Επιστήμη των δεδομένων

Αλεξάκης Γεώργιος 58093 Κομνηνός Βασίλειος 58051

10 Απριλίου 2023

Περιεχόμενα

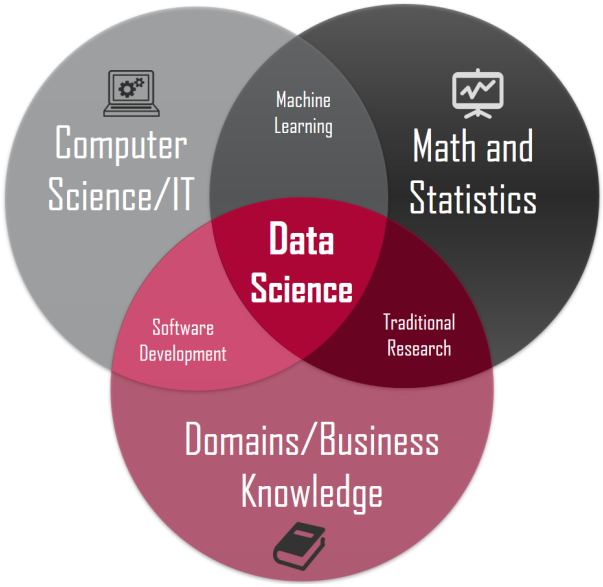
Κατάλογος Σχημάτων

1 Εισαγωγή

Η επιστήμη των δεδομένων (Data science) είναι μια επιστήμη που αφορά την εξαγωγή γνώσης από δεδομένα (δομημένα ή αδόμητα) και για να το πετύχει αυτό συνδυάζει γνώσεις από διάφορες άλλες επιστήμες όπως:

- Μαθηματικά
- Στατιστική
- Προγραμματισμός
- Προγνωστική ανάλυση (Predictive analysis)
- Εξόρυξη δεδομένων (Data mining)
- Τεχνητή Νοημοσύνη (Artificial intelligence)
- Μηχανική μάθηση (Machine learning)

Επιπλέον, η επιστήμη των δεδομένων συνδυάζει τα παραπάνω με τη γνώση κάποιου ειδικού τομέα (όπως η ιατρική) με σκοπό να δώσει λύση σε ένα συγκεκριμένο πρόβλημα [1]



Σχήμα 1: Γνώσεις που συνδυάζει η επιστήμη των δεδομένων

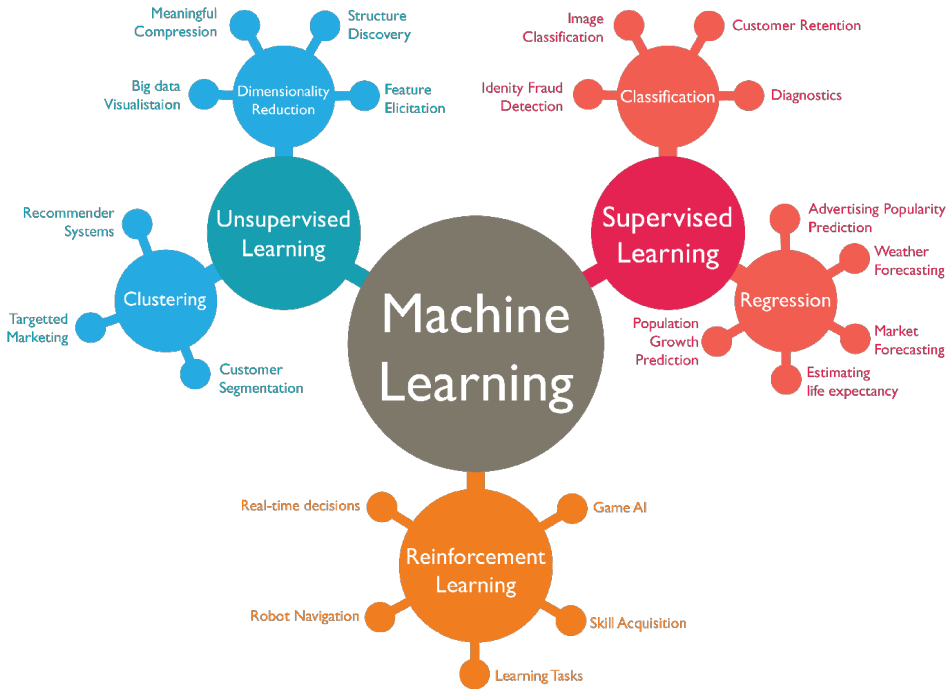
Σε αυτή την εργασία θα ασχοληθούμε κυρίως με την τεχνητή νοημοσύνη και τη μηχανική μάθηση και πιο συγκεκριμένα με τους αλγόριθμους που εφαρμόζονται στην επιστήμη των δεδομένων. Θα δοθεί αναλυτική εξήγηση για πολλούς χρήσιμους αλγόριθμους και θα υλοποιηθούν οι πιο σημαντικοί από αυτούς με τη χρήση της γλώσσας προγραμματισμού C++ η οποία θα μας δώσει τη δυνατότητα να φτιάξουμε γρήγορα προγράμματα.

Οι αλγόριθμοι που θα εξετάσουμε μπορούν να χωριστούν σε πέντε ομάδες με βάση τη χρήση τους και αυτές είναι:

- Αλγόριθμοι ταξινόμησής (Classification)
- Αλγόριθμοι παλινδρόμησης (Regression)
- Αλγόριθμοι ομαδοποίησης (Clustering)

- Αλγόριθμοι μείωσης διαστάσεων (Dimentionality reduction)
- Αλγόριθμοι ενισχυτικής μάθησης (Reinforcement learning)
- Αλγόριθμοι ανίχνευσης ανωμαλίας (Anomaly detection)

Μας ενδιαφέρουν ιδιαίτερα οι πρώτες τρεις ομάδες καθώς είναι αυτές που εμφανίζονται πιο συχνά.Οι παλινδρομητές και οι ταξινομητές είναι πολύ παρόμοιοι αλγόριθμοι οι οποίοι ανήκουν στους αλγόριθμους εποπτευόμενης μάθησης (Supervised learning) και χρησιμοποιούνται για να κάνουν προβλέψεις σύμφωνα με τα δεδομένα που τους έχουμε δώσει. Η διαφορά τους είναι ότι η ταξινόμηση προσπαθεί να προβλέψει την κλάση των καινούριων δεδομένων και να τα κατηγοριοποιήσει σύμφωνα με τις υπάρχουσες κατηγορίες, ενώ η παλινδρόμηση προσπαθεί να προβλέψει την τιμή κάποιου στοιχείου για τα καινούρια δεδομένα σύμφωνα με μια συνάρτηση που έφτιαξε από τα υπάρχοντα δεδομένα. Οι περισσότεροι αλγόριθμοι ταξινόμησης είναι και αλγόριθμοι παλινδρόμησης και το αντίστροφο. Από την άλλη οι αλγόριθμοι ομαδοποίησης είναι μη εποπτευόμενης μάθησης και χρησιμοποιούνται σε αδόμητα δεδομένα για να τα χωρίσουν σε ομάδες.



Σχήμα 2: Αλγόριθμοι μηχανικής μάθησης

2 Ταξινόμηση

2.1 Εισαγωγή στους ταξινομητές

Σε αυτή την ενότητα θα αναλύσουμε τους αλγορίθμους ταξινόμησης και θα δούμε τη χρήση τους. Για τους αλγορίθμους αυτούς χωρίζουμε το σύνολο των δεδομένων μας σε δυο μέρη:

- Δεδομένα εκπαίδευσης (training dataset)
- Δεδομένα επαλήθευσής (testing dataset)

Τα πρώτα τα χρησιμοποιούμε έτσι ώστε ο αλγόριθμος να βρει κάποιο μοτίβο στα δεδομένα με το οποίο θα μπορεί να κατατάσσει τα καινούρια δεδομένα που δέχεται σε κάποια από τις υπάρχουσες κλάσεις. Αυτή είναι και η διαδικασία εκπαίδευσης του μοντέλου. Αφού η εκπαίδευση τελειώσει τότε θα χρησιμοποιήσουμε τα δεδομένα επαλήθευσής για να επιβεβαιώσουμε την ορθή λειτουργία του μοντέλου. Υπάρχουν τρεις βασικοί τύποι ταξινομητών:

- Δυαδικοί (binary)
- Πολλαπλών κλάσεων (multy-class)
- Πολλαπλών ετικετών (multy-label)

Οι δυαδικοί ταξινομητές χρησιμοποιούνται όταν έχουμε μόνο δύο κλάσεις στις οποίες θέλουμε να εντάξουμε τα δεδομένα, ή όταν η απάντηση που θέλουμε να πάρουμε από το μοντέλο είναι δυαδικής φύσης. Για παράδειγμα, ένα πρόβλημα δυαδικής φύσης θα ήταν να προβλέψουμε εάν ένας ασθενής έχει ή δεν έχει μια ασθένειά σύμφωνα με τις εξετάσεις του.

Οι ταξινομητές πολλαπλών κλάσεων από την άλλη είναι ικανοί να αναγνωρίσουν περισσότερες από δύο κλάσεις και είναι πολύ χρήσιμοι για την αναγνώρισή προτύπων. Συνεχίζοντας με το προηγούμενο παράδειγμα θα θέλαμε χωρίσουμε τους ασθενείς σύμφωνα με την κατάσταση τους σε:

- υγιείς
- ήπια ασθένειά
- σοβαρή ασθένεια

΄Ετσι οι γιατροί θα μπορούν αν δράσουν ανάλογα.

Οι ταξινομητές πολλαπλών ετικετών δεν έχουν κάποια καινούρια λογική αλλά εφαρμόζουν τις λογικές των προηγούμενων προβλημάτων. Δηλαδή θα μπορούσαμε να θέλουμε να υλοποιήσουμε ένα μοντέλο το οποίο να κάνει και τις δύο προηγούμενες προβλέψεις που συζητήσαμε. Αυτά τα μοντέλα συνήθως δε χρησιμοποιούν ξεχωριστούς αλγορίθμους αλλά συνδυάζουν πολλούς ήδη γνωστούς αλγορίθμους για να φτάσουν στο αποτέλεσμα.

Σχήμα 3: Τύποι ταξινομητών

Στη συνέχεια θα αναλύσουμε τους διασημότερους αλγορίθμους και τη χρήση τους. Οι αλγόριθμοι αυτοί είναι[2, 3, 4]:

- Naive Bayes
- LDA (Linear Discriminant Analysis)
- QDA (Quadratic Discriminant Analysis)
- SVM (Support Vector Machine)
- k-NN (k Nearest Neighbors)
- Decision trees
- Random Forest
- Νευρωνικά δίκτυα (τα οποία θα αναλύσουμε στην παλινδρόμηση)

2.2 Naive Bayes

Σύμφωνα με το παρακάτω άρθρο [5] ο αλγόριθμος Naive Bayes είναι ένας αλγόριθμος που κάνει την υπόθεση ότι τα χαρακτηριστικά είναι υπό όρους ανεξάρτητα της δεδομένης κλάσης. Αυτή η υπόθεση στην πραγματικότητα δεν ισχύει αλλά ο αλγόριθμος πετυχαίνει πολύ υψηλή ακρίβεια και ταυτόχρονα μεγάλη υπολογιστική απόδοση. Αυτός είναι και ο λόγος που χρησιμοποιείται τόσο συχνά στην επιστήμη των δεδομένων.

Τα πλεονεκτήματα του αλγορίθμου είναι[6]:

- Η χρονική πολυπλοκότητα αυξάνεται γραμμικά με το πλήθος των δειγμάτων και των στοιχείων τους
- Χαμηλό variance (η αλλαγή των δεδομένων δεν επηρεάζει πολύ το μοντέλο)
- Μπορούμε εύκολα να προσθέσουμε και άλλα δεδομένα και το μοντέλο θα συνεχίσει την εκπαίδευση χωρίς πρόβλημα
- Δεν είναι επιρρεπής στον θόρυβο
- Δεν επηρεάζεται από έλλειψη τιμών στα δεδομένα

Τα μειονεκτήματά είναι:

- Η υπόθεση ότι τα δεδομένα είναι ανεξάρτητα τον κάνει ακατάλληλο για ορισμένα προβλήματα
- Οι συνδυασμοί στοιχείων που δεν εμφανίζονται στο σύνολο δεδομένων για κάποια πρόβλεψη θα έχει πάντα μηδενική πιθανότητα.
- Οι πιθανότητες που υπολογίζει ενδέχεται να είναι λάθος

Ο αλγόριθμος χρησιμοποιεί τον κανόνα του Bayes:

$$P(y|X) = \frac{P(y) \times P(X|y)}{P(X)}$$

Ο παραπάνω τύπος θα μας δώσει την πιθανότητα ενός δείγματος με στοιχεία X να ανήκει στην κλάση y. Το X είναι διάνυσμα με όλα τα στοιχεία του δείγματος μας. Στο παρακάτω παράδειγμα το X για το πρώτο δείγμα θα είναι $\langle 40, 85 \rangle$ και η κλάση θα είναι $y = 0$. Σκοπός μας είναι όταν πάρουμε τα

στοιχεία από έναν ασθενή να μπορούμε να προβλέψουμε αν έχει διαβήτη. Θα πρέπει δηλαδή να υπολογίσουμε τις πιθανότητες $P\left(y=0|X=\langle x_1,x_2\rangle\right)$ και $P\left(y=1|X=\langle x_1,x_2\rangle\right)$ για τον καινούριο ασθενή με μετρήσεις $x1, x2$ και η πρόβλεψη θα είναι αυτή με τη μεγαλύτερη πιθανότητα.

Γλυκόζη	Αρτηριακή πίεση	Διαβήτης
40	85	0
40	92	0
45	63	1
45	80	0
40	73	1
45	82	0
40	85	0
30	63	1
65	65	1
45	82	0
35	73	1
45	90	0
50	68	1
40	93	0
35	80	1
50	70	1

Πίνακας 1: Δείγματα για πρόβλεψη διαβήτη

Μέχρι εδώ ήταν η γνωστή στατιστική. Αλλά όσα περισσότερα στοιχεία έχουμε ανά δείγμα τόσο πιο πολύπλοκος γίνεται ο υπολογισμός του παράγοντα $P(X|y)$. Εδώ δίνει λύση στο πρόβλημα ο αλγόριθμος Naive Bayes με την ανεξαρτησία των στοιχείων. Ανεξαρτησία σημαίνει ότι

$$P(\langle x_1, x_2, \ldots, x_n\rangle|y) = \prod_{i=1}^{n} P(x_i|y)$$

Επίσης το $P(X)$ είναι πάντα σταθερό για τα δεδομένα στοιχεία που έχουμε άρα τελικά πρέπει να υπολογίσουμε το

$$P(y) \times \prod_{i=1}^{n} P(x_i|y)$$

για κάθε y και στο τέλος να διαλέξουμε εκείνο το y που έδωσε τη μεγαλύτερη τιμή (που θα είναι και η κλάση στην οποία ανήκει το δείγμα). Βέβαια, αν τα στοιχεία δεν είναι υπό όρους ανεξάρτητα δεν ισχύει αυτή η απλοποίηση. Παρ΄όλα αυτά μπορεί να χρησιμοποιηθεί σε πολλά σύνολα δεδομένων παρόλο που κάποιες φορές δεν ισχύει και για αυτόν τον λόγο καλείται και αφελής Bayes.

Ο αλγόριθμος είναι ιδανικός για προβλήματα πολλαπλών κλάσεων λόγω του εύκολου υπολογισμού. Πιο συγκεκριμένα ο Naive Bayes χρησιμοποιείται:

- Στην ανίχνευση spam. Μπορεί διαβάζοντας τις λέξεις ενός mail να βρει την πιθανότητα να είναι spam. Κάθε λέξη που διαβάζει θα αυξάνει ή θα μειώνει την πιθανότητα με κάποιον προκαθορισμένο κανόνα μέχρι που στο τέλος θα έχουμε πάρει μια ικανοποιητική απάντηση.

- Στη συναισθηματική ανάλυση. Οι εταιρίες μπορούν να κατατάξουν τα σχόλια των καταναλωτών τους σε θετικά και αρνητικά χρησιμοποιώντας τον αλγόριθμο. Μετά μπορούν να πράξουν ανάλογα έτσι ώστε να ευνοήσουν την εταιρία.

Υπάρχουν πολλές άλλες χρήσεις πέρα από αυτές. Για μια πιο αναλυτική εξήγηση των μαθηματικών του αλγορίθμου μπορείτε να διαβάσετε την εξής έρευνα [7].

Οι αλγόριθμοι QDA και QDA χρησιμοποιούν επίσης τον τύπο του Bayes. Η διαφορά είναι στον τρόπο υπολογισμού του παράγοντα $P(X|y)$. Όπως και Ο Αλγόριθμος Naive Bayes κάνει μια υπόθεση για να τον υπολογίσει, το ίδιο κάνουν και αυτοί οι αλγόριθμοι. Για να δείτε αυτούς τους τύπους με πλήρη εξήγηση μπορείτε να δείτε το παράρτημα Α.

2.3 Support Vector Machine

Οι μηχανές διανυσμάτων υποστήριξης (SVM) είναι ένα σύνολο μεθόδων εποπτευόμενης μάθησης μέθοδοι που χρησιμοποιούνται για ταξινόμηση και παλινδρόμηση. Ανήκουν σε μια οικογένεια γενικευμένων γραμμικών ταξινομητών. Ένα βασικό χαρακτηριστικό του είναι ότι είναι κάνει προβλέψεις που χρησιμοποιούν τη θεωρία μηχανικής μάθησης για να μεγιστοποιήσει την προγνωστική ακρίβεια ενώ ταυτόχρονα αποφεύγει την υπερβολική προσαρμογή στα δεδομένα (over-fitting)[8].

Τα πλεονεκτήματα του SVM είναι[9]:

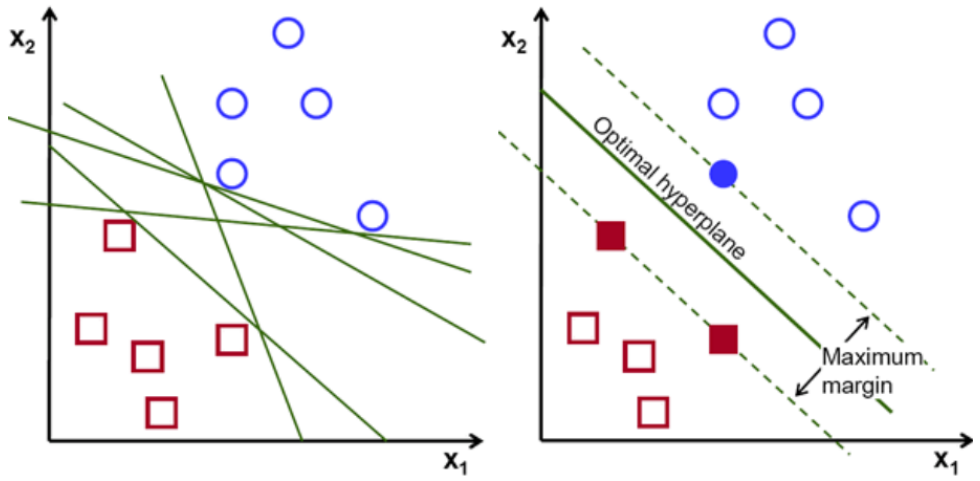
- Δουλεύει πολύ καλά όταν υπάρχει καθαρή διαφοροποίηση στα δεδομένα
- Δουλεύει καλύτερα σε μεγαλύτερες διστάσεις (πολλά στοιχεία ανά δείγμα)
- Είναι αποδοτικός όταν τα στοιχεία ανά δείγμα είναι περισσότερα από τα δείγματα
- Δεν κάνει μεγάλη χρήση μνήμης

Μειονεκτήματά:

- Δεν είναι κατάλληλος για μεγάλο σύνολο δεδομένων
- Είναι επιρρεπής στον θόρυβο
- Λόγω του τρόπου υλοποίησης του δεν μπορούμε να πάρουμε την πιθανότητα αλλά μόνο την πρόβλεψη

Ο αλγόριθμος αυτός είναι αρκετά απλός και χρησιμοποιείται κυρίως για δυαδική ταξινόμηση. Αρχικά για να τον καταλάβουμε θα πρέπει να αναπαραστήσουμε τα δεδομένα μας σαν σημεία σε έναν χώρο N διαστάσεων, όπου N τα στοιχεία που έχουμε ανά δείγμα. Το κάθε στοιχείο θα ανήκει σε μια από τις δύο κατηγορίες τις οποίες θέλουμε να αναγνωρίσουμε. Ο αλγόριθμος λοιπόν προσπαθεί να φέρει ένα υπερεπίπεδο τέτοιο ώστε να χωρίζει τα σημεία ανάλογα με την κατηγορία τους. Έτσι τα σημεία της μιας κατηγορίας θα βρίσκονται από τη μια μεριά και της δεύτερης από την άλλη.

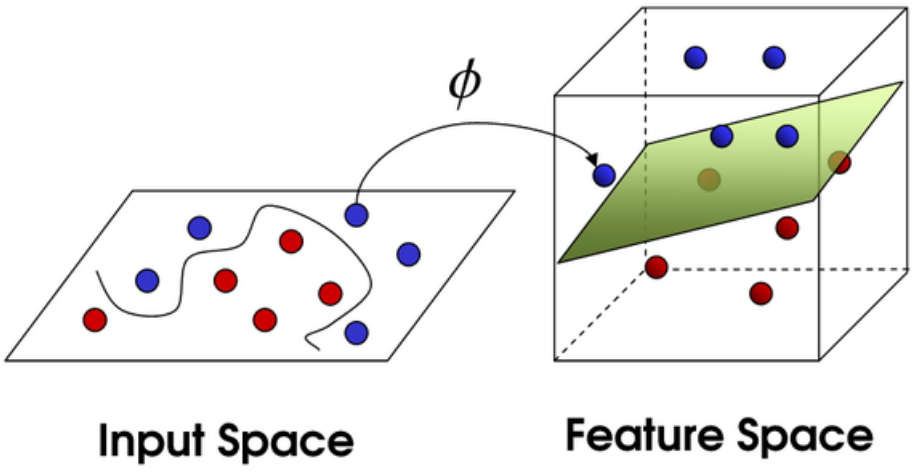
Για να κατανοήσουμε καλύτερα τον αλγόριθμο θα κάνουμε ένα παράδειγμα σε έναν χώρο δύο διαστάσεων που μπορεί εύκολα να αντιληφθεί το ανθρώπινο μάτι.



Σχήμα 4: Παράδειγμα SVM

Στον χώρο δύο διαστάσεων το υπερεπίπεδο είναι απλώς μια ευθεία. Εμείς λοιπόν θέλουμε να βρούμε μια ευθεία που να χωρίζει τα τετράγωνα με τους κύκλους. Όπως μπορούμε να δούμε από το πρώτο διάγραμμα υπάρχουν πολλές ευθείες που μπορεί να χωρίζουν τα σημεία μας αλλά εμείς θέλουμε να διαλέξουμε την ευθεία που να έχει τη μέγιστη απόσταση και από τις δύο μεριές. Επομένως, όταν δώσουμε ένα καινούριο δείγμα στον αλγόριθμο και θέλουμε να το κατατάξει σε μια από τις δύο ομάδες πρέπει απλά να το τοποθετήσει σε αυτόν τον χώρο και μετά να δει από ποια μεριά της ευθείας βρίσκεται. Ανάλογα με τη μεριά αυτή θα αποφασίσουμε και την κλάση του αντικειμένου.

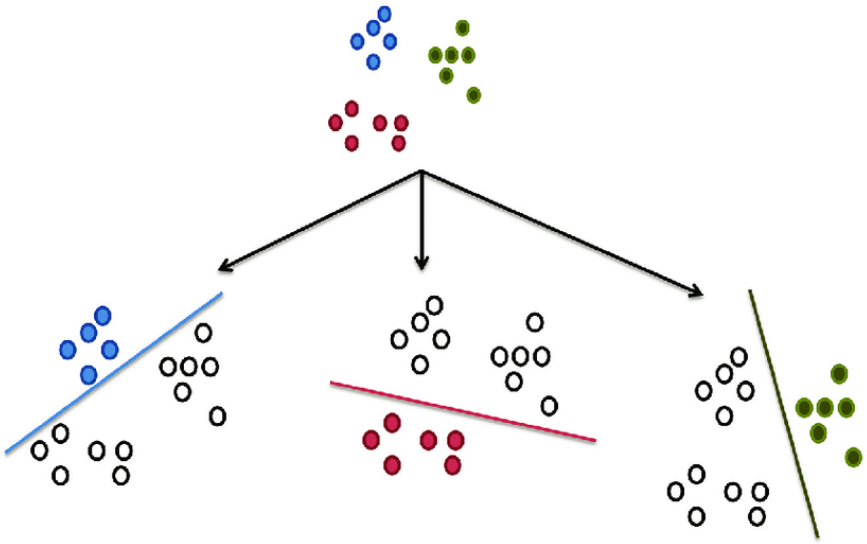
Λύσαμε το πρόβλημα όπου είχαμε πάνω από μία ευθεία αλλά υπάρχει ένα μεγαλύτερο πρόβλημα. Αυτό είναι η περίπτωση που δεν μπορούμε να χωρίσουμε τα στοιχεία μας με μια ευθεία. Εδώ έρχεται το λεγόμενο ¨kernel trick¨ για να λύσει το πρόβλημα. Το kernel είναι μια συνάρτηση σύμφωνα με την οποία επαυξάνουμε τη διάσταση του χώρου μας έτσι ώστε με την καινούρια διάσταση που θα δημιουργηθεί τα σημεία μας να είναι διαχωρίσιμα από πλέον ένα επίπεδο. Όταν γυρίσουμε πίσω στην αρχική διάσταση θα φαίνεται σαν τα σημεία μας να είναι διαχωρισμένα με μια καμπύλη.



Σχήμα 5: Επαύξηση διάστασης με SVM

Ένα τελευταίο πρόβλημα που θα μπορούσαμε να λύσουμε θα ήταν να τροποποιήσουμε τον αλγόριθμο έτσι ώστε να δουλεύει για πάνω από δύο κλάσεις. Ένας τρόπος να το κάνουμε αυτό θα ήταν με τη μέθοδο ONE versus ALL. Αυτό που κάνουμε είναι να φέρουμε μια ευθεία για κάθε κλάση που έχουμε η οποία θα χωρίζει τον χώρο στα στοιχεία που ανήκουν σε αυτήν την κλάση και σε όλα τα

υπόλοιπα. Έτσι συνδυάζοντάς την πληροφορία και από τις τρεις ευθείες μπορούμε να πάρουμε τη σωστή απάντηση.



Σχήμα 6: Ταξινόμηση πολλαπλών κλάσεων με SVM

Βέβεα κάνοντας αυτή τη μέθοδο θα εμφανιστουν σημεία στον χώρο τα οποία μπορεί να ανοίκουν σε πάνω από μια κλάση και και άλλα που δεν θα ανήκουν σε καμία. Θα πρέπει λοιπόν να λάβουμε υπόψη μας όχι μόνο από ποια μεριά της κάθε ευθείας βρισκόμαστε αλλά και την απόσταση από αυτές για να μπορέσουμε να καταλήξουμε σε ένα σωστό συμπέρασμα.

Μερικές χρήσεις του αλγορίθμου είναι[10]:

- Ανίχνευση προσώπου
- Κατηγοριοποίηση κειμένου
- Ταξινόμηση εικόνων
- Βιοπληροφορική (πχ. ανίχνευση καρκίνου)
- Αναγνώριση γραφικού χαρακτήρα
- Γενικευμένος προγνωστικος έλεγχος (έλεγχος χαοτικών συστημάτων)

2.4 K Nearest Neighbors

Ο αλγόριθμος k-NN είναι ένας μη παραμετρικός αλγόριθμος ο οποίος χρησιμοποιεί την εγγύτητα για να κατατάξει τα καινούρια στοιχεία σε μια από τις υπάρχουσες κλάσεις[11]. Αυτό σημαίνει ότι το κάθε καινούριο στοιχείο θα παίρνει την κλάση του βλέποντας τις κλάσεις των πιο κοντινών γειτόνων του διαλέγοντας την κλάση της πλειοψηφίας.

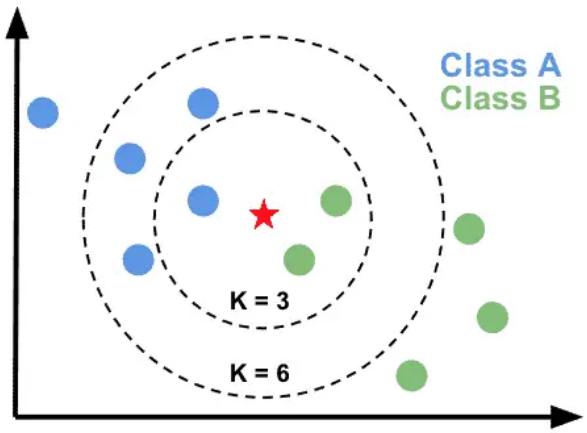
Τα πλεονεκτήματα του k-NN είναι:

- Εύκολη υλοποίηση
- Εύκολη προσαρμογή κατά την εμφάνιση καινούριων δεδομένων
- Δε χρειάζεται παραμέτρους παρά μόνο το k και μια μετρική απόστασης

Κάποια μειονεκτήματά του είναι:

- Δεν έχει καλή επεκτασιμότητα
- Δε δουλεύει καλά με μεγάλες διαστάσεις
- Μπορεί να γίνει εύκολα over fitting

Για να ξεκινήσουμε τον αλγόριθμο πρέπει πρώτα να ορίσουμε ένα k. Αυτό θα μας λέει πόσα γειτονικά δείγματα να λάβουμε υπόψη μας για τον προσδιορισμό της κλάσης του καινούριου δείγματος. Έπειτα βρίσκουμε τις αποστάσεις του σημείου από όλα τα σημεία με μια από τις μετρικές που θα αναλύσουμε παρακάτω και διαλέγουμε τα k σημεία με την κοντινότερη απόσταση στο δικό μας. Τέλος, η κλάση του σημείου μας θα είναι η κλάση που έχει η πλειοψηφία των k σημείων.



Σχήμα 7: Αλγορίθμος k-NN

Για της μετρικές απόστασης γενικότερα στην επιστήμη των δεδομένων και στη μηχανική μάθηση έχουμε τέσσερις βασικές επιλογές:

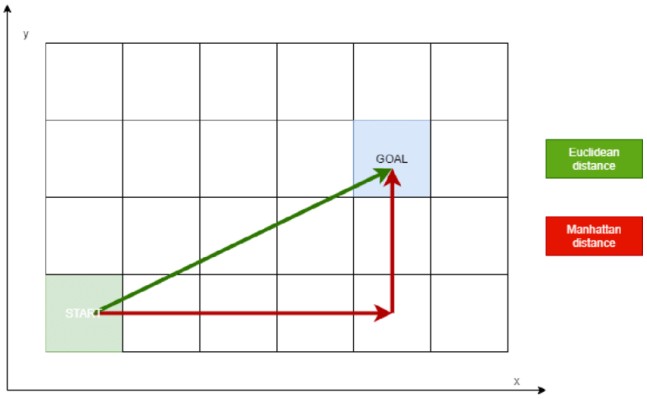
- Ευκλείδεια απόσταση
- Απόσταση Manhattan
- Απόσταση Minkowski
- Απόσταση Hamming

Η Ευκλείδεια απόσταση είναι η πιο γνωστή μετρική και γραφικά είναι το μήκος της ευθείας γραμμής που ενώνει τα δύο σημεία μεταξύ τους. Δίνεται από τον γνωστό τύπο:

$$D = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Η απόσταση Manhattan είναι επίσης πολύ γνωστή την οποί μπορούμε να παραστήσουμε γραφικά όπως το παρακάτω σχήμα και έχει τύπο:

$$D = \sum_{i=1}^{n}|x_i - y_i|$$

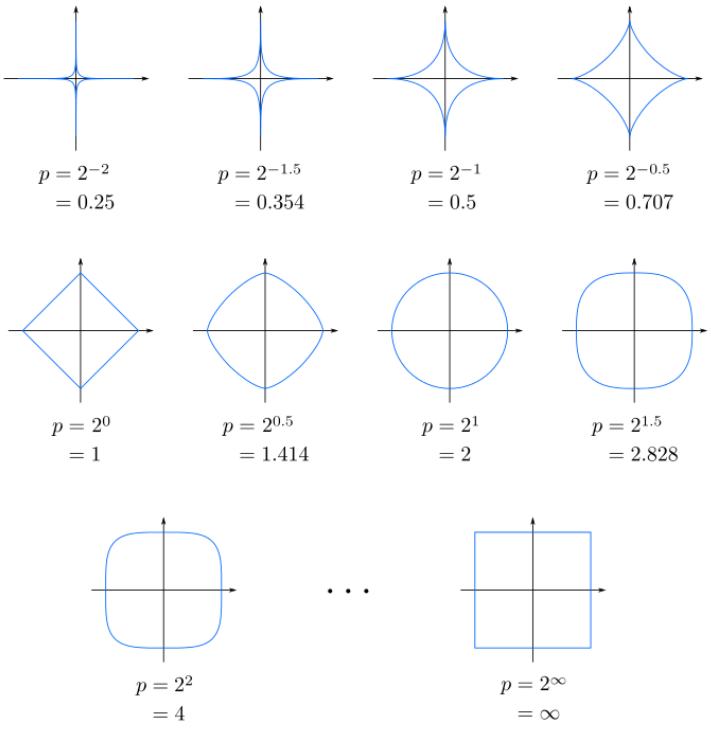


Σχήμα 8: Ευκλείδεια απόσταση και Απόσταση Manhattan

Η πόσταση Minkowski είναι ένα σύνολο αποστάσεων που αποτελεί μια γενίκευση των παραπάνω δύο με τύπο:

$$D = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

Όπως μπορούμε αν δούμε πως για $p = 1$ και $p = 2$ παίρνουμε την απόσταση Manhattan και την ευκλείδεια απόσταση αντίστοιχα. Δεν μπορούμε να την αναπαραστήσουμε γραφικά αλλά για περαιτέρω κατανόησή μπορούμε να δούμε το παρακάτω σχήμα όπου όλα τα σημεία στην μπλε γραμμή ισαπέχουν από το κέντρο τους με μετρική την απόσταση Minkowski για διαφορετικά p



Σχήμα 9: Απόσταση Minkowski

Τέλος, η απόσταση Hamming είναι πολύ απλή και εφαρμόζεται σε δυαδικά διανύσματα και είναι

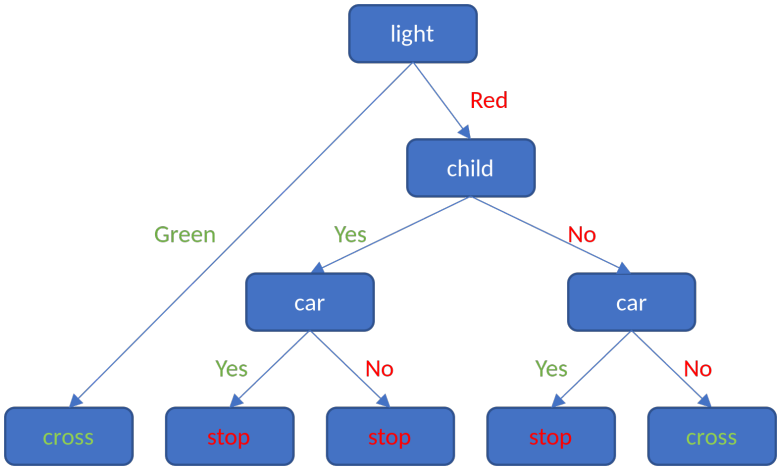
το πλήθος των συνιστωσών που είναι διαφορετικές μεταξύ τους. Για παράδειγμα, αν έχουμε τα διανύσματα $x = \langle 0, 1, 1, 0, 1 \rangle, y = \langle 1, 1, 0, 0, 0 \rangle$ η απόσταση τους θα είναι 3 γιατί οι πρώτες, τρίτες και πέμπτες συνιστώσες είναι διαφορετικές μεταξύ τους.

Οι χρήσεις του k-NN είναι πολύ παρόμοιες με αυτές του SVM. Μια ακόμα χρήση του είναι στην επιλογή προτεινόμενων που υπάρχουν σε πολλές μηχανές αναζήτησης. Μπορεί να βρει τις προτάσεις που θα ταίριαζαν στον κάθε χρήστη ανάλογα με τα ¨κλικ¨ που κάνει βρίσκοντας έτσι τα ενδιαφέροντα του.

2.5 Decision Trees

Τα δέντρα αποφάσεων είναι μη παραμετρικοί αλγόριθμοι που χρησιμοποιούνται για ταξινόμησή και παλινδρόμηση. Η αναπαράσταση τους γίνεται με τη μορφή δέντρου με κόμβους που χωρίζονται σε κόμβους αποφάσεων και στους τελικούς κόμβους.

Κάθε δέντρο ξεκινάει από έναν κόμβο απόφασης με δύο ή περισσότερους δρόμους που οδηγούν σε άλλους. Ο κάθε κόμβος απόφασης παρουσιάζει μια συνθήκη για τα δεδομένα και ο δρόμος που θα διαλέξουμε εξαρτάτε από το αν η συνθήκη είναι αληθείς ή όχι. Αν συνεχίσουμε αυτή τη διαδικασία κάποια στιγμή θα καταλήξουμε σε έναν τελικό κόμβο ο οποίος θα μας δίνει σαν απάντηση μία από τις κλάσεις που έχουμε ορίσει για το σύνολο δεδομένων.



Σχήμα 10: Παράδειγμα δέντρου απόφασης

Για παράδειγμα στο σχήμα 10 θέλουμε ο υπολογιστής να πάρει την απόφαση για το αν ο άνθρωπος πρέπει να περάσει τον δρόμο ή όχι. Για να το κάνει αυτό περνάει από πολλά στάδια αποφάσεων πριν καταλήξει στην τελική του απόφαση.

Στην πραγματικότητα όμως τα δεδομένα δε θα είναι τόσο απλά. Τα δεδομένα μπορεί να μη διαχωρίζονται με τις συνθήκες που έχουμε επιλέξει για τους κόμβους απόφασης. Αυτό όμως είναι ένα πρόβλημα γιατί ένας τελικός κόμβος θα πρέπει να μας υποδεικνύει μόνο μία κλάση που σημαίνει ότι το σύνολο των συνθηκών που περάσαμε για να φτάσουμε εκεί θα πρέπει να ισχύει μόνο για εκείνη την κλάση.

Για να μπορέσει ο υπολογιστής να βρει τις βέλτιστες συνθήκες για κάθε κόμβο πρέπει να χρησιμοποιήσουμε τη θεωρεία της πληροφορίας. Θα κάνουμε ένα παράδειγμα για καλύτερη κατανόηση του αλγορίθμου.

΄Εστω ότι έχουμε ένα πρόβλημα δυαδικής ταξινόμησης και έναν κόμβο απόφασης που στον οποίο έχουμε 20 στοιχεία και τα σημεία είναι διαχωρισμένα στις 2 κλάσεις στη μέση ($50\% - 50\%$). Τώρα θέλουμε να επιλέξουμε ανάμενα σε 2 συνθήκες:

- Συνθήκη Α: Χωρίζει τα σημεία σε 2 κόμβους όπου:

  - κόμβος 1: Έχει 14 σημεία με διαχωρισμό $42.8\% - 57.2\%$
  - κόμβος 2: Έχει 6 σημεία με διαχωρισμό $33.3\% - 66.7\%$

- Συνθήκη Β: Χωρίζει τα σημεία σε 2 κόμβους όπου:
  - κόμβος 1: Έχει 4 σημεία με διαχωρισμό $0\% - 100\%$
  - κόμβος 2: Έχει 16 σημεία με διαχωρισμό $37.5\% - 62.5\%$

Μπορούμε να υπολογίσουμε την εντροπία από τον τύπο:

$$E = \sum_{i=1}^{n} -p_i \log_2 p_i$$

όπου p_i η πιθανότητα της κλάσης i. Άρα έχουμε:

$$E_{parent} = -0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$$

$$E_{A1} = -0.428 \log_2 0.428 - 0.572 \log_2 0.572 = 0.99$$
$$E_{A2} = -0.333 \log_2 0.333 - 0.667 \log_2 0.667 = 0.91$$
$$E_{B1} = -0 \log_2 0 - 1 \log_2 1 = 0$$
$$E_{B2} = -0.375 \log_2 0.375 - 0.625 \log_2 0.625 = 0.95$$

Παρατηρούμε ότι όσο καλύτερος είναι ο διαχωρισμός των στοιχείων, τόσο μικρότερη είναι η εντροπία. Το κέρδος πληροφορίας για τις 2 συνθήκες είναι:

$$I_A = E_{parent} - \frac{14}{20} E_{A1} - \frac{6}{20} E_{A2} = 0.034$$

$$I_B = E_{parent} - \frac{4}{20} E_{B1} - \frac{16}{20} E_{B2} = 0.24$$

Ισχύει ότι $I_B > I_A$ και άρα θα προτιμήσουμε τη συνθήκη Β. Αυτός ο έλεγχος θα πρέπει να γίνει για όλες τις πιθανές συνθήκες και για όλους τους κόμβους ξεκινώντας από τον αρχικό κόμβο. Είναι σημαντικό να αναφέρουμε ότι αυτός είναι ένας άπληστος αλγόριθμος διότι επιλέγει την καλύτερη συνθήκη για έναν κόμβο αλλά μπορεί μια χειρότερη επιλογή να οδηγούσε σε καλύτερο αποτέλεσμα αν βλέπαμε τη συνολική εικόνα.

Ένα πρόβλημα με τα δέντρα είναι ότι έχουν μεγάλο variance και γι΄ αυτό τον λόγο πολλές φορές δεν μπορούν να γενικευτούν για καινούρια δεδομένα. Έτσι σαν μια εξέλιξη του αλγορίθμου δημιουργήθηκε ο αλγόριθμος του τυχαίου δάσους (Random Forest). Ο αλγόριθμος αυτός είναι συνδυασμός μεθόδων και όπως φαίνεται από το όνομα του αποτελείται από πολλά decision trees. Μετά για την τελική απόφαση αν έχουμε πρόβλημα ταξινόμησης θα επιλέξουμε την κλάση που δείχνει η πλειοψηφία, ενώ αν έχουμε πρόβλημα παλινδρόμησης θα πάρουμε τον μέση τιμή των απαντήσεων[12].

Το παραπάνω εξηγεί τη λέξη δάσος όμως δεν εξηγεί τη λέξη τυχαίο. Ο λόγος που είναι τυχαίο είναι επειδή το κάθε δέντρο δε συμπεριλαμβάνει όλα τα δείγματα αλλά ένα τυχαίο τμήμα αυτών και επιπλέον για αυτά τα δείγματα δε θα έχουν όλα τα χαρακτηριστικά τους αλλά θα επιλεχθούν κάποια τυχαία. Παρακάτω βλέπουμε ένα παράδειγμα για το πώς θα χωρίζαμε τα δεδομένα μας για ένα τυχαίο δάσος που θα αποτελείται από τέσσερα δέντρα.

id	x_0	x_1	x_2	x_3	x_4	y
0	4.3	4.9	4.1	4.7	5.5	0
1	3.9	6.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1

Πίνακας 2: Σύνολο δεδομένων για τον αλγόριθμο random forest

id
2
0
2
4
5
5

id
2
1
3
1
4
4

id
4
1
3
0
0
2

id
3
3
2
5
1
2

Πίνακας 3: Σύνολο δεδομένων για κάθε δέντρο

Επίσης, όπως είπαμε το κάθε υποσύνολο δεδομένων δε θα περιέχει όλα τα στοιχεία. Για παράδειγμα, το πρώτο θα μπορούσε να έχει μόνο τα στοιχεία x_1, x_2, το δεύτερο τα x_1, x_4 κ.ο.κ.

Τα πλεονεκτήματα των δέντρων είναι ότι δε χρειάζεται προ επεξεργασία των δεδομένων. Στα δεδομένα πολλές φορές λείπουν στοιχεία που πρέπει ή μπορεί να χρειάζεται κάποια κανονικοποίηση για να εφαρμόσουμε κάποιους αλγόριθμους. Τα δέντρα όμως δεν είναι ένας από αυτούς και αυτό τα κάνει πολύ καλή επιλογή σε τέτοιες περιπτώσεις. Επιπλέον, όπως είπαμε είναι μη παραμετρικοί και επίσης δεν είναι γραμμικοί που είναι κάτι που θέλουμε πολλές φορές για καλύτερο διαχωρισμό των κλάσεων[13]. Το βασικό πρόβλημα του αλγορίθμου είναι το overfitting το οποίο λύνει ο αλγόριθμος random forest.

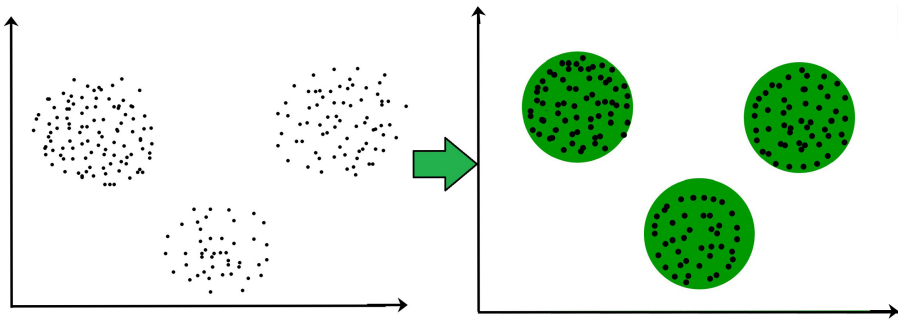
3 Παλινδρόμηση

4 Ομαδοποίηση

4.1 Εισαγωγή στην ομαδοποίηση

Η ενότητα αυτή αφορά τους αλγορίθμους ομαδοποίησης(clustering). Η δουλειά των αλγορίθμων αυτών είναι να χωρίσουν το σύνολο των δεδομένων σε ομάδες τέτοιες ώστε τα δεδομένα μιας ομάδας να εμφανίζουν περισσότερες ομοιότητες μεταξύ τους από ότι με τα δεδομένα των άλλων ομάδων. Αυτός είναι και ο βασικός στόχος της διερευνητικής και στατιστικής ανάλυσης και χρησιμοποιείται για[14]:

- Αναγνώριση προτύπων
- Ανάλυση εικόνων
- Εξαγωγή πληροφορίας
- Συμπίεση δεδομένων
- Γραφικά υπολογιστών
- Μηχανική μάθηση



Σχήμα 11: Γραφική αναπαράσταση ομαδοποίησης

Υπάρχουν εκατοντάδες αλγόριθμοι ομαδοποίησης οι οποίοι χρησιμοποιούν διαφορετικές μεθόδους και τεχνικές. Δεν υπάρχει κάποιος που να υπερτερεί πλήρως των υπολοίπων αλλά η καταλληλότητα του Κάθε αλγορίθμου εξαρτάται από το σύνολο δεδομένων. Παρά τις διαφορές τους μπορούμε να τους εντάξουμε σε κατηγορίες συμφωνά με τον τρόπο που κάνουν την ομαδοποίηση. Οι τέσσερεις πιο βασικές κατηγορίες είναι:

- Centroid based clustering
- Density based clustering
- Distribution based clustering
- Hierarchical clustering

Οι κατηγορίες αυτές θα αναλυθούν περισσότερο στη συνέχεια. Επιπλέον, θα δούμε και τους πιο διάσημους αλγορίθμους που ανήκουν σε αυτές της κατηγορίες και θα αναλύσουμε και τον τρόπο λειτουργίας τους.

4.2 Centroid based clustering

4.3 Density based clustering

4.4 Distribution based clustering

4.5 Hierarchical clustering

5 Συμπέρασμα

Αναφορές

[1] Wikipedia. https://en.wikipedia.org/wiki/Data_science. Accessed: 2023-03-13.

[2] Rohit Garg. 7 types of classification algorithms. https://analyticsindiamag.com/7-types-classification-algorithms/. Accessed: 2023-03-15.

[3] Machine learning classification - 8 algorithms for data science aspirants. https://data-flair.training/blogs/machine-learning-classification-algorithms/. Accessed: 2023-03-15.

[4] Ekin Keserer. 8 types of machine learning classification algorithms. https://www.akkio.com/post/5-types-of-machine-learning-classification-algorithms. Accessed: 2023-03-15.

[5] Webb, Geoffrey I and Keogh, Eamonn and Miikkulainen, Risto. Naïve Bayes. *Encyclopedia of machine learning*, 15:713–714, 2010.

[6] Pavan Vadapalli. Naive Bayes explained: Function, Advantages and disadvantages applications in 2023. https://www.upgrad.com/blog/naive-bayes-explained/. Accessed: 2023-03-18.

[7] Rish, Irina and others. An empirical study of the naive Bayes classifier. 3(22):41–46, 2001.

[8] Jakkula, Vikramaditya. Tutorial on support vector machine (svm). *School of EECS, Washington State University*, 37(2.5):3, 2006.

[9] Dhiraj K. Top 4 advantages and disadvantages of Support Vector Machine or SVM. https://dhirajkumarblog.medium.com/top-4-advantages-and-disadvantages-of-support-vector-machine-or-svm-a3c06a2b107. Accessed: 2023-03-18.

[10] Real-Life Applications of SVM (Support Vector Machines). https://data-flair.training/blogs/applications-of-svm/. Accessed: 2023-03-22.

[11] What is the k-nearest neighbors algorithm? https://www.ibm.com/topics/knn. Accessed: 2023-03-24.

[12] Random Forest. https://en.wikipedia.org/wiki/Random_forest. Accessed: 2023-04-08.

[13] Advantages and disadvantages of decision tree in machine learning. https://www.analytixlabs.co.in/blog/decision-tree-algorithm. Accessed: 2023-04-08.

[14] Cluster analysis. https://en.wikipedia.org/wiki/Cluster_analysis. Accessed: 2023-04-10.

Α΄ Τύποι για QDA, LDA

Στον αλγόριθμο QDA ο παράγοντας $P(X|y)$ δίνεται από τον τύπο:

$$P(X|y=k) = \frac{1}{(2\pi)^{\frac{n}{2}} \left|\sum_k\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(X-M_k)^T \sum_k^{-1}(X-M_k)}$$

όπου:

k η κλάση που εξετάζουμε

X το διάνυσμα χαρακτηριστικών του δείγματος

M_k το κέντρο των σημείων της κλάσης k

n η διάσταση του X

$\sum_k$ ο πίνακας συνδιακύμανσης

και

$$X = \langle x_1, x_2, \ldots, x_n \rangle$$

αν τα σημεία που ανήκουν στην κλάση k είναι $K_1, K_2, \ldots, K_m$ με:

$$K_i = \langle k_{i_1}, k_{i_2}, \ldots, k_{i_n} \rangle$$

τότε το M για αυτή την κλάση θα είναι:

$$M = \langle m_1, m_2, \ldots, m_n \rangle = \left\langle \frac{1}{m}\sum_{i=1}^{m} k_{i_1}, \frac{1}{m}\sum_{i=1}^{m} k_{i_2}, \ldots, \frac{1}{m}\sum_{i=1}^{m} k_{i_n} \right\rangle$$

Η διακύμανση είναι μας δείχνει τη διασπορά των σημείων της κλάσης από το κέντρο της κλάσης για μια συγκεκριμένη διάσταση και υπολογίζεται για κάθε διάσταση ξεχωριστά. Η διακύμανση για μια διάσταση u από τις n διαστάσεις θα είναι:

$$V_u = \frac{1}{m}\sum_{i=1}^{m} \left(k_{i_u} - m_u\right)^2$$

Πολύ παρόμοια είναι και η συνδιακύμανση η οποία όμως υπολογίζεται για δύο διαστάσεις u και z:

$$C_{uz} = \frac{1}{m}\sum_{i=1}^{m} \left(k_{i_u} - m_u\right)\left(k_{i_z} - m_z\right)$$

Ο πίνακας συνδιακύμανσης είναι ένας συμμετρικός πίνακας με διαστάσεις $n \times n$ και είναι:

$$\sum = \begin{bmatrix} V_1 & C_{12} & \ldots & C_{1n} \\ C_{12} & V_2 & \ldots & C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{1n} & C_{2n} & \ldots & V_n \end{bmatrix}$$

Ο αλγόριθμος LDA είναι μια απλοποίηση του QDA όπου θεωρούμε ότι όλες οι κλάσεις έχουν τον ίδιο πίνακα συνδιακύμανσης το οποίο απλοποιεί πάρα πολύ τις πράξεις.