις και στη συνέχεια ο υπολογισμός των συχνοτήτων εμφάνισης σε αυτές. Οι κλάσεις επιλέγονται με κριτήριο τις ανάγκες της έρευνας και συνήθως κυμαίνονται μεταξύ οκτώ και δεκαπέντε.

Πίνακας 4.7. Ποσοτικά δεδομένα

Τύπος δεδομένων	Οργάνωση
Ποσοτικά (αριθμητικά)	Πίνακας κατανομής συχνοτήτων (απόλυτων, σχετικών, ποσοστιαίων)
	Πίνακας αθροιστικής κατανομής συχνοτήτων (απόλυτων, σχετικών, ποσοστιαίων)
	Πίνακας διπλής εισόδου (για δύο μεταβλητές, μια ποσοτική και μια ποιοτική)

Τύπος δεδομένων	Παρουσίαση
Ποσοτικά (αριθμητικά)	Διάγραμμα μίσχου-φύλλου

	Διάγραμμα σημείων
	Ιστόγραμμα
	Θηκόγραμμα
	Διάγραμμα διασποράς (για δύο ποσοτικές μεταβλητές)

4.1.7. Περιγραφικά μέτρα ποσοτικών δεδομένων

Σε περίπτωση ποσοτικών δεδομένων οι παρουσιάσεις μέσω πινάκων και γραφημάτων αποτελούν το πρώτο βήμα στη διερευνητική ανάλυση, ωστόσο δεν είναι εύκολη η χρήση τους για περισσότερο σύνθετους υπολογισμούς, όπως για παράδειγμα σε αλγόριθμους μηχανικής μάθησης. Για το λόγο αυτό έχουν αναπτυχθεί αριθμητικά μέτρα τα οποία υπολογίζονται από τα δεδομένα ενός δείγματος. .Στην ουσία πρόκειται για στατιστικές συναρτήσεις, οι οποίες χρησιμοποιούν τιμές του δείγματος και παράγουν μια τιμή, η οποία ερμηνεύεται με κατάλληλο τρόπο. Τα περιγραφικά αριθμητικά μέτρα διακρίνονται σε μέτρα κεντρικής τάσης, μέτρα διασποράς, μέτρα σχήματος κατανομής.

Πίνακας 4.8. Περιγραφικά μέτρα ποσοτικών δεδομένων

Κατηγορία	Περιγραφή
Μέτρα κεντρικής τάσης	Εκτίμηση του σημείου συσσώρευσης των τιμών του δείγματος, κεντρικής τάσης.
Μέτρα διασποράς	Εκτίμηση του βαθμού μεταβλητότητας και απόστασης των τιμών από το σημείο κεντρικής τάσης.
Μέτρα σχήματος κατανομής	Εκτίμηση του βαθμού ασυμμετρίας της κατανομής των τιμών του δείγματος ως προς το σημείο κεντρικής τάσης (ασυμμετρία), και του βαθμού συγκέντρωσης και οξύτητας της κατανομής γύρω από αυτό (κύρτωση).
Μέτρα θέσης	Εκτίμηση κατανομής δεδομένων.

4.1.7.1 Μέτρα κεντρικής τάσης

Τα κύρια μέτρα κεντρικής τάσης για τις τιμές ενός δείγματος είναι ο αριθμητικός μέσος, η διάμεσος, και η επικρατούσα τιμή και ο υπολογισμός τους καθορίζεται με τον ανάλογο μαθηματικό τύπο. Σε περίπτωση όπου τα δεδομένα αφορούν σε πληθυσμό, υπάρχει διόρθωση του υπολογισμού, όπως

επίσης για την περίπτωση ομαδοποιημένων τιμών. Οι σχετικές θέσεις των τριών μέτρων δίνουν μια ένδειξη και του σχήματος της κατανομής των δεδομένων.

Πίνακας 4.9. Μέτρα κεντρικής τάσης

Μέτρα κεντρικής τάσης	**Ορισμός**	**Παρατηρήσεις**
Αριθμητικός μέσος	Για δείγμα n τιμών: Μη ομαδοποιημένα δεδομένα $\overline{X} = \frac{\sum_{i=1}^{n} X_i}{n}$ Ομαδοποιημένα δεδομένα $\overline{X} = \frac{\sum_{i=1}^{n} f_i m_i}{n}$ Για πληθυσμό Ν τιμών: Μη ομαδοποιημένα δεδομένα $\mu = \frac{\sum x}{N}$ Ομαδοποιημένα δεδομένα $\mu = \frac{\sum mf}{N}$	Ο αριθμητικός μέσος αποτελεί ένα πολύ δημοφιλές μέτρο, υπολογίζεται ως ο λόγος του αθροίσματος των τιμών δια του πλήθους των τιμών, και παρουσιάζει ευαισθησία σε ακραίες τιμές.
Διάμεσος	$Διάμεσος\ για\ περιττό\ πλήθος\ τιμών = x_{\frac{n+1}{2}}$ $Διάμεσος\ για\ άρτιο\ πλήθος\ τιμών = \frac{x_{(\frac{n}{2})} + x_{((\frac{n}{2})+1)}}{2}$	Η διάμεσος διαχωρίζει ένα σύνολο δεδομένων, το οποίο έχει προηγουμένως τεθεί σε αύξουσα διάταξη, σε δύο ίσα υποσύνολα. Οι τιμές του ενός υποσυνόλου είναι μικρότερες της διαμέσου, και οι τιμές του δεύτερου υψηλότερες.
Επικρατούσα τιμή	Δεν υπάρχει κάποιος τύπος, και ο υπολογισμός εξαρτάται από το σύνολο δεδομένων και τον μηχανισμό απαρίθμησης.	Η επικρατούσα τιμή έχει την υψηλότερη συχνότητα εμφάνισης. Ενδέχεται να μην υπάρχει, να έχει

		μοναδική τιμή, ή πολλαπλή τιμή.

4.1.7.2 Μέτρα διασποράς

Τα μέτρα διασποράς είναι η διακύμανση, η τυπική απόκλιση, το εύρος, το ενδοτεταρτομοριακό εύρος. Στα μέτρα μπορεί να περιληφθεί και ο συντελεστής μεταβλητότητας, ως μέτρο μεταβλητότητας.

Πίνακας 4.10. Μέτρα διασποράς

Μέτρα διασποράς	Ορισμός	Παρατηρήσεις
Διακύμανση	Για δείγμα n τιμών: Μη ομαδοποιημένα δεδομένα $s^2 = \frac{\sum(x-\overline{x})^2}{n-1}$ or $\sigma^2 = \frac{\sum x^2 - \frac{(\sum x)^2}{N}}{N}$ Ομαδοποιημένα δεδομένα $s^2 = \frac{\sum m^2 f - \frac{(\sum mf)^2}{n}}{n-1}$ or $s^2 = \frac{\sum f(m-\overline{x})^2}{n-1}$ Για πληθυσμό N τιμών: Μη ομαδοποιημένα δεδομένα $\sigma^2 = \frac{\sum(x-\mu)^2}{N}$ or $s^2 = \frac{\sum x^2 - \frac{(\sum x)^2}{n}}{n-1}$ Ομαδοποιημένα δεδομένα $\sigma^2 = \frac{\sum f(m-\mu)^2}{N}$ or $\sigma^2 = \frac{\sum m^2 f - \frac{(\sum mf)^2}{N}}{N}$	Υπολογίζεται ως ο μέσος των τετραγώνων των διαφορών από τον αριθμητικό μέσο. Η τιμή είναι στο τετράγωνο των μονάδων και αυτό την καθιστά δύσκολη στην ερμηνεία.
Τυπική απόκλιση	Για δείγμα $s = \sqrt{s^2}$ Για πληθυσμό $\sigma = \sqrt{\sigma^2}$	Υπολογίζεται ως η τετραγωνική τιμή της διακύμανσης και βρίσκεται στις ίδιες μονάδες με τα δεδομένα του δείγματος.

Εύρος	Εύρος = Μέγιστη – ελάχιστη τιμή	Η διαφορά μέγιστης και ελάχιστης τιμής σε ένα σύνολο τιμών.
Ενδοτεταρτομοριακό εύρος	IQR = Q_3 - Q_1	Η διαφορά μεταξύ τρίτου και δεύτερου τεταρτημορίων.
Συντελεστής μεταβλητότητας	Για δείγμα $\left[\frac{s}{\bar{x}} \text{ x } 100\right]$ % Για πληθυσμό $\left[\frac{\sigma}{\mu} \text{ x } 100\right]$ %	Παρέχει μέτρο της μεταβλητότητας σε σχέση με τον αριθμητικό μέσο. Επιτρέπει τη σύγκριση μεταξύ δειγμάτων ως προς τη μεταβλητότητα.

4.1.7.3 Μέτρα σχήματος κατανομής

Τα μέτρα σχήματος κατανομής περιλαμβάνουν τους συντελεστές ασυμμετρίας και κύρτωσης οι οποίοι χρησιμοποιούνται για τον έλεγχο της κανονικότητας μιας κατανομής δεδομένων.

Πίνακας 4.11. Μέτρα σχήματος κατανομής

Μέτρα σχήματος κατανομής	Ορισμός	Παρατηρήσεις
Ασυμμετρία	Για δείγμα τιμών $\beta_3 = \frac{1}{n}\frac{\sum(x-\bar{x})^3}{s^3}$ Για ομαδοποιημενα δεδομένα $\beta_3 = \frac{1}{n}\frac{\sum f(m-\bar{x})^3}{s^3}$	Χαρακτηρίζει την ασυμμετρία μιας κατανομής. • Για τιμές > 0 η ασυμμετρία είναι θετική (η ουρά της κατανομή αριστερά) • Για τιμές < 0 η ασυμμετρία είναι αρνητική (η ουρά της κατανομή δεξιά) • Για τιμές = 0 η κατανομή είναι συμμετρική
Κύρτωση	Για δείγμα τιμών $\beta_4 = \frac{1}{n}\frac{\sum(x-\bar{x})^4}{s^4}$ Για ομαδοποιημενα δεδομένα $\beta_4 = \frac{1}{n}\frac{\sum(x-\bar{x})^4}{s^4}$	Χαρακτηρίζει την οξύτητα της κορυφής της κατανομής. • Για τιμές > 3 η κατανομή είναι λεπτόκυρτη (οξεία) • Για τιμές < 3 η κατανομή είναι πλατύκυρτη (πλατειά)

		• Για τιμές = 0 η κατανομή προσεγγίζει την κανονική κατανομή

4.1.7.4 Μέτρα θέσης

Τέλος υπολογίζονται τα μέτρα θέσης τα οποία δίνουν μια εικόνα της κατανομής των τιμών, και διακρίνονται στα εκατοστημόρια, τεταρτημόρια, και ενδοτεταρτομοριακό εύρος.

Πίνακας 4.12. Μέτρα θέσης

Μέτρα διασποράς	Ορισμός	Παρατηρήσεις
Εκατοστημόρια	Ένα εκατοστημόριο είναι ένα ποσοστιαίο σημείο α% το οποίο χωρίζει ένα σύνολο δεδομένων σε δύο τμήματα, όπου τουλάχιστον α% τιμές είναι μικρότερες από αυτό και (1-α)% υψηλότερες.	Προϋπόθεση είναι το σύνολο να είναι διατεταγμένο.
Τεταρτημόρια	Q_1 Q_2 Q_3	Τα εκατοστημόρια για ποσοστό 25%. 50%, 75% αντίστοιχα. Το δεύτερο τεταρτημόριο είναι η διάμεσος, και το κάθε ένα από τα άλλα δύο είναι η διάμεσος των τιμών που είναι μικρότερες και υψηλότερες της διαμέσου.
Ενδοτεταρτομοριακό εύρος	IQR = Q_3 - Q_1	Η διαφορά μεταξύ τρίτου και δεύτερου τεταρτημορίων.

4.2. Τυχαίες μεταβλητές, συναρτήσεις κατανομών.

Οι μετρήσεις, ως αποτέλεσμα παρατηρήσεων ή εκτέλεση πειραμάτων αποτελούν την βασική πηγή συλλογής δεδομένων. Στα πειράματα τύχης η κάθε τιμή συνδέεται με μια πιθανότητα εμφάνισης,

καθώς η επανάληψη του πειράματος με ίδιες αρχικές συνθήκες δεν οδηγεί στο ίδιο αποτέλεσμα. Οι ενδεχόμενες τιμές αποτελούν στοιχεία του δειγματικού χώρου του πειράματος. Επειδή οι δειγματικοί χώροι διαφέρουν σημαντικά ως προς τους τύπους δεδομένων και την ποικιλία τους, ορίζεται η έννοια της τυχαίας μεταβλητής η οποία επιτρέπει τη εξέταση των πειραμάτων τύχης με ενιαίο τρόπο και τη θεωρητική θεμελίωση.

Ένα μέγεθος ονομάζεται τυχαία μεταβλητή όταν, στα διάφορα πειράματα τύχης που διεξάγονται κάτω από τις ίδιες συνθήκες, λαμβάνει εντελώς τυχαίες τιμές, δηλαδή δεν είναι εφικτή η πρόβλεψη της τιμής εκ των προτέρων. Η τυχαία μεταβλητή είναι εξ ορισμού μια συνάρτηση ορισμένη στον εκάστοτε δειγματικό χώρο. Συνήθως, συμβολίζεται με ένα κεφαλαίο γράμμα X, Y, Z, ενώ οι τιμές της συμβολίζονται με τα αντίστοιχα μικρά γράμματα x, y, z . Γενικά, μία τυχαία μεταβλητή λαμβάνει πεπερασμένο ή άπειρο πλήθος τιμών. Στην πρώτη περίπτωση ονομάζεται διακριτή τυχαία μεταβλητή, ενώ στη δεύτερη, συνεχής τυχαία μεταβλητή.

Για παράδειγμα, η μεταβλητή X που καταγράφει το αποτέλεσμα ρίψης ενός ζαριού λαμβάνει μόνο τις διακριτές τιμές $x = 1,2,3,4,5,6$. Είναι επομένως η μεταβλητή X διακριτή τυχαία μεταβλητή. Αντίθετα, μια μεταβλητή η οποία καταγράφει την ταχύτητα ενός αυτοκινήτου μπορεί να λάβει οποιαδήποτε τιμή και είναι συνεχής τυχαία μεταβλητή.

Ένα τυχαίο μέγεθος είναι εντελώς χαρακτηρισμένο εάν, εκτός από το πλήθος των τιμών τις οποίες μπορεί να λάβει, είναι γνωστή και η πιθανότητα κάθε τιμής. Με αυτό τον τρόπο, τα ζεύγη τιμής και πιθανότητας μπορούν να χρησιμοποιηθούν για τον ορισμό μιας συνάρτησης πιθανότητας ή συνάρτησης κατανομής.

4.2.1. Συνάρτηση πιθανότητας τυχαίας μεταβλητής

Έτσι, αν μία διακριτή τυχαία μεταβλητή X μπορεί να λάβει τις τιμές $x1, x2, \dots, xn$ και υποθέσουμε ότι οι πιθανότητες να λάβει η μεταβλητή τις τιμές αυτές είναι αντιστοίχως $P(x1), P(x2), .., P(\ xn)$ ή $P(xk) = f(xk)$ μπορούμε να ορίσουμε μία συνάρτηση πιθανότητας ή συνάρτηση κατανομής $P(X = x) = f(x)$ η οποία για διακριτή τυχαία μεταβλητή πρέπει να ικανοποιεί τις συνθήκες

$P(X \in S) = \sum_{x \in S} P_X(x)$ για κάθε σύνολο S

$\sum_{x \in X:} P_X(x) = 1$ για όλες τις δυνατές τιμές x

$P_X(x) \geq 0$ για κάθε x

Τα ζεύγη τιμών $(x, P(X = x))$ αποτελούν την κατανομή πιθανότητας της διακριτής μεταβλητής X.

4.2.2. Αριθμητικά χαρακτηριστικά (Μέση Τιμή - Διασπορά – Ροπές).

Για τις τυχαίες μεταβλητές, ισχύουν τα εξής.

Πίνακας 4.13. Μέση τιμή, διασπορά, τυπική απόκλιση

Αναμενόμενη Τιμή (Μέση Τιμή)	$E(X)=\mu=\sum_x xP(x)=\sum_x xf(x)$
Διασπορά	$V(X)=\sigma^2=E(X-\mu)^2=\sum_x (x-\mu)^2 f(x)=E(X^2)-\mu^2$
Τυπική Απόκλιση	$\sigma=\sqrt{\sigma^2}$

4.3. Κατανομές πιθανότητας για διακριτές τυχαίες μεταβλητές

Βασικές κατανομές για διακριτές τυχαίες μεταβλητές αποτελούν η κατανομή Bernoulli, η διωνυμική κατανομή, η κατανομή Poisson, η γεωμετρική και η υπεργεωμετρική κατανομή. Ενδεικτικά παρουσιάζονται τα στοιχεία της κατανομής Bernoulli και της κατανομή Poisson. Να τονιστεί ότι οι συγκεκριμένες κατανομές είναι θεωρητικές, δηλαδή συγκεκριμένες συναρτήσεις πιθανότητας, οι οποίες δεν εμφανίζονται ως έχουν στα δεδομένα του πραγματικού κόσμου παρά μόνο προσεγγιστικά. Ο στόχος της στατιστικής ανάλυσης είναι η εύρεση της κατανομής των πραγματικών δεδομένων, ώστε αν υπάρχει προσέγγιση με κάποια θεωρητική κατανομή, να αξιοποιηθεί η θεωρητική κατανομή για τη μελέτη των πραγματικών δεδομένων ως προς τη συμπερασματολογία.

4.3.1. Κατανομή Bernoulli.

Η κατανομή Bernoulli χαρακτηρίζει πειράματα τύχης όπου το αποτέλεσμα είναι είτε επιτυχές με πιθανότητα p, ή μη επιτυχές με πιθανότητα 1-p. Το ζητούμενο είναι η εύρεση του αριθμού των επιτυχιών στο πείραμα.

Πίνακας 4.14. Διωνυμική κατανομή

Διωνυμική Κατανομή	**Περιγραφή**
Συμβολισμός	$X \sim B(n,p)$

Συνάρτηση πιθανότητας	$f(x)=\binom{n}{x}p^x q^{n-x}, x=0,1,2,\ldots,n\ ,\ 0\le p\le 1$ $E(X)=np$ Μέση Τιμή της τ.μ. $V(X)=npq$ Διασπορά της τ.μ.

4.3.2. Κατανομή Poisson.

Η κατανομή Poisson χαρακτηρίζει πειράματα τύχης στα οποία εμφανίζονται ενδεχόμενα με τυχαίο τρόπο σε συγκεκριμένα συνεχή χρονικά διαστήματα.

Πίνακας 4.15. Κατανομή poisson

Κατανομή Poisson	Περιγραφή
Συμβολισμός	$X \sim P(\lambda)$
Συνάρτηση πιθανότητας	$f(x)=P(X=x)=\frac{e^{-\lambda}\lambda^x}{x!}, x=0,1,2,\ldots$ και $\lambda>0$ $E(X)=\lambda$ $V(X)=\lambda$

4.4. Κατανομές πιθανότητας για συνεχείς τυχαίες μεταβλητές

Βασικές κατανομές για συνεχείς τυχαίες μεταβλητές αποτελούν η ομοιόμορφη κατανομή, η κανονική κατανομή, η εκθετική κατανομή, όπως και οι κατανομές χ^2, t Student και F. Ενδεικτικά παρουσιάζονται τα στοιχεία της ομοιόμορφης κατανομής, της κανονικής και των χ^2, t Student και F. Οι συγκεκριμένες κατανομές είναι θεωρητικές, δηλαδή συγκεκριμένες συναρτήσεις πιθανότητας, οι οποίες δεν εμφανίζονται ως έχουν στα δεδομένα του πραγματικού κόσμου παρά μόνο προσεγγιστικά. Ο στόχος της στατιστικής ανάλυσης είναι η εύρεση της κατανομής των πραγματικών δεδομένων, ώστε αν υπάρχει προσέγγιση με κάποια θεωρητική κατανομή, να αξιοποιηθεί η θεωρητική κατανομή για τη μελέτη των πραγματικών δεδομένων ως προς τη συμπερασματολογία. Η κανονική κατανομή είναι θεμελιώδους σημασίας για τη στατιστική συμπερασματολογία, όπως και οι κατανομές χ^2, t Student και F.

4.4.1. Ομοιόμορφη κατανομή

Χαρακτηρίζει ισοπίθανα ενδεχόμενα.

Πίνακας 4.16. Ομοιόμορφη κατανομή

Ομοιόμορφη κατανομή	Περιγραφή	Παρατηρήσεις
Συμβολισμός	$X \sim U(a,b)$	Η απλούστερη μορφή συνεχούς κατανομής πιθανότητας
Συνάρτηση πιθανότητας	$f(x) = \begin{cases} \frac{1}{b-a}, a \le x \le b \\ 0, \quad \text{αλλού} \end{cases}$ $E(X) = \frac{a+b}{2}$ $V(X) \frac{(b-a)^2}{12}$	

4.4.2. Κανονική κατανομή

Αποτελεί τη σημαντικότερη κατανομή και θεμέλιο της στατιστικής συμπερασματολογίας.

Πίνακας 4.17. Κανονική κατανομή

Κανονική Κατανομή	Περιγραφή	Παρατηρήσεις
Συμβολισμός	$X \sim N(\mu, \sigma^2)$	Η σημαντικότερη κατανομή πιθανότητας
Συνάρτηση πιθανότητας	$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ Όπου $f(x)$ το ύψος της κατανομής, μ ο μέσος, σ^2 η διασπορά, σ η τυπική απόκλιση, $e = 2,7183$ και $\pi = 3,1416$ $E(X) = \mu$ $V(X) = \sigma^2$	

4.4.2.1 Παράδειγμα

Σε μια έρευνα ενός πολυκαταστήματος διαπιστώθηκε ότι ο χρόνος που οι πελάτες παραμένουν στο κατάστημα ακολουθεί κανονική κατανομή με τυπική απόκλιση 5 λεπτά. Λαμβάνεται ένα τυχαίο δείγμα n πελατών. Πόσο μεγάλο πρέπει να είναι το δείγμα ώστε η πιθανότητα η μέση τιμή του δείγματος να διαφέρει λιγότερο από μια ώρα από τη μέση τιμή του πληθυσμού να είναι τουλάχιστον 0,90;

Λύση

Έστω X= αριθμός των λεπτών που βρίσκονται οι πελάτες στο κατάστημα, τότε X~N(μ,5^2).

Το ζητούμενο είναι να υπολογιστεί n ώστε

$P(|\bar{X} - \mu| < 1) \geq 0{,}90.$

Ισχύει ότι $\bar{X} \sim N\left(\mu, \frac{5^2}{n}\right)$, οπότε

$$P(|\bar{X} - \mu| < 1) = P\left(\frac{|\bar{X} - \mu|}{5/\sqrt{n}} < \frac{1}{5/\sqrt{n}}\right) = P\left(|Z| < \frac{1}{5/\sqrt{n}}\right)$$

$$= P\left(|Z| < \frac{\sqrt{n}}{5}\right).$$

Από το πίνακα της τυποποιημένης κανονικής κατανομής αναζητείται ο αριθμός που αφήνει 5% του εμβαδού δεξιά, δηλαδή 1,645. Ο αριθμός είναι ίσος με $\frac{\sqrt{n}}{5} = 1{,}645$ οπότε n=68 με επίλυση.

4.4.3. Τυποποιημένη κανονική κατανομή

Αποτελεί κανονική κατανομή με μ=0 και σ=1. Με βάση αυτή υπολογίζονται οι πίνακες υπολογισμού και οι τιμές της κανονικής κατανομής με κατάλληλη αναγωγή.

Πίνακας 4.18. Τυποποιημένη κανονική κατανομή

Τυποποιημένη κανονική κατανομή	**Περιγραφή**	**Παρατηρήσεις**
Συμβολισμός	$X \sim N(\mu, \sigma^2)$	Η τυποποιημένη κανονική κατανομή
Συνάρτηση πιθανότητας	$Z \sim N(0,1)$ $f(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}$	

	$E(z)=0$ $V(z)=1$	

4.4.3.1 Παράδειγμα

Ένα εργοστάσιο κατασκευάζει λάμπες με μέση διάρκεια ζωής 25000 λεπτά και σ^2=5000 λεπτά. Αν ληφθεί ένα δείγμα 100 λαμπτήρων, ποια είναι η πιθανότητα η μέση τιμή του δείγματος να είναι κάτω από 30000 λεπτά?

Λύση

Με την προϋπόθεση ότι ισχύει η κανονική κατανομή, ισχύει ότι,

Έτσι μ_n=25, σ^2=5 και n=0,1 (σε χιλιάδες).

Το δείγμα n=0,1 χιλιάδες ακολουθεί κανονική κατανομή, οπότε ισχύει $\mu_n=\mu_x$=25. Η διακύμανση του δείγματος είναι ίση με 5/0,1=50.

Συνεπώς z= (x-μ)/σ= (30-25)/7,07=0,709

Οπότε P(x<30)=0,7852

4.4.4. Κατανομές χ^2, t Student, F

Οι κατανομές αυτές είναι πολύ χρήσιμες στη στατιστική συμπερασματολογία.

Πίνακας 4.19. Κατανομής χ2, t Student, F

Χρήσιμες κατανομές	Περιγραφή	Παρατηρήσεις
Κατανομή χ2	Αν $Z_1, Z_1, .., Z_n$είναι n ανεξάρτητες τυποποιημένες κανονικές τυχαίες μεταβλητές, δηλαδή, αν $Z_i \sim N(0,1)$, τότε η κατανομή της τυχαίας μεταβλητής $X = Z_1^2 + Z_2^2 + \cdots Z_n^2$. ονομάζεται κατανομή χι-τετράγωνο με n βαθμούς ελευθερίας και συμβολίζεται με χ_n^2.	Η μορφή της εξαρτάται από τους βαθμούς ελευθερίας. Ασύμμετρη θετικά.

Κατανομή t Student	Έστω Z μια τυχαία μεταβλητή η οποία ακολουθεί την τυποποιημένη κανονική κατανομή, δηλαδή) $Z \sim N(0,1)$, και S^n μια τυχαία μεταβλητή ανεξάρτητη από την Z η οποία ακολουθεί την κατανομή χ_n^2, δηλαδή $S \sim \chi_n^2$. Τότε, η κατανομή της τυχαίας μεταβλητής, $T = \frac{Z}{\sqrt{\frac{S}{n}}}$ ονομάζεται κατανομή t ή κατανομή Student με n βαθμούς ελευθερίας και συμβολίζεται με t_n.	Η μορφή της εξαρτάται από τους βαθμούς ελευθερίας. Συμμετρική γύρω από τη μέση τιμή. Για υψηλό αριθμό βαθμών ελευθερίας, προσεγγίζει την κανονική κατανομή.
Κατανομή F	Έστω S_n, S_m δύο ανεξάρτητες τυχαίες μεταβλητές οι οποίες ακολουθούν τις κατανομές χ_n^2 και χ_m^2 αντίστοιχα. Τότε, η κατανομή της τυχαίας μεταβλητής, $F = \frac{S_n/n}{S_m/m}$ ονομάζεται κατανομή F με n και m βαθμούς ελευθερίας και συμβολίζεται με $F_{n,m}$.	Η μορφή της εξαρτάται από τους βαθμούς ελευθερίας. Ασύμμετρη θετικά.

4.5. Συνδιακύμανση, συντελεστής συσχέτισης.

Στην περίπτωση κατά την οποία το ενδιαφέρον εστιάζει στο κατά πόσο δύο τυχαίες μεταβλητές συσχετίζονται, όχι με αιτιακή σχέση, αλλά ως μεγέθη τα οποία μεταβάλλονται με κοινό τρόπο, το ερώτημα που εξετάζεται είναι αν υπάρχει συσχέτιση, ποιος είναι ο βαθμός της και η κατεύθυνσή της.

Η ανάλυση συσχέτισης δίνει απαντήσεις στα ερωτήματα και γενικά δεν θεωρείται κάποια διάκριση μεταξύ των μεταβλητών. Δηλαδή θεωρούνται και οι δύο τυχαίες (μη ελεγχόμενες από τον ερευνητή), και δεν διαχωρίζονται σε ανεξάρτητη - εξαρτημένη. Και οι δύο έχουν την ίδια βαρύτητα και η σχέση τους είναι συμμετρική. Στην ανάλυση συσχέτισης το ενδιαφέρον είναι στον υπολογισμό ενός αριθμητικού μέτρου το οποίο εκφράζει το βαθμό στον οποίο οι μεταβλητές σχετίζονται μεταξύ

τους. Στην απλή περίπτωση δύο τυχαίων μεταβλητών, η συσχέτιση μπορεί να εκφραστεί τον Συντελεστή Συσχέτισης (Correlation Coefficient).

Υπάρχουν κυρίως δύο Συντελεστές Συσχέτισης, ο συντελεστής του Pearson και ο συντελεστής του Spearman. Ο Συντελεστής του Pearson χρησιμοποιείται όταν τα δεδομένα είναι συνεχή, αλλά παρουσιάζουν ταυτόχρονα και σχετική κανονικότητα (συμμετρικότητα δηλαδή). Η συμμετρία των δεδομένων μπορεί να ελεγχθεί με ένα ιστόγραμμα ή θηκόγραμμα. Ονομάζεται επίσης και Συντελεστής Γραμμικής Συσχέτισης. Ο Συντελεστής του Spearman, χρησιμοποιείται σε όλες τις άλλες περιπτώσεις, καθώς και στην περίπτωση διακριτών δεδομένων, αλλά μπορεί να χρησιμοποιηθεί και στα ποιοτικά διατάξιμα χαρακτηριστικά.

Πίνακας 4.20. Συντελεστής συσχέτισης

Συντελεστής συσχέτισης	**Ορισμός**	**Παρατηρήσεις**
Pearson	$r = \frac{\sum(X-\bar{X})(Y-\bar{Y})}{\sqrt{\sum(X-\bar{X})^2 \cdot \sum(Y-\bar{Y})^2 \cdot}}$	Ο συντελεστής συσχέτισης του Pearson, παίρνει τιμές στο διάστημα [-1,1]. Όταν είναι θετικός (δηλαδή από το 0 μέχρι το +1), υπάρχει Θετική Συσχέτιση, ενώ όταν είναι αρνητικός (από -1 μέχρι το 0), υπάρχει Αρνητική Συσχέτιση. Όσο πιο κοντά είναι τα δεδομένα σε ευθεία γραμμή, τόσο πιο υψηλός είναι ο συντελεστής (πλησιάζει τον αριθμό 1, ή -1) και η συσχέτιση ισχυρότερη (θετική ή αρνητική). Όταν ο συντελεστής συσχέτισης είναι ίσος με το 0 (r=0), τότε τα χαρακτηριστικά ΔΕΝ έχουν καμία συσχέτιση, και ονομάζονται

		ασυσχέτιστα ή ανεξάρτητα.
Spearman	$\rho = 1 - \frac{6\sum d^2}{n(n^2-1)}$	Έχει παρόμοιες ιδιότητες με αυτόν του Pearson, δηλαδή είναι κι αυτός αρνητικός, μηδέν ή θετικός, στην κλίμακα από το -1 μέχρι το +1. Χρησιμοποιείται όταν έχουμε ποιοτικά διατάξιμα χαρακτηριστικά (ή απαντήσεις στην κλίμακα 1-2-3-4-5 – καθόλου, λίγο, μέτρια, πολύ, πάρα πολύ), αλλά μπορεί να χρησιμοποιηθεί και στα ποσοτικά, κυρίως σε περιπτώσεις μη συμμετρικών κατανομών.

4.6. Κεντρικό οριακό θεώρημα.

Αποτελεί ένα από τα σημαντικότερα θεωρήματα της στατιστικής και σε πολύ απλή διατύπωση αναφέρει ότι αν σε ένα πληθυσμό με άγνωστη κατανομή ληφθούν μια σειρά τυχαίων δειγμάτων, τότε ο μέσος των μέσων των δειγμάτων (δειγματικός μέσος) ακολουθεί κανονική κατανομή (κατά προσέγγιση). Η θεμελιώδης σημασία του θεωρήματος για τη στατιστική συμπερασματολογία έγκειται στο ότι ανεξάρτητα από την κατανομή του πληθυσμού, αν ληφθεί ένα δείγμα σχετικά μεγάλο (>30), τότε μπορεί να χρησιμοποιηθεί η κανονική κατανομή για συμπερασματολογία ως προς τον πληθυσμό.

4.6.1.1 Κ.Ο.Θ. για δειγματικό μέσο

Ειδικότερα, για δείγμα $X_1, X_2, \dots, X_n$ προερχόμενο από κάποιο πληθυσμό με μέση τιμή μ και διακύμανση σ^2, αν $\bar{X}$ και S^2 η μέση τιμή και μεταβλητότητα του δείγματος που υπολογίζονται ως εξής:

$$\bar{X} = \frac{\sum X_i}{n}, \quad S^2 = \frac{\sum X_i^2 - n\bar{X}^2}{n-1}.$$

τότε, ανεξάρτητα από την κατανομή του πληθυσμού, ισχύει ότι

$$E(\bar{X}) = \mu, \quad var(\bar{X}) = \frac{\sigma^2}{n}.$$

Ειδικές περιπτώσεις υπάρχουν για την περίπτωση όπου ο πληθυσμός ακολουθεί κανονική, όταν η μεταβλητότητα του πληθυσμού υποτίθεται ότι είναι γνωστή, ή όταν είναι άγνωστη.

Αν το δείγμα είναι μεγάλο ($n > 30$), τότε, ανεξαρτήτως της κατανομής του πληθυσμού η τυχαία μεταβλητή $\frac{\bar{X}-\mu}{S/\sqrt{n}} \sim N(0,1)$ ακολουθεί την τυποποιημένη κανονική κατανομή.

4.7. Εκτιμητική

Στη στατιστική έρευνα στις περισσότερες περιπτώσεις ο υπό μελέτη πληθυσμός εκπροσωπείται από ένα δείγμα το οποίο αποτελεί το αντικείμενο μελέτης. Το σύνολο των υπολογισμών εκτελείται στα δεδομένα του δείγματος, ωστόσο ο στόχος είναι η αξιόπιστη εκτίμηση των παραμέτρων του πληθυσμού. Η εκτιμητική, αποτελεί τον κλάδο της στατιστικής συμπερασματολογίας η οποία έχει ως αντικείμενο την εκτίμηση των παραμέτρων του πληθυσμού. Οι εκτιμήσεις είναι είτε σημειακές, δηλαδή μια τιμή, ή ένα διάστημα εντός του οποίου αναμένεται να βρίσκεται η τιμή τη παραμέτρου του πληθυσμού. Καθώς είναι αδύνατη η βέβαιη εκτίμηση, υπάρχει πάντοτε ένας βαθμός εμπιστοσύνης ως προς την εκτίμηση. Οι παράμετροι του πληθυσμού οι οποίοι ενδιαφέρουν γενικά είναι η μέση τιμή και το ποσοστό ενός πληθυσμού ή η σύγκριση αυτών των τιμών μεταξύ δύο πληθυσμών. Στην εκτιμητική βασικό ρόλο έχει το κεντρικό οριακό θεώρημα, καθώς επιτρέπει την χρήση των ιδιοτήτων της κανονικής κατανομής για την εκτίμηση διαστημάτων εμπιστοσύνης ακόμη και για πληθυσμούς των οποίων η κατανομή είναι άγνωστη.

Πίνακας 4.21. Εκτιμητική

Εκτίμηση	Ορισμός	Παρατηρήσεις
Σημειακή	Υπολογισμός της τιμής μιας στατιστικής συνάρτησης με τιμές του δείγματος.	Υπάρχουν συνήθως αρκετές εκτιμήτριες συναρτήσεις.
Διάστημα εμπιστοσύνης	Υπολογισμός των άκρων ενός διαστήματος, με χρήση στατιστικής συνάρτησης με τιμές του δείγματος, εντός του οποίου η τιμή της παραμέτρου του πληθυσμού θα βρίσκεται με ορισμένο βαθμό	Ο βαθμός εμπιστοσύνης εκφράζει πιθανότητα και επηρεάζει το εύρος του διαστήματος. Υψηλότερος βαθμός εμπιστοσύνης, ευρύτερο διάστημα.

	εμπιστοσύνης.	

4.7.1. Σημειακή εκτίμηση

Η σημειακή εκτίμηση της τιμής μιας παραμέτρου ενός πληθυσμού εκτελείται με τον υπολογισμό της τιμής μιας στατιστικής συνάρτησης, δηλαδή μιας κατάλληλης συνάρτησης η οποία χρησιμοποιεί τις τιμές του δείγματος. Η εκτίμηση μπορεί να αφορά τη μέση τιμή, το ποσοστό ή άλλη παράμετρο. Η σημειακή εκτίμηση ωστόσο εμπεριέχει πάντοτε κάποιο σφάλμα λόγω της δειγματοληψίας.

4.7.2. Εκτίμηση με διάστημα εμπιστοσύνης για μέσο ενός πληθυσμού και σύγκριση μέσων δύο πληθυσμών, ποσοστό ενός πληθυσμού και σύγκριση ποσοστών δύο πληθυσμών

Σε σχέση με τη σημειακή εκτίμηση, είναι προτιμότερη η εκτίμηση ενός διαστήματος το οποίο θα περιλαμβάνει την τιμή της παραμέτρου του πληθυσμού με κάποια πιθανότητα. Επειδή όμως και πάλι δεν είναι εφικτός ο υπολογισμός του πόσο καλή είναι η εκτίμηση αυτή, χρησιμοποιείται ένας βαθμός εμπιστοσύνης, ο οποίος ορίζεται από τον ερευνητή με βάση το πρόβλημα, ο οποίος εκφράζει την επιθυμητή ακρίβεια της εκτίμησης. Ο βαθμός εμπιστοσύνης είναι συνήθως της τάξης του 95%, οπότε το διάστημα ονομάζεται '95%-Διάστημα Εμπιστοσύνης' (95%-Confidence Interval). Η επιλογή του εύρους του διαστήματος, δεν είναι αυθαίρετη, αλλά βασίζεται σε έναν απλό μαθηματικό τύπο. Οπότε με την εκτίμηση μέσω διαστήματος εμπιστοσύνης έχουμε 95% εμπιστοσύνη ότι το πραγματικό μέγεθος που ενδιαφέρει στον πληθυσμό, βρίσκεται εντός του διαστήματος, αλλά υπάρχει μια πιθανότητα μικρότερη από 5%. το πραγματικό ποσοστό να μην βρίσκεται μέσα εκεί.

Πίνακας 4.22. Σημειακή εκτίμηση

Διάστημα εμπιστοσύνης για ένα δείγμα	**Ορισμός**	**Παρατηρήσεις**
Μέση τιμή	Για μεγάλα δείγματα (>30), ένα $100(1-\alpha)\%$ διάστημα εμπιστοσύνης για την μέση τιμή του πληθυσμού δίνεται από τον τύπο	Ο βαθμός εμπιστοσύνης $100(1-\alpha)\%$ ορίζεται από τον ερευνητή και εκφράζει την επιθυμητή πιθανότητα

	$\bar{X} - z_{1-\frac{\alpha}{2}} \frac{S}{\sqrt{n}} \leq \mu \leq \bar{X} + z_{1-\frac{\alpha}{2}} \frac{S}{\sqrt{n}}$. Για πληθυσμούς που ακολουθούν κανονική κατανομή έχουμε: 1. Όταν η μεταβλητότητα είναι γνωστή, $\bar{X} - z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}} \leq \mu \leq \bar{X} + z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}}$. 2. Όταν το δείγμα είναι μικρό, $\bar{X} - t_{\frac{\alpha}{2},n-1} \frac{S}{\sqrt{n}} \leq \mu \leq \bar{X} + t_{\frac{\alpha}{2},n-1} \frac{S}{\sqrt{n}}$.	να βρίσκεται η παράμετρος του πληθυσμού εντός του διαστήματος. Χαμηλός βαθμός εμπιστοσύνης, παράγει στενό διάστημα, υψηλός βαθμός εμπιστοσύνης, παράγει ευρύ διάστημα.
Ποσοστό	Για μεγάλα δείγματα (>30), ένα $100(1-\alpha)\%$ διάστημα εμπιστοσύνης για το ποσοστό κάποιας κατηγορίας στον πληθυσμό δίνεται από τον τύπο $\hat{p} - z_{1-\frac{\alpha}{2}} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \leq p \leq \hat{p} + z_{1-\frac{\alpha}{2}} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$.	

Διάστημα εμπιστοσύνης για δύο δείγματα	**Ορισμός**	**Παρατηρήσεις**
Μέση τιμή	Για μεγάλα δείγματα (>30), υποθέτοντας πληθυσμούς με ίσες μεταβλητότητες, ένα $100(1-\alpha)\%$ διάστημα εμπιστοσύνης για την διαφορά των μέσων τιμών δύο πληθυσμών δίνεται από τον τύπο $\bar{X}_1 - \bar{X}_2 - z_{1-\frac{\alpha}{2}} S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}} \leq$ $\mu_1 - \mu_2 \leq \bar{X}_1 - \bar{X}_2 +$ $z_{1-\frac{\alpha}{2}} S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$. Για πληθυσμούς που ακολουθούν	Ο βαθμός εμπιστοσύνης $100(1-\alpha)\%$ ορίζεται από τον ερευνητή και εκφράζει την επιθυμητή πιθανότητα να βρίσκεται η παράμετρος του πληθυσμού εντός του διαστήματος. Χαμηλός βαθμός εμπιστοσύνης, παράγει στενό διάστημα, υψηλός βαθμός εμπιστοσύνης, παράγει ευρύ

	κανονική κατανομή έχουμε: 1. Όταν οι μεταβλητότητες είναι γνωστές, $\bar{X}_1 - \bar{X}_2 - z_{1-\frac{\alpha}{2}}\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}} \leq$ $\mu_1 - \mu_2 \leq \bar{X}_1 - \bar{X}_2 +$ $z_{1-\frac{\alpha}{2}}\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}.$ 2. Όταν τα δείγματα είναι μικρά, $\bar{X}_1 - \bar{X}_2 -$ $t_{\frac{\alpha}{2}, n_1+n_2-2} S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}} \leq \mu_1 -$ $\mu_2 \leq \bar{X}_1 - \bar{X}_2 +$ $t_{\frac{\alpha}{2}, n_1+n_2-2} S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}.$	διάστημα.
Ποσοστό	Για μεγάλα δείγματα (>30), ένα $100(1-\alpha)\%$ διάστημα εμπιστοσύνης για την διαφορά των ποσοστών κάποιας κατηγορίας σε δύο πληθυσμούς δίνεται από τον τύπο $\hat{p}_1 - \hat{p}_2 - z_{1-\frac{\alpha}{2}}\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2}} \leq$ $p_1 - p_2 \leq \hat{p}_1 - \hat{p}_2 +$ $z_{1-\frac{\alpha}{2}}\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2}}.$	

4.7.2.1 Παράδειγμα 1

Η κατανομή της ταχύτητας των αυτοκινήτων σε μια διασταύρωση είναι η κανονική χωρίς να είναι γνωστή η διακύμανση. Σε ένα δείγμα 14 αυτοκινήτων βρέθηκε ότι η μέση ταχύτητα ήταν $\overline{X}$ = 52,52 χλμ/ώρα και η τυπική απόκλιση S = 3,37 χλμ/ώρα. Να υπολογιστεί το διάστημα εμπιστοσύνης του μέσου μ σε επίπεδο σημαντικότητας 5%.

Λύση

Από τους πίνακες της κατανομής Student με βαθμούς ελευθερίας n-1=14-1 και επίπεδο σημαντικότητας α=5% είναι t_a = 2,16. Συνεπώς το 95% διάστημα εμπιστοσύνης για τη μέση ταχύτητα είναι:

$\overline{X}$ - ta (S/$\sqrt{n}$) ≤ μ ≤ $\overline{X}$ + ta (S/$\sqrt{n}$) ή

52,52 – 2,16 (3,37/3,74) ≤ μ ≤ 52,52 + 2,16 (3,37/3,74), ή

50,58 ≤ μ ≤ 54,47

4.7.2.2 Παράδειγμα 2

Σε ένα δείγμα 100 προϊόντων από μια μηχανή βρέθηκε ότι 24 είναι ελαττωματικά. Σε ένα δείγμα 100 προϊόντων από μια δεύτερη μηχανή βρέθηκε ότι 13 είναι ελαττωματικά. Να βρεθεί το διάστημα εμπιστοσύνης με πιθανότητα 95%, η διαφορά μεταξύ της αναλογίας ελαττωματικών προϊόντων στις δύο μηχανές.

Λύση

Το διάστημα εμπιστοσύνης δίνεται από τη σχέση:

$(P_1 - P_2) - Z_\alpha \sqrt{(p1q1)/n1 + (p2q2)/n2} \leq P_1 - P_2 \leq (P_1 - P_2) + Z_\alpha \sqrt{(p1q1)/n1 + (p2q2)/n2}$

P1 = 24/100 = 0,24, P2 = 13/100 = 0,13, P1 – P2 = 0,11

Συνεπώς 0,11 – 1,96(0,054) ≤ $P_1 - P_2$ ≤ 0,11 +1,96(0,054) ή

0,004 ≤ $P_1 - P_2$ ≤ 0,21

4.8. Στατιστικοί έλεγχοι (έλεγχος υποθέσεων για μέσο ενός πληθυσμού και σύγκριση μέσων δύο πληθυσμών, ποσοστό ενός πληθυσμού και σύγκριση ποσοστών δύο πληθυσμών)

Στην στατιστική έρευνα το ενδιαφέρον στρέφεται σε έναν ή περισσότερους πληθυσμούς οι οποίοι εκπροσωπούνται από αντίστοιχα δείγματα, τα οποία αποτελούν την πηγή δεδομένων για τους υπολογισμούς με στόχο την αξιόπιστη εκτίμηση των παραμέτρων του πληθυσμού. Εκτός της εκτίμησης των τιμών για παραμέτρους πληθυσμών, ένα πολύ σημαντικό σκέλος της στατιστικής έρευνας αφορά στην διερεύνηση ερωτημάτων για παραμέτρους πληθυσμών και την εξέταση του

αν ισχύουν ή όχι. Ο έλεγχος στατιστικών υποθέσεων αποτελεί τον κλάδο της στατιστικής συμπερασματολογίας, ο οποίος έχει ως αντικείμενο τον έλεγχο τέτοιων ερωτημάτων, των στατιστικών υποθέσεων, και την εξαγωγή συμπερασμάτων σχετικά με τις τιμές των παραμέτρων του πληθυσμού που ερευνάται. Οι έλεγχοι εμπεριέχουν πάντοτε ένα ποσοστό αβεβαιότητας οπότε υπάρχει πάντοτε ένας βαθμός εμπιστοσύνης ως προς τον έλεγχο. Οι παράμετροι του πληθυσμού οι οποίοι ενδιαφέρουν συνηθέστερα είναι η μέση τιμή και το ποσοστό ενός πληθυσμού ή η σύγκριση αυτών των τιμών μεταξύ δύο πληθυσμών.

4.8.1. Στατιστικές υποθέσεις

Ειδικότερα, ο έλεγχος υποθέσεων είναι μια συστηματική στατιστική διαδικασία με την οποία ελέγχονται υποθέσεις που αφορούν στον υπό μελέτη πληθυσμό, και συγκεκριμένα τις παραμέτρους του, με την επιλογή δειγμάτων μέσα από τον πληθυσμό. Για τον έλεγχο χρησιμοποιούνται τα δεδομένα ενός δείγματος το οποίο έχει επιλεγεί από τον πληθυσμό (με τυχαία δειγματοληψία υποχρεωτικά), προκειμένου να αποφασιστεί αν θα απορριφθεί μία υπόθεση ή όχι.

Μια στατιστική υπόθεση είναι μια πρόταση για μια ή περισσότερες παραμέτρους του πληθυσμού. Η πρόταση αυτή μπορεί να είναι αληθής ή όχι και διατυπώνεται πάντοτε ως ζεύγος αλληλοαποκλειόμενων προτάσεων οι οποίες καλύπτουν πλήρως τις δυνατές εναλλακτικές περιπτώσεις.

Υπάρχουν δύο είδη στατιστικών υποθέσεων. Η μηδενική (συμβολίζεται Η0) και η εναλλακτική (συμβολίζεται Η1).

- Η μηδενική υπόθεση (Η0 Null Hypothesis), εκφράζει την μη ύπαρξη διαφοράς ανάμεσα σε μια πληθυσμιακή παράμετρο και σε μία τιμή, ή ανάμεσα σε δύο παραμέτρους ή ότι υπάρχει ανεξαρτησία ή ότι δεν υπάρχει συσχέτιση. Η εναλλακτική υπόθεση εκφράζει την υφιστάμενη κατάσταση. Η εναλλακτική υπόθεση μπορεί να γίνει απορριφθεί, αλλά όχι να αποδειχθεί ως αληθής ή ψευδής.
- Η εναλλακτική υπόθεση (Η1, Alternative Hypothesis), εκφράζει την ύπαρξη διαφοράς ανάμεσα σε μια πληθυσμιακή παράμετρο και σε μία τιμή, ή ανάμεσα σε δύο παραμέτρους, ή ότι δεν υπάρχει ανεξαρτησία ή ότι υπάρχει συσχέτιση. Η εναλλακτική υπόθεση εκφράζει το ερευνητικό ερώτημα της μελέτης. Η εναλλακτική υπόθεση μπορεί να γίνει αποδεκτή, αλλά όχι να αποδειχθεί ως αληθής ή ψευδής.

Το αποτέλεσμα ενός ελέγχου υπόθεσης θα είναι είτε η απόρριψη της μηδενικής υπόθεσης και αποδοχή της εναλλακτικής, ή η μη απόρριψη της μηδενικής υπόθεσης και μη αποδοχή της εναλλακτικής. Ο έλεγχος δεν κρίνει την αλήθεια ή όχι των υποθέσεων.

4.8.2. Σφάλμα τύπου I και II

Λόγω της δειγματοληψίας υπάρχει πάντοτε το ενδεχόμενο λανθασμένης απόφασης, τα οποία αποτελούν σφάλματα και είναι σημαντικοί παράμετροι κατά τον έλεγχο υποθέσεων.

Πίνακας 4.23. Σφάλμα τύπου I και II

	H_0 Αληθής	H_0 Μη Αληθής
Απόρριψη H_0	Σφάλμα τύπου I	Σωστή απόφαση
Αποδοχή H_0	Σωστή απόφαση	Σφάλμα τύπου II

Το σφάλμα τύπου I, ονομάζεται επίπεδο σημαντικότητας, συμβολίζεται με το Ελληνικό γράμμα α και είναι η πιθανότητα λάθους να απορριφθεί η H_0, ενώ αυτή ισχύει. Προκαθορίζεται από τον ερευνητή. Συνηθέστερα επίπεδα σημαντικότητας: 1%, 5%, 10%. Το σφάλμα τύπου II, συμβολίζεται με το Ελληνικό γράμμα β. Η ποσότητα 1-β ονομάζεται ισχύς του ελέγχου και αποτελεί ένδειξη της ευαισθησίας του ελέγχου. Αυτό που επιδιώκεται στους ελέγχους στατιστικών υποθέσεων είναι η ελαχιστοποίηση τόσο του α όσο και του β. Αυτό δεν είναι εύκολο να συμβεί διότι η δειγματοληψία αποτελεί μερική εικόνα του πληθυσμού.

4.8.3. Έλεγχος υποθέσεων

Ο έλεγχος υποθέσεων ακολουθεί μια σειρά διαδοχικών βημάτων. Αρχικά διατυπώνονται οι δύο υποθέσεις (μηδενική και εναλλακτική) με ορθό και σαφή τρόπο. Η μηδενική υπόθεση θεωρείται ότι ισχύει, ως υπόθεση εργασίας. Στη συνέχεια ορίζεται το επιθυμητό επίπεδο σημαντικότητας. Ανάλογα με τον έλεγχο χρησιμοποιείται (με βάση τη θεωρητική στατιστική) κατάλληλη στατιστική συνάρτηση ελέγχου – test statistic, (της οποίας η θεωρητική κατανομή είναι γνωστή εκ των προτέρων, π.χ. κανονική, student, x^2, F κατανομή κ.λπ.). Η τιμή της συνάρτησης αυτής (test value) που προκύπτει από τα δεδομένα του δείγματος τις περισσότερες φορές δίνεται από τον τύπο:

$$\text{Test value}=\frac{\text{Sample Statistic - Hypothesized Parameter Value}}{\text{Standard Error of the Statistic}}$$ ή

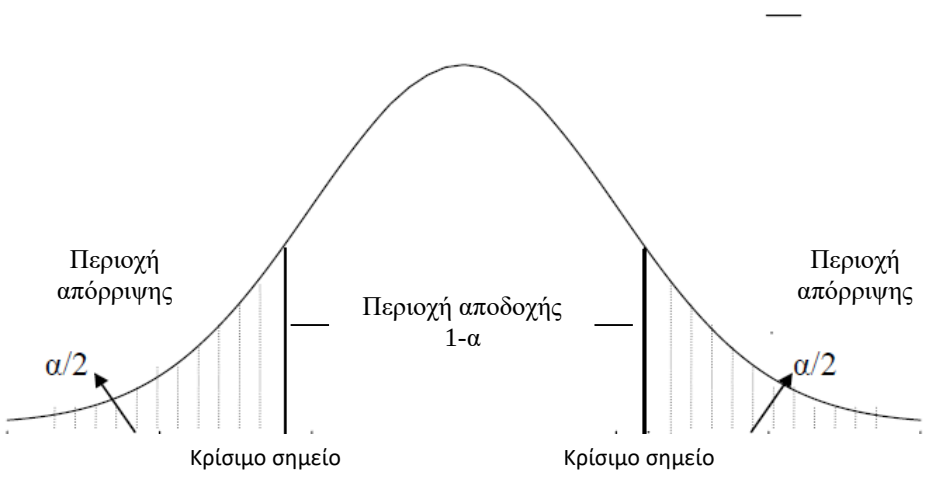
$$\text{Test value}=\frac{\text{Observed Values - Expected Values}}{\text{Standard Error of the Statistic}}$$

Εφόσον προσδιοριστεί το επίπεδο σημαντικότητας, από τους κατάλληλους πίνακες για την αντίστοιχη θεωρητική κατανομή της στατιστικής συνάρτησης ελέγχου, καθορίζονται οι κριτικές τιμές. Οι κριτικές τιμές καθορίζουν τις περιοχές απόρριψης και αποδοχής. Περιοχή απόρριψης ονομάζεται το εύρος τιμών για τη στατιστική συνάρτηση ελέγχου (τα test values που μπορεί προκύψουν ανάλογα το δείγμα) που δείχνει ότι υπάρχει σημαντική διαφορά ανάμεσα στην υποθετική τιμή της παραμέτρου και στην τιμή του δείγματος, η οποία δεν μπορεί να εξηγηθεί από

την τύχη, ή από σφάλματα δειγματοληψίας και έτσι η H_0 πρέπει να απορριφθεί. Περιοχή ονομάζεται εκείνο το εύρος τιμών για τη στατιστική συνάρτηση ελέγχου (τα test values που μπορεί προκύψουν ανάλογα το δείγμα) που δείχνει ότι δεν υπάρχει σημαντική διαφορά ανάμεσα στην υποθετική τιμή της παραμέτρου και στην τιμή του δείγματος, η οποία μπορεί να εξηγηθεί από την τύχη, ή από σφάλματα δειγματοληψίας και έτσι η H_0 δεν μπορεί να απορριφθεί.

Στη συνέχεια υπολογίζεται η τιμή της στατιστικής συνάρτησης ελέγχου από τις τιμές του δείγματος και η τιμή συγκρίνεται με τις τιμές απόρριψης οπότε λαμβάνεται η ανάλογη απόφαση για τη μηδενική υπόθεση.

Π.χ. Έστω ότι η συνάρτηση ελέγχου ακολουθεί την Κανονική Κατανομή. Για αμφίπλευρο έλεγχο:



	Κρίσιμες τιμές
Για α= 0,01	± 2,58
Για α=0,05	± 1,96
Για α =0,1	± 1,65

Διάγραμμα 4.1: Κανονική Κατανομή

Έλεγχος υποθέσεων	**Βήμα**	**Παρατηρήσεις**
Με βάση την περιοχή απόρριψης	1) Ορισμός των υποθέσεων H_0 και H_1	Η μηδενική υπόθεση περιέχει πάντοτε την ισότητα. Οι δύο υποθέσεις πρέπει να καλύπτουν τα δυνατά

		ενδεχόμενα πλήρως και να είναι αμοιβαία αποκλειόμενες. Η μηδενική υπόθεση θεωρείται (ως υπόθεση εργασίας) ότι ισχύει, οπότε ο έλεγχος θα καταλήξει είτε στην απόρριψη ή τη μη απόρριψή της.
	2) Καθορισμός της στατιστικής συνάρτησης ελέγχου (test statistic) και της κατανομής του	Ορίζεται από τον ερευνητή και προκύπτει από τα στοιχεία της έρευνας. Εξαρτάται από τα δεδομένα και κυρίως την ομοιογένεια του πληθυσμού.
	3) Καθορισμός του επιπέδου σημαντικότητας (α)	Συνήθως 1%, 5%, 10% ανάλογα με την επιθυμητή πιθανότητα απόρριψης της H_0 όταν αυτή είναι αληθής.
	4) Προσδιορισμός των τιμών απόρριψης (Decision Rule)	Υπολογίζεται ζώνη απόρριψης με βάση τη θεωρητική κατανομή της στατιστικής συνάρτησης ελέγχου και το συγκεκριμένο επίπεδο σημαντικότητας. Η ζώνη απόρριψης και ο έλεγχος απόρριψης μπορεί να είναι μονόπλευρος ή δίπλευρος.
	5) Συλλογή δεδομένων από το κατάλληλο δείγμα και υπολογισμός του στατιστικής συνάρτησης του δείγματος (test value)	Υπολογίζεται με βάση τη στατιστική συνάρτηση του δείγματος και τα δεδομένα του δείγματος.

	6) Απόφαση	Αν τιμή της στατιστικής συνάρτησης ελέγχου είναι εντός της ζώνης απόρριψης η μηδενική υπόθεση απορρίπτεται. Αν τιμή στατιστικής συνάρτησης ελέγχου είναι εκτός της ζώνης απόρριψης η μηδενική υπόθεση δεν απορρίπτεται.
Με βάση το p-value	Βήματα 1-5 ίδια με τον έλεγχο βάσει περιοχής απόρριψης	Πολλές φορές αντί για τις Κριτικές Τιμές χρησιμοποιείται το p-value. Ονομάζεται η πιθανότητα να προκύψει μια τιμή για την πληθυσμιακή παράμετρο υπό μελέτη, τόσο μεγάλη ή και μεγαλύτερη από αυτήν που υπολογίστηκε κάτω από την υπόθεση εργασίας ότι η H_0 ισχύει. Αποτελεί το παρατηρούμενο επίπεδο σημαντικότητας και συγκρίνεται με το επιθυμητό επίπεδο σημαντικότητας.
	6) Απόφαση	Αν p-value<α Τότε απορρίπτεται η μηδενική υπόθεση H_0 Αν p-value>α Τότε ΔΕΝ απορρίπτεται η μηδενική υπόθεση H_0

Οι βασικοί έλεγχοι για ένα και δύο δείγματα αφορούν τη μέση τιμή, τη διακύμανση και το ποσοστό είτε ως απόλυτες τιμές ή ως διαφορές.

4.8.4. Έλεγχος υποθέσεων για ένα δείγμα

Για ένα δείγμα οι βασικοί έλεγχοι αφορούν τη μέση τιμή του, τη διακύμανσή του και το ποσοστό εμφάνισης μιας τιμής. είτε ως απόλυτες τιμές ή ως διαφορές. Οι έλεγχοι μπορούν να εκτελεστούν είτε προς τη μια κατεύθυνση ή και τις δύο. Σημαντικό ρόλο στην επιλογή στατιστικής συνάρτησης παίζει η γνώση της μεταβλητότητας του πληθυσμού. Για μεγάλα δείγματα, ισχύει το κεντρικό οριακό θεώρημα για την δειγματική κατανομή του μέσου.

Πίνακας 4.24. Έλεγχος υποθέσεων για ένα δείγμα

Έλεγχος για ένα δείγμα	**Ορισμός**	**Παρατηρήσεις**
Μέση τιμή	Οι έλεγχοι υποθέσεων σχετικά με την μέση τιμή μ του πληθυσμού είναι της μορφής $\boldsymbol{H}_0 : \mu = \mu_0$, $\boldsymbol{H}_1 : \mu \neq \mu_0$; $\boldsymbol{H}_0 : \mu = \mu_0$, $\boldsymbol{H}_1 : \mu > \mu_0$; $\boldsymbol{H}_0 : \mu = \mu_0$, $\boldsymbol{H}_1 : \mu < \mu_0$. Έστω ότι η μεταβλητότητα του πληθυσμού είναι άγνωστη. Χρησιμοποιείται η στατιστική ελέγχου $t = \frac{\bar{x}-\mu}{s_{\bar{x}}}$ με $s_{\bar{x}} = \frac{s}{\sqrt{n}}$ (κατανομή t Student). Για μεγάλα δείγματα (>30), η υπόθεση $\boldsymbol{H}_0$ απορρίπτεται χάριν της $\boldsymbol{H}_1$ όταν αντίστοιχα ισχύουν $\lvert\bar{X} - \mu_0\rvert > z_{1-\frac{\alpha}{2}}\frac{S}{\sqrt{n}}$, $\bar{X} - \mu_0 > z_{1-\alpha}\frac{S}{\sqrt{n}}$, $\bar{X} - \mu_0 < -z_{1-\alpha}\frac{S}{\sqrt{n}}$, όπου α το επίπεδο σημαντικότητας.	Ο βαθμός εμπιστοσύνης $100(1-\alpha)\%$ ορίζεται από τον ερευνητή και εκφράζει την επιθυμητή πιθανότητα να υπάρχει σφάλμα τύπου Ι.

	Για πληθυσμούς που ακολουθούν κανονική κατανομή, η υπόθεση $\boldsymbol{H}_0$ απορρίπτεται χάριν της $\boldsymbol{H}_1$ όταν ισχύουν $\lvert\bar{X}-\mu_0\rvert > t_{\frac{\alpha}{2},n-1}\frac{S}{\sqrt{n}}$, $\bar{X}-\mu_0 > t_{\alpha,n-1}\frac{S}{\sqrt{n}}$, $\bar{X}-\mu_0 < -t_{\alpha,n-1}\frac{S}{\sqrt{n}}$..	
Ποσοστό	Οι έλεγχοι υποθέσεων σχετικά με το ποσοστό $\boldsymbol{p}$ κάποιας κατηγορίας στον πληθυσμό είναι της μορφής $\boldsymbol{H}_0 : \boldsymbol{p} = \boldsymbol{p}_0$ $\quad$ $\boldsymbol{H}_0 : \boldsymbol{p} = \boldsymbol{p}_0$ $\boldsymbol{H}_1 : \boldsymbol{p} \neq \boldsymbol{p}_0$, $\quad$ $\boldsymbol{H}_1 : \boldsymbol{p} > \boldsymbol{p}_0$, $\boldsymbol{H}_0 : \boldsymbol{p} = \boldsymbol{p}_0$ $\boldsymbol{H}_1 : \boldsymbol{p} < \boldsymbol{p}_0$. Για μεγάλα δείγματα, η υπόθεση $\boldsymbol{H}_0$ απορρίπτεται χάριν της $\boldsymbol{H}_1$ όταν αντίστοιχα ισχύουν $\lvert\hat{p}-p_0\rvert > z_{1-\frac{\alpha}{2}}\sqrt{\frac{p_0(1-p_0)}{n}}$, $\hat{p}-p_0 > z_{1-\alpha}\sqrt{\frac{p_0(1-p_0)}{n}}$, $\hat{p}-p_0 < -z_{1-\alpha}\sqrt{\frac{p_0(1-p_0)}{n}}$.	
Μεταβλητότητα	Οι έλεγχοι υποθέσεων σχετικά με την μεταβλητότητα σ^2 του πληθυσμού είναι της μορφής $\boldsymbol{H}_0 : \sigma^2 = \sigma_0^2$ $\quad$ $\boldsymbol{H}_0 : \sigma^2 = \sigma_0^2$ $\boldsymbol{H}_1 : \sigma^2 \neq \sigma_0^2$, $\quad$ $\boldsymbol{H}_1 : \sigma^2 > \sigma_0^2$, $\boldsymbol{H}_0 : \sigma^2 = \sigma_0^2$ $\boldsymbol{H}_1 : \sigma^2 < \sigma_0^2$. Για πληθυσμούς που ακολουθούν κανονική κατανομή, η υπόθεση $\boldsymbol{H}_0$ απορρίπτεται χάριν της $\boldsymbol{H}_1$ όταν αντίστοιχα ισχύουν	

	$\frac{(n-1)S^2}{\sigma_0^2} > \chi^2_{1-\frac{\alpha}{2},n-1}$ ή $\frac{(n-1)S^2}{\sigma_0^2} < \chi^2_{\frac{\alpha}{2},n-1}$, $\frac{(n-1)S^2}{\sigma_0^2} > \chi^2_{1-\alpha,n-1}$, $\frac{(n-1)S^2}{\sigma_0^2} < \chi^2_{\alpha,n-1}$.	

4.8.4.1 Παράδειγμα 1

Σε ένα κατάστημα ηλεκτρονικών πωλήσεων η μέση αξία του καλαθιού αγοράς είναι 68,5 €, με διακύμανση σ^2=7,29. Σε ένα τυχαίο δείγμα από 50 πελάτες η μέση αξία του καλαθιού αγοράς μετρήθηκε σε 69,7 €. Να ελεγχθεί η υπόθεση ότι η μέση αξία του καλαθιού αγοράς αυξήθηκε, σε επίπεδο σημαντικότητας α=2%.

Λύση

Ελέγχεται η αρχική υπόθεση H_0: μ=68,5, ως προς την εναλλακτική H_1: μ>68,5.

Η κρίσιμη περιοχή είναι:

$Z > Z_\alpha$ ή $Z > Z_{0,02} = 2,06$

Όπου $Z = (\overline{X} - \mu) / (\frac{\sigma}{\sqrt{n}}) = (69,7-68,5) / (2,7/7,07) = 3,14$

Επειδή $Z = 3,14 > 2,06$ η υπόθεση H_0: μ=68,5 απορρίπτεται και επομένως γίνεται δεκτή η H_1: μ>68,5.

Επομένως μέση αξία του καλαθιού αγοράς αυξήθηκε και αυτό δεν οφείλεται σε τυχαίους παράγοντες.

4.8.4.2 Παράδειγμα 2

Ορισμένοι διευθυντές εκτιμούν ότι οι αποδοχές των στελεχών έχουν μεταβληθεί σε σχέση με τις αποδοχές του προηγούμενου έτους, με μέση τιμή 69.000 €. Για τον έλεγχο του ισχυρισμού, λαμβάνεται ένα τυχαίο δείγμα 150 στελεχών και υπολογίζονται τα ακόλουθα.

$$\bar{X} = 72.000\ €, n = 150, s = 11.000€$$

Θα πραγματοποιηθεί έλεγχος υπόθεσης για να διαπιστωθεί εάν αυτά τα δεδομένα παρέχουν επαρκείς ενδείξεις αλλαγής των αποδοχών.

Το πρώτο βήμα σε έναν έλεγχο υποθέσεων είναι ο ορισμός της μηδενικής και της εναλλακτικής υπόθεσης.

- Η μηδενική υπόθεση είναι μια δήλωση σχετικά με τον πληθυσμό – που χρησιμοποιείται ως βάση για επιχείρημα – δεν έχει αποδειχθεί. Στην πραγματικότητα, ο ερευνητής αναζητά στοιχεία που την αμφισβητούν.
- Η εναλλακτική υπόθεση είναι μια εναλλακτική στη μηδενική. Είναι μια δήλωση του τι έχει αποφασιστεί να εξεταστεί με το πείραμα/παρατήρηση και γενικά εκφράζει το αναμενόμενο αποτέλεσμα της δοκιμής.

Σε αυτή την περίπτωση η μηδενική υπόθεση είναι:

H0: μ = 69.000 € (οι υφιστάμενες αποδοχές)

Η εναλλακτική υπόθεση είναι

H1: μ ≠ 69.000 € (οι αποδοχές δεν είναι ίσες με 69.000 €.)

Στη συνέχεια υπολογίζεται μια στατιστική ελέγχου. Η επιλογή στατιστικής ελέγχου εξαρτάται από το πρόβλημα (έλεγχος για μέση τιμή, ποσοστό ή μεταβλητότητα), και τη γνώση της μεταβλητότητας του πληθυσμού (κανονική κατανομή ή όχι). Σε περίπτωση όπου η πληροφορία για τον πληθυσμό είναι περιορισμένη επιλέγεται συντηρητική προσέγγιση (όπως για παράδειγμα η στατιστική t η οποία ακολουθεί κατανομή t Student).

Για αυτά τα δεδομένα ισχύει ότι (με τη βοήθεια λογισμικού)

$$t = \frac{\bar{x} - \mu}{s_{\bar{x}}} = 3{,}0131 \quad s_{\bar{x}} = \frac{s}{\sqrt{n}} = 0{,}874\,, df = 149$$

Τώρα πρέπει να ληφθεί μια στατιστική απόφαση. Το σκεπτικό είναι, το εξής. Εάν η μηδενική υπόθεση ισχύει, είναι πιθανό να προκύψει ένα δείγμα μέσου όρου που είναι περισσότερες από 3 τυπικές αποκλίσεις από τον μέσο όρο;

Ο κανόνας απόφασης της αποδοχής ή της απόρριψης της μηδενικής υπόθεσης καθορίζεται χρησιμοποιώντας μια κρίσιμη τιμή ή την τιμή p. Αυτή η τιμή είναι ένα μέτρο του πόσα στοιχεία έχουμε ενάντια στη μηδενική υπόθεση. Λέει το εξής: η πιθανότητα να παρουσιαστούν αυτά ή πιο ακραία αποτελέσματα με την προϋπόθεση ότι η μηδενική υπόθεση είναι αληθής.

Το επίπεδο για τη λήψη της απόφασης ονομάζεται επίπεδο σημαντικότητας και ορίζεται με το γράμμα α. (α είναι η πιθανότητα απόρριψης μιας αληθινής μηδενικής υπόθεσης). Αυτή η τιμή ορίζεται από τον ερευνητή πριν από την έρευνα. Συνήθως, οι ερευνητές απορρίπτουν μια υπόθεση εάν η τιμή p είναι μικρότερη από α = 0,05. Ο γενικός κανόνας είναι ότι μια μικρή τιμή p είναι απόδειξη έναντι της μηδενικής υπόθεσης ενώ μια μεγάλη τιμή p σημαίνει ελάχιστα ή καθόλου στοιχεία έναντι της μηδενικής υπόθεσης.

Άρα ο μέσος όρος του δείγματος είναι 3,01 τυπικές αποκλίσεις από τον υποτιθέμενο μέσο όρο του πληθυσμού.

Εφόσον: p = 0,00303 < 0,05 → απόρριψη H0.

Επομένως, το καταληκτικό επιχείρημα είναι ότι εάν οι μέσες αποδοχές του πληθυσμού είναι πραγματικά 69.000 €, είναι απίθανο να ληφθεί ένα δείγμα με μέση τιμή 72.000 €. Η πιθανότητα να απορριφθεί μια αληθινή μηδενική υπόθεση (σφάλμα τύπου I) είναι 0,003. Επομένως θεωρείται ότι υπάρχουν επαρκείς ενδείξεις ότι οι αποδοχές έχουν αλλάξει(έχουν αυξηθεί).

Εάν η τιμή p είναι μεγαλύτερη από το 0,05, πρέπει να μην απορριφθεί η μηδενική υπόθεση καθώς τα δεδομένα δεν επαρκούν για απόρριψη. Επομένως, δεν έχει αποδειχθεί ως αληθής η H0, απλά δεν υπάρχουν επαρκή στοιχεία για την απόρριψή της.

Σημείωση: Στην στατιστική, ένα αποτέλεσμα ονομάζεται στατιστικά σημαντικό εάν είναι απίθανο να προέκυψε τυχαία. Μια στατιστικά σημαντική διαφορά» σημαίνει απλώς ότι υπάρχουν στατιστικά στοιχεία που να λένε ότι υπάρχει διαφορά. Δεν σημαίνει ότι η διαφορά είναι απαραίτητα μεγάλη, σημαντική ή σημαντική με την κοινή σημασία της λέξης.

4.8.5. Έλεγχος υποθέσεων για δύο δείγματα

Για δύο δείγματα οι βασικοί έλεγχοι αφορούν διαφορές στη μέση τιμή τους, τη διακύμανσή του και το ποσοστό. Οι έλεγχοι μπορούν να εκτελεστούν είτε προς τη μια κατεύθυνση ή και τις δύο. Σημαντικό ρόλο στην επιλογή στατιστικής συνάρτησης παίζει η γνώση της μεταβλητότητας του πληθυσμού και το αν συσχετίζονται ή όχι. Για μεγάλα δείγματα, ισχύει το κεντρικό οριακό θεώρημα για την δειγματική κατανομή του μέσου.

Πίνακας 4.25. Έλεγχος υποθέσεων για δύο δείγματα

Έλεγχος για δύο δείγματα	**Ορισμός**	**Παρατηρήσεις**
Μέση τιμή	Η σύγκριση των μέσων τιμών μ_1, μ_2 δύο πληθυσμών γίνεται μέσω ελέγχων υποθέσεων της μορφής $\boldsymbol{H}_0 : \mu_1 = \mu_2$ $\quad$ $\boldsymbol{H}_0 : \mu_1 = \mu_2$ $\boldsymbol{H}_1 : \mu_1 \neq \mu_2$, $\quad$ $\boldsymbol{H}_1 : \mu_1 > \mu_2$. Για μεγάλα δείγματα, υποθέτοντας	Ο βαθμός εμπιστοσύνης $100(1-\alpha)\%$ ορίζεται από τον ερευνητή και εκφράζει την επιθυμητή πιθανότητα να υπάρχει σφάλμα τύπου I.

	πληθυσμούς με ίσες μεταβλητότητες, η υπόθεση $\boldsymbol{H}_0$ απορρίπτεται χάριν της $\boldsymbol{H}_1$ όταν αντίστοιχα ισχύουν $\lvert\bar{X}_1 - \bar{X}_2\rvert >$ $z_{1-\frac{\alpha}{2}} S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}, \quad \bar{X}_1 - \bar{X}_2 >$ $z_{1-\alpha} S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}.$ Για πληθυσμούς που ακολουθούν κανονική κατανομή, η υπόθεση $\boldsymbol{H}_0$ απορρίπτεται χάριν της $\boldsymbol{H}_1$ όταν ισχύουν $\lvert\bar{X}_1 - \bar{X}_2\rvert >$ $t_{\frac{\alpha}{2}, n_1+n_2-2} S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}, \quad \bar{X}_1 - \bar{X}_2 >$ $t_{\alpha, n_1+n_2-2} S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}.$	
Ποσοστό	Η σύγκριση των ποσοστών $\boldsymbol{p}_1, \boldsymbol{p}_2$ κάποιας κατηγορίας σε δύο πληθυσμούς γίνεται μέσω ελέγχων υποθέσεων της μορφής $\boldsymbol{H}_0 : \boldsymbol{p}_1 = \boldsymbol{p}_2 \quad \boldsymbol{H}_0 : \boldsymbol{p}_1 = \boldsymbol{p}_2$ $\boldsymbol{H}_1 : \boldsymbol{p}_1 \neq \boldsymbol{p}_2 \quad \boldsymbol{H}_1 : \boldsymbol{p}_1 > \boldsymbol{p}_2$. Ως γνωστόν οι έλεγχοι υποθέσεων βασίζονται σε κατανομές δειγματοληψίας όταν ισχύει η $\boldsymbol{H}_0$. Στην συγκεκριμένη περίπτωση η ισχύς της $\boldsymbol{H}_0$ σημαίνει $\boldsymbol{p}_1 = \boldsymbol{p}_2 = \boldsymbol{p}$, άρα ως εκτίμηση της άγνωστης παραμέτρου $\boldsymbol{p}$ χρησιμοποιούμε το ποσοστό $\hat{\boldsymbol{p}}$ της κατηγορίας συνολικά στα δύο δείγματα. Επομένως, για μεγάλα δείγματα, η υπόθεση $\boldsymbol{H}_0$ απορρίπτεται χάριν της $\boldsymbol{H}_1$ όταν αντίστοιχα ισχύουν	

	$\lvert\hat{p}_1 - \hat{p}_2\rvert > z_{1-\frac{\alpha}{2}}\sqrt{\hat{p}(1-\hat{p})\left(\frac{1}{n_1}+\frac{1}{n_2}\right)}$, $\hat{p}_1 - \hat{p}_2 > z_{1-\alpha}\sqrt{\hat{p}(1-\hat{p})\left(\frac{1}{n_1}+\frac{1}{n_2}\right)}$, όπου $\hat{\boldsymbol{p}} = \frac{\boldsymbol{n}_1\hat{\boldsymbol{p}}_1 + \boldsymbol{n}_2\hat{\boldsymbol{p}}_2}{\boldsymbol{n}_1 + \boldsymbol{n}_2}$.	
Μεταβλητότητα	Η σύγκριση των μεταβλητοτήτων σ_1^2, σ_2^2 δύο πληθυσμών γίνεται μέσω ελέγχων υποθέσεων της μορφής $\boldsymbol{H}_0 : \sigma_1^2 = \sigma_2^2$, $\boldsymbol{H}_1 : \sigma_1^2 \neq \sigma_2^2$ $\boldsymbol{H}_0 : \sigma_1^2 = \sigma_2^2$. $\boldsymbol{H}_1 : \sigma_1^2 > \sigma_2^2$ Για πληθυσμούς που ακολουθούν κανονική κατανομή, η υπόθεση $\boldsymbol{H}_0$ απορρίπτεται χάριν της $\boldsymbol{H}_1$ όταν αντίστοιχα ισχύουν $\frac{S_1^2}{S_2^2} > F_{1-\frac{\alpha}{2};n_1-1,n_2-1}$ ή $\frac{S_2^2}{S_1^2} < F_{1-\frac{\alpha}{2};n_2-1,n_1-1}$, $\frac{S_1^2}{S_2^2} > F_{1-\alpha;n_1-1,n_2-1}$.	

4.8.5.1 Παράδειγμα

Ένας καθηγητής επιθυμεί να προσδιορίσει εάν υπάρχει διαφορά στην απόδοση μεταξύ των φοιτητών που έχουν παρακολουθήσει ένα φροντιστηριακό μάθημα εκείνων που δεν έχουν παρακολουθήσει. Ένα δείγμα 100 φοιτητών που έχουν παρακολουθήσει αποκαλύπτει μέση απόδοση 74,3 / 100 με τυπική απόκλιση δείγματος 16 / 100. Ένα δείγμα 100 φοιτητών που δεν έχουν παρακολουθήσει το φροντιστηριακό μάθημα έχει μέση απόδοση 69,7 / 100 με τυπική απόκλιση 18 / 100. Υπάρχουν στοιχεία για διαφορά στην απόδοση των δύο ομάδων; Να δημιουργηθεί εκτίμηση του διαστήματος εμπιστοσύνης 95% της διαφοράς.

Μηδενική και Εναλλακτική Υπόθεση

H0: μ1 = μ2

H1: μ1 ≠ μ2

	Έχουν παρακολουθήσει	Δεν έχουν παρακολουθήσει
4.8.5.1.1 Μέση τιμή	74.3	69.7
Διακύμανση	16*16 = 256	18*18 = 324
Πλήθος	100	100

Διαφορά μέσων = 74.3 - 69.7 = 4.6

Τυπικό σφάλμα (Standard Error) = $\sqrt{\frac{256}{100} + \frac{324}{100}}$ = 2.408

Στατιστική z = 4.6/2.408 = 1.91

Κρίσιμη τιμή (από τον πίνακα τυποποιημένης κανονικής κατανομής) της z = 1.96

Συμπέρασμα:

Ισχύει ότι Στατιστική = 1.91 < Κρίσιμη τιμή = 1.96.

Επομένως δεν απορρίπτεται η H0. Τα δεδομένα δεν υποστηρίζουν διαφορά στην απόδοση μεταξύ των δύο ομάδων.

Εκτίμηση διαστήματος εμπιστοσύνης 95% της διαφοράς

Περιθώριο Σφάλματος (Margin of Error) = 1,96*2,408 = 4,72

Κάτω Όριο: 4,6 – 4,72 = -0,12

Ανώτερο όριο: 4,6 + 4,72 = 9,32

Το διάστημα [-0,12,9,32] περιέχει 0. Αυτό υποστηρίζει τη μηδενική υπόθεση ότι δεν υπάρχει διαφορά στους μέσους όρους.

4.9. Γραμμική παλινδρόμηση

Σε αρκετά προβλήματα της στατιστικής το ενδιαφέρον εστιάζεται στην ταυτόχρονη μελέτη δύο ή περισσότερων τυχαίων μεταβλητών, για να προσδιοριστεί αν και με ποιο τρόπο οι μεταβλητές αυτές σχετίζονται μεταξύ τους. Η ανάλυση παλινδρόμησης είναι ο κλάδος της στατιστικής που εξετάζει την σχέση μεταξύ δύο ή περισσότερων μεταβλητών με στόχο την πρόβλεψη της τιμή μιας μεταβλητής από τις τιμές μίας ή πολλών άλλων γνωστών μεταβλητών.

Η μεταβλητή Y, που θεωρείται ότι δέχεται την επιρροή της X ονομάζεται εξαρτημένη μεταβλητή (Response variable). Η μεταβλητή X η οποία επιλέγεται και καθοδηγείται από τον ερευνητή ονομάζεται ανεξάρτητη ή ερμηνευτική μεταβλητή (Predictor). Ο σκοπός της μεθόδου είναι να προσαρμοστούν τα δεδομένα ενός δείγματος σε ένα υποθετικό μοντέλο πρόβλεψης της σχέσης ανάμεσα στις μεταβλητές. Το μοντέλο θεωρείται υποθετικό, καθώς δεν είναι γνωστή η σχέση, αν υφίσταται, μεταξύ των μεταβλητών. Πρόκειται επομένως για ένα μοντέλο ερμηνείας και πρόβλεψης, με περιορισμούς. Η γραφική απεικόνιση των τιμών (X_i ,Y_i) καλείται διάγραμμα διασποράς (Scatter plot) και δίνει μια αρχική εικόνα της συσχέτισης των μεταβλητών.

Το μοντέλο μπορεί να είναι γραμμικό, ή μη γραμμικό. Τα γραμμικά μοντέλα μπορεί να περιλαμβάνουν μια ή περισσότερες ανεξάρτητων μεταβλητών. Απλή ονομάζεται η γραμμική παλινδρόμηση κατά την οποία χρησιμοποιούμε τις τιμές μίας μόνο μεταβλητής (ονομάζεται ερμηνευτική ή προβλεπτική μεταβλητή) για να προβλέψουμε τη μεταβλητή κριτήριο. Πολλαπλή ονομάζεται η γραμμική παλινδρόμηση κατά την οποία χρησιμοποιούμε τις τιμές πολλών προβλεπτικών μεταβλητών για να προβλέψουμε τη μεταβλητή κριτήριο.

Η σχέση μεταξύ των μεταβλητών X και Y δεν είναι σχέση συνάρτησης, αλλά στατιστική εκτίμηση, δηλαδή οι τιμές της Y δεν ορίζονται μονοσήμαντα από τις αντίστοιχες τιμές της X. Στην περίπτωση της γραμμικής παλινδρόμησης, το μοντέλο που εφαρμόζεται είναι μια ευθεία γραμμή (Επομένως, η σχέση περιγράφεται χρησιμοποιώντας την εξίσωση μιας ευθείας γραμμής).

Αν η σχέση μεταξύ X και Y είναι γραμμική, το υπόδειγμα παλινδρόμησης είναι της μορφής:

$Y = \beta_0 + \beta_1 X + \varepsilon_i$

- β_0 : ο σταθερός όρος (δηλ. η τιμή του Y όταν το X=0, το σημείο στο οποίο η γραμμή παλινδρόμησης τέμνει τον άξονα Y)
- β_1 : ο συντελεστής παλινδρόμησης για την προβλεπτική μεταβλητή ή η κλίση της ευθείας (δηλ. η γωνία που σχηματίζει η ευθεία με τον άξονα ψ) ή η κατεύθυνση/δύναμη της σχέσης (εκφράζει την κατά μέσον όρο μεταβολή της μεταβλητής Y όταν η X μεταβάλλεται κατά μία μονάδα)

- ε_i : Όρος σφάλματος, εκφράζει την απόκλιση των τιμών γύρω από την ευθεία παλινδρόμησης. Ο όρος σφάλματος περιλαμβάνεται, διότι το υπόδειγμα είναι μία προσέγγιση της πραγματικής σχέσης μεταξύ των μεταβλητών.

Η εκτίμηση των παραμέτρων β_0 και β_1 πραγματοποιείται με διάφορες μεθόδους, με την μέθοδο των ελαχίστων τετραγώνων να είναι αρκετά διαδεδομένη. Με βάση αυτή η ευθεία που προσαρμόζεται καλύτερα στα δεδομένα (ν σημεία στο επίπεδο) είναι αυτή που ελαχιστοποιεί το άθροισμα των τετραγώνων τετραγωνικών διαφορών μεταξύ των τιμών της μεταβλητής Y και των εκτιμήσεών τους,

$\min \sum(y_i - \hat{y}_i)^2$

Οπότε η εκτίμηση της ευθείας παλινδρόμησης (με b_0, b_1 τις σημειακές εκτιμήτριες των b_0, b_1.) δίνεται από τον τύπο

$$\hat{Y} = b_0 + b_1 X$$

με

$$b_1 = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}$$

$$b_0 = \bar{y} - b_1 \bar{x}$$

4.9.1. Συντελεστής προσδιορισμού

Μέτρο της προσαρμογής της ευθείας στα δεδομένα αποτελεί ο συντελεστής προσδιορισμού. Ορίζεται ως το ποσοστό της συνολικής μεταβλητότητας της εξαρτημένης μεταβλητής Y το οποίο ερμηνεύεται από τις μεταβολές της ανεξάρτητης μεταβλητής X στο υπόδειγμα παλινδρόμησης. Λαμβάνει τιμές στο διάστημα [0,1] και καλείται r-τετράγωνο και συμβολίζεται ως r^2. Τιμές κοντά στο μηδέν υποδηλώνουν ελάχιστη ή μη γραμμική συσχέτιση, ενώ τιμές κοντά στο ένα, υποδηλώνουν μεσαία ή ισχυρή συσχέτιση.

4.9.2. Παλινδρόμηση – Προϋποθέσεις

Για την εκτέλεση γραμμικής παλινδρόμησης (προσαρμογή δηλαδή ευθείας στο σύνολο δεδομένων) απαιτείται να πληρούνται ορισμένες προϋποθέσεις, διαφορετικά η αξιοπιστία του μοντέλου θα είναι χαμηλή και δεν θα είναι χρήσιμο για προβλέψεις.

Ειδικότερα, πρέπει να ελεγχθούν τα παρακάτω

- **Γραμμικότητα**. Η σχέση μεταξύ των τυχαίων μεταβλητών X και Y πρέπει να είναι γραμμική. Πριν την εκτέλεση την γραμμικής παλινδρόμησης πρέπει να γίνει έλεγχος της σχέσης μεταξύ της ανεξάρτητης και της εξαρτημένης μεταβλητής είτε με οπτικό τρόπο (διάγραμμα

διασποράς) ή μέσω του συντελεστή συσχέτισης ο οποίος πρέπει να είναι υψηλός και στατιστικά σημαντικός.

- **Σφάλματα**
    - Ανεξαρτησία των Σφαλμάτων. Οι τιμές των σφαλμάτων πρέπει να είναι στατιστικά ανεξάρτητες κάτι ιδιαίτερα σημαντικό όταν τα δεδομένα αποτελούν χρονοσειρά.
    - Κανονικότητα Σφαλμάτων ελέγξουμε την ύπαρξη ακραίων τιμών και αν τα κατάλοιπα ακολουθούν την κανονική κατανομή.
    - Ομοσκεδαστικότητα Σφαλμάτων. Βασική υπόθεση της γραμμικής παλινδρόμησης είναι ότι η διακύμανση των καταλοίπων παραμένει σταθερή, όποιες και εάν είναι οι τιμές των ερμηνευτικών μεταβλητών.

Επίσης πρέπει να ισχύουν τα εξής

- Ανεξαρτησία των παρατηρήσεων. Οι παρατηρήσεις πρέπει να είναι ανεξάρτητες, δηλαδή θα πρέπει να έχει εξασφαλιστεί πως μια παρατήρηση από το ένα δείγμα δεν πρόκειται να ανήκει και στο άλλο. Δεν βασίζεται σε κάποιο στατιστικό τεστ, αλλά στη λογική της έρευνας

Συνέχεια των μεταβλητών. Οι μεταβλητές μπορεί να είναι ποσοτικές, είτε διαστήματος (interval) είτε αναλογίας (ratio). Οι μεταβλητές πρέπει να είναι συνεχείς (Στην περίπτωση κατηγορικών μεταβλητών αυτές εισάγονται στο μοντέλο με μορφή ψευδομεταβλητών).

4.9.3. Έλεγχοι υποθέσεων και ερμηνεία

Εκτός του συντελεστή προσδιορισμού, κατά την ανάλυση παλινδρόμησης εξετάζεται και η στατιστική σημαντικότητα των παραμέτρων, ώστε να διασφαλιστεί η αξιοπιστία του μοντέλου.

- Κύριος στατιστικός έλεγχος.

  H0 : β1=0 έναντι της εναλλακτικής H1 : β1 ≠ 0, ισοδύναμος με τον έλεγχο για ύπαρξη συσχέτισης μεταξύ X και Y.

  Αν η H0 απορριφθεί, υπάρχουν επαρκείς ενδείξεις για ύπαρξη συσχέτισης μεταξύ X και Y.

  Αν η H0 δεν απορριφθεί, δεν υπάρχουν επαρκείς ενδείξεις για ύπαρξη συσχέτισης μεταξύ X και Y.
- Δευτερεύων Έλεγχος.

  H0 : β0=0 έναντι της εναλλακτικής H1 : β0 ≠ 0,

  Ερμηνεία: Η αναμενόμενη τιμή του Y όταν X=0. Πολλές φορές η τιμή αυτή δεν έχει ερμηνεία (διότι η τιμή X=0 δεν παρατηρείται ποτέ στην πράξη). Άλλες φορές θέτουμε β0=0 εκ-των-προτέρων και ανεξαρτήτως ελέγχου λόγω κοινής λογικής

4.10. Παραδείγματα

4.10.1. Παράδειγμα 1

Δίνονται παρακάτω οι μετρήσεις του χρόνου αναμονής πελατών στο κέντρο τηλεφωνικής εξυπηρέτησης μιας επιχείρησης για την επίλυση προβλημάτων.

8,79	6,33	8,53	8,84	7,74
6,98	6,71	10,5	7,39	8,38
13,45	12,93	13,17	13,18	20,35
13,62	18,08	13,64	11,87	14,54
16,23	21,21	11,75	18,23	14,51
12	19,69	19,95	20,83	11,52
18,26	17,35	19,52	12,96	14,99
21	12,05	20,45	18,89	15,06
18,48	18,86	13,3	18,09	21,2
16,1	18,4	17,1	15,02	15,11
14,51	16,05	18,33	11,7	16,19
14,56	14,39	18,42	13,57	18,91
19,58	22,69	17,99	19,15	19,31
17,96	21,83	23,18	22,73	21,3
18,74	25,07	22,6	23,96	21,89
24,13	19,35	16,72	16,45	22,15
16,77	19,12	19,28	23,65	25,37
47,16	27,29	56,99	36,1	53,51
69,79	49,14	41,6	61,85	36,75
31,09	56,71	52,71	32,46	46,75
70,33	76,46	62,89	53,43	36,1
21,63	47,36	64,48	35,97	53,94

1. Να υπολογιστούν οι ακόλουθες στατιστικές:

- α) Μέση τιμή
- β) Τυπική απόκλιση
- γ) Διάμεσος
- δ) Ελάχιστη τιμή δεδομένων
- ε) Μέγιστη τιμή δεδομένων
- ζ) Ενδοτεταρτημοριακό εύρος
- η) Συντελεστής κύρτωσης
- θ) Συντελεστής ασυμμετρίας

2. Να δημιουργηθεί το ιστόγραμμα των δεδομένων

3) Τι παρατήρηση μπορεί να εξαχθεί για τα δεδομένα και την κατανομή τους;

4.10.2. Παράδειγμα 2

Ένας μέσος χρήστης χρησιμοποιεί το κινητό τηλέφωνο κατά μέσο όρο 10 φορές την ημέρα, με τυπική απόκλιση ίση με 8. Η πρόσβαση στο κινητό τηλέφωνο ακολουθεί κανονική κατανομή.

- Ποια είναι η πιθανότητα ένας χρήστης να χρησιμοποιεί το κινητό του 35 φορές την ημέρα;
- Πόσες φορές την ημέρα χρησιμοποιεί το κινητό το 5% των πλέον ενεργών χρηστών;

4.10.3. Παράδειγμα 3

Ένα εργοστάσιο κατασκεύασε μια παρτίδα 2.000 ηλεκτρικών πλυντηρίων. Αν τα πλυντήρια έχουν μέση διάρκεια ζωής 1.000 ώρες και τυπική απόκλιση σ=200 ώρες, η εταιρία θέλει να γνωρίζει:

- Πόσα πλυντήρια θα έχουν πρόβλημα κάτω των 700 ωρών λειτουργίας?
- Πόσα πλυντήρια θα έχουν πρόβλημα κάτω των 900 ωρών και πάνω των 1.300 ωρών λειτουργίας?

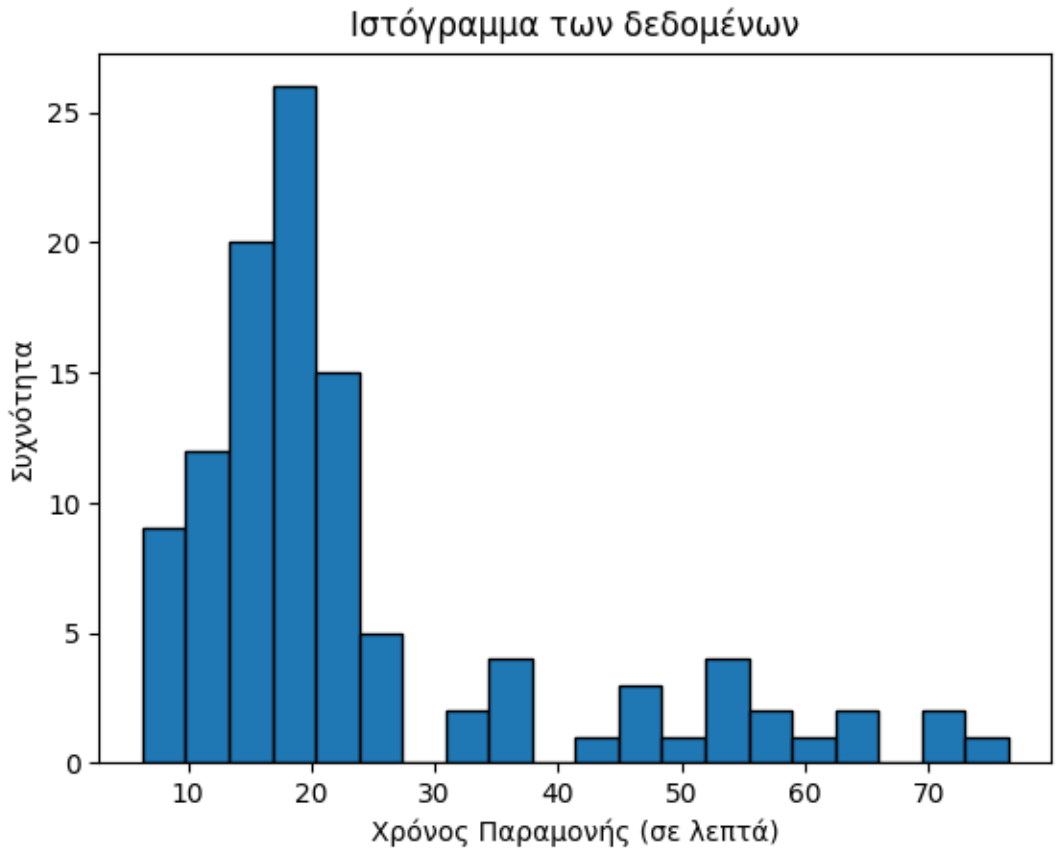
4.11. Ενδεικτικές λύσεις παραδειγμάτων με χρήση python

4.11.1. Παράδειγμα 1

```
import numpy as np
import matplotlib.pyplot as plt

# Δεδομένα
data = [
    [8.79, 6.33, 8.53, 8.84, 7.74],
    [6.98, 6.71, 10.5, 7.39, 8.38],
    [13.45, 12.93, 13.17, 13.18, 20.35],
    [13.62, 18.08, 13.64, 11.87, 14.54],
    [16.23, 21.21, 11.75, 18.23, 14.51],
    [12, 19.69, 19.95, 20.83, 11.52],
    [18.26, 17.35, 19.52, 12.96, 14.99],
    [21, 12.05, 20.45, 18.89, 15.06],
    [18.48, 18.86, 13.3, 18.09, 21.2],
    [16.1, 18.4, 17.1, 15.02, 15.11],
    [14.51, 16.05, 18.33, 11.7, 16.19],
    [14.56, 14.39, 18.42, 13.57, 18.91],
    [19.58, 22.69, 17.99, 19.15, 19.31],
    [17.96, 21.83, 23.18, 22.73, 21.3],
    [18.74, 25.07, 22.6, 23.96, 21.89],
    [24.13, 19.35, 16.72, 16.45, 22.15],
    [16.77, 19.12, 19.28, 23.65, 25.37],
    [47.16, 27.29, 56.99, 36.1, 53.51],
    [69.79, 49.14, 41.6, 61.85, 36.75],
```

```
    [31.09, 56.71, 52.71, 32.46, 46.75],
    [70.33, 76.46, 62.89, 53.43, 36.1],
    [21.63, 47.36, 64.48, 35.97, 53.94]
]

# Υπολογισμός στατιστικών
mean = np.mean(data)
std_dev = np.std(data)
median = np.median(data)
min_val = np.min(data)
max_val = np.max(data)
range_within_quartiles = np.percentile(data, 75) - np.percentile(data,
25)
skewness = np.mean((data - mean) ** 3) / (std_dev ** 3)
kurtosis = np.mean((data - mean) ** 4) / (std_dev ** 4)

# Εκτύπωση στατιστικών
print("Μέση τιμή:", mean)
print("Τυπική απόκλιση:", std_dev)
print("Διάμεσος:", median)
print("Ελάχιστη τιμή δεδομένων:", min_val)
print("Μέγιστη τιμή δεδομένων:", max_val)
print("Ενδοτεταρτημοριακό εύρος:", range_within_quartiles)
print("Συντελεστής κύρτωσης:", kurtosis)
print("Συντελεστής ασυμμετρίας:", skewness)

# Δημιουργία ιστογράμματος
plt.hist(np.array(data).flatten(), bins=20, edgecolor='black')
plt.title('Ιστόγραμμα των δεδομένων')
plt.xlabel('Χρόνος Παραμονής (σε λεπτά)')
plt.ylabel('Συχνότητα')
plt.show()

# Παρατηρήσεις για τα δεδομένα και την κατανομή τους
print("Οι περισσότεροι πελάτες εξυπηρετούνται σε διαστήματα περίπου από
17 έως 20 λεπτά. "Το ιστόγραμμα δείχνει μια δεξιά ασύμμετρη κατανομή με
λίγους πελάτες να αναμένουν για περισσότερο από 30 λεπτά.")
```

```
Μέση τιμή: 23.938363636363636
Τυπική απόκλιση: 15.652628563876346
Διάμεσος: 18.875
Ελάχιστη τιμή δεδομένων: 6.33
Μέγιστη τιμή δεδομένων: 76.46
Ενδοτεταρτημοριακό εύρος: 9.365
Συντελεστής κύρτωσης: 4.7818643206007705
Συντελεστής ασυμμετρίας: 1.6250905201323393
```



Οι περισσότεροι πελάτες εξυπηρετούνται σε διαστήματα περίπου από 17 έως 20 λεπτά. Το ιστόγραμμα δείχνει μια δεξιά ασύμμετρη κατανομή με λίγους πελάτες να αναμένουν για περισσότερο από 30 λεπτά.

Διάγραμμα 4.2: Ιστόγραμμα δεδομένων

4.11.2. Παράδειγμα 2

```
import scipy.stats as stats

# Δεδομένα
mean = 10  # Μέσος όρος χρήσης του κινητού τηλεφώνου
std_dev = 8  # Τυπική απόκλιση

# α) Υπολογισμός πιθανότητας χρήσης του κινητού 35 φορές την ημέρα
probability_a = stats.norm.cdf(35, mean, std_dev)
print("Πιθανότητα χρήσης του κινητού 35 φορές την ημέρα:",
probability_a)

# β) Υπολογισμός αριθμού χρήσεων για το 5% των πλέον ενεργών χρηστών
percentile_b = stats.norm.ppf(0.95, mean, std_dev)
print("Αριθμός χρήσεων για το 5% των πλέον ενεργών χρηστών:",
percentile_b)
```

```
Πιθανότητα χρήσης του κινητού 35 φορές την ημέρα: 0.9991109747008916
Αριθμός χρήσεων για το 5% των πλέον ενεργών χρηστών: 23.15882901561178
```

4.11.3. Παράδειγμα 3

```
import scipy.stats as stats

# Δεδομένα
μ = 1000  # Μέση διάρκεια ζωής
σ = 200   # Τυπική απόκλιση
N = 2000  # Συνολικός αριθμός πλυντηρίων

# Υπολογισμός πιθανότητας προβλήματος κάτω από 700 ώρες
prob_below_700 = stats.norm.cdf(700, loc=μ, scale=σ)

# Υπολογισμός πιθανότητας προβλήματος ανάμεσα σε 900 και 1300 ώρες
prob_between_900_1300 = stats.norm.cdf(1300, loc=μ, scale=σ) -
stats.norm.cdf(900, loc=μ, scale=σ)

# Υπολογισμός πλήθους πλυντηρίων για κάθε περίπτωση
num_below_700 = N * prob_below_700
num_between_900_1300 = N * prob_between_900_1300

# Εκτύπωση αποτελεσμάτων
print("Πλήθος πλυντηρίων με πρόβλημα κάτω από 700 ώρες:",
int(num_below_700))
print("Πλήθος πλυντηρίων με πρόβλημα μεταξύ 900 και 1300 ώρες:",
int(num_between_900_1300))
```

```
Πλήθος πλυντηρίων με πρόβλημα κάτω από 700 ώρες: 133
Πλήθος πλυντηρίων με πρόβλημα μεταξύ 900 και 1300 ώρες: 1249
```

4.12. *Ερωτήσεις αυτοαξιολόγησης*

4.1 Η ηλικία οκτώ αυτοκινήτων σε ένα χώρο στάθμευσης είναι 1, 15, 7, 4, 4, 9, 2, 6. Ο αριθμητικός μέσος της ηλικίας και η διάμεση ηλικία των αυτοκινήτων είναι αντίστοιχα:
α) 5 και 5
β) 7 και 6
γ) 6 και 4,5
δ) 6 και 5

4.2 Ποια είναι η πιθανότητα εμφάνισης τουλάχιστον μιας "ΚΕΦΑΛΗ (HEAD)" κατά τη ρίψη ενός νομίσματος τέσσερις διαδοχικές φορές:
α) 1/16
β) 4/16
γ) 14/16
δ) 15/16

4.3 Η μηδενική υπόθεση και η εναλλακτική υπόθεση αφορούν:
α) παραμέτρους του πληθυσμού
β) παραμέτρους του δείγματος
γ) σε μερικές περιπτώσεις παραμέτρους του δείγματος και σε άλλες περιπτώσεις παραμέτρους του πληθυσμού
δ) στατιστικές του δείγματος

4.4 Η βαθμολογία 200 φοιτητών στο μάθημα της Στατιστικής ακολουθεί κανονική κατανομή με αριθμητικό μέσο = 5 και τυπική απόκλιση = 1,5. Πόσοι φοιτητές έλαβαν τουλάχιστον πέντε και πέρασαν το μάθημα;
α) 50
β) 136
γ) 100
δ) δεν μπορούμε να υπολογίσουμε με βάση τα δεδομένα

4.5 Εάν η p-τιμή είναι ίση με 0.03, τί συμπέρασμα μπορείτε να βγάλετε για τον συντελεστή, εάν το επίπεδο σημαντικότητας είναι 5%;
α) Μία p-τιμή ίση με 0.03 με επίπεδο σημαντικότητας 5% σημαίνει ότι δεν μπορεί να απορριφθεί η μηδενική υπόθεση
β) Μία p-τιμή ίση με 0.03 με επίπεδο σημαντικότητας 5% σημαίνει ότι μπορεί να απορριφθεί η μηδενική υπόθεση ότι ο αντίστοιχος συντελεστής είναι ίσος με το μηδέν (0)
γ) Μπορεί να εξαχθεί συμπέρασμα για τον συντελεστή όταν η p-τιμή είναι 0.03, μόνο εάν το επίπεδο σημαντικότητας είναι μεγαλύτερο από 50%.
δ) Δεν δύναται να εξαχθεί συμπέρα για τον συντελεστή εάν η p-τιμή είναι 0.03

ΚΕΦΑΛΑΙΟ 5: Παλινδρόμηση

5.1. Εισαγωγή

Αναμενόμενα αποτελέσματα

- Αναγνώριση της ανάλυσης παλινδρόμησης
- Γνώση της απλής και πολλαπλής Παλινδρόμησης
- Ικανότητα εφαρμογής της μεθόδου ελαχίστων τετραγώνων (Least squares method)
- Ικανότητα προσδιορισμού της ευθείας που ταιριάζει περισσότερα σε σημεία που δίνονται
- Λογιστική παλινδρόμηση

Εξέταση της σχέσης μεταξύ μεταβλητών

Σε διάφορα προβλήματα της Στατιστικής, το ενδιαφέρον εστιάζεται στην:

- ταυτόχρονη μελέτη δύο ή περισσότερων μεταβλητών,
- με σκοπό να προσδιοριστεί με ποιο τρόπο οι μεταβλητές αυτές σχετίζονται μεταξύ τους

Παραδείγματα

- Η ηλικία και το βάρος ενός παιδιού έχουν κάποια θετική εξάρτηση/συσχέτιση μεταξύ τους (όσο μεγαλύτερη η ηλικία του παιδιού τόσο μεγαλύτερο βάρος θα έχει)
- Η διάρκεια ζωής των ζώντων οργανισμών σε μια περιοχή και το επίπεδο μόλυνσης της περιοχής έχουν αρνητική εξάρτηση μεταξύ τους (όσο πιο μεγάλη είναι η μόλυνση της περιοχής τόσο μικρότερη είναι η διάρκεια ζωής των οργανισμών που ζουν στην περιοχή)

5.1.1. Ανάλυση παλινδρόμησης (regression analysis)

Ο κλάδος της Στατιστικής που εξετάζει τη σχέση μεταξύ δύο ή περισσότερων μεταβλητών με κύριο στόχο την πρόβλεψη μιας απ' αυτές μέσω των άλλων λέγεται ανάλυση παλινδρόμησης (regression analysis). Ο όρος "reggresion" χρησιμοποιήθηκε για πρώτη φορά από τον Άγγλο ανθρωπολόγο Galton (1822-1911) το 1885. Ο Galton, κατά τη μελέτη του ύψους των παιδιών σε σχέση με το ύψος των γονέων, διαπίστωσε ότι παιδιά υψηλών γονέων τείνουν, κατά μέσο όρο, να είναι κοντύτερα

των γονιών τους, ενώ παιδιά κοντών γονέων τείνουν, κατά μέσο όρο, να γίνουν ψηλότερα των γονιών τους.

5.1.2. Απλή παλινδρόμηση

Στην απλή παλινδρόμηση, χρησιμοποιούμε μόνο μία μεταβλητή *X* και μία δεύτερη μεταβλητή *Y* η οποία μπορεί να προσεγγιστεί ικανοποιητικά από μία συνάρτηση του *X.*

- Για παράδειγμα η *Y* μπορεί να εκφραστεί μέσω της *X* ως $Y \approx 3X+5$
- *X*: ανεξάρτητη μεταβλητή (independent or input variable)
- *Y*: εξαρτημένη μεταβλητή (depended or response variable)

Ορισμός

Η παλινδρόμηση στην οποία υπάρχει μόνο μία ανεξάρτητη μεταβλητή καλείται απλή παλινδρόμηση ενώ αν υπάρχουν περισσότερες από μία ανεξάρτητες μεταβλητές ονομάζεται πολλαπλή παλινδρόμηση.

5.1.3. Απλή και πολλαπλή Ανάλυση παλινδρόμησης: παραδείγματα

Η εύρεση της σχέσης μεταξύ της συνολικής παραγωγής ενός αγρού και της ποσότητας λιπάσματος που χρησιμοποιήθηκε (απλή παλινδρόμηση)

Η εύρεση της σχέσης μεταξύ της συνολικής παραγωγής ενός αγρού και της ποσότητας λιπάσματος που χρησιμοποιήθηκε, της θερμοκρασίας της περιοχής και της υγρασίας της περιοχής (πολλαπλή παλινδρόμηση)

5.1.4. Ανεξάρτητες και εξαρτημένες μεταβλητές: πόσο ξεκάθαρο ρόλο έχει η καθεμιά;

Κλινικές μελέτες: ο ερευνητής καθορίζει από πριν τις δόσεις ενός φαρμάκου (ανεξάρτητη μεταβλητή) που δίνει στους ασθενείς και μετρά τις αντιδράσεις τους στο φάρμακο (εξαρτημένη μεταβλητή).

Με την παλινδρόμηση ενδιαφέρεται να προσδιορίσει μία σχέση δόσης-αντίδρασης για το συγκεκριμένο φάρμακο: για δεδομένη δόση να προβλέπει την αντίδραση.

5.1.5. Ανεξάρτητες και εξαρτημένες μεταβλητές: πόσο ξεκάθαρο ρόλο έχει η καθεμιά;

- Σε ένα δείγμα 30 μαθητών μετράμε, το βάρος και το ύψος τους

- Η διάκριση εδώ μεταξύ ανεξάρτητης και εξαρτημένης μεταβλητής είναι δύσκολη
- Αν αυτό που μας ενδιαφέρει είναι το "τι συμβαίνει με το βάρος των παιδιών όταν αλλάζει το ύψος τους", τότε θεωρούμε ως ανεξάρτητη μεταβλητή *X* το ύψος και ως εξαρτημένη μεταβλητή *Y* το βάρος
- Οπότε, ενδιαφερόμαστε για την παλινδρόμηση του βάρους (*Y*) πάνω στο ύψος (*X*)

Ανεξάρτητες και εξαρτημένες μεταβλητές: Ρόλος τους;

- Σε ένα δείγμα 30 μαθητών μετράμε, το βάρος και το ύψος τους
- Η διάκριση μεταξύ ανεξάρτητης και εξαρτημένης μεταβλητής είναι σχετικά δύσκολη
- Αντίθετα, αν αυτό που ενδιαφέρει είναι το "τι συμβαίνει με το ύψος των παιδιών όταν αλλάζει το βάρος τους", τότε ως ανεξάρτητη μεταβλητή *X* μπορεί να θεωρηθεί το βάρος, ενώ ως εξαρτημένη μεταβλητή *Y* το ύψος
- Οπότε έχουμε παλινδρόμηση του ύψους (*Y*) πάνω στο βάρος (*X*)

Ανεξάρτητες και εξαρτημένες μεταβλητές: πόσο ξεκάθαρο ρόλο έχει η καθεμιά;

5.1.6. Παράδειγμα

Μία τράπεζα προσπαθεί να μελετήσει εάν το εισόδημα ενός ατόμου αποτελεί ένδειξη για το εάν θα πληρώνει το άτομο κανονικά τις δόσεις ενός δανείου ή όχι.

Απάντηση:

- Αν θεωρηθεί ότι η εξαρτημένη μεταβλητή είναι εάν το άτομο θα πληρώνει κανονικά τις δόσεις δανείου ή όχι, το πρόβλημα αυτό δεν μπορεί να αναλυθεί με τη μέθοδο της παλινδρόμησης, διότι η εξαρτημένη μεταβλητή (αν το άτομο θα πληρώνει κανονικά τις δόσεις δανείου ή όχι) δεν είναι συνεχής μεταβλητή αλλά διακριτή που μάλιστα λαμβάνει δύο μη-αριθμητικές τιμές: ΝΑΙ/ΌΧΙ
- Οπότε, η εξαρτημένη μεταβλητή: εισόδημα (*Y*)
- Ανεξάρτητη μεταβλητή: εάν το άτομο θα πληρώνει κανονικά τις δόσεις δανείου ή όχι (*X*)
- Ενδιαφερόμαστε για την παλινδρόμηση του εισοδήματος (*Y*) πάνω στο εάν το άτομο θα πληρώνει κανονικά τις δόσεις δανείου ή όχι (*X*)

Άσκηση

Σε κάθε μία από τις παρακάτω περιπτώσεις, εντοπίστε την εξαρτημένη και την ανεξάρτητη μεταβλητή εάν πρόκειται να μελετηθούν οι σχέσεις των μεταβλητών αυτών με τη μέθοδο της ανάλυσης παλινδρόμησης:

1. Ο λογαριασμός ρεύματος κυμαίνεται ανάλογα με την κατανάλωση ενός νοικοκυριού
2. Μία μελέτη προσπαθεί να εξακριβώσει εάν ηλικιωμένοι οδηγοί αυτοκινήτων εμπλέκονται σε περισσότερα ατυχήματα απ' ότι άλλοι οδηγοί. Ο αριθμός των ατυχημάτων ανά 10000 οδηγοί συγκρίνεται με την ηλικία του οδηγού
3. Μία μελέτη προσπαθεί να εξετάσει εάν το εβδομαδιαίο ποσό που ξοδεύει ένα νοικοκυριό στο super market μεταβάλλεται με τον αριθμού των ατόμων του νοικοκυριού
4. Μία μελέτη προσπαθεί να εξακριβώσει εάν το επίπεδο εκπαίδευσης των ατόμων (μετρούμενο σε έτη που βρίσκεται σε οποιαδήποτε εκπαιδευτική διαδικασία) μειώνει το ποσοστό εγκληματικότητας σε έναν πληθυσμό
5. Ασφαλιστικές εταιρείες καθορίζουν το πόσο θα πληρώνεται κάθε μήνα σε ασφάλιστρα σε πολλά συμβόλαια βάσει της ηλικίας του ασφαλισμένου

5.1.7. Απλή παλινδρόμηση

Για την εύρεση του κατάλληλου μοντέλου για την περιγραφή της σχέσης μεταξύ δύο μεταβλητών που μας ενδιαφέρουν, πιο συχνά ξεκινάμε σχεδιάζοντας το διάγραμμα διασποράς (scatter plot) στο επίπεδο των παρατηρήσεων που είναι διαθέσιμο. Στο διάγραμμα αυτό, οι τιμές της μεταβλητής *X* τοποθετούνται στον οριζόντιο άξονα και της μεταβλητής *Y* στον κατακόρυφο άξονα.

5.1.8. Απλή γραμμική παλινδρόμηση

Η απλούστερη περίπτωση παλινδρόμησης είναι η απλή γραμμική παλινδρόμηση (simple linear regression), κατά την οποία χρησιμοποιούμε μόνο μια μεταβλητή *X*, και μια δεύτερη μεταβλητή *Y* η οποία μπορεί να προσεγγιστεί ικανοποιητικά από μία γραμμική συνάρτηση του *X*.

- *X*: ανεξάρτητη μεταβλητή (*independent or input variable*)
- *Y*: εξαρτημένη μεταβλητή (*dependent or response variable*)

5.1.9. Απλή γραμμική παλινδρόμηση: παράδειγμα 1

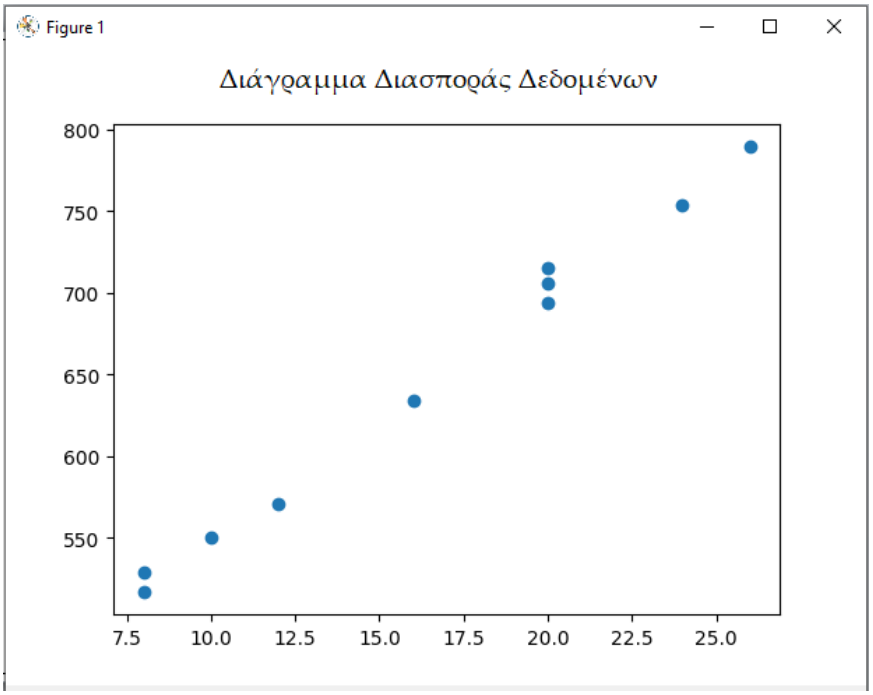
Ένας αγρότης ενδιαφέρεται να προσδιορίσει τον τρόπο με τον οποίο η ποσότητα *X* του λιπάσματος που χρησιμοποιείται σε ένα αγροτεμμάχιο επηρεάζει την παραγωγή *Y* του αγροκτήματος.

Πειραματίζεται λοιπόν με *v=10* όμοια αγροτεμάχια (ίδιου εμβαδού, σε περιοχές που επικρατούν παρόμοιες κλιματολογικές συνθήκες) έτσι ώστε οι όποιες διαφοροποιήσεις παρατηρούνται στην παραγωγή των αγρών να οφείλονται κατά κύριο λόγο στις διαφορετικές ποσότητες λιπάσματος που χρησιμοποιήθηκαν.

Στον παρακάτω πίνακα δίνεται η παραγωγή *Y* (σε χιλιάδες *kgr*) για *v=10* όμοια αγροτεμάχια όπως και η ποσότητα *X* του λιπάσματος που χρησιμοποιήθηκε στο καθένα (σε εκατοντάδες *kgr*).

Πίνακας 5.1. Δεδομένα

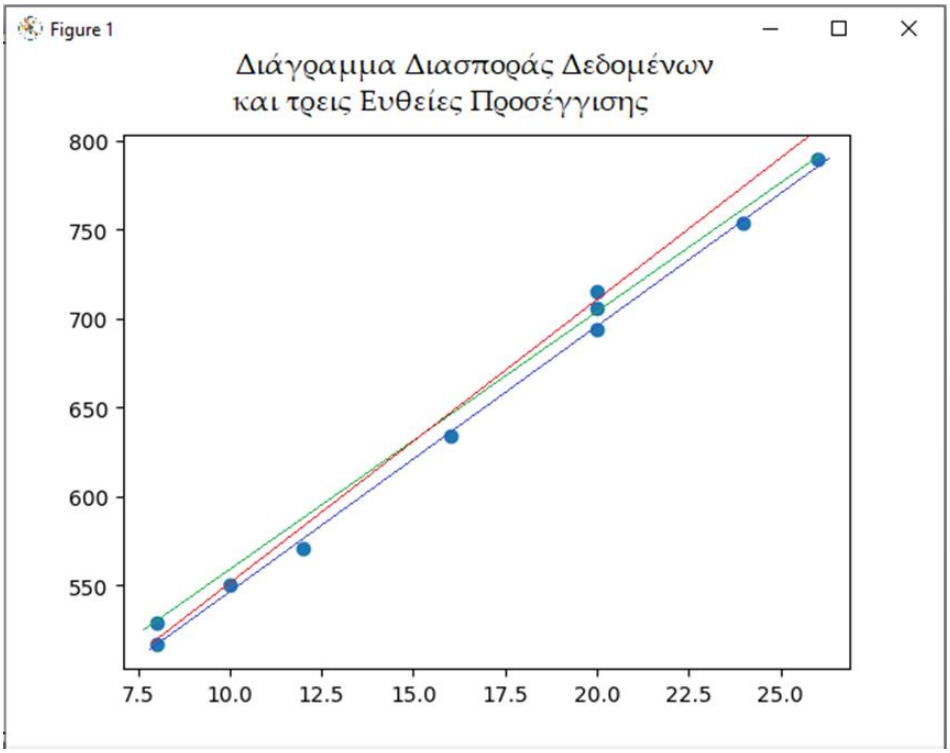
i	x_i	y_i
1	20	706
2	10	550
3	26	790
4	8	517
5	20	694
6	16	634
7	20	715
8	12	571
9	8	529
10	24	754



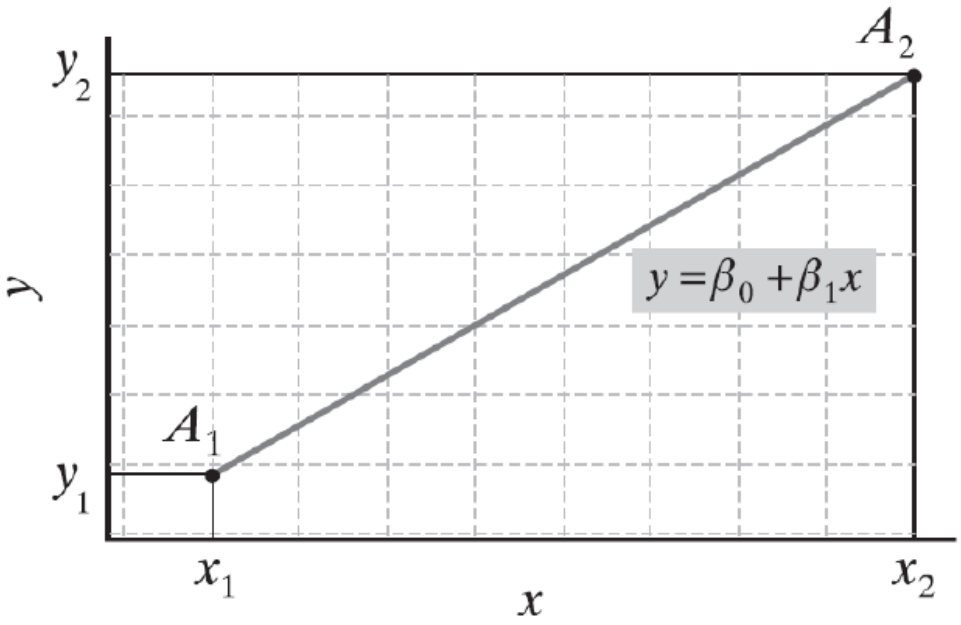
Διάγραμμα 5.1: Διασπορά Δεδομένων

Κώδικας στην *Python για Scatter Plot*

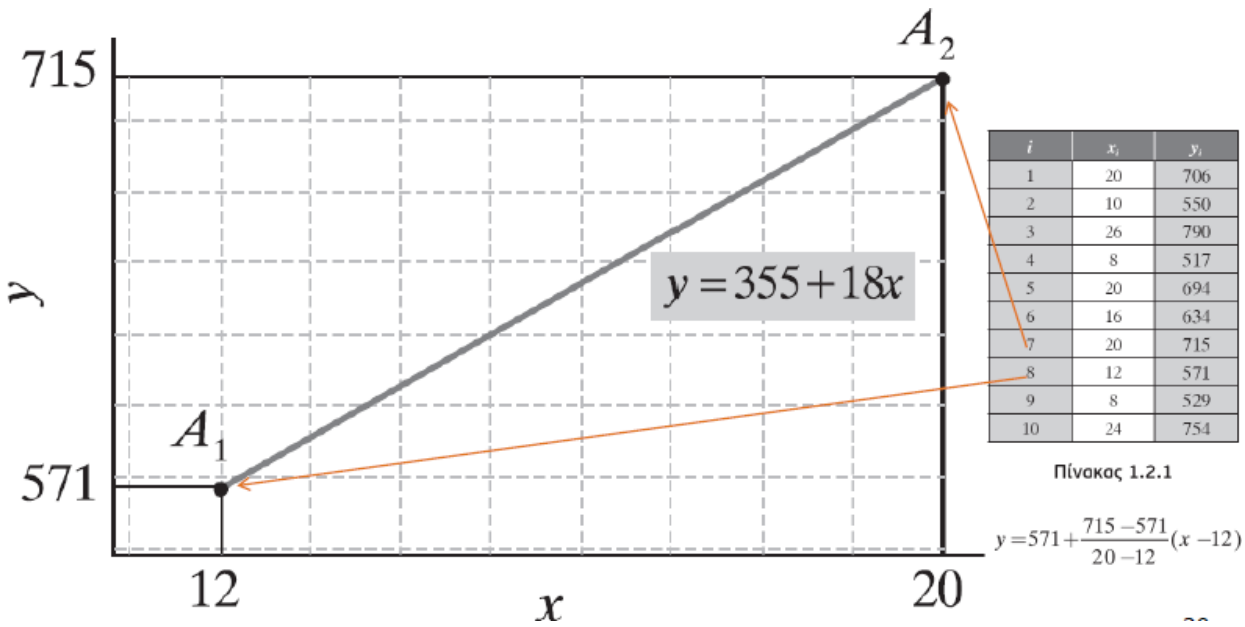
```
# Scatter Plot
from numpy import array
from matplotlib import pyplot
X = array([[1.0, 20],[1.0, 10],[1.0, 26],[1.0, 8], [1.0, 20],[1.0, 16],
        [1.0, 20],[1.0, 12],[1.0, 8],[1.0, 24]])
y=array([[706],[550],[790],[517],[694],[634],[715],[571],[529],[754]])
pyplot.scatter(X[:,1], y)
pyplot.show()
```



Διάγραμμα 5.2: Διασποράς Δεδομένων και τρεις Ευθείες Προσέγγισης



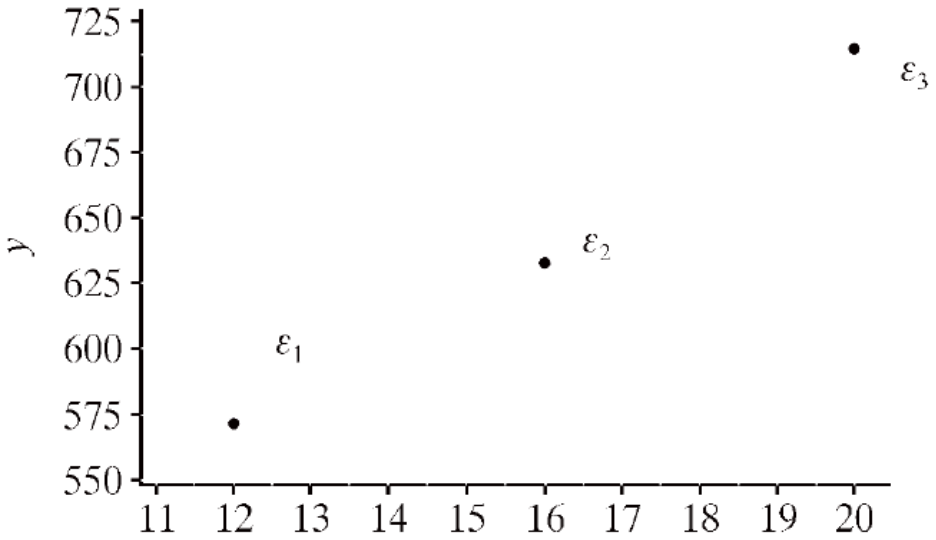
Διάγραμμα 5.3: Προσδιορίζοντας την ευθεία που ταιριάζει περισσότερο όταν έχουμε 2 σημεία



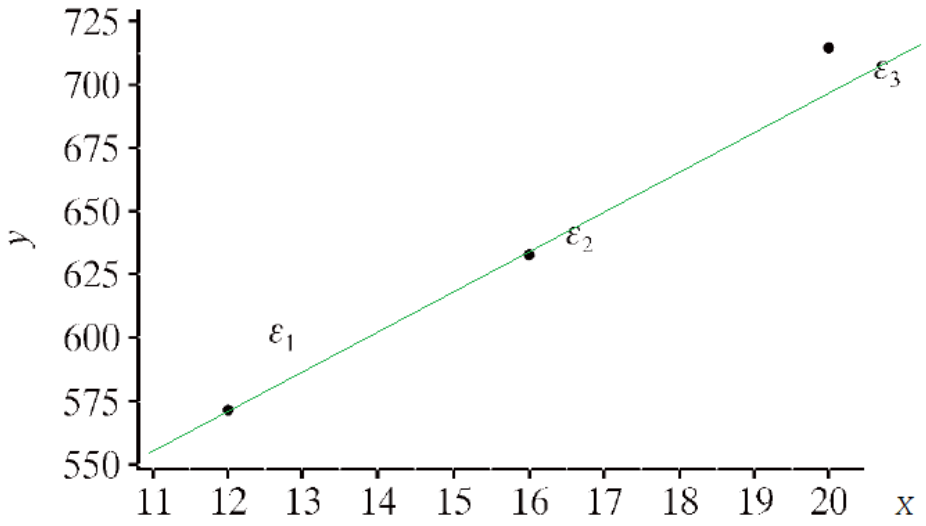
i	x_i	y_i
1	20	706
2	10	550
3	26	790
4	8	517
5	20	694
6	16	634
7	20	715
8	12	571
9	8	529
10	24	754

Πίνακας 1.2.1

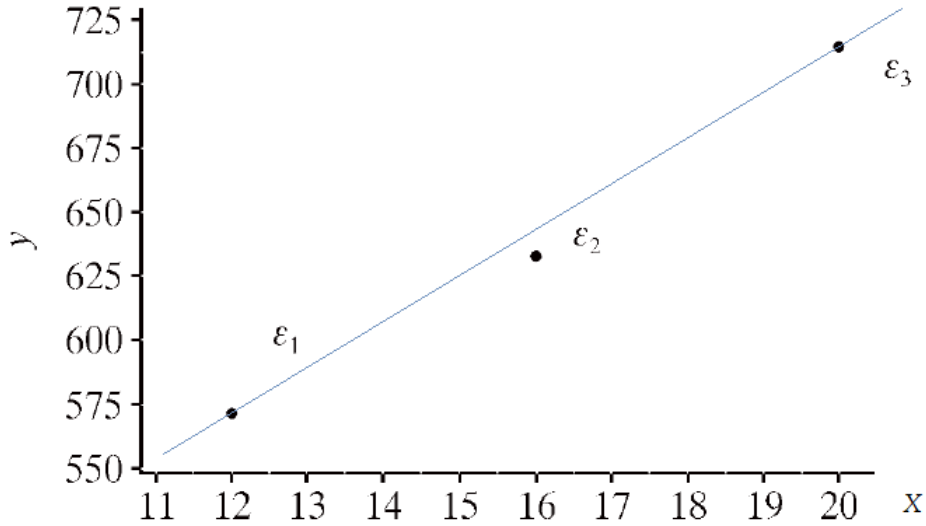
Διάγραμμα 5.4: Προσδιορίζοντας την ευθεία που ταιριάζει περισσότερο, όταν τα σημεία είναι δύο



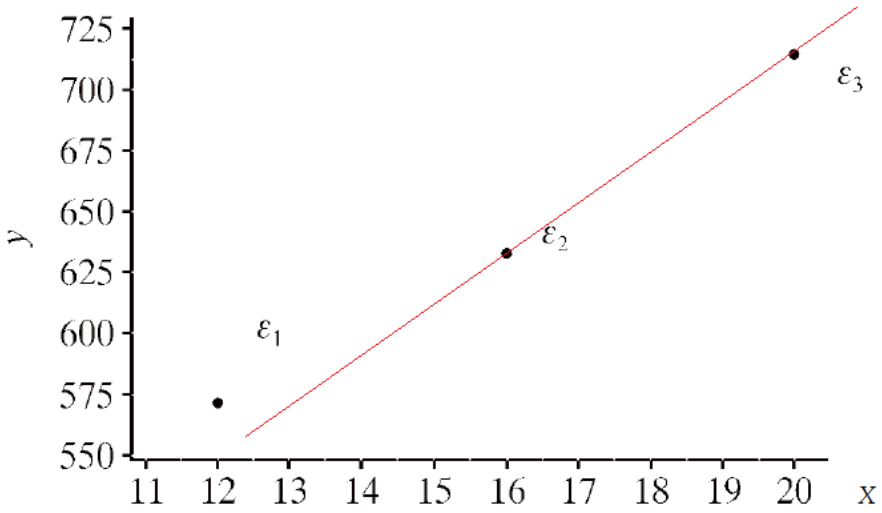
Διάγραμμα 5.5: Προσδιορίζοντας την ευθεία που ταιριάζει περισσότερο, όταν τα σημεία είναι τρία



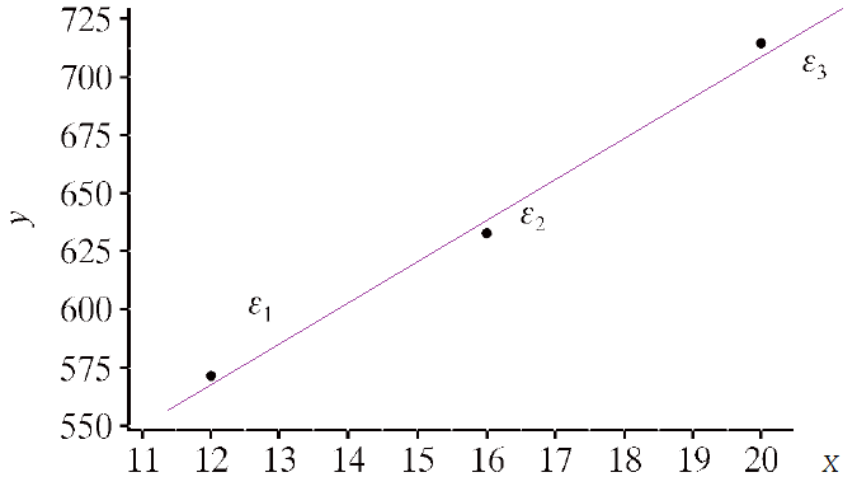
Διάγραμμα 5.6: Προσδιορίζοντας την ευθεία που ταιριάζει περισσότερο, όταν τα σημεία είναι τρία



Διάγραμμα 5.7: Προσδιορίζοντας την ευθεία που ταιριάζει περισσότερο, όταν τα σημεία είναι τρία

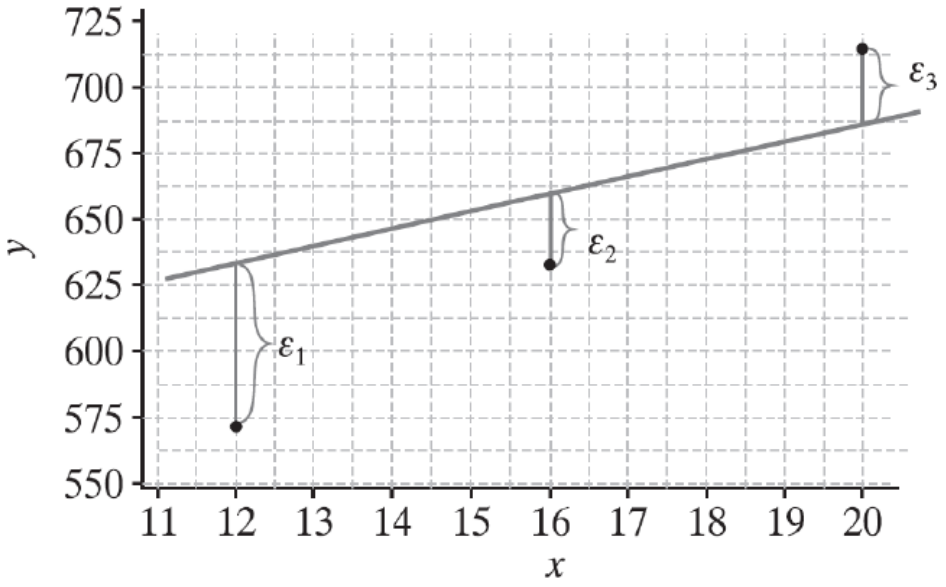


Διάγραμμα 5.8: Προσδιορίζοντας την ευθεία που ταιριάζει περισσότερο, όταν τα σημεία είναι τρία

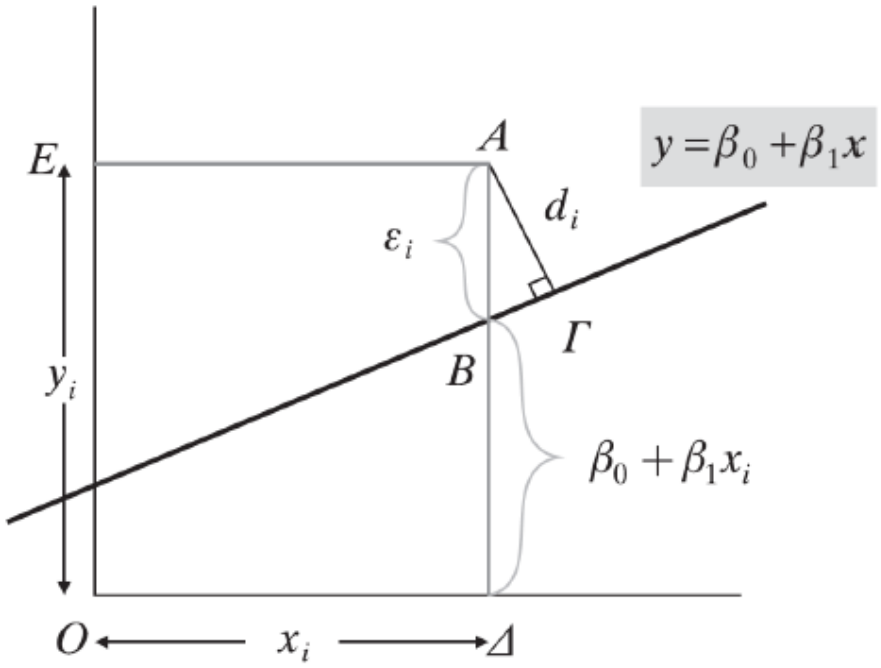


Διάγραμμα 5.9: Προσδιορίζοντας την ευθεία που ταιριάζει περισσότερο, όταν τα σημεία είναι τρία

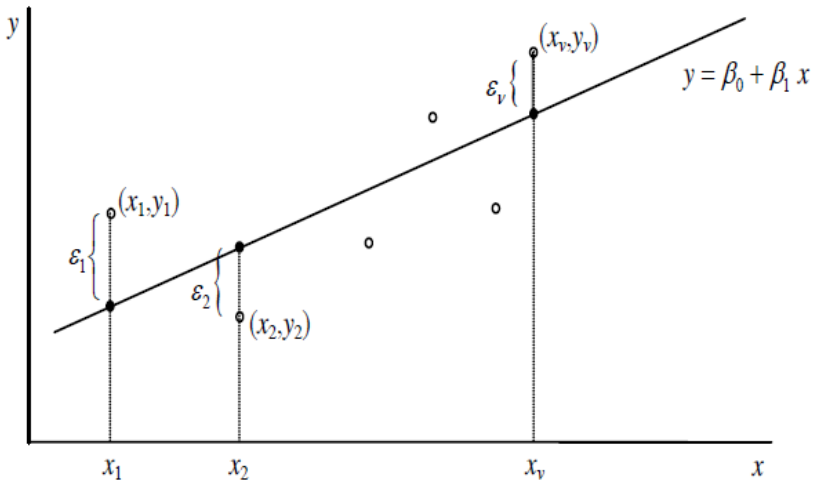
Υπάρχουν πολλές υποψήφιες γραμμές παλινδρόμησης που μπορούν να αποτυπωθούν για ένα σύνολο δεδομένων (Διαγράμματα 5.6-5.9). Η γραμμή που ελαχιστοποιεί τις αποστάσεις λέγεται ότι συλλαμβάνει και εξηγεί τη διακύμανση καλύτερα από οποιοδήποτε άλλο μοντέλο.



Διάγραμμα 5.10: Προσδιορίζοντας την ευθεία που ταιριάζει περισσότερο, όταν τα σημεία είναι τρία: κατακόρυφες αποκλίσεις



Διάγραμμα 5.11: Προσδιορίζοντας την ευθεία που ταιριάζει περισσότερο στα σημεία Κατακόρυφες αποστάσεις και συνήθεις αποστάσεις



$$\varepsilon_i = y_i - (\beta_0 + \beta_1 x_i) \Leftrightarrow y_i = \beta_0 + \beta_1 x_i + \varepsilon_i,\ i = 1,2,\ldots,\nu$$

Διάγραμμα 5.12: Προσδιορίζοντας την ευθεία που ταιριάζει περισσότερο στα σημεία

5.1.10. Προσδιορίζοντας την ευθεία που ταιριάζει περισσότερο στα σημεία

Μια μέθοδος που χρησιμοποιείται για την εύρεση της εξίσωσης της καλύτερης ευθείας που προσαρμόζεται σε (δισδιάστατα) δεδομένα, είναι η "*μέθοδος ελαχίστων τετραγώνων*" .

Η πρώτη αναφορά με ολοκληρωμένη ανάπτυξη της μεθόδου των ελαχίστων τετραγώνων εμφανίζεται το 1805 σε μια εργασία του Γάλλου μαθηματικού Legendre (1752-1833) και στη συνέχεια από το Γερμανό μαθηματικό Gauss, (1777-1855) στην εργασία "*Theoria Motus*" .

5.2. Μέθοδος ελαχίστων τετραγώνων (Least squares method)

Η ευθεία που προσαρμόζεται καλύτερα στα δεδομένα (*ν* σημεία στο επίπεδο) είναι, αυτή που ελαχιστοποιεί το άθροισμα των τετραγώνων των υπολοίπων e_i , δηλαδή το

$$\sum_{i=1}^{\nu} \varepsilon_i^2 = \sum_{i=1}^{\nu} \left(y_i - (\beta_0 + \beta_1 x_i)\right)^2$$

5.2.1. Mean Absolute Deviation (MAD method)

Θα μπορούσαμε να πάρουμε άθροισμα απόλυτων αποκλίσεων αντί για άθροισμα τετραγώνων;

$$\sum_{i=1}^{\nu} |\varepsilon_i| = \sum_{i=1}^{\nu} |y_i - (\beta_0 + \beta_1 x_i)|$$

Ναι, αλλά δεν θα ασχοληθούμε με αυτό στο παρών μάθημα

5.2.2. Least squares estimators (εκτιμήτριες ελαχίστων τετραγώνων)

Οι εκτιμήτριες ελάχιστων τετραγώνων για τις παραμέτρους β_0, β_1 της ευθείας $y = \beta_0 + \beta_1 x$, με βάση ν ζεύγη σημείων (x_i, y_i), $i = 1, 2, \ldots, \nu$ δίνονται από τους τύπους

$$\hat{\beta}_1 = \frac{\nu \sum_{i=1}^{\nu} x_i y_i - \left(\sum_{i=1}^{\nu} x_i\right)\left(\sum_{i=1}^{\nu} y_i\right)}{\nu \sum_{i=1}^{\nu} x_i^2 - \left(\sum_{i=1}^{\nu} x_i\right)^2}, \qquad \hat{\beta}_0 = \frac{1}{\nu}\sum_{i=1}^{\nu} y_i - \hat{\beta}_1 \cdot \frac{1}{\nu}\sum_{i=1}^{\nu} x_i. \qquad (1.2.5)$$

Η αντίστοιχη ευθεία $y = \hat{\beta}_0 + \hat{\beta}_1 x$ καλείται **ευθεία ελάχιστων τετραγώνων** ή **ευθεία παλινδρόμησης** της Y (πάνω) στη X.

Least squares estimators (εκτιμήτριες ελαχίστων τετραγώνων)

$$g(\beta_0, \beta_1) = \sum_{i=1}^{\nu} \varepsilon_i^2 = \sum_{i=1}^{\nu} (y_i - (\beta_0 + \beta_1 x_i))^2$$

$$\frac{\partial g(\beta_0, \beta_1)}{\partial \beta_0} = \sum_{i=1}^{\nu} 2(y_i - (\beta_0 + \beta_1 x_i)) \cdot (-1) = -2\left(\sum_{i=1}^{\nu} y_i - \nu \cdot \beta_0 - \beta_1 \sum_{i=1}^{\nu} x_i\right)$$

$$\frac{\partial g(\beta_0, \beta_1)}{\partial \beta_1} = \sum_{i=1}^{\nu} 2(y_i - (\beta_0 + \beta_1 x_i)) \cdot (-x_i) = -2\left(\sum_{i=1}^{\nu} x_i y_i - \beta_0 \sum_{i=1}^{\nu} x_i - \beta_1 \sum_{i=1}^{\nu} x_i^2\right)$$

$$\sum_{i=1}^{\nu} \varepsilon_i^2 = \sum_{i=1}^{\nu} (y_i - (\beta_0 + \beta_1 x_i))^2$$

Εάν θέσουμε τις μερικές παραγώγους ως προς β_0 και β_1 ίσες με 0, υπολογίζουμε το σύστημα

$$\sum_{i=1}^{\nu} y_i - \nu\beta_0 - \beta_1 \sum_{i=1}^{\nu} x_i = 0$$

$$\sum_{i=1}^{\nu} x_i y_i - \beta_0 \sum_{i=1}^{\nu} x_i - \beta_1 \sum_{i=1}^{\nu} x_i^2 = 0$$

Έχουμε τις κανονικές εξισώσεις:

$$\begin{aligned} \nu\beta_0 + \left(\sum_{i=1}^{\nu} x_i\right)\beta_1 &= \sum_{i=1}^{\nu} y_i \\ \left(\sum_{i=1}^{\nu} x_i\right)\beta_0 + \left(\sum_{i=1}^{\nu} x_i^2\right)\beta_1 &= \sum_{i=1}^{\nu} x_i y_i \end{aligned}$$

Λύνουμε ως προς β_0 και β_1:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{\nu}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{\nu}(x_i - \bar{x})^2}, \quad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

Αντικαθιστούμε:

$$\bar{y} = \frac{1}{\nu}\sum_{i=1}^{\nu} y_i, \quad \bar{x} = \frac{1}{\nu}\sum_{i=1}^{\nu} x_i$$

$$S_{xy} = \sum_{i=1}^{\nu}(x_i - \bar{x})(y_i - \bar{y}),$$

$$S_{xx} = \sum_{i=1}^{\nu}(x_i - \bar{x})^2$$

Οπότε βρίσκουμε ως λύσεις:

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}}, \qquad \hat{\beta}_0 = \bar{y} - \frac{S_{xy}}{S_{xx}}\bar{x}$$

5.2.3. Η ευθεία ελαχίστων τετραγώνων

Η ευθεία $y = \hat{\beta}_0 + \hat{\beta}_1 x$ καλείται ευθεία ελαχίστων τετραγώνων ή ευθεία παλινδρόμησης της *Y* (πάνω) στη *X*

Αντικαθιστώντας το $\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$ η ευθεία ελαχίστων τετραγώνων, παίρνει τη μορφή

$y - \bar{y} = \hat{\beta}_1 (x - \bar{x})$

η οποία φανερώνει ότι διέρχεται από το σημείο με συντεταγμένες $(\bar{x}, \bar{y})$ και έχει συντελεστή διεύθυνσης το $\hat{\beta}_1$

5.2.4. Παράδειγμα 2 (least squares calculation)

Ένας αγρότης ενδιαφέρεται να προσδιορίσει τον τρόπο με τον οποίο η ποσότητα *X* του λιπάσματος που χρησιμοποιείται σε ένα αγροτεμμάχιο επηρεάζει την παραγωγή *Y* του αγροκτήματος.

Πειραματίζεται λοιπόν με *ν=10* όμοια αγροτεμάχια (ίδιου εμβαδού, σε περιοχές που επικρατούν παρόμοιες κλιματολογικές συνθήκες) έτσι ώστε οι όποιες διαφοροποιήσεις παρατηρούνται στην παραγωγή των αγρών να οφείλονται κατά κύριο λόγο στις διαφορετικές ποσότητες λιπάσματος που χρησιμοποιήθηκαν.

Πίνακας 5.2. Δεδομένα παραδείγματος

i	x_i	y_i
1	20	706
2	10	550
3	26	790
4	8	517
5	20	694
6	16	634
7	20	715
8	12	571
9	8	529
10	24	754

Στον πάνω πίνακα δίνεται η παραγωγή *Y* (σε χιλιάδες *kgr*) για *ν=10* όμοια αγροτεμάχια όπως και η ποσότητα *X* του λιπάσματος που χρησιμοποιήθηκε στο καθένα (σε εκατοντάδες *kgr*)

$$\hat{\beta}_0 = \frac{1}{\nu}\sum_{i=1}^{\nu} y_i - \hat{\beta}_1 \cdot \frac{1}{\nu}\sum_{i=1}^{\nu} x_i$$

$$\hat{\beta}_1 = \frac{\nu\sum_{i=1}^{\nu} x_i y_i - \left(\sum_{i=1}^{\nu} x_i\right)\left(\sum_{i=1}^{\nu} y_i\right)}{\nu\sum_{i=1}^{\nu} x_i^2 - \left(\sum_{i=1}^{\nu} x_i\right)^2}$$

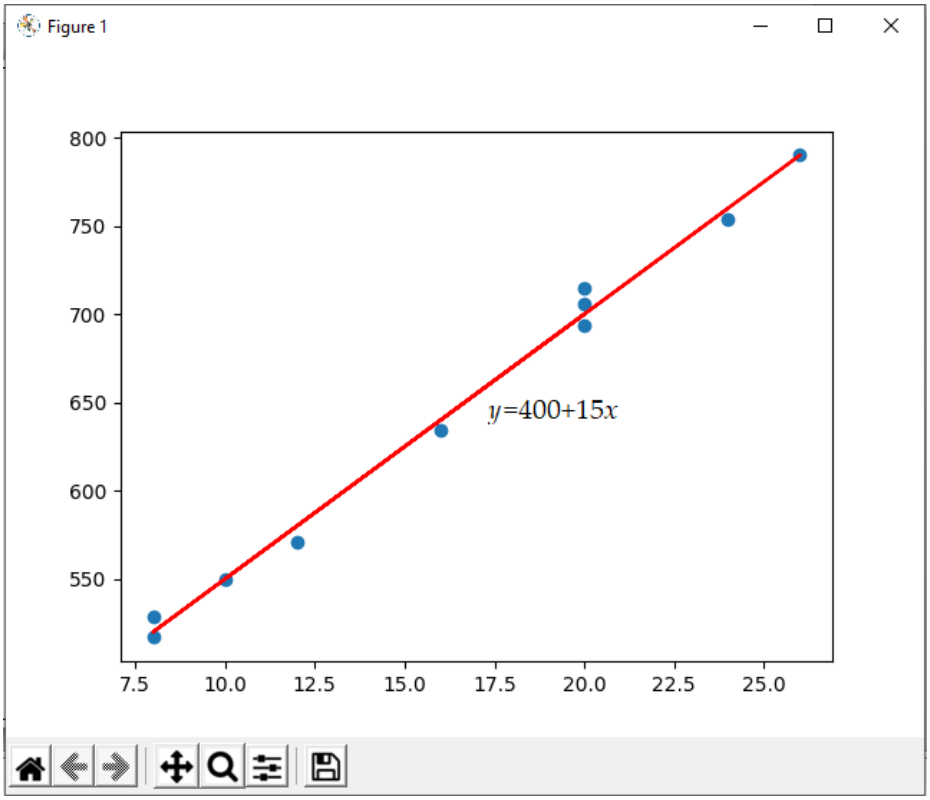
Πίνακας 5.3. Υπολογισμός του x_i^2 και $x_i * y_i$

i	x_i	y_i	x_i^2	$x_i * y_i$
1	20	706	400	14120
2	10	550	100	5500
3	26	790	676	20540
4	8	517	64	4136
5	20	694	400	13880
6	16	634	256	10144
7	20	715	400	14300
8	12	571	144	6852
9	8	529	64	4232
10	24	754	576	18096
Σύνολο	164	6460	3080	111800

5.2.5. Παράδειγμα 3 (least squares calculation)

Δίνονται: $\hat{\beta}_1 = 15$ και $\hat{\beta}_0 = 400$

και $y = \hat{\beta}_0 + \hat{\beta}_1 x = 400 + 15x$



Διάγραμμα 5. 13. Σχεδίαση της ευθείας από τα σημεία

$$y = \hat{\beta}_0 + \hat{\beta}_1 x = 400 + 15x$$

Σύμφωνα με την ευθεία παλινδρόμησης που προέκυψε, η εκτίμησή μας για την παραγωγή *Y* (σε χιλιάδες kgr) όταν δεν χρησιμοποιηθεί λίπασμα (*x*=0) είναι

$$y = 400 + 15 \cdot 0 = 400 = \hat{\beta}_0$$

Η ερμηνεία των εκτιμητριών ελαχίστων τετραγώνων

$$y = \hat{\beta}_0 + \hat{\beta}_1 x$$

ας θεωρήσουμε δύο διαδοχικές τιμές x_0 και $x_0' = x_0 + 1$ της ανεξάρτητης μεταβλητής *X*.

- Τότε η διαφορά των αντίστοιχων προβλεπόμενων τιμών της εξαρτημένης μεταβλητής θα είναι ίση με

$$y_0' - y_0 = (\hat{\beta}_0 + \hat{\beta}_1 x_0') - (\hat{\beta}_0 + \hat{\beta}_1 x_0) = \hat{\beta}_0 + \hat{\beta}_1 (x_0 + 1) - (\hat{\beta}_0 + \hat{\beta}_1 x_0) = \hat{\beta}_1$$

  - Συνεπώς $y_0' = y_0 + \hat{\beta}_1$ και μπορούμε να πούμε ότι η εκτιμήτρια $\hat{\beta}_1$, παριστάνει τη μεταβολή της εξαρτημένης μεταβλητής *Y*, όταν το *X*′ μεταβληθεί κατά μια μονάδα

Συγκεκριμένα, όταν το *X* αυξηθεί κατά μια μονάδα, τότε το *Y* αυξάνεται κατά $\hat{\beta}_1$ μονάδες, όταν $\hat{\beta}_1 > 0$, ή ελαττώνεται κατά $\hat{\beta}_1$ μονάδες, όταν $\hat{\beta}_1 < 0$

5.2.6. Στην εξίσωση ελαχίστων τετραγώνων

Η τιμή της εκτιμήτριας $\hat{\beta}_0$, της παραμέτρου β_0 παριστάνει την τεταγμένη του σημείου στο οποίο η ευθεία τέμνει τον άξονα *y΄y* (τιμή της εξαρτημένης μεταβλητής *Y* όταν *x=0*).

Όταν το $\hat{\beta}_0 = 0$ τότε η ευθεία διέρχεται από την αρχή των αξόνων ο συντελεστής διεύθυνσης $y = \hat{\beta}_0 + \hat{\beta}_1 x$ της ευθείας παριστάνει τη μεταβολή της εξαρτημένης μεταβλητής *Y* όταν το *X* μεταβληθεί κατά μια μονάδα. Έτσι, όταν το *x* αυξηθεί κατά μια μονάδα τότε το $\hat{y}$ αυξάνεται κατά $\hat{\beta}_1$ μονάδες όταν $\hat{\beta}_1 > 0$ ή ελαττώνεται κατά $\hat{\beta}_1$ μονάδες όταν $\hat{\beta}_1 < 0$

5.2.7. Προσδιορισμός των συντελεστών της εξίσωσης παλινδρόμησης

Προσδιορισμός της ευθείας ελαχίστων τετραγώνων (που αντιπροσωπεύει τα σημεία των παρατηρήσεων από τα ζεύγη των τιμών των *X* και *Y*).

Ανεξάρτητη μεταβλητή *X*

Εξαρτημένη μεταβλητή *Y*

Όταν οι ανεξάρτητες μεταβλητές είναι περισσότερες από δύο, έστω k με k>2, τότε το νέφος των n σημείων των παρατηρήσεων θα βρίσκεται σε ένα χώρο k+1 διαστάσεων ($X_1,X_2,\ldots,X_k$ *οι ανεξάρτητες μεταβλητές και Y η εξαρτημένη μεταβλητή).*

Στη γενική περίπτωση με το μέθοδο least squares προσδιορίζονται οι συντελεστές $A_1,A_2,\ldots,A_k$ *καθώς και ο σταθερός όρος* y_n *από τη σχέση:* $Y=A_1X_1+A_2X_2+\ldots+A_kX_k+y_n$

Άσκηση 1η

- Δίνονται τα σημεία (x_i, y_i), i = 1,2, ,.... ,v του επιπέδου
- Από όλες τις ευθείες της μορφής $y = \beta x$ που περνάνε από την αρχή των αξόνων, θα προσδιοριστεί εκείνη που τα προσεγγίζει καλύτερα με βάση τη μέθοδο least squares

α. Θα εκτιμηθεί η παράμετρος β της ευθείας $y = \beta x$

β. Θα βρεθεί η εκτιμήτρια $\hat{\beta}$ για τα επόμενα δεδομένα

Πίνακας 5.4. Δεδομένα εισόδου

i	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
x_i	30	20	60	80	40	50	70	90
y_i	75	52	120	170	86	110	153	194

5.3. Πολλαπλή παλινδρόμηση

Έλεγχος της πολλαπλής παλινδρόμησης με τη βοήθεια των υπολοίπων (σφάλματος)

- Με τη χρήση των υπολογιστών σήμερα, ο προσδιορισμός της παλινδρόμησης δεν είναι ιδιαίτερα δύσκολος
- Η έκφραση όμως του νέφους των παρατηρήσεων από την εξίσωση της παλινδρόμησης ή των ελαχίστων τετραγώνων, όπως αναφέραμε ήδη, ορίζει έναν υποχώρο του αρχικού των *k + 1* διαστάσεων
- Αυτή η παρουσίαση των δεδομένων είναι η «καλύτερη δυνατή»
- Πρέπει να βρούμε έναν τρόπο να τον μετρήσουμε ποσοτικά, αυτόν τον καθαρό ποιοτικό όρο «καλύτερη δυνατή»), και αυτό επιτυγχάνεται με τη βοήθεια των υπολοίπων
- Υπόλοιπα στην παλινδρόμηση ονομάζουμε τις διαφορές που υπάρχουν μεταξύ των *πραγματικών μετρούμενων παρατηρηθεισών τιμών της εξαρτημένης μεταβλητής Y* και *των τιμών που παίρνει η Y όταν ισχύει η εξίσωση της παλινδρόμησης*
- Δηλαδή όταν οι παρατηρήσεις είναι *n* θα έχουμε και *n* τιμές των υπολοίπων
- Συμβολίζουμε με SCR το άθροισμα των υπολοίπων

$$\sum_{i=1}^{n}(y_{0i})^2 \qquad SCR = y_{01}^2 + y_{02}^2 + \ldots + y_{0n}^2$$

Συντελεστής παλινδρόμησης *R*

- και με S^2 το λόγο SCR προς η — k — 1 $\qquad S^2 = \dfrac{SCR}{n-k-1}$
- Το S^2 μπορούμε να το θεωρήσουμε σαν μια διακύμανση

- Οι δύο ποσότητες SCR και S^2 δίνουν μια καλή ιδέα για την ολική ποιότητα της παλινδρόμησης καθώς και για κάθε έναν από τους συντελεστές της

- Η ολική ποιότητα της παλινδρόμησης προσδιορίζεται από το *συντελεστή παλινδρόμησης R*:

$$R = \sqrt{1 - \frac{SCR}{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

- Όταν έχουμε R > 0.80 τότε λέμε ότι η παλινδρόμηση εκφράζει αρκετά ικανοποιητικά το φαινόμενο που μελετούμε

5.3.1. Έλεγχος για τους συντελεστές A_1, A_2, ..., A_i

Ο *έλεγχος για τους συντελεστές* A_1, A_2, ..., A_i γίνεται αν πολλαπλασιάσουμε την απόκλισή τους από την υπολογισθείσα τιμή, με την τετραγωνική ρίζα του αντίστοιχου σημείου της κύριας διαγωνίου του αντίστροφου του πίνακα αδράνειας Δ = X' X

$S - \sqrt{\delta_{ii}} = \delta_{\iota}$ όπου δ_{ii} στοιχείο της διαγωνίου του πίνακα $\Delta^{-1} = (X' X)^{-1}$

Όταν ένας συντελεστής A_i είναι πολύ μεγαλύτερος από την αντίστοιχη απόκλιση του δ_i, τότε θεωρείται πολύ καλός συντελεστής, και η αντίστοιχη μεταβλητή είναι ενδιαφέρουσα στην παλινδρόμηση.

Δηλαδή, η μεταβλητή αυτή συντελεί πολύ στη δημιουργία του υποδείγματος της παλινδρόμησης.

Προσέχουμε όμως και το εξής:

- Πολλές φορές συμβαίνει να διαπιστώσουμε ότι μερικές ανεξάρτητες μεταβλητές X_i επιδρούν σοβαρά στην εξαρτημένη Y και συγχρόνως είναι έντονα συσχετισμένες μεταξύ τους

5.3.2. Επιλογή ανεξάρτητων μεταβλητών

Αν διατηρήσουμε όλες αυτές τις μεταβλητές στη διαδικασία της εξίσωσης παλινδρόμησης, είναι σαν να λαμβάνουμε υπόψη την κάθε μια δυο φορές.

Για το λόγο αυτόν είναι απαραίτητο να *επιλέξουμε ποιες ανεξάρτητες μεταβλητές* θα λάβουμε υπόψη για το τελικό υπόδειγμα της παλινδρόμησης.

Αυτό γίνεται αν οι μεταβλητές *επιλεχθούν μια μια*, **ξεκινώντας από τη σημαντικότερη** και σταματώντας σ' ένα σημείο σημαντικότητας, πέραν του οποίου η πρόσθεση μιας ακόμη ή και

περισσοτέρων μεταβλητών από τις υπόλοιπες να μη προσφέρει καμιά επιπλέον σημαντική πληροφορία.

5.3.3. Στάδια του ελέγχου της εξίσωσης παλινδρόμησης

Τα στάδια του ελέγχου της εξίσωσης παλινδόμησης είναι τα εξής:

1. Εφαρμόζουμε από μια απλή παλινδρόμηση για κάθε μια από τις ποσοτικές ανεξάρτητες μεταβλητές και θεωρούμε σημαντικότερη αυτή που δίνει τη μεγαλύτερη τιμή στο R ή την πιο μικρή τιμή στο SCR

2. Εφαρμόζουμε τη διπλή παλινδρόμηση θεωρώντας σαν ανεξάρτητες μεταβλητές, την πιο σημαντική και μια μια όλες τις υπόλοιπες. Θεωρούμε δεύτερη από πλευράς σημαντικότητας αυτή που δίνει τη μικρότερη τιμή στο SCR

3. Συνεχίζουμε τη διαδικασία αυτή μέχρι το τέλος. Σε κάθε ένα από τα στάδια το SCR θα ελαττώνεται, εάν συμβολίσουμε με Δ(SCR) τη μείωση που παρατηρούμε κάθε φορά, μπορούμε να υπολογίσουμε την ποσότητα
$$F = \frac{\Delta(SCR)}{S^2}$$

4. Στη συνέχεια συγκρίνουμε τις τιμές της F με τις τιμές των πινάκων της F κατανομής για (n— k— 1) και 1 βαθμούς ελευθερίας. Όταν το F που υπολογίσαμε βρεθεί μικρότερο από την τιμή της F κατανομής, δηλαδή όταν ο έλεγχος είναι αρνητικός, τότε σταματάμε τη διαδικασία βήμα-βήμα

5.3.4. Διακριτή ανάλυση (analyse discriminante) (ανάλυση ως προς δύο κριτήρια)

- Μία ειδική περίπτωση, πολύ ενδιαφέρουσα όμως, είναι αυτή κατά την οποία εξαρτημένη μεταβλητή μπορεί να πάρει μόνο δύο τιμές
- Δηλαδή η εξαρτημένη μεταβλητή Y θεωρείται ποιοτική με δύο μόνο κλάσεις
- Υποθέτουμε ότι θέλουμε να προσδιορίσουμε τους παράγοντες που συντελούν ώστε μια μικρομεσαία επιχείρηση να χρησιμοποιήσει την πληροφορική
- Θεωρούμε το σκοπό αυτόν ένα αντιπροσωπευτικό δείγμα από εκατό μικρομεσαίες επιχειρήσεις και τις χωρίζουμε σε δύο κατηγορίες
- Αυτές που χρησιμοποιούν ήδη την πληροφορική και αυτές που δεν τη χρησιμοποιούν

- Στην πρώτη περίπτωση θεωρούμε ότι η εξαρτημένη μεταβλητή παίρνει την τιμή 1 (Y = 1) και στη δεύτερη την τιμή 0 (Y = 0)

5.3.5. Διακριτή ανάλυση (analyse discriminante)

Με τον τρόπο αυτόν το διάνυσμα Y θα είναι ένα διάνυσμα με μηδενικά και μονάδες.

Οι ανεξάρτητες όμως μεταβλητές μπορεί να είναι οποιασδήποτε μορφής (αριθμός εργαζομένων, ετήσιος αριθμός πωλήσεων, ετήσιες δαπάνες, κέρδη, επενδύσεις).

Η περίπτωση αυτή είναι ακριβώς η ίδια μ' αυτή που αντιμετωπίσαμε προηγουμένως, αλλά επειδή το διάνυσμα Y είναι της μορφής Y =(1,0,0,1,1,…,0)' , ονομάζεται διακριτή ανάλυση.

5.3.6. Παραδείγματα πολλαπλής παλινδρόμησης Παράδειγμα 2

Έστω ότι ένας αντιπρόσωπος (ιδιοκτήτης) μιας εταιρίας αυτοκινήτων έχει 8 αντιπροσώπους σε οκτώ περιοχές της χώρας μας. Σε κάθε μια απ' αυτές τις γεωγραφικές περιοχές έχει κι ένα κόστος διαφήμισης. Ο αντιπρόσωπος (ιδιοκτήτης) θέλει να διαπιστώσει την απόδοση της διαφήμισης. Τα δεδομένα του αντιπροσώπου παρουσιάζονται στον παρακάτω πίνακα και αφορούν το εξάμηνο για το οποίο κάνει αυτόν τον έλεγχο.

Πίνακας 5.5. Πίνακας Αριθ. Πωλήσεων – Δαπάνη διαφήμισης

Περιοχή	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Αριθ. Πωλήσεων	75	76	82	82	76	83	76	74
Δαπάνη διαφήμισης	110	95	75	90	85	80	105	120

Απλή παλινδρόμηση

Πωλήσεις (*Y*) – Διαφήμιση (*X*)

```
# solve directly
import numpy as np
import pandas as pd
from numpy import array
from numpy.linalg import inv
from matplotlib import pyplot
```

```
X = array([[1.0, 110],
      [1.0, 95],
      [1.0, 75],
      [1.0, 90],
      [1.0, 85],
      [1.0, 80],
      [1.0, 105],
      [1.0, 120]])
y=array([[75],[76],[82],[82],[76],[83],[76],[74]])
# linear least squares
b = inv(X.T.dot(X)).dot(X.T).dot(y)
print('Οι εκτιμήτριες β0 και β1 είναι αντίστοιχα ',b)
# predict using coefficients
yhat = X.dot(b)
# 1st way - Calculate the correlation coefficient with numpy library
r = np.corrcoef(X[:,1], y.T)
print('R(Π,Δ)=',r[0,1])
# 2nd way - Calculate the correlation coefficient with pandas library
x1=pd.Series(X[:,1])
y1=pd.Series([75,76,82,82,76,83,76,74])
print('R(Π,Δ)=',y1.corr(x1))
```

Οι εκτιμήτριες β_0 και β_1 είναι αντίστοιχα:

[[95.88235294]

[-0.18823529]]

R(Π,Δ)= -0.8005

y=95.88+(-0.188)*x

Εφαρμόζοντας την απλή παλινδρόμηση στα δεδομένα αυτά

$$\Pi=-0.188\Delta+95.9$$

- όπου Δ = η δαπάνη διαφήμισης και Π ο αριθμός των πωλήσεων
- Ο συντελεστής παλινδρόμησης είναι R = -0.8005

Παρατηρούμε ότι ο συντελεστής παλινδρόμησης R είναι αρκετά σημαντικός αλλά αρνητικός, πράγμα που οδηγεί στο συμπέρασμα ότι η αύξηση της δαπάνης διαφήμισης επέφερε μείωση στις πωλήσεις. Επειδή αυτό το γεγονός φαινόταν ως μη φυσιολογικό, θέλησε να εξετάσει την επίδραση στις πωλήσεις που έχουν και τα έξοδα κίνησης που δίνει ο κάθε αντιπρόσωπος στους υπαλλήλους του. Προσθέτουμε λοιπόν και μια νέα γραμμή στον αρχικό πίνακα δεδομένων,

Πίνακας 5.6. Έξοδα κίνησης

Περιοχή	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Έξοδα κίνησης	26	28	34	31	29	36	25	23

Δαπάνη διαφήμισης – πωλήσεις

Να γραφεί πρόγραμμα σε Python το οποίο να υπολογίζει τις εκτιμήτριες β_0 και β_1

```
# solve directly
from numpy import array
from numpy.linalg import inv
from matplotlib import pyplot
X = array([[1.0, 110],
           [1.0, 95],
           [1.0, 75],
           [1.0, 90],
           [1.0, 85],
           [1.0, 80],
           [1.0, 105],
           [1.0, 120]])
y=array([[75],[76],[82],[82],[76],[83],[76],[74]])
# linear least squares
b = inv(X.T.dot(X)).dot(X.T).dot(y)
print('Οι εκτιμήτριες β0 και β1 είναι αντίστοιχα: ',b)
# predict using coefficients
yhat = X.dot(b)
```

Οι εκτιμήτριες β_0 και β_1 είναι αντίστοιχα:

[[95.88235294]

[-0.18823529]]

y=95.88+(-0.188)*x

5.3.7. Εύρεση των συσχετίσεων

Εφαρμόζοντας τώρα δύο απλές συσχετίσεις:

α) πωλήσεις - έξοδα κίνησης

β) δαπάνη διαφήμισης - έξοδα κίνησης

γ) μια διπλή συσχέτιση: πωλήσεις - (δαπάνη διαφήμισης + έξοδα κίνησης) έχουμε τα παρακάτω αποτελέσματα

(α) Π=0.75Κ + 56.04

(β) Δ=-3.214Κ+ 188.2

(γ) Π=1.020Κ + 0.082Δ+ 40.67

- Οι παραπάνω σχέσεις ερμηνεύουν το φαινόμενο και συγχρόνως μπορούν να υποδείξουν τι θα συμβάλλει περισσότερο στην αύξηση των πωλήσεων
- Ένας πολύ ενδιαφέρων παράγοντας, που επιδρά στον αριθμό των πωλήσεων, είναι τα έξοδα κίνησης, και ένας δευτερεύοντας η δαπάνη διαφήμισης.
- Οι δύο αυτοί παράγοντες λόγω της διαφοράς που παρουσιάζουν στο σύνολό τους είναι αρνητικά συσχετισμένοι μεταξύ τους R(Δ,Π)=-0,8005

<u>Δαπάνη διαφήμισης - έξοδα κίνησης</u>

Να γραφεί πρόγραμμα σε Python το οποίο να υπολογίζει την ευθεία παλινδρόμησης για την δαπάνη διαφήμισης σε σχέση με τα έξοδα κίνησης. Δίνονται τα δεδομένα εισόδου X = array([[1.0, 26], [1.0, 28],[1.0, 34],[1.0, 31],[1.0, 29],[1.0, 36],[1.0, 25],[1.0, 23]]) και τα δεδομένα εξόδου y= array([[110],[95],[75],[90],[85],[80],[105],[120]])

Απάντηση:

```
# solve directly
from numpy import array
from numpy.linalg import inv
```

```
from matplotlib import pyplot
X = array([[1.0, 26],
      [1.0, 28],
      [1.0, 34],
      [1.0, 31],
      [1.0, 29],
      [1.0, 36],
      [1.0, 25],
      [1.0, 23]])
y=array([[110],[95],[75],[90],[85],[80],[105],[120]])
# linear least squares
b = inv(X.T.dot(X)).dot(X.T).dot(y)
print('Οι εκτιμήτριες β0  και β1 είναι αντίστοιχα: ',b)
# predict using coefficients
yhat = X.dot(b)
```

```
[[188.21428571]
 [ -3.21428571]]
y=-3.214*x+188.21
```

Πολλαπλή παλινδρόμηση πωλήσεις - (δαπάνη διαφήμισης + έξοδα κίνησης)

Να γραφεί πρόγραμμα σε Python το οποίο να υπολογίζει την ευθεία παλινδρόμησης για την δαπάνη διαφήμισης σε σχέση με τα έξοδα κίνησης. Δίνονται τα δεδομένα εισόδου X = array([[1.0, 26, 110],[1.0, 28, 95],[1.0, 34, 75],[1.0, 31, 90],[1.0, 29, 85],[1.0, 36, 80],[1.0, 25, 105],[1.0, 23, 120]]) και τα δεδομένα εξόδου y= array([[75],[76],[82],[82],[76],[83],[76],[74]])

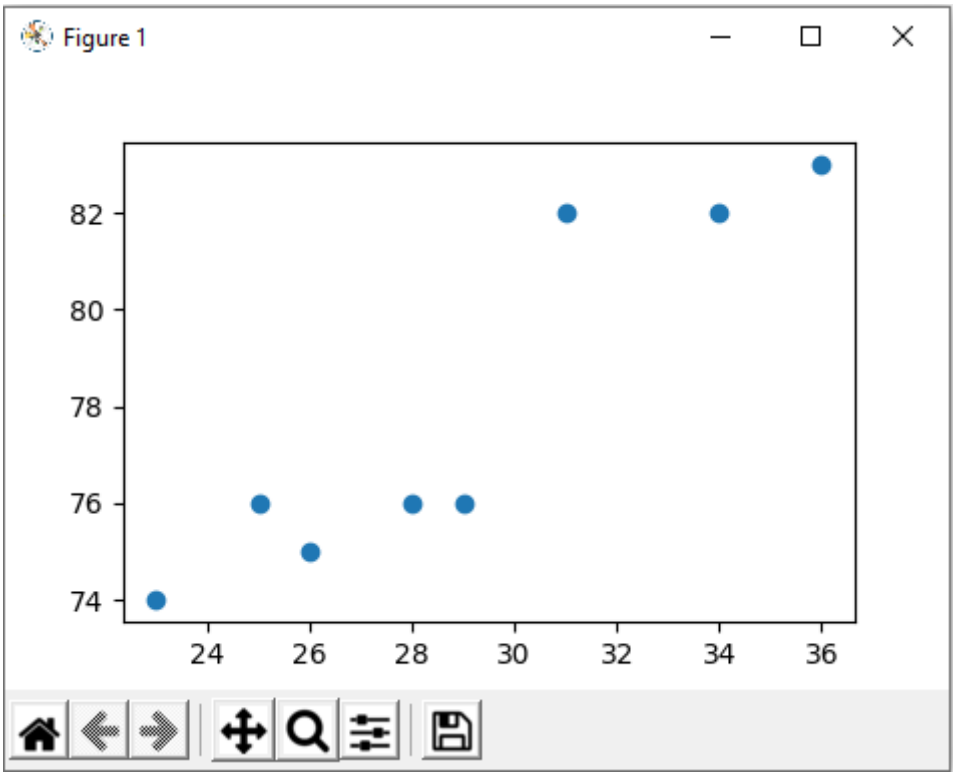
Απάντηση:

```
# solve directly
from numpy import array
from numpy.linalg import inv
from matplotlib import pyplot
X = array([  [1.0, 26, 110],
             [1.0, 28,  95],
             [1.0, 34,  75],
             [1.0, 31,  90],
```

```
                [1.0, 29,  85],
                [1.0, 36,  80],
                [1.0, 25, 105],
                [1.0, 23, 120]])
y=array([[75],[76],[82],[82],[76],[83],[76],[74]])
# linear least squares
b = inv(X.T.dot(X)).dot(X.T).dot(y)
print('Οι εκτιμήτριες β0, β1 και β2 είναι αντίστοιχα: ',b)
# predict using coefficients
yhat = X.dot(b)
# plot data and predictions
pyplot.scatter(X[:,1], y)
#pyplot.plot(X[:,1], yhat, color='red')
pyplot.show()
```

[[40.66760563]
 [1.01971831]
 [0.08169014]]

$y=1.020x_1 + 0.082x_2 + 40.67$



Διάγραμμα 5.14. Scatter plot

5.3.8. Συμπεράσματα

Αν λάβουμε υπόψη μόνο το δευτερεύοντα παράγοντα που είναι η διαφήμιση, οδηγούμαστε σε λανθασμένα συμπεράσματα, λόγω της έντονης επίδρασης του που είναι τα έξοδα κίνησης.

Η μοναδική λύση που μας οδηγεί σε μια ξεκάθαρη εικόνα των φαινομένων είναι η διπλή συσχέτιση, που μας παρέχει την επίδραση του κάθε παράγοντα ανεξάρτητα από τον άλλο.

Στην εξίσωση της διπλής παλινδρόμησης παρατηρούμε, ότι ο συντελεστής των εξόδων κίνησης (1,020) πολύ μεγαλύτερος από τον αντίστοιχο της διαφήμισης (0,082), ο οποίος όμως είναι θετικός (άσχετα αν είναι πολύ μικρός).

Το συμπέρασμα λοιπόν είναι, ότι η επιχείρηση για να αυξήσει τις πωλήσεις θα πρέπει αυξήσει τα έξοδα κίνησης των εργαζομένων σε κάθε υποκατάστημά της, χωρίς να μεταβάλλει τις δαπάνες διαφήμισης.

5.3.9. Παράδειγμα 3

Ας θεωρήσουμε *n*=12 παρατηρήσεις ενός μεγέθους Y (εξαρτημένη μεταβλητή) που εξαρτάται από k = 4 παραμέτρους X_1, X_2, X_3, X_4 (ανεξάρτητες μεταβλητές) όπως παρουσιάζονται στον παρακάτω πίνακα. Να δημιουργηθεί η ευθεία παλινδρόμησης.

Πίνακας 5.7. Δεδομένα εισόδου

Y	**X_1**	**X_2**	**X_3**	**X_4**
78.5	7	26	6	60
74.5	1	29	15	52
104.3	11	56	8	20
87.6	11	31	8	47
95.9	7	52	6	33
109.2	11	55	9	22
102.7	3	71	17	6
93.1	2	54	18	22
115.9	21	47	4	26
83.8	1	40	21	34
113.3	11	66	9	12
109.4	10	68	8	12

Επίλυση:

```
# solve directly
from numpy import array
from numpy.linalg import inv
from matplotlib import pyplot
X = array([  [1.0,  7,  26,   6,  60],
             [1.0,  1,  29,  15,  52],
             [1.0, 11, 56,   8,  20],
             [1.0, 11, 31,   8,  47],
             [1.0,  7, 52,   6,  33],
             [1.0, 11, 55,   9,  22],
             [1.0,  3, 71,  17,   6],
             [1.0,  2, 54,  18,  22],
             [1.0, 21, 47,   4,  26],
             [1.0,  1, 40,  21,  34],
             [1.0, 11, 66,   9,  12],
             [1.0, 10, 68,   8,  12]])
y=array([[78.5],[74.5],[104.3],[87.6],[95.9],[109.2],[102.7],[93.1],[115.9],[83.8],[113.3],[109.4]])
# linear least squares
b = inv(X.T.dot(X)).dot(X.T).dot(y)
print('Οι εκτιμήτριες β0, β1, β2 , β3 και β4 είναι αντίστοιχα: ',b)
# predict using coefficients
yhat = X.dot(b)
```

[[31.3412438]

[1.96241902]

[0.78258179]

[0.64012212]

[0.16040731]]

$y=1.96*x_1+0.78*x_2+0.64*x_3+0.16*x_4+31.34$

5.3.10. Δημιουργία πίνακα συσχέτισης

Τα αποτελέσματα θα τα παρουσιάσουμε όπως αυτά εκτυπώνονται εφαρμόζοντας ένα πρόγραμμα που παρέχει αυτές τις δυνατότητες ανάλυσης στον ηλεκτρονικό υπολογιστή. Κύριο ενδιαφέρον παρουσιάζει η μελέτη του πίνακα της συσχέτισης των 4 ανεξάρτητων μεταβλητών μεταξύ τους. Ο πίνακας συσχέτισης είναι ένας συμμετρικός τετραγωνικός πίνακας που δίνει τους συντελεστές συσχέτισης των μεταβλητών ανά δύο.

Κατασκευάζεται ως εξής:

- Πρώτα κατασκευάζουμε τον πίνακα των δεδομένων
- Στη συνέχεια την κάθε μεταβλητή την κανονικοποιούμε
- Δηλαδή την προσαρμόζουμε στην κανονική κατανομή ως εξής: Αφαιρούμε από κάθε τιμή X_{ij} τη μέση τιμή της αντίστοιχης μεταβλητής ακολούθως τη διαφορά τη διαιρούμε με την αντίστοιχη τυπική απόκλιση σ_j
- Αν ο νέος πίνακας συμβολιστεί με Τ, ο πίνακας συσχέτισης είναι το γινόμενο Τ'Τ αφού διαιρεθεί με τον αριθμό των παρατηρήσεων *n*

Πίνακας 5.8. Πίνακας Συσχέτισης

	Y	**X_1**	**X_2**	**X_3**	**X_4**
Y	1	0.731	0.816	-0.535	**-0.821**
X_1	0.731	1	0.229	-0.824	-0.245
X_2	0.816	0.229	1	-0.139	*-0.973*
X_3	-0.535	-0.824	-0.139	1	-0.029
X_4	**-0.821**	-0.245	*-0.973*	-0.029	1

Η απλή ευθεία παλινδρόμησης μεταξύ της Y και της X_4 είναι

- Σύμφωνα με τον πίνακα συσχέτισης, διαπιστώνουμε ότι:
- Η πρώτη και πιο σημαντική μεταβλητή είναι η X_4
- Η X_4 είναι η μεταβλητή η πιο έντονα συσχετισμένη με την Y διότι (COR(Y, X_4) = -0.821)

$Y=-0.676X_4 + 116.84$

με τυπική απόκλιση $\sigma_4 = 0.155$ και $F_4 = 22.8$

συντελεστή συσχέτισης R=-0.821 και ΔSCR= 0.674

- Η δεύτερη μεταβλητή που επιλέγεται είναι η X_1 με εξίσωση Παλινδρόμησης μεταξύ της Y και των X_4, X_1 την:

  $Y = - 0.614X_4 + 1.44X_1 + 103.1$

  με τυπικές αποκλίσεις $\sigma_4 = 0.049$ και $F_4 = 159.3$

  $\sigma_1 = 0.138$ και $F_1 = 108.3$

  συντελεστή συσχέτισης R = - 0.986 και ΔSCR=0.297

- Επειδή οι τιμές F_4 και F_1 είναι θετικές συνεχίζουμε τη διαδικασία βήμα-βήμα

Η τρίτη μεταβλητή που επιλέγεται είναι η X_2 με εξίσωση παλινδρόμησης μεταξύ της Y και των X_4, X_1, X_2

$Y = - 0.237X_4 + 1.452X_1 + 0.416X_2 + 71.65$

με τυπικές αποκλίσεις σ_4= 0,173 και $F_4 = 1.86$

σ_1= 0.117 και $F_1 = 154.01$

σ_2=0.186 και $F_2 = 5.02$

συντελεστή συσχέτισης R = 0.991 και ΔSCR= πολύ μικρό

- Αν ανατρέξουμε στον πίνακα συσχέτισης θα διαπιστώσουμε ότι οι μεταβλητές X_2 και X_4 είναι εξαιρετικά συσχετισμένες μεταξύ τους (COR(X_4, X_2)=-0973)

- Η εισαγωγή λοιπόν της μεταβλητής X_2 στην εξίσωση παλινδρόμησης, μειώνει αισθητά την επίδραση της X_4

- Αυτό το διαπιστώνουμε από το ότι στην τελευταία εξίσωση παλινδρόμησης η τυπική απόκλιση της X_4(0.173) είναι πολύ κοντά στο συντελεστή A_4(-0.237)

- Κύρια όμως το διαπιστώνουμε από την τιμή της F που είναι πολύ πιο κάτω από την τιμή σημαντικότητας που μας παρέχει ο πίνακας της F κατανομής

Όπως έχουμε αναφέρει, όταν δύο μεταβλητές είναι έντονα συσχετισμένες, θα πρέπει να αποσύρουμε τη μια από αυτές. Στο παράδειγμά μας θα αποσύρουμε τη X_4

- Καταλήγουμε έτσι στην εξίσωση παλινδρόμησης μεταξύ της Y και των X_1 και X_2

$$Y = 1.43X_1 + 0.65X_2 + 53.78$$

- με τυπικές αποκλίσεις $\sigma_1 = 0.121$ και $F_1 = 146.5$

  $\sigma_2 = 0.0459$ και $F_2 = 208.6$

- Καθώς οι δύο τιμές του F (F_1 και F_2) είναι εξαιρετικά μεγάλες (οι μεγαλύτερες που βρήκαμε σ' όλη τη διαδικασία βήμα-βήμα) αυτή η εξίσωση παλινδρόμησης δίνει την καλύτερη δυνατή εικόνα του φαινομένου. (Το καλύτερο θεωρητικό υπόδειγμα για το φαινόμενο που μελετούμε).

- Η μεταβλητή X_3 δεν προσθέτει τίποτε το σημαντικό στην Y —αυτό διαπιστώνεται από τον πίνακα συσχέτισης— γι' αυτό δεν τη λαμβάνουμε υπόψη.

Η εξίσωση παλινδρόμησης μεταξύ της Y και των X_1 και X_2

```
import numpy as np
# solve directly
from numpy import array
from numpy.linalg import inv
from matplotlib import pyplot
X = array([[1.0,  7,  26],
      [1.0,  1,  29],
      [1.0, 11, 56],
      [1.0, 11, 31],
      [1.0,  7, 52],
      [1.0, 11, 55],
      [1.0,  3, 71],
      [1.0,  2, 54],
      [1.0, 21, 47],
      [1.0,  1, 40],
      [1.0, 11, 66],
      [1.0, 10, 68]])
y=array([[78.5],[74.5],[104.3],[87.6],[95.9],[109.2],[102.7],[93.1],[115.9],[83.8],[113.3],[109.4]])
# linear least squares
b = inv(X.T.dot(X)).dot(X.T).dot(y)
print('Οι εκτιμήτριες β0, β1 και β2 είναι αντίστοιχα: ',b)
```

```
# predict using coefficients
yhat = X.dot(b)
```

Οι εκτιμήτριες β_0, β_1 και β_2 είναι αντίστοιχα: [[53.79306648] [1.43042] [0.64766871]]

$y=1.43*x_1+0.64*x_2+ 53.79$

5.4. Λογιστική παλινδρόμηση (logistic regression)

Η λογιστική παλινδρόμηση (Logistic regression) αποτελεί στην ουσία ένα μοντέλο ταξινόμησης των τιμών μιας μεταβλητής απόκρισης Y με βάση τη θεωρία των πιθανοτήτων. Στο μοντέλο αυτό όπου η μεταβλητή Y συνήθως έχει δυαδικό χαρακτήρα (λαμβάνει δύο τιμές) στοχεύεται η πρόβλεψη της έκβασης αυτής από ένα πλήθος προβλεπτικών μεταβλητών που μπορεί να είναι ονομαστικές, τακτικές ή ποσοτικές. Η σημαντικότερη διαφοροποίηση μεταξύ γραμμικής και λογιστικής παλινδρόμησης βασίζεται στη φύση της επιλεγμένης μεταβλητής απόκρισης, η οποία στην μεν πρώτη μπορεί να είναι αποκλειστικά ποσοτική, στη δε δεύτερη κατηγορική, (τακτική βλ. Garson (2011)) ή ονομαστική. Ενώ κατά την κλασική γραμμική παλινδρόμηση η εκτίμησητων παραμέτρων *α* και b_i γίνεται με τη μέθοδο των ελάχιστων τετραγώνων, κατά τη λογιστική παλινδρόμηση η εκτίμηση των παραμέτρων γίνεται με τη μέθοδο του λόγου πιθανοφάνειας (μέθοδος συνήθως εφαρμοζόμενη στα γενικευμένα γραμμικά υποδείγματα), δηλαδή επιλέγονται οι πιο πιθανοφανείς τιμές των παραμέτρων, προκειμένου να οδηγήσουν στα παρατηρούμενα αποτελέσματα. Ως επακόλουθο, η πρώτη παραδέχεται την ύπαρξη ομοιογένειας (ομοσκεδαστικότητας) στα υπολείμματα των αποκρίσεων ενώ στη δεύτερη αναπτύσσεται πάντα ετεροσκεδαστικότητα σε κάθε προβλεπόμενη τιμή εξαιτίας του μεταβαλλόμενου ποσοστού διακύμανσης που αναλογεί σε αυτήν.

Διακρίνονται τρεις τύποι λογιστικής παλινδρόμησης ανάλογα με την ιδιαίτερη φύση της εξαρτημένης κατηγορικής μεταβλητής η οποία μπορεί να είναι:

1. Δίτιμη ή δυαδική ή διχοτομική (binary) ή διμερής εξαρτημένη μεταβλητή. Συνίσταται από δύο κατηγορίες, όπως π.χ. είναι οι εκβάσεις επιτυχία/αποτυχία, ΝΑΙ/ΟΧΙ, γ εγονός απόν/παρόν.

2. Τακτική (ordinal) μεταβλητή. Η εξαρτημένη μεταβλητή συνίσταται από τρεις ή περισσότερες κατηγορίες μεταξύ των οποίων ισχύει η έννοια της ανισότητας, όπως π.χ. σε μια ερώτηση της κλίμακας διαφωνώ καθόλου, λίγο, μέτρια, αρκετά, πολύ, στην κατάταξη ενός στρώματος υλικού ως λεπτού, μεσαίου, παχέος.

3. Ονομαστική (Nominal) ή πολυωνυμική (polynomial) ή κατηγορική αδιαβάθμητη (non-ordered categorical) ή πολυμερής μεταβλητή απόκρισης. Περιέχει τρεις ή περισσότερες κατηγορίες χωρίς

κάποια φυσική διαβάθμιση, όπως π.χ. ο χαρακτηρισμός του χρώματος αντικειμένων ως ερυθρού, πράσινου, κίτ ρινου ή ενός τροφίμου ως τραγανού, μαλακού, εύθρυπτου.

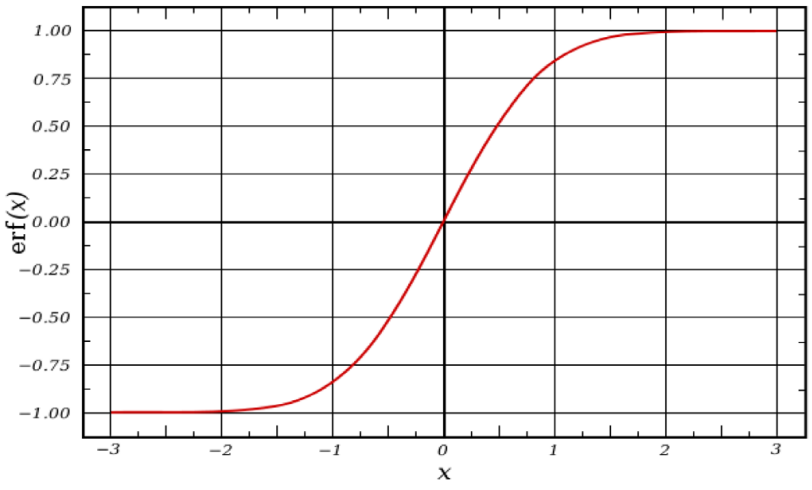
Η λογιστική παλινδρόμηση επινοήθηκε ως εναλλακτική επιλογή της γραμμικής διακριτικής ανάλυσης για την ταξινόμηση των στοιχείων (ονομαστικών ή τακτικών) της εξαρτημένης, με ευρεία απήχηση σε πολλά διαφορετικά επιστημονικά πεδία και κυρίως στην ιατρική και τις κοινωνικές επιστήμες. Χαρακτηριστικά, χρησιμοποιείται στην πρόβλεψη της:

- εμφάνισης ή μη μιας νόσου (π.χ. διαβήτης) από ένα σύνολο διαφορετικών χαρακτηριστικών του πάσχοντος ατόμου (ηλικία, φύλο, αιματολογικά, ηλεκτροκαρδιογράφημα κτλ.)
- πρόβλεψη της πρόθεσης αγοράς ενός αγαθού από έναν καταναλωτή (έρευνα αγοράς)
- πιθανότητας αποτυχίας μιας διεργασίας παραγωγής προϊόντος σε ένα εργοστάσιο τροφίμων
- επιλογής ενός πολιτικού κόμματος με βάση την καταγραφή των δημογραφικών στοιχείων των πολιτών, όπως είναι η ηλικία, φύλο, φυλή, τόπος διαμονής, εισόδημα, προηγούμενη ψηφοφορία
- πιθανότητας αθέτησης από δανειολήπτη της αποπληρωμής του δανείου του.

Για περισσότερες λεπτομέρειες σχετικά με τις μεθόδους της λογιστικής παλινδρόμησης μπορείτε να ανετρέξετε στις εργασίες και τα συγγράματα των Cox & Snell (1989), των Hosmer & Lemeshow (2000), των Long & Freese (2014) και συνδυαστικά με τη χρήση των πινάκων ενδεχομένων από τους Everitt (1992) και Agresti (1996).

5.4.1. Ανάπτυξη του μοντέλου

Στην επιστήμη της στατιστικής, η λογιστική παλινδρόμηση χρησιμοποιείται για την πρόβλεψη της πιθανότητας εμφάνισης ενός γεγονότος προσαρμόζοντας τα δεδομένα της μελέτης στην εξίσωση της λογιστικής καμπύλης όπως αυτή παρουσιάζεται στο διάγραμμα 5.15.



Διάγραμμα 5.15: Εξίσωση της λογιστικής καμπύλης

Η καμπύλη αυτή έχει σιγμοειδή μορφή και χαρακτηρίζεται από ένα στάδιο εκθετικής ανάπτυξης στο οποίο ο ρυθμός αύξησης επιβραδύνεται βαθμιαία και περατώνεται στο ασυμπτωτικό στάδιο κορεσμού της ανάπτυξης (η ευθεία βαίνει τελικά παράλληλα στον άξονα *X*).

Η δυαδική λογιστική παλινδρόμηση αποτελεί μια διωνυμική εξίσωση στην οποία η μεταβλητή απόκρισης *Y* είναι το τυχαίο αποτέλεσμα εμφάνισης μιας από δύο δυνητικές εκβάσεις του τύπου επιτυχία ή αποτυχία όπως για παράδειγμα είναι η ρίψη ενός ζαριού όπου το αποτέλεσμα εμφάνισης του αριθμού έξι (6) θεωρείται επιτυχία και των λοιπών αριθμών αποτυχία, το αποτέλεσμα της ρίψης ενός νομίσματος δύο διαφορετικών όψεων (κορώνα-γράμματα), η θετική ψήφος εκλογής ενός πολιτικού εκπροσώπου κ.α.

Η δίτιμη λογιστική παλινδρόμηση έχει τη μορφή

$$f(z)=\frac{e^z}{1+e^z}=\frac{1}{1+e^{-z}}$$

όπου *z* είναι η μεταβλητή εισόδου και *f(z)* το αποτέλεσμα αυτής. Στα πλεονεκτήματα της εξίσωσης συγκαταλέγεται και το γεγονός ότι η μεταβλητή εισόδου λαμβάνει θετικές και αρνητικές τιμές ενώ το αποτέλεσμα αυτής *f(z)* περιορίζεται σε εύρος τιμών μεταξύ 0 και 1. Αναλυτικότερα, η μεταβλητή *z* εκπροσωπεί τη δράση μιας ομάδας ανεξάρτητων μεταβλητών ενώ η *f(z)* προσδιορίζει την πιθανότητα ενός συγκεκριμένου αποτελέσματος λόγω της δράσης της ομάδας αυτής. Η μεταβλητή *z* (λογιστική) εκφράζει επίσης το μέτρο της ολικής συνεισφοράς όλων των συμμετεχουσών ανεξάρτητων μεταβλητών στο μοντέλο και ορίζεται ως

$$z=\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_k X_k$$

όπου β_0 είναι το ύψος της κλίσης της γραμμής παλινδρόμησης και ισούται με την τιμή z όταν οι τιμές όλων των ανεξάρτητων μεταβλητών ισούνται με 0, ενώ β_i είναι οι συντελεστές παλινδρόμησης καθένας των οποίων εκφράζει το μέγεθος συνεισφοράς της αντίστοιχης μεταβλητής. Θετική τιμή του συντελεστή δηλώνει ότι η επεξηγηματική μεταβλητή αυξάνει την πιθανότητα της επιτυχημένης έκβασης (να συμβεί δηλαδή το γεγονός), αρνητική τιμή σημαίνει ότι η μεταβλητή μειώνει την πιθανότητα αυτής της έκβασης. Υψηλή τιμή του συντελεστή σημαίνει ότι η ανεξάρτητη μεταβλητή επηρεάζει πολύ ισχυρά την πιθανότητα να συμβεί το γεγονός ή μη, ενώ χαμηλή τιμή δηλώνει μικρή επίδραση της ανεξάρτητης μεταβλητής στην πιθανότητα εμφάνισης της ανάλογης έκβασης.

$$logit(p)=\log_e\left(\frac{p}{1-p}\right)=\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_k X_k$$

Οι συντελεστές της παλινδρόμησης υπολογίζονται με τη βοήθεια της εκτίμησης της μέγιστης πιθανοφάνειας (Maximum Likelihood Estimate – MLE), ως

$$L = \prod_{i=1}^{n} f(x_i \theta)$$

ή με τη λογαριθμική μορφή,

$$L = \sum_{i=1}^{n} \log_e f(x_i \theta)$$

όπου ϑ είναι μια παράμετρος της μεταβλητής η οποία μπορεί να μεταβάλλεται ελεύθερα. Η προβλεπόμενη τιμή για κάθε παρατήρηση θα ισούται με

$$\hat{l} = \frac{1}{n} \log_e L$$

Η συνάρτηση της πιθανοφάνειας έκβασης ενός γεγονότος (likelihood) δείχνει πόσο κατάλληλα ένα παρατηρούμενο δείγμα περιγράφεται από κάποιες τιμές παραμέτρων όπως για παράδειγμα ο μέσος όρος, η τυπική απόκλιση. Επομένως, η μεγιστοποίηση της συνάρτησης της πιθανότητας έκβασης καθορίζει τις παραμέτρους εκείνες που είναι οι πλέον ικανές να παράγουν τα παρατηρούμενα στοιχεία. Από άποψη στατιστικής βαρύτητας, η MLE προτείνεται για εφαρμογές σε μεγάλα δείγματα καθόσον είναι ευέλικτη, προσαρμόζεται εύκολα στην παραγωγή πολλών διαφορετικού τύπου μοντέλων, το χειρισμό διαφορετικής φύσης στοιχείων και περιέχει ακριβέστερες μετρήσεις. Η αξιοπιστία των αποτελεσμάτων της λογιστικής παλινδρόμησης επηρεάζεται κατά πολύ από το δειγματοληπτικό μέγεθος της έρευνας.

5.5. Μελέτη περίπτωσης στην Παλινδρόμηση

Κατεβάστε από την ιστοσελίδα https://archive.ics.uci.edu/ml/datasets/Forest+Fires σύνολο δεδομένων για πυρκαγιές από περιοχές στην Πορτογαλία (δίνεται επίσης το αρχείο δεδομένων forestfires.xlsx και σε μορφή xlsx). Τα δεδομένα περιέχουν γεωγραφικά και μετεωρολογικά στοιχεία όταν εκδηλώθηκαν πυρκαγιές καθώς επίσης και την επιφάνεια που κάηκε που μετριέται σε εκτάρια (hectars, 1 εκτάριο = 10 στρέμματα). Έχοντας ως στόχο την πρόβλεψη της επιφάνειας που θα καεί βάση των μετεωρολογικών συνθηκών που επικρατούν συγγράψτε κώδικα Python που εκτιμάει τους συντελεστές του παρακάτω μοντέλου παλινδρόμησης με τον τρόπο που ζητείται: $area = \beta_1 temp + \beta_2 wind + \beta_3 rain + \beta_0$

Εκτιμήστε τους συντελεστές του παραπάνω μοντέλου παλινδρόμησης με τη μέθοδο των ελαχίστων τετραγώνων, αλλά χρησιμοποιείστε μόνο εκείνες τις παρατηρήσεις όπου η τιμή επιφάνειας (μεταβλητή area) είναι μεγαλύτερη από 0.1 εκτάρια και μικρότερη από 3.2 εκτάρια (area>0.1 και area < 3.2) και χαρακτηρίζει μικρές πυρκαγιές.

1. Να υπολογιστούν οι εκτιμήτριες β_0 , β_1 ,β_2 και β_3 της εξίσωσης ελαχίστων τετραγώνων.
2. Να σχεδιαστεί το διάγραμμα διασποράς (scatter plot) μεταξύ της θερμοκρασίας *temp* και της επιφάνειας *area* στη Python και να δοθεί ο κώδικας του script της Python.

Επομένως, θα πρέπει να ανέβουν συνολικά 2 αρχεία (μπορεί και ένα αρχείο zip με 2 αρχεία ή ξεχωριστά αρχεία).

Ένα γράφημα διασποράς (αρχείο εικόνας) και **ένα** αρχείο της Python.

5.6. *Ερωτήσεις αυτοαξιολόγησης*

5.1 Σε ποια περίπτωση η τιμή του συντελεστή προσδιορισμού R^2 είναι ίση με την τιμή του διορθωμένου συντελεστή προσδιορισμού (Adjusted R^2);

α) Οι τιμές του συντελεστή προσδιορισμού R^2 και διορθωμένου συντελεστή προσδιορισμού R^2adj είναι ίσες μόνο στην περίπτωση όπου το πλήθος των μεταβλητών στο μοντέλο παλινδρόμησης είναι ίσο με το 0

β) Σε καμία περίπτωση δεν είναι ίση η τιμή του συντελεστή προσδιορισμού R^2 με την τιμή του διορθωμένου συντελεστή προσδιορισμού (Adjusted R^2)

γ) Μόνο στην περίπτωση όπου το πλήθος των μεταβλητών στο μοντέλο παλινδρόμησης είναι διάφορο του μηδενός (συγκεκριμένα είναι μεγαλύτερο του 0)

δ) Μόνο στην περίπτωση όπου το πλήθος των μεταβλητών στο μοντέλο παλινδρόμηση είναι αρνητικό

5.2 Γιατί η συνάρτηση κόστους της Σταδιακής Καθόδου επιτρέπει τη σύγκριση των τιμών της συνάρτησης κόστους για δύο διαφορετικά μοντέλα παλινδρόμησης με την ίδια εξαρτημένη μεταβλητή, αν το μόνο που αλλάζει είναι το σύνολο δεδομένων;

α) Η σύγκριση των τιμών της συνάρτησης κόστους επιτρέπεται επειδή η συνάρτηση κόστους J() υπολογίζει το σφάλμα ως απόλυτο αριθμό

β) Στη μέθοδο Σταδιακής Καθόδου η συνάρτηση κόστους J(), επειδή διαιρεί με το πλήθος των παρατηρήσεων στο σύνολο εκπαίδευσης m, υπολογίζει επί της ουσίας το σφάλμα κατά μέσο όρο (το μέσο σφάλμα), και όχι ως απόλυτο αριθμό. Επειδή ακριβώς υπολογίζει το μέσο σφάλμα, μπορούν οι τιμές δύο διαφορετικών συναρτήσεων κόστους να συγκριθούν

γ) Δεν μπορεί να γίνει η σύγκριση

5.3 Στην περίπτωση της μεθόδου των ελαχίστων τετραγώνων, μπορεί να γίνει σύγκριση των τιμών της συνάρτησης κόστους για δύο διαφορετικά μοντέλα παλινδρόμησης με την ίδια εξαρτημένη μεταβλητή, αν το μόνο που αλλάζει είναι το σύνολο δεδομένων;

α) Μπορεί να γίνει η σύγκριση των τιμών της συνάρτησης κόστους μόνο αν δεν αλλάζει το σύνολο των δεδομένων

β) Δεν μπορεί να γίνει η σύγκριση των τιμών της συνάρτησης κόστους για τη μέθοδο των ελαχίστων τετραγώνων, επειδή αυτή υπολογίζει το απόλυτο σφάλμα και όχι το μέσο σφάλμα

γ) Μπορεί να γίνει σύγκριση επειδή στη μέθοδο ελαχίστων τετραγώνων, η συνάρτηση κόστους υπολογίζει το σφάλμα κατά μέσο όρο

δ) Μπορεί να γίνει σύγκριση επειδή στη μέθοδο ελαχίστων τετραγώνων, η συνάρτηση κόστους υπολογίζει το απόλυτο σφάλμα

5.4 Ποια από τις παρακάτω καμπύλες χρησιμοποιείται συνήθως για να απεικονίσει τη σχέση μεταξύ των ανεξάρτητων μεταβλητών και της πιθανότητας στην λογιστική παλινδρόμηση;

α) Γραμμική

β) Εκθετική

γ) Παράβολική

δ) Σιγμοειδής

ΚΕΦΑΛΑΙΟ 6: Αξιολόγηση Απόδοσης

6.1. Διαχωρισμός δεδομένων σε δοκιμής/εκπαίδευσης (train/test)

Κατά τη δημιουργία ενός μοντέλου μηχανικής μάθησης, είναι σημαντικό να αξιολογήσετε το εκπαιδευμένο μοντέλο σε δεδομένα που δεν χρησιμοποιήθηκαν κατά την εκπαίδευσή του, καθώς η γενίκευση (generalization) είναι περισσότερο σημαντική από την απομνημόνευση (που σημαίνει ότι θέλουμε έναν κανόνα που γενικεύεται σε νέα δεδομένα και όχι σύγκριση με δεδομένα που απομνημονεύσαμε).

Είναι πιο δύσκολο να καταλήξουμε σε σωστό συμπέρασμα από περιστατικά που δεν έχουν δει ποτέ στο παρελθόν από ό,τι σε αυτά που έχουμε ήδη δει. Η σωστή αξιολόγηση γίνεται εύκολα αφήνοντας έξω ένα υποσύνολο των δεδομένων κατά την εκπαίδευση του μοντέλου και χρησιμοποιώντας το στη συνέχεια για την αξιολόγηση του μοντέλου. Τα δεδομένα που χρησιμοποιήθηκαν στην προσαρμογή ενός μοντέλου ονομάζονται δεδομένα εκπαίδευσης (training data) ενώ τα δεδομένα που χρησιμοποιούνται για την αξιολόγηση ενός μοντέλου ονομάζονται δεδομένα δοκιμής (test data).

Θεωρητικά, μπορούμε να έχουμε δύο ξεχωριστά σύνολα δεδομένων: ένα εκπαιδευτικό (train) και ένα δοκιμαστικό (test). Ωστόσο, η ύπαρξη ξεχωριστών συνόλων δεδομένων σε δύο διαφορετικά αρχεία είναι ασυνήθιστο: τις περισσότερες φορές έχουμε ένα μόνο αρχείο που περιέχει όλα τα δεδομένα και πρέπει να τα χωρίσουμε μόλις φορτωθεί στη μνήμη. Το Scikit-learn παρέχει τη βοηθητική συνάρτηση sklearn.model_selection.train_test_split που χρησιμοποιείται για να χωρίσει αυτόματα το σύνολο δεδομένων σε δύο υποσύνολα.

```
import pandas as pd
from sklearn.model_selection import train_test_split

# Αρχικά, ας φορτώσουμε το πλήρες σύνολο δεδομένων adult census.
adult_census = pd.read_csv("adult-census.csv")

# Διαχωρίζουμε το στόχο από τα δεδομένα
data, target = adult_census.drop(columns="class"),
adult_census["class"]

data_train, data_test, target_train, target_test = train_test_split(
data, target, random_state=42, test_size=0.25
)
```

Όταν καλούμε τη συνάρτηση train_test_split, καθορίσαμε ότι θα θέλαμε να έχει το 25% των δειγμάτων στο σετ δοκιμών, ενώ τα υπόλοιπα δείγματα (75%) ανατίθενται στο σετ εκπαίδευσης.

6.2. Μετρικές ταξινόμησης

Στη μηχανική μάθηση οι μετρικές ταξινόμησης είναι απαραίτητες για την αξιολόγηση της απόδοσης των μοντέλων που προσπαθούν να κατηγοριοποιήσουν δεδομένα σε διακριτές κατηγορίες. Οι πιο κοινές μετρικές για την αξιολόγηση των αλγορίθμων ταξινόμησης περιλαμβάνουν την ακρίβεια (accuracy), την precision, το recall, τον πίνακα σύγχυσης (confusion matrix), το F1-score και τα ROC-AUC.

6.2.1. Κατώφλια (threshold) και ο πίνακας σύγχυσης (confusion matrix)

Ας υποθέσουμε ότι έχετε ένα μοντέλο λογιστικής παλινδρόμησης για την ανίχνευση ανεπιθύμητων μηνυμάτων (spam-email), το οποίο προβλέπει μια τιμή μεταξύ 0 και 1, αντιπροσωπεύοντας την πιθανότητα ότι ένα δεδομένο email είναι ανεπιθύμητο. Μια πρόβλεψη 0,50 σηματοδοτεί 50% πιθανότητα ότι το email είναι ανεπιθύμητο, μια πρόβλεψη 0,75 σηματοδοτεί 75% πιθανότητα και ούτω καθεξής.

Θα θέλατε να αναπτύξετε αυτό το μοντέλο σε μια εφαρμογή ηλεκτρονικού ταχυδρομείου για να φιλτράρει τα ανεπιθύμητα μηνύματα σε έναν ξεχωριστό φάκελο. Για να το κάνετε αυτό, πρέπει να μετατρέψετε την αριθμητική έξοδο του μοντέλου (π.χ., 0,75) σε μία από τις δύο κατηγορίες: "ανεπιθύμητο" ή "όχι ανεπιθύμητο".

Για να κάνετε αυτήν τη μετατροπή, επιλέγετε ένα όριο πιθανότητας, το οποίο ονομάζεται κατώφλι ταξινόμησης (classification threshold). Τα παραδείγματα με πιθανότητα πάνω από την τιμή του ορίου ταξινομούνται στη θετική κατηγορία, την κατηγορία που δοκιμάζετε (εδώ, ανεπιθύμητο). Τα παραδείγματα με χαμηλότερη πιθανότητα ταξινομούνται στην αρνητική κατηγορία, την εναλλακτική κατηγορία (εδώ όχι ανεπιθύμητο).

Ας υποθέσουμε ότι το μοντέλο βαθμολογεί ένα email με 0,99, προβλέποντας ότι το email έχει 99% πιθανότητα να είναι ανεπιθύμητο, και ένα άλλο email με 0,51, προβλέποντας ότι έχει 51% πιθανότητα να είναι ανεπιθύμητο. Αν ορίσετε το κατώφλι ταξινόμησης στο 0,5, το μοντέλο θα ταξινομήσει και τα δύο email ως ανεπιθύμητα. Αν ορίσετε το κατώφλι στο 0,95, μόνο το email που βαθμολογήθηκε με 0,99 θα ταξινομηθεί ως ανεπιθύμητο.

Αν και το 0,5 μπορεί να φαίνεται μια λογική επιλογή για το όριο, δεν είναι πάντα η καλύτερη επιλογή αν το κόστος μιας λανθασμένης ταξινόμησης διαφέρει πολύ (μεταξύ των δύο κλάσεων) ή

αν οι κατηγορίες δεν είναι ισοβαρείς. Για παράδειγμα, αν μόνο το 0,01% των email είναι ανεπιθύμητο, ή αν είναι χειρότερο να ταξινομήσετε ένα νόμιμο email ως ανεπιθύμητο από το να αφήσετε ένα ανεπιθύμητο email στο εισερχόμενο, τότε το να χαρακτηριστούν όλα τα email που το μοντέλο θεωρεί τουλάχιστον 50% πιθανό ως ανεπιθύμητα, θα έχει αρνητικά αποτελέσματα.

6.2.1.1 Πίνακας σύγχυσης (confusion matrix)

Η βαθμολογία πιθανότητας δεν είναι η πραγματικότητα ή η αλήθεια (groud truth). Υπάρχουν τέσσερα πιθανά αποτελέσματα για κάθε έξοδο από έναν δυαδικό ταξινομητή. Για το παράδειγμα του ταξινομητή ανεπιθύμητων μηνυμάτων, αν τοποθετήσετε την αλήθεια ως στήλες και την πρόβλεψη του μοντέλου ως γραμμές, προκύπτει ο παρακάτω πίνακας, ο οποίος ονομάζεται πίνακας σύγχυσης (confusion matrix):

	Πραγματικό θετικό	**Πραγματικό αρνητικό**
Προβλεπόμενο θετικό	**Αληθές θετικό (True Positive - TP):** Ένα ανεπιθύμητο μήνυμα που ταξινομήθηκε σωστά ως ανεπιθύμητο. Αυτά είναι τα ανεπιθύμητα μηνύματα που μεταφέρονται αυτόματα στο φάκελο ανεπιθύμητων.	**Ψευδές θετικό (False positive - FP):** Ένα μη ανεπιθύμητο μήνυμα που ταξινομήθηκε λανθασμένα ως ανεπιθύμητο. Αυτά είναι τα νόμιμα μηνύματα που καταλήγουν στο φάκελο ανεπιθύμητων.
Προβλεπόμενο αρνητικό	**Ψευδές αρνητικό (False Negative - FN):** Ένα ανεπιθύμητο μήνυμα που ταξινομήθηκε λανθασμένα ως μη ανεπιθύμητο. Αυτά είναι τα ανεπιθύμητα μηνύματα που δεν πιάστηκαν από το φίλτρο και φτάνουν στα εισερχόμενα.	**Αληθές αρνητικό (True Negative - TN):** Ένα μη ανεπιθύμητο μήνυμα που ταξινομήθηκε σωστά ως μη ανεπιθύμητο. Αυτά είναι τα νόμιμα μηνύματα που στέλνονται κατευθείαν στα εισερχόμενα.

Πίνακας 6.1: Πίνακας σύγχυσης για το πρόβλημα καηγοριοποίησης ανεπιθύμητων email

Παρατηρήστε ότι το άθροισμα σε κάθε σειρά δίνει όλα τα προβλεπόμενα θετικά (TP + FP) και όλα τα προβλεπόμενα αρνητικά (FN + TN), ανεξαρτήτως εγκυρότητας. Το άθροισμα σε κάθε στήλη, από την άλλη, δίνει όλα τα πραγματικά θετικά (TP + FN) και όλα τα πραγματικά αρνητικά (FP + TN), ανεξαρτήτως ταξινόμησης του μοντέλου. Όταν το σύνολο των πραγματικών θετικών δεν είναι κοντά στο σύνολο των πραγματικών αρνητικών, το σύνολο δεδομένων είναι μη ισορροπημένο.

6.2.2. Ακρίβεια (Accuracy)

Ορίζει το ποσοστό των σωστών προβλέψεων σε σχέση με το συνολικό αριθμό προβλέψεων. Είναι μια απλή μετρική που χρησιμοποιείται ευρέως, αλλά μπορεί να παραπλανήσει σε περιπτώσεις ανισορροπίας κλάσεων, όπου μία από τις κλάσεις εμφανίζεται πολύ πιο συχνά από την άλλη.

$$\text{Accuracy} = \frac{\text{correct classifications}}{\text{total classifications}} = \frac{TP + TN}{TP + TN + FP + FN}$$

6.2.3. Precision

Το Precision ορίζει το ποσοστό των σωστών προβλέψεων για μία συγκεκριμένη κλάση, σε σχέση με το σύνολο των προβλέψεων αυτής της κλάσης. Είναι ιδιαίτερα χρήσιμο όταν θέλουμε να εστιάσουμε στην ποιότητα των προβλέψεων μιας θετικής κατηγορίας.

$$\text{Precision} = \frac{\text{correctly classified actual positives}}{\text{everything classified as positive}} = \frac{TP}{TP + FP}$$

Στο παράδειγμα ταξινόμησης ανεπιθύμητων μηνυμάτων, το precision μετρά το ποσοστό των email που ταξινομήθηκαν ως ανεπιθύμητα και ήταν πράγματι ανεπιθύμητα. Ένα υποθετικά τέλειο μοντέλο θα είχε μηδενικά ψευδώς θετικά (false positives) και, επομένως, ακρίβεια 1,0.

6.2.4. Recall ή αλλιώς true positive rate (TPR)

Το Recall ορίζει το ποσοστό των σωστών θετικών προβλέψεων σε σχέση με το σύνολο των πραγματικών θετικών δειγμάτων. Είναι σημαντικό σε περιπτώσεις όπου θέλουμε να αποφύγουμε την παράλειψη θετικών προβλέψεων.

$$\text{Recall (or TPR)} = \frac{\text{correctly classified actual positives}}{\text{all actual positives}} = \frac{TP}{TP + FN}$$

Τα ψευδώς αρνητικά (FN) είναι πραγματικά θετικά που ταξινομήθηκαν λανθασμένα ως αρνητικά, γι' αυτό εμφανίζονται στον παρονομαστή. Στο παράδειγμα ταξινόμησης ανεπιθύμητων μηνυμάτων, το recall μετρά το ποσοστό των ανεπιθύμητων μηνυμάτων που ταξινομήθηκαν σωστά ως ανεπιθύμητα. Γι' αυτό ένας άλλος όρος για το recall είναι η πιθανότητα ανίχνευσης: απαντά στο ερώτημα "Ποιο ποσοστό των ανεπιθύμητων μηνυμάτων ανιχνεύεται από αυτό το μοντέλο;"

Ένα υποθετικά τέλειο μοντέλο θα είχε μηδενικά ψευδώς αρνητικά και, επομένως, Recall (TPR) 1,0, δηλαδή ποσοστό ανίχνευσης 100%. Σε ένα μη ισορροπημένο σύνολο δεδομένων, όπου ο αριθμός των πραγματικών θετικών είναι πολύ, πολύ χαμηλός, για παράδειγμα 1-2 παραδείγματα συνολικά, η ανάκληση είναι λιγότερο χρήσιμη ως μετρική.

6.2.5. Ρυθμός ψευδώς θετικών (False Positive Rate - FPR)

Ο ρυθμός ψευδώς θετικών (False Positive Rate - FPR) είναι το ποσοστό όλων των πραγματικών αρνητικών που ταξινομήθηκαν λανθασμένα ως θετικά. Ο μαθηματικός ορισμός του είναι:

$$\text{FPR} = \frac{\text{incorrectly classified actual negatives}}{\text{all actual negatives}} = \frac{FP}{FP + TN}$$

Τα ψευδώς θετικά είναι πραγματικά αρνητικά που ταξινομήθηκαν λανθασμένα, γι' αυτό εμφανίζονται στον παρονομαστή. Στο παράδειγμα ταξινόμησης ανεπιθύμητων μηνυμάτων, ο FPR μετρά το ποσοστό των νόμιμων email που ταξινομήθηκαν λανθασμένα ως ανεπιθύμητα.

Ένα τέλειο μοντέλο θα είχε μηδενικά ψευδώς θετικά και, επομένως, FPR 0,0. Σε ένα μη ισορροπημένο σύνολο δεδομένων, όπου ο αριθμός των πραγματικών αρνητικών είναι πολύ, πολύ χαμηλός, για παράδειγμα 1-2 παραδείγματα συνολικά, ο FPR λιγότερο χρήσιμος ως μετρική.

6.2.6. F1-Score

Η βαθμολογία F1 (F1 score) είναι ο αρμονικός μέσος (ένας τύπος μέσου όρου) της ακρίβειας (precision) και της ανάκλησης (recall). Μαθηματικά, δίνεται από τον εξής τύπο:

$$\text{F1} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$$

Αυτή η μετρική ισορροπεί τη σημασία των Precision και Recall, και προτιμάται σε σύνολα δεδομένων με μη ισορροπημένες κατηγορίες έναντι της ακρίβειας (accuracy). Όταν τα Precision και Recall έχουν και τα δύο τέλεια βαθμολογία 1,0, η F1 θα έχει επίσης τέλεια βαθμολογία 1,0. Σε γενικότερο πλαίσιο, όταν Precision και Recall έχουν παρόμοια τιμή, η F1 θα είναι κοντά σε αυτή την τιμή. Όταν Precision και Recall διαφέρουν πολύ, η F1 θα είναι πιο κοντά στη χειρότερη από τις δύο μετρικές.

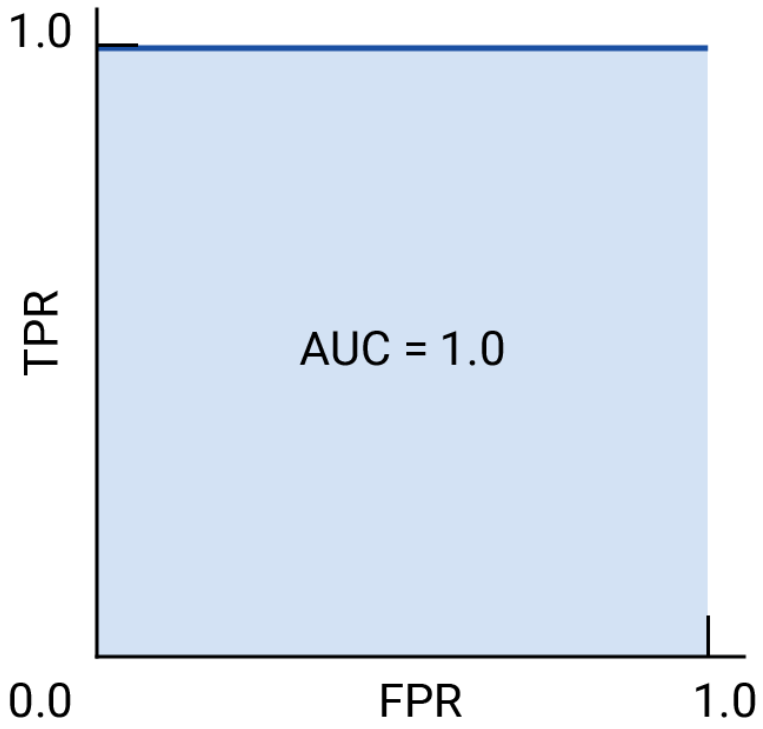
6.2.7. ROC και AUC

Οι προηγούμενες μετρικές υπολογίζονται σε μια συγκεκριμένη τιμή κατωφλίου ταξινόμησης. Ωστόσο, αν θέλετε να αξιολογήσετε την ποιότητα ενός μοντέλου σε όλα τα πιθανά κατώφλια, χρειάζεστε διαφορετικά εργαλεία.

6.2.7.1 Receiver-operating characteristic – ROC

Η καμπύλη ROC είναι μια οπτική αναπαράσταση της απόδοσης του μοντέλου σε όλα τα πιθανά κατώφλια ταξινόμησης. Η πλήρης ονομασία της, "receiver operating characteristic", προέρχεται από την εποχή του Β' Παγκοσμίου Πολέμου και χρησιμοποιήθηκε για την ανίχνευση μέσω ραντάρ.

Η καμπύλη ROC σχεδιάζεται υπολογίζοντας τον ρυθμό αληθών θετικών (TPR) και τον ρυθμό ψευδώς θετικών (FPR) σε κάθε πιθανό κατώφλι (στην πράξη, σε επιλεγμένα διαστήματα), και στη συνέχεια, γίνεται η γραφική παράσταση του TPR πάνω από τον FPR. Ένα τέλειο μοντέλο, το οποίο σε κάποιο κατώφλι έχει TPR 1.0 και FPR 0.0, μπορεί να αναπαρασταθεί είτε από ένα σημείο στο (0, 1) αν αγνοηθούν τα υπόλοιπα κατώφλια, είτε ως εξής:



Διάγραμμα 6.1: ROC και AUC από ένα υποθετικό «τέλειο» μοντέλο

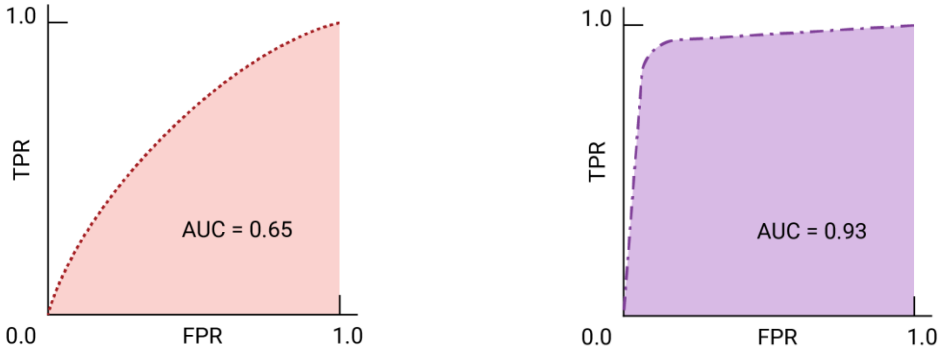
6.2.8. Area under the ROC curve (AUC)

Η περιοχή κάτω από την καμπύλη ROC (Area under the ROC curve - AUC) αντιπροσωπεύει την πιθανότητα ότι το μοντέλο, αν του δοθεί ένα τυχαία επιλεγμένο θετικό και αρνητικό παράδειγμα, θα ταξινομήσει το θετικό υψηλότερα από το αρνητικό.

Το τέλειο μοντέλο, που σχηματίζει ένα τετράγωνο με πλευρές μήκους 1, έχει περιοχή (εμβαδό) κάτω από την καμπύλη (AUC) ίση με 1,0. Αυτό σημαίνει ότι υπάρχει 100% πιθανότητα το μοντέλο να ταξινομήσει σωστότερα ένα τυχαία επιλεγμένο θετικό παράδειγμα από ένα τυχαία επιλεγμένο αρνητικό παράδειγμα. Με άλλα λόγια, ένας ταξινομητής ανεπιθύμητων μηνυμάτων με AUC ίση με 1,0 θα αποδίδει πάντα υψηλότερη πιθανότητα ότι ένα τυχαίο ανεπιθύμητο email είναι

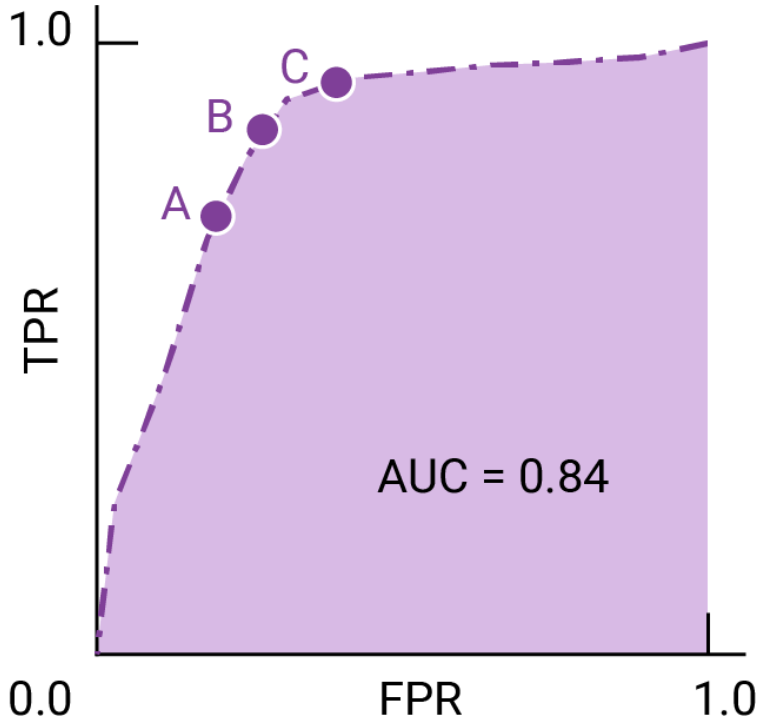
ανεπιθύμητο από ένα τυχαίο νόμιμο email. Η πραγματική ταξινόμηση κάθε email εξαρτάται από το όριο που θα επιλέξετε.

Η AUC είναι μια χρήσιμη μετρική για τη σύγκριση της απόδοσης δύο διαφορετικών μοντέλων, εφόσον το σύνολο δεδομένων είναι περίπου ισορροπημένο. Το μοντέλο με τη μεγαλύτερη περιοχή κάτω από την καμπύλη είναι γενικά το καλύτερο.



Διάγραμμα 6.2: ROC και AUC δύο υποθετικών μοντέλων. Η καμπύλη στα δεξιά, με μεγαλύτερη AUC, αντιπροσωπεύει το καλύτερο από τα δύο μοντέλα.

Τα σημεία σε μια καμπύλη ROC που βρίσκονται πιο κοντά στο (0,1) αντιπροσωπεύουν μια περιοχή με τα καλύτερα κατώφλια για το εκάστοτε μοντέλο. Όπως συζητήθηκε στην ενότητα «Κατώφλια και Πίνακας σύγχυσης», το κατώφλι που θα επιλέξετε εξαρτάται από το ποια μετρική είναι πιο σημαντική για τη συγκεκριμένη εφαρμογή που θα χρησιμοποιηθεί το μοντέλο. Δείτε τα σημεία A, B και C στο παρακάτω διάγραμμα, το καθένα από τα οποία αντιπροσωπεύει ένα όριο:

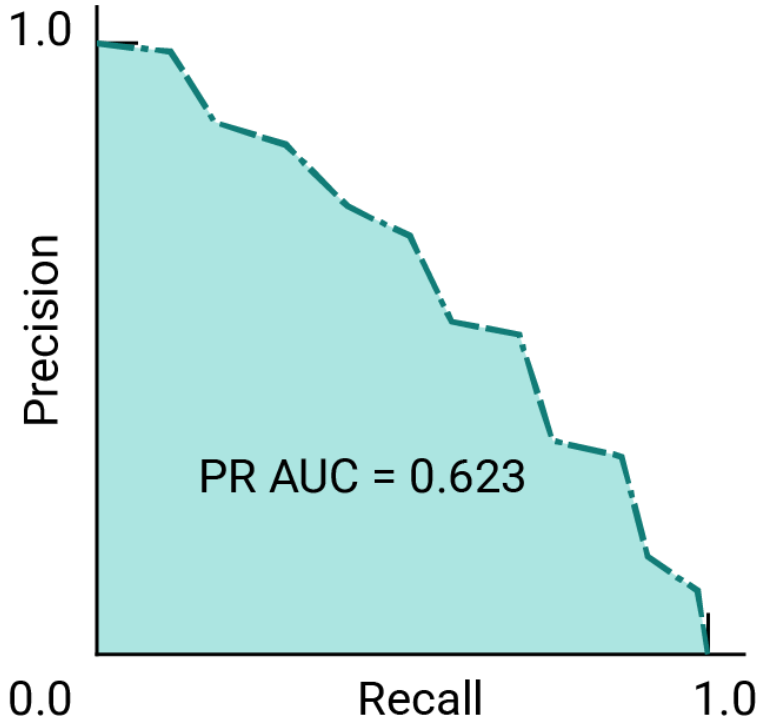


Διάγραμμα 6.3: Καμπύλη ROC με AUC=0.84, που δείχνει τρία σημεία (A,B,C) πλησιέστερα στο (0,1).

Εάν τα ψευδώς θετικά είναι ιδιαίτερα δαπανηρά, μπορεί να έχει νόημα να επιλέξετε ένα κατώφλι που δίνει χαμηλότερο FPR, όπως το σημείο A, ακόμα κι αν μειωθεί το TPR. Αντιθέτως, αν τα ψευδώς αρνητικά είναι πολύ δαπανηρά, το κατώφλι για το σημείο C, που μεγιστοποιεί το TPR, μπορεί να είναι προτιμότερο. Εάν τα κόστη είναι περίπου ισοδύναμα, το σημείο B μπορεί να προσφέρει την καλύτερη ισορροπία μεταξύ TPR και FPR.

6.2.9. Precision-Recall Curve (PRC)

Η AUC και η ROC λειτουργούν ικανοποιητικά για τη σύγκριση μοντέλων όταν το σύνολο δεδομένων είναι περίπου ισορροπημένο μεταξύ των κατηγοριών. Όταν το σύνολο δεδομένων είναι μη ισορροπημένο, οι καμπύλες precision-recall (PRCs) και η περιοχή κάτω από αυτές τις καμπύλες μπορεί να προσφέρουν μια καλύτερη συγκριτική οπτική αναπαράσταση της απόδοσης του μοντέλου. Οι καμπύλες precision-recall δημιουργούνται με την απεικόνιση του Precision στον κατακόρυφο άξονα και του Recall στον οριζόντιο άξονα, σε όλα τα κατώφλια ταξινόμησης.



Διάγραμμα 6.4: Παράδειγμα Precision-Recall Curve (PRC)

6.2.10. Παραδείγματα στο scikit-learn

Η ενότητα sklearn.metrics του scikit-learn παρέχει τις πιο συνηθισμένες μετρικές σε μορφή συναρτήσεων.

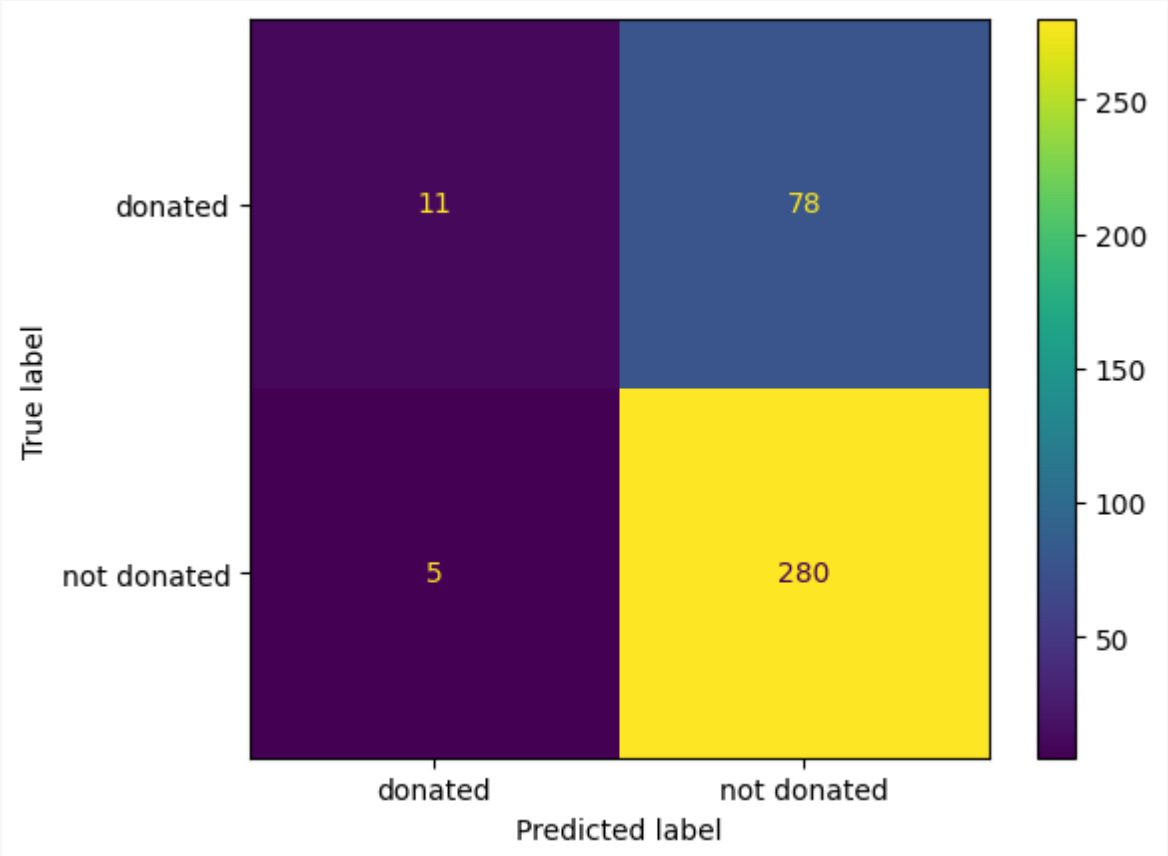
6.2.10.1 Παράδειγμα για την Ακρίβεια

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

# Αρχικά, ας φορτώσουμε το πλήρες σύνολο δεδομένων blood_transfusion.
blood_transfusion = pd.read_csv("blood_transfusion.csv")
data = blood_transfusion.drop(columns="Class")
target = blood_transfusion["Class"]

# Διαχωρισμός σε σετ εκπαίδεσης/τεστ
data_train, data_test, target_train, target_test = train_test_split(
data, target, shuffle=True, random_state=0, test_size=0.5
)
```

```
# Προσαρμογή μοντέλου λογιστικής παλινδρόμησης
classifier = LogisticRegression()
classifier.fit(data_train, target_train)

# Υπολογισμός ακρίβειας
accuracy = accuracy_score(target_test, target_predicted)
print(f"Accuracy: {accuracy:.3f}")
# Accuracy: 0.778

# Η LogisticRegression έχει επίσης μια μέθοδο που ονομάζεται score
#(μέρος του scikit-learn API), το οποίο υπολογίζει τη μετρική της
#ακρίβειας
classifier.score(data_test, target_test)
# 0.7780748663101604
```

6.2.10.2 *Παράδειγμα για το Confusion Matrix*

```
from sklearn.metrics import ConfusionMatrixDisplay
_ = ConfusionMatrixDisplay.from_estimator(classifier, data_test,
target_test)
```



6.2.10.3 Παράδειγμα για το Precision-Recall

```
from sklearn.metrics import precision_score, recall_score
precision = precision_score(target_test, target_predicted,
pos_label="donated")
recall = recall_score(target_test, target_predicted,
pos_label="donated")
print(f"Precision score: {precision:.3f}")
print(f"Recall score: {recall:.3f}")

# Precision score: 0.688
# Recall score: 0.124
```

Αυτά τα αποτελέσματα συμφωνούν με αυτό που φάνηκε στον Confusion matrix. Κοιτάζοντας στην αριστερή στήλη, περισσότερες από τις μισές προβλέψεις "donated" ήταν σωστές, οδηγώντας σε Precision πάνω από 0,5. Ωστόσο, ο ταξινομητής μας σημείωσε εσφαλμένα πολλά άτομα που έδωσαν αίμα ως "not donated", οδηγώντας σε πολύ χαμηλή μετρική Recall γύρω στο 0,1.

6.2.10.4 Παράδειγμα για το F1-Score

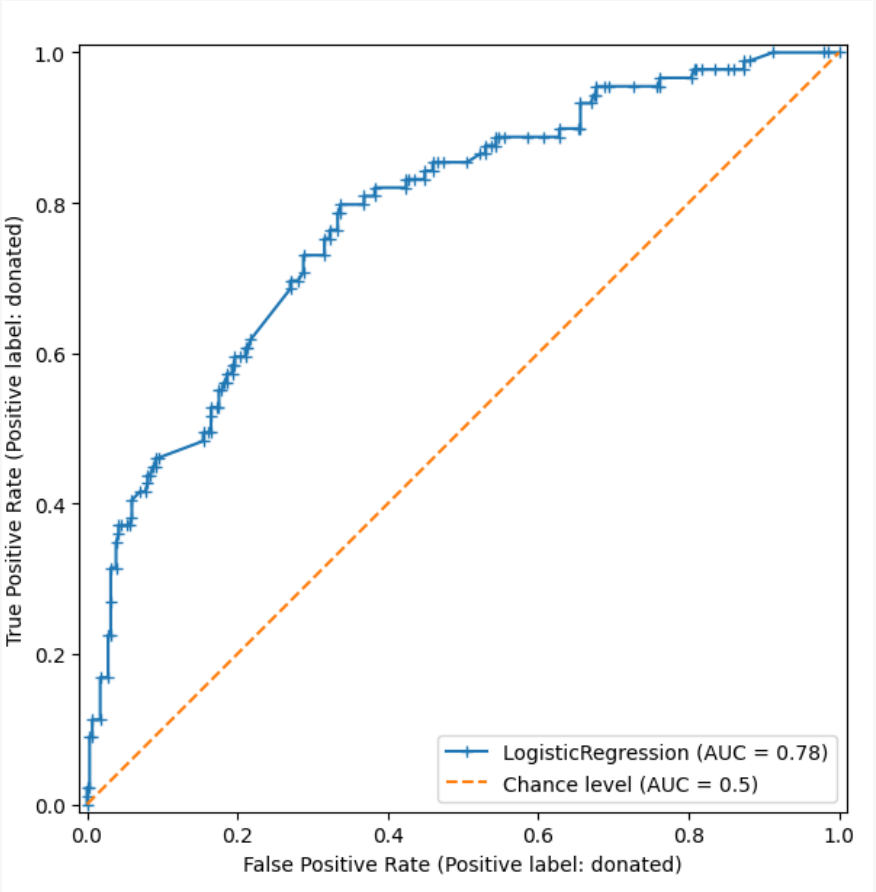
```
from sklearn.metrics import f1_score
f1_score(target_test, target_predicted, pos_label = 'donated')

# 0.20952380952380953
```

Όπως αναμέναμε, η τιμή του F1-Score (0,2095) είναι πιο κοντά στο χειρότερο από τα Precision (0,688) – Recall (0.124).

6.2.10.5 Παράδειγμα για τα ROC-AUC

Η συνάρτηση RocCurveDisplay.from_estimator μας εμφανίζει την καμπύλη ROC αλλά και την περιοχή κάτω από την καμπύλη (μετρική AUC) για το μοντέλο και τα δεδομένα που της θέτουμε. Επιλέγοντας την παράμετρο plot_chance_level=True, βλέπουμε πόσο διαφέρει το μοντέλο από τη θεωρητική περίπτωση της τυχαίας διαλογής (AUC = 0,5).

```
from sklearn.metrics import RocCurveDisplay
disp = RocCurveDisplay.from_estimator(
      classifier,
      data_test,
      target_test,
      pos_label="donated",
      marker="+",
      plot_chance_level=True,
      chance_level_kw={"color": "tab:orange", "linestyle": "--"}
)
```



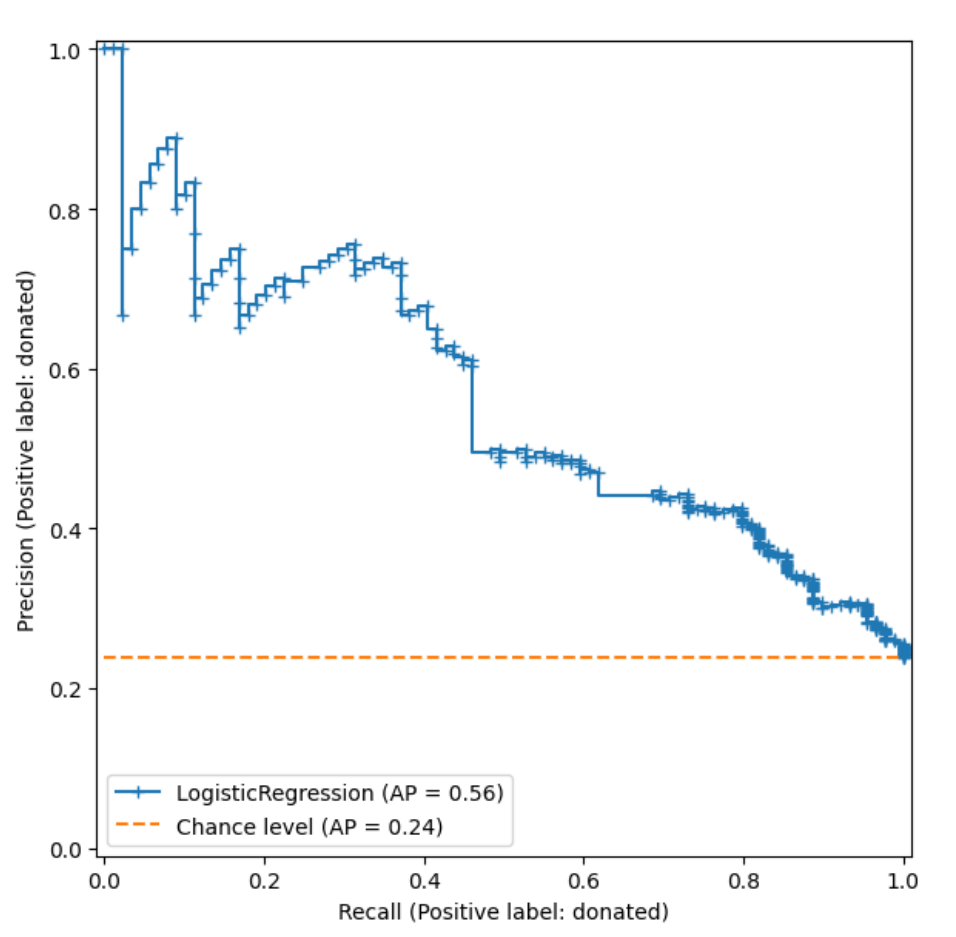
Εάν μας ενδιαφέρει μόνο η περιοχή AUC και όχι οι καμπύλες ROC, τότε μπορούμε να χρησιμοποιήσουμε την συνάρτηση roc_auc_score με τον ακόλουθο τρόπο:

```
from sklearn.metrics import roc_auc_score
roc_auc_score(target_test, classifier.decision_function(data_test))

# 0.7839345554898483
```

6.2.10.6 Παράδειγμα για το Precision-Recall Curve

Η συνάρτηση PrecisionRecallDisplay.from_estimator μας εμφανίζει την καμπύλη Precision-Recall αλλά και την περιοχή κάτω από την καμπύλη (αλλιώς και PR AUC) για το μοντέλο και τα δεδομένα που της θέτουμε. Επιλέγοντας την παράμετρο plot_chance_level=True, βλέπουμε πόσο διαφέρει το μοντέλο από τη θεωρητική περίπτωση της επιλογής της πιο συνηθισμένης κλάσης ανεξάρτητα από τις εισόδους.

```
from sklearn.metrics import PrecisionRecallDisplay

PrecisionRecallDisplay.from_estimator(
      classifier,
      data_test,
      target_test,
      pos_label="donated",
      marker="+",
      plot_chance_level=True,
      chance_level_kw={"color": "tab:orange", "linestyle": "--"},
)
```



6.3. Μετρικές Αξιολόγησης σε Προβλήματα Παλινδρόμησης

Στα προβλήματα παλινδρόμησης, ο στόχος είναι η πρόβλεψη μιας συνεχούς μεταβλητής από ένα σύνολο χαρακτηριστικών. Για να αξιολογήσουμε την απόδοση ενός μοντέλου παλινδρόμησης, χρησιμοποιούμε διάφορες μετρικές που μας επιτρέπουν να εκτιμήσουμε την ποιότητα των προβλέψεων σε σχέση με τις πραγματικές τιμές. Σε αυτό το κεφάλαιο θα δούμε τις πιο σημαντικές

από αυτές τις μετρικές, όπως το Μέσο Τετραγωνικό Σφάλμα (MSE), το R^2, το Μέσο Απόλυτο Σφάλμα (MAE), το Μέσο Απόλυτο Ποσοστιαίο Σφάλμα (MAPE) και τα υπολείμματα (residuals).

6.3.1. Μέσο Τετραγωνικό Σφάλμα (Mean Squared Error - MSE)

Το Μέσο Τετραγωνικό Σφάλμα (MSE) είναι μια από τις πιο διαδεδομένες μετρικές σε προβλήματα παλινδρόμησης. Υπολογίζει τη μέση τιμή των τετραγώνων των διαφορών μεταξύ των πραγματικών τιμών και των προβλέψεων. Το MSE τιμωρεί τα μεγαλύτερα λάθη, καθώς το σφάλμα αυξάνεται εκθετικά λόγω της τετραγωνικής φύσης του. Ο τύπος του είναι:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

όπου y_i είναι οι πραγματικές τιμές και $\hat{y}_i$ οι προβλεπόμενες τιμές.

Το MSE προσφέρει μια συνολική εικόνα της απόδοσης του μοντέλου, με τις τιμές του να είναι πάντα μη αρνητικές. Όσο μικρότερη είναι η τιμή του MSE, τόσο καλύτερη είναι η απόδοση του μοντέλου. Ωστόσο, επειδή εκφράζεται σε τετραγωνικές μονάδες, μπορεί να μην είναι εύκολα ερμηνεύσιμο από μόνο του.

6.3.2. R^2 (Συντελεστής Προσδιορισμού)

Το R^2, γνωστό και ως συντελεστής προσδιορισμού, μετρά την αναλογία της διακύμανσης που εξηγείται από το μοντέλο σε σχέση με τη συνολική διακύμανση των δεδομένων. Ουσιαστικά, το R^2 δείχνει πόσο καλά το μοντέλο προβλέπει τα δεδομένα σε σχέση με έναν απλό μέσο όρο. Ο τύπος του είναι:

$$R^2 = 1 - \frac{\mathrm{SS}_{\mathrm{res}}}{\mathrm{SS}_{\mathrm{tot}}}$$

όπου $\mathrm{SS}_{\mathrm{res}}$ είναι το Άθροισμα των Τετραγώνων των Υπολειμμάτων:

$$\mathrm{SS}_{\mathrm{res}} = \sum_{i=1}(y_i - \hat{y}_i)^2$$

και $\mathrm{SS}_{\mathrm{tot}}$ το συνολικό άθροισμα των τετραγώνων της διαφοράς μεταξύ των πραγματικών τιμών και της μέσης τιμής τους:

$$\mathrm{SS}_{\mathrm{tot}} = \sum_{i=1}(y_i - \overline{y})^2$$

Το R^2 παίρνει τιμές από 0 έως 1, όπου το 1 σημαίνει ότι το μοντέλο εξηγεί τέλεια τη διακύμανση των δεδομένων. Ωστόσο, μπορεί επίσης να πάρει αρνητικές τιμές, όταν το μοντέλο προβλέπει χειρότερα από ένα μοντέλο που απλώς χρησιμοποιεί τον μέσο όρο.

6.3.3. Μέσο Απόλυτο Σφάλμα (Mean Absolute Error - MAE)

Το Μέσο Απόλυτο Σφάλμα (MAE) είναι μια άλλη συνηθισμένη μετρική, που μετρά την απόλυτη διαφορά μεταξύ των πραγματικών τιμών και των προβλέψεων. Σε αντίθεση με το MSE, το MAE δεν τετραγωνίζει τις διαφορές, οπότε δεν τιμωρεί τα μεγαλύτερα λάθη τόσο έντονα. Αυτό το καθιστά πιο εύκολο να ερμηνευτεί. Ο τύπος του είναι:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Το MAE εκφράζεται στις ίδιες μονάδες με τις τιμές που προβλέπουμε, και έτσι είναι πιο άμεσο για να κατανοήσουμε πόσο απέχει κατά μέσο όρο η πρόβλεψή μας από τις πραγματικές τιμές.

6.3.4. Διάμεσο Απόλυτο Σφάλμα (Median Absolute Error - MedAE)

Το Median Absolute Error (MedAE) είναι μια μετρική παλινδρόμησης που υπολογίζει τη διάμεσο των απόλυτων σφαλμάτων μεταξύ των πραγματικών τιμών και των προβλέψεων του μοντέλου. Είναι μια πιο ανθεκτική μετρική σε ακραίες τιμές (outliers) σε σύγκριση με το Μέσο Απόλυτο Σφάλμα (MAE), επειδή βασίζεται στη διάμεσο, η οποία επηρεάζεται λιγότερο από εξαιρετικά υψηλά ή χαμηλά σφάλματα. Για να υπολογίσουμε το MedAE, παίρνουμε το απόλυτο σφάλμα όλων των προβλέψεων και μετά βρίσκουμε τη διάμεσο:

$$MedAE = median(|y_1 - \hat{y}_1|, |y_2 - \hat{y}_2|, \ldots, |y_n - \hat{y}_n|)$$

6.3.5. Μέσο Απόλυτο Ποσοστιαίο Σφάλμα (Mean Absolute Percentage Error - MAPE)

Το MAPE υπολογίζει το ποσοστό του απόλυτου σφάλματος σε σχέση με την πραγματική τιμή. Αυτό είναι ιδιαίτερα χρήσιμο όταν θέλουμε να κατανοήσουμε την ακρίβεια του μοντέλου σε σχετικούς όρους, δηλαδή ως ποσοστό. Ο τύπος του είναι:

$$\text{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

Το MAPE εκφράζεται σε ποσοστό και είναι εύκολο να κατανοηθεί από μια ευρεία γκάμα χρηστών. Ωστόσο, το MAPE μπορεί να επηρεαστεί πολύ από μικρές πραγματικές τιμές, καθώς τα ποσοστά που προκύπτουν μπορεί να είναι πολύ μεγάλα.

6.3.6. Υπολείμματα (Residuals)

Τα υπολείμματα (residuals) είναι οι διαφορές μεταξύ των πραγματικών τιμών και των προβλέψεων του μοντέλου. Ουσιαστικά, το υπόλειμμα για κάθε παρατήρηση i ορίζεται ως:

$$\text{Υπόλειμμα}_i = y_i - \hat{y}_i$$

Η ανάλυση των υπολειμμάτων είναι κρίσιμη για την αξιολόγηση της ποιότητας του μοντέλου. Ένα καλό μοντέλο θα πρέπει να έχει υπολείμματα που κατανέμονται τυχαία, χωρίς συστηματικά μοτίβα. Εάν τα υπολείμματα παρουσιάζουν κάποιο πρότυπο (π.χ., αυξάνονται ή μειώνονται συστηματικά), αυτό μπορεί να είναι ένδειξη ότι το μοντέλο δεν ταιριάζει καλά στα δεδομένα.

6.3.7. Συμπεράσματα

Η επιλογή της κατάλληλης μετρικής για την αξιολόγηση της απόδοσης ενός μοντέλου παλινδρόμησης εξαρτάται από το είδος των δεδομένων και τις απαιτήσεις της εκάστοτε εφαρμογής. Το MSE και το R^2 παρέχουν πληροφορίες για την ακρίβεια του μοντέλου σε σχέση με τη συνολική διακύμανση, ενώ το MAE και το MAPE είναι πιο εύκολα κατανοητά και παρέχουν πληροφορίες για το απόλυτο μέγεθος των λαθών. Η ανάλυση των υπολειμμάτων είναι επίσης σημαντική, καθώς αποκαλύπτει μοτίβα που μπορεί να υποδεικνύουν ανεπάρκειες του μοντέλου.

6.3.8. Παραδείγματα στο scikit-learn

Οι μετρικές παλινδρόμησης βρίσκονται και αυτές στο πακέτο sklearn.metrics του scikit-learn. Για τα επόμενα παραδείγματα θα χρησιμοποιήσουμε το σύνολο δεδομένων κατοικίας Ames. Στόχος είναι να προβλεφθεί η τιμή των σπιτιών στην πόλη Έιμς (Αϊόβα). Όπως και με την ταξινόμηση, χρησιμοποιούμε μόνο ένα διαχωρισμό εκπαίδευσης-δοκιμής για να εστιάσουμε αποκλειστικά στις μετρικές παλινδρόμησης.

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression

ames_housing = pd.read_csv("house_prices.csv")
data = ames_housing.drop(columns="SalePrice")
target = ames_housing["SalePrice"]
data = data.select_dtypes(np.number)
```

```
# Μετασχηματίζουμε το στόχο σε χιλιάδες δολλαρίων (k$).
target /= 1000

# Διαχωρίζουμε τα δεδομένα μας σε σύνολα εκπαίδευσης και δοκιμής.
from sklearn.model_selection import train_test_split
data_train, data_test, target_train, target_test = train_test_split(
data, target, shuffle=True, random_state=0)

# Εκπαιδεύουμε ένα μοντέλο γραμμικής παλινδρόμησης στα δεδομένα
εκπαίδευσης
regressor = LinearRegression()
regressor.fit(data_train, target_train)
```

6.3.8.1 Παράδειγμα για το Μέσο Τετραγωνικό Σφάλμα (Mean Squared Error - MSE)

Ορισμένα μοντέλα μηχανικής εκμάθησης έχουν σχεδιαστεί για να επιλυθούν ως προβλήματα βελτιστοποίησης, δηλαδή ελαχιστοποιώντας ένα σφάλμα (γνωστό και ως συνάρτηση απώλειας-loss function) με το σετ εκπαίδευσης. Μια βασική συνάρτηση απώλειας που χρησιμοποιείται στην παλινδρόμηση είναι το μέσο τετραγωνικό σφάλμα (Mean Squared Error-MSE). Έτσι, αυτή η μετρική χρησιμοποιείται μερικές φορές και για την αξιολόγηση του μοντέλου, διότι βελτιστοποιείται από το εν λόγω μοντέλο.

Το μοντέλο μας γραμμικής παλινδρόμησης ελαχιστοποιεί το μέσο τετραγωνικό σφάλμα στο σετ εκπαίδευσης. Στη συνέχεια, μπορούμε να υπολογίσουμε το μέσο τετράγωνο σφάλμα στο σύνολο δοκιμής.

```
from sklearn.metrics import mean_squared_error

target_predicted = regressor.predict(data_test)
print(
      "Mean squared error on the testing set: "
      f"{mean_squared_error(target_test, target_predicted):.3f}"
)

# Mean squared error on the testing set: 2064.736
```

6.3.8.2 Παράδειγμα για το R^2 (Συντελεστής Προσδιορισμού)

Το ακατέργαστο MSE μπορεί να είναι δύσκολο να ερμηνευτεί. Ένας τρόπος είναι να αναπροσαρμόσετε το MSE χρησιμοποιώντας τη διακύμανση του στόχου. Αυτή η μετρική είναι γνωστή R^2 (Συντελεστής Προσδιορισμού). Πράγματι, αυτή είναι η προεπιλεγμένη μετρική που χρησιμοποιείται στο scikit-learn καλώντας τη μέθοδο score.

```
regressor.score(data_test, target_test)
# 0.6872520581075561
```

Η μετρική αυτή αντιπροσωπεύει το ποσοστό διακύμανσης του στόχου που εξηγείται από τις ανεξάρτητες μεταβλητές του μοντέλου. Η καλύτερη δυνατή βαθμολογία είναι 1 αλλά δεν υπάρχει κάτω όριο.

6.3.8.3 Παράδειγμα για το Μέσο Απόλυτο Σφάλμα (Mean Absolute Error - MAE)

Η R^2 δίνει μια εικόνα για την ποιότητα του εκπαιδευμένου μοντέλου. Ωστόσο, αυτή η βαθμολογία δεν μπορεί να συγκριθεί από το ένα σύνολο δεδομένων στο άλλο και η τιμή που λαμβάνεται δεν έχει ουσιαστική ερμηνεία σε σχέση με τον αρχικό στόχο. Αν θέλαμε να πάρουμε μια ερμηνεύσιμη βαθμολογία, θα ενδιαφερόμασταν για το διάμεσο ή το μέσο απόλυτο σφάλμα (Mean absolute error- MAE).

```
from sklearn.metrics import mean_absolute_error
target_predicted = regressor.predict(data_test)
print(
      "Mean absolute error: "
      f"{mean_absolute_error(target_test, target_predicted):.3f} k$"
)

# Mean absolute error: 22.608 k$
```

Υπολογίζοντας το μέσο απόλυτο σφάλμα (MAE), μπορούμε να ερμηνεύσουμε ότι το μοντέλο μας προβλέπει κατά μέσο όρο 22,6 k$ μακριά από την πραγματική τιμή του σπιτιού.

6.3.8.4 Παράδειγμα για το Διάμεσο Απόλυτο Σφάλμα (Median Absolute Error - MedAE)

Ενα μειονέκτημα του MAE είναι ότι ο μέσος όρος μπορεί να επηρεαστεί από μεγάλο σφάλμα. Για ορισμένες εφαρμογές, μπορεί να μην θέλουμε αυτά τα μεγάλα σφάλματα να έχουν τόσο μεγάλη επιρροή στη μετρική μας. Σε αυτή την περίπτωση μπορούμε να χρησιμοποιήσουμε το διάμεσο απόλυτο σφάλμα (Median absolute error-MedAE).

```
from sklearn.metrics import median_absolute_error
print(
      "Median absolute error: "
      f"{median_absolute_error(target_test, target_predicted):.3f} k$"
)
# Median absolute error: 14.137 k$
```

6.3.8.5 *Παράδειγμα για το Μέσο Απόλυτο Ποσοστιαίο Σφάλμα (Mean Absolute Percentage Error - MAPE)*

Το μέσο απόλυτο σφάλμα (ή το διάμεσο απόλυτο σφάλμα) εξακολουθεί να έχει έναν περιορισμό: έχοντας σφάλμα 50 k$ για ένα σπίτι αξίας 50 k$ έχει το ίδιο αντίκτυπο με ένα σφάλμα 50 k$ για ένα σπίτι αξίας 500 k$. Πράγματι, το μέσο απόλυτο σφάλμα δεν είναι σχετικό. Το μέσο απόλυτο ποσοστό σφάλματος (mean absolute percentage error) εισάγει αυτήν τη σχετική κλιμάκωση.

```
from sklearn.metrics import mean_absolute_percentage_error
print(
      "Mean absolute percentage error: "
      f"{mean_absolute_percentage_error(target_test, target_predicted)
* 100:.3f} %"
)
# Mean absolute percentage error: 13.574 %
```

6.3.8.6 *Παράδειγμα για τα Υπολείμματα (Residuals)*

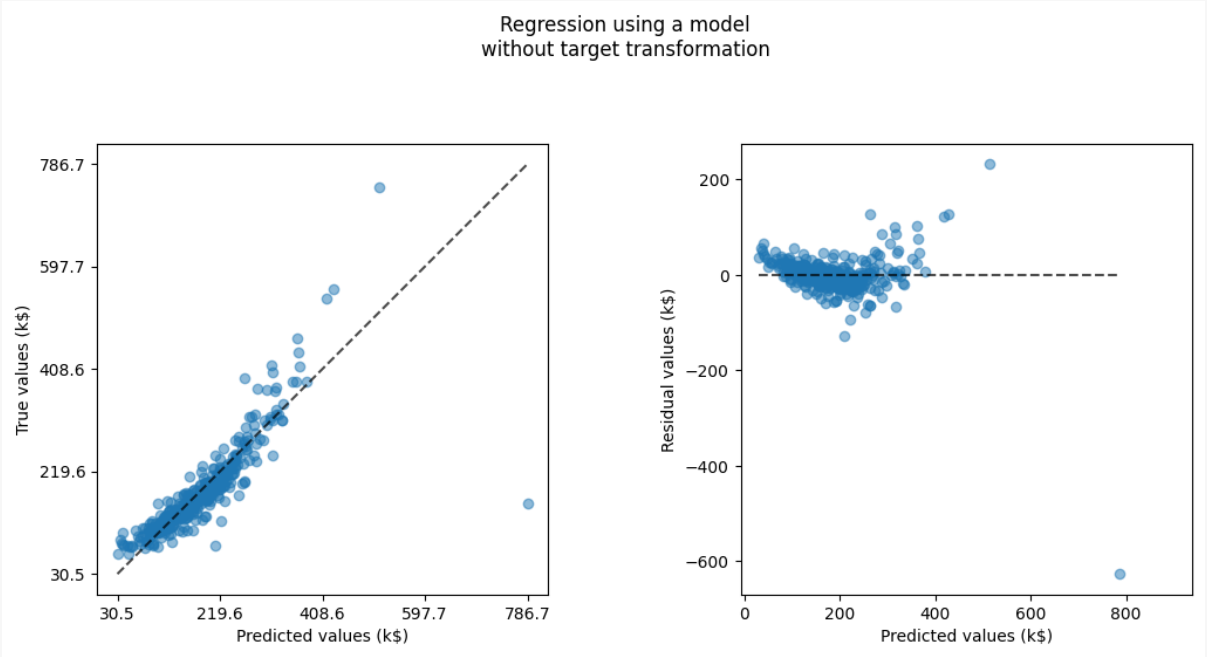
Εκτός από τη χρήση μετρικών, μπορούμε να οπτικοποιήσουμε τα αποτελέσματα σχεδιάζοντας τις προβλεπόμενες τιμές έναντι των πραγματικών τιμών. Σε ένα ιδανικό σενάριο όπου όλες οι παραλλαγές στον στόχο θα μπορούσαν να εξηγηθούν τέλεια από τα παρατηρούμενα χαρακτηριστικά και έχοντας επιλέξει ένα βέλτιστο μοντέλο θα περιμέναμε όλες οι προβλέψεις να πέσουν κατά μήκος της διαγώνιας γραμμής του πρώτου διαγράμματος παρακάτω.

Σε πραγματικές συνθήκες, αυτό δεν συμβαίνει σχεδόν ποτέ. Κάποια άγνωστη διακύμανση του στόχου δεν μπορεί να εξηγηθεί από τις διακυμάνσεις στα δεδομένα: προέρχεται από εξωτερικούς παράγοντες που δεν αντιπροσωπεύονται στα παρατηρούμενα χαρακτηριστικά. Για να αποκτήσετε περισσότερες πληροφορίες, μπορεί να είναι χρήσιμο να σχεδιάσετε τα υπολείμματα (residuals), τα οποία αντιπροσωπεύουν τη διαφορά μεταξύ των πραγματικών και των προβλεπόμενων τιμών έναντι των προβλεπόμενων τιμών. Αυτό φαίνεται στο δεύτερο διάγραμμα.

Στο scikit-learn μπορούμε να χρησιμοποιήσουμε τη PredictionErrorDisplay του πακέτου sklearn.metrics για να οπτικοποιήσουμε τα υπολείμματα:

```
import matplotlib.pyplot as plt
from sklearn.metrics import PredictionErrorDisplay
fig, axs = plt.subplots(ncols=2, figsize=(13, 5))
PredictionErrorDisplay.from_predictions(
      y_true=target_test,
      y_pred=target_predicted,
      kind="actual_vs_predicted",
      scatter_kwargs={"alpha": 0.5},
      ax=axs[0],
```

```
)
axs[0].axis("square")
axs[0].set_xlabel("Predicted values (k$)")
axs[0].set_ylabel("True values (k$)")
PredictionErrorDisplay.from_predictions(
      y_true=target_test,
      y_pred=target_predicted,
      kind="residual_vs_predicted",
      scatter_kwargs={"alpha": 0.5},
      ax=axs[1],
)
axs[1].axis("square")
axs[1].set_xlabel("Predicted values (k$)")
axs[1].set_ylabel("Residual values (k$)")
_ = fig.suptitle(
      "Regression using a model\nwithout target transformation", y=1.1
)
```



Σε αυτά τα διαγράμματα βλέπουμε ότι το μοντέλο μας τείνει να υποτιμά την τιμή των σπιτιών τόσο για τις χαμηλότερες όσο και για τις υψηλές πραγματικές τιμές. Αυτό σημαίνει ότι τα υπολείμματα εξακολουθούν να διατηρούν κάποια δομή που είναι συνήθως ορατή ως σχήμα "μπανάνα" ή "χαμόγελο" στο διάγραμμα των υπολειμμάτων. Αυτό είναι συχνά μια ένδειξη ότι το μοντέλο μας θα μπορούσε να βελτιωθεί, είτε μετασχηματίζοντας τα χαρακτηριστικά, τον στόχο ή μερικές φορές αλλάζοντας τον τύπο του μοντέλου ή τις παραμέτρους του.

6.4. Διασταυρούμενη Επικύρωση

Στη μηχανική μάθηση, η αξιολόγηση ενός μοντέλου είναι κρίσιμη για την κατανόηση της ικανότητάς του να γενικεύει, δηλαδή να κάνει ακριβείς προβλέψεις σε νέα, άγνωστα δεδομένα. Μια κοινή πρακτική είναι η διάσπαση του συνόλου δεδομένων σε σύνολα εκπαίδευσης και δοκιμής, όμως αυτή η μέθοδος μπορεί να οδηγήσει σε παραπλανητικές εκτιμήσεις απόδοσης, ιδιαίτερα όταν το μέγεθος του συνόλου δεδομένων είναι μικρό ή όταν η κατανομή των δεδομένων ποικίλλει.

Η **διασταυρούμενη επικύρωση** (cross-validation) έρχεται να αντιμετωπίσει αυτό το πρόβλημα, παρέχοντας μια πιο αξιόπιστη μέτρηση της απόδοσης του μοντέλου μέσω πολλαπλών διαιρέσεων και εκπαιδεύσεων. Αυτή η τεχνική βοηθά να αποφεύγονται υπερεκτιμήσεις ή υποεκτιμήσεις της ακρίβειας του μοντέλου και να βελτιώνεται η γενική του απόδοση.

6.4.1. Η Ανάγκη για Διασταυρούμενη Επικύρωση

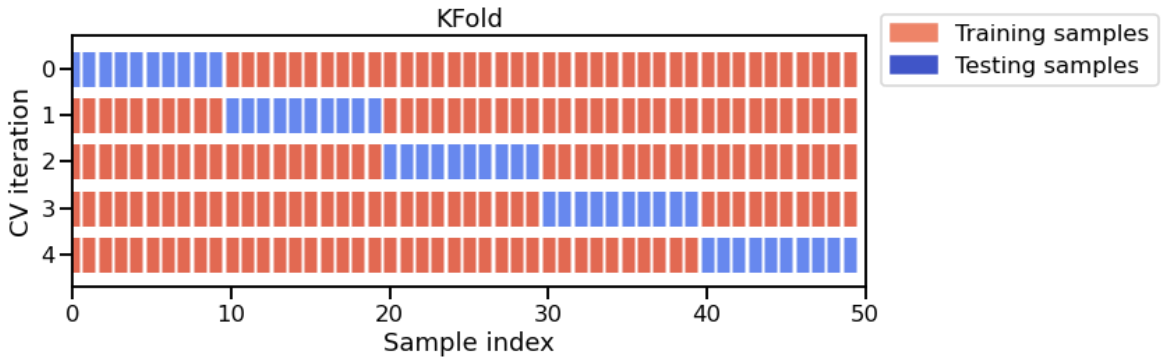
Στην απλή προσέγγιση του διαχωρισμού των δεδομένων σε εκπαίδευση και δοκιμή, μπορεί να προκύψουν προβλήματα. Για παράδειγμα, η απόδοση ενός μοντέλου μπορεί να φανεί υπερβολικά αισιόδοξη αν, κατά τύχη, το σύνολο δοκιμής περιέχει τα πιο "εύκολα" παραδείγματα. Αυτό έχει ως αποτέλεσμα λανθασμένες εκτιμήσεις της πραγματικής του ικανότητας. Επιπλέον, η μείωση του αριθμού των δειγμάτων για εκπαίδευση και δοκιμή μπορεί να προκαλέσει προβλήματα ειδικά σε μικρά σύνολα δεδομένων.

Η διασταυρούμενη επικύρωση επιλύει αυτά τα ζητήματα κάνοντας πολλαπλές διαιρέσεις των δεδομένων και χρησιμοποιώντας κάθε υποσύνολο τόσο για εκπαίδευση όσο και για δοκιμή. Έτσι, επιτυγχάνεται μια πιο αξιόπιστη εκτίμηση της απόδοσης του μοντέλου, καθώς μειώνεται η πιθανότητα το αποτέλεσμα να οφείλεται σε τυχαίες διακυμάνσεις. Στον αντίποδα, η διαδικασία cross-validation είναι υπολογιστικά δαπανηρή επειδή απαιτεί εκπαίδευση πολλών μοντέλων αντί για ένα.

6.4.2. Στρατηγικές Διασταυρούμενης Επικύρωσης

6.4.2.1 K-fold Cross-Validation

Η πιο κοινή τεχνική διασταυρούμενης επικύρωσης είναι η **K-fold cross-validation**, όπου τα δεδομένα χωρίζονται σε K ίσα μέρη (folds). Το μοντέλο εκπαιδεύεται K φορές, κάθε φορά χρησιμοποιώντας τα K-1 τμήματα για εκπαίδευση και το υπόλοιπο για δοκιμή. Ο μέσος όρος της απόδοσης σε όλες τις K διαιρέσεις χρησιμοποιείται ως εκτίμηση της γενικής απόδοσης.



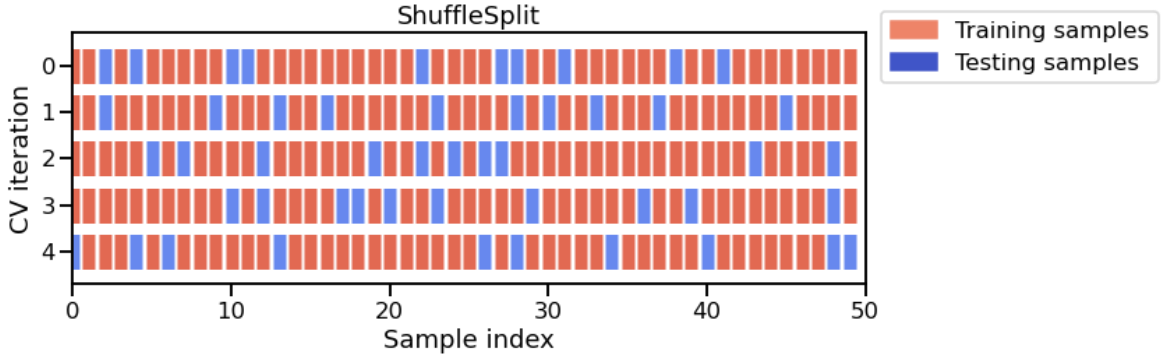
Διάγραμμα 6.5: 5-fold cross-validation. Η εκπαίδευση του μοντέλου γίνεται στα κόκκινα δείγματα και η αξιολόγηση στα μπλε δείγματα.

6.4.2.2 Shufflesplit

Μια άλλη στρατηγική cross-validation ονομάζεται "shuffle-split". Σε κάθε επανάληψη αυτής της στρατηγικής:

- ανακατεύουμε τυχαία τη σειρά των δειγμάτων ενός αντιγράφου του πλήρους συνόλου δεδομένων.
- χωρίζουμε το ανακατεμένο σύνολο δεδομένων σε εκπαίδευσης και δοκιμών.
- Εκπαιδεύουμε ένα νέο μοντέλο στο σετ εκπαίδευσης.
- Αξιολογούμε το μοντέλο στο σετ δοκιμής.

Επαναλαμβάνουμε αυτήν τη διαδικασία n φορές. Λάβετε υπόψη ότι το υπολογιστικό κόστος αυξάνεται με n.



Διάγραμμα 6.6: Shufflesplit cross-validation για n=5. Η εκπαίδευση του μοντέλου γίνεται στα κόκκινα δείγματα και η αξιολόγηση στα μπλε δείγματα.

6.4.2.3 *Stratified K-fold Cross-Validation*

Στην περίπτωση προβλημάτων ταξινόμησης χρησιμοποιείται συχνά η **διαστρωματωμένη διασταυρούμενη επικύρωση** (stratified cross-validation). Εδώ, κάθε fold διατηρεί την ίδια αναλογία κατηγοριών όπως στο αρχικό σύνολο δεδομένων. Αυτή η τεχνική είναι χρήσιμη όταν οι κλάσεις είναι ανισοβαρείς (π.χ. όταν η μία κατηγορία έχει πολύ περισσότερα δείγματα από τις άλλες).

6.4.2.4 *Leave-One-Out Cross-Validation (LOOCV)*

Στην **Leave-One-Out cross-validation**, κάθε δείγμα των δεδομένων χρησιμοποιείται μεμονωμένα ως σύνολο δοκιμής, ενώ όλα τα υπόλοιπα χρησιμοποιούνται για εκπαίδευση. Αυτή η μέθοδος είναι πιο ακριβής, αλλά πολύ απαιτητική υπολογιστικά, ιδιαίτερα για μεγάλα σύνολα δεδομένων.

6.4.3. Παραδείγματα Διασταυρούμενης Επικύρωσης στο scikit-learn

Η βιβλιοθήκη **scikit-learn** παρέχει πολλές δυνατότητες για την υλοποίηση της διασταυρούμενης επικύρωσης. Η συνάρτηση **cross_validate** είναι μια από τις επιλογές που υπάρχουν για cross-validation στο scikit-learn. Χρειάζεται να περάσετε το μοντέλο, τα δεδομένα και τον στόχο. Από εκεί και πέρα υπάρχουν πολλές στρατηγικές, το cross_validate παίρνει μια παράμετρο cv που καθορίζει τη στρατηγική διαχωρισμού.

6.4.3.1 *Παράδειγμα k-fold cross validation*

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import cross_validate


# Αρχικά, ας φορτώσουμε το πλήρες σύνολο δεδομένων adult census.
adult_census = pd.read_csv("adult-census.csv")
target_name = "class"
target = adult_census[target_name]
data = adult_census.drop(columns=target_name)

# Επίσης, επιλέγουμε μόνο τις αριθμητικές στήλες και δημιουργούμε το
εκπαιδευτικό pipeline.
numerical_columns = ["age", "capital-gain", "capital-loss", "hours-per-
week"]
data_numeric = data[numerical_columns]
model = make_pipeline(StandardScaler(), LogisticRegression())

# Εφαρμογή 5-fold cross-validation
model = make_pipeline(StandardScaler(), LogisticRegression())
cv_result = cross_validate(model, data_numeric, target, cv=5)
cv_result
```

```
# Αποτέλεσμα:
# {'fit_time': array([0.04736733, 0.04696965, 0.04235411, 0.03123236,
0.04744959]),
# 'score_time': array([0.        , 0.02077174, 0.0157063 , 0.01563454,
0.        ]),
# 'test_score': array([0.79557785, 0.80049135, 0.79965192, 0.79873055,
0.80456593])}
```

Η έξοδος του `cross_validate` είναι ένα λεξικό Python, το οποίο από προεπιλογή περιέχει τρεις καταχωρίσεις:

- τον χρόνο εκπαίδευσης του μοντέλου στα δεδομένα εκπαίδευσης για κάθε μέρος (fold), `fit_time`
- τον χρόνο πρόβλεψης με το μοντέλο στα δεδομένα δοκιμών για κάθε μέρος,`score_time`
- την προεπιλεγμένη βαθμολογία στα δεδομένα δοκιμών για κάθε μέρος, `test_score`.

Η ρύθμιση `cv=5` δημιούργησε 5 διακριτές διαιρέσεις για να λάβουμε 5 παραλλαγές για την εκπαίδευση και τη δοκιμή. Κάθε σετ εκπαίδευσης χρησιμοποιείται για να εκπαιδεύσει ένα μοντέλο και μετά το μοντέλο αυτό βαθμολογήθηκε στο αντίστοιχο σετ δοκιμών. Η προεπιλεγμένη στρατηγική κατά τον ορισμό `cv=<ακέραιος αριθμός>` είναι η K-fold cross-validation όπου το `K` αντιστοιχεί στον (ακέραιο) αριθμό των διαχωρισμών. Μια ρύθμιση `cv=5` ή `cv=10` είναι μια συνηθισμένη πρακτική, καθώς είναι ένας καλός συμβιβασμός μεταξύ του χρόνου υπολογισμού και της σταθερότητας της εκτιμώμενης μεταβλητότητας.

Σημειώστε ότι από προεπιλογή η συνάρτηση `**cross_validate**` απορρίπτει τα μοντέλα που εκπαιδεύτηκαν στο διαφορετικό επικαλυπτόμενο υποσύνολο του συνόλου δεδομένων. Ο στόχος της διασταυρούμενης επικύρωσης δεν είναι η εκπαίδευση ενός μοντέλου, αλλά μάλλον η εκτίμηση περίπου της απόδοσης γενίκευσης ενός μοντέλου εάν θα είχε εκπαιδευτεί στο πλήρες σετ εκπαίδευσης, μαζί με μια εκτίμηση της μεταβλητότητας.

Μπορείτε να ορίσετε πρόσθετες παραμέτρους στην **cross-validate** για τη συλλογή πρόσθετων πληροφοριών, όπως οι βαθμολογίες εκπαίδευσης των μοντέλων που λαμβάνονται σε κάθε γύρο ή ακόμη και την επιστροφή των ίδιων μοντέλων αντί για την απόρριψή τους.

```
# Έχοντας τα αποτελέσματα του cross-validation για όλα τα μέρη
# υπολογίζουμε το μέσο όρο και την τυπική απόκλιση:

scores = cv_result["test_score"]
```

```
print(
    "The mean cross-validation accuracy is: "
    f"{scores.mean():.3f} ± {scores.std():.3f}"
)

# The mean cross-validation accuracy is: 0.800 ± 0.003
```

Σημειώστε ότι υπολογίζοντας την τυπική απόκλιση των βαθμολογιών διασταυρούμενης επικύρωσης, μπορούμε να εκτιμήσουμε την αβεβαιότητα της απόδοσης γενίκευσης του μοντέλου μας. Αυτό είναι το κύριο πλεονέκτημα της διασταυρούμενης επικύρωσης και μπορεί να είναι κρίσιμο στην πράξη, για παράδειγμα όταν συγκρίνετε διαφορετικά μοντέλα για να καταλάβετε αν κάποιο είναι καλύτερο από το άλλο ή αν η απόδοση γενίκευσης του κάθε μοντέλου βρίσκεται μέσα στις γραμμές σφαλμάτων ενός άλλου μοντέλου.

6.4.3.2 Παράδειγμα shuffle-split cross-validation

Στο scikit-learn μπορούμε να χρησιμοποιήσουμε τη συνάρτηση `cross_validate` με ένα "shuffle-split" αντικείμενο:

```
# Χρησιμοποιούμε το προηγούμενο σύνολο δεδομένων και μοντέλο

from sklearn.model_selection import ShuffleSplit

cv = ShuffleSplit(n_splits=5, test_size=0.3, random_state=0)
cv_result = cross_validate(model, data_numeric, target, cv=cv)
cv_result
# Result:
# {'fit_time': array([0.04723573, 0.04687357, 0.04688239, 0.06301165,
0.05387878]),
# 'score_time': array([0.03126001, 0.0160141 , 0.01562452, 0.01562691,
0.0090065 ]),
# 'test_score': array([0.80004095, 0.79935849, 0.80140586, 0.79908551,
0.79703815])}
```

6.4.3.3 Παράδειγμα stratified cross-validation

Ας ξεκινήσουμε με την έννοια της διαστρωμάτωσης δίνοντας μια περίπτωση όπου μπορεί να αντιμετωπίσουμε πρόβλημα αν δεν είμαστε προσεκτικοί. Ας φορτώσουμε το σύνολο δεδομένων iris και ας δημιουργήσουμε ένα μοντέλο λογιστικής παλινδρόμησης.

```
from sklearn.datasets import load_iris
data, target = load_iris(as_frame=True, return_X_y=True)
from sklearn.preprocessing import StandardScaler
```

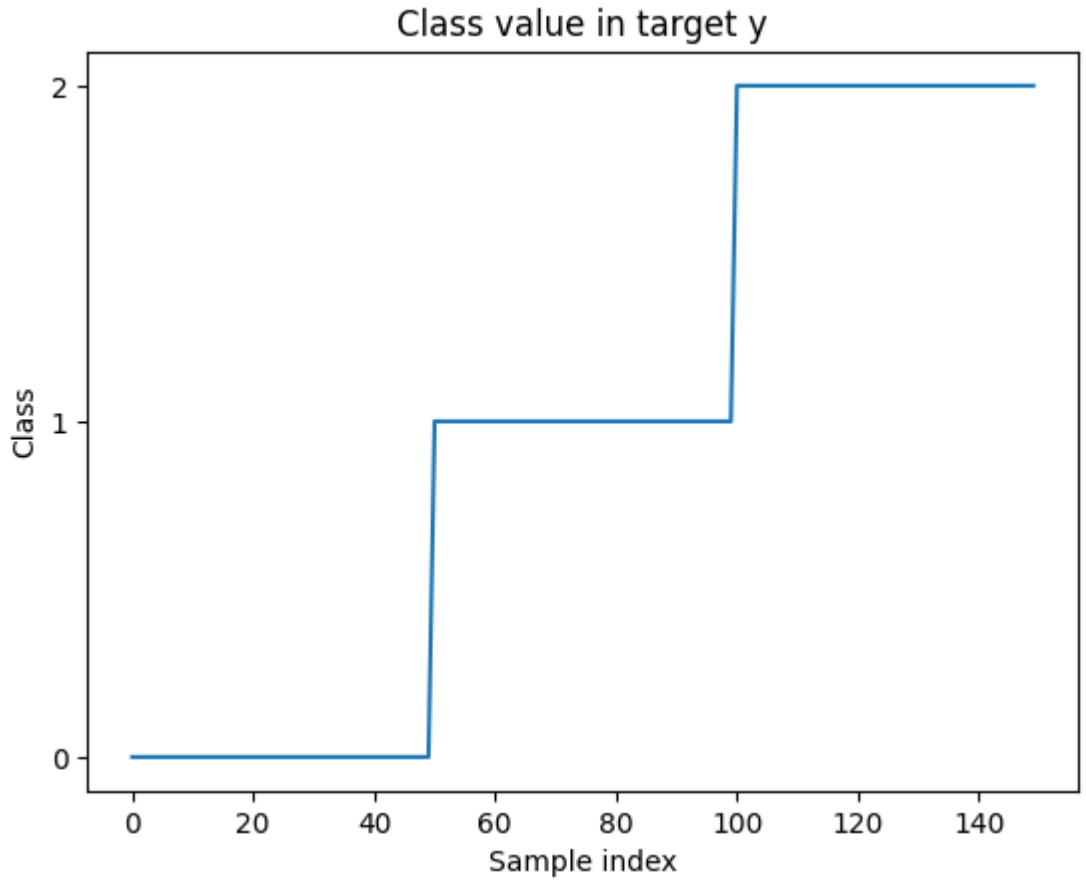
```
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import make_pipeline
model = make_pipeline(StandardScaler(), LogisticRegression())
from sklearn.model_selection import ShuffleSplit
```

Ορίζοντας τρεις διαχωρισμούς (k=3), θα χρησιμοποιήσουμε το 1/3 των δειγμάτων για δοκιμή και τα υπόλοιπα για εκπαίδευση. Σημειώνουμε ότι το KFold δεν ανακατεύει τα δεδομένα από προεπιλογή. Αυτό σημαίνει ότι θα επιλέξει το πρώτο 1/3 των δειγμάτων για το σετ δοκιμών κατά την πρώτη διαίρεση, στη συνέχεια το επόμενο τρίτο των δειγμάτων για τον δεύτερο διαχωρισμό και το τελευταίο τρίτο των δειγμάτων για την τελευταία διάσπαση. Στο τέλος, όλα τα δείγματα θα έχουν χρησιμοποιηθεί σε δοκιμές τουλάχιστον μία φορά ανάμεσα στους διαφορετικούς διαχωρισμούς.

Τώρα, ας εφαρμόσουμε αυτήν την στρατηγική για να ελέγξουμε την απόδοση γενίκευσης του δικού μας μοντέλου.

```
from sklearn.model_selection import cross_validate
from sklearn.model_selection import KFold
cv = KFold(n_splits=3)
results = cross_validate(model, data, target, cv=cv)
test_score = results["test_score"]
print(
f"The average accuracy is {test_score.mean():.3f} ±
{test_score.std():.3f}"
)

# The average accuracy is 0.000 ± 0.000
```

Είναι μια πραγματική έκπληξη που το μοντέλο μας δεν μπορεί να ταξινομήσει σωστά κανένα δείγμα σε οποιαδήποτε διάσπαση διασταυρούμενης επικύρωσης. Θα ελέγξουμε τώρα την τιμή του στόχου μας για να καταλάβουμε το θέμα.



Διάγραμμα 6.7: Διάταξη κλάσεων στο σετ δεδομένων Iris

Βλέπουμε ότι το διάνυσμα-στόχος target είναι διατεταγμένο. Αυτό έχει μερικές απροσδόκητες συνέπειες κατά τη χρήση της διασταυρούμενης επικύρωσης KFold: σε κάθε μέρος του διαχωρισμού υπάρχουν μόνο δύο από τις τρεις κατηγορίες στο σετ εκπαίδευσης και όλα τα δείγματα της υπόλοιπης κατηγορίας χρησιμοποιούνται ως σετ δοκιμής. Έτσι το μοντέλο μας δεν είναι σε θέση να προβλέψει την κατηγορία που δεν ήταν εμφανής κατά τη διάρκεια της εκπαίδευσης.

Ωστόσο, κάποιος μπορεί να θέλει να χωρίσει τα δεδομένα μας διατηρώντας τη συχνότητα των κλάσεων: θέλουμε δηλαδή να διαστρωματώσουμε (stratify) τα δεδομένα μας ανά κλάση. Στο scikit-learn υπάρχουν μερικές στρατηγικές διασταυρούμενης επικύρωσης που εφαρμόζουν τη διαστρωμάτωση (περιέχουν τη λέξη Stratified στα ονόματά τους).

```
from sklearn.model_selection import StratifiedKFold
cv = StratifiedKFold(n_splits=3)
results = cross_validate(model, data, target, cv=cv)
test_score = results["test_score"]
print(f"The average accuracy is {test_score.mean():.3f} ±
test_score.std():.3f}")

# The average accuracy is 0.967 ± 0.009
```

6.5. Υπερπροσαρμογή (Overfitting) και Υποπροσαρμογή (Underfitting)

Κατά τη διάρκεια κατασκευής μοντέλων μηχανικής μάθησης, η υπερπροσαρμογή και η υποπροσαρμογή αποτελούν δύο από τις μεγαλύτερες προκλήσεις που μπορεί να συναντήσει κάποιος. Και οι δύο αυτές καταστάσεις σχετίζονται με την ικανότητα του μοντέλου να γενικεύει καλά σε νέα, άγνωστα δεδομένα. Σε αυτό το κεφάλαιο θα εξετάσουμε τι είναι η υπερπροσαρμογή και η υποπροσαρμογή, πώς αναγνωρίζονται και πώς μπορούν να αντιμετωπιστούν.

6.5.1. Υποπροσαρμογή (Underfitting)

Η υποπροσαρμογή συμβαίνει όταν ένα μοντέλο δεν μπορεί να «μάθει» επαρκώς από τα δεδομένα εκπαίδευσης και αποτυγχάνει να ενσωματώσει τις υποκείμενες τάσεις τους. Στην πράξη, αυτό σημαίνει ότι το μοντέλο είναι πολύ απλό για να περιγράψει τη σχέση μεταξύ των μεταβλητών εισόδου και εξόδου.

6.5.1.1 Χαρακτηριστικά της υποπροσαρμογής:

- Χαμηλή ακρίβεια τόσο στα δεδομένα εκπαίδευσης όσο και στα δεδομένα ελέγχου.
- Απλό μοντέλο που δεν μπορεί να αιχμαλωτίσει τη σύνθετη συμπεριφορά των δεδομένων.
- Μεγάλο σφάλμα σε όλες τις φάσεις, καθώς το μοντέλο δεν μπορεί να γενικεύσει σωστά.

6.5.1.2 *Αιτίες υποπροσαρμογής:*

- Πολύ απλός αλγόριθμος: Για παράδειγμα, η χρήση γραμμικής παλινδρόμησης σε ένα μη γραμμικό πρόβλημα.
- Ανεπαρκής χρόνος εκπαίδευσης.
- Ανεπαρκής ποσότητα χαρακτηριστικών ή έλλειψη σημαντικών χαρακτηριστικών στο σετ δεδομένων.

6.5.1.3 *Τρόποι αντιμετώπισης της υποπροσαρμογής:*

- **Χρήση πιο περίπλοκου μοντέλου:** Ένα πιο ισχυρό μοντέλο μπορεί να προσαρμοστεί καλύτερα στα δεδομένα.
- **Επιλογή περισσότερων χαρακτηριστικών:** Προσθήκη σημαντικών παραμέτρων που μπορούν να ενισχύσουν το μοντέλο.
- **Περισσότερος χρόνος εκπαίδευσης:** Αύξηση του αριθμού των εποχών (epochs) ή της ακρίβειας κατά την εκπαίδευση.

6.5.2. Υπερπροσαρμογή (Overfitting)

Η υπερπροσαρμογή συμβαίνει όταν ένα μοντέλο είναι τόσο προσαρμοσμένο στα δεδομένα εκπαίδευσης που αποτυγχάνει να γενικεύσει σε νέα δεδομένα. Το μοντέλο «μαθαίνει» πολύ καλά τα δεδομένα εκπαίδευσης, περιλαμβάνοντας ακόμη και θόρυβο ή μικρές τυχαίες διακυμάνσεις που δεν αντιπροσωπεύουν την πραγματική σχέση μεταξύ εισόδου και εξόδου.

6.5.2.1 *Χαρακτηριστικά της υπερπροσαρμογής:*

- Πολύ καλή απόδοση στα δεδομένα εκπαίδευσης, αλλά σημαντική μείωση της απόδοσης στα δεδομένα ελέγχου.
- Το μοντέλο γίνεται πολύ περίπλοκο και αιχμαλωτίζει θόρυβο αντί για πραγματικά μοτίβα.
- Χαμηλό σφάλμα εκπαίδευσης, αλλά υψηλό σφάλμα γενίκευσης.

6.5.2.2 *Αιτίες υπερπροσαρμογής:*

- **Πολύ περίπλοκο μοντέλο:** Μοντέλα με πολλούς παραμέτρους ή μεγάλη ευελιξία.
- **Υπερβολικά μεγάλη εκπαίδευση:** Το μοντέλο προσαρμόζεται υπερβολικά στα δεδομένα εκπαίδευσης αν εκπαιδεύεται για πάρα πολύ χρόνο.
- **Μικρό σύνολο δεδομένων:** Όταν τα δεδομένα δεν επαρκούν, το μοντέλο μπορεί να απομνημονεύσει τις λεπτομέρειες των δεδομένων εκπαίδευσης, χωρίς να μάθει τις γενικές τάσεις.

6.5.2.3 Τρόποι αντιμετώπισης της υπερπροσαρμογής:

- **Χρήση κανονικοποίησης (Regularization):** Χρήση ειδικών τεχνικών όπως L1, L2, σχεδιασμένων ειδικά για να περιοριστεί η πολυπλοκότητα του μοντέλου.
- **Επιλογή/μείωση χαρακτηριστικών:** Μείωση της πολυπλοκότητας του μοντέλου επιλέγοντας μόνο τα πιο σημαντικά χαρακτηριστικά ή χρησιμοποιώντας τεχνικές όπως η ανάλυση κύριων συνιστωσών (PCA) για τη μείωση της διαστατικότητας των δεδομένων.
- **Ρύθμιση υπερπαραμέτρων:** Ρυθμίζοντας τις υπερπαραμέτρους του μοντέλου (όπως ο ρυθμός μάθησης, το βάθος των δέντρων στα δέντρα απόφασης κ.λπ.) χρησιμοποιώντας τεχνικές όπως η αναζήτηση πλέγματος ή η τυχαία αναζήτηση, μπορούμε να βρούμε τη βέλτιστη διαμόρφωση που εξισορροπεί τη μεροληψία και τη διακύμανση (bias–variance tradeoff).
- **Χρήση περισσότερων δεδομένων:** Η αύξηση του μεγέθους του συνόλου δεδομένων μπορεί να βοηθήσει το μοντέλο να μάθει τις γενικές τάσεις χωρίς να προσαρμόζεται υπερβολικά στις λεπτομέρειες των δεδομένων εκπαίδευσης.
- **Πρόωρη διακοπή (Early Stopping):** Διακοπή της εκπαίδευσης όταν η απόδοση στα δεδομένα ελέγχου αρχίζει να επιδεινώνεται, αποφεύγοντας έτσι την υπερπροσαρμογή.

6.5.3. Πώς να εντοπίσετε την υπερπροσαρμογή και την υποπροσαρμογή

Ο καλύτερος τρόπος για να εντοπιστεί η υπερπροσαρμογή και η υποπροσαρμογή είναι η παρακολούθηση της απόδοσης του μοντέλου κατά τη διάρκεια της εκπαίδευσης στα δεδομένα εκπαίδευσης και τα δεδομένα ελέγχου.

- **Υποπροσαρμογή:** Η απόδοση είναι κακή τόσο στα δεδομένα εκπαίδευσης όσο και στα δεδομένα ελέγχου. Το μοντέλο δεν έχει μάθει επαρκώς τα χαρακτηριστικά.
- **Υπερπροσαρμογή:** Πολύ χαμηλό σφάλμα εκπαίδευσης, αλλά υψηλό σφάλμα στα δεδομένα ελέγχου. Το μοντέλο έχει μάθει πολύ καλά τα δεδομένα εκπαίδευσης, αλλά δεν μπορεί να γενικεύσει.

Η χρήση ενός διαγράμματος μάθησης (learning curve) μπορεί να βοηθήσει στην καλύτερη κατανόηση της κατάστασης. Σε αυτό το διάγραμμα, απεικονίζεται το σφάλμα εκπαίδευσης και το σφάλμα ελέγχου ως συνάρτηση του χρόνου εκπαίδευσης. Η υπερπροσαρμογή φαίνεται όταν το σφάλμα ελέγχου αρχίζει να αυξάνεται ενώ το σφάλμα εκπαίδευσης μειώνεται.

Η ισορροπία μεταξύ υπερπροσαρμογής και υποπροσαρμογής είναι καθοριστικής σημασίας για την κατασκευή ενός επιτυχημένου μοντέλου. Στόχος είναι να βρούμε ένα μοντέλο που να είναι αρκετά

απλό για να μην υπερπροσαρμόζεται, αλλά ταυτόχρονα αρκετά σύνθετο ώστε να μαθαίνει τις σημαντικές πληροφορίες από τα δεδομένα.

6.6. *Ερωτήσεις αυτοαξιολόγησης*

6.1 Ποιος είναι ο σκοπός του διαχωρισμού των δεδομένων σε εκπαίδευσης/ελέγχου (train/test) στη μηχανική μάθηση;

α) Για να δημιουργήσουμε πολλαπλά υποσύνολα του συνόλου δεδομένων στην τεχνική cross–validation

β) Για να αξιολογηθεί η απόδοση του μοντέλου σε νέα δεδομένα

γ) Για να αυξηθεί η πολυπλοκότητα του μοντέλου για καλύτερες προβλέψεις

δ) Προεπεξεργασία και μετατροπή του συνόλου δεδομένων πριν από την εκπαίδευση

6.2 Ποια από τις παρακάτω μετρικές απόδοσης χρησιμοποιείται για προβλήματα δυαδικής ταξινόμησης (binary classification);

α) Mean Absolute Error (MAE)

β) F1 score

γ) R-squared

6.3 Κατά την αξιολόγηση ενός μοντέλου παλινδρόμησης (regression model), ποια από τις παρακάτω μετρική αξιολόγησης είναι η καταλληλότερη εάν θέλουμε να τιμωρήσουμε μεγαλύτερα σφάλματα παρά μικρότερα σφάλματα;

α) Root Mean Squared Error (RMSE)

β) Mean Absolute Error (MAE)

γ) R-squared

6.4 Όταν έχουμε υπερπροσαρμογή (overfitting), η διαφορά του σφάλματος μεταξύ των δεδομένων εκπαίδευσης (training error) και των δεδομένων ελέγχου (test error) είναι

α) Μεγάλη

β) Μικρή

γ) Ίδια

ΚΕΦΑΛΑΙΟ 7: Naïve Bayes

7.1. Βασική αρχή ταξινομητή Naive Bayes

Οι μέθοδοι μηχανικής μάθησης οι οποίοι ανήκουν στην ομάδα Naive Bayes είναι ένα σύνολο εποπτευόμενων αλγορίθμων μάθησης οι οποίοι βασίζονται στην εφαρμογή του θεωρήματος του Bayes με την «αφελή (naive)» υπόθεση ότι υπάρχει υπό συνθήκη ανεξαρτησία (conditional independence) μεταξύ κάθε ζεύγους χαρακτηριστικών (feature) του δείγματος (κάτι το οποίο δεν ισχύει στα σύνθετα πραγματικά δεδομένα, αλλά ως υπόθεση εργασίας είναι σχετικά αξιόπιστη).

Ένα πρόβλημα ταξινόμησης, στη γενική του μορφή, αποτελείται από σειρά παρατηρήσεων σε χαρακτηριστικά των μονάδων ενός πληθυσμού (μια παρατήρηση θεωρείται ότι αποτελεί ένα σύνολο τιμών για τα χαρακτηριστικά υπό μελέτη), και αριθμό κατηγοριών στις οποίες η κάθε παρατήρηση (πλειάδα τιμών) είναι επιθυμητό να ενταχθεί με κάποιο κανόνα ταξινόμησης.

Στην προσέγγιση των μεθόδων της ομάδας Naive Bayes η βασική αρχή είναι η ακόλουθη.

Για ένα τυπικό πρόβλημα ταξινόμησης, θεωρείται:

- Ένα σύνολο n χαρακτηριστικών $\{x_1, x_2, \dots, x_n\}$ για τα οποία εκτελούνται μετρήσεις/παρατηρήσεις (για ένα οποιοδήποτε πρόβλημα ταξινόμησης). Κάθε μέτρηση αποτελείται από πλειάδα τιμών, μια για κάθε χαρακτηριστικό $\{x_1, x_2, \dots, x_n\}$.
- Ένα σύνολο i παρατηρήσεων, δηλαδή i πλειάδων $(x_1, x_2, \dots, x_n)$, όπου η κάθε μια αντιστοιχεί σε μια μέτρηση των χαρακτηριστικών (κάθε παρατήρηση αντιστοιχεί σε μια οντότητα του πληθυσμού ή δείγματος υπό μελέτη).
- Ένα σύνολο κατηγοριών στις οποίες η κάθε πλειάδα τιμών (μια παρατήρηση) είναι επιθυμητό να ενταχθεί, τις οποίες παριστάνουμε με τη μεταβλητή κλάσης y.

Με βάση το θεώρημα του Bayes ισχύει ότι:

$$P(y|x_1, x_2, \dots, x_n) = \frac{P(y)P(x_1, x_2, \dots, x_n|y)}{P(x_1, x_2, \dots, x_n)}$$

Χρησιμοποιώντας «αφελή (naive)» υπόθεση ότι υπάρχει υπό συνθήκη ανεξαρτησία (conditional independence) μεταξύ κάθε ζεύγους χαρακτηριστικών (feature) του δείγματος, δηλαδή

$$P(x_i|y, x_1, x_2, .., x_{i-1}, x_{i+1}, .., x_n) = P(x_i|y)$$

για κάθε i, η σχέση μπορεί να απλοποιηθεί στην παρακάτω, η οποία επιτρέπει τον υπολογισμό των δεσμευμένων πιθανοτήτων:

$$P(y|x_1, x_2, \dots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, x_2, \dots, x_n)}$$

Με δεδομένο ότι ο όρος $P(x_1, x_2, \dots, x_n)$ είναι σταθερός για συγκεκριμένο σύνολο δεδομένων (είσοδο), ο κανόνας ταξινόμησης που μπορεί να χρησιμοποιηθεί είναι ο παρακάτω:

$$P(y|x_1, x_2, \dots, x_n) \propto P(y)\prod\nolimits_{i=1}^{n} P(x_i|y)$$

Οπότε με τον υπολογισμό των πιθανοτήτων $P(y)$ και $P(x_i|y)$ (με τη μέθοδο μεγιστοποίησης της εκ των υστέρων πιθανότητας / Maximum A Posteriori) η κλάση στην οποία «εκτιμάται» ότι ανήκει η i πλειάδα (παρατήρηση) είναι αυτή η οποία μεγιστοποιεί την πιθανότητα, δηλαδή

$$\hat{y} = argmax P(y)\prod\nolimits_{i=1}^{n} P(x_i|y)$$

Οι ταξινομητές της ομάδας Naive Bayes διαφέρουν κυρίως ως προς τις υποθέσεις τι οποίες έχουν σχετικά με την κατανομή $P(x_i|y)$.

Παρά τις φαινομενικά υπεραπλουστευμένες υποθέσεις, οι ταξινομητές της ομάδας Naive Bayes λειτουργούν αρκετά καλά σε πολλές πραγματικές καταστάσεις, όπως ταξινόμηση εγγράφων και φιλτράρισμα ανεπιθύμητων μηνυμάτων και απαιτούν μικρό όγκο δεδομένων εκπαίδευσης για την εκτίμηση των απαραίτητων παραμέτρων. Επίσης, είναι εξαιρετικά γρήγοροι σε σύγκριση με περισσότερο εξελιγμένες μεθόδους και συχνά χρησιμοποιούνται ως σημείο αναφοράς (benchmark) για νέους αλγόριθμους.

7.2. Gaussian Naive Bayes

Ο Gaussian Naive Bayes αλγόριθμος θεωρεί ότι η κατανομή πιθανότητας των χαρακτηριστικών είναι Gaussian:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)}$$

Οι παράμετροι σ_y και μ_y εκτιμώνται χρησιμοποιώντας τη μέθοδο της μέγιστης πιθανοφάνειας (maximum likelihood).

7.3. Bernoulli Naive Bayes

Ο Bernoulli Naive Bayes θεωρεί ότι τα δεδομένα κατανέμονται με βάση πολυμεταβλητές κατανομές Bernoulli. Δηλαδή, μπορεί να υπάρχουν πολλαπλά χαρακτηριστικά, αλλά το καθένα υποτίθεται ότι είναι μια μεταβλητή δυαδικής αξίας (Bernoulli, boolean). Επομένως, αυτή η μέθοδος απαιτεί τα δείγματα να αναπαρίστανται ως διανύσματα δυαδικών τιμών χαρακτηριστικών (0,1).

Ο κανόνας απόφασης για τον Bernoulli Naive Bayes βασίζεται στον κανόνα

$$P(x_i|y) = P(x_i = 1|y)x_i + (1 - P(x_i = 1|y))(1 - x_i)$$

Ο οποίος έχει την ιδιαιτερότητα ότι τιμωρεί τη μη εμφάνιση ενός χαρακτηριστικού (κάτι το οποίο είναι χρήσιμο σε ορισμένα προβλήματα).

7.4. Παράδειγμα κατανόησης 1

Ας υποτεθεί το παρακάτω σύνολο δεδομένων το οποίο αφορά καταγεγραμμένα ιστορικά στοιχεία φορολογουμένων με την ταξινόμησή τους σε δύο κλάσεις, ανάλογα με την υποψία για φοροαποφυγή. Το ζητούμενο είναι να δημιουργηθεί κατάλληλος ταξινομητής για την ταξινόμηση νέων παρατηρήσεων (φορολογουμένων) ως προς την υποψία για φοροαποφυγή. Για παράδειγμα, πώς θα πρέπει να ταξινομηθεί η παρατήρηση με

$$X = (επιστροφή = ΟΧΙ, οικογ.κατ. = \text{ έγγαμος η}, εισόδημα = 120)$$

	Χαρακτηριστικό (feature)			**Κλάση**
Παρατήρηση	**Φορολογητέο εισόδημα**	**Οικογενακή κατάσταση**	**Επιστροφή φόρου**	**Υποψία για Φοροαποφυγή**
1	125	Άγαμος/η	Ναι	Όχι
2	100	Έγγαμος/η	Όχι	Όχι
3	70	Άγαμος/η	Όχι	Όχι
4	120	Έγγαμος/η	Ναι	Όχι
5	95	Διαζευγμένος/η	Όχι	Ναι
6	60	Έγγαμος/η	Όχι	Όχι
7	220	Διαζευγμένος/η	Ναι	Όχι
8	85	Άγαμος/η	Όχι	Ναι
9	75	Έγγαμος/η	Όχι	Όχι
10	90	Άγαμος/η	Όχι	Ναι

Με βάση την ιδέα του ταξινομητή Naïve Bayes, το ζητούμενο είναι να υπολογιστούν οι πιθανότητες της X για κάθε κλάση, και να επιλεγεί στη συνέχεια η κλάση με την υψηλότερη πιθανότητα (κριτήριο Maximum A Posteriori Probability/MAP).

Δηλαδή

$P(\ φοροαποφυγή = OXI \mid X) =?$

$P(\ φοροαποφυγή = NAI \mid X)=?$

Ακολουθώντας τηνπροσέγγιση του Naïve Bayes, αρχικά υπολογίζεται η πιθανότητα εμφάνισης κάθε κλάσης, δηλαδή φοροαποφυγή ή όχι (με βάση πάντοτε τα διαθέσιμα δεδομένα). Υπολογίζονται ως συχνότητες με βάση τον ορισμό της πιθανότητας.

$$P(κλάσης) = \frac{συχνότητα\ κλάσης}{συνολικό\ πλήθος} = \frac{N_C}{N}$$

Οπότε είναι

$$P(φοροαποφυγή = OXI) = \frac{3}{10}$$

$$P(φοροαποφυγή = NAI) = \frac{7}{10}$$

Στη συνέχεια υπολογίζονται οι δεσμευμένες πιθανότητες για κάθε χαρακτηριστικό και κάθε κλάση.

$$P(x_i|C_k) = \frac{|x_{ik}|}{N_C}$$

όπου $|x_{ik}|$ το πλήθος των χαρακτηριστικών που έχουν την τιμή i.

Οπότε είναι

$$P(επιστροφή = NAI \mid φοροαποφυγή = OXI) = \frac{3}{7}$$

$$P(επιστροφή = OXI \mid φοροαποφυγή = OXI) = \frac{4}{7}$$

$$P(επιστροφή = NAI \mid φοροαποφυγή = NAI) = 0$$

$P(\text{επιστροφή} = OXI \mid \text{φοροαποφυγή} = NAI) = 1$

$P(\text{οικογ.κατ.} = \text{Άγαμος} \mid \text{φοροαποφυγή} = OXI) = \frac{2}{7}$

$P(\text{οικογ.κατ.} = \text{Διαζευγμένος} \mid \text{φοροαποφυγή} = OXI) = \frac{1}{7}$

$P(\text{οικογ.κατ.} = \text{Έγγαμος} \mid \text{φοροαποφυγή} = OXI) = \frac{4}{7}$

$P(\text{οικογ.κατ.} = \text{Άγαμος} \mid \text{φοροαποφυγή} = NAI) = \frac{2}{7}$

$P(\text{οικογ.κατ.} = \text{Διαζευγμένος} \mid \text{φοροαποφυγή} = NAI) = \frac{1}{7}$

$P(\text{οικογ.κατ.} = \text{Έγγαμος} \mid \text{φοροαποφυγή} = NAI) = 0$

Ως προς το εισόδημα, το οποίο αποτελεί συνεχή μεταβλητή, ισχύει

Για την κλάση $\text{φοροαποφυγή} = OXI$ η μέση τιμή = 110 και η διακύμανση = 2975.

Για την κλάση $\text{φοροαποφυγή} = NAI$ η μέση τιμή = 90 και η διακύμανση = 25

Οπότε

$$P(X \mid \text{φοροαποφυγή} = OXI) = P(\text{επιστροφή} = OXI \mid \text{φοροαποφυγή} = OXI) * P(\text{οικογ.κατ.} = \text{έγγαμος/η} \mid \text{φοροαποφυγή} = OXI) * P(\text{εισόδημα} = 120 \mid \text{φοροαποφυγή} = OXI) = \frac{4}{7} * \frac{4}{7} * 0{,}0072 = 0{,}0024$$

$$P(X \mid \text{φοροαποφυγή} = NAI) = P(\text{επιστροφή} = OXI \mid \text{φοροαποφυγή} = NAI) * P(\text{οικογ.κατ.} = \text{έγγαμος/η} \mid \text{φοροαποφυγή} = NAI) * P(\text{εισόδημα} = 120 \mid \text{φοροαποφυγή} = NAI) = 1 * 0 * 10^{-9} = 0$$

Με βάση το παραπάνω σύνολο τιμών, έχουν υπολογιστεί οι πιθανότητες, και ο 'ταξινομητής' είναι έτοιμος για την ταξινόμηση νέας παρατήρησης.

Για παράδειγμα, για να ταξινομηθεί η παρατήρηση με

$$X = (\text{επιστροφή} = OXI, \text{οικογ.κατ.} = \text{έγγαμος } \eta, \text{εισόδημα} = 120)$$

πρέπει να υπολογιστούν οι πιθανότητες της X για κάθε κλάση και να επιλεγεί η κλάση με την υψηλότερη πιθανότητα (κριτήριο Maximum A Posteriori Probability/MAP). Από τα προηγούμενα προκύπτει ότι

$$P(X \mid \text{φοροαποφυγή} = OXI) * P(\text{φοροαποφυγή} = OXI) > P(X \mid \text{φοροαποφυγή} = NAI) * P(\text{φοροαποφυγή} = NAI)$$

Επομένως

$$P(\text{φοροαποφυγή} = OXI \mid X) > P(\text{φοροαποφυγή} = NAI \mid X)$$

Οπότε η κλάση για την παρατήρηση

$$X = (\text{επιστροφή} = OXI, \text{οικογ.κατ.} = \text{έγγαμος } \eta, \text{εισόδημα} = 120)$$

είναι η $\text{φοροαποφυγή} = OXI$.

7.5. Παράδειγμα κατανόησης 2

Δίνεται το παρακάτω σύνολο δεδομένων, με τέσσερα χαρακτηριστικά του καιρού, δεκατέσσερις παρατηρήσεις και δύο κλάσεις. Το ζητούμενο είναι να ταξινομηθεί η νέα παρατήρηση

$$X = (\text{Εικόνα} = \text{Ηλιοφάνεια}, \quad \text{Άνεμος} = \text{Ισχυρός}, \quad \text{Θερμοκρασία} = \text{Χαμηλή}, \text{Υγρασία} = \text{Υψηλή})$$

με βάση τον ταξινομητή Naïve Bayes,

	Χαρακτηριστικό (feature)				Κλάση
Ημέρα	**Εικόνα**	**Άνεμος**	**Θερμοκρασία**	**Υγρασία**	**Ευνοϊκός καιρός**
1	Ηλιοφάνεια	Ασθενής	Υψηλή	Υψηλή	Όχι
2	Ηλιοφάνεια	Ισχυρός	Υψηλή	Υψηλή	Όχι
3	Νεφελώδης	Ασθενής	Υψηλή	Υψηλή	Ναι
4	Βροχή	Ασθενής	Μέση	Υψηλή	Ναι
5	Βροχή	Ασθενής	Χαμηλή	Κανονική	Ναι
6	Βροχή	Ισχυρός	Χαμηλή	Κανονική	Όχι
7	Νεφελώδης	Ισχυρός	Χαμηλή	Κανονική	Ναι
8	Ηλιοφάνεια	Ασθενής	Μέση	Υψηλή	Όχι
9	Ηλιοφάνεια	Ασθενής	Χαμηλή	Κανονική	Ναι

10	Βροχή	Ασθενής	Μέση	Κανονική	Ναι
11	Ηλιοφάνεια	Ισχυρός	Μέση	Κανονική	Ναι
12	Νεφελώδης	Ισχυρός	Μέση	Υψηλή	Ναι
13	Νεφελώδης	Ασθενής	Υψηλή	Κανονική	Ναι
14	Βροχή	Ισχυρός	Μέση	Υψηλή	Όχι

Επίλυση

Υπολογισμός πιθανοτήτων για κάθε χαρακτηριστικό.

Εικόνα	**Ευνοϊκός καιρός** =ΝΑΙ	**Ευνοϊκός καιρός** =ΟΧΙ
Ηλιοφάνεια	2/9	3/5
Νεφελώδης	4/9	0/5
Βροχή	3/9	2/5

Θερμοκρασία	**Ευνοϊκός καιρός** =ΝΑΙ	**Ευνοϊκός καιρός** =ΟΧΙ
Υψηλή	2/9	2/5
Μέση	4/9	2/5
Χαμηλή	3/9	1/5

Άνεμος	**Ευνοϊκός καιρός** =ΝΑΙ	**Ευνοϊκός καιρός** =ΟΧΙ
Ισχυρός	3/9	3/5
Ασθενής	6/9	2/5

Υγρασία	**Ευνοϊκός καιρός** =ΝΑΙ	**Ευνοϊκός καιρός** =ΟΧΙ
Υψηλή	3/9	4/5
Κανονική	6/9	1/5

Για τη νέα παρατήρηση

$$X = (Εικόνα = Ηλιοφάνεια, \quad Άνεμος = Ισχυρός,$$
$$Θερμοκρασία = Χαμηλή, Υγρασία = Υψηλή)$$

Υπολογισμός πιθανοτήτων

$$\mathrm{P}(Εικόνα = Ηλιοφάνεια \mid \textbf{Ευνοϊκός καιρός} = \text{ΝΑΙ}) = 2/9$$

$\mathrm{P}(\Theta\varepsilon\rho\mu o\kappa\rho\alpha\sigma\acute{\iota}\alpha = X\alpha\mu\eta\lambda\acute{\eta} \mid$ **Ευνοϊκός καιρός** =ΝΑΙ) = $3/9$

$\mathrm{P}(\Upsilon\gamma\rho\alpha\sigma\acute{\iota}\alpha = \Upsilon\psi\eta\lambda\acute{\eta} \mid$ **Ευνοϊκός καιρός** =ΝΑΙ) = $3/9$

$\mathrm{P}(\acute{A}\nu\varepsilon\mu o\varsigma = I\sigma\chi\upsilon\rho\acute{o}\varsigma \mid$ **Ευνοϊκός καιρός** =ΝΑΙ) = $3/9$

$\mathrm{P}($**Ευνοϊκός καιρός** =ΝΑΙ) = $9/14$

Και

$\mathrm{P}(E\iota\kappa\acute{o}\nu\alpha = H\lambda\iota o\phi\acute{\alpha}\nu\varepsilon\iota\alpha \mid$ **Ευνοϊκός καιρός** =ΟΧΙ) = $3/5$

$\mathrm{P}(\Theta\varepsilon\rho\mu o\kappa\rho\alpha\sigma\acute{\iota}\alpha = X\alpha\mu\eta\lambda\acute{\eta} \mid$ **Ευνοϊκός καιρός** =ΟΧΙ) = $1/5$

$\mathrm{P}(\Upsilon\gamma\rho\alpha\sigma\acute{\iota}\alpha = \Upsilon\psi\eta\lambda\acute{\eta} \mid$ **Ευνοϊκός καιρός** =ΟΧΙ) = $4/5$

$\mathrm{P}(\acute{A}\nu\varepsilon\mu o\varsigma = I\sigma\chi\upsilon\rho\acute{o}\varsigma \mid$ **Ευνοϊκός καιρός** =ΟΧΙ) = $3/5$

$\mathrm{P}($**Ευνοϊκός καιρός** =ΟΧΙ) = $5/14$

Με βάση το κριτήριο Maximum A Posteriori Probability/MAP, το ζητούμενο είναι να υπολογιστούν οι πιθανότητες της X για κάθε κλάση και να επιλεγεί η κλάση με την υψηλότερη πιθανότητα.

Οπότε,

$\mathrm{P}($**Ευνοϊκός καιρός** =ΝΑΙ$\mid \mathbf{X})$: $[\mathrm{P}(E\iota\kappa\acute{o}\nu\alpha = H\lambda\iota o\phi\acute{\alpha}\nu\varepsilon\iota\alpha \mid$ **Ευνοϊκός καιρός** =ΝΑΙ $)\mathrm{P}(\Theta\varepsilon\rho\mu o\kappa\rho\alpha\sigma\acute{\iota}\alpha = X\alpha\mu\eta\lambda\acute{\eta} \mid$ **Ευνοϊκός καιρός** =ΝΑΙ $)\mathrm{P}(\Upsilon\gamma\rho\alpha\sigma\acute{\iota}\alpha = \Upsilon\psi\eta\lambda\acute{\eta} \mid$ **Ευνοϊκός καιρός** =ΝΑΙ $)\mathrm{P}(\acute{A}\nu\varepsilon\mu o\varsigma = I\sigma\chi\upsilon\rho\acute{o}\varsigma \mid$ **Ευνοϊκός καιρός** =ΝΑΙ $)]\mathrm{P}(\mathrm{Play}=Yes) = 0.0053$

$\mathrm{P}($**Ευνοϊκός καιρός** =ΟΧΙ $\mid \mathbf{X})$: $[\mathrm{P}(E\iota\kappa\acute{o}\nu\alpha = H\lambda\iota o\phi\acute{\alpha}\nu\varepsilon\iota\alpha \mid$ **Ευνοϊκός καιρός** =ΟΧΙ$)$ $\mathrm{P}(\Theta\varepsilon\rho\mu o\kappa\rho\alpha\sigma\acute{\iota}\alpha = X\alpha\mu\eta\lambda\acute{\eta} \mid$ **Ευνοϊκός καιρός** =ΟΧΙ$)\mathrm{P}(\Upsilon\gamma\rho\alpha\sigma\acute{\iota}\alpha = \Upsilon\psi\eta\lambda\acute{\eta} \mid$ **Ευνοϊκός καιρός** =ΟΧΙ$)\mathrm{P}(\acute{A}\nu\varepsilon\mu o\varsigma = I\sigma\chi\upsilon\rho\acute{o}\varsigma \mid$ **Ευνοϊκός καιρός** =ΟΧΙ$)]\mathrm{P}($**Ευνοϊκός καιρός** =ΟΧΙ$) = 0.0206$

Με δεδομένο ότι $\mathrm{P}($**Ευνοϊκός καιρός** =ΝΑΙ$\mid \mathbf{X}) < \mathrm{P}($**Ευνοϊκός καιρός** =ΟΧΙ $\mid \mathbf{X})$

Η παρατήρηση ταξινομείται στην κλάση **Ευνοϊκός καιρός** =ΟΧΙ.

7.6. Βιβλιοθήκη sklearn.naive_bayes

Η βιβλιοθήκη sklearn της python περιλαμβάνει τις παρακάτω υλοποιήσεις βασικών αλγορίθμων για την ομάδα ταξινομητών Naive Bayes (αποτελούν κλάσεις python και στο κώδικα πρέπει να δημιουργηθεί ένα αντικείμενο τέτοιου τύπου).

Sklearn Αλγόριθμος	Περιγραφή
BernoulliNB	Ταξινομητής Naive Bayes για μοντέλα με multivariate Bernoulli κατανομή δεδομένων.
CategoricalNB	Ταξινομητής Naive Bayes για κατηγορικά δεδομένα.
GaussianNB	Ταξινομητής για μοντέλα με Gaussian κατανομή δεδομένων.
MultinomialNB	Ταξινομητής Naive Bayes για πολυμεταβλητά μοντέλα.

Η διαδικασία για τη χρήση του κάθε αλγόριθμου είναι τυποποιημένη και το μόνο που απαιτείται είναι η προσεκτική χρήση των παραμέτρων κάθε κλάσης/αλγόριθμου, καθώς επηρεάζουν την ακρίβεια της ταξινόμησης.

Σε μια διαδικασία ταξινόμησης με μοντέλο Naive Bayes τα τυπικά βήματα είναι τα παρακάτω.

- Δημιουργία συνόλου δεδομένων παραδείγματος με κατάλληλη προεπεξεργασία και επεξεργασία ώστε να συμφωνούν με τις προδιαγραφές του αλγόριθμου.
- Διαμοιρασμός των δεδομένων σε σύνολο εκπαίδευσης και ελέγχου (συνήθως Εκπαίδευση/Τεστ = 80% / 20%). Η διαδικασία είναι δειγματοληψία και καθορίζει σε μεγάλο βαθμό το βαθμό ακρίβειας σε περίπτωση ανομοιογενούς πληθυσμού.
- Εκπαίδευση και προσαρμογή του μοντέλου Naive Bayes. Υπολογισμός των πιθανοτήτων και παραμέτρων.
- Αξιολόγηση της ακρίβειας του μοντέλου με μετρικές.
- Έκθεση αποτελεσμάτων ταξινόμησης (Confusion matrix, Confusion matrix σε διάγραμμα)
- Οπτικοποίηση των ορίων των κλάσεων για χαμηλής διάστασης δεδομένα.
- Ταξινόμηση νέων δεδομένων
- Οπτικοποίηση των δεδομένων και της ταξινόμησης για χαμηλής διάστασης δεδομένα.

Κύριες συναρτήσεις κοινές σε όλους τους αλγόριθμους του sklearn είναι οι,

- `fit(X, y, sample_weight=None)`

Fit Gaussian Naive Bayes according to X, y.

- partial_fit(X, y, classes=None, sample_weight=None)
  Incremental fit on a batch of samples.

- predict(X)
  Perform classification on an array of test vectors X.

- score(X, y, sample_weight=None)
  Return the mean accuracy on the given test data and labels.

- set_params(**params)
  Set the parameters of this estimator.

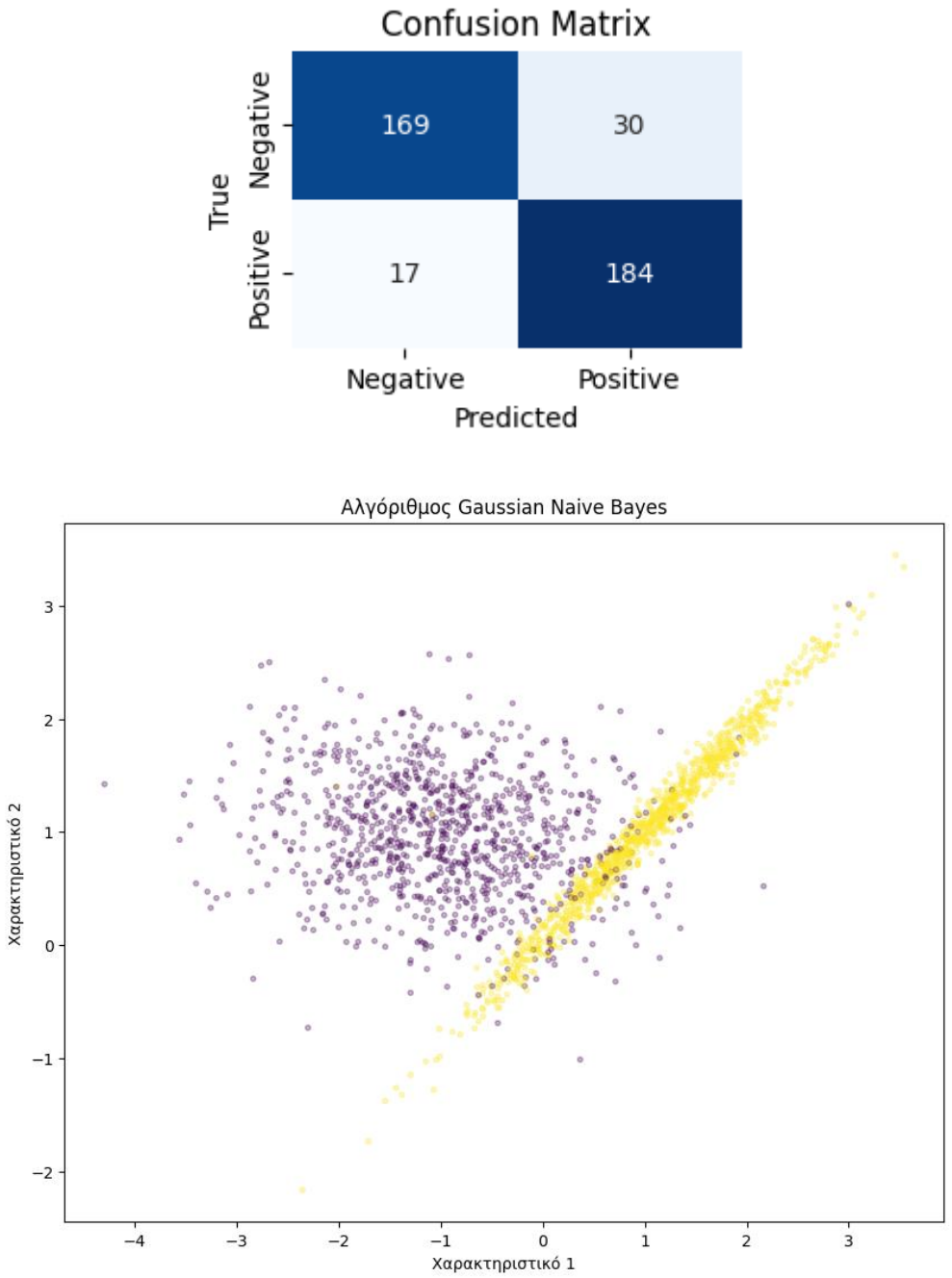
7.7. Παράδειγμα Gaussian Naive Bayes με python sklearn

```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Δημιουργία με τυχαίο τρόπο συνόλου δεδομένων παραδείγματος
# με τη χρήση της βιβλιοθήκης sklearn
# (2000 παρατηρήσεις, 2 χαρακτηριστικά, 2 κλάσεις)
X, y = make_classification(n_samples=2000, n_features=2,
n_informative=2, n_redundant=0, n_clusters_per_class=1,
random_state=42)

#print(X[1:3])

# Διαμοιρασμός των δεδομένων σε σύνολο εκπαίδευσης και ελέγχου
# Εκπαίδευση/Τεστ = 80% / 20%
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Εκπαίδευση του μοντέλου Gaussian Naive Bayes model
gaussian_nb = GaussianNB()
gaussian_nb.fit(X_train, y_train)

# Αξιολόγηση της ακρίβειας του μοντέλου
y_pred = gaussian_nb.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
```

```
print(f'Model Accuracy: {accuracy:.2f}')

# Έκθεση αποτελεσμάτων ταξινόμησης
print("Classification Report:")
print(classification_report(y_test, y_pred))

# Confusion matrix σε πίνακα
print("Confusion Matrix:")
cm = confusion_matrix(y_test, y_pred)
print(cm)

# Confusion matrix σε διάγραμμα σε seaborn
plt.figure(figsize=(3, 2))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", cbar=False,
            xticklabels=["Negative", "Positive"],
            yticklabels=["Negative", "Positive"])
plt.xlabel("Predicted")
plt.ylabel("True")
plt.title("Confusion Matrix")
plt.show()

# Οπτικοποίηση των δεδομένων και της ταξινόμησης σε matplotlib.pyplot
# Δεν είναι αναγκαίο, αλλά βοηθά την εποπτεία
# Σε δεδομένα υψηλών διαστάσεων (>3) δεν είναι εφικτή η οπτικοποίηση,
# οπότε γίνεται επιλογή διαστάσεων
plt.figure(figsize=(10, 8))

plt.scatter(X[:, 0], X[:, 1], c=y,  marker='o', alpha=0.3, s=10)

plt.title('Αλγόριθμος Gaussian Naive Bayes')
plt.xlabel('Χαρακτηριστικό 1')
plt.ylabel('Χαρακτηριστικό 2')

plt.show()
```

```
Model Accuracy: 0.88

Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.85      0.88       199
           1       0.86      0.92      0.89       201

    accuracy                           0.88       400
   macro avg       0.88      0.88      0.88       400
weighted avg       0.88      0.88      0.88       400

Confusion Matrix:
```

```
[[169  30]
 [ 17 184]]
```



Διάγραμμα 7.1 Πίνακας σύγχυσης και γράφημα διασποράς

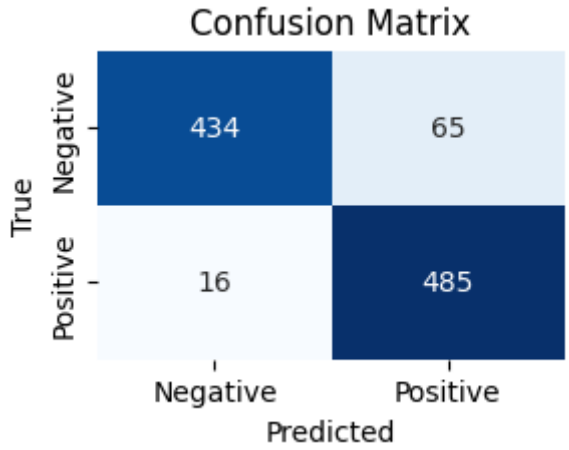
7.8. Παράδειγμα Bernoulli Naive Bayes με python sklearn

```
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import BernoulliNB
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
```

```
# Δημιουργία τυχαίου συνόλου δεδομένων παραδείγματος
# (5000 παρατηρήσεις, 15 χαρακτηριστικά, 2 κλάσεις)
X, y = make_classification(n_samples=5000, n_features=15, n_classes=2,
n_clusters_per_class=1, random_state=42)

#print(X[1:3])

# Διαμοιρασμός των δεδομένων σε σύνολο εκπαίδευσης και ελέγχου
# Εκπαίδευση/Τεστ = 80% / 20%
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Εκπαίδευση του μοντέλου Bernoulli Naive Bayes
bernoulli_nb = BernoulliNB()
bernoulli_nb.fit(X_train, y_train)

# Αξιολόγηση της ακρίβειας του μοντέλου
y_pred = bernoulli_nb.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f'Model Accuracy: {accuracy:.2f}')

# Έκθεση αποτελεσμάτων ταξινόμησης
print("Classification Report:")
print(classification_report(y_test, y_pred))

# Confusion matrix σε πίνακα
print("Confusion Matrix:")
cm = confusion_matrix(y_test, y_pred)
print(cm)

# Confusion matrix σε διάγραμμα
plt.figure(figsize=(3, 2))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", cbar=False,
            xticklabels=["Negative", "Positive"],
            yticklabels=["Negative", "Positive"])
plt.xlabel("Predicted")
plt.ylabel("True")
plt.title("Confusion Matrix")
plt.show()
```

```
Model Accuracy: 0.92

Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.87      0.91       499
```

```
           1       0.88      0.97      0.92       501

    accuracy                           0.92      1000
   macro avg       0.92      0.92      0.92      1000
weighted avg       0.92      0.92      0.92      1000

Confusion Matrix:
[[434  65]
 [ 16 485]]
```



Διάγραμμα 7.2 Πίνακας σύγχυσης

7.9. *Ερωτήσεις αυτοαξιολόγησης*

7.1 Ποια είναι η υπόθεση σε έναν ταξινομητή Naïve Bayes:
α) όλες οι τάξεις (class) είναι ανεξάρτητες μεταξύ τους
β) όλα τα χαρακτηριστικά μιας τάξης είναι ανεξάρτητα μεταξύ τους
γ) το πιο πιθανό χαρακτηριστικό για μια τάξη είναι το πιο σημαντικό χαρακτηριστικό που πρέπει να εξεταστεί για ταξινόμηση
δ) όλα τα χαρακτηριστικά μιας τάξης εξαρτώνται υπό όρους το ένα από το άλλο

7.2 Ο ταξινομητής Naïve Bayes ανήκει στην κατηγορία μοντέλων μηχανικής μάθησης:
α) υπό επίβλεψη
β) χωρίς επίβλεψη
γ) και υπό επίβλεψη και χωρίς επίβλεψη
δ) με ενίσχυση

7.3 Μειονέκτημα του ταξινομητή Naïve Bayes αποτελεί το ότι:
α) Ο Naive Bayes υποθέτει ότι όλα τα γνωρίσματα (features) είναι ανεξάρτητα ή μη σχετιζόμενα, επομένως δεν μπορεί να μάθει τη σχέση μεταξύ των γνωρισμάτων.
β) Έχει καλή απόδοση σε προβλέψεις με πολλαπλές τάξεις (class) σε σύγκριση με άλλους αλγόριθμους.
γ) Μπορεί να χρησιμοποιηθεί για δυαδικές ταξινομήσεις (binary) καθώς και σε ταξινομήσεις πολλαπλών τάξεων.
δ) Είναι η πιο δημοφιλής επιλογή για προβλήματα ταξινόμησης κειμένου.

7.4 Παραδείγματα χρήσης του αλγόριθμου Naïve Bayes αποτελεί το:
α) Φιλτράρισμα ανεπιθύμητων μηνυμάτων (spam filtering)
β) Ανάλυση συναισθήματος (sentiment analysis)
γ) Ταξινόμηση άρθρων
δ) Όλα τα παραπάνω

ΚΕΦΑΛΑΙΟ 8: Μελέτη περίπτωσης Naïve Bayes

8.1. Εισαγωγικά

Στη μηχανική μάθηση, οι απλοί ταξινομητές Bayes είναι μια οικογένεια απλών «πιθανοτικών ταξινομητών» οι οποίοι βασίζονται στην εφαρμογή του θεωρήματος Bayes με ισχυρές (αλλά αφελείς, καθώς δεν ισχύουν εκτενώς στην πράξη) υποθέσεις ανεξαρτησίας μεταξύ των χαρακτηριστικών. Στη θεωρία και τη στατιστική πιθανοτήτων, το θεώρημα του Bayes (εναλλακτικά νόμος Bayes ή κανόνας Bayes) περιγράφει την πιθανότητα ενός γεγονότος, ενημερωμένη με βάση την προηγούμενη γνώση των συνθηκών που μπορεί να σχετίζονται με το γεγονός.

Στο παρόν κεφάλαιο θα παρουσιαστεί αναλυτικά μια μελέτη περίπτωσης μηχανικής μάθησης με εφαρμογή ταξινομητών της ομάδας Naïve Bayes. Ο στόχος είναι η κατανόηση αφενός της χρήσης των ταξινομητών σε πραγματικά προβλήματα, και αφετέρου η ανάδειξη της πολυπλοκότητας τόσο των δεδομένων του πραγματικού κόσμου, όσο και της διαδικασίας επίλυσης ενός σύνθετου προβλήματος.

Το πρόβλημα που θα παρουσιαστεί είναι η δημιουργία ταξινομητή για την ταξινόμηση μηνυμάτων κειμένου (SMS) σε ανεπιθύμητα (spam) και μη (ham). Οι όροι spam/ham έχουν καθιερωθεί για τον χαρακτηρισμό μηνυμάτων κειμένου, όπως ηλεκτρονική αλληλογραφία, κείμενο κλπ. Το πρόβλημα παρουσιάζεται σε σχετικά απλοποιημένη μορφή για καλύτερη κατανόηση της ροής των εργασιών.

Το πρόβλημα θα επιλυθεί με εναλλακτικά μοντέλα ώστε να πραγματοποιηθεί σύγκριση και αξιολόγηση τους. Αρχικά θα παρουσιαστεί η δημιουργία ενός φίλτρου ανεπιθύμητων μηνυμάτων χρησιμοποιώντας τον αλγόριθμο Naive Bayes με βασική υλοποίηση στην python. Το μοντέλο θα κατασκευαστεί για να προβλέψει την πιθανότητα ένα συγκεκριμένο κείμενο να είναι ανεπιθύμητο ή μη ανεπιθύμητο με βάση τα δεδομένα εκπαίδευσης στα οποία έχει εκτεθεί το μοντέλο. Η επίλυη θα πραγματοποιηθεί με τον υπολογισμό των πιθανοτήτων με βάση το θεώρημα του Bayes. Στη συνέχεια θα αναπτυχθούν εναλλακτικά μοντέλα με τη χρήση των κλάσεων Naive Bayes της βιβλιοθήκης Naïve Bayes sklearn, και θα εκτελεστεί σύγκριση μεταξύ τους.

Το σύνολο δεδομένων εκπαίδευσης το οποίο θα χρησιμοποιηθεί αποτελείται από μηνύματα κειμένου (συμβολοσειρές) και μια μεταβλητή στόχο, η οποία είναι η μεταβλητή της κλάσης και καθορίζει εάν το μήνυμα κειμένου ήταν ανεπιθύμητο ή όχι (spam/ham). Το σύνολο δεδομένων είναι ανοικτό και διαθέσιμο στο αποθετήριο μηχανικής μάθησης του UCI (https://archive.ics.uci.edu/dataset/228/sms+spam+collection). Το αρχικό αρχείο όπως και

λεπτομέρειες για τη σύνθεση του αρχείου και την προέλευση των δεδομένων περιέχονται στη σχετική ιστοσελίδα. Το αρχείο περιλαμβάνει 5.574 μηνύματα, τα οποία είναι ταξινομημένα σε ανεπιθύμητα (spam) και μη (ham), με 4.827 μη ανεπιθύμητα μηνύματα SMS (86,6%), και συνολικά 747 (13,4%) ανεπιθύμητα μηνύματα.

Το αρχείο περιέχει ένα μήνυμα ανά γραμμή. Κάθε γραμμή αποτελείται από δύο στήλες: μία με την ετικέτα (spam ή ham) και μια δεύτερη με το ακατέργαστο κείμενο του SMS.

Ακολουθούν μερικά παραδείγματα μηνυμάτων:

```
ham   What you doing?how are you?

ham   Ok lar... Joking wif u oni...

ham   dun say so early hor... U c already then say...

ham   MY NO. IN LUTON 0125698789 RING ME IF UR AROUND! H*

ham   Siva is in hostel aha:-.

ham   Cos i was out shopping wif darren jus now n i called him 2 ask wat present he wan lor. Then he started guessing who i was wif n he finally guessed darren lor.

spam   FreeMsg: Txt: CALL to No: 86888 & claim your reward of 3 hours talk time to use from your phone now! ubscribe6GBP/ mnth inc 3hrs 16 stop?txtStop

spam   Sunshine Quiz! Win a super Sony DVD recorder if you canname the capital of Australia? Text MQUIZ to 82277. B

spam   URGENT! Your Mobile No 07808726822 was awarded a L2,000 Bonus Caller Prize on 02/09/03! This is our 2nd attempt to contact YOU! Call 0871-872-9758 BOX95QU
```

Για την παρούσα εργασία έχει ήδη πραγματοποιηθεί προεπεξεργασία σε αρκετά σημεία στο αρχικό σύνολο δεδομένων για την αφαίρεση ειδικών χαρακτήρων και ομογενοποίηση (την οποία σε ένα πραγματικό πρόβλημα εκτελεί ο αναλυτής). Στο τελικό αρχείο το οποίο θα χρησιμοποιηθεί στη μελέτη περιλαμβάνονται 5.525 μηνύματα, με 4.806 μη ανεπιθύμητα μηνύματα SMS και 747 ανεπιθύμητα μηνύματα. Επίσης το αρχείο μορφοποιήθηκε με μορφότυπο csv.

Το τελικό αρχείο SMS είναι το παρακάτω.

SMSCollection.csv

Να σημειωθεί ότι για την υλοποίηση της μελέτης χρησιμοποιούνται βιβλιοθήκες ανάλυσης κειμένου και οπτικοποίησης της python, για τις οποίες παρουσιάζεται η βασική φιλοσοφία της προσέγγισης και όχι αναλυτική παρουσίαση. Ο ενδιαφερόμενος αναγνώστης/προγραμματιστής θα πρέπει να επισκεφθεί τη σχετική τεκμηρίωση για λεπτομέρειες (κάτι το οποίο ισχύει για κάθε βιβλιοθήκη της python η οποία παρουσιάζεται στο παρόν υλικό).

Επίσης, θα πρέπει να επισημανθεί ότι ανάλογα με το περιβάλλον υλοποίησης (google colab, anaconda, jupyter lab κλπ) ορισμένες βιβλιοθήκες ενδέχεται να είναι προεγκατεστημένες ή διαθέσιμες σε αυτό. Σε διαφορετική περίπτωση θα πρέπει να εγκατασταθούν (με το πρόγραμμα pip ή άλλο τρόπο).

8.2. Ταξινομητής Naïve Bayes με βασική υλοποίηση python

Σε κάθε εργασία μηχανικής μάθησης είναι σκόπιμο να δαπανάται επαρκής χρόνος για την επισκόπηση των δεδομένων και κατανόηση της δομής, κενών, ατελειών, και λοιπών στοιχείων ώστε να αποφασίζεται στη συνέχεια η πλέον δόκιμη προσέγγιση. Η φάση της διερευνητικής ανάλυσης (exploratory analysis) απαιτεί πολλές φορές δέσμευση σημαντικού ποσοστού χρόνου ως εργασία επί του συνόλου ενός έργου μηχανικής μάθησης, και σε μερικές περιπτώσεις ξεπερνά και το 50%-60%. Είναι σχετικά εύκολο να κατανοηθεί η σημασία της αναγνώρισης των κατάλληλων χαρακτηριστικών, καθώς και της τυποποίησής τους, όσο και της συλλογής τους. Εάν τα δεδομένα δεν δομηθούν σωστά, ή δεν γίνει επιλογή κατάλληλων χαρακτηριστικών το αποτέλεσμα, ειδικά σε πολυδιάστατα δεδομένα δεν θα είναι το αναμενόμενο.

8.2.1. Φόρτωση αρχείου δεδομένων

Το αρχείο δεδομένων δίνεται σε μορφή csv, οπότε αρχικά εισάγεται σε ένα pandas dataframe για διερεύνηση.

Αν χρησιμοποιηθεί το περιβάλλον google colab, το αρχείο csv πρέπει να τοποθετηθεί στο google drive και να προσπελαστεί μέσω του παρακάτω.

```
from google.colab import drive
drive.mount('/content/gdrive')

import pandas as pd

df = pd.read_csv('gdrive/MyDrive/SMSCollection.csv',  encoding='utf-8')

df.columns = ['Label','SMS']
```

```
print(df.shape)

df.head(5)
```

Αν το αρχείο βρίσκεται τοπικά στον σταθμό εργασίας η πρόσβαση γίνεται απευθείας.

```
df = pd.read_csv('SMSCollection.csv',  encoding='utf-8')
```

Οπότε εμφανίζονται οι διαστάσεις,

```
(5524, 2)
```

και οι αρχικές εγγραφές.

```
      Class       Message
0     ham   Go until jurong point, crazy.. Available only ...
1     ham   Ok lar... Joking wif u oni...
2     spam  Free entry in 2 a wkly comp to win FA Cup fina...
3     ham   U dun say so early hor... U c already then say...
4     ham   Nah I don't think he goes to usf, he lives aro...
```

Είναι εμφανής ο διαχωρισμός σε ετικέτα και κείμενο, όπου το κείμενο ενδέχεται να περιέχει οποιονδήποτε χαρακτήρα.

8.2.2. Διερευνητική ανάλυση δεδομένων

Στα αρχικά βήματα δημιουργίας και εκπαίδευσης ενός ταξινομητή, είναι σκόπιμο να διερευνάται το ποσοστό των ταξινομημένων παρατηρήσεων ανά κλάση (η κατανομή τους γενικότερα) ώστε να είναι γνωστό εκ των προτέρων εάν ο ταξινομητής πρόκειται να εκτεθεί σε ομοιογενή δεδομένα ή όχι. Στο συγκεκριμένο σύνολο η αναλογία ανεπιθύμητης αλληλογραφίας προς μη ανεπιθύμητης είναι σημαντική για τον ταξινομητή.

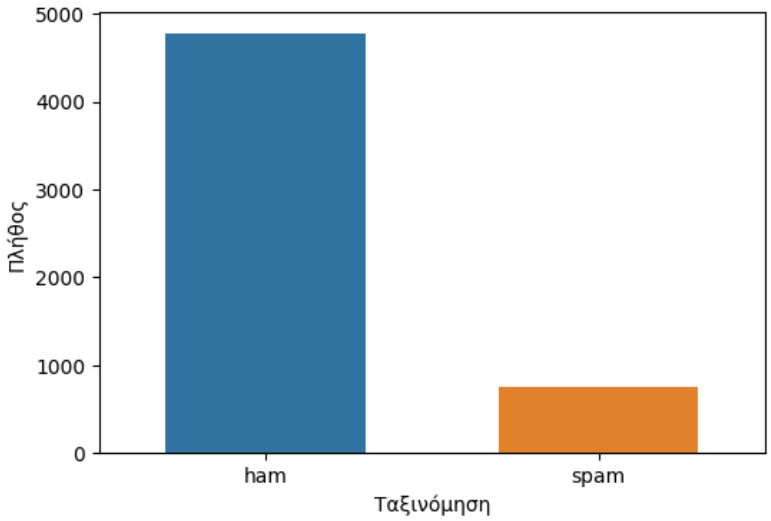
```
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline

plt.figure(figsize=(6,4))
```

```
sns.countplot(x=df.Class, width=0.6, hue=df.Class)

plt.ylabel('Πλήθος')
plt.xlabel('Ταξινόμηση')

df.Class.value_counts(normalize=True)
```

Αυτό δίνει το παρακάτω αποτέλεσμα.

```
      proportion

Class

ham   0.865134

spam  0.134866
```

και το γράφημα



Διάγραμμα 8.1: Ο κανόνας για τον συγκεκριμένο

Το σύνολο δεδομένων όπως είναι εμφανές είναι εξαιρετικά ασύμμετρο, δηλαδή η μεταβλητή στόχος (επιθυμητό ή όχι) δεν έχει ίσες αναλογίες των κλάσεων της. Το συγκεκριμένο σημείο πρέπει να ληφθεί υπόψη κατά την επιλογή υποσυνόλου εκπαίδευσης και ελέγχου, διαφορετικά η εκπαίδευση και οι εκτιμήσεις των παραμέτρων ενδέχεται να οδηγήσουν σε μη αξιόπιστο ταξινομητή.

Υπάρχουν αρκετοί τρόποι επίλυσης του προβλήματος. Μπορεί να χρησιμοποιηθεί η τεχνική Under Sample, δηλαδή να επιλεγούν τυχαία δείγματα της πλειοψηφικής τάξης ίσα με την κατηγορία

μειοψηφίας ή η τεχνική Over Sample, δηλαδή με τη χρήση παρεμβολής, οι βαθμοί κλάσης μειοψηφίας να αυξάνονται σε αριθμό για να ανταγωνιστούν την πλειοψηφική τάξη. Στη συγκεκριμένη περίπτωση δεν χρησιμοποιείται καμία από τις δύο τεχνικές, καθώς ο ταξινομητής είναι ένα πιθανοκρατικό μοντέλο, η πιθανότητα είναι σχετική και ως εκ τούτου θα αναπληρώσει την ανισοκατανομή.

8.2.3. Δημιουργία συνόλων εκπαίδευσης και ελέγχου

Εφόσον ολοκληρωθεί η διερευνητική αξιολόγηση των δεδομένων (συνήθως εξετάζονται περισσότερα στοιχεία για την κατανομή και τις διαστάσεις), στη συνέχεια το σύνολο δεδομένων διαχωρίζεται σε υποσύνολο εκπαίδευσης και υποσύνολο ελέγχου. Συνήθως ο διαμοιρασμός ακολουθεί την αναλογία 80:20, όπου τα δεδομένα εκπαίδευσης αποτελούν το 80% και τα δεδομένα ελέγχου 20% του συνόλου του πληθυσμού. Η επιλογή είναι σκόπιμο να εκτελείται με τυχαίο τρόπο ή στατιστική δειγματοληψία. Διαφορετικά το δείγμα δεν θα αντιπροσωπεύει τον πληθυσμό επαρκώς και η όποια στατιστική συμπερασματολογία χρησιμοποιηθεί δεν θα είναι αξιόπιστη σε μεγάλο βαθμό.

```
# Χρήση της συνάρτηση sample του pandas για δημιουργία τυχαίου
δείγματος
# Με frac=1 ουσιαστικά γίνεται 'τυχαίο ανακάτεμα' των δεδομένων
(shaffling)
# και επιστρέφεται το σύνολο δεδομένων
shaffled = df.sample(frac=1,random_state=1)

# Επιλογή 80% για εκπαίδευση
train_size = round(len(shaffled) * 0.8)
train = shaffled[:train_size].reset_index(drop=True)

# Επιλογή 20% για έλεγχο
test = shaffled[train_size:].reset_index(drop=True)

print(train.shape)
print(test.shape)
```

Τα σύνολα περιέχουν το παρακάτω πλήθος παρατηρήσεων.

```
(4419, 2)
(1105, 2)
```

Εναλλακτικά, η βιβλιοθήκη sklearn παρέχει την ενσωματωμένη συνάρτησης train_test_split() στην sklearn.model_selection. Οπότε η δημιουργία υποσυνόλων δημιουργείται με σχετικά απλό τρόπο.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(df['Class'],
                                                    df['Message'],
                                                    random_state=1,
                                                    test_size=0.2)

print('Πλήθος παρατηρήσεων στο σύνολο: {}'.format(df.shape[0]))
print('Πλήθος παρατηρήσεων στο training set:
{}'.format(X_train.shape[0]))
print('Πλήθος παρατηρήσεων στο test set: {}'.format(X_test.shape[0]))
```

Τα σύνολα όπως και προηγουμένως, αλλά με διαφορετικά τυχαία επιλεγμένες παρατηρήσεις.

```
Πλήθος παρατηρήσεων στο σύνολο: 5524
Πλήθος παρατηρήσεων στο training set: 4419
Πλήθος παρατηρήσεων στο test set: 1105
```

8.2.4. Έλεγχος συνόλων εκπαίδευσης και ελέγχου

Να σημειωθεί ότι κάθε επανάληψη επιλογής δείγματος οδηγεί σε διαφορετικό τυχαίο δείγμα, οπότε είναι σημαντικό το προηγούμενο βήμα του shaffling. Αν για παράδειγμα οι παρατηρήσεις είναι αρχικά ταξινομημένες σε κατηγορίες, θα προκύψουν δείγματα εξαιρετικά μεροληπτικά. Η στατιστική προσέγγιση είναι επομένως πολύ σημαντική.

Ένα δείγμα ενός πληθυσμού πρέπει να είναι αντιπροσωπευτικό του πληθυσμού, διαφορετικά τα αποτελέσματα που λαμβάνονται μπορεί να είναι ασύμμετρα ή μη αξιόπιστα. Επομένως, είναι πολύ σημαντικό να ελεγχθεί σε αυτό το σημείο, πριν προχωρήσει η υλοποίηση του ταξινομητή, αν τα δείγματα είναι αντιπροσωπευτικά του πληθυσμού.

```
train.Class.value_counts(normalize=True)
```

```
proportion

Label

ham   0.865807

spam  0.134193
```

```
test.Class.value_counts(normalize=True)
```

```
proportion

Label

ham   0.862443

spam  0.137557
```

Συγκρίνοντας τις διασπάσεις, η αναλογία των τάξεων για τις δύο διαιρέσεις καθώς και ολόκληρου του πληθυσμού (που βρέθηκε παραπάνω) είναι πολύ κοντά η μία στην άλλη. Αυτό είναι ένα αρκετά καλό μέτρο για να αποδείξει ότι τα διαχωρισμένα δείγματα είναι αμερόληπτοι εκπρόσωποι του πληθυσμού για αυτό το σύνολο δεδομένων.

8.2.5. Κανόνας ταξινόμησης

Στη συνέχεια είναι αναγκαίο να οριστεί η γενική ιδέα της ταξινόμησης, δηλαδή για ποιο λόγο και με βάση ποιο κριτήριο ένα μήνυμα κατατάσσεται στην κλάση των ανεπιθύμητων και αντίστροφα. Μια από τις προσεγγίσεις που ακολουθείται, ως γενική ιδέα για ταξινομητές αυτού του τύπου (στην ουσία πρόκειται για ταξινομητή κειμένου) είναι η θεώρηση του κειμένου ως ένα σύνολο λέξεων (bag of words), όπου το ενδιαφέρον εστιάζει στη συχνότητα εμφάσης. Δηλαδή, για κάθε λέξη στην είσοδο (σε ένα μήνυμα), υπολογίζεται η πιθανότητα να εμφανιστεί αυτή η λέξη είτε σε ανεπιθύμητα μηνύματα είτε σε μη ανεπιθύμητα μηνύματα και στη συνέχεια, αυτές οι πιθανότητες συγκρίνονται. Εάν οι πιθανότητες υποδεικνύουν αθροιστικά ότι η ύπαρξη αυτής της λέξης ή μιας συλλογής λέξεων συνδέεται στενά με ανεπιθύμητα (μη ανεπιθύμητα) μηνύματα, τότε αυτή η είσοδος ταξινομείται ως ανεπιθύμητη (μη ανεπιθύμητη). Με λίγα λόγια το κάθε κείμενο αντιμετωπίζεται ως ένα σύνολο λέξεων (bag of words) και με βάση τις συχνότητές τους καθορίζεται ο κανόνας απόφασης. Η παραπάνω προσέγγιση (αρκετά γνωστή στον κλάδο της Επεξεργασίας Φυσικής Γλώσσας / Natural Language Processing) οδηγεί στην ανάγκη του καθαρισμού του κειμένου ώστε να παραμείνουν μόνο οι όροι (λέξεις) οι οποίοι είναι χρήσιμοι και να αφαιρεθούν ενδιάμεσες λέξεις, σημεία στίξης και οτιδήποτε υπάρχει στα κείμενα και δεν αποτελεί λέξη με νόημα. Αν και αυτό οδηγεί σε μείωση της πληροφορίας των κειμένων, από την άλλη πλευρά οδηγεί και σε μείωση του θορύβου (όπως για παράδειγμα ένας όρος με δύο τελείες, ή τρία θαυμαστικά), διατηρώντας λέξεις με νόημα.

Για παράδειγμα, οι αρχικές εγγραφές περιέχουν κεφαλαία, πεζά, χαρακτήρες ελέγχου κλπ.

```
train.Message.head(5)
```

```
SMS

0     Nite nite pocay wocay luv u more than n e thin...
```

```
1     Awesome, lemme know whenever you're around

2     I'm leaving my house now.

3     LOL .. *grins* .. I'm not babe, but thanks for...

4     Hi, Mobile no. # has added you in their co...
```

8.2.6. Βασική επεξεργασία κειμένου

Εργασίες επεξεργασίας κειμένου μπορούν να πραγματοποιηθούν είτε μαζικά είτε τμηματικά, είτε με απλές συναρτήσεις της python ή με εξειδικευμένες βιβλιοθήκες.

Με συναρτήσεις μπορεί να εκτελεστεί μαζική τροποποίηση σε ένα pandas dataframe. Για παράδειγμα, το '\W' αντιστοιχεί σε οποιονδήποτε χαρακτήρα που δεν είναι γράμμα, ψηφίο ή υπογράμμιση και με τη συνάρτηση replace αντικαθίσταται από χαρακτήρα διαστήματος.

```
train.Message = train['Message'].str.replace('\W',' ')
train.Message = train['Message'].str.lower()
train.Message.head(5)
```

```
Message

0     nite nite pocay wocay luv u more than n e thin...

1     awesome, lemme know whenever you're around

2     i'm leaving my house now.

3     lol .. *grins* .. i'm not babe, but thanks for...

4     hi, mobile no. # has added you in their co...
```

8.2.7. Δημιουργία μήτρας όρων κειμένου

Ο αναλυτής είναι αυτός ο οποίος καθορίσει με βάση το πρόβλημα τις εργασίες καθαρισμού του κειμένου. Τα κανονικοποιημένα (ως προς τους η αποδεκτούς χαρακτήρες κλπ) μηνύματα χρησιμοποιούνται για τη δημιουργία της μήτρας των όρων κειμένου (Term document matrix). Αυτή είναι ένας πίνακας δύο διαστάσεων και αποτελείται από την αντιστοίχιση κάθε λέξης με το μήνυμά της και τον αριθμό των εμφανίσεων στο μήνυμα. Μια μήτρα αποτελείται από τα κειμένων ως δείκτες (σειρές) και κάθε αναγνωριστικό κειμένου (λέξη) ως δείκτες στηλών (στήλη). Επομένως, η

τιμή 'n' (> 0) στη θέση -> (γραμμή, στήλη) σημαίνει ότι η λέξη της ‘στήλης’ εμφανίζεται n φορές στο κείμενο ‘γραμμή’.

Το πρώτο βήμα για τη δημιουργία της μήτρας των όρων κειμένου είναι η δημιουργία του λεξιλογίου των κειμένων. Το λεξιλόγιο είναι ένα σύνολο από όλες τις μοναδικές λέξεις σε όλα τα κείμενα, ώστε στη συνέχεια να μετρηθούν οι συχνότητες εμφάνισής τους.

Το σύνολο εκπαίδευσης αποτελείται από τις παρακάτω λέξεις (bag of words).

```
from itertools import chain

messages = train.Message.str.split()

# η συνάρτηση chain δημιουργεί μια ενιαία λίστα από
# τις επιμέρους λίστες λέξεων κάθε μηνύματος
words = list(chain(*messages))

# μια λίστα με το σύνολο λέξεων όλων των μηνυμάτων
bag_of_words = pd.Series(words).unique()

print(len(bag_of_words))

print(bag_of_words[:30])
```

```
11763

['nite' 'pocay' 'wocay' 'luv' 'u' 'more' 'than' 'n' 'e' 'thing' '4eva' 'i'
 'promise' 'ring' '2morrowxxxx' 'awesome,' 'lemme' 'know' 'whenever'
 "you're" 'around' "i'm" 'leaving' 'my' 'house' 'now.' 'lol' '..'
 '*grins*' 'not']11763
```

Αυτές οι λέξεις έχουν ανακτηθεί από τα μηνύματα. Χρησιμοποιώντας το λεξιλόγιο, προκύπτει η μήτρα όρων (Term Document Matrix). Για ευκολία στη χρήση, έχει κατασκευαστεί ένα λεξικό που αντιπροσωπεύει την ίδια έννοια.

Η παραπάνω μέθοδος δημιούργησε ένα λεξικό με κάθε όρο στο λεξιλόγιο ως κλειδί και για κάθε όρο ένα διάνυσμα που προσδιορίζει σε ποια μηνύματα εμφανίζεται ο όρος και πόσες φορές, ανάλογο με τον Term Document Matrix. Η μέθοδος αυτή είναι ένας τρόπος για να γίνει αυτό από την αρχή και δεν λαμβάνει υπόψη τις λέξεις στοπ (stopw words), δηλαδή λέξεις οι οποίες πρέπει να αποκλειστούν (για παράδειγμα άρθρα, σύνδεσμοι κλπ). Αυτό μπορεί επίσης να επιτευχθεί μέσω της συνάρτησης CountVetorizer της βιβλιοθήκης sklearn.feature_extraction.text. Εδώ εξετάζεται και η αφαίρεση των λέξεων.

Για παράδειγμα χρησιμοποιείται παρακάτω η CountVetorizer, η οποία διαχωρίζει τα κείμενα σε όρους με αλφαριθμητικούς χαρακτήρες (tokens), και χρησιμοποιεί τη βιβλιοθήκη nltk (βιβλιοθήκη επεξεργασίας φυσικής γλώσσας) για αποκλεισμό λέξεων που δεν έχουν νόημα (stop words). Οι λέξεις αυτές είναι συνήθως τυποποιημένες για διάφορες γλώσσες σε βιβλιοθήκες και έτοιμες για χρήση.

```
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from nltk.tokenize import RegexpTokenizer

# Ορίζεται ότι κάθε έγκυρος όρος/λέξη θα περιέχει
# μόνο αριθμούς και γράμματα
token = RegexpTokenizer(r'[a-z0-9A-Z]+')

# δημιουργία αντικείμενου τύπου CountVectorizer
# το οποίο μετατρέπει ένα κείμενο
# σε μήτρα όρων (term document matrix)
vectorizer = CountVectorizer(
    lowercase=True,
    ngram_range=(1,1),
    tokenizer = token.tokenize
)

# δημιουργία του term_document_matrix
term_document_matrix = vectorizer.fit_transform(train['Message'])

# πόσα features δηλαδή στήλες, δηλαδή λέξεις
# έχουν αναγνωριστεί
print(len(vectorizer.get_feature_names_out()))

# ανανέωση του bag_of_words με τα features
bag_of_words = vectorizer.get_feature_names_out()

# δημιουργία λεξικού για την αναπαράσταση του term_document_matrix
term_document_matrix =
dict(zip(vectorizer.get_feature_names_out(),np.transpose(term_document_
matrix.toarray())))
```

Οπότε το λεξιλόγιο (bag_of_words) περιέχει `7763` όρους (δεν είναι ακριβώς λεξεις, καθώς υπάρχουν συνδυασμοί αριθμών και γραμμάτων που δεν είναι λέξεις, όπως 0011 κλπ). Το πρόβλημα με αυτή την προσέγγιση είναι οι συχνότερα εμφανιζόμενοι όροι είναι άρθρα, συνδετικά κλπ, τα οποία δεν έχουν νόημα για την ταξινόμηση.

8.2.8. Αφαίρεση stop words

Για την εκκαθάριση του bag_of_words από τέτοιους όρους, ακολουθεί ένα βήμα με την αφαίρεση προκαθορισμένων όρων (stop words). Κάθε γλώσσα περιλαμβάνει τέτοιους όρους και είναι εφικτή η αφαίρεσή τους αν στον παραπάνω κώδικα προστεθεί η σχετική παράμετρος για την αγγλική γλώσσα.

```
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from nltk.tokenize import RegexpTokenizer
from nltk.corpus import stopwords

# Ορίζεται ότι κάθε έγκυρος όρος/λέξη θα περιέχει
# μόνο αριθμούς και γράμματα
token = RegexpTokenizer(r'[a-z0-9A-Z]+')

# δημιουργία αντικείμενου τύπου CountVectorizer
# το οποίο μετατρέπει ένα κείμενο
# σε μήτρα όρων (term document matrix)
vectorizer = CountVectorizer(
    lowercase=True,
    ngram_range=(1,1),
    tokenizer = token.tokenize,
    stop_words='english'
)

# δημιουργία του term_document_matrix
term_document_matrix = vectorizer.fit_transform(train['Message'])

# πόσα features δηλαδή στήλες, δηλαδή λέξεις
# έχουν αναγνωριστεί
print(len(vectorizer.get_feature_names_out()))

# ανανέωση του bag_of_words με τα features
bag_of_words = vectorizer.get_feature_names_out()

# δημιουργία λεξικού για την αναπαράσταση του term_document_matrix
term_document_matrix =
dict(zip(vectorizer.get_feature_names_out(),np.transpose(term_document_
matrix.toarray())))
```

Οπότε το ενημερωμένο λεξιλόγιο (bag_of_words) περιέχει `7496` όρους.

8.2.9. Οπτικοποίηση όρων με νέφος λέξεων (wordcloud)

Η αξιολόγηση του term_document_matrix που έχει δημιουργηθεί απαιτεί συστηματική εξέταση καθώς ο πίνακας περιέχει χιλιάδες στήλες και γραμμές. Μια σχετικά εύκολη και γρήγορη

αξιολόγηση αποτελεί η οπτικοποίηση με διάφορους τρόπους και διαστάσεις, κάτι το οποίο δίνει σχετικά καλή εποπτική εικόνα. Για δεδομένα αυτού του τύπου το νέφος λέξεων (WordCloud) παρέχει μια αρκετά καλή εποπτική εικόνα των συχνοτήτων εμφάνισης. Διαισθητικά, αναμένεται ότι τα WordCloud θα είναι διαφορετικά για τα ανεπιθύμητα μηνύματα και τα μηνύματα ham (μη ανεπιθύμητα). Αυτό θα έδινε μια αίσθηση του είδους των λέξεων που υπάρχουν και στα δύο είδη μηνυμάτων.

Το WordCloud υπάρχει στο πακέτο wordcloud, με δυνατότητα λήψης και εγκατάστασης με

pip install wordcloud

conda install wordcloud

Η συνάρτηση WordCloud δημιουργεί ένα αντικείμενο εικόνας, το οποίο προβάλλεται χρησιμοποιώντας τη συνάρτηση imshow() του matplotlib. Το πλάτος, το ύψος και άλλες παράμετροι ορίζονται στο αντικείμενο wordcloud. Υπάρχει πρόβλεψη για την αφαίρεση των στοπ λέξεων και στο αντικείμενο. Η παράμετρος stopwords πρέπει να παρέχεται με το αντικείμενο STOPWORDS από το nltk.corpus.

Το πρώτο WordCloud δημιουργείται από τα μηνύματα κειμένου, που χαρακτηρίζονται ως ανεπιθύμητα.

```
from wordcloud import WordCloud,ImageColorGenerator
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

text = ""

# για λέξεις στο train set
for message in train[train.Class == 'spam'].Message:
    words = message.split()
    text = text + " ".join(words) + " "

wordcloud = WordCloud(width=800,height=600,background_color='linen',
                      stopwords=stopwords.words('english')).generate(te
xt)

plt.figure(figsize=(8,6))

plt.imshow(wordcloud,interpolation='bilinear')
plt.axis('off')
```

Από το WordCloud είναι εμφανές ότι λέξεις όπως free, call, now, txt, tone, reply, mobile, text εμφανίζονται συχνά στα ανεπιθύμητα μηνύματα του συνόλου εκπαίδευσης. Αυτό είναι διαισθητικά αναμενόμενο καθώς στην πραγματικότητα αυτού του είδους οι λέξεις εμφανίζονται στα ανεπιθύμητα μηνύματα που λαμβάνονται γενικά.

Στη συνέχεια, δημιουργείται το WordCloud για τα μη ανεπιθύμητα μηνύματα.

```
from wordcloud import WordCloud,ImageColorGenerator
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords

text = ""

# για λέξεις στο train set
for message in train[train.Class == 'ham'].Message:
    words = message.split()
    text = text + " ".join(words) + " "

wordcloud = WordCloud(width=800,height=600,background_color='linen',
                      stopwords=stopwords.words('english')).generate(te
xt)
```

```
plt.figure(figsize=(8,6))

plt.imshow(wordcloud,interpolation='bilinear')
plt.axis('off')
```

Υπάρχουν σαφείς διαφορές στην ποικιλία και το είδος των λέξεων που εμφανίζονται για τις δύο κατηγορίες/ετικέτες. Οι λέξεις που εμφανίζονται στα μηνύματα ham (non-spam) είναι γενικά κοινές λέξεις όπως δείχνει το WordCloud, όπως love, come, know, got, home, need, dear κ.λπ.

Η οπτική αναπαράσταση,, δίνει σε αρκετές περιπτώσεις μια καλή εικόνα ενός πληθυσμού και βοηθά την αποφυγή παρερμηνειών.

8.2.10. Μετατροπή μήτρας σε pandas dataframe

Στη συνέχεια, το λεξικό term_document_matrix μετατρέπεται σε pandas DataFrame και αντικαθιστά το αρχικό σύνολο δεδομένων εκπαίδευσης. Με αυτόν τον τρόπο η αναπαράσταση για κάθε μήνυμα γίνεται εύκολη. Κάθε σειρά είναι ένα μεμονωμένο κείμενο και κάθε στήλη μετά τις στήλες Class και Message αντιπροσωπεύει μια λέξη από το λεξιλόγιο.

```
df_term_document_matrix = pd.DataFrame(term_document_matrix)

new_train = pd.concat([train,df_term_document_matrix],axis=1)

print(new_train.head(3))

new_train.shape
```

```
      Class                                            Message  0  00  000
\

0   ham  nite nite pocay wocay luv u more than n e thin...  0   0    0

1   ham         awesome, lemme know whenever you're around  0   0    0

2   ham                          i'm leaving my house now.  0   0    0


   000pes  008704050406  0089  01223585236  01223585334  ...  zaher  zed  \

0       0             0     0            0            0  ...      0    0

1       0             0     0            0            0  ...      0    0

2       0             0     0            0            0  ...      0    0


   zeros  zhong  zindgi  zoe  zogtorius  zoom  zouk  zyada

0      0      0       0    0          0     0     0      0

1      0      0       0    0          0     0     0      0

2      0      0       0    0          0     0     0      0
```

Ο πίνακας έχει χιλιάδες στήλες, καθώς κάθε στήλη αντιστοιχεί σε ένα όρο, οπότε είναι κατανοητό το μέγεθος του προβλήματος, όταν η κλίμακα αυξάνει. Συνήθως, όπως αναφέρθηκε, πριν ξεκινήσει η κατασκευή μοντέλων, οι stop words αφαιρούνται οριστικά από τις στήλες Message καθώς οι ενδιάμεσες λέξεις (άρθρα, σύνδεσμοι, κλπ) συνήθως υπάρχουν σε όλα τα μηνύματα και δεν είναι σχετικές κατά τη διάκριση μεταξύ ανεπιθύμητων και μη (μη ανεπιθύμητων) μηνυμάτων. Η επεξεργασία του αρχικού σώματος κειμένου είναι επαναληπτική διαδικασία έως ότου καταλήξει στο επιθυμητό σύνολο εκπαίδευσης.

Το σύνολο εκπαίδευσης έχει τα παρακάτω μηνύματα

```
new_train['Class'].value_counts()
```

```
      count
Class
ham   3826
spam  593
```

Ωστόσο, το ενδιαφέρον είναι όχι τα μηνύματα αλλά οι λέξεις οι οποίες εμφανίζονται σε μηνύματα, τα οποία χαρακτηρίζονται ως ham ή spam. Επομένως για τον υπολογισμό των πιθανοτήτων του μοντέλου Naïve Bayes απαιτείται ο υπολογισμός των συχνοτήτων εμφάνισης των λέξεων σε μηνύματα ham ή spam, και με βάση αυτές των πιθανοτήτων.

8.2.11. Υπολογισμός πιθανοτήτων μοντέλου Naïve Bayes

Στη συνέχεια, όπως αναφέρθηκε, δημιουργείται το μοντέλο Naïve Bayes με απλή python για λόγους επίδειξης. Το σύνολο δεδομένων εκπαίδευσης new_train είναι έτοιμο για κατασκευή μοντέλου. Το ζητούμενο είναι να δημιουργηθεί κανόνας απόφασης, ο οποίος θα εντάσσει ένα μήνυμα στην κλάση ham ή την κλάση spam. Η λογική για την κατασκευή του μοντέλου Naïve Bayes είναι η ακόλουθη.

Το μοντέλο θα λάβει ένα μήνυμα ως είσοδο (το οποίο θα αποτελείται από ένα σύνολο λέξεων $\{w_1, w_2,\ldots,w_n\}$) και θα υπολογίσει τις δεσμευμένες πιθανότητες εμφάνισης των δύο κλάσεων, με δεδομένες τις λέξεις οι οποίες εμφανίζονται στο μήνυμα:

$P(\text{spam}|\ w_1, w_2,\ldots,w_n)$

$P(\text{ham}|\ w_1, w_2,\ldots,w_n)$

Συγκρίνοντας τις δύο τιμές, εάν η πρώτη είναι μεγαλύτερη από τη δεύτερη, το μοντέλο ταξινομεί το μήνυμα ως ανεπιθύμητο, ενώ αν είναι μικρότερη, το μοντέλο ταξινομεί το μήνυμα ως μη ανεπιθύμητο. Εάν οι τιμές είναι ίσες, τότε σε αυτό το απλό μοντέλο ταξινομείται ως ham (στην πραγματικότητα, τα μοντέλα είναι περισσότερο εξελιγμένα με περισσότερες παραμέτρους). Ο κανόνας βασίζεται στον παρακάτω τύπο

$Y_{pred} = \text{argmax}_y\ (P(y)P(w_1|y)P(w_2|y)\ldots P(w_n|y))$

Για τον υπολογισμό των πιθανοτήτων του αλγόριθμου Naive Bayes απαιτείται ο υπολογισμός των ακόλουθων όρων:

- N(spam) - ο συνολικός αριθμός λέξεων στα ανεπιθύμητα μηνύματα
- N(ham) – ο συνολικός αριθμός λέξεων σε μη ανεπιθύμητα μηνύματα
- N(bag_of_words) – ο συνολικός αριθμός λέξεων στο λεξιλόγιο
- P(spam) – η πιθανότητα εμφάνισης μηνύματος ως ανεπιθύμητης αλληλογραφίας
- P(ham) – η πιθανότητα εμφάνισης μηνύματος ως μη ανεπιθύμητης αλληλογραφίας

και ο εκ των προτέρων υπολογισμός τους εξοικονομεί χρόνο κατά την εκπαίδευση του μοντέλου.

Οι συχνότητες είναι

```
# δημιουργία λίστας με τις λέξεις και καταμέτρηση
number_of_words_in_spam = len(list(chain(*messages[new_train.Class == 
'spam'])))
number_of_words_in_ham = len(list(chain(*messages[new_train.Class == 
'ham'])))

number_of_words_in_bug_of_words = len(bag_of_words)

print('number_of_words_in_spam={0},number_of_words_in_ham={1},
number_of_words_in_bug_of_words={2}'.format(
    number_of_words_in_spam,
    number_of_words_in_ham,
    number_of_words_in_bug_of_words))
```

```
number_of_words_in_spam=14184,number_of_words_in_ham=54846,
number_of_words_in_bug_of_words=7496
```

Οι πιθανότητες είναι

```
total_number_of_ham = new_train['Class'].value_counts().iloc[0]
total_number_of_spam = new_train['Class'].value_counts().iloc[1]

total_number_of_messages = len(new_train)

Prob_spam_message = total_number_of_spam / total_number_of_messages
Prob_ham_message = total_number_of_ham / total_number_of_messages


print('Prob_ham={0}, Prob_spam={1}'.format(Prob_ham_message,
Prob_spam_message))
```

```
Prob_ham=0.8658067436071509, Prob_spam=0.13419325639284907
```

Οι παράμετροι

$P(w_i|spam)$

$P(w_i|ham)$

(το w_i σημαίνει κάθε λέξη στο λεξιλόγιο) υπολογίζουν την πιθανότητα να εμφανιστεί μια λέξη στο κείμενο με την ετικέτα του μηνύματος. Αυτοί οι υπολογισμοί είναι στατικοί όπως αυτοί που έγιναν προηγουμένως. Ως εκ τούτου, ο υπολογισμός τους μια φορά είναι αρκετός.

Διατηρούνται δύο ξεχωριστά λεξικά, για spam και ham (non-spam), αντιστοιχίζοντας τη λέξη με την πιθανότητα να εμφανιστεί στην αντίστοιχη ετικέτα.

```
# δημιουργία και αρχικοποίση λεξικών
# για λέξεις και πιθανότητα εμφάνισης σε ham και spam
# με αρχική πιθανότητα μηδέν
Prob_of_word_given_spam = {word: 0 for word in bag_of_words}
Prob_of_word_given_ham = {word: 0 for word in bag_of_words}

Spam_messages = new_train[train.Class == 'spam']
Ham_messages = new_train[train.Class == 'ham']

# για κάθε λέξη στο bag_of_words
# υπολογισμός της πιθανότητας
# P(λέξη | spam μήνυμα)
# P(λέξη | ham μήνυμα)
for word in bag_of_words:

    Number_of_word_in_spam = Spam_messages[word].sum()
    Number_of_word_in_ham = Ham_messages[word].sum()

    Prob_of_word_given_spam[word] = Number_of_word_in_spam /
number_of_words_in_spam
    Prob_of_word_given_ham[word] = Number_of_word_in_ham /
number_of_words_in_ham
```

Για παράδειγμα

```
Prob_of_word_given_spam
```

```
'annie': 0.0,
 'anniversary': 0.0,
 'annoncement': 7.050197405527355e-05,
 'announced': 0.0,
 'announcement': 0.00021150592216582064,
```

```
 'annoying': 0.0,
 'anonymous': 7.050197405527355e-05,
 'anot': 0.0,
 'ans': 0.0003525098702763677,
 'ansr': 0.00021150592216582064,
 'answer': 0.0003525098702763677,
 'answered': 0.0,
 'answerin': 0.0,
 'answering': 0.0,
```

Και

```
Prob_of_word_given_ham
```

```
anjie': 1.8232870218429784e-05,
 'anjola': 1.8232870218429784e-05,
 'anna': 1.8232870218429784e-05,
 'annie': 1.8232870218429784e-05,
 'anniversary': 1.8232870218429784e-05,
 'annoncement': 0.0,
 'announced': 1.8232870218429784e-05,
 'announcement': 0.0,
 'annoying': 3.646574043685957e-05,
 'anonymous': 0.0,
 'anot': 1.8232870218429784e-05,
 'ans': 5.469861065528935e-05,
 'ansr': 0.0,
```

8.2.12. Ταξινόμηση με το μοντέλο

Η επόμενη φάση είναι η κατασκευή του μοντέλου Naïve Bayes. Το μοντέλο θα λάβει ένα μήνυμα και θα υπολογίσει τις πιθανότητες:

$P(spam| w_1, w_2,\ldots,w_n)$

$P(ham| w_1, w_2,\ldots,w_n)$

Συγκρίνοντας τα δύο, εάν το πρώτο είναι μεγαλύτερο από το δεύτερο, το μοντέλο ταξινομεί το μήνυμα ως ανεπιθύμητο, εάν είναι ίσα τότε το μοντέλο απαιτεί ανθρώπινη βοήθεια για να τα ταξινομήσει καλύτερα, διαφορετικά το μήνυμα ταξινομείται ως ham (non-spam).

Η παρακάτω συνάρτηση δέχεται ένα μήνυμα εισόδου και το ταξινομεί.

```
import re

def spam_or_ham(message):

    # καθαρισμός μηνύματος από μη αλφαρηθμητικούς χαρακτήρες
    message = re.sub('\W',' ',message)
```

```
    message = message.lower()
    message = message.split()

    spam = 1
    ham = 1

    for word in message:

      # αναζήτηση στα λεξικά
        if word in Prob_of_word_given_spam.keys():
          # πολλαπλασιασμός πιθανοτήτων
          # ανεξάρτητα ενδοεχόμενα
          # οι εμφανίσεις τη λέξης
            spam *= Prob_of_word_given_spam[word]
        if word in Prob_of_word_given_ham.keys():
            ham *= Prob_of_word_given_ham[word]

    Prob_of_spam_given_message = Prob_spam_message * spam
    Prob_of_ham_given_message = Prob_ham_message * ham

    print("P(spam|message) = ",Prob_of_spam_given_message)
    print("P(ham|message) = ",Prob_of_ham_given_message)

    if Prob_of_spam_given_message > Prob_of_ham_given_message:
            classification = 'Κλάση: spam'
    elif Prob_of_spam_given_message < Prob_of_ham_given_message:
            classification= 'Κλάση: ham'
    else:
            classification = 'Αδύνατη ταξινόμηση'

    return classification
```

Η ταξινόμηση μηνυμάτων είναι απλή,

```
spam_or_ham('I want to see you tomorrow')
```

```
P(spam|message) =  1.1739400370740495e-07
P(ham|message) =  2.5818053031061797e-06

Κλάση: ham
```

```
spam_or_ham('I can lend you some money -- let me know -- send me your
code ')
```

```
P(spam|message) =  1.1543497503208857e-16
P(ham|message) =  9.122865119190628e-17
```

```
Κλάση: spam
```

8.2.13. Αξιολόγση του μοντέλου

Το μοντέλο φαίνεται σχετικά καλό από τις προκαταρκτικές δοκιμές. Ως τελευταίο μέρος του έργου, το μοντέλο πρέπει να δοκιμαστεί στο σύνολο δοκιμών και να εξαχθούν συμπεράσματα από τις προβλέψεις. Οι προβλέψεις αποθηκεύονται σε μια νέα στήλη, για να συγκριθούν με την πραγματική στήλη Class.

```
test['predicted'] = test.Class.apply(spam_or_ham)
```

```
      Class       Message     predicted

0     ham   Later i guess. I needa do mcat study too.     ham

1     ham   Christmas is An occasion that is Celebrated as...   ham

2     ham   awesome, how do I deal with the gate? Charles ...   ham

3     ham   All sounds good. Fingers . Makes it difficult ...   ham

4     ham   Nt joking seriously i told   ham
```

Και

```
test.predicted.value_counts(normalize=True)
```

```
proportion

                  predicted

ham               0.827149

spam              0.103167

Αδύνατη ταξινόμηση     0.069683
```

Από το δείγμα της δοκιμής, το μοντέλο ταξινόμησε περίπου το 82% ως μη ανεπιθύμητο και περίπου το 10% ως ανεπιθύμητο, ενώ για 6% δεν ήταν δυνατή η ταξινόμηση. Το μοντέλο δείχνει να τα πήγε καλά καθώς αυτές οι αναλογίες είναι ανάλογες με την αναλογία κλάσεων του δείγματος. Για καλύτερη αξιολόγηση υπολογίζεται η ακρίβεια.

```
accuracy = sum(test.Class == test.predicted)/len(test)
accuracy
```

```
0.9176470588235294
```

Το μοντέλο έχει επιτύχει υψηλή ακρίβεια περίπου 92% στο σετ ελέγχου. Δείχνει αρκετά καλό αλλά έχει σχέση με το λεξιλόγιο του πληθυσμού και την προετοιμασία των δεδομένων.

Η ακρίβεια δίνει μια γενική ιδέα για το μοντέλο, αλλά όχι την συνολική εικόνα. Ο πίνακας Confusion matrix δίνει πληροφορίες για τα Αληθινά θετικά και τα Λάθος αρνητικά (σωστές προβλέψεις) και τα Σωστά αρνητικά και Ψευδώς θετικά (λανθασμένες προβλέψεις) του μοντέλου. Με αυτά, σημαντικές μετρήσεις για αναφορά είναι η Ακρίβεια, η Ανάκληση και η βαθμολογία F1.

Η λειτουργική μονάδα sklearn.metrics παρέχει την ενσωματωμένη συνάρτηση classification_report η οποία δημιουργείται με βάση τις αληθινές ταξινομήσεις και τις προβλεπόμενες ταξινομήσεις.

```
from sklearn.metrics import classification_report

print(classification_report(test.predicted,test.Label))
```

```
                           precision    recall  f1-score   support

                      ham       0.95      0.99      0.97       914
                     spam       0.70      0.94      0.80       114
Αδύνατη ταξινόμηση       0.00      0.00      0.00        77

                 accuracy                           0.92      1105
                macro avg       0.55      0.64      0.59      1105
             weighted avg       0.86      0.92      0.89      1105
```

Το μοντέλο καταφέρνει να έχει ακρίβεια 95% στο σετ ελέγχου. Και εξίσου καλή ακρίβεια και ανάκληση.

Σε αυτό το σημείο η βασική εργασία έχει ολοκληρωθεί. Το επόμενο βήμα είναι η αξιολόγηση του μοντέλου με διάφορα μηνύματα και η αναζήτηση βελτιώσεων.

8.2.14. Έλεγχος λανθασμένων ταξινομήσεων

Καλή πρακτική σε προβλήματα κειμένου είναι να εξετάζονται τα εσφαλμένα ταξινομημένα δεδομένα, καθώς θα δώσουν ένδειξη πιθανόν αστοχιών που μπορεί να οφείλονται σε λανθασμένα δεδομένα ή σε μη καθαρισμένο κείμενο σε αυτήν την περίπτωση. Μια γρήγορη εικόνα μπορεί να αποδοθεί με τα wordcloud για τα λάθος ταξινομημένα μηνύματα spam και ham.

Για τον έλεγχο των εσφαλμένα ταξινομημένων μηνυμάτων δημιουργείται ένα νέο σύνολο.

```
def clean_message(message):

    message = re.sub('\W',' ',message)
    message = message.lower()

    return message

test.Class = test.Message.apply(clean_message)
df_missclassified = test[test.Class != test.predicted]

df_missclassified
```

```
      Class  Message      predicted

      9      ham    by march ending i should be ready but will c...      Αδύνατη
ταξινόμηση

      32     ham    sure thing big man i have hockey elections at...     Αδύνατη
ταξινόμηση

      55     ham    what is your account number      spam

      56     spam   want 2 get laid tonight want real dogging loc...     Αδύνατη
ταξινόμηση

      69     spam   ur balance is now 600 next question complet...       Αδύνατη
ταξινόμηση

      ...    ...    ...    ...

      1068   spam   santa calling would your little ones like a c...     ham

      1078   spam   collect your valentine s weekend to paris inc ...    Αδύνατη
ταξινόμηση

      1081   spam   you are now unsubscribed all services get ton...     Αδύνατη
ταξινόμηση
```

Παρακάτω δημιουργούνται WordClouds και για τις δύο κατηγορίες, μόνο για τα κείμενα που δεν ταξινομήθηκαν σωστά για την ανακάλυψη τυχόν σημαντικών στοιχείων. Το πρώτο αφορά τα ham (non-spam) μηνύματα που ταξινομήθηκαν ως spam.

```
text = ""

for message in df_missclassified[df_missclassified.Class ==
'ham'].Message:
    words = message.split()
    text = text + " ".join(words) + " "

wordcloud =
WordCloud(width=1200,height=800,stopwords=stopwords.words('english'),ba
ckground_color='wheat').generate(text)

plt.figure(figsize=(12,8))
plt.imshow(wordcloud,interpolation='bilinear')
plt.axis('off')
```

Το WordCloud εμφανίζει τις πιο συχνές λέξεις για τα εσφαλμένα (μη ανεπιθύμητα) μηνύματα που έχουν ταξινομηθεί. Οι λέξεις call, phone, connected, messages, executive, mobile κ.λπ. είναι παρόμοιες με τις λέξεις που βρίσκονται συνήθως στα ανεπιθύμητα μηνύματα. Μια ερμηνεία είναι

ότι το μοντέλο δεν ταξινόμησε όλα εκείνα τα ham (μη ανεπιθύμητα) μηνύματα που είχαν πιο επίσημο τόνο ή μιμούνταν κατά κάποιο τρόπο τα ανεπιθύμητα μηνύματα.

Η επόμενη γραφική παράσταση αφορά τα μηνύματα spam που ταξινομήθηκαν λανθασμένα ως ham.

```
text = ""

for message in df_missclassified[df_missclassified.Class ==
'spam'].Message:
    words = message.split()
    text = text + " ".join(words) + " "


wordcloud =
WordCloud(width=1200,height=800,stopwords=stopwords.words('english'),ba
ckground_color='wheat').generate(text)

plt.figure(figsize=(12,8))
plt.imshow(wordcloud,interpolation='bilinear')
plt.axis('off')
```

Η εικόνα δείχνει πολλές λέξεις που δεν αποτελούν μέρος του λεξιλογίου. Λέξεις όπως – 150p, msg κ.λπ. Λόγω του ότι αυτές οι λέξεις δεν εμφανίζονται στο λεξιλόγιο, οι λέξεις ham υπερίσχυαν των

λέξεων ανεπιθύμητης αλληλογραφίας. Το μήνυμα είναι ουσιαστικά ανεπιθύμητο, καθώς οι λέξεις που θα μπορούσαν πράγματι να συμβάλουν στην ταξινόμησή του ως τέτοιο δεν υπάρχουν στο λεξιλόγιο και οι λέξεις που υπάρχουν στο μήνυμα και ήταν μέρος του λεξιλογίου το ταξινόμησαν ως μήνυμα ham (μη ανεπιθύμητο).

Τα εσφαλμένα ταξινομημένα μηνύματα εκθέτουν το πρόβλημα αυτού του απλού μοντέλου (το οποίο ωστόσο αποτελεί περιορισμό του μοντέλου επεξεργασίας φυσικής γλώσσας που επιλέχθηκε, δηλαδή το bag of words). Το μοντέλο αγνοεί το περιεχόμενο που κρύβεται στα μηνύματα. Εξετάζει μόνο τη λέξη μεμονωμένα και ελέγχει εάν αυτή η λέξη μπορεί να είναι ανεπιθύμητη ή όχι. Ενώ παρατηρείται ότι οι λέξεις μεμονωμένα δεν μεταφέρουν όλη την ιστορία.

Ως γενικό συμπέρασμα, δημιουργήθηκε ένα επιτυχημένο μοντέλο για φίλτρο ανεπιθύμητης αλληλογραφίας χρησιμοποιώντας τον αλγόριθμο Naive Bayes. Το μοντέλο θεωρεί την ανεξαρτησία μεταξύ των μηνυμάτων και των λέξεων στα μηνύματα, αγνοώντας το πλαίσιο του μηνύματος. Με βάση αυτή την απλή υπόθεση, το μοντέλο απέδωσε πολύ καλά για αυτό το σύνολο δεδομένων. Το μοντέλο επιτυγχάνει καλή βαθμολογία ακρίβειας και ανάκλησης. Ωστόσο, είναι σημαντικό να σημειωθεί ότι τα ανεπιθύμητα μηνύματα έχουν μικρότερη ανάκληση. Αυτό μπορεί να οφείλεται στο γεγονός ότι υπάρχει μια ανισορροπία στις κατηγορίες της μεταβλητής στόχου. Συνολικά το μοντέλο έχει μάθει καλά τα δεδομένα εκπαίδευσης και έχει καλές επιδόσεις στο σύνολο δοκιμών για τα μηνύματα κειμένου στο σύνολο δεδομένων που χρησιμοποιείται.

8.3. Ταξινομητής Naïve Bayes με χρήση της κλάσης sklearn Gaussian Naive Bayes

Το πρόβλημα ταξινόμησης μηνυμάτων το οποίο παρουσιάστηκε στην προηγούμενη ενότητα, θα επιλυθεί στην παρούσα ενότητα με τη βοήθεια της κλάσης GaussianNB της βιβλιοθήκης sklearn.naive_bayes, η οποία υλοποιεί τον αλγόριθμο Naïve Bayes με την υπόθεση ότι τα δεδομένα ακολουθούν την κανονική κατανομή (Gauss). Η προσέγγιση ως προς το χειρισμό του κειμένου είναι ίδια, δηλαδή bag of words.

8.3.1. Φόρτωση αρχείου δεδομένων

Αρχικά φορτώνεται το αρχείο σε ένα αντικείμενο pandas dataframe.

```
from google.colab import drive
drive.mount('/content/gdrive')

import pandas as pd
```

```
df =
pd.read_csv('gdrive/MyDrive/SMSSpamCollection.csv',  encoding='utf-8')

df.columns = ['Class','Message']

print(df.shape)

df.head(5)
```

Οι ετικέτες των δύο κλάσεων αντικαθίστανται μα δυαδικές τιμές.

```
df['Class'] = df.Class.map({'ham':0,'spam':1})
```

8.3.2. Δημιουργία συνόλων εκπαίδευσης και ελέγχου

Εκτελείται διαχωρισμός σε σύνολο εκπαίδευσης και ελέγχου σε αναλογία 80:20.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(df['Message'],
                                                    df['Class'],
                                                    random_state=1,
                                                    test_size=0.2)

print('Πλήθος παρατηρήσεων στο σύνολο: {}'.format(df.shape[0]))
print('Πλήθος παρατηρήσεων στο training set:
{}'.format(X_train.shape[0]))
print('Πλήθος παρατηρήσεων στο test set: {}'.format(X_test.shape[0]))
```

```
Πλήθος παρατηρήσεων στο σύνολο: 5524
Πλήθος παρατηρήσεων στο training set: 4419
Πλήθος παρατηρήσεων στο test set: 1105
```

8.3.3. Δημιουργία μήτρας όρων κειμένου

Επειδή το σώμα του κειμένου αποτελείται από λέξεις δημιουργείται ο πίνακας όρων κειμένου.

```
# Αρχικοποίηση της κλάσης CountVectorizer η οποία μετατρέπει ένα
κείμενο σε μήτρα όρων με συχνότητες
from sklearn.feature_extraction.text import CountVectorizer
from nltk.tokenize import RegexpTokenizer


# Ορίζεται ότι κάθε έγκυρος όρος/λέξη θα περιέχει
# μόνο γράμματα
token = RegexpTokenizer(r'[a-zA-Z]{3,}')
```

```
# δημιουργία αντικείμενου τύπου CountVectorizer
# το οποίο μετατρέπει ένα κείμενο
# σε μήτρα όρων (term document matrix)
vectorizer = CountVectorizer(
    lowercase=True,
    ngram_range=(1,1),
    tokenizer = token.tokenize
)

# Προσαρμογή του συνόλου εκπαίδευσης και δημιουργία της μήτρας όρων
κειμένου
training_data = vectorizer.fit_transform(X_train)

# Προσαρμογή του συνόλου εκπαίδευσης και δημιουργία της μήτρας όρων
κειμένου
testing_data = vectorizer.transform(X_test)

#print(testing_data)
```

8.3.4. Εκπαίδευση μοντέλου

Εκπαίδευση του μοντέλου Gaussian NB.

```
from sklearn.naive_bayes import MultinomialNB, GaussianNB

naive_bayes = GaussianNB()

naive_bayes.fit(training_data.toarray(), y_train)
```

Δημιουργία των προβλέψεων για το σύνολο ελέγχου, ώστε να ελεγχθεί η απόδοση και οι υπόλοιπες μετρικές.

8.3.5. Αξιολόγηση μοντέλου

```
predictions = naive_bayes.predict(testing_data.toarray())
```

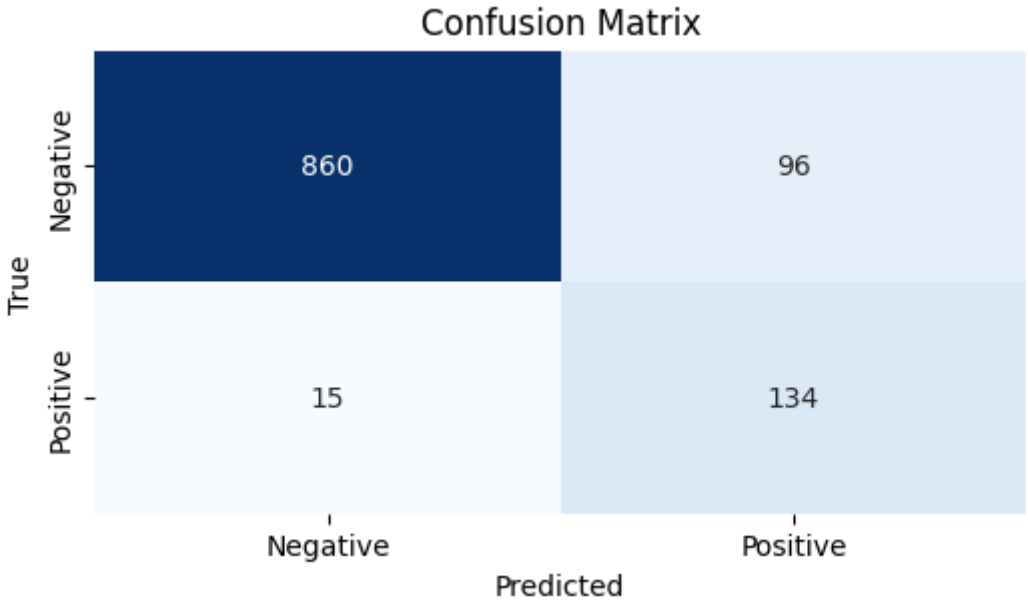
Υπολογισμός των μετρικών απόδοσης.

```
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score

print('Accuracy score: ', format(accuracy_score(y_test,predictions)))
print('Precision score: ', format(precision_score(y_test,predictions)))
print('Recall score: ', format(recall_score(y_test,predictions)))
print('F1 score: ', format(f1_score(y_test,predictions)))
```

```
Accuracy score:  0.8995475113122172
Precision score:  0.5826086956521739
Recall score:  0.8993288590604027
F1 score:  0.7071240105540897
```

Δημιουργία Confusion matrix σε διάγραμμα 8.2.

```
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Confusion matrix
cm = confusion_matrix(y_test, predictions)

# Confusion matrix σε διάγραμμα
plt.figure(figsize=(6, 3))

sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", cbar=False,
            xticklabels=["Negative", "Positive"],
            yticklabels=["Negative", "Positive"])
plt.xlabel("Predicted")
plt.ylabel("True")
plt.title("Confusion Matrix")

plt.show()
```



Διάγραμμα 8.2: Πίνακας σύγχυσης

Η γενική εκτίμηση είναι ότι το μοντέλο είναι σχετικά ικανοποιητικό, αλλά η απόδοσή του είναι χαμηλότερη από το απλοϊκό μοντέλο, κοντά στο 92%.

8.3.6. Ταξινόμηση νέου μηνύματος

```
import numpy as np

sms = vectorizer.transform(np.array(['I want to see you tomorrow']))

prediction = naive_bayes.predict(sms.toarray())

print(prediction)
```

Το μήνυμα ταξινομείται ως spam.

```
[1]
```

Και το μήνυμα

```
import numpy as np

sms = vectorizer.transform(np.array(['I can lend you some money -- let
me know -- send me your code']))

prediction = naive_bayes.predict(sms.toarray())

print(prediction)
```

```
[1]
```

Το μήνυμα ταξινομείται ως spam.

Παρατηρείται ότι εκτελείται διαφορετική ταξινόμηση σε σχέση με τον απλοϊκό ταξινομητή.

8.3.7. Βελτιώσεις

Δίνεται ως άσκηση στον αναγνώστη ως άσκηση ο πειραματισμός στο μοντέλο GaussianNB με στόχο τη βελτίωση της απόδοσής του. Ενέργειες που μπορούν να εξεταστούν είναι

- η μεταβολή της δειγματοληψίας σε ποσοστά και παραμέτρους,
- μεταβολή των παραμέτρων της κλάσης GaussianNB,
- μεταβολή της προσέγγισης bag of words, με μεταβολή των παραμέτρων της κλάσης CountVectorizer με βάση την τεκμηρίωσή της.

8.4. Ταξινομητής Naïve Bayes με χρήση της κλάσης sklearn MultinomialNB Naive Bayes

Το πρόβλημα ταξινόμησης μηνυμάτων το οποίο παρουσιάστηκε στην προηγούμενη ενότητα, θα επιλυθεί στην παρούσα ενότητα με τη βοήθεια της κλάσης MultinomialNB της βιβλιοθήκης sklearn.naive_bayes, η οποία υλοποιεί τον αλγόριθμο Multinomial Naïve Bayes. Η προσέγγιση ως προς το χειρισμό του κειμένου είναι ίδια, δηλαδή bag of words, ωστόσο υπάρχουν διαφοροποιήσεις στον τύπο των δεδομένων εισόδου.

8.4.1. Φόρτωση αρχείου δεδομένων

Αρχικά φορτώνεται το αρχείο σε ένα αντικείμενο pandas dataframe.

```
from google.colab import drive
drive.mount('/content/gdrive')

import pandas as pd

df =
pd.read_csv('gdrive/MyDrive/SMSSpamCollection.csv',  encoding='utf-8')

df.columns = ['Class','Message']

print(df.shape)

df.head(5)
```

Οι ετικέτες των δύο κλάσεων αντικαθίστανται μα δυαδικές τιμές.

```
df['Class'] = df.Class.map({'ham':0,'spam':1})
```

8.4.2. Δημιουργία συνόλων εκπαίδευσης και ελέγχου

Εκτελείται διαχωρισμός σε σύνολο εκπαίδευσης και ελέγχου σε αναλογία 80:20.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(df['Message'],
                                                    df['Class'],
                                                    random_state=1,
                                                    test_size=0.2)
```

```
print('Πλήθος παρατηρήσεων στο σύνολο: {}'.format(df.shape[0]))
print('Πλήθος παρατηρήσεων στο training set:
{}'.format(X_train.shape[0]))
print('Πλήθος παρατηρήσεων στο test set: {}'.format(X_test.shape[0]))
```

```
Number of rows in the total set: 5524
Number of rows in the training set: 4143
Number of rows in the test set: 1381
```

8.4.3. Δημιουργία μήτρας όρων κειμένου

Επειδή το σώμα του κειμένου αποτελείται από λέξεις δημιουργείται ο πίνακας όρων κειμένου.

```
# Αρχικοποίηση της κλάσης CountVectorizer η οποία μετατρέπει ένα
κείμενο σε μήτρα όρων με συχνότητες
from sklearn.feature_extraction.text import CountVectorizer
from nltk.tokenize import RegexpTokenizer


# Ορίζεται ότι κάθε έγκυρος όρος/λέξη θα περιέχει
# μόνο γράμματα
token = RegexpTokenizer(r'[a-zA-Z]{3,}')

# δημιουργία αντικείμενου τύπου CountVectorizer
# το οποίο μετατρέπει ένα κείμενο
# σε μήτρα όρων (term document matrix)
vectorizer = CountVectorizer(
    lowercase=True,
    ngram_range=(1,1),
    tokenizer = token.tokenize
)

# Προσαρμογή του συνόλου εκπαίδευσης και δημιουργία της μήτρας όρων
κειμένου
training_data = vectorizer.fit_transform(X_train)

# Προσαρμογή του συνόλου εκπαίδευσης και δημιουργία της μήτρας όρων
κειμένου
testing_data = vectorizer.transform(X_test)
```

8.4.4. Εκπαίδευση μοντέλου

Εκπαίδευση του μοντέλου Multinomial NB.

```
from sklearn.naive_bayes import MultinomialNB

naive_bayes = MultinomialNB()
```

```
naive_bayes.fit(training_data,y_train)
```

8.4.5. Αξιολόγηση μοντέλου

Δημιουργία των προβλέψεων για το σύνολο ελέγχου, ώστε να ελεγχθεί η απόδοση και οι υπόλοιπες μετρικές.

```
predictions = naive_bayes.predict(testing_data)
```

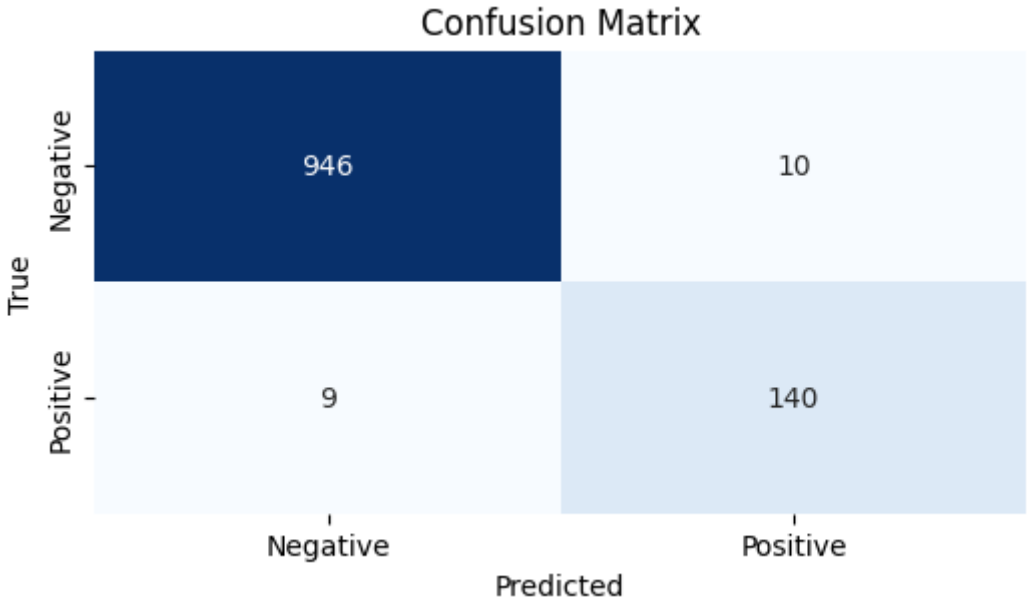
Υπολογισμός των μετρικών απόδοσης.

```
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score

print('Accuracy score: ', format(accuracy_score(y_test,predictions)))
print('Precision score: ', format(precision_score(y_test,predictions)))
print('Recall score: ', format(recall_score(y_test,predictions)))
print('F1 score: ', format(f1_score(y_test,predictions)))
```

```
Accuracy score:  0.9862418537291817
Precision score:  0.9662921348314607
Recall score:  0.9297297297297298
F1 score:  0.9476584022038568
```

Δημιουργία Confusion matrix σε διάγραμμα 8.3.

```
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Confusion matrix
cm = confusion_matrix(y_test, predictions)

# Confusion matrix σε διάγραμμα
plt.figure(figsize=(6, 3))

sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", cbar=False,
            xticklabels=["Negative", "Positive"],
            yticklabels=["Negative", "Positive"])
plt.xlabel("Predicted")
plt.ylabel("True")
plt.title("Confusion Matrix")

plt.show()
```



Διάγραμμα 8.3: Πίνακας σύγχυσης

Η γενική εκτίμηση είναι ότι το μοντέλο είναι πάρα πολύ ικανοποιητικό, με απόδοση υψηλότερη από το απλοϊκό μοντέλο και το μοντέλο Gaussian Naive Bayes, κοντά στο 99%.

8.4.6. Ταξινόμηση νέου μηνύματος

```
import numpy as np

sms = vectorizer.transform(np.array(['I want to see you tomorrow']))

prediction = naive_bayes.predict(sms.toarray())

print(prediction)
```

```
[0]
```

Το μήνυμα ταξινομείται ως ham.

Ταξινόμηση ενός νέου μηνύματος.

```
import numpy as np

sms = vectorizer.transform(np.array(['I can lend you some money -- let
me know -- send me your code']))
```

```
prediction = naive_bayes.predict(sms.toarray())

print(prediction)
```

```
[0]
```

Το μήνυμα ταξινομείται ως ham.

Παρατηρείται ότι εκτελείται διαφορετική ταξινόμηση σε σχέση με τον απλοϊκό ταξινομητή. Αυτό είναι αναμενόμενο καθώς η κάθε επανάληψη δημιουργεί διαφορετικά σύνολα εκπαίδευσης και ελέγχου, όπως και λεξιλόγια.

8.4.7. Βελτιώσεις

Ενέργειες που μπορούν να εξεταστούν είναι

- η μεταβολή της δειγματοληψίας σε ποσοστά και παραμέτρους,
- μεταβολή των παραμέτρων,
- μεταβολή της προσέγγισης bag of words, με μεταβολή των παραμέτρων της κλάσης CountVectorizer με βάση την τεκμηρίωσή της.

ΚΕΦΑΛΑΙΟ 9: Δέντρα αποφάσεων

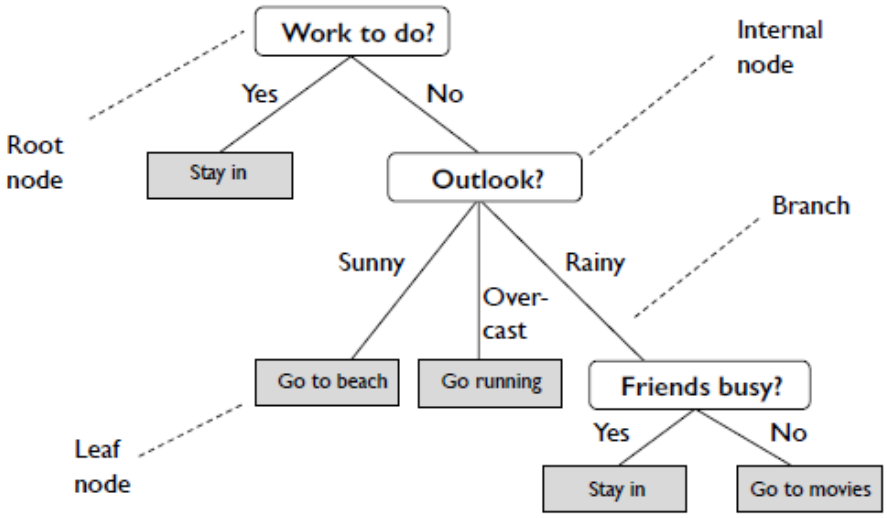
9.1. Εισαγωγή

Τα Δέντρα Απόφασης (ΔΑ) αναπαριστούν ένα μοντέλο πρόβλεψης το οποίο χτίζεται με μια επαναληπτική διαδικασία μέσα από μια σειρά δυαδικών αποφάσεων τύπου ΝΑΙ/ΟΧΙ, μεγαλύτερο/μικρότερο, κλπ.

Τα ΔΑ μπορούν να αναπαραστήσουν οποιαδήποτε δυαδική συνάρτηση και ο χώρος υπόθεσης που αναζητείται είναι ολόκληρος ο χώρος των δυαδικών συναρτήσεων. Το μέγεθος του χώρου της υπόθεσης καθορίζεται από το σύνολο των δεδομένων.

Θεωρώντας ότι υπάρχουν μόνο δυαδικά χαρακτηριστικά σε κάθε κόμβο, υπάρχουν 2^{μ} πιθανές διασπάσεις που πρέπει να αξιολογηθούν δεδομένου ότι το σύνολο δεδομένων έχει μ χαρακτηριστικά.

Οι αλγόριθμοι των ΔΑ αναζητούν τον χώρο των υποθέσεων ψάχνοντας όλα τα πιθανά δέντρα. Μια εξαντλητική αναζήτηση δεν είναι εφικτή λόγω της εκθετικής φύσης του προβλήματος. Δηλαδή, αν υποθέσουμε ότι έχουμε μ δυαδικά χαρακτηριστικά, τότε υπάρχουν 2^{μ} πιθανοί συνδυασμοί χαρακτηριστικών. Τότε, αν θεωρήσουμε ότι έχουμε πρόβλημα δυαδικής ταξινόμησης, υπάρχουν $(2^2)^{\mu}$ πιθανοί τρόποι κατηγοριοποίησης των δεδομένων. Εάν κάθε δέντρο αντιστοιχεί σε μια μοναδική συνάρτηση αναζήτησης της κλάσης, μπορείτε εύκολα να δείτε πως είναι δύσκολο να κάνετε μια αναζήτηση σε όλα τα πιθανά δέντρα αποφάσεων για ένα σύνολο δεδομένων (ειδικά, εάν έχουμε μη δυαδικά χαρακτηριστικά και κλάσεις).



Διάγραμμα 9.1 : Παράδειγμα ΔΑ με κατηγορικά χαρακτηριστικά.

9.1.1. Ορολογία ΔΑ

- **Κόμβος ρίζας (root node):** Είναι η αρχή του δέντρου. Δεν έχει καμία εισερχόμενη ακμή, μηδέν ή περισσότερα εξερχόμενα κλαδιά.
- **Εσωτερικός κόμβος (internal node):** μία εισερχόμενη ακμή, δύο (ή περισσότερα) εξερχόμενα κλαδιά.
- **Καθαρότητα (purity):** Αυτή η έννοια βασίζεται στο κλάσμα των στοιχείων δεδομένων που ανήκουν στο υποσύνολο. Ένας τρόπος με τον οποίο μπορεί να οριστεί η καθαρότητα ενός συνόλου είναι η συχνότητα του πιο συνηθισμένου συστατικού του. Για παράδειγμα, εάν ένα σετ αποτελείται από 60% αντικείμενα της κατηγορίας Α, 30% στην κατηγορία Β και 10% στην κατηγορία Γ, τότε η καθαρότητά του είναι 60%.
- **Κόμβος φύλλου (leaf node):** σε κάθε κόμβο φύλλου εκχωρείται μια ετικέτα κλάσης εάν οι κόμβοι είναι καθαροί. Διαφορετικά, η ετικέτα της τάξης καθορίζεται με πλειοψηφία. Στα φύλλα δεν έχουμε περαιτέρω διακλαδώσεις.
- **Γονικός (parent) κόμβος και κόμβοι-παιδιά (child):** Εάν ένας κόμβος διαχωρίζεται, αναφερόμαστε σε αυτόν τον κόμβο ως τον γονικό κόμβο και οι κόμβοι που προκύπτουν ονομάζονται κόμβοι-παιδιά.

9.1.2. Χαρακτηριστικά ΔΑ

- Τα ΔΑ μπορούν να χρησιμοποιηθούν σε προβλήματα ταξινόμησης και σε προβλήματα παλινδρόμησης.
- Τα ΔΑ είναι μια από τις δημοφιλέστερες κατηγορίες μοντέλων για προβλήματα ταξινόμησης.
- Χρησιμοποιούν άπληστη αναζήτηση του χώρου των πιθανών δέντρων: η αναζήτηση γίνεται με μια σειρά τοπικών αναζητήσεων που μπορεί να μην οδηγήσουν στο καθολικό βέλτιστο (global optimum).
- Αλγόριθμος τύπου batch, δηλαδή χρησιμοποιεί πολλά (ή όλα) παραδείγματα σε κάθε επανάληψη και όχι ένα-ένα κάθε φορά.
- Είναι μη παραμετρικά μοντέλα: δηλαδή ο αριθμός των παραμέτρων που πρέπει να προσαρμοστούν σε σχέση με το πλήθος των δεδομένων εκπαίδευσης δεν είναι προκαθορισμένος (όπως π.χ. στη γραμμικά παλινδρόμηση).
- Είναι ντετερμινιστικός τύπος αλγορίθμου. Αυτό σημαίνει, ότι δεδομένης μιας συγκεκριμένης εισόδου, θα παράγει πάντα το ίδιο αποτέλεσμα, με το μοντέλο να περνά πάντα από την ίδια αλληλουχία καταστάσεων.

9.1.3. Σχέση ΔΑ με τη μάθηση μέσω κανόνων

Διαισθητικά, μπορούμε επίσης να σκεφτούμε το δέντρο αποφάσεων ως φωλιασμένους κανόνες «if-else». Ένας κανόνας είναι απλώς ένας συνδυασμός συνθηκών. Παράδειγμα:

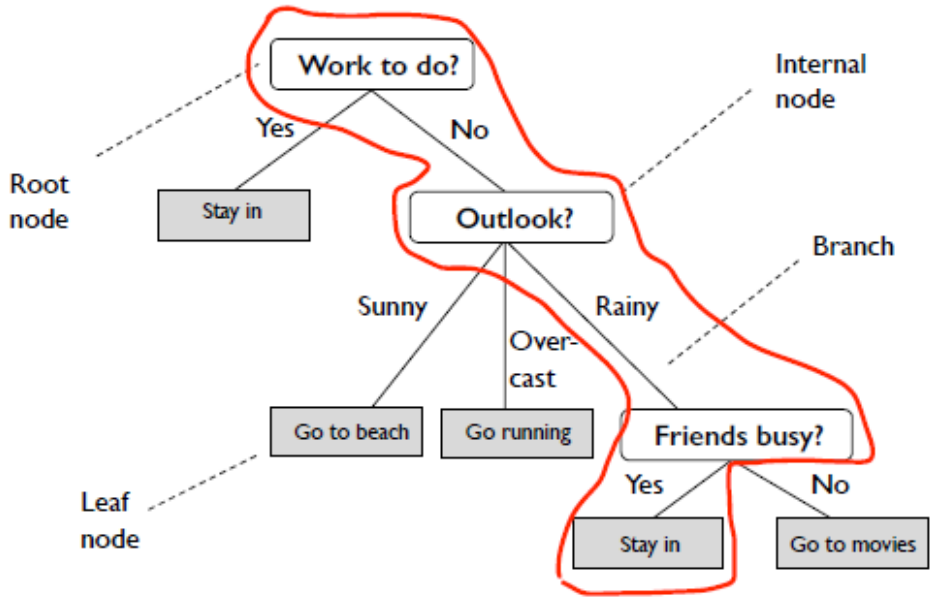
Κανόνας 1 : (if x = 1) AND (if y = 2) AND …

Πολλοί κανόνες μπορούν να συνδεθούν μεταξύ τους για να προβλέψουν την αξία του στόχου ενός παραδείγματος στο σετ εκπαίδευσης ή δοκιμής. Π.χ.:

Κλάση 1 : (Rule1 = True) OR (Rule2 = True) OR …

Κάθε κόμβος φύλλου σε ένα δέντρο αποφάσεων αντιπροσωπεύει ένα τέτοιο σύνολο κανόνων όπως φαίνεται στο παρακάτω σχήμα, το οποίο απεικονίζει τον κανόνα:

(Work to do? = False) AND (Outlook? = Rainy) AND (Friends busy? = Yes)



Διάγραμμα 9.2: Ο κανόνας για τον συγκεκριμένο κόμβο φύλου:
(Work to do? = False) AND (Outlook? = Rainy) AND (Friends busy? = Yes)

Λαμβάνοντας υπόψη το πλήρες δέντρο που απεικονίζεται στο προηγούμενο σχήμα, ο κανόνας απόφασης για την ετικέτα κλάσης "Stay In" μπορεί στη συνέχεια να γραφτεί ως το ακόλουθο σύνολο κανόνων:

((Work to do? = False) AND (Outlook? = Rainy) AND (Friends busy? = Yes)) OR (Work to do? = True)

9.2. Κατασκευή ΔΑ

Τα βασικά στάδια ενός αλγορίθμου κατασκευής ενός ΔΑ μπορούν να αποτυπωθούν σε ένα αναδρομικό αλγόριθμο με τα ακόλουθα βήματα:

1. Επιλογή του κατάλληλου χαρακτηριστικού στον κόμβο-πατέρα ώστε όταν γίνει η διάσπαση να έχουμε το μεγαλύτερο «κέρδος πληροφορίας».
2. Σταματάμε και επιστρέφουμε το δέντρο εάν όλα τα παιδιά-κόμβοι είναι «καθαρά» ή δεν μπορούμε να βελτιώσουμε άλλο την καθαρότητα των κλάσεων.
3. Επανάληψη της διαδικασίας στο επόμενο επίπεδο για κάθε παιδί του κόμβου που προκύπτει από το βήμα 1.

Στην πράξη προκύπτουν αρκετά θέματα για κάποιες ακραίες/μη συνηθισμένες καταστάσεις που πρέπει να αντιμετωπιστούν:

- Πώς αποφασίζουμε ποιο χαρακτηριστικό θα επιλέξουμε για τον διαχωρισμό ενός γονικού κόμβου σε κόμβους-παιδιά; Δηλαδή, ποιο είναι το κριτήριο για να μετρηθεί πόσο καλή είναι η διάσπαση;
- Εφόσον ένας διαχωρισμός πολλαπλών κατηγοριών μπορεί να εκφραστεί ως μια σειρά δυαδικών διαχωρισμών, ποια προσέγγιση πρέπει να προτιμηθεί;
- Ενώ ο διαχωρισμός κατηγορικός χαρακτηριστικών είναι εύκολα κατανοητός, πώς μπορούμε να αντιμετωπίσουμε τα συνεχή χαρακτηριστικά;
- Πότε σταματάμε να μεγαλώνουμε ένα δέντρο (γιατί ο πλήρης διαχωρισμός μπορεί εύκολα να οδηγήσει σε υπερπροσαρμογή);
- Πώς κάνουμε προβλέψεις εάν δεν υπάρχουν τρόποι για να διαχωρίσουμε τέλεια τους μη καθαρούς κόμβους περαιτέρω; (Καλές επιλογές είναι συνήθως η πλειοψηφία στα προβλήματα ταξινόμησης και ο μέσος όρος των δειγμάτων για τα προβλήματα παλινδρόμησης)

Οι σημαντικότερες διαφορές των αλγορίθμων κατασκευής ΔΑ είναι οι ακόλουθες:

- Το κριτήριο διάσπασης. Πιθανά κριτήρια:
    - Information gain (Shannon Entropy, Gini impurity, misclassication error)
    - Στατιστικά τεστ
    - Κάποια συνάρτηση βελτιστοποίησης
- Δυαδικός διαχωρισμός / Διαχωρισμός πολλαπλών μερών (πάνω από δύο)
- Δυνατότητα αξιοποίησης διακριτών / συνεχών μεταβλητών
- Κλάδεμα δέντρου πριν ή μετά την κατασκευή.

Οι πιο γνωστοί αλγόριθμοι ΔΑ είναι οι :

- ID3
- CART
- C4.5

9.2.1. Αλγόριθμος ID3 - Iterative Dichotomizer 3

- Περιγράφεται στο: Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1 (1), 81-106.
- Ένας από τους παλαιότερους αλγόριθμους δέντρων αποφάσεων
- Διακριτά χαρακτηριστικά, δεν μπορεί να χειριστεί αριθμητικά χαρακτηριστικά
- Διαχωρίσεις πολλών κατηγοριών
- Χωρίς κλάδεμα, επιρρεπής σε υπερπροσαρμογή (overfitting)

- Κοντά και πλατιά δέντρα (σε σύγκριση με το CART)
- Μεγιστοποιεί το κέρδος πληροφοριών/ελαχιστοποιεί την εντροπία
- Διακριτά χαρακτηριστικά, δυαδικά και πολλαπλών κατηγοριών χαρακτηριστικά

9.2.2. C4.5

- Περιγράφεται στο: Quinlan, J. R. (1993). C4.5: Programming for machine learning. Morgan Kauffmann, 38, 48.
- Χειρίζεται συνεχή και διακριτά χαρακτηριστικά (ο διαχωρισμός των συνεχών χαρακτηριστικών είναι πολύ απαιτητικός υπολογιστικά γιατί πρέπει να λαμβάνονται υπόψη όλα τα πιθανά εύρη)
- Το κριτήριο διαχωρισμού υπολογίζεται μέσω του λόγου κέρδους (gain ratio)
- Μπορεί να χειριστεί χαρακτηριστικά που λείπουν (τα αγνοεί στον υπολογισμό του κέρδους πληροφορίας)
- Εκτελεί κλάδεμα του δέντρου μετά την κατασκευή (post-pruning)

9.2.3. CART

- Περιγράφεται στο: Breiman, L. (1984). Classification and regression trees. Belmont, Calif: Wadsworth International Group.
- Συνεχή και διακριτά χαρακτηριστικά
- Αυστηρά δυαδικές διασπάσεις (τα δέντρα που προκύπτουν είναι ψηλότερα σε σύγκριση με τα ID3 και C4.5)
- Οι δυαδικές διαιρέσεις μπορούν να δημιουργήσουν καλύτερα δέντρα από το C4.5, αλλά τείνουν να είναι μεγαλύτερα και πιο δύσκολο να ερμηνευτούν. Δηλαδή για κ χαρακτηριστικά, έχουμε $2^{k-1} - 1$ τρόπους για να δημιουργηθεί μια δυαδικά κατάτμηση
- Μείωση διακύμανσης (variance reduction) στα δέντρα παλινδρόμησης
- Χρησιμοποιεί το κριτήριο Gini Impurity σε δέντρα ταξινόμησης
- Εκτελεί κλάδεμα κόστους-πολυπλοκότητας (δείτε πιο κάτω)

9.2.4. Κέρδος Πληροφορίας (Information Gain)

Το τυπικό κριτήριο που χρησιμοποιείται για τη διάσπαση σε δέντρα απόφασης είναι το λεγόμενο κέρδος πληροφορίας. Με απλά λόγια, όσο καλύτερη είναι η διάσπαση, τόσο υψηλότερο είναι το κέρδος πληροφοριών.

- Το κέρδος πληροφοριών βασίζεται στην έννοια της αμοιβαίας πληροφόρησης: Η μείωση της εντροπίας μιας μεταβλητής γνωρίζοντας την άλλη. Στην περίπτωσή μας (ταξινόμηση κλάσεων) η μία μεταβλητή είναι η κλάση και η άλλη ένα χαρακτηριστικό.

- Θέλουμε να μεγιστοποιήσουμε την αμοιβαία πληροφόρηση κατά τον καθορισμό κριτηρίων διαχωρισμού.
- Δηλαδή, ορίζουμε το κριτήριο σε έναν κόμβο έτσι ώστε να μεγιστοποιεί το κέρδος πληροφοριών:

$$GAIN(D, x_j) = I(D) - \sum_{u \in Values(x_j)} \frac{|D_u|}{|D|} I(D_u)$$

Όπου D είναι το σύνολο εκπαίδευσης στον γονικό κόμβο, D_u είναι ένα σύνολο δεδομένων σε έναν κόμβο-παιδί μετά τον διαχωρισμό, και I είναι μια συνάρτηση που μετρά την «ακαθαρσία» ενός δεδομένου κόμβου.

9.2.5. Υπερπροσαρμογή στα ΔΑ

Εάν τα δέντρα απόφασης δεν κλαδευτούν, έχουν υψηλό κίνδυνο να υπερπροσαρμόσουν τα δεδομένα εκπαίδευσης σε υψηλό βαθμό. Μια γενική προσέγγιση για την ελαχιστοποίηση της υπερπροσαρμογής στα δέντρα απόφασης είναι το κλάδεμα των ΔΑ. Υπάρχουν γενικά δύο προσεγγίσεις: κλάδεμα πριν και μετά (pre/post pruning).

9.2.5.1 Προ-Κλάδεμα (Pre-Pruning)

- Ορίζουμε μια αποκοπή βάθους (μέγιστο βάθος δέντρου) εκ των προτέρων.
- Κλάδεμα κόστους-πολυπλοκότητας: $I + \alpha|N|$, όπου I είναι μέτρο μη καθαρότητας, α είναι μια παράμετρος συντονισμού και $|N|$ είναι ο συνολικός αριθμός των κόμβων.
- Σταματάμε την ανάπτυξη εάν μια διάσπαση δεν είναι στατιστικά σημαντική (π.χ. με τεστ χ^2).
- Ορίζουμε έναν ελάχιστο αριθμό σημείων δεδομένων για κάθε κόμβο.

9.2.5.2 Μετά-Κλάδεμα (Post-Pruning)

- Αναπτύσσουμε πρώτα το πλήρες δέντρο και μετά αφαιρούμε τους κόμβους.
- Κλάδεμα με αξιολόγηση του σφάλματος: αφαιρούμε κόμβους μέσω αξιολόγησης του συνόλου δεδομένων επικύρωσης (προβληματικό όταν έχουμε λίγα δεδομένα).
- Μπορούμε επίσης να μετατρέψουμε τα δέντρα σε κανόνες πρώτα και μετά να κλαδέψουμε τους κανόνες.
- Υπάρχει ένας κανόνας ανά κόμβο φύλλου.
- Εάν οι κανόνες δεν ταξινομηθούν, τα σύνολα κανόνων έχουν κόστος κατά την αξιολόγηση, αλλά είναι πιο εκφραστικά.
- Δεν είναι απαραίτητο να αφαιρέσουμε και τους δύο κόμβους-παιδιά εάν αφαιρέσουμε τον κόμβο-ρίζα.

9.3. ΔΑ σε προβλήματα παλινδρόμησης

Τα δέντρα απόφασης μπορούν επίσης να χρησιμοποιηθούν σε προβλήματα παλινδρόμησης, η δυνατότητα αυτή εισήχθη μέσω του αλγορίθμου CART.

Όταν χρησιμοποιήσουμε ΔΑ για παλινδρόμηση, μεγαλώνουμε το δέντρο (δηλαδή αποφασίζουμε για κριτήρια διάσπασης σε κάθε κόμβο) μέσω της μείωσης διακύμανσης (variance reduction) σε κάθε κόμβο. Εδώ, η διακύμανση αναφέρεται στη διακύμανση μεταξύ των μεταβλητών στόχου στον γονικό κόμβο και στα παιδιά του.

Για τα προβλήματα παλινδρόμησης μπορούμε να χρησιμοποιήσουμε μια μετρική που να συγκρίνει τις (συνεχείς) μεταβλητές στόχου με τις προβλέψεις όπως το μέσο τετραγωνικό σφάλμα (MSE) σε ένα κόμβο:

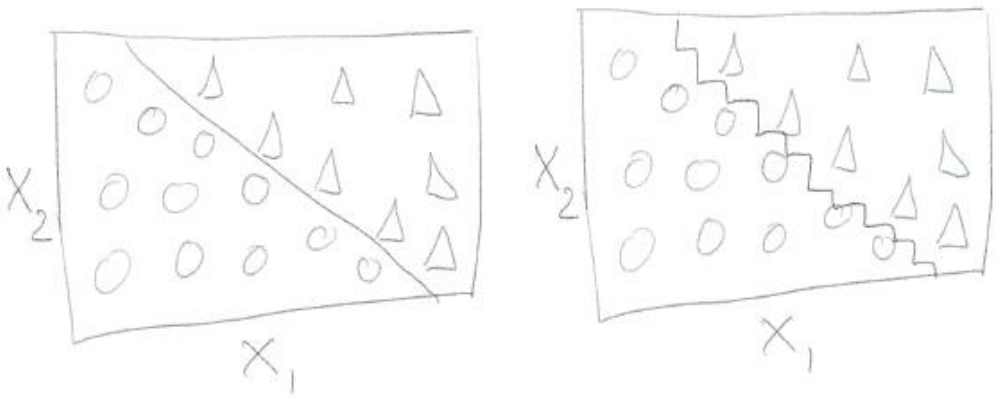
$$MSE = \frac{1}{n_t} \sum_{i=1, i \in Dt}^{n} \left(y_t^{[i]} - h\left(x^{[i]}\right)_t \right)^2$$

Σημειώστε ότι η προβλεπόμενη τιμή στόχου σε έναν κόμβο t, $h(X)_t$, υπολογίζεται ως η μέση τιμή δείγματος του υποσυνόλου εκπαίδευσης σε αυτόν τον κόμβο:

$$h(x)_t = \frac{1}{n_t} \sum_{i \in D_t} y^{[i]}$$

Αυτό το σφάλμα MSE στον εξεταζόμενο κόμβο ονομάζεται επίσης και «διακύμανση μες στον κόμβο» (within-node variance) και το κριτήριο διάσπασης έτσι ονομάζεται «μείωση διακύμανσης».

Σημειώστε ότι τα δέντρα απόφασης αντιμετωπίζουν το ίδιο πρόβλημα με τα δέντρα ταξινόμησης, καθώς δεν είναι κατάλληλα στην προσέγγιση διαγώνιων υπερεπιπέδων (hyperplains).



Διάγραμμα 9.3: Δέντρο ταξινόμησης που προσεγγίζει ένα διαγώνιο όριο απόφασης (δεξιά). Οι διαιρέσεις είναι πάντα κάθετες στους άξονες χαρακτηριστικών.

9.4. Συμπεράσματα

Παρακάτω παρατίθενται μερικά από τα πλεονεκτήματα και τα μειονεκτήματα της χρήσης των δέντρων αποφάσεων ως προγνωστικού μοντέλου.

- (+) Εύκολη ερμηνεία και επικοινωνία
- (+) Ανεξαρτησία από την κλιμάκωση χαρακτηριστικών
- (-) Υπερπροσαρμόζουν εύκολα
- (-) Απαιτούν περίτεχνο κλάδεμα
- (-) Αρκετά κοστοβόρο όταν απλώς πρέπει να προσαρμόσει μια "διαγώνια γραμμή"
- (-) Στα δέντρα παλινδρόμησης, το εύρος εξόδου περιορίζεται σε αυτό που υπάρχει στα δεδομένα εκπαίδευσης

9.5. Παράδειγμα εφαρμογής ΔΑ σε πρόβλημα ταξινόμισης με το πακέτο scikit-learn

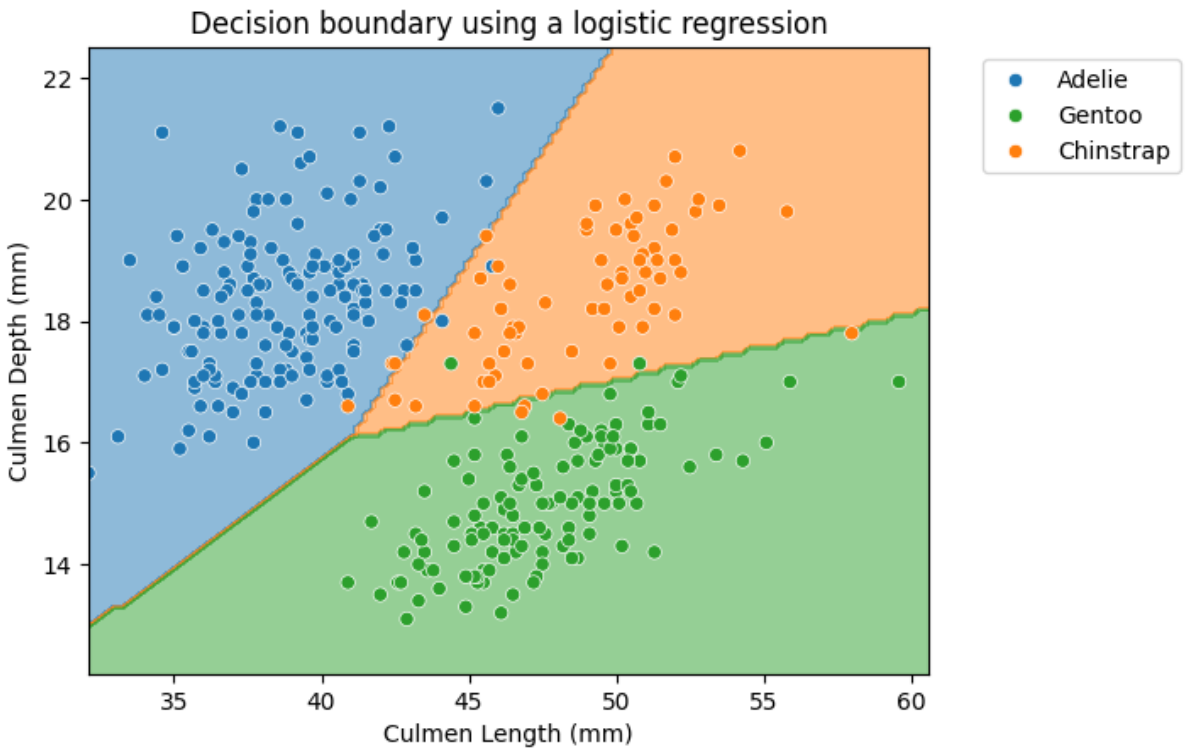
Σε αυτό το παράδειγμα παρουσιάζουμε τα δέντρα αποφάσεων σε ένα πρόβλημα ταξινόμησης πολλαπλών κλάσεων χρησιμοποιώντας ένα σύνολο δεδομένων με 2 χαρακτηριστικά και 3 κλάσεις (penguins). Για λόγους απλότητας, εστιάζουμε τη συζήτηση στην παράμετρο max_depth, η οποία ελέγχει το μέγιστο βάθος του δέντρου αποφάσεων.

```
import pandas as pd

penguins = pd.read_csv("https://raw.githubusercontent.com/INRIA/scikit-
learn-mooc/main/datasets/penguins_classification.csv")

culmen_columns = ["Culmen Length (mm)", "Culmen Depth (mm)"]
target_column = "Species"

# Αρχικά, χωρίζουμε τα δεδομένα σε δύο υποσύνολα εκπαίδευσης/δοκιμής.

from sklearn.model_selection import train_test_split

data, target = penguins[culmen_columns], penguins[target_column]
data_train, data_test, target_train, target_test = train_test_split(
    data, target, random_state=0
)
```

Για να δείξουμε τις διαφορές των ΔΑ με γραμμικά μοντέλα θα δημιουργήσουμε και οπτικοποιήσουμε πρώτα ένα μοντέλο λογιστικής παλινδόμισης. Οι γραμμικοί ταξινομητές ορίζουν ένα γραμμικό διαχωρισμό για να ξεχωρίζουν τις κλάσεις χρησιμοποιώντας έναν γραμμικό συνδυασμό των χαρακτηριστικών εισόδου. Αυτό σημαίνει ότι, στον δισδιάστατο χώρο των

χαρακτηριστικών μας, ένας γραμμικός ταξινομητής βρίσκει τις γραμμές που χωρίζουν καλύτερα τις κλάσεις. Αυτό εξακολουθεί να ισχύει για προβλήματα πολλαπλών κλάσεων, με τη διαφορά ότι τοποθετούνται περισσότερες από μία γραμμές. Μπορούμε να χρησιμοποιήσουμε το `DecisionBoundaryDisplay` του scikit-learn για να σχεδιάσουμε τα όρια απόφασης που μαθαίνει ο ταξινομητής.

```
from sklearn.linear_model import LogisticRegression

linear_model = LogisticRegression()
linear_model.fit(data_train, target_train)

import matplotlib.pyplot as plt
import matplotlib as mpl
import seaborn as sns

from sklearn.inspection import DecisionBoundaryDisplay
tab10_norm = mpl.colors.Normalize(vmin=-0.5, vmax=8.5)
palette = ["tab:blue", "tab:green", "tab:orange"]

dbd = DecisionBoundaryDisplay.from_estimator(
    linear_model,
    data_train,
    response_method="predict",
    cmap="tab10",
    norm=tab10_norm,
    alpha=0.5,
)

sns.scatterplot(
    data=penguins,
    x=culmen_columns[0],
    y=culmen_columns[1],
    hue=target_column,
    palette=palette,
)
plt.legend(bbox_to_anchor=(1.05, 1), loc="upper left")
_ = plt.title("Decision boundary using a logistic regression")
```



Διάγραμμα 9.4: Οπτικοποίηση ορίου απόφασης με λογιστική παλινδόμηση

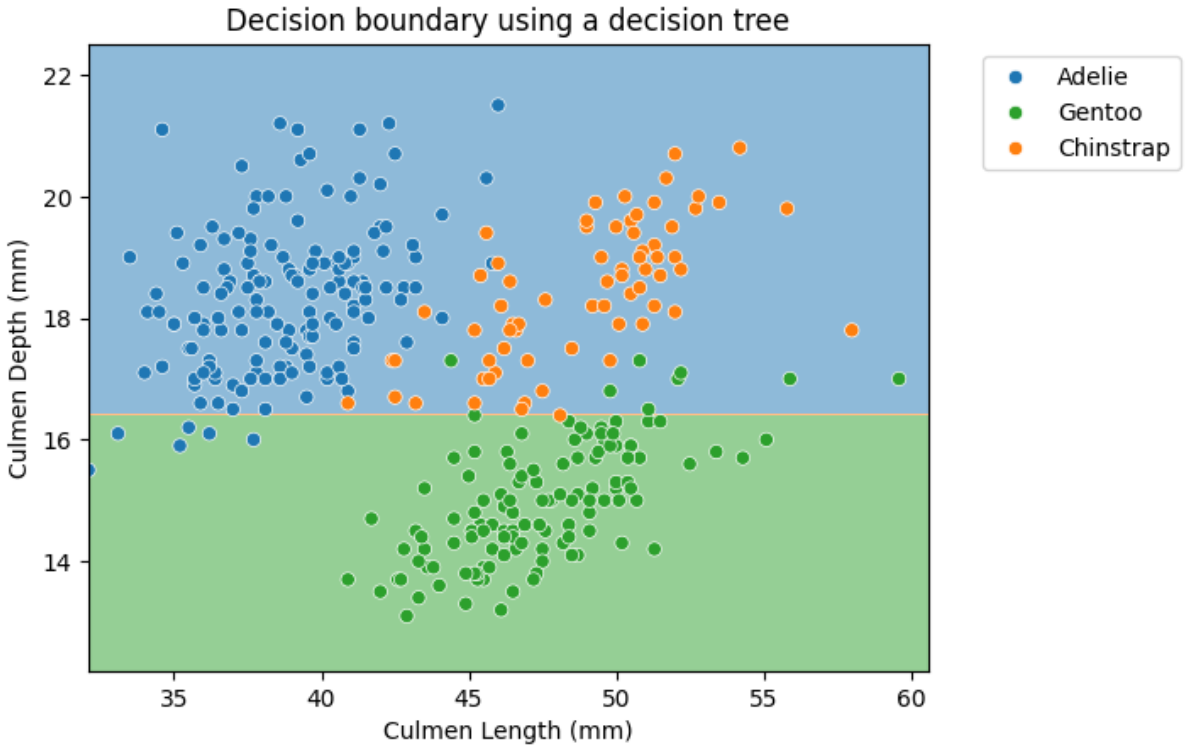
Βλέπουμε ότι οι γραμμές είναι ένας συνδυασμός των χαρακτηριστικών εισόδου αφού δεν είναι κάθετες σε συγκεκριμένο άξονα. Φαίνεται ότι το γραμμικό μοντέλο είναι καλή επιλογή για αυτό το πρόβλημα καθώς δίνει καλή ακρίβεια:

```
linear_model.fit(data_train, target_train)
test_score = linear_model.score(data_test, target_test)
print(f"Accuracy of the LogisticRegression: {test_score:.2f}")

#  Accuracy of the LogisticRegression: 0.98
```

Ας εκπαιδεύσουμε ένα ΔΑ όταν ορίσουμε την παράμετρο `max_depth` ώστε να επιτρέπει μόνο έναν διαχωρισμό στον χώρο των χαρακτηριστικών.

```
from sklearn.tree import DecisionTreeClassifier

tree = DecisionTreeClassifier(max_depth=1)
tree.fit(data_train, target_train)


DecisionBoundaryDisplay.from_estimator(
    tree,
```

```
    data_train,
    response_method="predict",
    cmap="tab10",
    norm=tab10_norm,
    alpha=0.5,
)
sns.scatterplot(
    data=penguins,
    x=culmen_columns[0],
    y=culmen_columns[1],
    hue=target_column,
    palette=palette,
)

plt.xlabel(culmen_columns[0])
plt.ylabel(culmen_columns[1])
#plt.colorbar(scatter, label=target_column)
plt.legend(bbox_to_anchor=(1.05, 1), loc="upper left")
_ = plt.title("Decision boundary using a decision tree")
```

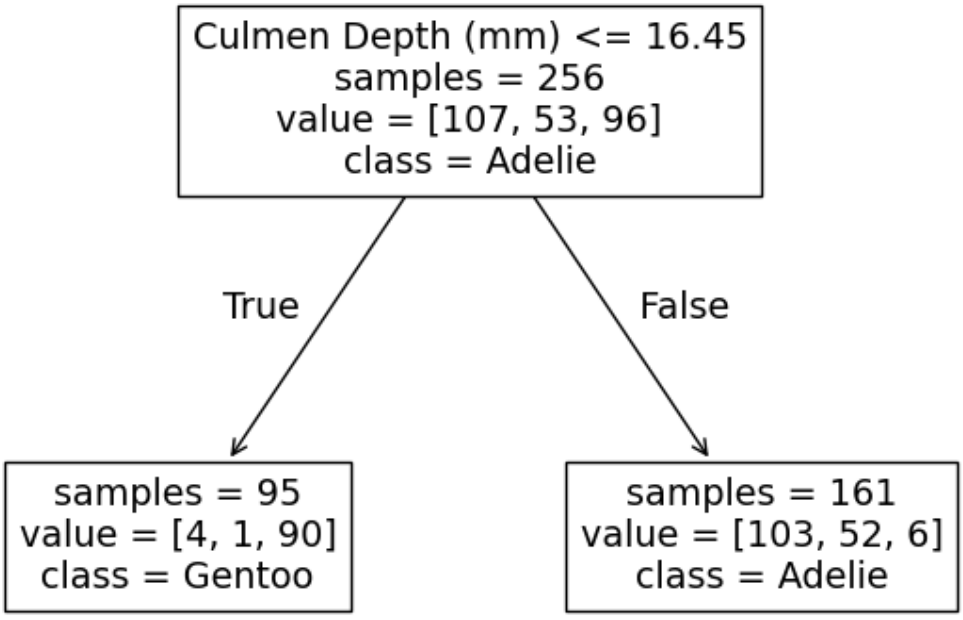


Διάγραμμα 9.5: Οπτικοποίηση ορίου απόφασης με ΔΑ (max_length = 1)

Οι κατατμήσεις που βρέθηκαν από τον αλγόριθμο διαχωρίζουν τα δεδομένα κατά μήκος του άξονα "Culmen Depth", απορρίπτοντας το χαρακτηριστικό "Culmen Length". Έτσι, φαίνεται ότι ένα δέντρο

αποφάσεων δεν χρησιμοποιεί συνδυασμό χαρακτηριστικών όταν κάνει μια διάσπαση. Μπορούμε να δούμε σε μεγαλύτερο βάθος τη δομή του δέντρου.

```
from sklearn.tree import plot_tree

_, ax = plt.subplots(figsize=(8, 6))
_ = plot_tree(
    tree,
    feature_names=culmen_columns,
    class_names=tree.classes_.tolist(),
    impurity=False,
    ax=ax,
)
```



Διάγραμμα 9.6: Οπτικοποίηση ΔΑ (max_length = 1)

Βλέπουμε ότι η διάσπαση έγινε στο χαρακτηριστικό culmen depth. Τα αρχικά δεδομένα διαιρέθηκαν σε 2 σύνολα με βάση το βάθος culmen depth (κατώτερο ή ανώτερο από 16,45 mm).

Αυτή η κατάτμηση του συνόλου δεδομένων ελαχιστοποιεί την ποικιλομορφία κλάσεων σε καθεμία από τις υποδιαιρέσεις. Αυτό το μέτρο είναι επίσης γνωστό ως **κριτήριο (criterion)** και είναι μια ρυθμιζόμενη παράμετρος.

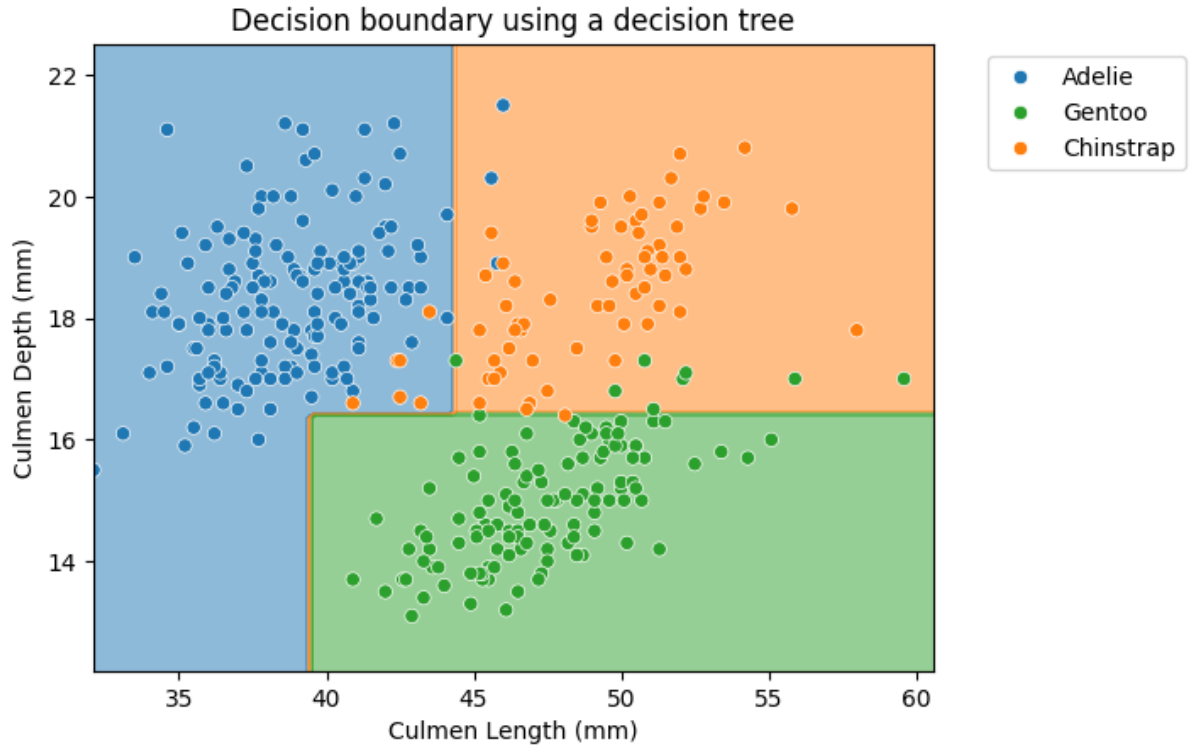
Αν κοιτάξουμε πιο προσεκτικά τους διαχωρισμούς, βλέπουμε ότι όταν το δείγμα είναι ανώτερο από 16.45 τότε αυτό ανήκει κυρίως στην κατηγορία «Adelie». Βλέποντας τις αξίες, όντως παρατηρούμε 103 άτομα «Adelie» σε αυτόν την κλάση. Μετράμε επίσης 52 "Chinstrap" δείγματα και 6 δείγματα "Gentoo". Μπορούμε να κάνουμε παρόμοια ερμηνεία για το διαχωριστικό που ορίζεται από κατώφλι μικρότερο από 16,45 mm. Στην προκειμένη περίπτωση η πιο αντιπροσωπευτική κλάση είναι η "Gentoo". Ας δούμε πώς θα λειτουργούσε το δέντρο μας ως μηχανισμός πρόγνωσης. Ας ξεκινήσουμε με μια περίπτωσηόπου το culmen depth είναι κατώτερο από το κατώφλι.

```
test_penguin_1 = pd.DataFrame(
    {"Culmen Length (mm)": [0], "Culmen Depth (mm)": [15]}
)
tree.predict(test_penguin_1)

# array(['Gentoo'], dtype=object)
# Η κλάση που προβλέπει το μοντέλο είναι η "Gentoo".
```

Επιστρέφοντας στο πρόβλημα ταξινόμησης, η διάσπαση που βρέθηκε με μέγιστο βάθος 1 δεν είναι αρκετά ισχυρή για να χωρίσει τις τρεις κλάσεις και η ακρίβεια του μοντέλου είναι χαμηλή σε σύγκριση με το γραμμικό μοντέλο. Πράγματι, αυτό δεν αποτελεί έκπληξη. Είδαμε νωρίτερα ότι ένα μεμονωμένο χαρακτηριστικό δεν είναι δυνατό να χωρίσει και τις τρεις κλάσεις: το μοντέλο υποπροσασμόζει. Ωστόσο, από την προηγούμενο ανάλυση είδαμε ότι χρησιμοποιώντας και τα δύο χαρακτηριστικά θα πρέπει να είμαστε σε θέση να πάρουμε καλά αποτελέσματα. Στο επόμενο παράδειγμα θα αυξήσουμε την παράμετρο max_depth σε 2:

```
tree = DecisionTreeClassifier(max_depth=2)
tree.fit(data_train, target_train)

tab10_norm = mpl.colors.Normalize(vmin=-0.5, vmax=8.5)

palette = ["tab:blue", "tab:green", "tab:orange"]
DecisionBoundaryDisplay.from_estimator(
    tree,
    data_train,
    response_method="predict",
    cmap="tab10",
    norm=tab10_norm,
    alpha=0.5,
```

```
)
ax = sns.scatterplot(
    data=penguins,
    x=culmen_columns[0],
    y=culmen_columns[1],
    hue=target_column,
    palette=palette,
)
plt.legend(bbox_to_anchor=(1.05, 1), loc="upper left")
_ = plt.title("Decision boundary using a decision tree")
```

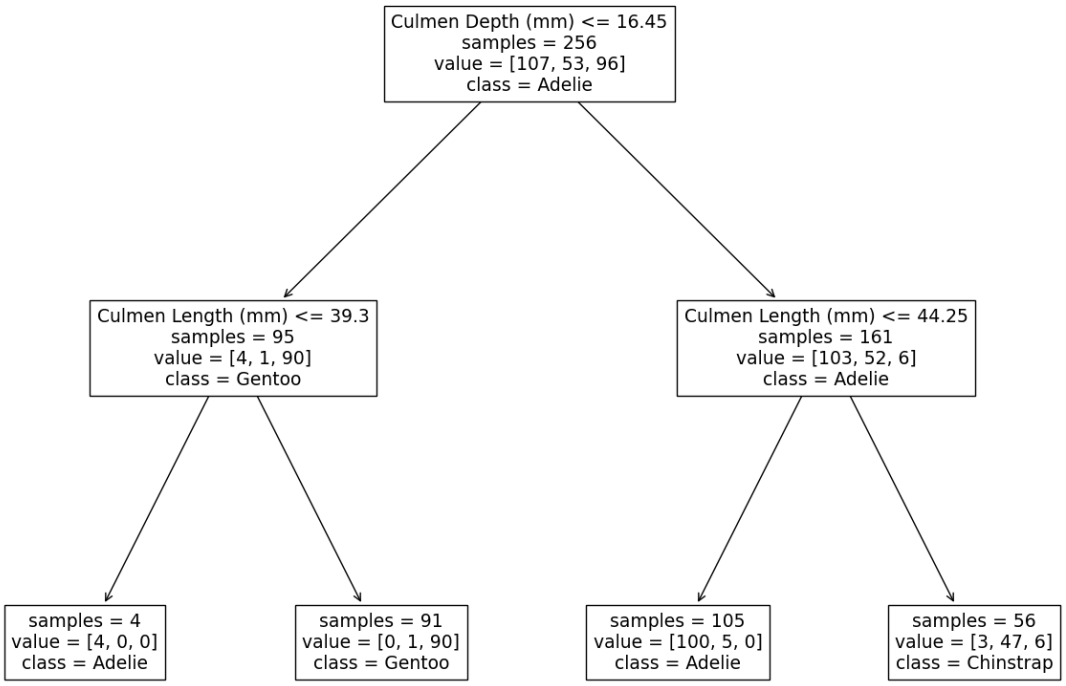


Διάγραμμα 9.7: Οπτικοποίηση ορίου απόφασης με ΔΑ (max_length = 2)

Βλέπουμε ότι τώρα το μοντέλο κατανέμει καλύτερα τις κλάσεις. Οπτικοποιούμε και το διάγραμμα:

```
from sklearn.tree import plot_tree

_, ax = plt.subplots(figsize=(16, 12))
_ = plot_tree(
    tree,
    feature_names=culmen_columns,
    class_names=tree.classes_.tolist(),
    impurity=False,
```

```
    ax=ax,
)
```



Διάγραμμα 9.8: Οπτικοποίηση ΔΑ (max_length = 2)

Το δέντρο που προκύπτει έχει 7 κόμβους: 1 κόμβο ρίζα, 2 εσωτερικούς κόμβους και 4 είναι κόμβοι "φύλλα", οργανωμένοι σε 2 επίπεδα. Βλέπουμε ότι το δεύτερο επίπεδο δέντρου χρησιμοποίησε το "Culmen Length" για να πάρει δύο νέες αποφάσεις. Ποιοτικά, είδαμε ότι ένα τόσο απλό δέντρο ήταν αρκετό για να ταξινομήσει τα είδη των πιγκουίνων.

9.6. Παράδειγμα εφαρμογής ΔΑ σε πρόβλημα παλινδρόμισης με το πακέτο scikit-learn

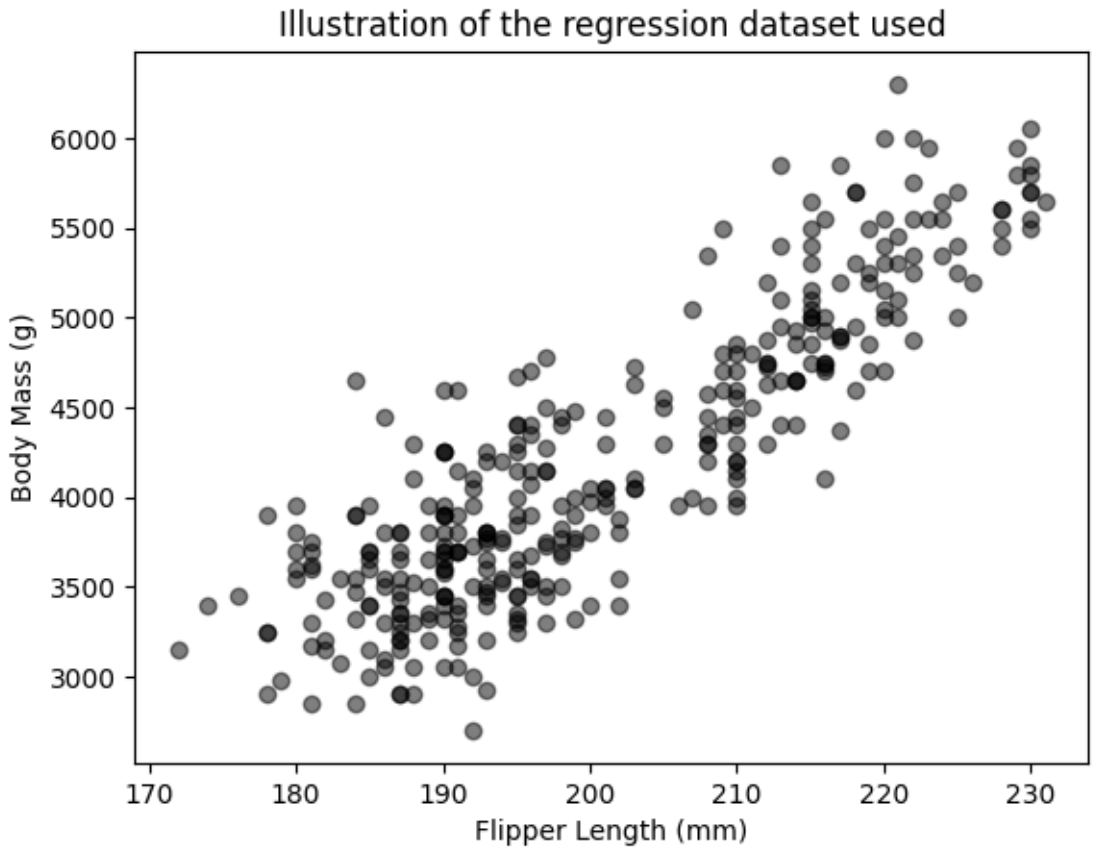
Σε αυτό το παράδειγμα, παρουσιάζουμε πώς λειτουργούν τα ΔΑ σε προβλήματα παλινδρόμησης. Δείχνουμε διαφορές με τα δέντρα αποφάσεων που παρουσιάστηκαν προηγουμένως για

ταξινόμηση. Αρχικά, φορτώνουμε ένα ειδικά διαμορφωμένο σύνολο δεδομένων "penguins" για την επίλυση ενός προβλήματος παλινδρόμησης.

```
import pandas as pd

penguins = pd.read_csv("https://raw.githubusercontent.com/INRIA/scikit-
learn-mooc/main/datasets/penguins_regression.csv")
feature_name = "Flipper Length (mm)"
target_name = "Body Mass (g)"
data_train, target_train = penguins[[feature_name]],
penguins[target_name]
```

Για να δείξουμε πώς προβλέπουν τα δέντρα απόφασης σε μια κατάσταση παλινδρόμησης, δημιουργούμε ένα συνθετικό σύνολο δεδομένων που περιέχει μερικές από τις πιθανές τιμές μήκους πτερυγίου μεταξύ του ελάχιστου και του μέγιστου των αρχικών δεδομένων. Η χρήση του όρου "test" εδώ αναφέρεται σε δεδομένα που δεν χρησιμοποιήθηκαν για εκπαίδευση. Αυτό δεν πρέπει να συγχέεται με τα δεδομένα που προέρχονται από μια διάσπαση train-test, καθότι τα δεδομένα εδώ δημιουργήθηκαν σε ίσα διαστήματα για την οπτική αξιολόγηση των προβλέψεων. Σημειώστε ότι αυτό ισχύει μεθοδολογικά εδώ γιατί ο στόχος μας είναι να πάρουμε κάποια διαισθητική κατανόηση σχετικά με το σχήμα της συνάρτησης απόφασης του εκπαιδευμένου δέντρου αποφάσεων. Ωστόσο, ο υπολογισμός μιας μέτρησης αξιολόγησης σε ένα τέτοιο συνθετικό σύνολο δοκιμών θα ήταν χωρίς νόημα αφού το συνθετικό σύνολο δεδομένων δεν ακολουθεί την ίδια κατανομή με τα δεδομένα του πραγματικού κόσμου στα οποία θα αναπτυχθεί το μοντέλο.

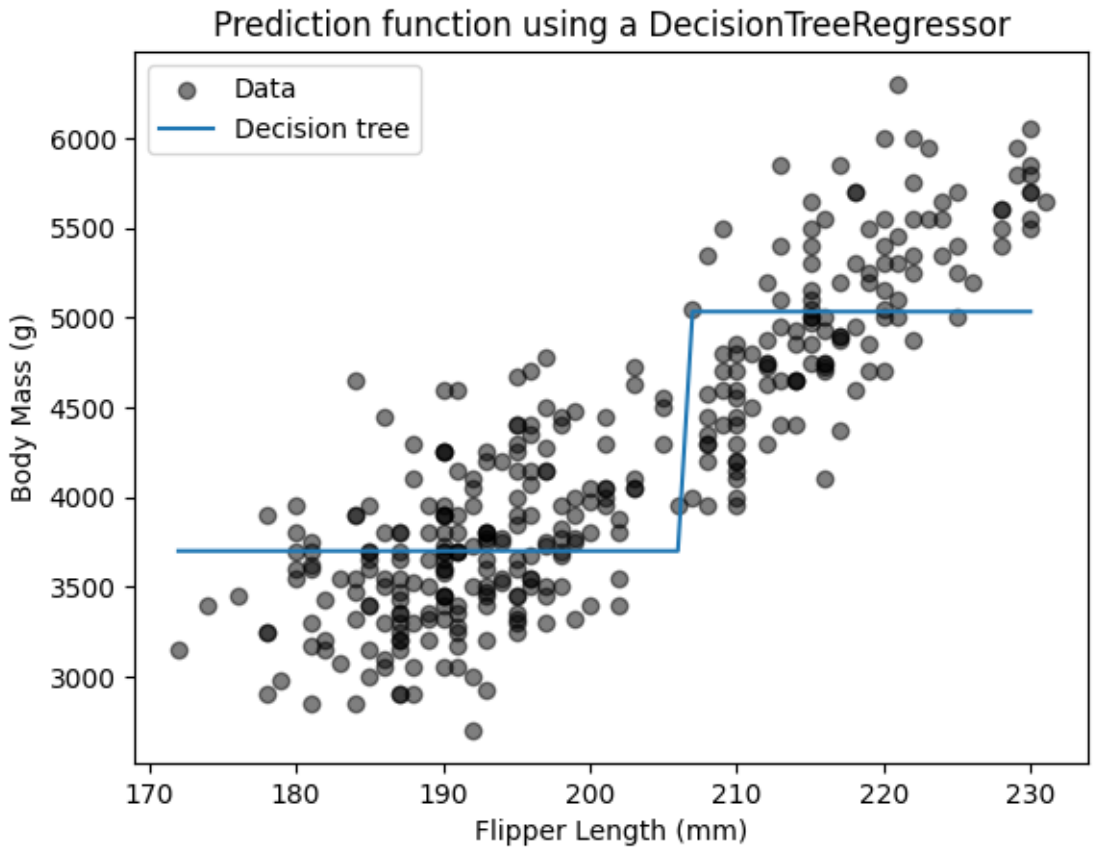
```
import matplotlib.pyplot as plt

plt.scatter(penguins[feature_name], penguins[target_name],
color="black", alpha=0.5)
plt.xlabel(feature_name)
plt.ylabel(target_name)
_ = plt.title("Illustration of the regression dataset used")
```



Διάγραμμα 9.9: Δεδομένα που θα χρησιμοποιηθούν στο παράδειγμα παλινδρόμισης

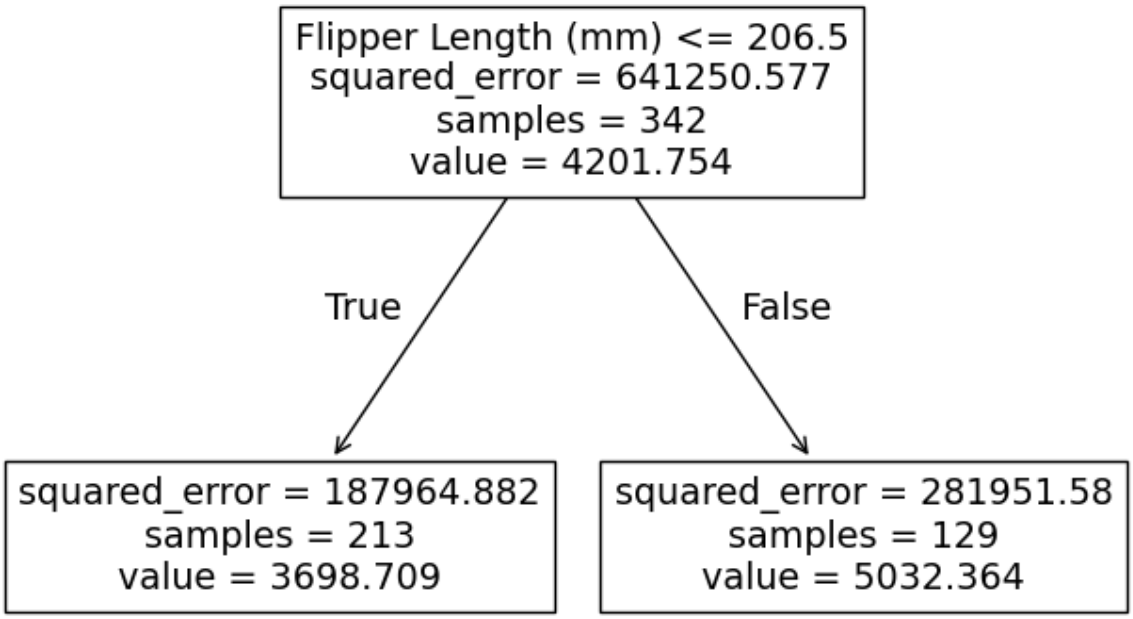
Δημιουργία μοντέλου ΔΑ με max_depth = 1.

```
from sklearn.tree import DecisionTreeRegressor

tree = DecisionTreeRegressor(max_depth=1)
tree.fit(data_train, target_train)
target_predicted = tree.predict(data_test)

plt.scatter(penguins[feature_name], penguins[target_name],
color="black", alpha=0.5, label="Data")
plt.plot(data_test[feature_name], target_predicted, label="Decision
tree")
plt.xlabel(feature_name)
plt.ylabel(target_name)
plt.legend()
_ = plt.title("Prediction function using a DecisionTreeRegressor")
```



Διάγραμμα 9.10: Συνάρτηση πρόβλεψης ΔΑ με max_depth=1

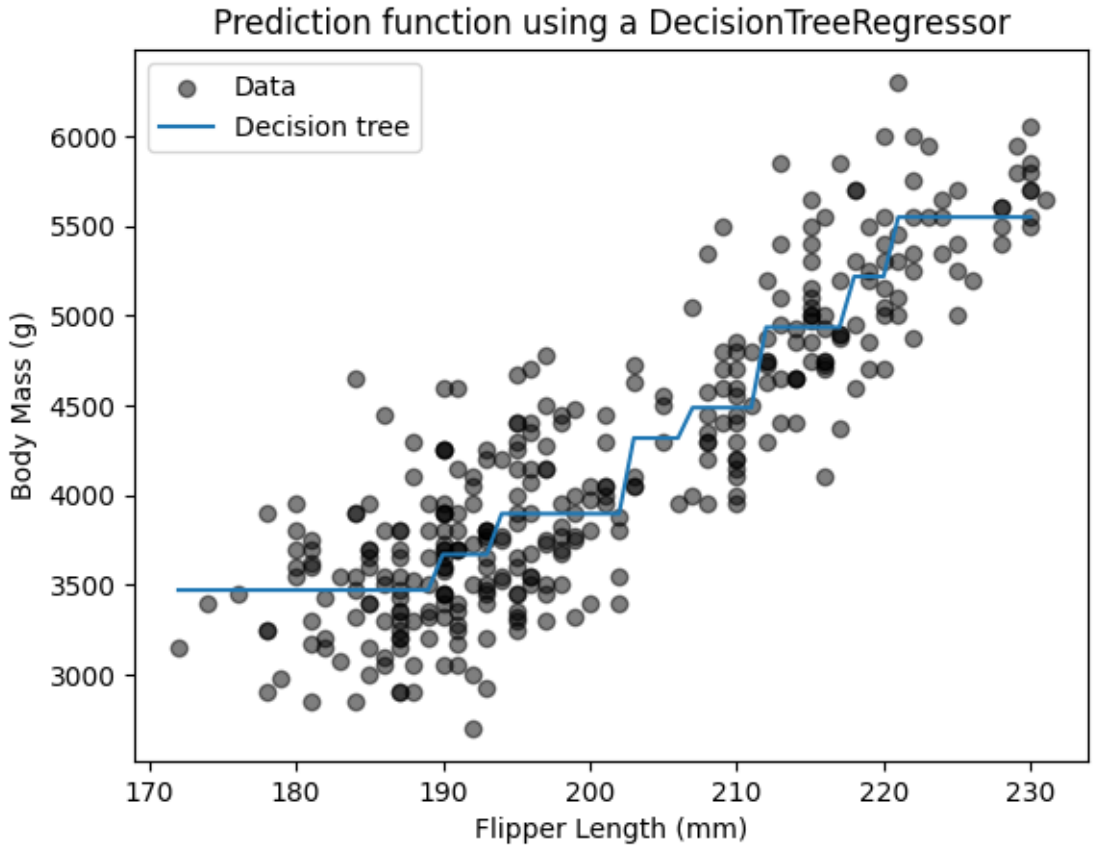
```
from sklearn.tree import plot_tree

_, ax = plt.subplots(figsize=(8, 6))
_ = plot_tree(tree, feature_names=[feature_name], ax=ax)
```



Διάγραμμα 9.11: Οπτικοποίηση ΔΑ (max_depth = 1)

Δημιουργία μοντέλου ΔΑ με max_depth = 3:

```
tree = DecisionTreeRegressor(max_depth=3)
tree.fit(data_train, target_train)
target_predicted = tree.predict(data_test)

plt.scatter(penguins[feature_name], penguins[target_name],
color="black", alpha=0.5, label="Data")
plt.plot(data_test[feature_name], target_predicted, label="Decision
tree")
plt.xlabel(feature_name)
plt.ylabel(target_name)
plt.legend()
_ = plt.title("Prediction function using a DecisionTreeRegressor")
```



Διάγραμμα 9.12: Συνάρτηση πρόβλεψης ΔΑ με max_depth=3

Η αύξηση του βάθους του δέντρου αυξάνει τον αριθμό των κατατμήσεων και έτσι ο αριθμός των σταθερών τιμών που το δέντρο μπορεί να προβλέψει.

9.7. *Ερωτήσεις αυτοαξιολόγησης*

9.1 Ένας από τους τρόπους για να αντιμετωπίσουμε την υπερπροσαρμογή (overfitting) στα δέντρα απόφασης είναι αυξάνοντας την παράμετρο του μεγίστου βάθος (max_depth)

α) Σωστό

β) Λάθος

9.2 Σε ένα δέντρο αποφάσεων, ποιος είναι ο σκοπός του κόμβου ρίζας;

α) Αντιπροσωπεύει τις ετικέτες κλάσεων των δεδομένων εκπαίδευσης.

β) Αποθηκεύει τις τιμές χαρακτηριστικών των δεδομένων εκπαίδευσης.

γ) Αντιπροσωπεύει την τελική πρόβλεψη που γίνεται από το δέντρο αποφάσεων.

δ) Αντιπροσωπεύει το χαρακτηριστικό που παρέχει τον καλύτερο διαχωρισμό για το σύνολο δεδομένων.

9.3Ποιο από τα παρακάτω είναι πλεονέκτημα των δέντρων αποφάσεων (επίλεξε όλα όσα είναι σωστά);

α) Μπορούν να χειριστούν τόσο κατηγορικά όσο και αριθμητικά χαρακτηριστικά.

β) Μπορούν να χειριστούν με ευκολία δεδομένα πολλών διαστάσεων..

γ) Έχουν ανοσία στην υπερπροσαρμογή (overfitting).

δ) Είναι εύκολα στη κατανόηση και ερμηνεία.

9.4 Ποιος είναι ο σκοπός των κόμβων «φύλλα» στα δέντρα απόφασης

α) Αναπαριστούν την ετικέτα της κατηγορίας ή την τιμή που θα προβλεφθεί

β) Αποθηκεύουν τις συνθήκες για τη διάσπαση των δεδομένων

γ) Υποδεικνύουν τη σημασία ενός χαρακτηριστικού

δ) Αναπαριστούν το βάθος του δέντρου

ΚΕΦΑΛΑΙΟ 10: K-NN (K-NEAREST NIGHBORS)

10.1. Εισαγωγή

Οι αλγόριθμοι του πλησιέστερου γείτονα (Nearest Neighbors - NN) είναι από τους απλότερους αλγόριθμους μηχανικής μάθησης και έχουν μελετηθεί εξονυχιστικά στον τομέα της αναγνώρισης προτύπων τον περασμένο αιώνα. Αναπτύχθηκαν αρχικά το 1957 από τους Fix και Hodges[1] και εξελίχθηκαν το 1967 από τον Corver[2]. Ο αλγόριθμος K-NN (K-NEAREST NEIGHBORS) αφορά την εφαρμογή των τεχνικών αναζήτησης πλησιέστερου γείτονα σε προβλήματα επιβλεπόμενης μάθησης (supervised learning). Αν και οι αλγόριθμοι του πλησιέστερου γείτονα δεν είναι τόσο δημοφιλείς όσο ήταν κάποτε, εξακολουθούν να χρησιμοποιούνται ευρέως στην πράξη ειδικά σε συστήματα συστάσεων (recommendation systems) και στην ανίχνευση ακραίων στοιχείων (outlier detection).

10.2. Ο αλγοριθμοσ κ-νν

10.2.1. Βασικές έννοιες

Η ιδέα πίσω από τον k-NN είναι σχετικά απλή: είναι ένας αλγόριθμος επιβλεπόμενης μάθησης που απλά αποθηκεύει τα παραδείγματα εκπαίδευσης,

$$\langle x^{[i]}, y^{[i]} \rangle \in D \quad (|D| = n),$$

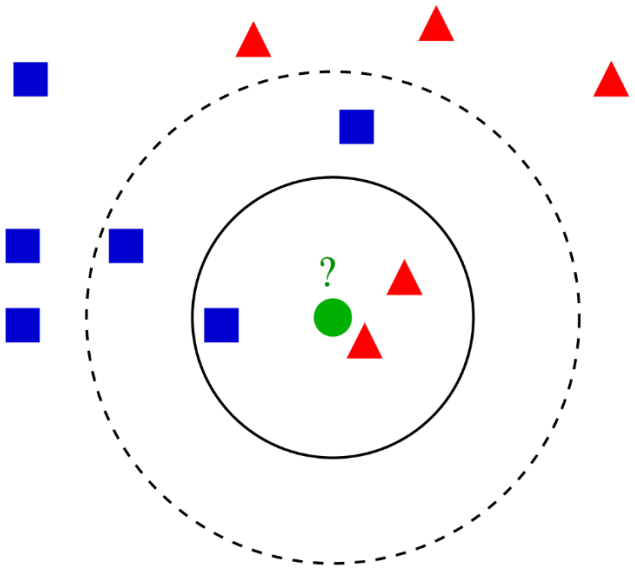
κατά τη φάση της εκπαίδευσης. Για τον λόγο αυτό, ο k-NN ονομάζεται επίσης αλγόριθμος «τεμπέλικης» (lazy) μάθησης, διότι η επεξεργασία των παραδειγμάτων εκπαίδευσης αναβάλλεται μέχρι να γίνουν προβλέψεις.

Στη συνέχεια, για να κάνει μια πρόβλεψη (ετικέτα κλάσης ή συνεχής στόχος), ένα εκπαιδευμένο μοντέλο k-NN βρίσκει τους k πλησιέστερους γείτονες του σημείου ερωτήματος και υπολογίζει την ετικέτα κλάσης (ταξινόμηση) ή τον συνεχή στόχο (παλινδρόμηση) με βάση τα k πλησιέστερα (πιο «όμοια») σημεία. Στις επόμενες ενότητες θα εξηγηθεί αναλυτικά αυτός ο μηχανισμός. Ωστόσο, η γενική ιδέα είναι ότι αντί να προσεγγίζει συνολικά τη συνάρτηση στόχο f(x) = y, κατά τη διάρκεια κάθε πρόβλεψης, ο k-NN προσεγγίζει τη συνάρτηση του στόχου τοπικά. Στην πράξη, είναι πιο εύκολο να μάθει κανείς να προσεγγίζει μια συνάρτηση τοπικά παρά καθολικά.

---

[1] *Fix, Evelyn; Hodges, Joseph L. (1951). Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties (PDF) (Report). USAF School of Aviation Medicine, Randolph Field, Texas. Archived (PDF) from the original on September 26, 2020.*

[2] *Cover, Thomas M.; Hart, Peter E. (1967). "Nearest neighbor pattern classification" (PDF). IEEE Transactions on Information Theory. **13** (1): 21–27*



Διάγραμμα 10.13: Παράδειγμα ταξινόμησης k-NN. Το δείγμα δοκιμής (πράσινη κουκκίδα) πρέπει να ταξινομηθεί είτε σε μπλε τετράγωνα είτε σε κόκκινα τρίγωνα. Αν k = 3 (κύκλος συμπαγούς γραμμής) τότε η κλάση που δίνεται είναι τα κόκκινα τρίγωνα επειδή υπάρχουν 2 τρίγωνα και μόνο 1 τετράγωνο μέσα στον εσωτερικό κύκλο. Αν k = 5 (διακεκομμένος κύκλος) η κλάση που δίνεται είναι τα μπλε τετράγωνα (3 τετράγωνα έναντι 2 τριγώνων μέσα στον εξωτερικό κύκλο). Πηγή εικόνας: wikipedia[3]

10.2.2. Περιπτωση κ=1: αλγοριθμος 1-νν

Η ενότητα παρέχει μια πιο τεχνική περιγραφή του αλγορίθμου για k=1, 1-πλησιέστερου γείτονα (1-NN). Οι παρακάτω αλγόριθμοι αναγράφονται σε μορφή ψευδοκώδικα τύπου python.

Αλγόριθμος εκπαίδευσης:

```
for i = 1,…,n στο n-διάστατο σύνολο δεδομένων εκπαίδευσης D (|D|=n):
      Αποθήκευσε το παράδειγμα εκπαίδευσης ⟨x[i], y[i]⟩
```

Αλγόριθμος πρόβλεψης (όπου q το τεστ παράδειγμα που θέλουμε να κάνουμε πρόβλεψη):

```
closest_point = None
closest_distance = ∞
for i = 1,…,n:
      current_distance = d(x[i], x[q])
      if current_distance < closest_distance :
            closest_distance = current_distance
```

[3] https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

```
            target_value_of_closest_point = y[i]
return target_value_of_closest_point
```

Η πρόβλεψη που παράγεται από το μοντέλο 1-NN, $h(x^{[q]})$ είναι η τιμή στόχος του πλησιέστερου σημείου. Η πιο συνηθισμένη συνάρτηση μέτρησης απόστασης των αλγορίθμων k-NN είναι η Ευκλείδεια απόσταση, που υπολογίζει την απόσταση δύο σημείων $x^{[a]}$ και $x^{[b]}$ ως εξής:

$$d\left(x^{[a]}, x^{[b]}\right) = \sqrt{\sum_{j=1}^{m} \left(x_j^{[a]} - x_j^{[b]}\right)^2}$$

10.2.3. k-nn σε προβλήματα ταξινόμησης και παλινδρόμησης

Προηγουμένως, περιγράψαμε τον αλγόριθμο 1-NN, ο οποίος κάνει μια πρόβλεψη εκχωρώντας την ετικέτα κλάσης ή την τιμή συνεχούς στόχου του πλησιέστερου παραδείγματος εκπαίδευσης από το σημείο που θέλουμε να προβλέψουμε (όπου η ομοιότητα τυπικά υπολογίζεται χρησιμοποιώντας τη μέτρηση της Ευκλείδειας απόστασης για συνεχή χαρακτηριστικά).

Αντί να βασίζεται στην πρόβλεψη του απλού, πλησιέστερου παραδείγματος εκπαίδευσης, ο αλγόριθμος k-NN λαμβάνει υπόψη τους k πλησιέστερους γείτονες όταν προβλέπει μια ετικέτα κλάσης (σε προβλήματα ταξινόμησης) ή μια συνεχή τιμή στόχου (σε προβλήματα παλινδρόμησης).

10.2.3.1 Ταξινόμηση

Στα προβλήματα ταξινόμησης, ένα τυπικό μοντέλο k-NN συνήθως προβλέπει την ετικέτα κλάσης στόχου ως την ετικέτα κλάσης που αναπαρίσταται συχνότερα ανάμεσα στα k πλησιέστερα παραδείγματα εκπαίδευσης και του δεδομένου σημείου που ζητάμε την πρόβλεψη.

10.2.3.2 Παλινδρόμηση

Η γενική ιδέα του k-NN στην παλινδρόμηση είναι η ίδια όπως και για την ταξινόμηση: πρώτον, βρίσκουμε τους k πλησιέστερους γείτονες στο σύνολο δεδομένων. Δεύτερον, κάνουμε μια πρόβλεψη με βάση τις ετικέτες των k πλησιέστερων γειτόνων. Ωστόσο, στην παλινδρόμηση η συνάρτηση-στόχος είναι μια συνάρτηση πραγματικής αντί για διακριτή τιμή, οπότε μια προσέγγιση του συνεχούς στόχου είναι ο υπολογισμός της μέσης τιμής του στόχου από τους k πλησιέστερους γείτονες. Μια άλλη επιλογή που χρησιμοποιείται αντί του μέσου όρου είναι η διάμεσος.

10.2.3.3 Απλή υλοποίηση αλγορίθμου k-NN σε ψευδοκώδικα

```
Dk = {}

while |Dk| < k

      closest_distance = ∞

      for i = 1, … , n, :  ∀ i  ∉ Dk

            current_distance = d(x[i], x[q])

            if current_distance < closest_distance:

                  closest_distance = current_distance

                  closest_point = x[i]

      add closest_point to Dk
```

Η συγκεκριμένη υλοποίηση είναι αρκετά απαιτητική υπολογιστικά. Για την επιτάχυνση του αλγορίθμου έχουν βρεθεί αρκετές τεχνικές μερικές από τις οποίες παρουσιάζονται περιληπτικά παρακάτω:

10.2.3.4 Τεχνικές βελτίωσης αλγορίθμου k-NN

- Πιο αποδοτικές δομές δεδομένων για την αποθήκευση των δεδομένων εκπαίδευσης (Bucketing[4], KD-tree[5], Ball-Tree[6]). Τα μοντέλα k-NN που παρέχει το scikit-learn έχουν μια παράμετρο algorithm που μπορεί να πάρει τις ακόλουθες τιμές: {‘auto’, ‘ball_tree’, ‘kd_tree’, ‘brute’}, default=’auto’. Η προεπιλεγμένη τιμή auto προσπαθεί να βρει αυτόματα ποια είναι η καλύτερη τεχνική.
- Διαγραφή παραδειγμάτων: μπορούμε να αφαιρέσουμε οριστικά σημεία δεδομένων που δεν επηρεάζουν το όριο απόφασης. Για παράδειγμα, θεωρήστε ένα απομονωμένο σημείο δεδομένων (γνωστός και ως “outlier”) που περιβάλλεται από πολλά σημεία δεδομένων από μια διαφορετική κλάση. Αυτό το σημείο μπορούμε να το αφαιρέσουμε με ασφάλεια.
- Μείωση διαστάσεων: Εάν ο αριθμός των χαρακτηριστικών μπορεί να μειωθεί με ασφάλεια τότε έχουμε μια σημαντική βελτίωση της ταχύτητας του k-NN καθώς αυτή εξαρτάται από τον υπολογισμό των αποστάσεων.

[4] Ronald L Rivest. On the Optimality of Elia's Algorithm for Performing Best-Match Searches." In: IFIP Congress. 1974, pp. 678-681.
[5] Jon Louis Bentley. Multidimensional binary search trees used for associative searching". In: Communications of the ACM 18.9 (1975), pp. 509-517.
[6] Stephen M Omohundro. Five balltree construction algorithms. International Computer Science Institute Berkeley, 1989.

- Παραλληλοποίηση k-NN: υπάρχουν πολλοί τρόποι να παραλληλοποιήσουμε τον αλγόριθμο k-NN εύκολα. Στο scikit-learn υπάρχει η παράμετρος n_jobs, όπου μπορούμε να ορίσουμε το αριθμό των παράλληλων εργασιών. Με την τιμή -1 το scikit-learn χρησιμοποιεί όλους τους επεξεργαστές/threads που βρίσκει διαθέσιμους.

10.2.3.5 Συναρτήσεις μέτρησης απόστασης

Υπάρχουν πολλές συναρτήσεις μέτρησης της απόστασης έτσι ώστε να επιλέξουμε τους k πλησιέστερους γείτονες. Η επιλογή της «καλύτερης» συνάρτησης είναι συνήθως διαφορετική ανάλογα με το είδος προβλήματος που έχουμε. Για συνεχή χαρακτηριστικά η πιο συνηθισμένη συνάρτηση είναι η ευκλείδεια απόσταση. Άλλες συναρτήσεις που χρησιμοποιούνται συνήθως είναι οι

- Απόσταση Manhattan
- Απόσταση Minkowski
- Απόσταση Mahalanobis

Όμως ένα πρόβλημα με τις πιο «περίπλοκες» συναρτήσεις είναι ότι κάνουν τον αλγόριθμο πιο αργό στην εκτέλεση. Το scikit-learn έχει δύο παραμέτρους που ρυθμίζουν αυτή τη συνάρτηση (p, metric)[7]. Η προεπιλεγμένη συνάρτηση είναι η ευκλείδεια απόσταση.

10.2.3.6 Στάθμιση του μέτρου απόστασης

Στον «κανονικό» k-NN όλοι οι k γείτονες συμμετέχουν ομοίως στην επιλογή κλάσεων ή στον μέσο όρο. Ωστόσο, ειδικά εάν η ακτίνα που περικλείει ένα σύνολο γειτόνων είναι μεγάλη, μπορεί να θέλουμε να δώσουμε ισχυρότερα βάρη σε γείτονες που είναι πιο κοντά στο σημείο του ερωτήματος. Η παράμετρος που ρυθμίζει αυτήν την επιλογή στο scikit-learn είναι η «weights» με τιμές : {‘uniform’, ‘distance’}, callable, default=’uniform’. Η επιλογή ‘uniform’ είναι η «κλασική» ρύθμιση, ενώ με τη ‘distance’ τα σημεία σταθμίζονται ανάλογα με την απόσταση: πιο κοντινά σημεία έχουν περισσότερο βάρος στην απόφαση από αυτά που είναι πιο μακριά. Με την callable μπορούμε να ορίσουμε εμείς τη συνάρτηση των βαρών.

10.2.4. Βελτιώνοντας την απόδοση

Υπάρχουν αρκετοί διαφορετικοί τρόποι για να βελτιώσουμε την προγνωστική απόδοση των αλγορίθμων k-NN:

- Επιλέγοντας την τιμή του k.
- Κλιμάκωση των αξόνων χαρακτηριστικών.

[7] Περισσότερες πληροφορίες στο manual: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html#sklearn.neighbors.KNeighborsClassifier

- Επιλογή μέτρησης απόστασης.
- Στάθμιση του μέτρου απόστασης.

Στην πράξη, τιμές του k μεταξύ 3 και 15 είναι συνήθως αρκετές για να δώσουν καλή απόδοση. Όπως σε όλους τους αλγόριθμους μηχανικής μάθησης η εύρεση των σωστών τιμών για τις παραμέτρους γίνεται συνήθως με cross-validation.

10.3. Πλεονεκτηματα – Μειονεκτηματα

10.3.1. Πλεονεκτηματα

- Εύκολη υλοποίηση.
- Εύκολη ερμηνεία.
- Εύκολη προσθήκη νέων δεδομένων αφού δεν χρειάζεται να «εκπαιδευτεί» νέο μοντέλο.

10.3.2. Μειονκτηματα

- Η πρόβλεψη είναι πολύ απαιτητική υπολογιστικά όσο μεγαλώνει το σετ δεδομένων. Με χρήση «έξυπνων» δομών δεδομένων πάντως το πρόβλημα μειώνεται. Επίσης οι απαιτήσεις σε μνήμη είναι ανάλογες με το μέγεθος των δεδομένων.
- Δυσκολεύονται σε προβλήματα με μεγάλο αριθμό χαρακτηριστικών καθώς ο υπολογισμός των αποστάσεων γίνεται πιο απαιτητικός.
- Συνήθως τα χαρακτηριστικά χρειάζονται κλιμάκωση.

10.4. Παράδειγμα εφαρμογής k-NN σε πρόβλημα ταξινόμισης με το πακέτο scikit-learn

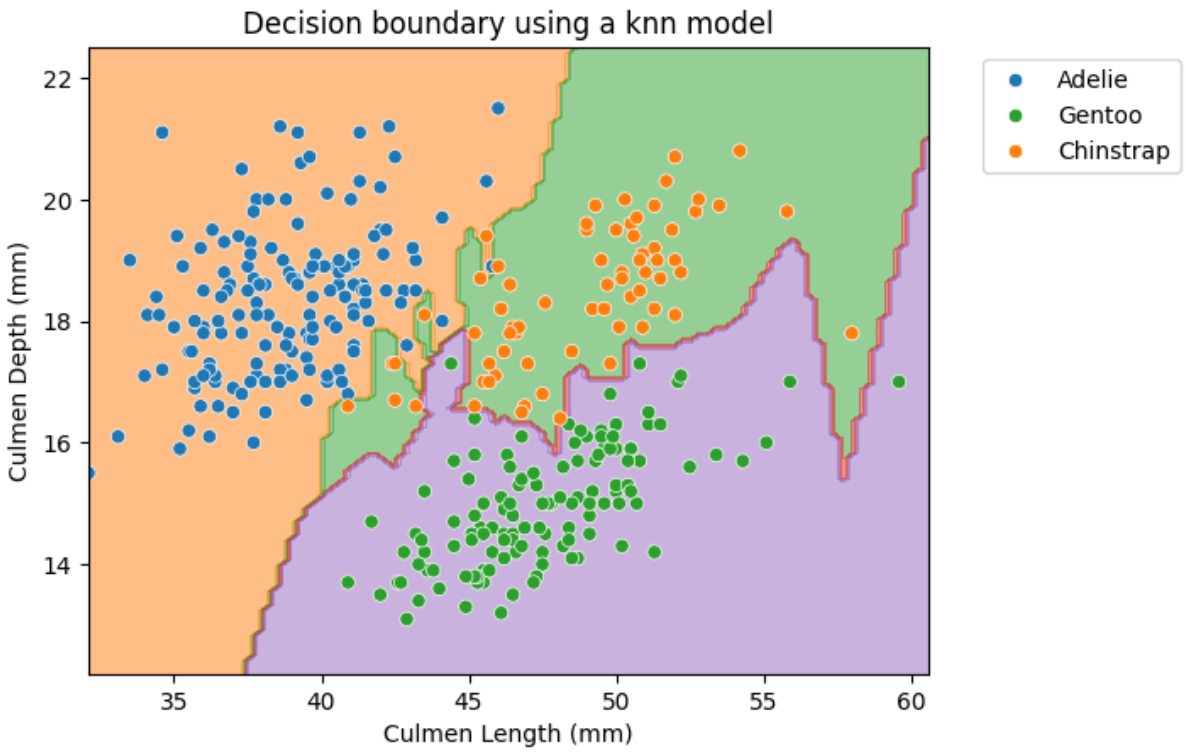
Το πακέτο scikit-learn περιέχει τον αλγόριθμο KNeighborsClassifier για να εφαρμόζει τον k-NN σε προβλήματα ταξινόμισης. Σε αυτό το παράδειγμα παρουσιάζουμε τον αλγόριθμο αυτό στο ίδιο πρόβλημα ταξινόμησης πολλαπλών κλάσεων που είδαμε και στα δέντρα αποφάσεων (penguins). Για λόγους απλότητας, εστιάζουμε τη συζήτηση στην παράμετρο k (n_neighbors).

```
import pandas as pd

penguins = pd.read_csv("https://raw.githubusercontent.com/INRIA/scikit-
learn-mooc/main/datasets/penguins_classification.csv")

culmen_columns = ["Culmen Length (mm)", "Culmen Depth (mm)"]
target_column = "Species"

# Αρχικά, χωρίζουμε τα δεδομένα σε δύο υποσύνολα εκπαίδευσης/δοκιμής.

from sklearn.model_selection import train_test_split
```

```
data, target = penguins[culmen_columns], penguins[target_column]
data_train, data_test, target_train, target_test = train_test_split(
    data, target, random_state=0
)
```

Ξεκινάμε με την απλή περίπτωση όπου k = 1. Σε αυτήν την περίπτωση η ταξινόμηση γίνεται παρατηρώντας το πιο κοντινό σημείο στα δεδομένα εκπαίδευσης. Σημειώνουμε ότι όταν έχουμε αριθμητικά δεδομένα καλό είναι να γίνεται κλιμάκωση με τον StandardScaler πριν από την "εκπαίδευση" μοντέλων k-NN. Εδώ δεν θα το κάνουμε αυτό για να μπορεί να γίνει πιο εύκολα η σύγκριση με τα όρια απόφασης των δέντρων απόφασης.

```
from sklearn.neighbors import KNeighborsClassifier
import matplotlib.pyplot as plt
import matplotlib as mpl
import seaborn as sns

knn_model = KNeighborsClassifier(n_neighbors=1)
knn_model.fit(data_train, target_train)

from sklearn.inspection import DecisionBoundaryDisplay
tab10_norm = mpl.colors.Normalize(vmin=-0.5, vmax=4.5)
# create a palette to be used in the scatterplot
palette = ["tab:blue", "tab:green", "tab:orange"]

dbd = DecisionBoundaryDisplay.from_estimator(
    knn_model,
    data_train,
    response_method="predict",
    cmap="tab10",
    norm=tab10_norm,
    alpha=0.5,
)

sns.scatterplot(
    data=penguins,
    x=culmen_columns[0],
    y=culmen_columns[1],
    hue=target_column,
    palette=palette,
)
# put the legend outside the plot
plt.legend(bbox_to_anchor=(1.05, 1), loc="upper left")
_ = plt.title("Decision boundary using a knn model")
```



Διάγραμμα 10.14: Όρια απόφασης του μοντέλου 1-nn

Βλέπουμε ότι τα όρια έχουν αρκετές διακυμάνσεις. Είναι φανερό ότι το μοντέλο υπερπροσαρμόζει.

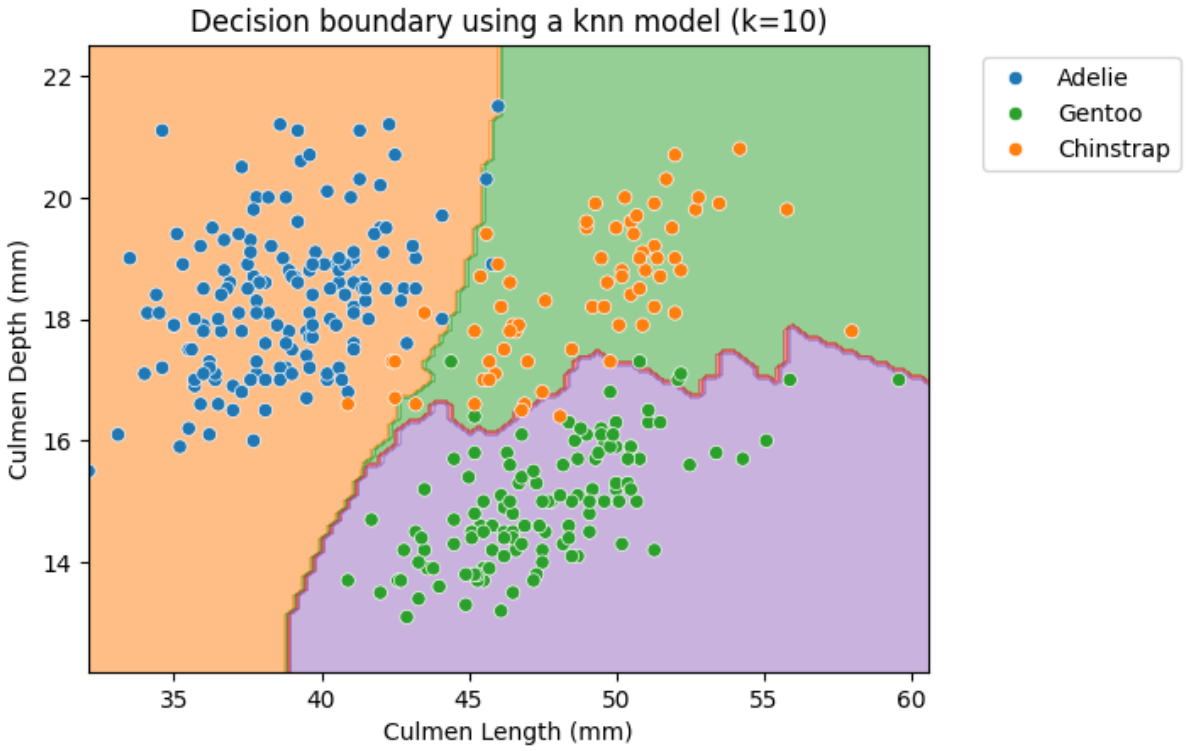
Ας δούμε και την ακρίβεια του μοντέλου:

```
knn_model.fit(data_train, target_train)
test_score = knn_model.score(data_test, target_test)
print(f"Accuracy of the knn model: {test_score:.2f}")

# Accuracy of the knn model: 0.99
```

Τώρα ας πραγματοποιήσουμε την ίδια διαδικασία για k = 10:

```
knn_model = KNeighborsClassifier(n_neighbors=10)
knn_model.fit(data_train, target_train)

dbd = DecisionBoundaryDisplay.from_estimator(
    knn_model,
    data_train,
    response_method="predict",
    cmap="tab10",
    norm=tab10_norm,
    alpha=0.5,
)
```

```
sns.scatterplot(
    data=penguins,
    x=culmen_columns[0],
    y=culmen_columns[1],
    hue=target_column,
    palette=palette,
)
# put the legend outside the plot
plt.legend(bbox_to_anchor=(1.05, 1), loc="upper left")
_ = plt.title("Decision boundary using a knn model (k=10)")
```



Διάγραμμα 10.15: Όρια απόφασης του μοντέλου 10-nn

Βλέπουμε ότι τα όρια απόφασης εδώ είναι πιο ομαλά.

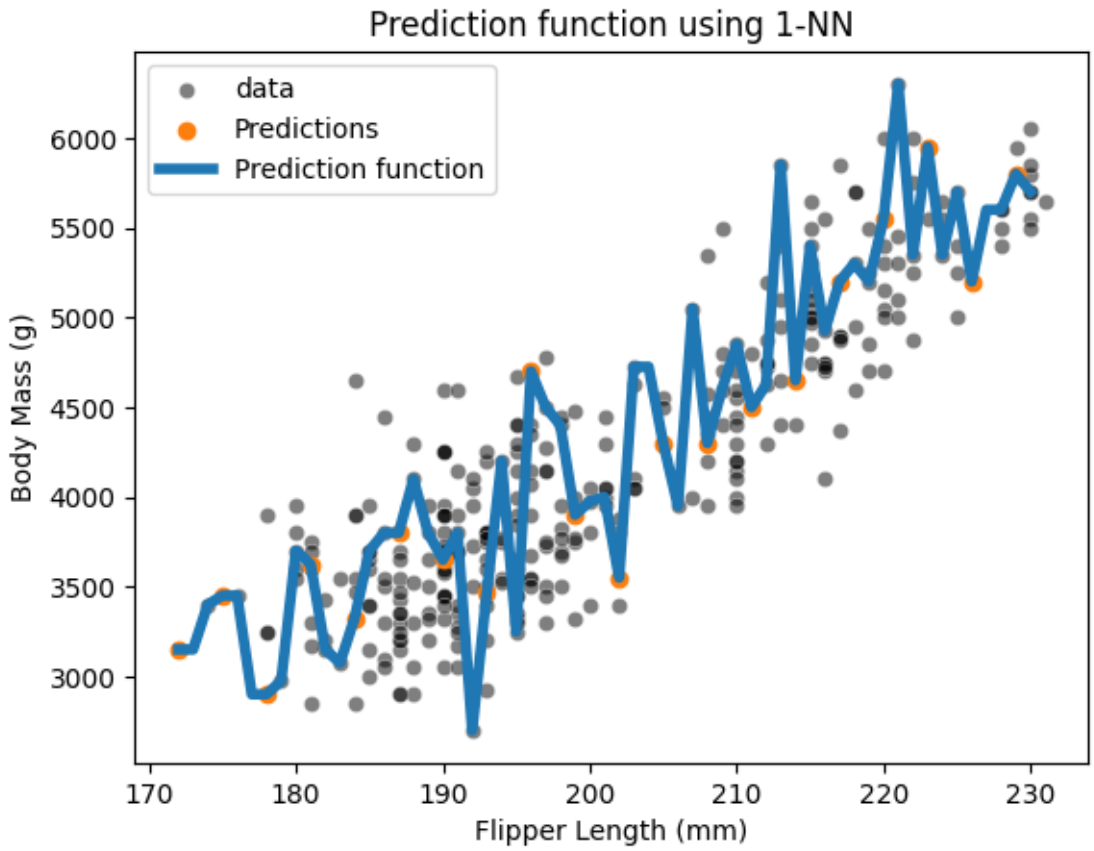
10.5. Παράδειγμα εφαρμογής k-NN σε πρόβλημα παλινδρόμισης με το πακέτο scikit-learn

Σε αυτό το παράδειγμα παρουσιάζουμε πώς λειτουργεί ο αλγόριθμος k-NN στο ίδιο πρόβλημα ταξινόμησης πολλαπλών κλάσεων που είδαμε και στα δέντρα αποφάσεων (penguins). Αρχικά, φορτώνουμε το ειδικά διαμορφωμένο σύνολο δεδομένων "penguins" για την επίλυση ενός προβλήματος παλινδρόμησης.

```
import pandas as pd

penguins = pd.read_csv("https://raw.githubusercontent.com/INRIA/scikit-
learn-mooc/main/datasets/penguins_regression.csv")
feature_name = "Flipper Length (mm)"
target_name = "Body Mass (g)"
data_train, target_train = penguins[[feature_name]],
penguins[target_name]
```

Ας προσαρμόσουμε αρχικά ένα μοντέλο k-NN όταν k = 1. Σε αυτή την περίπτωση η παλινδρόμηση γίνεται παρατηρώντας το πιο κοντινό σημείο στα δεδομένα εκπαίδευσης. Χρησιμοποιούμε μια βοηθητική συνάρτηση για την εκπαίδευση/οπτικοποίηση.

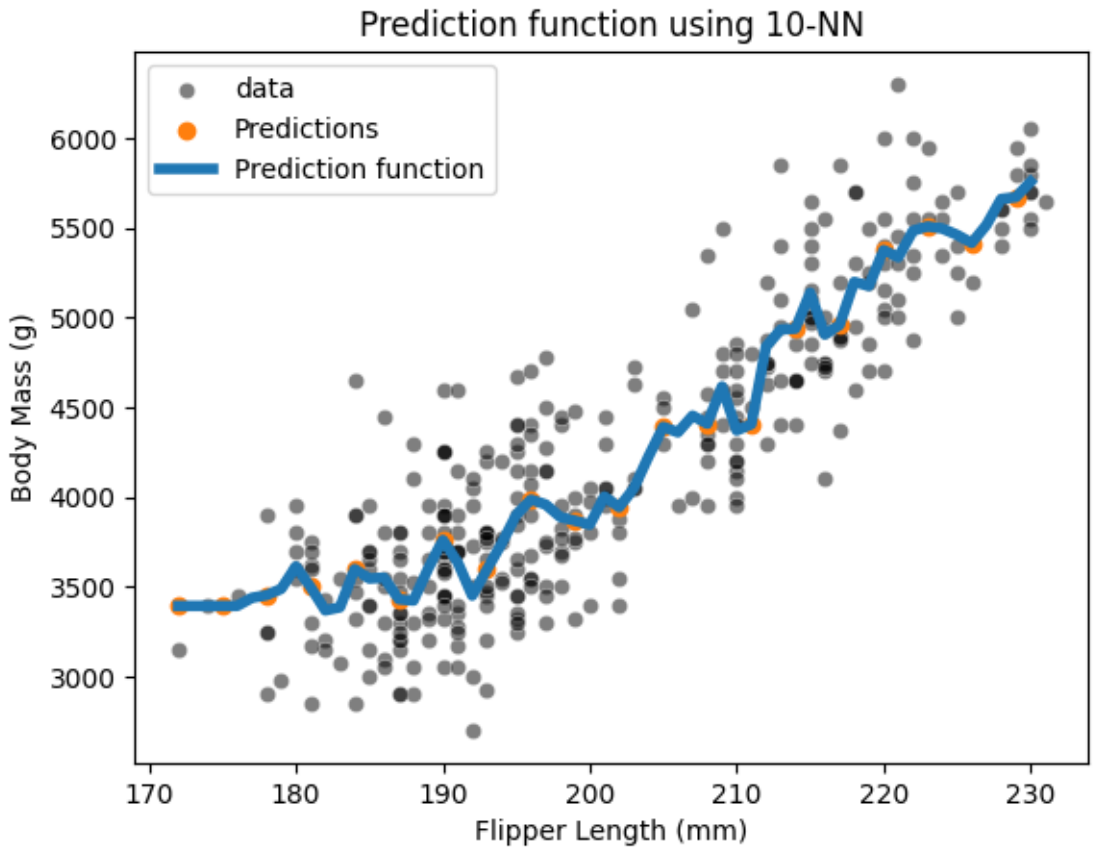
```
import matplotlib.pyplot as plt
import matplotlib as mpl
import seaborn as sns
from sklearn.neighbors import  KNeighborsRegressor

def fit_and_plot_regression(model, data, feature_names, target_names):
    model.fit(data[feature_names], data[target_names])
    data_test = pd.DataFrame(
        np.arange(data.iloc[:, 0].min(), data.iloc[:, 0].max()),
        columns=data[feature_names].columns,
    )
    target_predicted = model.predict(data_test)

    sns.scatterplot(
        x=data.iloc[:, 0], y=data[target_names], color="black",
alpha=0.5, label="data"
    )
    plt.scatter(
        data_test[::3],
        target_predicted[::3],
        label="Predictions",
        color="tab:orange",
    )
    plt.plot(data_test.iloc[:, 0], target_predicted, linewidth=4,
label="Prediction function")
    plt.legend()


knn = KNeighborsRegressor(n_neighbors = 1)
fit_and_plot_regression(knn, penguins, data_reg_columns, target_name)

_ = plt.title("Prediction function using 1-NN")
```



Διάγραμμα 10.16: Συνάρτηση πρόβλεψης του μοντέλου 1-nn

Βλέπουμε οι προβλέψεις και η γραμμή πρόβλεψης έχουν αρκετές διακυμάνσεις. Αυτό δεν είναι παράξενο καθώς όταν ο αλγόριθμος ψάχνει να βρει το κοντινότερο σημείο και αυτό μπορεί να είναι αρκετά έξω από τη γραμμή που προβλέπει η γραμμική παλινδρόμηση. Ας δούμε τώρα πώς αλλάζει η γραμμή των προβλέψεων όταν μεγαλώσουμε το k σε 10:

```
knn = KNeighborsRegressor(n_neighbors=10)

fit_and_plot_regression(knn, penguins, data_reg_columns, target_name)

_ = plt.title("Prediction function using 10-NN")
```



Διάγραμμα 10.17: Συνάρτηση πρόβλεψης του μοντέλου 10-nn

Βλέπουμε ότι η γραμμή πρόβλεψης είναι πιο ομαλή σε σχέση με όταν το k = 1. Αυτό είναι λογικό γιατί ο αλγόριθμος θα υπολογίζει τις προβλέψεις παίρνοντας τη μέση τιμή των 10 (αντί για 1) κοντινότερων σημείων του σετ εκπαίδευσης.

10.6. *Ερωτήσεις αυτοαξιολόγησης*

10.1 Ο αλγόριθμος knn είναι πιο απαιτητικός υπολογιστικά κατά την εκπαίδευση παρά κατά τις προβλέψεις

α) Σωστό

β) Λάθος

10.2 Ποια από τις παρακάτω προτάσεις ισχύει για τη μέτρηση απόστασης που χρησιμοποιείται στον αλγόριθμο kNN;

α) Η Ευκλείδεια απόσταση είναι η πιο συχνά χρησιμοποιούμενη μέτρηση απόστασης.

β) Η απόσταση του Μανχάταν ισχύει μόνο για κατηγορικά δεδομένα.

γ) Η μέτρηση της απόστασης δεν επηρεάζει την απόδοση του αλγορίθμου.

δ) Η επιλογή της μέτρησης απόστασης εξαρτάται από τα συγκεκριμένα χαρακτηριστικά του συνόλου δεδομένων και το πρόβλημα που αντιμετωπίζουμε.

10.3 Ποιες από τις παρακάτω μετρικές μέτρησης απόστασης μπορούν να χρησιμοποιηθούν στον knn

α) Manhattan

β) Minkowski

γ) Mahalanobis

δ) όλα τα παραπάνω

10.4 Ποια από τις παρακάτω είναι η Ευκλείδεια Απόσταση μεταξύ δύο σημείων δεδομένων Α(1,3) και Β(2,3);

α) 1

β) 2

γ) 4

δ) 8

Απαντήσεις ερωτήσεων αυτοαξιολόγησης ανά κεφάλαιο

2.1 α 2.2 β 2.3 γ 2.4 β 2.5 δ 2.6 γ

4.1 δ 4.2 δ 4.3 α 4.4 γ 4.5 β

5.1 α 5.2 β 5.3 β 5.4 δ

6.1 β 6.2 β 6.3 α 6.4 α

7.1 δ 7.2 α 7.3 α 7.4 δ

9.1 β 9.2 δ 9.3 α & δ 9.4 α

10.1 β 10.2 δ 10.3 δ 10.4 α

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Αποθετήριο UCI https://archive.ics.uci.edu/dataset/228/sms+spam+collection

  Το αποθετήριο για το αρχικό σύνολο δεδομένων του κεφαλαίου μελέτης περίπτωση Naïve Bayes

  Almeida,Tiago and Hidalgo,Jos. (2012). SMS Spam Collection. UCI Machine Learning Repository. https://doi.org/10.24432/C5CC84.
- https://scikit-learn.org/stable/api/index.html

  Το api της βιβλιοθήκης sklearn, η οποία χρησιμοποιείται για την υλοποίηση των παραδειγμάτων.
- https://scikit-learn.org/stable/api/sklearn.naive_bayes.html

  Το api της βιβλιοθήκης sklearn, για τις κλάσεις Naïve Bayes, η οποία χρησιμοποιείται για την υλοποίηση των παραδειγμάτων.
- https://www.nltk.org/

  Natural Language Toolkit

  Το NLTK είναι μια κορυφαία πλατφόρμα για τη δημιουργία προγραμμάτων Python για εργασία με δεδομένα ανθρώπινης γλώσσας. Παρέχει εύχρηστες διεπαφές σε περισσότερα από 50 σώματα και λεξιλογικούς πόρους, όπως το WordNet, μαζί με μια σουίτα βιβλιοθηκών επεξεργασίας κειμένου για ταξινόμηση, δημιουργία διακριτικών, stemming, σήμανση, ανάλυση και σημασιολογική ανάλυση, wrappers για βιβλιοθήκες NLP, και ενεργό φόρουμ συζήτησης.
- Καμαράτος, Μ. (2022). Εισαγωγή στις Πιθανότητες και τη Στατιστική [Προπτυχιακό εγχειρίδιο]. Κάλλιπος, Ανοικτές Ακαδημαϊκές Εκδόσεις. https://dx.doi.org/10.57713/kallipos-48

  Ακαδημαϊκό σύγγραμα ανοικτής πρόσβασης. Για εμβάθυνση σε πιθανότητες και στατιστική.
- Οικονόμου, Π., Μαλεφάκη, Σ., & Μπατσίδης, Α. (2022). Πιθανότητες – Στατιστική [Προπτυχιακό εγχειρίδιο]. Κάλλιπος, Ανοικτές Ακαδημαϊκές Εκδόσεις. https://dx.doi.org/10.57713/kallipos-101

  Ακαδημαϊκό σύγγραμα ανοικτής πρόσβασης. Για εμβάθυνση σε πιθανότητες και στατιστική.
- Ζωγράφος, Κ., & Τσαϊρίδης, Χ. (2024). Στατιστική και Στοιχεία Πιθανοτήτων [Προπτυχιακό εγχειρίδιο]. Κάλλιπος, Ανοικτές Ακαδημαϊκές Εκδόσεις. https://dx.doi.org/10.57713/kallipos-346
- Ακαδημαϊκό σύγγραμα ανοικτής πρόσβασης. Για εμβάθυνση σε πιθανότητες και στατιστική.
- Κούτρας, Μ., Ευαγγελαράς, Χ., 2010. *Ανάλυση Παλινδρόμησης*. Εκδόσεις Σταμούλης. ISBN 9789603517894

- Παπαδημητρίου, Γ., 2007. *Η Ανάλυση Δεδομένων*. Εκδόσεις Τυπωθήτω. Αθήνα.
- Agresti A. (1996). *An Introduction to Categorical Data Analysis. John Wiley and Sons*, New York, 372 p.
- Cox D. R. & Snell E. J. (1989). *The Analysis of Binary Data, 2nd ed.* Chapman and Hall, London, 236 p.
- Everitt B.S. (1992). *The analysis of contingency tables*. Chapman & Hall, London, 164 p.
- Garson G.B (2011). Ordinal regression. In Statnotes: Topics in Multivariate Analysis. http://faculty.chass.ncsu.edu./garson/pa765/statnote.htm
- Hosmer D.W. & Lemeshow S. (2000*). Applied Logistic Regression. 2nd ed*. John Wiley & Sons, N. Jersey, 373 p.
- McCullagh P. & Nelder J.A. (1989). *Generalized Linear Models. 2nd ed*. Chapman & Hall, London, 511 p.
- Long J.C. & Freese J. (2014). *Regression Models for Categorical Dependent Variables Using Stata, 3rd ed*. College Station: Stata Press, 589 p.