M.Sc. Data Science
Σχολή Πληροφορικής
Μητροπολιτικό Κολλέγιο Αθηνών - University of East London

ΤΙΤΛΟΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ:

“ Ανάπτυξη εφαρμογής για εποπτευόμενη και μη εποπτευόμενη μηχανική μάθηση και αξιοποίηση αυτής σε σύνολο δεδομένων καταγραφής ενεργειακών χαρακτηριστικών κτιρίου.”

Συγγραφέας: Nikos Kougianos
UEL Number: 2018506

Επιβλέπων καθηγητής: Dr Dimitris Sklavounos

Αθήνα, 2021

Περίληψη

Στην παρούσα πτυχιακή εργασία επιχειρούμε να προσεγγίσουμε από μια θεωρητική σκοπιά τις άκρως ενδιαφέρουσες επιστήμες των δεδομένων και της μηχανικής εκμάθησης. Θα αναλύσουμε τις κύριες κατηγορίες αλγορίθμων, εστιάζοντας στις ομοιότητες και διαφορές τους, και θα εμβαθύνουμε στους δημοφιλέστερους αλγορίθμους εποπτευόμενης και μη εποπτευόμενης μάθησης, που χρησιμοποιούνται ευρύτατα σε έρευνες και αναλύσεις δεδομένων. Συνεχίζοντας στο πιο απαιτητικό σκέλος της εργασίας, θα υλοποιήσουμε μια εύχρηστη δίγλωσση command line εφαρμογή που αναπτύχθηκε σε γλώσσα Python, μέσω της οποίας ο χρήστης θα μπορεί να εκτελέσει αλγορίθμους, να ανακτήσει πληροφορίες από εκτελέσεις δικές του αλλά και άλλων χρηστών (χρησιμοποιώντας μια απομακρυσμένη βάση δεδομένων MongoDB), και να αποθηκεύσει σε excel αρχεία πληροφορίες για ένα CSV dataset της επιλογής του. Στο ερευνητικό κομμάτι, θα εκπαιδεύσουμε και θα εκτελέσουμε 4 γνωστούς αλγορίθμους εποπτευόμενης μάθησης, χρησιμοποιώντας ένα προκαθορισμένο σύνολο δεδομένων ενεργειακών χαρακτηριστικών, και θα συμπεράνουμε ότι ο Naive Bayes παρουσιάζει τα καλύτερα αποτελέσματα από τους άλλους τρεις ανταγωνιστές του. Οι αλγόριθμοι που θα χρησιμοποιηθούν είναι οι SVM, Logistic Regression, Naive Bayes και Decision Tree. Τέλος, θα χρησιμοποιήσουμε και τον αλγόριθμο μη εποπτευόμενης μάθησης K-Means στο ίδιο σύνολο δεδομένων, και με τη βοήθεια κατάλληλων βιβλιοθηκών οπτικοποίησης της Python θα παράξουμε και τις αντίστοιχες συστάδες σε δισδιάστατα γραφήματα.

Λέξεις κλειδιά: Μηχανική εκμάθηση, Python, SVM, Logistic Regression, Naive Bayes, Decision Trees, K-Means, MongoDB

Abstract

In this thesis we attempt to approach from a theoretical perspective the interesting disciplines of Data Science and Machine Learning. We will analyze the main categories of ML algorithms focusing on their similarities and differences, and we are going to delve into the most popular supervised and unsupervised algorithms that are being used in research and data analysis projects. Moving on to the more demanding part of the thesis, we will implement a user-friendly, bilingual, Python command line application that will offer users various capabilities, such as executing an algorithm of their choice on a predefined dataset of building energy features, retrieving algorithm execution data from a remote Mongo database and creating useful excel files that contain information regarding a CSV dataset of their choice. As far as research is concerned, we will train and execute 4 widely used supervised learning algorithms and conclude that Naive Bayes is the best overall algorithm compared with its other 3 competitors. The algorithms that will be used in this experiment are SVM, Logistic Regression, Naive Bayes and Decision Tree. Last but not least, we will use the unsupervised learning algorithm K-Means on the same dataset, and with the help of appropriate data visualization Python libraries we will produce the corresponding clusters in 2D graphs.

Keywords: Machine Learning, Python, SVM, Logistic Regression, Naive Bayes, Decision Trees, K-Means, MongoDB

Πίνακας περιεχομένων

Πίνακας εικόνων

ΚΕΦΑΛΑΙΟ 1 - Εισαγωγή

1.1 Κίνητρο

Το κίνητρο πίσω από την συγκεκριμένη πτυχιακή εργασία είναι να παρουσιαστούν σε ένα συμπαγές κείμενο, σημαντικές πληροφορίες σχετικά με την επιστήμη δεδομένων και την μηχανική εκμάθηση. Οι πληροφορίες αυτές περιλαμβάνουν τις κατηγορίες των αλγορίθμων, δημοφιλείς αλγορίθμους (με ιδιαίτερη μνεία στα πλεονεκτήματα και μειονεκτήματα τους) καθώς και αναφορά των σημαντικότερων δημοσιεύσεων των τελευταίων 2 ετών, το ερευνητικό έργο των οποίων μπορεί να αποτελέσει τη βάση για μελλοντικές ανακαλύψεις και θεωρίες στο Machine/Deep Learning.

Επιπλέον κίνητρο υπάρχει πίσω και από το δεύτερο σκέλος της εργασίας, το οποίο αφορά αποκλειστικά hands on υλοποιήσεις αλγορίθμων, ανάλυση και παρουσίαση των αποτελεσμάτων τους, καθώς και πιο τεχνικές λεπτομέρειες τις οποίες θα μπορεί να εκμεταλλευτεί ένας data scientist. Ο σκοπός του συγγραφέα είναι η δημιουργία ανοιχτού (open source) λογισμικού, το οποίο μέσω κατάλληλων οδηγιών να δίνει τη δυνατότητα σε αρχάριους αλλά και προχωρημένους προγραμματιστές να το εκτελέσουν σε δικό τους περιβάλλον, να το προσαρμόσουν στις δικές τους ανάγκες, και εφόσον επιθυμούν να το επεκτείνουν με δικές τους ιδέες.

1.2 Σκοποί και στόχοι

Οι στόχοι της παρούσας εργασίας είναι 2, εκ των οποίων ο πρώτος αφορά την θεωρητική προσέγγιση, αναφορά και βιβλιογραφική έρευνα στους δημοφιλέστερους αλγορίθμους μηχανικής εκμάθησης, όπως αυτοί έχουν διαμορφωθεί έως σήμερα. Ο δεύτερος στόχος είναι η υλοποίηση εφαρμογής που θα δίνει στον χρήστη διάφορες δυνατότητες σχετικές με αλγορίθμους και σύνολα δεδομένων.

Οι παραπάνω στόχοι που έχουν τεθεί, θα πραγματοποιηθούν επιγραμματικά με τους ακόλουθους σκοπούς:

- Εκτενής βιβλιογραφική ανασκόπηση και αναφορά σε ακαδημαϊκές δημοσιεύσεις από τον χώρο του Data Science και του Machine Learning.
- Θεωρητική ανάλυση των προτεινόμενων αλγορίθμων, διαχωρίζοντας τους ανάλογα με την κατηγορία και τον τρόπο λειτουργίας τους.
- Ιδιαίτερη μνεία σε σχετικά επιστημονικά έργα, παγκόσμιου βεληνεκούς.
- Δημιουργία και διαμόρφωση μιας online MongoDB βάσης δεδομένων.

- Ανάπτυξη Command Line εφαρμογής και υλοποίηση αυτόνομων python scripts και jupyter notebooks.
- Διασύνδεση της εφαρμογής με την βάση δεδομένων.

1.3 Οργάνωση της εργασίας

Η εργασία είναι οργανωμένη σε 5 διακριτά κεφάλαια, τα οποία περιγράφονται συνοπτικά στα παρακάτω bullets:

- **ΚΕΦΑΛΑΙΟ 1:** Εισαγωγικό κεφάλαιο το οποίο παρουσιάζει επιγραμματικά τα κίνητρα και τους στόχους που θα επιτευχθούν μετά από την ολοκλήρωση της εργασίας.
- **ΚΕΦΑΛΑΙΟ 2**: Το συγκεκριμένο κεφάλαιο είναι αμιγώς θεωρητικό και περιλαμβάνει την παρουσίαση αλγορίθμων μηχανικής εκμάθησης και την αναφορά σε σημαντικά έργα που σχετίζονται με τον χώρο του Machine Learning. Περιέχει επίσης κάποιες βασικές πληροφορίες για το εμπλεκόμενο λογισμικό που έχει χρησιμοποιηθεί στα επόμενα κεφάλαια.
- **ΚΕΦΑΛΑΙΟ 3**: Σύντομο κεφάλαιο που περιγράφει τις μέθοδους οι οποίες χρησιμοποιήθηκαν από τον συγγραφέα τόσο για τη συγγραφή της εργασίας (βιβλιογραφική αναφορά, ανάγνωση σχετικών δημοσιεύσεων κλπ.) όσο και για την υλοποίηση της εφαρμογής (προγραμματιστικές καλές πρακτικές και μεθοδολογία ανάπτυξης λογισμικού).
- **ΚΕΦΑΛΑΙΟ 4**: Εδώ αναγράφονται αναλυτικά όλες οι λεπτομέρειες που αφορούν το πρακτικό κομμάτι της εργασίας, τα αποτελέσματα που παράχθηκαν καθώς και εκτενής περιγραφή των λειτουργιών που προσφέρει η διαδραστική εφαρμογή που υλοποιήθηκε.
- **ΚΕΦΑΛΑΙΟ 5**: Η κατακλείδα της πτυχιακής εργασίας, στην οποία αναφέρονται συνοπτικά τα παραχθέντα αποτελέσματα καθώς και πιθανό μελλοντικό έργο.

ΚΕΦΑΛΑΙΟ 2 - Βιβλιογραφική ανασκόπηση

2.1 Ιστορική αναδρομή στο Machine Learning και στο Data Science

Σε μια σύντομη ιστορική αναδρομή, ο όρος "Machine Learning" ή αλλιώς μηχανική εκμάθηση χρησιμοποιήθηκε για πρώτη φορά το μακρινό 1952, όταν ο Arthur Samuel της IBM έφτιαξε ένα πρόγραμμα το οποίο μπορούσε να παίξει το γνωστό επιτραπέζιο παιχνίδι στρατηγικής checkers, ή αλλιώς ντάμα στα ελληνικά (Foote, 2019). 15 χρόνια αργότερα γεννήθηκε ο αλγόριθμος Nearest Neighbor, ο οποίος ήταν η αρχή του βασικού pattern recognition[1].

Μέχρι τις αρχές του 1980, το machine learning και το AI[2] είχαν ένα κοινό μονοπάτι, αλλά στη συνέχεια διαχωρίστηκαν και η επιστήμη της μηχανικής εκμάθησης διατήρησε τον προσανατολισμό που είχε στα Νευρωνικά Δίκτυα και ευδοκίμησε την δεκαετία του 1990, γεγονός που οφείλεται και στην ραγδαία αύξηση του Ίντερνετ. Το 2006 αναπτύχθηκαν οι πρώτοι αλγόριθμοι αναγνώρισης προσώπου, και το 2007 το μοντέλο νευρωνικού δικτύου LSTM[3] άρχισε να ξεπερνά σε απόδοση πιο παραδοσιακά μοντέλα αναγνώρισης φωνής.

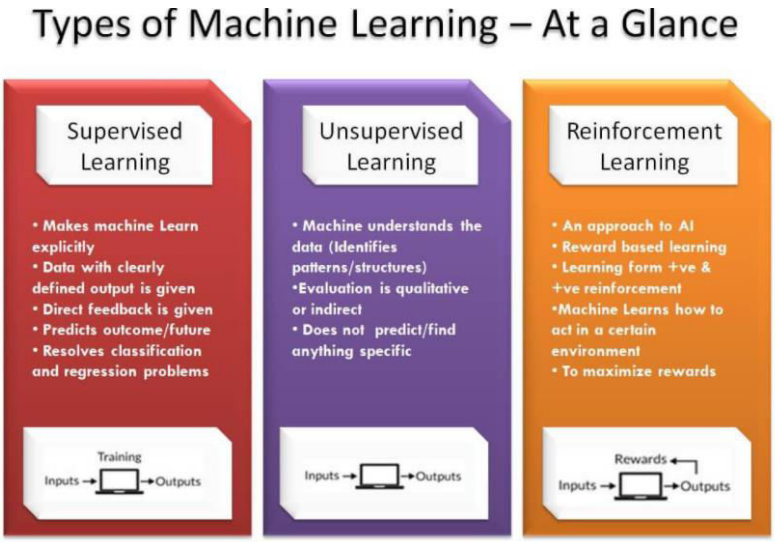
Το 2012, η ομάδα X Lab της Google ανέπτυξε έναν αλγόριθμο που μπορούσε αυτόνομα να περιηγηθεί και να βρει βίντεο που περιέχουν γάτες, και το 2014 η Facebook υλοποίησε το DeepFace, έναν προχωρημένο αλγόριθμο ο οποίος μπορούσε να αναγνωρίσει πρόσωπα σε φωτογραφίες με την ίδια ακρίβεια που θα το έκανε και ένας άνθρωπος. Για περισσότερες λεπτομέρειες αναφορικά με την ιστορία του machine learning, η Google έχει φτιάξει ένα εξαιρετικό διάγραμμα το οποίο περιέχει πληροφορίες από 60 πηγές (https://cloud.withgoogle.com/build/data-analytics/explore-history-machine-learning/).

[1] Pattern recognition είναι η αυτόματη αναγνώριση μοτίβων και κανονικοτήτων σε ένα σύνολο δεδομένων (Wikipedia, 2020)

[2] AI: Τεχνητή νοημοσύνη, είναι η προσομοίωση της ανθρώπινης νοημοσύνης σε μηχανές (υπολογιστές) οι οποίες είναι προγραμματισμένες να σκέφτονται και να δρουν σαν ανθρώπινα όντα (Frankenfield, 2020)

[3] Long Short-Term Memory, αρχιτεκτονική επαναλαμβανόμενου νευρωνικού δικτύου (recurrent neural network), που χρησιμοποιείται στον τομέα του deep learning (Hochreiter & Schmidhuber, 1997)

2.2 Κατηγορίες αλγορίθμων μηχανικής εκμάθησης



Εικόνα 1 - Βασικές κατηγορίες αλγορίθμων (Tripathi, 2019)

Κάτω από την ομπρέλα της μηχανικής εκμάθησης (machine learning), υπάρχουν αρκετοί αλγόριθμοι οι οποίοι έχουν σκοπό να προβλέψουν, να αναλύσουν, να ταξινομήσουν και να κατηγοριοποιήσουν δεδομένα. Αυτοί οι αλγόριθμοι χωρίζονται σε τρεις βασικές κατηγορίες, τις οποίες αναλύει σε ηλεκτρονικό επιστημονικό άρθρο ο Data Scientist David Fumo (Fumo, 2017):

- **Supervised:** Οι αλγόριθμοι εποπτευόμενης μάθησης χρησιμοποιούν δεδομένα με ετικέτα (labeled data) και προσπαθούν να βρουν τη συσχέτιση ανάμεσα στα δεδομένα εισόδου και σε ένα ή περισσότερα δεδομένα εξόδου.
- **Unsupervised:** Οι αλγόριθμοι μη εποπτευόμενης μάθησης χρησιμοποιούν δεδομένα χωρίς ετικέτα, και "χρησιμοποιούνται σε περιπτώσεις όπου ο αναλυτής δε γνωρίζει ακριβώς τη δομή των δεδομένων και δε ξέρει τι ακριβώς πρέπει να κοιτάξει μέσα στο dataset" (Fumo, 2017)
- **Reinforcement:** Οι αλγόριθμοι ενισχυόμενης μάθησης εκπαιδεύουν συνεχώς τον εαυτό τους μέσω της trial and error διαδικασίας, και δρουν με σκοπό την μεγιστοποίηση της ανταμοιβής (πχ high score σε ένα video game) σε κάθε επανάληψη.

Υπάρχει και η μικρότερη κατηγορία των **Semi-supervised** αλγορίθμων, οι οποίοι είναι ένα κράμα εποπτευόμενης και μη εποπτευόμενης μάθησης. Επειδή το κόστος της ύπαρξης ετικέτας σε όλα τα δεδομένα είναι υψηλό, πολλές φορές χρησιμοποιείται ο ανθρώπινος παράγοντας για να μπουν ετικέτες μόνο σε κάποια από τα δεδομένα, δημιουργώντας έτσι ένα ανάμεικτο dataset από labeled και unlabeled data.

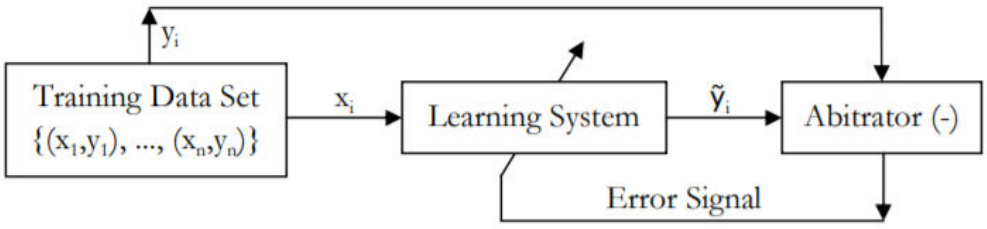
2.3 Αλγόριθμοι εποπτευόμενης μάθησης

2.3.1 Ορισμός και τρόπος λειτουργίας

Η εποπτευόμενη μάθηση είναι ένα πρότυπο μηχανικής εκμάθησης το οποίο χρησιμοποιεί ένα δείγμα δεδομένων (ζευγάρια εισόδου-εξόδου), για να εκπαιδευτεί και να αναπτύξει συσχετισμό και λογική μεταξύ τους, προκειμένου να προβλέψει εκ νέου το αποτέλεσμα από ένα νέο, άγνωστο σύνολο δεδομένων. Καθώς το αποτέλεσμα (output) θεωρείται ως η «ετικέτα» των δεδομένων εισόδου (input data), το δείγμα των δεδομένων ονομάζεται και ως labeled data ή αλλιώς εποπτευόμενα δεδομένα (Liu & Wu, 2012).

Ο σκοπός της εποπτευόμενης μάθησης είναι να χτίσει ένα τεχνητό σύστημα το οποίο θα συσχετίσει και θα μάθει την αντιστοίχιση ανάμεσα στα δεδομένα, με σκοπό να προβλέψει δεδομένα εξόδου έχοντας ως είσοδο δεδομένα τα οποία είναι καινούργια για το εν λόγω σύστημα. Εάν το αποτέλεσμα αποτελείται από πεπερασμένο αριθμό διακριτών τιμών, τότε η εποπτευόμενη μάθηση οδηγεί σε ταξινόμηση (classification) των δεδομένων εισόδου. Στην αντίθετη περίπτωση όπου οι τιμές είναι συνεχείς, τότε θεωρείται ότι γίνεται παλινδρόμηση (regression).

Εν αντιθέσει με την μη εποπτευόμενη μάθηση, η οποία θα αναλυθεί παρακάτω, οι αλγόριθμοι εποπτευόμενης μάθησης δουλεύουν με δεδομένα τα οποία έχουν ετικέτα (labeled data)[4]. Αυτό σημαίνει πως και τα δεδομένα εκπαίδευσης (training data) πρέπει να έχουν ετικέτα, και όπως είναι λογικό να υποθέσει κανείς, όσο πιο σωστά εκπαιδευτεί ένας τέτοιος αλγόριθμος τόσο πιο αποδοτικός θα είναι. Σε περίπτωση που στο στάδιο της εκμάθησης χρησιμοποιούνται και δεδομένα χωρίς ετικέτα, τότε ο υπό ανάπτυξη αλγόριθμος ανήκει στην κατηγορία της ημι-εποπτευόμενης μάθησης (semi-supervised learning). Μια σημαντική λεπτομέρεια είναι πως “αν ο αλγόριθμος ρωτάει ενεργά τον χρήστη για ετικέτες κατά την διάρκεια της εκμάθησης, τότε αυτή η επαναλαμβανόμενη διαδικασία ονομάζεται Ενεργή Εκμάθηση” (Liu & Wu, 2012).



[4] Labeled data είναι τα δεδομένα που χαρακτηρίζονται από διακριτές κολώνες-στήλες, οι οποίες έχουν και έναν τίτλο που αποτελεί και το όνομα του χαρακτηριστικού που περιέχει η κάθε κολώνα.

Εικόνα 2 - Διάγραμμα που δείχνει τον τρόπο λειτουργίας της εποπτευόμενης μάθησης

Στην εικόνα 2 απεικονίζεται ο τρόπος λειτουργίας των αλγορίθμων εποπτευόμενης μάθησης. Στο εν λόγω διάγραμμα, τα ζευγάρια (x_i ,y_i) είναι τα δεδομένα εκμάθησης, όπου το x αναπαριστά την είσοδο και το y την έξοδο. Μόλις ξεκινήσει η εκπαίδευση, τα x_i τροφοδοτούνται στο σύστημα (Learning System), το οποίο παράγει ένα αποτέλεσμα $\tilde{y}_i$. Στη συνέχεια το $\tilde{y}_i$ συγκρίνεται με το αληθές και πραγματικό y_i, από τον arbitrator, ο οποίος αναλαμβάνει το ρόλο του κριτή. Η διαφορά που προκύπτει αναπαρίσταται ως error signal, και ουσιαστικά είναι η πληροφορία που ανατροφοδοτείται πίσω στο σύστημα εκμάθησης με σκοπό να προσαρμοστεί κατάλληλα ο αλγόριθμος και να βελτιωθεί. Η συγκεκριμένη διαδικασία θα επαναληφθεί πολλές φορές, με σκοπό εν τέλει να μειωθεί στο ελάχιστο η διαφορά ανάμεσα στο αποτέλεσμα που προβλέπει το σύστημα ($\tilde{y}_i$), και στο πραγματικό (y_i).

Αξίζει να σημειωθεί ωστόσο, ότι μια ελάχιστη δυνατή τιμή σφάλματος (η οποία επετεύχθη κατά τη διάρκεια της εκμάθησης) δεν εγγυάται απόλυτα και την άριστη απόδοση του αλγορίθμου. Ο αλγόριθμος εκπαιδεύεται με σκοπό να ελεγχθεί σε δεδομένα τα οποία δεν έχει ξανασυναντήσει, για τα οποία καλείται να κάνει προβλέψεις βάσει της εκμάθησης που έχει υποστεί. Ο κύριος λόγος για τον οποίο ένας αλγόριθμος μπορεί να μην έχει καλή απόδοση, ενώ είχε καταφέρει μια πολύ καλή διαδικασία εκμάθησης, είναι το λεγόμενο φαινόμενο overfitting. Το overfitting παρατηρείται όταν έχουν γίνει υπερβολικά πολύπλοκες ενέργειες αντιστοίχισης των εκπαιδευτικών δεδομένων εισόδου-εξόδου, με αποτέλεσμα ο αλγόριθμος να μπορεί να λειτουργήσει αποδοτικά μόνο με τα δεδομένα τα οποία χρησιμοποιήθηκαν στην φάση της εκμάθησης. Με άλλα λόγια, ο αλγόριθμος έχει εκπαιδευτεί τόσο πολύ και έχει αναπτύξει λανθασμένους κανόνες οι οποίοι έχουν εφαρμογή μόνο στα συγκεκριμένα δεδομένα, οδηγώντας σε λανθασμένα αποτελέσματα όταν τροφοδοτείται με νέα.

Για τον παραπάνω λόγο, ένας καλός αλγόριθμος θα πρέπει να παρουσιάζει και έναν καλό δείκτη γενίκευσης. Θα πρέπει δηλαδή να μην “παντρεύεται” τα δεδομένα εκμάθησης και να μην αναπτύσσει συσχετίσεις οι οποίες δεν θα έχουν εφαρμογή σε δεδομένα ελέγχου ή άλλα δεδομένα του ίδιου τύπου. Κάτι τέτοιο επιτυγχάνεται βρίσκοντας την χρυσή τομή ανάμεσα στην ελαχιστοποίηση της τιμής σφάλματος και στην δημιουργία αχρείαστων και πολύπλοκων κανόνων οι οποίοι θα είναι λανθασμένοι, με σκοπό να μειωθεί ακόμη περισσότερο το σφάλμα κατά τη

διάρκεια της εκμάθησης. Για παράδειγμα, στις Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines), ο βαθμός γενίκευσης του μαθητευόμενου αλγορίθμου χαρακτηρίζεται από την διάσταση Vapnik–Chervonenkis (VC dimension[5]) και τον αριθμό χαρακτηριστικών (dimensionality) των δεδομένων. Όσο μειώνονται αυτά τα δύο τόσο πιο γενικός και αποδοτικός γίνεται ο υπό ανάπτυξη αλγόριθμος (Seetha, Murty, & Saravanan, 2011).

Τέλος, να αναφερθεί ότι το μοντέλο εποπτευόμενης μάθησης δεν παρουσιάζει κάποιους περιορισμούς αναφορικά με τις πηγές δεδομένων. Τα δεδομένα εισόδου μπορεί να ανήκουν σε ένα διανυσματικό χώρο ή να είναι διακριτές τιμές. Περιορισμοί δεν υπάρχουν ούτε στο κομμάτι του arbitrator· εάν το y_i υπολογίζεται σε ένα συνεχές διάστημα, τότε το σφάλμα υπολογίζεται ως y_i - $\tilde{y}_i$. Σε διακριτές τιμές, το τμήμα του arbitrator συνήθως υπολογίζει την τιμή σφάλματος βασιζόμενο στην ισότητα των y_i και $\tilde{y}_i$. Για παράδειγμα, για κάθε ισότητα μπορεί να παράγεται 0 και για κάθε διαφορά 1, καταλήγοντας με αυτόν τον τρόπο σε μεγάλη τιμή σφάλματος όταν παρατηρούνται αρκετές διαφορές στα y_i, $\tilde{y}_i$.

2.3.2 Πλεονεκτήματα και περιορισμοί

Σαφώς, όπως όλες οι επιστημονικές διεργασίες, έτσι και η διαδικασία της εποπτευόμενης μάθησης παρουσιάζει κάποια θετικά και κάποια αρνητικά στοιχεία, τα οποία την καθιστούν κατάλληλη για επιλογή και χρήση κάτω από ορισμένες συνθήκες. Πιο συγκεκριμένα, ο data scientist Ashwin Joy αναλύει με κατανοητό τρόπο τα πιο σημαντικά πλεονεκτήματα και μειονεκτήματα των αλγορίθμων, σε ηλεκτρονικό επιστημονικό του άρθρο (Joy, 2020).

Τα πλεονεκτήματα άξια αναφοράς μπορούν να συνοψιστούν στα παρακάτω bullets:

- Ο ερευνητής έχει ξεκάθαρη εικόνα όλων των χαρακτηριστικών των δεδομένων, καθώς τα δεδομένα εκμάθησης και τα πραγματικά δεδομένα έχουν την ίδια μορφή.
- Πρόκειται για μια σχετικά απλή διαδικασία, μέρος της οποίας μπορεί εύκολα να εξηγηθεί, όπως και έγινε στην παραπάνω ενότητα. Σε αντίθεση με την μη εποπτευόμενη μάθηση, όπου εκεί η διαδικασία εκμάθησης αποτελεί κατά κύριο λόγο ένα μαύρο κουτί, αφού οι

[5] Είναι ένας δείκτης που (εν συντομία) απεικονίζει το πόσο μπορεί να εκπαιδευτεί αποτελεσματικά ένας δυαδικός αλγόριθμος ταξινόμησης βάσει πολυποκότητας, προσαρμοστικότητας και άλλων χαρακτηριστικών (Wikipedia, Wikipedia, 2021)

αλγόριθμοι πολλές φορές δημιουργούν συσχετισμούς σε non labeled data οι οποίοι δεν είναι κατανοητοί σε έναν άνθρωπο αλλά μόνο σε μια μηχανή.

- Ο ερευνητής μπορεί εκ των προτέρων να είναι πολύ συγκεκριμένος σχετικά με το ποια χαρακτηριστικά από το dataset τον ενδιαφέρουν, ούτως ώστε να προσαρμοστεί και ο αλγόριθμος κατάλληλα.
- Αφού ολοκληρωθεί η εκπαίδευση του υπό ανάπτυξη αλγορίθμου, τα δεδομένα που χρησιμοποιήθηκαν στην εν λόγω διαδικασία μπορούν να διαγραφούν, και να αποθηκευτεί μόνο η παραγόμενη μαθηματική φόρμουλα και οι κανόνες που αναπτύχθηκαν. Μια τέτοια ενέργεια θα εξοικονομούσε χώρο και πόρους στο υπολογιστικό σύστημα.
- Όπως θα αποδειχθεί και στο πρακτικό κομμάτι, οι αλγόριθμοι εποπτευόμενης μάθησης παράγουν εξαιρετικά αποτελέσματα σε προβλήματα αμιγούς ταξινόμησης.

Στην αντίπερα όχθη, κάποια αρνητικά στοιχεία και περιορισμοί που παρουσιάζει η εποπτευόμενη μάθηση:

- Λόγω του τρόπου λειτουργίας της (χρήση μόνο labeled data, εκπαίδευση με συγκεκριμένα δεδομένα) η εποπτευόμενη μάθηση περιορίζεται σε πολύ συγκεκριμένα προβλήματα, τα οποία αποτελούν ένα αρκετά μικρό μέρος της πολυπλοκότητας που κρύβει η επιστήμη της μηχανικής εκμάθησης.
- Σε συνέχεια του πρώτου bullet, δεν υπάρχει περίπτωση να παραχθούν κάποιες νέες, άγνωστες προς τον χρήστη πληροφορίες (όπως για παράδειγμα ορισμένα μοτίβα τα οποία δεν είχε φανταστεί ότι υπάρχουν), κάτι που συμβαίνει αρκετά συχνά με την μη εποπτευόμενη μάθηση όπου ο αλγόριθμος έχει περισσότερη ελευθερία.
- Τα αποτελέσματα και η αποδοτικότητα του αλγορίθμου περιορίζονται σε δεδομένα που έχουν τα ίδια χαρακτηριστικά με αυτά για τα οποία έχει εκπαιδευτεί. Για παράδειγμα, ένας ταξινομητής εικόνων εποπτευόμενης μάθησης έχει περάσει την διαδικασία της εκπαίδευσης με γάτες και σκύλους. Εάν ως δεδομένο εισόδου δοθεί ένα άλλο ζώο, πχ ελέφαντας, τότε ο αλγόριθμος λανθασμένα θα δώσει ως output γάτα ή σκύλο.
- Για να είναι αποδοτικός ένας αλγόριθμος και με πραγματικά δεδομένα, θα πρέπει να αφιερωθεί πολύς χρόνος στην διαδικασία της εκμάθησης. Πιο συγκεκριμένα, θα πρέπει τα δεδομένα εκμάθησης να περιέχουν όσο το δυνατόν πιο ευρύ δείγμα από όλα τα χαρακτηριστικά που ενδιαφέρουν τον ερευνητή, αλλιώς θα τίθεται σε ρίσκο η ακρίβεια του αλγορίθμου.

- Σε μεγάλα μεγέθη δεδομένων, η διαδικασία εκμάθησης καθώς και η ταξινόμηση αυτή καθαυτή είναι εξαιρετικά χρονοβόρες, και όπως είναι λογικό ο χρόνος είναι ένα πολύτιμο αγαθό και στην Επιστήμη των Δεδομένων και του Machine Learning.

2.4 Αλγόριθμοι μη εποπτευόμενης μάθησης

2.4.1 Ορισμός και τρόπος λειτουργίας

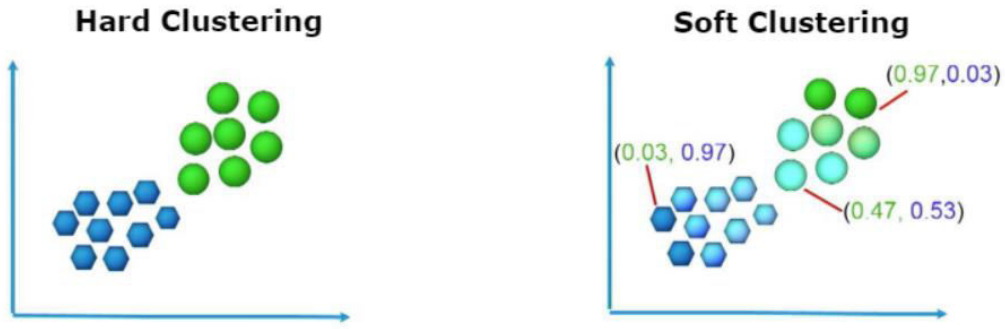
Η μη εποπτευόμενη μάθηση χρησιμοποιεί αλγορίθμους μηχανικής εκμάθησης με σκοπό να αναλύσει και να ομαδοποιήσει σύνολα δεδομένων χωρίς ετικέτα. Οι αλγόριθμοι αυτοί ανακαλύπτουν κρυφά μοτίβα ή συστάδες από παρόμοια δεδομένα, δίχως να είναι αναγκαία η ανθρώπινη παρέμβαση. Αυτή η ιδιότητα τους καθιστά κατάλληλους για διερευνητική ανάλυση δεδομένων, τεχνικές cross-selling[6], συσταδοποίηση πελατών και αναγνώριση εικόνας (IBM, 2021).

Οι τεχνικές μη εποπτευόμενης μάθησης αξιοποιούνται κυρίως για τρεις σκοπούς:

- Συσταδοποίηση (Clustering)
- Συσχέτιση (Association rules)
- Μείωση διαστάσεων (Dimensionality reduction)

Συσταδοποίηση (Clustering)

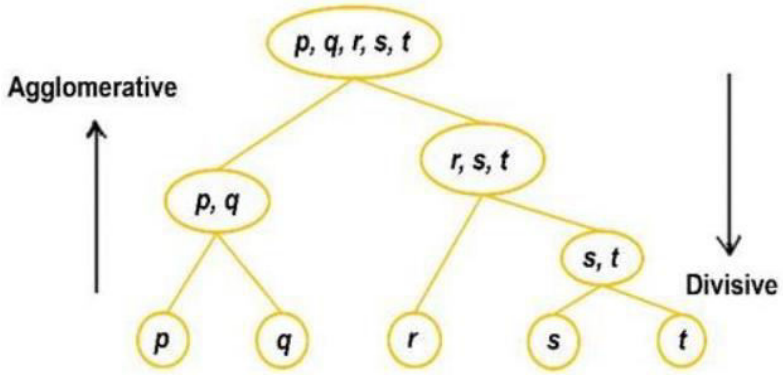
Η συσταδοποίηση είναι η πρώτη τεχνική που θα περιγραφεί, η οποία ουσιαστικά ομαδοποιεί σύνολα δεδομένων ανάλογα με τις ομοιότητες και τις διαφορές τους. Οι clustering αλγόριθμοι χρησιμοποιούνται για να επεξεργαστούν ακατέργαστα δεδομένα και να τα αναθέσουν σε ομάδες με κοινά χαρακτηριστικά και χωρίζονται σε υποκατηγορίες της αποκλειστικής, της επικαλυπτόμενης, της ιεραρχικής και της πιθανολογικής συσταδοποίησης.



[6] Δημοφιλής τεχνική πώλησης η οποία προτείνει στον πελάτη παρόμοια προϊόντα με αυτά που τον ενδιαφέρουν, ή σκοπεύει να αγοράσει (Hayes, Investopedia, 2021).

Εικόνα 3 - Hard vs Soft clustering

Η αποκλειστική και η επικαλυπτόμενη συσταδοποίηση είναι δύο αντίθετες κατηγορίες. Η πρώτη ονομάζεται επίσης και “αυστηρή” (hard), υπό την έννοια ότι μια παρατήρηση από το σύνολο δεδομένων μπορεί ανά πάσα χρονική στιγμή να ανήκει μόνο σε μια συστάδα. Αντιθέτως, στην επικαλυπτόμενη, μια παρατήρηση μπορεί να ανήκει ταυτόχρονα σε δύο ή περισσότερες συστάδες, με διακριτά ποσοστά συμμετοχής (membership) σε κάθε μια από αυτές. Παραδείγματος χάριν, ένα σημείο μπορεί να ανήκει κατά 70% στο cluster A, και 30% στο cluster B. Ο αλγόριθμος K-means είναι ένα χαρακτηριστικό παράδειγμα αποκλειστικής συσταδοποίησης, και αντιστοίχως υπάρχει και ο fuzzy K-means ο οποίος εφαρμόζει επικαλυπτόμενη συσταδοποίηση.



Εικόνα 4 - Agglomerative and divisive clustering

Η ιεραρχική συσταδοποίηση, γνωστή και ως ιεραρχική ανάλυση συστάδων (hierarchical cluster analysis, HCA), εφαρμόζεται από αλγορίθμους δύο αντίθετων κατηγοριών: Συσσωρευτικούς (agglomerative clustering) και διαχωριστικούς (divisive clustering). Στη συσσωρευτική συσταδοποίηση, αρχικά κάθε σημείο είναι μια συστάδα από μόνο του και μέσω μιας επαναλαμβανόμενης συγχωνευτικής διαδικασίας ενώνονται κάθε φορά τα πιο κοντινά data points και σχηματίζουν όλο και λιγότερα clusters, μέχρι να επιτευχθεί ο επιθυμητός αριθμός. Στη διαχωριστική συσταδοποίηση αντίστοιχα, στο αρχικό στάδιο όλα τα δεδομένα ανήκουν σε μια μόνο συστάδα, και σε κάθε βήμα τα clusters πληθαίνουν, πάλι έως ότου επιτευχθεί ο στόχος (Kumar S. , 2020). Η περιγραφή των συγκεκριμένων διαδικασιών απεικονίζεται και στην εικόνα 4 για καλύτερη κατανόηση.

Αναφορικά με τη συσσωρευτική συσταδοποίηση, οι μέθοδοι που χρησιμοποιούνται για να υπολογιστεί ο βαθμός ομοιότητας ανάμεσα στα δεδομένα και να καθοριστούν οι συστάδες, είναι οι εξής (IBM, 2021):

- Συσχετισμός τετραγώνων (Ward’s linkage): Η απόσταση ανάμεσα σε δύο συστάδες καθορίζεται από την αύξηση του αθροίσματος των τετραγώνων, κατόπιν της συγχώνευσης.

- Συσχετισμός μέσης τιμής (Average linkage): Η μέθοδος αυτή υπολογίζει τη μέση απόσταση ανάμεσα σε 2 σημεία κάθε συστάδας.
- Μέγιστος συσχετισμός (Maximum linkage): Η μέθοδος αυτή υπολογίζει τη μέγιστη απόσταση 2 σημείων κάθε συστάδας.
- Ελάχιστος συσχετισμός (Minimum linkage): Αντιστοίχως, η μέθοδος αυτή υπολογίζει την ελάχιστη απόσταση 2 σημείων κάθε συστάδας.

Για τις παραπάνω μεθόδους, χρησιμοποιείται συνήθως η ευκλείδεια απόσταση. Παρ' όλα αυτά θα πρέπει να αναφερθεί και η απόσταση Manhattan, η οποία χρησιμοποιείται πιο σπάνια.

Η πιθανολογική συσταδοποίηση είναι μια τεχνική μη εποπτευόμενης μάθησης η οποία βοηθάει στην επίλυση προβλημάτων πρόβλεψης πυκνότητας. Πιο συγκεκριμένα, τα data points στη συγκεκριμένη τεχνική ομαδοποιούνται βάσει της πιθανότητας τους να ανήκουν σε κάποια συστάδα. Η πιο διαδεδομένη μέθοδος πιθανολογικής συσταδοποίησης είναι το Gaussian Mixture Model[7].

Κανόνες συσχέτισης (Association rules)

Ένας κανόνας συσχέτισης είναι μια μέθοδος βασισμένη σε κανόνες, ιδανική για να ανακαλύπτει συσχετισμούς ανάμεσα στις μεταβλητές, σε ένα σύνολο δεδομένων. Οι μέθοδοι αυτές χρησιμοποιούνται ευρέως για "ανάλυση καλαθιού" (market basket analysis), μέσω της οποίας οι εταιρίες μπορούν να κατανοήσουν καλύτερα τις σχέσεις που έχουν τα προϊόντα που εμπορεύονται. Ο απώτερος σκοπός της ανάλυσης αυτής είναι να αναπτυχθούν καλύτερες cross-selling τεχνικές και μηχανές προτάσεων, με το καλύτερο παράδειγμα να αποτελεί το παγκοσμίως γνωστό site της Amazon (amazon.com). Ο πιο γνωστός αλγόριθμος που χρησιμοποιεί κανόνες συσχέτισης είναι ο Apriori.

Μείωση διαστάσεων (Dimensionality reduction)

Παρ' όλο που ο συνήθης κανόνας λέει ότι όσο περισσότερα δεδομένα υπάρχουν, τόσο καλύτερο θα είναι και το τελικό αποτέλεσμα της ανάλυσης, υπάρχουν αρκετές περιπτώσεις που κάτι τέτοιο δε συμβαίνει. Αναλύσεις που οδηγούν έναν αλγόριθμο σε overfitting (έχει περιγραφεί

[7] "Τα Gaussian Mixture Models είναι πιθανολογικά μοντέλα τα οποία αναπαριστούν κανονικά κατανεμημένους υποπληθυσμούς που περιέχονται σε έναν ενιαίο πληθυσμό." (McGonagle, Geoff, & Dobre, 2020)

παραπάνω) αποδεικνύονται προβληματικές, όπως επίσης και περιπτώσεις όπου τα δεδομένα είναι τόσα πολλά που καθίσταται ανούσια και πολύ δύσκολη η οπτικοποίηση του dataset. Η μείωση διαστάσεων είναι μια τεχνική που εφαρμόζεται όταν το πλήθος των χαρακτηριστικών (ή αλλιώς οι διαστάσεις) ενός dataset, είναι υπερβολικά μεγάλο σε αριθμό. Το αποτέλεσμα είναι η μείωση του αριθμού των χαρακτηριστικών, διασφαλίζοντας όσο είναι δυνατόν την ακεραιότητα και την ακρίβεια του dataset. Η μείωση διαστάσεων χρησιμοποιείται συνήθως στο στάδιο προεπεξεργασίας των δεδομένων, και οι τρεις πιο διαδεδομένες τεχνικές είναι οι παρακάτω:

- Principal component analysis
- Singular value decomposition
- Autoencoders

Η περιγραφή και ανάλυση του τρόπου λειτουργίας των παραπάνω τεχνικών δε θα καλυφθεί στην παρούσα πτυχιακή εργασία.

2.4.2 Πλεονεκτήματα και περιορισμοί

Όπως η εποπτευόμενη μάθηση, έτσι και η μη εποπτευόμενη απαιτεί καλή γνώση προκειμένου να χρησιμοποιηθεί σωστά και να οδηγήσει στα επιθυμητά αποτελέσματα. Παρακάτω τα κυριότερα πλεονεκτήματα της συγκεκριμένης μεθόδου (Valcheva, 2020):

- Σχετικά μικρή πολυπλοκότητα συγκριτικά με το supervised learning. Στη μη εποπτευόμενη μάθηση δεν είναι προαπαιτούμενο η κατανόηση των δεδομένων και η προσθήκη ετικετών.
- Η εφαρμογή των αλγορίθμων και η ανάλυση γίνεται σε πραγματικό χρόνο. Κάτι τέτοιο βοηθάει τους ερευνητές να γνωρίζουν άμεσα τη μορφή των δεδομένων και να ανακαλύπτουν συσχετίσεις που δε θα μπορούσαν ποτέ να βρουν οι ίδιοι.
- Σχεδόν πάντοτε, είναι πολύ πιο εύκολο να βρεθούν unlabeled δεδομένα. Εν έτει 2021 υπάρχει καταιγισμός δεδομένων, εκ των οποίων τα περισσότερα είναι raw data τα οποία δεν έχουν υποστεί επεξεργασία και προσθήκη ετικετών από άνθρωπο.
- Η προσθήκη ετικετών μπορεί να πραγματοποιηθεί με πολύ μεγαλύτερη ευκολία, αφού πρώτα εφαρμοστεί μια clustering μέθοδος.
- Χρησιμοποιώντας τη μείωση διαστάσεων, πολλά σύνολα δεδομένα τα οποία εξαρχής ήταν πολύπλοκα, απλοποιούνται και μπορούν να οπτικοποιηθούν και να υποστούν περαιτέρω ανάλυση πιο αβίαστα.

Τα αρνητικά στοιχεία της μη εποπτευόμενης μάθησης, από την άλλη, είναι τα παρακάτω:

- Δεν υπάρχει μεγάλος έλεγχος στα δεδομένα. Ένας ερευνητής που εφαρμόζει μη εποπτευόμενη μάθηση δε μπορεί εκ των προτέρων να γνωρίζει το αποτέλεσμα, ούτε να αναμένει κάποιο συγκεκριμένο format το οποίο εν συνεχεία θα μπορεί να εκμεταλλευτεί.
- Μικρότερη ακρίβεια. Από τη στιγμή που τη δουλειά της προσθήκης ετικετών την αναλαμβάνει η ίδια η μηχανή και όχι ο άνθρωπος, τότε πιθανώς να δημιουργηθούν λανθασμένα κάποια χαρακτηριστικά.
- Τα αποτελέσματα της ανάλυσης δε μπορούν να επικυρωθούν. Από τη στιγμή που δεν υπάρχουν δεδομένα εκμάθησης ούτε αναμενόμενα αποτελέσματα, το αποτέλεσμα ενός αλγορίθμου μη εποπτευόμενης μάθησης δε μπορεί να διασταυρωθεί με κάποιο τρόπο, παρά μόνο με το μάτι (σε πολύ μικρά datasets).

2.5 Θεωρητική ανάλυση γνωστών Supervised learning αλγορίθμων

Στην ενότητα αυτή θα παρουσιαστούν και θα περιγραφούν οι επικρατέστεροι αλγόριθμοι εποπτευόμενης μάθησης που χρησιμοποιούνται σε αναλύσεις και έρευνες στον κλάδο του Machine Learning. Πιο συγκεκριμένα θα αναλυθούν, σε θεωρητικό πλαίσιο, οι ακόλουθοι αλγόριθμοι:

- Logistic Regression
- Naive Bayes
- Decision Trees

2.5.1 Λογιστική παλινδρόμηση (Logistic Regression)

Περιγραφή

Η data scientist Anamika Thanda περιγράφει σε ένα εξαιρετικά συμπαγές και κατανοητό άρθρο της, τους αλγορίθμους λογιστικής παλινδρόμησης (Thanda, 2020). Η λογιστική παλινδρόμηση ανήκει στην οικογένεια των αλγορίθμων ταξινόμησης και χρησιμοποιείται για να προβλέψει συνήθως ένα δυαδικό αποτέλεσμα (αλλά όχι πάντοτε δυαδικό) το οποίο εξαρτάται από ένα σύνολο, ανεξάρτητων μεταξύ τους, μεταβλητών.

Δυαδικό αποτέλεσμα χαρακτηρίζεται ένα αποτέλεσμα που μπορεί να έχει μόνο δύο διακριτές τιμές, κάτι το οποίο με λόγια μπορεί να μεταφραστεί ως “ένα ενδεχόμενο είτε συνέβη (με τιμή 1/true) είτε δε συνέβη (με τιμή 0/false)”. Οι ανεξάρτητες μεταβλητές είναι οι μεταβλητές

εκείνες που δεν επηρεάζουν η μία την άλλη, αλλά επηρεάζουν την έκβαση του τελικού αποτελέσματος. Συνεπώς, ένας logistic regression αλγόριθμος ενδείκνυται να χρησιμοποιηθεί όταν έχουμε να κάνουμε με δεδομένα τα οποία περιέχουν κατηγορικές εξαρτώμενες μεταβλητές[8].

Οι ανεξάρτητες μεταβλητές ενός dataset μπορούν να ανήκουν σε κάποια από τις ακόλουθες κατηγορίες:

- **Συνεχείς**: Παραδείγματος χάριν η θερμοκρασία σε βαθμούς Κελσίου ή το βάρος σε γραμμάρια. Χρησιμοποιώντας πιο τεχνικούς όρους, τα συνεχή δεδομένα μπορεί να είναι είτε interval data, δηλαδή να υπάρχουν ίσα διαστήματα ανάμεσα στις τιμές, είτε ratio data, δηλαδή να υπάρχουν ίσα διαστήματα ανάμεσα στις τιμές και να έχει κάποιο ουσιαστικό νόημα η μηδενική τιμή.
    - Επί παραδείγματι, η θερμοκρασία Κελσίου χαρακτηρίζεται ως interval data, καθώς η διαφορά που υπάρχει ανάμεσα στους 10 και 11 βαθμούς είναι ίδια με τη διαφορά ανάμεσα στους 30 και 31. Παρ' όλα αυτά η θερμοκρασία 0 βαθμοί Κελσίου είναι μια ακόμη θερμοκρασία, χωρίς να σημαίνει ότι σε αυτή την τιμή δεν υπάρχει θερμοκρασία.
    - Το βάρος σε γραμμάρια χαρακτηρίζεται ως ratio data, καθώς ισχύει ο κανόνας με τα ίσα διαστήματα ανάμεσα στις τιμές, αλλά η τιμή 0 γραμμάρια έχει ξεχωριστή σημασία. Σημαίνει πρακτικά την απουσία βάρους.
- **Διακριτές, σειριακές:** Οι μεταβλητές αυτές μπορούν να τοποθετηθούν σε ένα σύνολο αριθμητικών διακριτών τιμών, μιας συγκεκριμένης κλίμακας. Χαρακτηριστικά παραδείγματα είναι ένα μεγάλο ποσοστό των ερευνών ικανοποίησης πελάτη (customer satisfaction surveys), όπου αρκετά ερωτήματα απαντώνται σε κλίμακες (Πχ Σε μια κλίμακα από 1-5 πόσο ευχαριστημένος είναι κάποιος πελάτης από ένα προϊόν).
- **Διακριτές, συμβολικές:** Αντίστοιχες με τις σειριακές μεταβλητές, με τη διαφορά ότι αυτό το είδος δεν τοποθετείται σε αριθμητικές κλίμακες, αλλά σε σύνολα που περιέχουν λέξεις. Ένα τέτοιο παράδειγμα μπορεί να είναι το χρώμα της ίριδας του ματιού, το οποίο μπορεί να είναι καφέ, πράσινο, μπλε κλπ. Κάτι σημαντικό που πρέπει να τονιστεί είναι ότι δεν υπάρχει συγκεκριμένη ιεραρχία όπως στις σειριακές μεταβλητές, όπου οι μεγάλες τιμές

[8] Εξαρτώμενη μεταβλητή σε ένα σύνολο δεδομένων είναι το χαρακτηριστικό το οποίο βρίσκεται υπό ανάλυση/πρόβλεψη.

σημαίνουν κάτι διαφορετικό από τις μικρότερες τιμές. Στις συμβολικές τιμές κάθε ετικέτα είναι ανεξάρτητη και δεν σχετίζεται με τις υπόλοιπες.

Μέχρι στιγμής έχει περιγραφεί μόνο ένα είδος λογιστικής παλινδρόμησης, που είναι και το επικρατέστερο, και αυτό είναι η δυαδική λογιστική παλινδρόμηση. Στην πραγματικότητα όμως υπάρχουν 3 ξεχωριστά είδη, τα οποία θα περιγραφούν στη συνέχεια και είναι και ένας από τους λόγους που οι logistic regression αλγόριθμοι έχουν τόσο μεγάλη απήχηση:

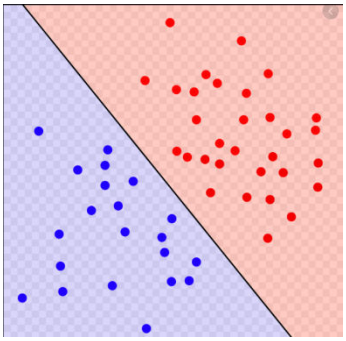
- **Δυαδική (Binary)**: Όπως έχει προαναφερθεί, η εξαρτώμενη μεταβλητή μπορεί να πάρει μόνο δυο διακριτές τιμές. Χαρακτηριστικά παραδείγματα μπορεί να είναι τα δεδομένα τύπου “ΝΑΙ/ΟΧΙ”, “ΑΛΗΘΕΣ/ΨΕΥΔΕΣ”, “0/1”, “ΕΠΙΤΥΧΙΑ/ΑΠΟΤΥΧΙΑ”.
- **Πολυωνυμική (Multinomial)**: Η εξαρτώμενη μεταβλητή μπορεί να έχει περισσότερες από 2 διακριτές τιμές, οι οποίες όμως δε σχετίζονται μεταξύ τους και δε μπορούν να μπουν σε κάποια κλίμακα (πχ από το μικρότερο στο μεγαλύτερο). Χαρακτηριστικό παράδειγμα είναι τα μέσα συγκοινωνίας στη στεριά, με πιθανές τιμές “ΑΥΤΟΚΙΝΗΤΟ”, “ΜΗΧΑΝΗ”, “ΤΡΕΝΟ”, “ΠΟΔΗΛΑΤΟ”.
- **Σειριακή (ordinal)**: Όπως και στην προηγούμενη κατηγορία, και εδώ έχουμε περισσότερες από 2 διακριτές τιμές, οι οποίες όμως σχετίζονται μεταξύ τους και μπορούν να τοποθετηθούν σε μια κλίμακα. Ένα παράδειγμα θα μπορούσε να είναι το μέγεθος από ένα ρούχο, με πιθανές τιμές XS/S/M/L/XL, και ένα άλλο παράδειγμα θα μπορούσε να είναι περιγραφικές κατηγορίες απόδοσης ενός μαθητή στο σχολείο, με πιθανές τιμές “ΚΑΚΑ”, “ΜΕΤΡΙΑ”, “ΚΑΛΑ”, “ΑΡΙΣΤΑ”.

Πλεονεκτήματα και περιορισμοί

Ακολούθως θα παρουσιαστούν κάποια θετικά και αρνητικά στοιχεία της λογιστικής παλινδρόμησης, ξεκινώντας από τα θετικά:

- **Ευκολία εφαρμογής.** Αποτελεί μια σχετικά απλή μέθοδο μηχανικής εκμάθησης, η οποία παρουσιάζει ευκολία τόσο στην εφαρμογή της όσο και στη διαδικασία εκμάθησης των αλγορίθμων από τα δεδομένα εκμάθησης. Η απλότητα αυτή είναι και ένας από τους κύριους λόγους που χρησιμοποιείται σε μεγάλο βαθμό από ερευνητές.
- **Καλή απόδοση σε περιπτώσεις όπου τα δεδομένα διαχωρίζονται γραμμικά (linearly separable).** Ένα dataset (ή υποσύνολο dataset) λέμε ότι είναι γραμμικά διαχωρίσιμο όταν τα δεδομένα δυο διαφορετικών χαρακτηριστικών μπορούν να διαχωριστούν από μια ευθεία γραμμή. Κάτι τέτοιο συνήθως υποδηλώνει μικρή πολυπλοκότητα στο σύνολο

δεδομένων, και πως το κάθε χαρακτηριστικό είναι όντως ανεξάρτητο από τα υπόλοιπα (loosely coupled).



Εικόνα 5 - Linearly separable data (πηγή: Wikipedia)

- **Παροχή σημαντικών πληροφοριών για τα δεδομένα.** Μέσω της λογιστικής παλινδρόμησης ένας ερευνητής μπορεί να αποκτήσει πληροφορίες για τα χαρακτηριστικά των δεδομένων, και πώς ένα συγκεκριμένο χαρακτηριστικό επηρεάζει (και κατά πόσο) την εξαρτώμενη μεταβλητή. Μπορεί επίσης να βρεθεί και ο συσχετισμός ανάμεσα στα χαρακτηριστικά, ο οποίος χαρακτηρίζεται είτε ως θετικός είτε ως αρνητικός. Στην περίπτωση του θετικού συσχετισμού, όταν πραγματοποιείται αύξηση στην αριθμητική τιμή ενός χαρακτηριστικού, τότε θα παρατηρείται η αντίστοιχη αύξηση και στο ζευγάρι του. Στον αρνητικό συσχετισμό θα συμβαίνει το αντίθετο, δηλαδή σε ενδεχόμενο μείωσης της τιμής του ενός χαρακτηριστικού, θα πραγματοποιείται αύξηση της τιμής του άλλου.

Και κάποια από τα αρνητικά στοιχεία που παρουσιάζει η συγκεκριμένη κατηγορία αλγορίθμων:

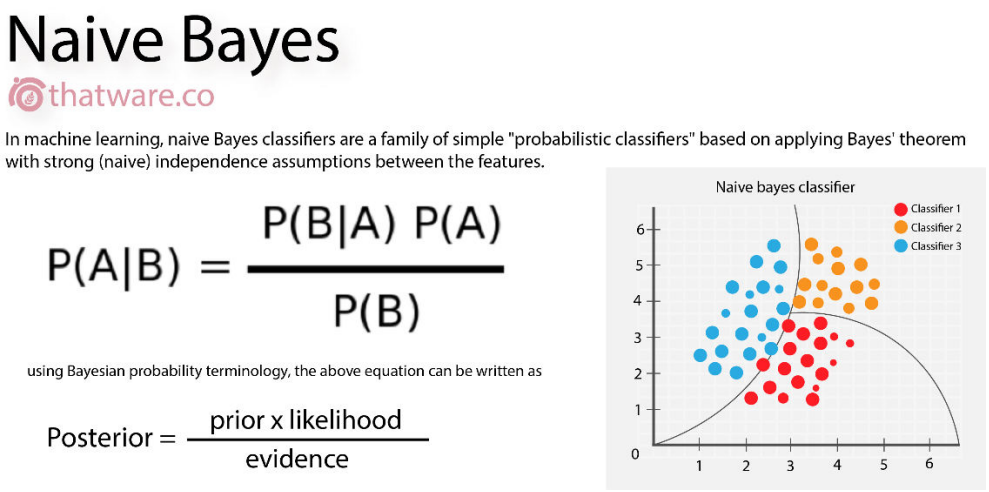
- **Αδυναμία πρόβλεψης συνεχών τιμών.** Όπως έχει προαναφερθεί, υπάρχουν διάφορα είδη λογιστικής παλινδρόμησης, αλλά κανένα από αυτά δεν έχει εφαρμογή σε προβλήματα πρόβλεψης συνεχών τιμών. Κάτι τέτοιο σαφώς και αποτελεί σημαντικό περιορισμό της συγκεκριμένης κατηγορίας, καθώς υπάρχει μεγάλο ποσοστό ερευνών που βασίζονται αποκλειστικά πάνω σε συνεχή δεδομένα.
- **Υπόθεση γραμμικότητας ανάμεσα στην εξαρτώμενη και τις ανεξάρτητες μεταβλητές.** Οι logistic regression αλγόριθμοι έχουν καλές επιδόσεις, αλλά υποθέτουν ότι πάντα υπάρχει γραμμικότητα (βλ. εικόνα 5) ανάμεσα στην υπό διερεύνηση μεταβλητή και στο υπόλοιπο dataset. Σε πραγματικά δεδομένα όμως κάτι τέτοιο είναι σπάνιο, καθώς υπάρχει μεγαλύτερη πολυπλοκότητα και επιρροή ανάμεσα στα διάφορα χαρακτηριστικά του

dataset. Χαρακτηριστικό παράδειγμα είναι το iris plant dataset[9], όπου φυτά με μέγεθος πετάλου 2 εκατοστά μπορούν να ανήκουν και στην κατηγορία Iris Setosa αλλά και στην κατηγορία Iris Versicolour, χωρίς να υπάρχει ξεκάθαρη διάκριση ανάμεσα στα 2.

- **Χαμηλή ακρίβεια όταν δεν υπάρχει ικανοποιητικός όγκος δεδομένων εκπαίδευσης.** Ένα μειονέκτημα το οποίο το έχουν αρκετές οικογένειες αλγορίθμων, είναι αυτό της κακής απόδοσης όταν τα δεδομένα εκπαίδευσης είναι λίγα. Σε μια τέτοια περίπτωση ενδεχομένως να παρατηρηθεί overfitting, όχι λόγω υπερβολικής εκπαίδευσης αλλά λόγω ελλιπούς. Να μην προλάβει δηλαδή ο αλγόριθμος να αναπτύξει και να γενικεύσει τους κανόνες για τα δεδομένα, με αποτέλεσμα εν τέλει να βασίζεται σε ημιτελείς υπολογισμούς που μειώνουν σε σημαντικό βαθμό την ακρίβεια του.

2.5.2 Naive Bayes

Περιγραφή



Εικόνα 6 - Θεώρημα Bayes (αριστερά) & ταξινομητής Naive Bayes (δεξιά). Πηγή: towardsdatascience.com

Οι Naive Bayes αλγόριθμοι είναι η επόμενη κατηγορία που θα παρουσιαστεί στην παρούσα εργασία, και ανήκουν στην οικογένεια των πιθανολογικών αλγορίθμων υπό συνθήκη (conditional

[9] https://archive.ics.uci.edu/ml/datasets/iris

probability algorithms). Για να γίνει κατανοητός ο τρόπος λειτουργίας τους θα πρέπει πρώτα να αναλυθούν οι μαθηματικές βάσεις πάνω στις οποίες πατάνε οι συγκεκριμένοι αλγόριθμοι. "Στην θεωρία των πιθανοτήτων, η πιθανότητα υπό συνθήκη είναι ο υπολογισμός της πιθανότητας πραγματοποίησης ενός συμβάντος με δεδομένο ότι ένα άλλο συμβάν έχει πραγματοποιηθεί (είτε από υπόθεση, είτε αποδεδειγμένα) " (Barone, 2020). Με άλλα λόγια, για δύο ενδεχόμενα Α και Β, η πιθανότητα υπό συνθήκη για το Β ορίζεται ως η πιθανότητα της τομής των δύο ενδεχομένων διά την πιθανότητα πραγματοποίησης του Α[10].

Απλό παράδειγμα υπολογισμού πιθανοτήτων υπό συνθήκη (Barone, 2020): Έστω ένας μαθητής που κάνει αίτηση για να σπουδάσει σε ένα πανεπιστήμιο, και ελπίζει να πάρει και υποτροφία. Το συγκεκριμένο πανεπιστήμιο έχει ως πολιτική να δέχεται 100 υποψηφίους ανά 1000 αιτήσεις, δηλαδή ένα ποσοστό 10% (P(A)). Ανά 500 υποψηφίους που δέχεται, παρέχει δωρεάν υποτροφία στους 10 εξ' αυτών (δηλαδή σε ένα ποσοστό 2% (P(B))). Η γενναιοδωρία του πανεπιστημίου δε σταματά εκεί, καθώς παρέχει δωρεάν στέγαση, διατροφή και βιβλία στο 50% (P(C)) των υποτρόφων. Σύμφωνα με τα παραπάνω δεδομένα λοιπόν, η πιθανότητα για ένα φοιτητή να γίνει δεκτός στο πανεπιστήμιο και στη συνέχεια να λάβει υποτροφία είναι η εξής:

P(B|A) = 0.1*0.02 = 0.002 = 0.2%

Με το ίδιο σκεπτικό, η πιθανότητα ο φοιτητής να γίνει αποδέκτης και των δωρεάν παροχών του πανεπιστημίου είναι 0.1*0.02*0.5 = 0.1%, δηλαδή μόλις 1 στους 1000 υποψηφίους γίνεται δεκτός και απολαμβάνει όλα τα προνόμια.

Οι αλγόριθμοι πιθανοτήτων υπό συνθήκη βασίζονται πάνω σε μια θεμελιώδη μαθηματική αρχή των πιθανοτήτων: το θεώρημα του Bayes. Ονομάστηκε έτσι από τον Βρετανό μαθηματικό του 18ου αιώνα Thomas Bayes, και είναι μια μαθηματική φόρμουλα που χρησιμοποιείται για τον υπολογισμό πιθανοτήτων υπό συνθήκη. Ο τύπος της είναι ο εξής:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \qquad (1)$$

Με λόγια περιγράφεται ως η υπό συνθήκη πιθανότητα ενός ενδεχομένου Α δεδομένου ότι έχει συμβεί το Β, η οποία ισούται με το γινόμενο της υπό συνθήκη πιθανότητας του Β δεδομένου ότι

[10] P(B|A) = P(A∩B) / P(A) (2)

έχει συμβεί το Α επί την πιθανότητα του Α, διαιρεμένο με την πιθανότητα του Β. Οι αλγόριθμοι που βασίζονται στο θεώρημα του Bayes γνωρίζουν πολλές εφαρμογές, με μια εξ' αυτών να είναι στον χώρο των οικονομικών και στον υπολογισμό του ρίσκου που υπάρχει από τις τράπεζες όταν καλούνται να δώσουν δάνειο σε έναν πιθανό δανειολήπτη (Hayes, Investopedia, 2020). Στην μηχανική εκμάθηση, χρησιμοποιούνται κατά κόρον Naive Bayes αλγόριθμοι που έχουν ως βασική αρχή, όπως λέει και το όνομα τους, το προαναφερθέν θεώρημα.

Σε ηλεκτρονικό επιστημονικό άρθρο που έχει γνωρίσει μεγάλη απήχηση λόγω της καλής επεξήγησης που παρέχει (Gandhi, 2018), αναλύονται με σαφήνεια οι 3 τύποι ταξινομητών Naive Bayes που χρησιμοποιούνται στο Machine Learning:

- **Gaussian Naive Bayes:** Χρησιμοποιείται για attributes των οποίων οι τιμές είναι συνεχείς και πραγματικές. Λαμβάνεται η υπόθεση ότι τα δείγματα ακολουθούν κανονική κατανομή[11].
- **Multinomial Naive Bayes:** Η κύρια χρήση του είναι για προβλήματα ταξινόμησης εγγράφων, όπως για παράδειγμα η απόδοση του είδους σε ένα έγγραφο ανάλογα με τις λέξεις τις οποίες περιέχει. Τα attributes πάνω στα οποία γίνεται η ανάλυση/πρόβλεψη είναι οι ίδιες οι λέξεις του κειμένου.
- **Bernoulli Naive Bayes:** Ο αλγόριθμος αυτός είναι όμοιος με τον προηγούμενο, αλλά οι παράμετροι που χρησιμοποιούνται για την πρόβλεψη παίρνουν μόνο δύο τιμές, που αναφέρονται σε πραγματοποίηση ή μη (true/false) ενός χαρακτηριστικού, ή μιας λέξης στο ανωτέρω παράδειγμα.

Σύμφωνα με το ίδιο άρθρο, οι αλγόριθμοι Naive Bayes χρησιμοποιούνται κατά κύριο λόγο σε 4 είδη εφαρμογών:

- **Πρόβλεψη σε πραγματικό χρόνο.** Οι Naive Bayes εκπαιδεύονται γρήγορα και παράγουν αποτελέσματα σε μικρό χρόνο, και για αυτόν τον λόγο χρησιμοποιούνται σε εφαρμογές όπου είναι αναγκαία η άμεση ανατροφοδότηση και αναπροσαρμογή σε νέα δεδομένα.
- Πρόβλεψη σε πολλαπλά χαρακτηριστικά.

[11] Κανονική κατανομή είναι βασική αρχή της στατιστικής και αναφέρεται σε δείγματα πραγματικών τιμών τα οποία τείνουν να συγκεντρώνονται γύρω από μια μέση τιμή.

- **Ταξινόμηση κειμένου**. Οι Naive Bayes αλγόριθμοι λόγω της αφέλειας τους τα καταφέρνουν περίφημα σε επεξεργασία κειμένου, και γι' αυτό χρησιμοποιούνται κατά κόρον για spam filtering[12] καθώς και για sentiment analysis[13].
- **Συστήματα προτάσεων (recommendation systems).** Οι ταξινομητές Naive Bayes σε συνδυασμό με την τεχνική του collaborative filtering[14] χτίζουν ένα δυνατό recommendation system το οποίο μέσω του machine learning και του data mining φιλτράρει πληροφορίες και προβλέπει την πιθανότητα μια συγκεκριμένη πηγή να αρέσει σε έναν χρήστη.

Πλεονεκτήματα και περιορισμοί

Ακολουθώντας τον ίδιο τρόπο παρουσίασης όπως και στη λογιστική παλινδρόμηση, στην συγκεκριμένη ενότητα θα αναφερθούν κάποια πλεονεκτήματα και μειονεκτήματα της οικογένειας των Naive Bayes αλγορίθμων, ξεκινώντας πάντοτε από τα θετικά στοιχεία τους:

- **Εξαιρετικές επιδόσεις του και στους 4 επιστημονικούς τομείς[15]** ***accuracy, prediction, recall, F1.*** Σύμφωνα με μια δημοσίευση του 2009 τριών εισηγητών του Loughborough University, στην οποία συγκρίνεται ο Naive Bayes (NB) με τα Δέντρα Αποφάσεων και τα Νευρωνικά Δίκτυα, αποδεικνύεται ότι τα καταφέρνει πολύ καλά τόσο σε επιδόσεις όσο και σε χρόνους εκτέλεσης (Xhemali, Hinde, & Stone, 2009).
- **Μικρός χρόνος εκπαίδευσης του μοντέλου,** η οποία συνδυάζεται και με την καλή προσαρμογή που παρουσιάζουν σε αλλαγές του dataset (όταν για παράδειγμα εμφανίζονται νέα data points στα δεδομένα) (Catanzarite, 2018).
- **Χαμηλή κατανάλωση υπολογιστικών πόρων (CPU, RAM).** Άλλο ένα σημαντικό πλεονέκτημα το οποίο καθιστά τους Naive Bayes αλγορίθμους κατάλληλους για εφαρμογή σε οποιοδήποτε υπολογιστικό σύστημα, καθώς σε κάθε βήμα δε φορτώνεται όλο το

[12] Τεχνική που χρησιμοποιείται από email providers η οποία αναγνωρίζει την ανεπιθύμητη αλληλογραφία βάσει του περιεχομένου της.
[13] Τεχνική που χρησιμοποιείται σε κοινωνικά δίκτυα και αναγνωρίζει το συναίσθημα (θετικό, αρνητικό) που κρύβεται πίσω από μια πρόταση/παράγραφο ενός χρήστη.
[14] Μέθοδος που χρησιμοποιείται από συστήματα προτάσεων για να προβλέψει τις προτιμήσεις ενός χρήστη.
[15] Οι 4 όροι (accuracy, prediction, recall, F1) καθορίζουν πόσο αποδοτικός είναι ένας αλγόριθμος (Riggio, 2019)

$$\boldsymbol{Accuracy} = \frac{\text{αριθμός σωστών προβλέψεων}}{\text{συνολικός αριθμός προβλέψεων}} \quad (3) \qquad \boldsymbol{Precision} = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (4)$$

$$\boldsymbol{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (5) \qquad \boldsymbol{F1} = 2 * \frac{precision * recall}{precision + recall} \quad (6)$$

dataset στη μνήμη του συστήματος προκειμένου να υπολογιστούν οι πιθανότητες και να βγουν προβλέψεις.

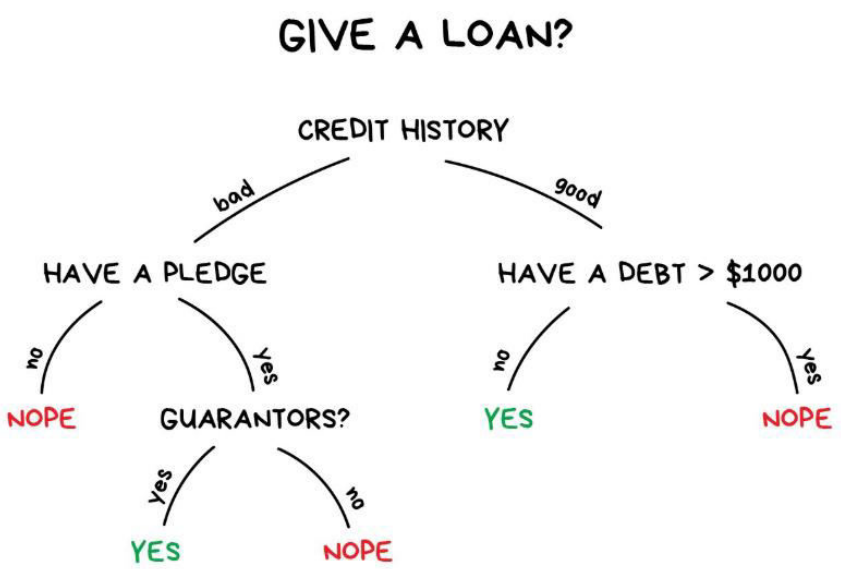
- **Γραμμική αύξηση πολυπλοκότητας σε αναλογία με το μέγεθος των δεδομένων** (αριθμός εγγραφών και χαρακτηριστικών, ή αλλιώς γραμμών και στηλών), γεγονός που τους καθιστά κατάλληλους σε πολύπλοκα datasets.
- **Καλή απόδοση σε ελλιπή δεδομένα,** υπολογίζοντας μέσους όρους ή επιλέγοντας να αγνοήσουν εντελώς τα attributes που παρουσιάζουν κενά.

Οι περιορισμοί που παρουσιάζει η ανωτέρω οικογένεια πιθανολογικών αλγορίθμων υπό συνθήκη είναι απόρροια του τρόπου λειτουργίας τους και σχετίζεται άμεσα και με κάποια από τα πλεονεκτήματα τους:

- **Χαμηλότερη ακρίβεια σε σχέση με άλλους πολυπλοκότερους αλγορίθμους.** Το γεγονός ότι για κάθε πρόβλεψη δεν λαμβάνεται υπόψιν ολόκληρο το dataset (για να υπάρχει καλύτερη απόδοση σε πόρους και χρόνο), πολλές φορές έχει ως αποτέλεσμα η ακρίβεια του μοντέλου να είναι χαμηλότερη σε σχέση με άλλους αλγορίθμους.
- **Υπόθεση ανεξαρτησίας.** Όπως έχει προαναφερθεί, σε δεδομένα πραγματικού κόσμου και όχι εργαστηριακά, είναι πολύ σπάνιο φαινόμενο η πλήρης ανεξαρτησία των μεταβλητών των δεδομένων. Οι Naive Bayes όμως βασίζουν όλη την θεωρία τους πάνω στην υπόθεση της ανεξαρτησίας.
- **Πρόβλημα μηδενικής συχνότητας (zero-frequency problem).** Σύμφωνα με το πρόβλημα μηδενικής συχνότητας, αν ένα χαρακτηριστικό λείπει από την διαδικασία εκπαίδευσης του αλγορίθμου, τότε σε περίπτωση μεταγενέστερης εμφάνισης δεν λαμβάνεται υπόψιν και η πιθανότητα του θα είναι πάντοτε 0. Παρ’ όλα αυτά με περαιτέρω διαμόρφωση του μοντέλου μηχανικής εκμάθησης το συγκεκριμένο πρόβλημα μπορεί να επιλυθεί, όπως αναφέρεται και σε ηλεκτρονικό άρθρο του Raghav Vashisht (Vashisht, 2020).

2.5.3 Δένδρα αποφάσεων (Decision trees)

Περιγραφή



Εικόνα 7 - Δέντρο Αποφάσεων που καθορίζει αν θα πρέπει να δοθεί δάνειο σε υποψήφιο δανειολήπτη, βάσει σχετικών μεταβλητών. Πηγή: medium.com

Οι αλγόριθμοι που βασίζονται σε δέντρα αποφάσεων αποτελούν ένα από τα μοντέλα πρόβλεψης που χρησιμοποιούνται στη στατιστική, το data mining[16] και το machine learning. Τα δέντρα αποφάσεων χωρίζονται σε 2 κύριες κατηγορίες, αναλόγως του τύπου της προβλεπόμενης μεταβλητής:

- Δέντρα ταξινόμησης, τα οποία καλούνται να προβλέψουν μια διακριτή μεταβλητή ως αποτέλεσμα.
- Δέντρα παλινδρόμησης, όπου η πρόβλεψη πρέπει να γίνει σε συνεχή μεταβλητή η οποία μπορεί να πάρει πραγματικές τιμές (πχ χρονική διάρκεια, τιμή, θερμοκρασία κλπ.)

Ο όρος Classification And Regression Tree (CART) χρησιμοποιείται για να αναφερθεί και στις 2 παραπάνω κατηγορίες (Breiman, Friedman, Olshen, & Stone, 1984). Και τα 2 είδη έχουν κάποιες ομοιότητες όσον αφορά στον τρόπο λειτουργίας τους, αλλά και κάποιες διαφορές, όπως για παράδειγμα την διαδικασία που χρησιμοποιείται για να καθοριστεί το που θα γίνει ο διαχωρισμός στις τιμές. "Η κύρια διαφορά ανάμεσα στα ανάμεσα στα δέντρα ταξινόμησης και στα δέντρα παλινδρόμησης, είναι ότι τα πρώτα δημιουργούνται με μη ταξινομημένες εξαρτώμενες

[16] "Data mining είναι η διαδικασία εύρεσης ανωμαλιών, συσχετίσεων και μοτίβων σε μεγάλα σύνολα δεδομένων με σκοπό την πρόβλεψη αποτελεσμάτων" (SAS, 2020)

μεταβλητές, ενώ τα δεύτερα παίρνουν ταξινομημένες μεταβλητές με συνεχείς τιμές" (Pulipaka, 2016).

Οι κατηγορίες προβλημάτων στα οποία χρησιμοποιούνται δέντρα αποφάσεων έχουν τα ακόλουθα κοινά χαρακτηριστικά (Teggi, 2020):

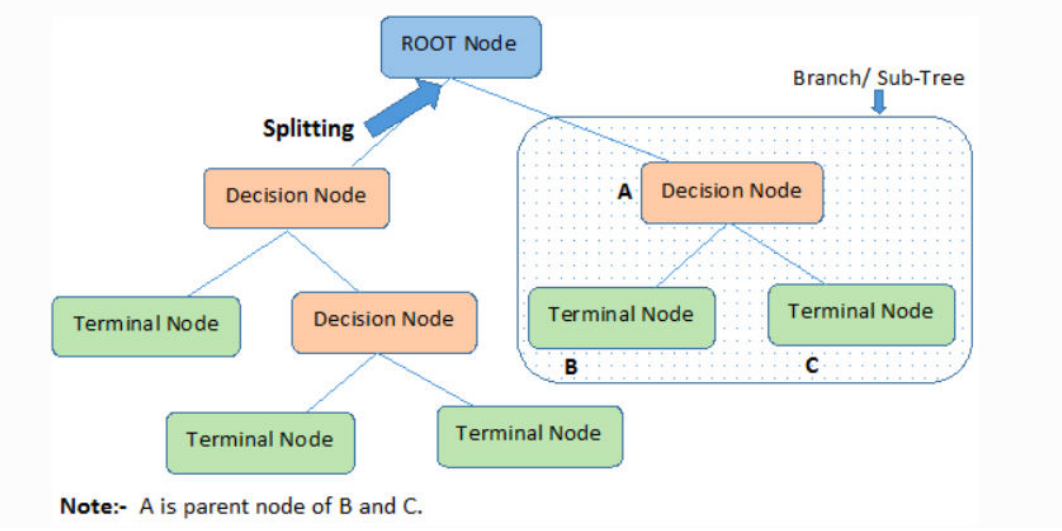
- Οι εγγραφές αναπαρίστανται από ζευγάρια χαρακτηριστικών-τιμών. Περιγράφονται από ένα συγκεκριμένο σύνολο χαρακτηριστικών (πχ Θερμοκρασία) και των τιμών τους (πχ ζεστό, χλιαρό, κρύο). Η ευκολότερη κατάσταση για ένα δέντρο αποφάσεων είναι όταν οι τιμές παίρνουν λίγες και διακριτές τιμές (όπως στο προηγούμενο παράδειγμα), αλλά με τις κατάλληλες μορφοποιήσεις ο αλγόριθμος μπορεί να διαχειριστεί και πραγματικές τιμές (στην προκειμένη περίπτωση την τιμή της θερμοκρασίας σε μία από τις γνωστές κλίμακες, πχ βαθμοί Κελσίου).
- Το χαρακτηριστικό που μελετά ο αλγόριθμος, και για το οποίο πρέπει να γίνει πρόβλεψη, έχει διακριτές τιμές. Σε ιδανικές συνθήκες παίρνει τις τιμές 0 και 1, που σημαίνει εκφράζει την πραγματοποίηση ή μη του ενδεχομένου. Και σε αυτήν την περίπτωση, με τις κατάλληλες επεκτάσεις και τροποποιήσεις ο αλγόριθμος μπορεί να υποστηρίξει και ενδεχόμενα με συνεχείς τιμές.
- Υπάρχουν διαχωρισμοί που χωρίζουν το dataset σε κατηγορίες. Τα δέντρα αποφάσεων λειτουργούν με αυτόν τον τρόπο και απεικονίζουν διαχωριστικές εκφράσεις, ακόμα και σε συνεχείς τιμές (πχ Ηλικία > 40, Ηλικία < 40).
- Τα δεδομένα εκμάθησης (training dataset) του αλγορίθμου μπορεί να περιέχουν λάθη. Οι μέθοδοι εκμάθησης των δέντρων αποφάσεων είναι ισχυρές απέναντι σε σφάλματα τόσο στις διακριτές τιμές των χαρακτηριστικών όσο και στην περιγραφή των ίδιων των attributes.
- Τα δεδομένα εκμάθησης μπορεί να περιέχουν ελλιπείς τιμές. Σε αυτήν την περίπτωση τα δέντρα αποφάσεων μπορούν να εκπαιδευτούν με μεγαλύτερη ακρίβεια και αποτελεσματικότητα σε σχέση με άλλες κατηγορίες αλγορίθμων.

Σε επιστημονικό άρθρο (Chauhan, 2019) που έχει γραφτεί από τον Data Scientist Nagesh Singh Chauhan, επεξηγείται αναλυτικά ο τρόπος λειτουργίας των Δέντρων Αποφάσεων, και αναλύονται βασικές ορολογίες που βοηθούν στην καλύτερη κατανόηση των κανόνων βάσει των οποίων δρουν οι συγκεκριμένοι αλγόριθμοι.

Για την καλύτερη αντίληψη σχετικά με τα decision trees πρέπει να γίνουν κατανοητοί οι όροι που χρησιμοποιούνται κατά κόρον:

1. **Ριζικός κόμβος**: Ο αρχικός κόμβος από τον οποίο ξεκινάει όλο το dataset, στον οποίο προφανώς περιέχονται όλα τα δεδομένα αφού δεν έχει γίνει ακόμα κάποιος διαχωρισμός.
2. **Διαχωρισμός**: Η διαδικασία κατά την οποία ένας κόμβος διαχωρίζεται σε δύο ή περισσότερους υποκόμβους.
3. **Κόμβος απόφασης**: Κόμβος εκτός του ριζικού, ο οποίος διαχωρίζεται περαιτέρω σε επόμενα βήματα εκτέλεσης του αλγορίθμου.
4. **Φύλλο/Τερματικός κόμβος**: Τελικός κόμβος ο οποίος δεν διαχωρίζεται περαιτέρω.
5. **Κλάδεμα**: Το αντίθετο του διαχωρισμού, η ελάττωση των κόμβων που γίνεται μέσω της αφαίρεσης υποκόμβων που ανήκουν σε ένα κόμβο απόφασης. Η συγκεκριμένη ενέργεια γίνεται για να επιτευχθεί μείωση της πολυπλοκότητας του αλγορίθμου έτσι ώστε να μπορούν να εξαχθούν συμπεράσματα με μεγαλύτερη ευκολία.
6. **Κλαδί/Υποδέντρο:** Μια υποενότητα του δέντρου αποφάσεων.
7. **Γονέας/Παιδί:** Ένας κόμβος απόφασης που περιέχει υποκόμβους αποκαλείται γονέας, και οι υποκόμβοι αποκαλούνται παιδιά.

Στην παρακάτω εικόνα είναι εμφανείς όλοι οι προαναφερθέντες όροι εκτός από την ενέργεια του κλαδέματος (pruning).



Εικόνα 8 - Δέντρο αποφάσεων στο οποίο απεικονίζονται οι σημαντικότεροι όροι και διαδικασίες. Πηγή: medium.com

"Οι αποφάσεις που καθορίζουν τα σημεία στα οποία θα γίνει διαχωρισμός επηρεάζουν σε μεγάλο βαθμό την ακρίβεια και την απόδοση ενός δέντρου αποφάσεων" (Chauhan, 2019). Χρησιμοποιούνται διάφοροι αλγόριθμοι οι οποίοι αποφασίζουν πότε χρειάζεται διαχωρισμός και

εάν ένας κόμβος πρέπει να διαχωριστεί σε δύο ή περισσότερους υποκόμβους. Οι κυριότεροι εξ' αυτών είναι οι παρακάτω:

- **ID3**: επέκταση του D3
- **C4.5**: διάδοχος του ID3
- **CART**: δέντρο ταξινόμησης και παλινδρόμησης
- **CHAID**: κάνει αυτόματη ανίχνευση της τιμής chi-square[17] και πραγματοποιεί διαχωρισμούς πολλών επιπέδων κατά τον υπολογισμό δέντρων ταξινόμησης
- **MARS**: multivariate adaptive regression splines. Μέθοδος παλινδρομικής ανάλυσης που εφευρέθη από τον Jerome H. Friedman το 1991 (Friedman, 1991)

Ένας εκ των παραπάνω αλγορίθμων, ο ID3 είναι ένας άπληστος αλγόριθμος που δημιουργεί δέντρα αποφάσεων χρησιμοποιώντας την καλύτερη λύση για κάθε χρονική στιγμή χωρίς να ανατρέχει προς τα πίσω. Τα βήματα που ακολουθούνται είναι τα εξής:

1. Επιλογή αρχικού κόμβου S που περιλαμβάνει το σύνολο δεδομένων
2. Σε κάθε επανάληψη του αλγορίθμου, παρατηρείται το πιο άσημο attribute του dataset και υπολογίζονται οι τιμές Entropy (H)[18] και Information Gain (IG)[19].
3. Στη συνέχεια επιλέγεται το χαρακτηριστικό που έχει το μικρότερο Entropy ή το μεγαλύτερο Information Gain.
4. Ο αρχικός κόμβος S διαχωρίζεται βάσει του επιλεγμένου χαρακτηριστικού και παράγει δύο ή περισσότερους υποκόμβους.
5. Ο αλγόριθμος συνεχίζει με αναδρομή[20] σε κάθε υποκόμβο, λαμβάνοντας υπόψιν μόνο attributes που δεν έχουν χρησιμοποιηθεί σε προηγούμενες επαναλήψεις.

Πλεονεκτήματα και περιορισμοί

[17] Chi-square: $x^2 = \sum \frac{(O-E)^2}{E}$ (7) , όπου O είναι τα observed frequencies (οι φορές που ένα ενδεχόμενο πραγματικά συνέβη) και E είναι τα expected frequencies (οι φορές που ένα ενδεχόμενο αναμένεται να συμβεί)

[18] Entropy είναι μια μεταβλητή που δείχνει το ποσοστό τυχαιότητας στα δεδομένα που είναι υπό επεξεργασία. Όσο μεγαλύτερη η τιμή της τόσο δυσκολότερο είναι να βγουν συμπεράσματα για αυτά τα δεδομένα. (Sujan, 2018)

[19] Information gain είναι μια στατιστική ιδιότητα που υπολογίζει πόσο αποτελεσματικά ένα δεδομένο χαρακτηριστικό (attribute) διαχωρίζει το training dataset ανάλογα με τον στόχο που έχει καθοριστεί. (Sujan, 2018)

[20] Αναδρομή στην Πληροφορική είναι η διαδικασία κατά την οποία μια μέθοδος καλεί τον εαυτό της συνεχώς και σπάει ένα αρχικό πρόβλημα σε πολλά μικρότερα προβλήματα βρίσκοντας την λύση με μια σχετική ευκολία.

Σύμφωνα με τον Data Scientist Dhiraj K., τα Δέντρα Αποφάσεων έχουν ορισμένα πλεονεκτήματα που τα καθιστούν κατάλληλα σε ορισμένους τύπους προβλημάτων. Έχουν όμως, όπως και όλοι οι αλγόριθμοι άλλωστε, και συγκεκριμένους περιορισμούς, οι οποίοι πρέπει να λαμβάνονται σοβαρά υπόψιν προκειμένου να μην χρησιμοποιούνται σε λανθασμένα use cases[21] (Dhiraj, 2019). Ξεκινώντας, ως συνήθως, από τα πλεονεκτήματα:

- **Δεν απαιτείται μεγάλη προσπάθεια και χρόνος στην προ-επεξεργασία των δεδομένων.** Πιο συγκεκριμένα, δεν είναι απαραίτητα βήματα ούτε η κανονικοποίηση (normalization) ούτε η προτυποποίηση (standardization/scaling) των δεδομένων.
- Οι ελλιπείς τιμές στα δεδομένα (missing data) δεν επηρεάζουν σε μεγάλο βαθμό την τελική έκβαση και τον τρόπο με τον οποίο θα δημιουργηθεί το δέντρο.
- **Υψηλή εμπορική αξία (business value).** Τα δέντρα αποφάσεων έχουν αποκτήσει τόσο μεγάλη απήχηση λόγω της απλότητας τους και της ευκολίας να επεξηγηθούν τόσο σε μια τεχνική ομάδα όσο και σε ένα διοικητικό συμβούλιο από stakeholders που ενδιαφέρεται μόνο για οπτικοποιημένα αποτελέσματα.
- **Ευκολία κατανόησης,** που συνδυάζεται άρρητα και με την ευκολία και της απλότητα της οπτικοποίησης τους. Θεωρείται τετριμμένη διαδικασία η απεικόνιση ενός δέντρου, όπως και η επεξήγηση του σε έναν άνθρωπο που δεν κατέχει τις απαραίτητες τεχνικές γνώσεις (if-else statements)

Και οι περιορισμοί που έχουν τα δέντρα αποφάσεων:

- **Αστάθεια.** Μια μικρή αλλαγή στο dataset στο οποίο εφαρμόζεται ένας tree based algorithm, μπορεί να προκαλέσει μεγάλη αλλαγή στη δομή του δέντρου, οδηγώντας τον αλγόριθμο στην δημιουργία νέου δέντρου με αυξημένη πολυπλοκότητα. Ο συγκεκριμένος περιορισμός κατά την προσωπική άποψη του συγγραφέα είναι αρκετά σημαντικός, καθώς στην καθημερινή εργασία ενός data scientist τα δεδομένα τα οποία μελετά μπορούν να υποστούν πολλές αλλαγές στο πέρασμα του χρόνου.
- **Overfitting.** Ένας tree based αλγόριθμος από την φύση του θα προσπαθήσει να καλύψει όλες τις περιπτώσεις/ενδεχόμενα ενός dataset, δημιουργώντας πολλές φορές αρκετούς

[21] "Ένα use case είναι μια λίστα βημάτων που καθορίζουν την αλληλεπίδραση ανάμεσα σε ένα ρόλο και σε ένα σύστημα με σκοπό την επίτευξη ενός στόχου" (Wikipedia, 2020). Με πιο απλά λόγια είναι ή περίπτωση στην οποία χρησιμοποιείται ένα εργαλείο για έναν συγκεκριμένο σκοπό.

κόμβους και αυξάνοντας σε αχρείαστο βαθμό την πολυπλοκότητα του. Κάτι τέτοιο, στο τέλος της ημέρας, θα οδηγήσει σε λανθασμένες προβλέψεις.

- **Κακή απόδοση σε μεγάλα datasets.** Σε συνέχεια του δεύτερου bullet, βγαίνει το συμπέρασμα ότι τα δέντρα αποφάσεων δεν αποδίδουν ικανοποιητικά σε μεγάλα και πολύπλοκα datasets, λόγω της δημιουργίας υπεράριθμων κόμβων.
- **Αδυναμία πρόβλεψης αμιγώς συνεχών τιμών.** Οι tree based αλγόριθμοι από κατασκευής τους δε μπορούν να προβλέψουν συνεχείς τιμές. Μπορούν μόνο να λειτουργήσουν ως regression trees, δηλαδή να δημιουργούν κόμβους βασισμένους σε συνεχείς ανεξάρτητες μεταβλητές (με χρήση ανισοτήτων), αλλά η εξαρτώμενη μεταβλητή πάνω στην οποία θα γίνει η πρόβλεψη θα πρέπει να είναι διακριτή (δυαδική ή περισσότερων κλάσεων).

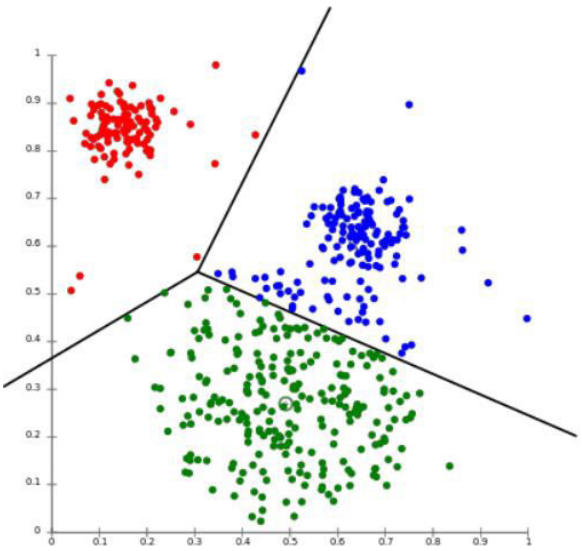
2.6 Θεωρητική ανάλυση γνωστών Unsupervised learning αλγορίθμων

Παρακάτω θα παρουσιαστούν δύο δημοφιλείς αλγόριθμοι μη εποπτευόμενης μάθησης, εκ των οποίων ο ένας (PCA) χρησιμοποιείται κυρίως στο στάδιο της προεπεξεργασίας δεδομένων:

- K-means clustering
- Principal Component Analysis (PCA)

2.6.1 K-means clustering

Περιγραφή



Εικόνα 9 - k-means clustering με 3 διακριτά clusters. Πηγή: amazon.com

Το k-means clustering είναι μια μη εποπτευόμενη μέθοδος κβαντοποίησης διανυσμάτων (vector quantization), διαχωρισμού δηλαδή ενός μεγάλου αριθμού παρατηρήσεων σε συγκεκριμένο αριθμό από συστάδες. Η ομαδοποίηση αυτή έχει ως στόχο να διαχωρίσει n παρατηρήσεις σε k ομάδες (όπου k ≤ n), έτσι ώστε κάθε παρατήρηση να ανήκει στη συστάδα με το

κοντινότερο μέσο, το οποίο χρησιμεύει ως ένα χαρακτηριστικό δείγμα της συστάδας. Αυτό οδηγεί σε μια διαμέριση του χώρου δεδομένων σε κελιά Voronoi[22].

Υπάρχουν διάφορα είδη αλγορίθμων που ανήκουν στην οικογένεια των k-means, με το πιο δημοφιλές εξ' αυτών να είναι ο αφελής (naive) k-means αλγόριθμος. Ο συγκεκριμένος χρησιμοποιεί μια επαναλαμβανόμενη τεχνική βελτίωσης και διόρθωσης, προκειμένου να καταλήξει στο τελικό αποτέλεσμα, με τα εξής βήματα:

- **Βήμα ανάθεσης**: Στο συγκεκριμένο βήμα κάθε παρατήρηση ανατίθεται στη συστάδα με τον κοντινότερο μέσο (δηλαδή αυτόν που έχει τη μικρότερη ευκλείδεια απόσταση στο τετράγωνο από την παρατήρηση)
- **Βήμα ενημέρωσης**: Μετά από την ανάθεση, υπολογίζονται εκ νέου οι μέσοι (centroids) του κάθε cluster. Αυτό γίνεται γιατί μετά από την είσοδο μιας παρατήρησης μέσα σε ένα cluster (ειδικά εάν πρόκειται για ακραία τιμή) επηρεάζεται η τιμή που θα έχει ο μέσος της συστάδας.

Η παραπάνω επαναλαμβανόμενη διαδικασία σταματά μόλις στο βήμα ενημέρωσης δεν προκύπτει σημαντική διαφορά στους μέσους, που σημαίνει ότι τα clusters έχουν λάβει την τελική τους μορφή. Η προαναφερθείσα διαδικασία δεν εγγυάται το βέλτιστο αποτέλεσμα και τον ιδανικό αριθμό δημιουργίας των cluster, ο οποίος συνήθως υπολογίζεται με άλλες τεχνικές (όπως πχ το Elbow method) (Steinley & Brusco, 2007).

Αφού παρουσιάστηκε η διαδικασία της δημιουργίας των συστάδων, στην ακόλουθη παράγραφο θα παρουσιαστούν και οι δυο επικρατέστερες μέθοδοι αρχικοποίησης των k-means αλγορίθμων:

- **Μέθοδος Forgy:** Η μέθοδος αυτή διαλέγει τυχαία k παρατηρήσεις (όσα δηλαδή θα είναι και τα clusters) και χρησιμοποιεί αυτές ως αρχικούς μέσους. Είναι μια από τις γρηγορότερες μεθόδους αρχικοποίησης (Thakur, 2020) και έχει αρκετά καλή απόδοση.
- **Random Partition:** Σε αυτή τη μέθοδο ανατίθεται τυχαία ένα cluster σε κάθε παρατήρηση και μετά ακολουθεί το βήμα της ενημέρωσης, δηλαδή ο υπολογισμός των μέσων.

[22] "Στα μαθηματικά, ένα διάγραμμα Voronoi είναι ένα διαχωρισμός ενός επιπέδου σε περιοχές που βασίζονται στην απόσταση από τα σημεία ενός συγκεκριμένου υποσυνόλου του επιπέδου." (Wikipedia, 2020)

Σύμφωνα με τους Hamerly Greg & Elkan Charles (Greg & Charles, 2002), η μέθοδος random partition χρησιμοποιείται περισσότερο για αφηρημένους k-means αλγορίθμους (όπως harmonic και fuzzy k-means) ενώ για τον κλασικό k-means και για μεγιστοποίηση πιθανοτήτων (expectation maximization) προτιμάται η πρώτη μέθοδος.

Οι k-means αλγόριθμοι, λόγω της δημοφιλίας και της ευκολίας χρήσης τους, χρησιμοποιούνται σε αρκετές εφαρμογές με τις πιο σημαντικές εξ΄ αυτών να είναι:

- **Κβαντοποίηση διανυσμάτων.** Η συνηθέστερη εφαρμογή που γνωρίζουν οι k-means αλγόριθμοι είναι στην κβαντοποίηση χρωμάτων, που είναι η διαδικασία μείωσης της χρωματικής παλέτας σε μια εικόνα, σε έναν συγκεκριμένο αριθμό χρωμάτων k (όπου k = clusters). Κάτι τέτοιο σαφώς μειώνει δραστικά το μέγεθος μιας εικόνας και χρησιμοποιείται σε τεχνικές συμπίεσης.
- **Ανάλυση συστάδων.** Χρησιμοποιώντας μη εποπτευόμενη μάθηση, δηλαδή μόνο δεδομένα εισόδου, οι k-means αλγόριθμοι δημιουργούν ορισμένο αριθμό συστάδων. Η εύρεση του βέλτιστου αριθμού ωστόσο, όπως προαναφέρθηκε, αποτελεί ξεχωριστή διαδικασία και παίζει μεγάλο ρόλο στην τελική απόδοση του αλγορίθμου.
- **Εκμάθηση χαρακτηριστικών.** Στην συγκεκριμένη εφαρμογή οι k-means χρησιμοποιούνται συνδυαστικά και με άλλες μεθόδους supervised ή semi-supervised learning. Χαρακτηριστικό παράδειγμα η συνδυαστική χρήση με γραμμικούς ταξινομητές (linear classifiers) για αναγνώριση εικόνας και φυσική επεξεργασία γλώσσας (Natural Language Processing).

Πλεονεκτήματα και περιορισμοί

Ακολουθούν κάποια πλεονεκτήματα και κάποια μειονεκτήματα της συγκεκριμένης κατηγορίας αλγορίθμων, όπως παρουσιάζονται σε επίσημο άρθρο από την Google (Google, 2020):

- **Σχετικά απλή υλοποίηση** χωρίς ιδιαίτερες πολυπλοκότητες
- **Αποδοτικό scaling** σε μεγάλα μεγέθη από datasets
- **Εγγύηση προσέγγισης (convergence)**. Αυτό σημαίνει ότι έστω και προσεγγιστικά ο αλγόριθμος θα παράξει ένα αποτέλεσμα.
- Τα centroids μπορούν να αρχικοποιηθούν αμέσως χωρίς πολύπλοκους υπολογισμούς.
- **Προσαρμοστικότητα** σε νέα παραδείγματα και παρατηρήσεις.
- **Γενικεύεται σε διάφορα μεγέθη και σχήματα από clusters**, αναλόγως του είδους k-means που θα χρησιμοποιηθεί. Ενδεικτικά στη σελίδα της Wikipedia αναφέρονται περίπου 15

διαφορετικά είδη k-means αλγορίθμων, τα οποία σαφώς απαιτούν και εξειδικευμένες γνώσεις.

Και τα σημαντικότερα αρνητικά σημεία τους:

- **Χειροκίνητη εύρεση k**. Έχει αναφερθεί και παραπάνω πως δεν υπάρχει αυτοματοποιημένος τρόπος ο οποίος υπολογίζει με σιγουριά τον βέλτιστο αριθμό από συστάδες που πρέπει να φτιαχτούν. Αυτό σημαίνει πως το τελικό αποτέλεσμα εξαρτάται και από τον ερευνητή και την προεπεξεργασία που θα κάνει.
- **Μεγάλη εξάρτηση στις αρχικές μεταβλητές**. Σχετίζεται και με το παραπάνω, και ουσιαστικά σημαίνει πως οι αρχικές μεταβλητές (k και initial centroids) επηρεάζουν σε μεγάλο βαθμό το τελικό αποτέλεσμα. Για μικρά k κάτι τέτοιο μπορεί να περιοριστεί τρέχοντας το πείραμα αρκετές φορές, αλλά όσο αυξάνεται το k πιο προχωρημένες τεχνικές θα πρέπει να χρησιμοποιηθούν.
- **Δυσκολία συσταδοποίησης δεδομένων για clusters διαφορετικού μεγέθους και πυκνότητας**. Αναφέρθηκε ως πλεονέκτημα η ικανότητα γενίκευσης, η οποία όμως απαιτεί αρκετά εξειδικευμένες γνώσεις προκειμένου να υλοποιηθεί σωστά και να παράξει τα επιθυμητά αποτελέσματα.
- **Περιορισμένη απόδοση σε μεγάλες διαστάσεις**. Όσο αυξάνονται οι διαστάσεις θα πρέπει να χρησιμοποιούνται τεχνικές μείωσης διαστάσεων (dimensionality reduction) για να παραμένει σε ανεκτά επίπεδα η απόδοση.

2.6.2 Principal Component Analysis (PCA)

Περιγραφή

Ο τελευταίος αλγόριθμος που θα παρουσιαστεί στο θεωρητικό μέρος της πτυχιακής εργασίας είναι ο Principal Component Analysis (PCA). Ο συγκεκριμένος αλγόριθμος ανήκει στην οικογένεια των unsupervised learning αλγορίθμων, αλλά πολλές φορές χρησιμοποιείται στο στάδιο της προεπεξεργασίας των δεδομένων, για λόγους που θα αναλυθούν παρακάτω. Η μαθηματική θεωρία που κρύβεται πίσω από το PCA είναι αρκετά πολύπλοκη και δε θα καλυφθεί σε αυτήν την ενότητα· αντ' αυτού θα περιγραφεί απλούστερα ο τρόπος λειτουργίας και τα βήματα που ακολουθούνται και φυσικά θα γίνει αναφορά στα κυριότερα πλεονεκτήματα και μειονεκτήματα που έχει ο συγκεκριμένος αλγόριθμος.

"Το PCA είναι μια μέθοδος μείωσης διαστάσεων η οποία χρησιμοποιείται για να ελαχιστοποιήσει τον αριθμό χαρακτηριστικών σε μεγάλα και πολύπλοκα datasets, μετατρέποντας τα σε απλούστερα τα οποία συνεχίζουν να περιέχουν την πλειονότητα της σημαντικής

πληροφορίας” (Jaadi, 2021). Όπως είναι φυσικό η μείωση διαστάσεων ενός συνόλου δεδομένων πραγματοποιείται εις βάρος της ακρίβειας, αλλά τις περισσότερες φορές είναι σημαντικότερη η ανάλυση και η οπτικοποίηση απλών datasets από την απόλυτη ακρίβεια.

Παρακάτω τα κυριότερα στάδια από τα οποία περνάει η μέθοδος PCA:

1. **Κανονικοποίηση δεδομένων.** Το πρώτο είναι και το σημαντικότερο στάδιο, καθώς χωρίς την κανονικοποίηση δεν υπάρχει κανένα νόημα να γίνουν τα επόμενα βήματα και εν τέλει να αφαιρεθεί μέρος των attributes. Η κανονικοποίηση πραγματοποιείται στην αρχή του αλγορίθμου ούτως ώστε κάθε χαρακτηριστικό να έχει την ίδια βαρύτητα και να συνεισφέρει εξίσου στην ανάλυση. Μετά το συγκεκριμένο βήμα όλα τα χαρακτηριστικά θα βρίσκονται στην ίδια κλίμακα.
2. **Υπολογισμός πίνακα συνδιακύμανσης (covariance).** Ο σκοπός αυτού του βήματος είναι να κατανοήσει πώς οι μεταβλητές μεταβάλλονται από την μέση τιμή τους και πώς επηρεάζουν η μία την άλλη. Πολλές φορές χαρακτηριστικά που παρουσιάζουν πολύ υψηλή συσχέτιση (high correlation) μπορούν να αφαιρεθούν καθώς θεωρούνται περιττά. Ο πίνακας είναι δισδιάστατος διαστάσεων n*n, όπου n ο αριθμός των χαρακτηριστικών, και σε κάθε εγγραφή του περιέχεται ένας αριθμός που είναι η συνδιακύμανση του κάθε ζεύγους. Εάν η τιμή είναι θετική σημαίνει πως τα χαρακτηριστικά είναι ανάλογα, ενώ αν είναι αρνητική σημαίνει πως είναι αντιστρόφως ανάλογα.

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

Εικόνα 10 - Πίνακας συνδιακύμανσης 3 μεταβητών x,y,z. Πηγή: builtin.com

3. **Υπολογισμός ιδιοδιανυσμάτων και ιδιοτιμών (eigenvectors & eigenvalues).** Το τρίτο βήμα του αλγορίθμου, βάσει του οποίου θα φτιαχτούν τα κύρια συστατικά (principal components). Τα ιδιοδιανύσματα και οι ιδιοτιμές είναι οι οντότητες γραμμικής άλγεβρας που υπολογίζονται βάσει του πίνακα συνδιακύμανσης. Τα principal components από την άλλη είναι τα νέα χαρακτηριστικά, τα οποία είναι ανεξάρτητα μεταξύ τους και ουσιαστικά περιέχουν συμπυκνωμένη την πληροφορία όλων των χαρακτηριστικών που θα αφαιρεθούν. Χωρίς αναλυτική περιγραφή, η δημιουργία των principal components γίνεται εφικτή και βασίζεται αποκλειστικά στα ιδιοδιανύσματα και τις ιδιοτιμές· με φθίνουσα ταξινόμηση των eigenvectors αναγνωρίζονται τα πιο σημαντικά components του dataset. Στο τέλος του βήματος τα principal components θα είναι σε αριθμό όσα ήταν και τα αρχικά

χαρακτηριστικά, αλλά θα είναι ταξινομημένα βάσει σημαντικότητας, κάτι το οποίο θα βοηθήσει στο επόμενο βήμα.

4. **Μείωση διαστάσεων.** Σε αυτό το στάδιο ο αναλυτής επιλέγει να κρατήσει έναν αριθμό από τα πιο σημαντικά components, αφού πλέον τα έχει ταξινομημένα βάσει της πληροφορίας που περιέχουν και της σημαντικότητας τους. Να σημειωθεί ότι τα υποχρεωτικά βήματα ανάλυσης είναι τα πρώτα τρία που προαναφέρθηκαν, καθώς το συγκεκριμένο βήμα είναι προαιρετικό και είναι στην ευχέρεια του ερευνητή αν θα προχωρήσει σε μείωση διαστάσεων.

Πλεονεκτήματα και περιορισμοί

Σε επιστημονικό άρθρο του διαδικτύου (i2tutorials, 2019) παρουσιάζονται και αναλύονται τα πλεονεκτήματα και τα μειονεκτήματα του αλγορίθμου Principal Component Analysis:

- **Αφαίρεση υψηλά συσχετισμένων μεταβλητών.** Σε datasets που περιέχουν πραγματικά δεδομένα πολλές φορές παρατηρείται πληθώρα χαρακτηριστικών εκ των οποίων αρκετά είναι περιττά. Με το PCA εντοπίζονται τα χαρακτηριστικά εκείνα που παρουσιάζουν υψηλή συσχέτιση και αφαιρούνται, και στη θέση τους δημιουργούνται τα principal components τα οποία είναι ανεξάρτητα.
- **Βελτίωση απόδοσης machine learning αλγορίθμων.** Όπως προαναφέρθηκε το PCA χρησιμοποιείται κυρίως στο στάδιο της προεπεξεργασίας των δεδομένων, ούτως ώστε στη συνέχεια να εφαρμοστεί κάποιος machine learning αλγόριθμος πάνω στο επεξεργασμένο dataset. Εφόσον έχει προηγηθεί μείωση διαστάσεων προφανώς η απόδοση του αλγορίθμου θα είναι καλύτερη τόσο σε ακρίβεια όσο και σε χρόνους ολοκλήρωσης.
- **Ελαχιστοποίηση overfitting.** Το overfitting στα δεδομένα συμβαίνει συνήθως όταν υπάρχει μεγάλος αριθμός από attributes. Με το PCA μειώνεται αυτός ο αριθμός με αποτέλεσμα να ελαχιστοποιείται και το overfitting στο dataset.
- **Καλύτερη οπτικοποίηση δεδομένων.** Προφανές πλεονέκτημα, αφού καθίσταται από δύσκολο έως αδύνατο να οπτικοποιηθούν δεδομένα τα οποία έχουν πάρα πολλά χαρακτηριστικά.

Ακολουθούν τα μειονεκτήματα του PCA:

- **Χάνεται το νόημα των αρχικών χαρακτηριστικών.** Στην παρουσίαση των βημάτων του PCA αναφέρθηκε ότι δημιουργούνται νέα χαρακτηριστικά (Principal Components) τα οποία περιέχουν συμπυκνωμένη πληροφορία. Αυτό έχει ως αποτέλεσμα να μην έχουν κάποιο

διακριτό και ξεκάθαρο νόημα στο ανθρώπινο μάτι και συνεπώς να μη μπορούν να διαβαστούν εύκολα από τους αναλυτές.

- **Απαιτείται κανονικοποίηση των δεδομένων σε κάθε περίπτωση.** Το πρώτο και απαραίτητο βήμα προκειμένου να έχει νόημα η εφαρμογή PCA είναι η κανονικοποίηση των δεδομένων. Αυτό έχει υπολογιστικό κόστος, καθώς και το μειονέκτημα της μετατροπής των κατηγορικών μεταβλητών σε αριθμητικές.
- **Απουσία πληροφορίας.** Τα Principal Components, όση πληροφορία και αν περιέχουν, ποτέ δε θα φτάσουν στο 100% της αρχικής πληροφορίας που υπήρχε στο dataset πριν την μετατροπή. Κάτι τέτοιο μπορεί να έχει σοβαρές επιπτώσεις αν στο στάδιο της μείωσης διαστάσεων δεν επιλεχθούν τα σωστά Principal Components, αφού ενδέχεται αν χαθεί σημαντικό μέρος της πληροφορίας.

2.7 Θεωρητικά στοιχεία εμπλεκόμενου λογισμικού

Για την υλοποίηση του πρακτικού μέρους της πτυχιακής εργασίας έχει χρησιμοποιηθεί κυρίως η γλώσσα προγραμματισμού Python και η NoSQL βάση δεδομένων MongoDB. Η συγκεκριμένη ενότητα περιέχει κάποιες γενικές πληροφορίες θεωρητικής φύσεως για τα προαναφερθέντα, όπως και μια αναφορά διασύνδεσης των δυο τεχνολογιών:

- **Python.** Η Python είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου και γενικού σκοπού. Δίνει έμφαση στην καλή αναγνωσιμότητα του κώδικα και στην ευκολία εκμάθησης, και για αυτό το λόγο αποτελεί μια από τις δημοφιλέστερες και πιο αγαπημένες γλώσσες προγραμματισμού τα τελευταία χρόνια (stackoverflow, 2020). Με τη χρήση εξωτερικών βιβλιοθηκών που θα αναφερθούν και στην ενότητα «Μεθοδολογία», η Python χρησιμοποιείται σε μεγάλο βαθμό για Machine Learning και Data Analysis.
- **MongoDB.** Η mongo είναι μια NoSQL, document-oriented βάση δεδομένων η οποία χρησιμοποιείται για την αποθήκευση και γρήγορη ανάκτηση δεδομένων. Η πρώτη έκδοση βγήκε το 2009 και έκτοτε έχει γνωρίσει μεγάλη άνθηση. Ο όρος NoSQL σημαίνει “Not Only SQL” και πρακτικά σημαίνει ότι δεν περιέχει τους σχεσιακούς (και πολλές φορές αυστηρούς) κανόνες που υπάρχουν στις SQL βάσεις. Το document-oriented από την άλλη σημαίνει ότι στη βάση αποθηκεύονται έγγραφα τύπου BSON (που είναι συνδυασμός JSON με δυαδικά δεδομένα) τα οποία δε χρειάζεται να έχουν κάποια προκαθορισμένη μορφή.

Αναφορικά με τη διασύνδεση μεταξύ τους, στο επίσημο documentation της MongoDB αναφέρεται πως “ο PyMongo είναι ο επίσημος MongoDB Python driver. Συνιστούμε να τον χρησιμοποιήσετε για να δουλέψετε πάνω στην MongoDB μέσω Python” (docs.mongodb, 2021).

2.8 Σχετικό ερευνητικό έργο

Στην συγκεκριμένη ενότητα θα παρουσιαστούν πρόσφατα έργα και δημοσιεύσεις σχετικές με την επιστήμη της μηχανικής εκμάθησης. Έχουμε φτάσει σε μια εποχή όπου κάθε ανεπτυγμένη εταιρία εφαρμόζει τεχνολογίες Machine Learning και Big Data, προκειμένου να αποκτήσει πλεονέκτημα στην αγορά σε σχέση με τους ανταγωνιστές της. Για τον λόγο αυτό έχουν αυξηθεί ραγδαία και οι δημοσιεύσεις σε σχετικά επιστημονικά πεδία, με χιλιάδες papers να υποβάλλονται κάθε χρόνο σε δημοφιλείς πλατφόρμες δημοσίευσης όπως NeurIPS, ICML, ICLR, ACL, και MLDS (Kumar P. , 2020). Να σημειωθεί ότι ο κώδικας που έχει γραφτεί για τα περισσότερα projects ανήκει στην κατηγορία του ανοιχτού λογισμικού, που σημαίνει ότι είναι δημόσια διαθέσιμα στο διαδίκτυο και μπορεί ο καθένας να τα δει και να τα επεκτείνει.

The Tree Ensemble Layer: Differentiability meets Conditional Computation

Η πρώτη δημοσίευση που σαφώς είναι άξια αναφοράς, είναι μια από τις σημαντικότερες για το έτος 2020. Πρόκειται για το paper με τίτλο «The Tree Ensemble Layer: Differentiability meets Conditional Computation» (Hazimeh, Ponomareva, Mol, Tan, & Mazumder, 2020) , στο οποίο οι ερευνητές συνδυάζουν τα δέντρα αποφάσεων με τα νευρωνικά δίκτυα σε έναν υβριδικό αλγόριθμο. Το αποτέλεσμα αυτής της συγχώνευσης είναι η εκμετάλλευση των πλεονεκτημάτων και των δυο κατηγοριών. Πιο συγκεκριμένα, από τα δέντρα αποφάσεων εκμεταλλεύτηκαν την υπολογιστική τους ικανότητα υπό συνθήκη (conditional computation), ενώ από τα νευρωνικά δίκτυα την ιδιότητα επεξεργασίας και διαμόρφωσης των χαρακτηριστικών (feature engineering), κάτι το οποίο το κάνουν εκ φύσεως. Ενδεικτικό της αξίας του συγκεκριμένου έργου είναι τα αστέρια (18.000) και η τεράστια δημοτικότητα που έχουν στο github[23].

End-to-End Object Detection with Transformers

Συνεχίζοντας με ένα ακόμη πολύ ενδιαφέρον paper, από ερευνητές του τμήματος Έρευνας και Ανάπτυξης του Facebook (Carion, et al., 2020). Στο συγκεκριμένο παρουσιάζεται ένα νέο μοντέλο αναγνώρισης εικόνας το οποίο αποφεύγει τους πολύπλοκους υπολογισμούς και την ανάγκη για ανθρώπινη παρέμβαση κατά τη διάρκεια της εκμάθησης. Σε αντίθεση με ανταγωνιστικά μοντέλα τα οποία έχουν τον ίδιο σκοπό, στο DETR (Detection Transformer όπως

[23] Δημόσιο (και ιδιωτικό) αποθετήριο της Microsoft στο οποίο ο καθένας μπορεί να ανεβάσει κάποιο προγραμματιστικό project.

ονομάστηκε) έχουν αφαιρεθεί τόσο η προεπεξεργασία όσο και τα βήματα που γίνονται κατόπιν της ανάλυσης, τα οποία επηρεάζουν σε μεγάλο βαθμό την αποδοτικότητα των αλγορίθμων. Με σχετικά απλή αρχιτεκτονική και ξεκάθαρη υλοποίηση, το συγκεκριμένο μοντέλο φαίνεται πολλά υποσχόμενο για το μέλλον του image recognition.

Language Models are Few-Shot Learners

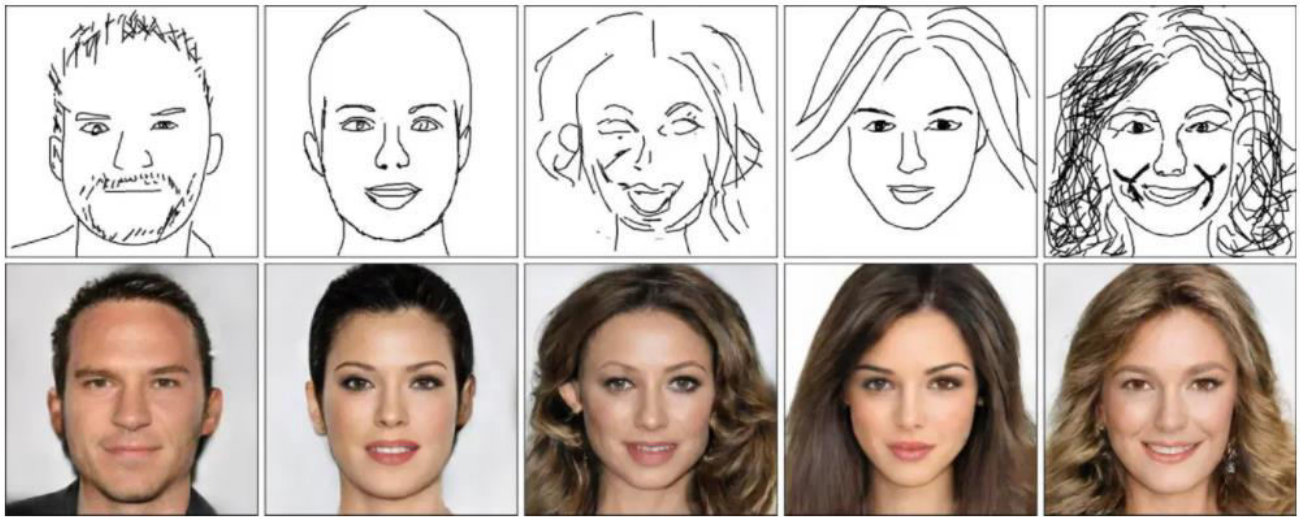
Προχωράμε στο ευρύτατο πεδίο επεξεργασίας φυσικής γλώσσας, πάνω στο οποίο βασίζονται chatbots, ψηφιακοί βοηθοί και τεχνικές AI οι οποίες αναγνωρίζουν και κατακερματίζουν γραπτό και προφορικό κείμενο με ταχύτατους ρυθμούς. Μηχανικοί της εταιρίας Open AI προχώρησαν σε επιτυχή δημοσίευση στην οποία εκπαίδευσαν το NLP μοντέλο GPT-3[24] με 175 δισεκατομμύρια μεταβλητές (Brown, et al., 2020). Για να γίνει κατανοητό το συγκεκριμένο επίτευγμα αξίζει να αναφερθεί πως μέχρι και τη στιγμή της δημοσίευσης το αποδοτικότερο NLP μοντέλο ήταν στην κατοχή της Microsoft και είχε "μόλις" 17 δισεκατομμύρια μεταβλητές.

Unsupervised Translation of Programming Languages

Εκτός από τις παραπάνω κατηγορίες δημοσιεύσεων που ανήκουν στο ευρύτερο επιστημονικό πεδίο του Data Science, αξίζει να αναφερθεί και ένα paper το οποίο σχετίζεται αποκλειστικά με προγραμματισμό και λογισμικό. Πρόκειται για τη δημοσίευση «Unsupervised Translation of Programming Languages» (Lachaux, Roziere, Chanussot, & Lample, 2020) από το τμήμα AI του Facebook, στην οποία δημιουργήθηκε ένας ικανότατος μεταφραστής προγραμματιστικών γλωσσών. Πιο συγκεκριμένα, όπως αναφέρουν και οι ερευνητές, δημιουργήθηκε μεταγλωττιστής με τη χρήση unsupervised learning σε νευρωνικά δίκτυα, ο οποίος μπορεί να μεταφράζει πηγαίο κώδικα με πολύ καλές αποδόσεις ανάμεσα σε 3 δημοφιλείς γλώσσες προγραμματισμού (Java, Python, C++). Μια τέτοια μετάφραση συνήθως αποτελεί μεγάλη πρόκληση και απαιτεί τεράστιο κόστος υλοποίησης για εταιρίες λογισμικού, πρόβλημα που πιθανότατα θα μετριαστεί μελλοντικά από μεταγλωττιστές αυτού του είδους.

[24] Generative Pre-trained Transformer 3. Χαρακτηρίζεται ως το πιο επιτυχές μοντέλο πρόβλεψης γλώσσας έως σήμερα (Marr, 2020)

DeepFaceDrawing: Deep Generation of Face Images from Sketches



Εικόνα 11 - Αποτέλεσμα DeepFaceDrawing. Πηγή: https://rubikscode.net/

Τέλος, θα αναφερθεί μια αρκετά επαναστατική δημοσίευση του 2020, μέσω της οποίας παράγονται πορτραίτα υψηλής ευκρίνειας από απλά σκίτσα. Χαρακτηριστική είναι η εικόνα 9, στην οποία βλέπουμε τις ικανότητες του DeepFaceDrawing μοντέλου που ανέπτυξαν Κινέζοι ερευνητές (Chen, Su, Gao, Xia, & Fu, 2020). Οι εντυπωσιακές αυτές επιδόσεις μπορούν να χρησιμοποιηθούν τόσο για καλλιτεχνικούς σκοπούς (δημιουργία χαρακτήρων) όσο και για πιο σοβαρούς λόγους όπως στην αναγνώριση προσώπων στην εγκληματολογία. Το μοντέλο που παράγει τα παραπάνω αποτελέσματα ήταν τόσο αποτελεσματικό γιατί, σύμφωνα με τη δημοσίευση, βασίστηκε πάνω σε ανεπτυγμένες τεχνικές Deep Learning μέσω των οποίων κατάφερνε να συνθέτει με τέτοια ακρίβεια τα πορτραίτα έχοντας ως input ανακριβή και απλά σκίτσα.

ΚΕΦΑΛΑΙΟ 3 - Μεθοδολογία

Στο κεφάλαιο αυτό θα αναφερθούν οι μέθοδοι που ακολουθήθηκαν τόσο σχετικά με τη βιβλιογραφική ανασκόπηση, όσο και για την ανάπτυξη του σχετικού λογισμικού.

3.1 Θεωρητική προσέγγιση

Η μεθοδολογία που χρησιμοποιήθηκε για την συγγραφή του θεωρητικού μέρους της παρούσας εργασίας ήταν ενδελεχής έρευνα βασισμένη σε έμπιστες πηγές του διαδικτύου, άρθρα και δημοσιεύσεις σχετικές με αλγορίθμους και machine learning, η οποία σε συνδυασμό με τις ήδη υπάρχουσες γνώσεις του συγγραφέα από τον χώρο της πληροφορικής διαμόρφωσαν το τελικό αποτέλεσμα. Οι συγγραφείς των άρθρων που χρησιμοποιήθηκαν ως πηγές είναι στην πλειοψηφία επαγγελματίες data scientists οι οποίοι έχουν εμπειρία στην ανάλυση δεδομένων και στην υλοποίηση των αλγορίθμων που αναλύθηκαν. Υποστηρίζεται ακράδαντα η άποψη που λέει ότι στην σύγχρονη εποχή, οποιοσδήποτε έχει πρόσβαση στο διαδίκτυο και γνωρίζει πώς να κάνει σωστή έρευνα μπορεί να βρει πληροφορίες για οποιοδήποτε θέμα, επιστημονικό ή μη, αλλά και να διασταυρώσει αυτές τις πληροφορίες βασιζόμενος σε επιστημονικές μελέτες και ακαδημαϊκές δημοσιεύσεις.

3.2 Πρακτικές υλοποίησης

Στο προγραμματιστικό σκέλος, έγινε μελέτη του επίσημου documentation[25] της python, καθώς και των κύριων βιβλιοθηκών[26] της που χρησιμοποιούνται σε Data Science projects και όχι μόνο. Όλες οι υλοποιήσεις/δοκιμές πραγματοποιήθηκαν σε υπολογιστικό σύστημα με τις εξής προδιαγραφές:

- Λειτουργικό σύστημα: Windows 10 64-bit
- Επεξεργαστής: Intel Core i7-8550U @ 1.80Ghz

[25] Python 3 official documentation: https://docs.python.org/3/
[26] Pandas official documentation: https://pandas.pydata.org/docs/
Scikit-learn official documentation: https://scikit-learn.org/stable/
Matplotlib official documentation: https://matplotlib.org/3.2.1/index.html
NumPy official documentation: https://numpy.org/doc/1.18/user/index.html

- RAM: 8,00 GB

Κάθε Jupyter notebook που υλοποιήθηκε, μετατράπηκε στη συνέχεια σε Python script και χρονομετρήθηκε, προκειμένου να αξιολογηθεί και χρονικά ο κάθε αλγόριθμος. Σημαντικό να τονιστεί επίσης πως οι χρόνοι εκτέλεσης επηρεάζονται σε περίπτωση που το μηχάνημα είναι λάπτοπ, αναλόγως του αν είναι στην πρίζα, καθώς ο επεξεργαστής αποδίδει σε υψηλότερες συχνότητες όταν παίρνει ρεύμα και από τον φορτιστή. Οι δοκιμές του συγγραφέα έγιναν σε λάπτοπ το οποίο βρισκόταν σε φόρτιση συνεχώς.

Μια σύντομη αναφορά στο βοηθητικό λογισμικό που χρησιμοποιήθηκε, το οποίο προτείνεται από τον συγγραφέα αλλά δεν αποτελεί απαραίτητη προϋπόθεση για την εκτέλεση των προγραμμάτων που αναπτύχθηκαν:

- **JetBrains PyCharm (Community Edition)**: Δωρεάν IDE (Integrated Development Environment), ειδικά σχεδιασμένο για την Python, ανεπτυγμένο από την τσέχικη εταιρία JetBrains.
- **Visual Studio Code**: Open-Source text editor γενικού σκοπού από την Microsoft, ο οποίος προσφέρει πολλές δυνατότητες μέσω κατάλληλων επεκτάσεων (πχ jupyter notebook support).
- **Robo 3T**: Open-Source δημοφιλές γραφικό περιβάλλον (GUI) διασύνδεσης με βάσεις δεδομένων MongoDB, ανεπτυγμένο από την εταιρία 3T Software Labs.

ΚΕΦΑΛΑΙΟ 4 - Ανάπτυξη Machine Learning λογισμικού

Το παρόν κεφάλαιο περιέχει λεπτομέρειες σχετικά με την ανάπτυξη του λογισμικού που έχει αναφερθεί, διερευνητική ανάλυση των datasets, καθώς και χρήσιμα συμπεράσματα που προκύπτουν από τις εκτελέσεις των αλγορίθμων.

4.1 Διερευνητική ανάλυση δεδομένων

Τα datasets που επιλέχθηκαν να αναλυθούν ανήκουν στον τομέα «Ενέργεια και Κτίρια», ο οποίος είναι αρκετά δημοφιλής αυτήν την περίοδο λόγω των δυσοίωνων προβλέψεων της επιστημονικής κοινότητας για το περιβάλλον του πλανήτη μας. Έχουν αντληθεί από δημόσιο αποθετήριο[27] το οποίο αποτελεί μέρος της δημοσίευσης "Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models" (Candanedo & Feldheim, 2015). Στο συγκεκριμένο paper έχουν υλοποιηθεί στατιστικά μοντέλα μηχανικής εκμάθησης σε R βάσει των οποίων έχουν εξαχθεί χρήσιμα συμπεράσματα τα οποία αναγράφονται λεπτομερώς σε αυτό. Στο πλαίσιο αυτής της πτυχιακής εργασίας δε θα γίνει περαιτέρω ανάλυση της προαναφερθείσας δημοσίευσης· ο σκοπός είναι να χρησιμοποιηθούν τα ίδια σύνολα δεδομένων και να γίνει εκ νέου υλοποίηση machine learning αλγορίθμων σε περιβάλλον Python.

Τα δεδομένα χωρίζονται σε τρία CSV αρχεία (datatest.csv, datatest2.csv, datatraining.csv) εκ των οποίων το ένα αρχείο (datatraining) χρησιμοποιείται για την εκπαίδευση των αλγορίθμων. Στον παρακάτω πίνακα, μπορούμε να δούμε αναλυτικά τα χαρακτηριστικά (attributes) των datasets:

Attribute	**Values (approx.)**	**Description**
SN	1-10.000	Sequence number, αύξων αριθμός της κάθε εγγραφής.
Date	YYYY-MM-DD HH:MM:SS	Ημερομηνία και ώρα που έλαβαν χώρα οι μετρήσεις της εγγραφής.

[27] https://github.com/LuisM78/Occupancy-detection-data

Temperature	19 - 25 ◦C	Η θερμοκρασία του χώρου σε βαθμούς Κελσίου.
Humidity	20-40 RH (%)	Ποσοστό υγρασίας στον χώρο.
Light	0-1600 Lux	Μονάδα μέτρησης του φωτός.
CO2	480 - 2076 Ppm	Μονάδα μέτρησης του διοξειδίου του άνθρακα.
HumidityRatio	$W = 0.622 \frac{pw}{p-pw}$	Σχετική υγρασία στο χώρο, η οποία υπολογίζεται βάσει της θερμοκρασίας και της υγρασίας.
Occupancy	0-1	Η εξαρτώμενη μεταβλητή, πάνω στην οποία θα γίνουν οι προβλέψεις. Το 0 υποδηλώνει απουσία ανθρώπων στον χώρο που έγινε η μέτρηση, και το 1 υποδηλώνει παρουσία.

Πίνακας 1 - Πίνακας Χαρακτηριστικών

Παρατηρώντας τον ανωτέρω πίνακα και σε συνδυασμό με τις πληροφορίες που αναγράφονται στο paper, συμπεραίνεται ότι τα δεδομένα αφορούν περιβαλλοντικές μετρήσεις από διάφορους χώρους ενός κτιρίου στο οποίο είχαν τοποθετηθεί κατάλληλες συσκευές μετρήσεων (σένσορες). Η εξαρτώμενη μεταβλητή occupancy συμπληρώθηκε με το χέρι από ερευνητές, έπειτα από φωτογραφίες των δωματίων που τραβούσαν ψηφιακές κάμερες.

Ξεκινώντας την προγραμματιστική ανάλυση, με τη βοήθεια ενός Python script (df_info_to_excels.py), παράγουμε 6 αρχεία excel τα οποία μας δίνουν βασικές πληροφορίες για τα 3 datasets που έχουμε (2 αρχεία ανά dataset). Το ένα αρχείο περιέχει βασικές πληροφορίες όπως αριθμό στηλών (χαρακτηριστικών) και γραμμών (εγγραφών), ενώ το δεύτερο αρχείο περιέχει περισσότερες λεπτομέρειες όπως μέση τιμή, τυπική απόκλιση, ελάχιστες / μέγιστες τιμές κλπ. Στον παρακάτω πίνακα υπάρχουν συγχωνευμένα τα 3 πρώτα αρχεία excel, που δείχνουν τις βασικές πληροφορίες για τα δεδομένα:

	datatest	**datatest2**	**datatraining**	**Total**
lines	2665	9752	8143	20560
columns	8	8	8	
total_elements	21320	78016	65144	164480
null_elements	0	0	0	0

Πίνακας 2 - Γενικές πληροφορίες για τα 3 datasets

Αρχικά παρατηρείται ότι δεν υπάρχουν κενές τιμές, οπότε δε θα χρειαστεί να γίνει κάποια ιδιαίτερη διαχείριση σε αυτό το κομμάτι. Επίσης βλέπουμε ότι και τα 3 datasets έχουν τον ίδιο αριθμό στηλών, κάτι το οποίο μπορεί να επιβεβαιωθεί με το μάτι, καθώς και σε συνδυασμό με τον πίνακα 1. Στη συνέχεια βλέπουμε ότι έχουμε συνολικά 20.560 εγγραφές, εκ των οποίων οι 8.143 θα χρησιμοποιηθούν για την εκπαίδευση των αλγορίθμων, αριθμοί οι οποίοι σε γενικές γραμμές μπορούν να κριθούν ικανοί έτσι ώστε να εξαχθούν χρήσιμα συμπεράσματα.

Θα συνεχίσουμε με τους εναπομείναντες 3 πίνακες, οι οποίοι δείχνουν περισσότερες λεπτομέρειες σχετικά με την κατανομή των στοιχείων:

datatest	Temperature	Humidity	Light	CO2	HumidityRatio	Occupancy
count	2665	2665	2665	2665	2665	2665
mean	21,43387629	25,3539368	193,2275556	717,9064701	0,00402701	0,364727955
std	1,028024176	2,436842325	250,2109058	292,6817184	0,000610573	0,481444128
min	20,2	22,1	0	427,5	0,003303314	0
25%	20,65	23,26	0	466	0,003529482	0
50%	20,89	25	0	580,5	0,00381507	0
75%	22,35666667	26,85666667	442,5	956,3333333	0,004531535	1
max	24,40833333	31,4725	1697,25	1402,25	0,005377759	1

Πίνακας 3 - datatest.csv details

datatest2	Temperature	Humidity	Light	CO2	HumidityRatio	Occupancy
count	9752	9752	9752	9752	9752	9752
mean	21,00176844	29,89191021	123,0679297	753,2248317	0,004588778	0,210110747
std	1,020693215	3,952843807	208,2212751	297,0961136	0,000530985	0,407407954
min	19,5	21,865	0	484,6666667	0,003274764	0
25%	20,29	26,64208333	0	542,3125	0,004196307	0
50%	20,79	30,2	0	639	0,00459331	0
75%	21,53333333	32,7	208,25	831,125	0,004997966	0
max	24,39	39,5	1581	2076,5	0,005768608	1

Πίνακας 4 - datatest2.csv details

datatraining	Temperature	Humidity	Light	CO2	HumidityRatio	Occupancy
count	8143	8143	8143	8143	8143	8143
mean	20,61908364	25,73150729	119,5193745	606,5462432	0,003862507	0,212329608
std	1,016916441	5,531211	194,7558048	314,3208769	0,000852331	0,408982013
min	19	16,745	0	412,75	0,002674127	0
25%	19,7	20,2	0	439	0,003078284	0
50%	20,39	26,2225	0	453,5	0,00380077	0
75%	21,39	30,53333333	256,375	638,8333333	0,00435193	0
max	23,18	39,1175	1546,333333	2028,5	0,006476013	1

Πίνακας 5 - datatraining.csv details

Παρατηρείται ότι οι 3 πίνακες βρίσκονται σχετικά κοντά στις τιμές, κάτι που σημαίνει ότι δεν υπάρχει μεγάλη απόκλιση στα datasets. Με κίτρινο έχουν τονιστεί οι τιμές εκείνες οι οποίες παρουσιάζουν μεγαλύτερη απόκλιση σε σχέση με τις αντίστοιχες των άλλων 2 datasets.

4.2 Προεπεξεργασία δεδομένων

Όπως προαναφέρθηκε δεν υπάρχουν πολλές ενέργειες για να γίνουν αναφορικά με το στάδιο του preprocessing, καθώς τα δεδομένα βρίσκονται σε αρκετά καλή μορφή, χωρίς ακραίες τιμές (outliers) και δίχως κενές (null) παρατηρήσεις. Σε διαφορετική περίπτωση θα έπρεπε να

αντιμετωπιστούν τόσο τα outliers (με αφαίρεση, αγνόηση ή πιο εξειδικευμένες μεθόδους) όσο και τα null values (με προσθήκη μέσων τιμών).

Παρ' όλα αυτά, για να μειωθεί ο αριθμός των χαρακτηριστικών και να υπάρχει μεγαλύτερη αποτελεσματικότητα στα πειράματα, θα αφαιρεθούν οι 2 πρώτες στήλες, οι οποίες είναι ο άυξων αριθμός και η ημερομηνία. Το SN (άυξων αριθμός) είναι απλά ένα μοναδικό αναγνωριστικό για τις εγγραφές και δεν παρέχει κάποια ουσιαστική πληροφορία για τα δεδομένα, ενώ η ημερομηνία είναι σε μορφή κειμένου και σε ακρίβεια δευτερολέπτων και στα πλαίσια αυτής της πτυχιακής θα αγνοηθεί τελείως. Αυτό αφήνει τα 3 datasets με τις 6 κύριες μεταβλητές, εκ των οποίων οι 5 είναι οι θεωρητικά ανεξάρτητες και η έκτη η εξαρτώμενη μεταβλητή (occupancy). Η αφαίρεση των δύο στηλών είναι πολύ απλή διαδικασία και θα γίνει και αυτή προγραμματιστικά μέσω python.

Συνεχίζουμε με τη δημιουργία correlation matrix, το οποίο μας δείχνει την συσχέτιση που έχουν οι μεταβλητές μεταξύ τους και σε τι ποσοστό επηρεάζουν η μια την άλλη. Για το συγκεκριμένο βήμα χρησιμοποιήθηκαν και τα 3 datasets, αλλά για καλύτερα αποτελέσματα συγχωνεύθηκαν σε 1 συνολικό dataset και εφαρμόστηκε εκεί η δημιουργία του πίνακα συσχέτισης:

	Temperature	Humidity	Light	CO2	HumidityRatio	Occupancy
Temperature	1	-0.16	0.69	0.45	0.21	0.56
Humidity	-0.16	1	-0.029	0.3	0.93	0.046
Light	0.69	-0.029	1	0.45	0.22	0.91
CO2	0.45	0.3	0.45	1	0.48	0.5
HumidityRatio	0.21	0.93	0.22	0.48	1	0.26
Occupancy	0.56	0.046	0.91	0.5	0.26	1

Εικόνα 12 - Correlation matrix for all 3 datasets

Από την εικόνα 12 άμεσα συμπεραίνεται ότι, η εξαρτώμενη μεταβλητή Occupancy, παρουσιάζει αρκετά υψηλή συσχέτιση με το χαρακτηριστικό Light. Αυτό πρακτικά σημαίνει ότι η παρουσία / απουσία ανθρώπων σε ένα δωμάτιο επηρεάζεται σε μεγάλο βαθμό από το πόσο φως υπάρχει σε αυτό το δωμάτιο· κάτι το οποίο είναι λογικό αφού όταν λείπουν όλοι από το χώρο πολλές φορές τα φώτα είναι σβηστά. Από την άλλη, παρατηρείται ότι το correlation του Occupancy σε σχέση με τα Humidity & HumidityRatio είναι αρκετά μικρό. Η δημιουργία correlation matrix μπορεί να βοηθήσει έναν ερευνητή σε μετέπειτα στάδιο εκτέλεσης των αλγορίθμων, όταν θα πρέπει να ληφθεί απόφαση που να αφορά τα χαρακτηριστικά του dataset (πχ σειρά των attributes σε ένα δέντρο αποφάσεων).

4.3 Υλοποίηση Supervised Machine Learning αλγορίθμων

4.3.1 Support Vector Machine

Ο πρώτος machine learning αλγόριθμος που υλοποιήθηκε πάνω στα δεδομένα είναι ο SVM (Support Vector Machine). Ο συγκεκριμένος αλγόριθμος είναι εποπτευόμενης μάθησης και είναι ευρέως χρησιμοποιούμενος τόσο σε προβλήματα ταξινόμησης όσο και σε παλινδρόμησης. Υλοποιήθηκε σε jupyter notebook και σε python script (svm.py) και αξιολογήθηκε βάσει των παρακάτω μετρικών:

- Accuracy
- Precision
- Recall
- F1
- Completion time

Οι εξισώσεις υπολογισμού των πρώτων τεσσάρων μετρικών αναγράφονται ως υποσημείωση στη σελίδα 26 και ουσιαστικά είναι τιμές που δείχνουν την απόδοση ενός αλγορίθμου, ενώ το Completion time είναι ο χρόνος εκπαίδευσης και εκτέλεσης του αλγορίθμου. Δεν χρονομετρήθηκαν τα υπόλοιπα στάδια του python script (load datasets, data preprocessing) καθώς αξιολογούνται αποκλειστικά οι αλγόριθμοι.

- Accuracy: 0.9896110171539019
- Precision: 0.9640564826700898
- Recall: 0.9943727242634889
- F1 Score: 0.9943727242634889
- Completion time: 0:00:21.699700

Παρατηρείται ότι η εκτέλεση ολοκληρώθηκε σε 21 δευτερόλεπτα και οι δείκτες απόδοσης είναι εξαιρετικοί, αφού και οι 4 αγγίζουν τη μονάδα που είναι η ανώτατη τιμή.

4.3.2 Logistic Regression

Στη συνέχεια δοκιμάστηκε ο Logistic Regression αλγόριθμος (με τον default solver lbfgs[28]) χρησιμοποιώντας τα ίδια μετρικά για την αξιολόγηση του και παρήγαγε τα παρακάτω αποτελέσματα:

- Accuracy: 0.9825239590883467
- Precision: 0.9648541114058355
- Recall: 0.9632571996027806
- F1 Score: 0.9632571996027806
- Completion time: 0:00:00.097992

Εδώ βλέπουμε ότι ο αλγόριθμος είναι πάλι εξαιρετικός σε ακρίβεια (με ελάχιστα χειρότερες αποδόσεις σε σχέση με τον SVM) αλλά ολοκληρώθηκε σε μόλις 1 δέκατο του δευτερολέπτου!

4.3.3 Naive Bayes

Ο επόμενος αλγόριθμος που δοκιμάστηκε είναι ο Naive Bayes, ο οποίος έχει αναλυθεί εκτενώς στην ενότητα 2.5.2. Οι εξαιρετικές επιδόσεις του και στους 4 επιστημονικούς τομείς accuracy, prediction, recall, F1 επιβεβαιώνονται και στην πράξη, όπως επίσης και οι μικροί χρόνοι:

- Accuracy: 0.9854232101151646
- Precision: 0.9487989886219975
- Recall: 0.9937106918238994
- F1 Score: 0.9937106918238994
- Completion time: 0:00:00.016961

Εκτός από το Precision το οποίο βρίσκεται σχετικά χαμηλά, οι υπόλοιπες τιμές είναι εξαιρετικές, με ιδιαίτερη αναφορά να γίνεται στον χρόνο εκτέλεσης ο οποίος βρίσκεται στα 16/100 του δευτερολέπτου, δηλαδή περίπου 6 φορές πιο χαμηλά από τον Logistic Regression.

4.3.4 Decision Tree

Η συγκεκριμένη ενότητα θα κλείσει με την εκτέλεση εντός ταξινομητή δέντρου αποφάσεων (Decision tree classifier). Περισσότερες πληροφορίες για τον τρόπο λειτουργίας των decision trees

[28] Περισσότερες πληροφορίες για Logistic regression solvers: https://towardsdatascience.com/dont-sweat-the-solver-stuff-aea7cddc3451

υπάρχουν στην ανάλυση που έχει προηγηθεί, στην ενότητα 2.5.3. Όπως θα δούμε στα μετρικά, η απόδοση του αλγορίθμου για τα συγκεκριμένα datasets είναι συγκριτικά η χειρότερη σε σχέση με τους υπόλοιπους αλγορίθμους, αλλά ο χρόνος εκτέλεσης είναι ιδιαίτερα χαμηλός και σε αυτήν την περίπτωση:

- Accuracy: 0.939840541193525
- Precision: 0.9300302571860817
- Recall: 0.8139688844753393
- F1 Score: 0.8139688844753393
- Completion time: 0:00:00.019999

Οι μεγαλύτερες διαφορές παρατηρούνται στα metrics F1 Score και Recall, τα οποία βρίσκονται περίπου 20% χαμηλότερα από τους υπόλοιπους αλγορίθμους. Αξίζει να σημειωθεί πως τα αποτελέσματα από εκτέλεση σε εκτέλεση του δέντρου αποφάσεων ποικίλλουν.

4.3.5 Σύνοψη αποτελεσμάτων - Συμπεράσματα

Ακολουθεί συνολικός πίνακας από τις παραπάνω εκτελέσεις των αλγορίθμων:

Algorithm	Accuracy	Precision	Recall	F1 Score	Time
SVM	0.99	0.96	0.99	0.99	**0:00:21.699**
Logistic Regression	0.98	0.96	0.96	0.96	0:00:00.097
Naive Bayes	0.98	0.95	0.99	0.99	0:00:00.016
Decision Tree	0.94	0.93	0.81	0.81	0:00:00.019

Πίνακας 6 - Συνολικός πίνακας αποτελεσμάτων

Με κόκκινο έχουν σημειωθεί οι χειρότερες αποδόσεις και με πράσινο ο καλύτερος συνολικά αλγόριθμος, που είναι ο Naive Bayes. Ο SVM μπορεί με ασφάλεια να θεωρηθεί ο χειρότερος στο συγκεκριμένο πείραμα, μόνο και μόνο από τον υπερβολικά μεγάλο χρόνο εκπαίδευσης/εκτέλεσης που είχε σε σχέση με τους ανταγωνιστές του.

Συνοψίζοντας, στην υποενότητα 4.3 δημιουργήθηκε Jupyter notebook και αντίστοιχα python scripts, στα οποία υλοποιήθηκαν 4 αλγόριθμοι εποπτευόμενης μάθησης και καταγράφηκαν οι επιδόσεις του καθένα πάνω σε 5 μετρικά. Το jupyter notebook βρίσκεται στο path app/helpers/algorithms.ipynb και τα python scripts (ένα για κάθε αλγόριθμο) στο path app/helpers/python_algorithms_scripts.

4.4 Υλοποίηση K-means clustering αλγορίθμου

Σε συνέχεια της υλοποίησης των supervised learning αλγορίθμων, υλοποιήθηκε μαζί με visualization και ο unsupervised learning αλγόριθμος K-means clustering, που έχει παρουσιαστεί

στην ενότητα 2.6.1. Το jupyter notebook που χρησιμοποιήθηκε είναι το ίδιο με παραπάνω και το αντίστοιχο python script ονομάζεται “kmeans.py”.

Αναφορικά με τα clusters δόθηκε η εντολή να κατασκευαστούν 2 στο σύνολο, καθώς η ομαδοποίηση που θα γίνει θα αφορά την εξαρτημένη μεταβλητή Occupancy, η οποία παίρνει τις τιμές 0 και 1, για απουσία και παρουσία ανθρώπων σε ένα δωμάτιο αντίστοιχα. Τα centroids επιλέχθηκαν με το χέρι (και φαίνονται στις εικόνες παρακάτω), μετά από αρκετές δοκιμές για την επίτευξη καλύτερων αποτελεσμάτων. Αυτό έγκειται στο γεγονός ότι για την βελτιστοποίηση ενός k-means αλγορίθμου χρειάζονται και χειροκίνητες ενέργειες από τον ερευνητή (όπως αναφέρεται στα αρνητικά στοιχεία σελ. 35).

Ένας k-means αλγόριθμος γίνεται άμεσα πιο κατανοητός εφόσον οπτικοποιηθεί κιόλας, και για τον λόγο αυτό επιλέχθηκαν τα 2 χαρακτηριστικά με το μεγαλύτερο correlation σε σχέση με το Occupancy (βλ. εικόνα 12), τα οποία είναι το Light (correlation=0.91) και Temperature (correlation=0.56). Το αποτέλεσμα είναι η δημιουργία δισδιάστατου γραφήματος με άξονες το φως και τη θερμοκρασία, και τις χρωματισμένες συστάδες να υποδεικνύουν εάν μια παρατήρηση έχει Occupancy 0 ή 1.

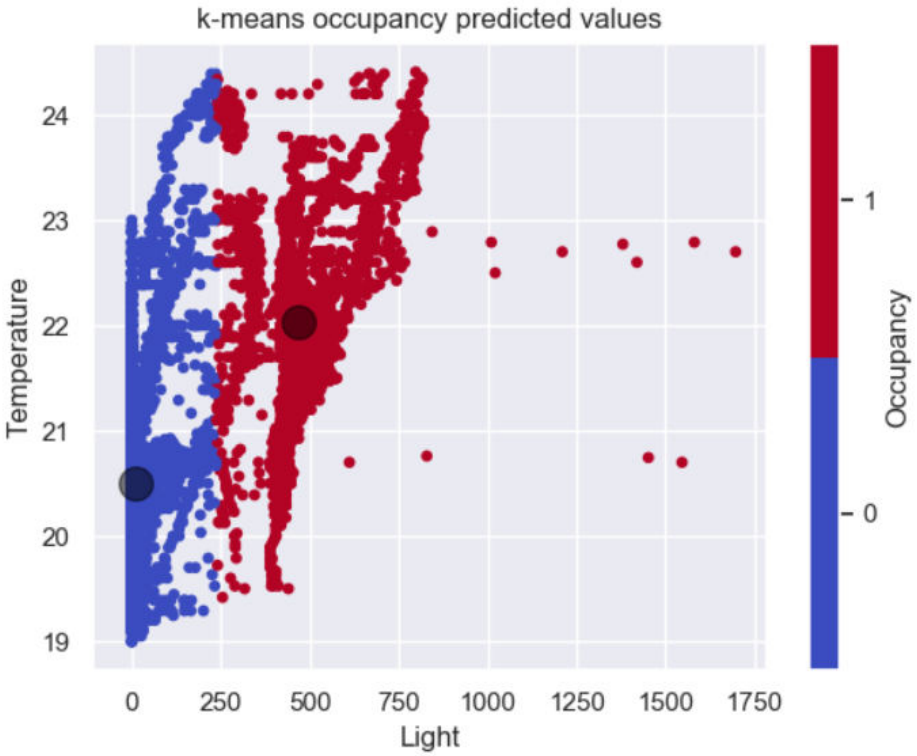
Όσον αφορά την αξιολόγηση και τη φύση (unsupervised) του αλγορίθμου, να αναφερθεί ότι δεν υπάρχει κάποια έτοιμη βιβλιοθήκη η οποία παράγει αυτόματα μετρικά για τον αλγόριθμο. Επίσης να τονιστεί ότι εδώ δεν χωρίζουμε το dataset σε test και train (όπως έγινε για τους supervised), συνεπώς και τα 3 datasets που έχουμε στη διάθεση μας συγχωνεύθηκαν σε 1 το οποίο συνολικά έχει 20560 παρατηρήσεις.

Η μεθοδολογία που ακολουθήσαμε για την εκτέλεση, αξιολόγηση και οπτικοποίηση του αλγορίθμου περιγράφεται αναλυτικά στα παρακάτω βήματα:

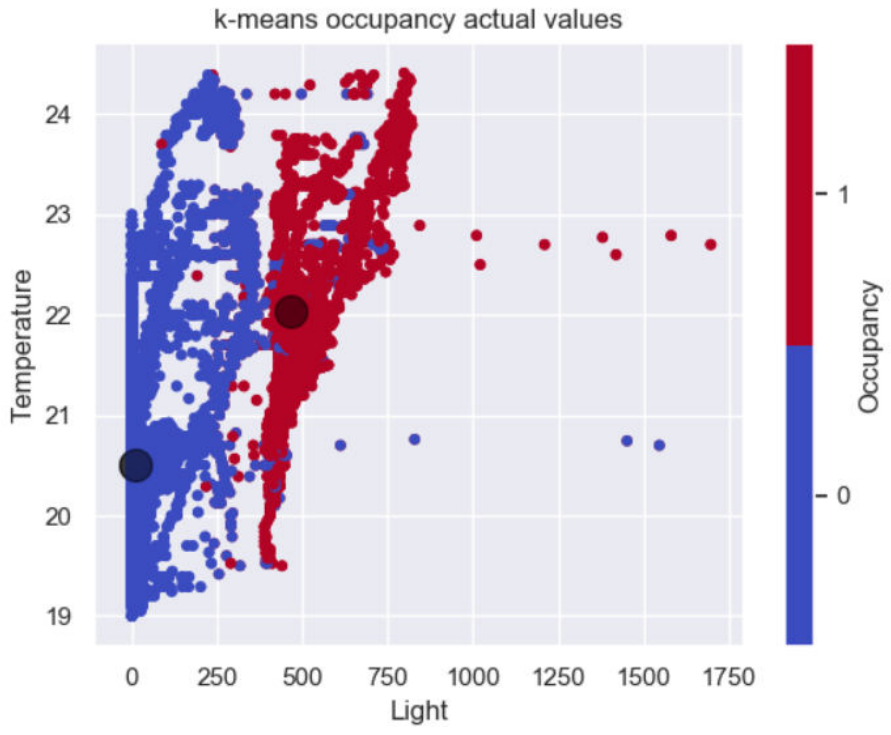
1. Συγχώνευση και των τριών datasets σε ένα και διαχωρισμός σε x (Light, Temperature) και y (Occupancy) υποσύνολα. Το y θα χρησιμοποιηθεί αποκλειστικά στην αξιολόγηση.
2. Εκπαίδευση και εκτέλεση αλγορίθμου, ο οποίος παράγει ένα σύνολο y_kmeans, το οποίο είναι της ίδιας μορφής με το y (20560 στοιχεία με τιμές 0 και 1).
3. Σύγκριση του y (actual values from dataset) με το y_kmeans (predicted values from k-means) και υπολογισμός ορθών και λανθασμένων προβλέψεων.
4. Δημιουργία και αποθήκευση 2 γραφημάτων, ένα με τις πραγματικές συστάδες και ένα με τις προβλεφθείσες.

Η ολοκλήρωση των παραπάνω παράγει τα ακόλουθα αποτελέσματα:

- Total elements: 20560
- Correct predictions: 19863
- Wrong predictions: 697
- Accuracy 96.61%



Εικόνα 13 - Προβλεφθέντα clusters (y_kmeans)



Εικόνα 14 - Πραγματικά clusters (y)

Αυτό που παρατηρείται είναι ότι λόγω του εξαιρετικά υψηλού correlation που έχει το attribute Light σε σχέση με το Occupancy, είναι το μόνο που πρακτικά επηρεάζει τις παρατηρήσεις. Χαρακτηριστική είναι η νοητή κάθετη γραμμή που διαχωρίζει τα 2 clusters, στις τιμές ~250 (εικόνα 13) και ~375 Lux (εικόνα 14). Αναφορικά με την ακρίβεια βλέπουμε ότι επιτυγχάνεται στο εξαιρετικό 96% , με το λανθασμένο 4% να βρίσκεται κυρίως στη ζώνη 250-375 Lux στην οποία ο αλγόριθμος έχει αποτύχει.

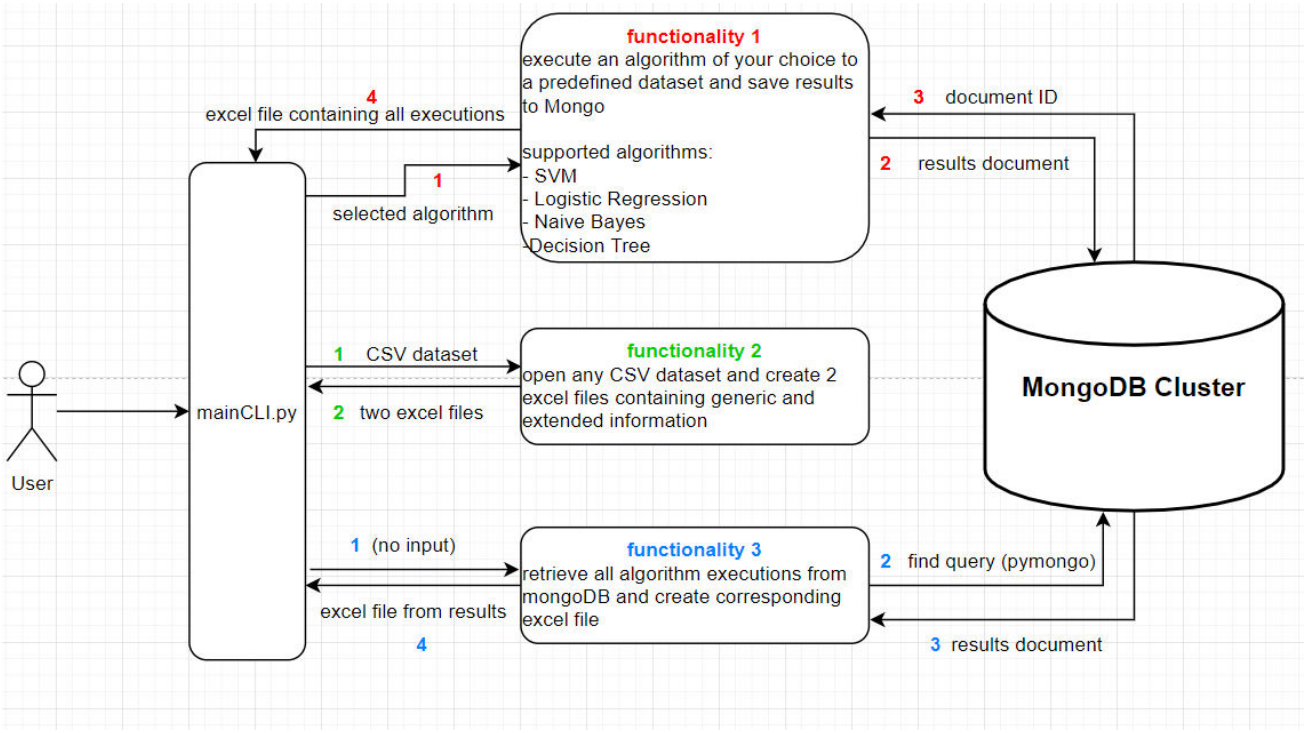
4.5 Δημιουργία και διαμόρφωση βάσης δεδομένων MongoDB

Για τις ανάγκες της εφαρμογής επιλέχθηκε η NoSQL βάση δεδομένων MongoDB, στην οποία θα αποθηκεύονται όλες οι πληροφορίες από τις εκτελέσεις των αλγορίθμων, μέσω διασύνδεσης της βάσης με την εφαρμογή. Αφού έχει προηγηθεί η αποθήκευση κάποιων δεδομένων, στη συνέχεια η ίδια βάση θα χρησιμοποιείται και για την ανάκτηση τους και αποθήκευση τους σε ένα αρχείο excel.

Η δημιουργία και διαμόρφωση της βάσης έγινε διαδικτυακά σε cloud παγκόσμιο πάροχο (MongoDB Atlas), ο οποίος έχει φτιαχτεί και συντηρείται από τους ίδιους ανθρώπους που εφηύραν και την MongoDB (docs.mongodb, 2021). Πρόκειται για σχετικά απλή διαδικασία η οποία απαιτεί βασικές γνώσεις database administration και εγγραφή στην πλατφόρμα MongoDB Atlas η οποία παρέχει δωρεάν κάποια MB χώρου σε προγραμματιστές για σκοπούς development. Επιγραμματικά τα βήματα που ακολουθήθηκαν ούτως ώστε να υπάρχει μια έτοιμη ολοκληρωμένη βάση:

1. Δημιουργία λογαριασμού και εγγραφή στην πλατφόρμα.
2. Δημιουργία βάσης δεδομένων με όνομα ‘algorithmsDb’ και δημιουργία αντίστοιχου collection σε αυτήν την βάση με όνομα ‘algorithmsCollection’. Τα collections είναι τα αντίστοιχα tables στον κόσμο της MongoDB.
3. Ρύθμιση του database cluster ώστε να επιτρέπει τη σύνδεση από οποιαδήποτε IP. Από προεπιλογή, για λόγους ασφαλείας, το cluster επέτρεπε την σύνδεση μόνο στην IP του υπολογιστή που είχε χρησιμοποιηθεί για την εγγραφή στην πλατφόρμα.
4. Αποθήκευση και χρήση στην command line εφαρμογή, του connection URL μέσω του οποίου η βάση επιτρέπει εισερχόμενες (inbound) συνδέσεις.

Για καλύτερη κατανόηση, στην παρακάτω εικόνα παρουσίαζεται ξεκάθαρα ο ρόλος που κατέχει η προαναφερθείσα βάση δεδομένων στην εφαρμογή.



Εικόνα 15 - Ολοκληρωμένος σχεδιασμός εφαρμογής

4.6 Υλοποίηση Command Line εφαρμογής

Η ανάπτυξη λογισμικού συνεχίζεται με την υλοποίηση ολοκληρωμένης Command Line εφαρμογής, η οποία δίνει στον χρήστη έναν εύχρηστο τρόπο να εκτελέσει κάποιες Data Science/Machine Learning λειτουργίες. Οι οδηγίες εκτέλεσης του προγράμματος περιγράφονται αναλυτικά στο παράρτημα. Σκοπός είναι η συγκεκριμένη εφαρμογή να συνεχίσει να επεκτείνεται και μετά το πέρας της παρούσας πτυχιακής, προσθέτοντας νέες λειτουργικότητες και βελτιώσεις. Στα πλαίσια αυτής της εργασίας ωστόσο, και όπως έχει αναφερθεί και σε προηγούμενες ενότητες, υλοποιήθηκαν τα παρακάτω:

1. Επιλογή και εκτέλεση ενός εκ των αλγορίθμων της ενότητας 4.3, στα προκαθορισμένα datasets που αναλύθηκαν στην ενότητα 4.1. Αποθήκευση των αποτελεσμάτων της εκτέλεσης σε μια online βάση δεδομένων MongoDB.
2. Άνοιγμα ενός CSV dataset επιλογής του χρήστη, και δημιουργία 2 πινάκων (σε αρχεία excel) τα οποία περιέχουν σημαντικές πληροφορίες για το dataset.
3. Ανάκτηση των εκτελέσεων που γίνονται στο βήμα 1, από την απομακρυσμένη βάση δεδομένων, και αποθήκευση των αποτελεσμάτων σε ένα αρχείο excel.

Το πρόγραμμα ξεκινάει ζητώντας από τον χρήστη να επιλέξει τη γλώσσα που προτιμά - οι γλώσσες που υποστηρίζονται είναι τα ελληνικά και τα αγγλικά. Στη συνέχεια εκτυπώνεται ένα εισαγωγικό σύντομο κείμενο το οποίο εξηγεί εν συντομία τα 3 διαφορετικά “μονοπάτια” που

μπορεί να ακολουθήσει ο χρήστης. Επιλέγοντας έναν εκ των αριθμών 1-3 το πρόγραμμα συνεχίζει σε μια από τις τρεις βασικές λειτουργίες του.

4.6.1 Περιγραφή λειτουργίας 1

Η πρώτη και σημαντικότερη λειτουργία της εφαρμογής είναι η εκτέλεση των αλγορίθμων και η αποθήκευση των αποτελεσμάτων online. Το πρώτο πράγμα που ζητείται από τον χρήστη είναι να καταχωρήσει το username του, το οποίο θα χρησιμοποιηθεί αργότερα στην ανάκτηση των αποτελεσμάτων (λειτουργία 3). Εφόσον δεν επιθυμεί να δώσει κάποιο όνομα, μπορεί απλά να πατήσει Enter και να χρησιμοποιηθεί το προκαθορισμένο username “defaultUser”.

Στη συνέχεια επιλέγει έναν εκ των αριθμών 1-4 για να τρέξει κάποιον αλγόριθμο, με την αντιστοιχία να είναι η εξής:

- 1 -> SVM
- 2 -> Logistic Regression
- 3 -> Naive Bayes
- 4 -> Decision Tree

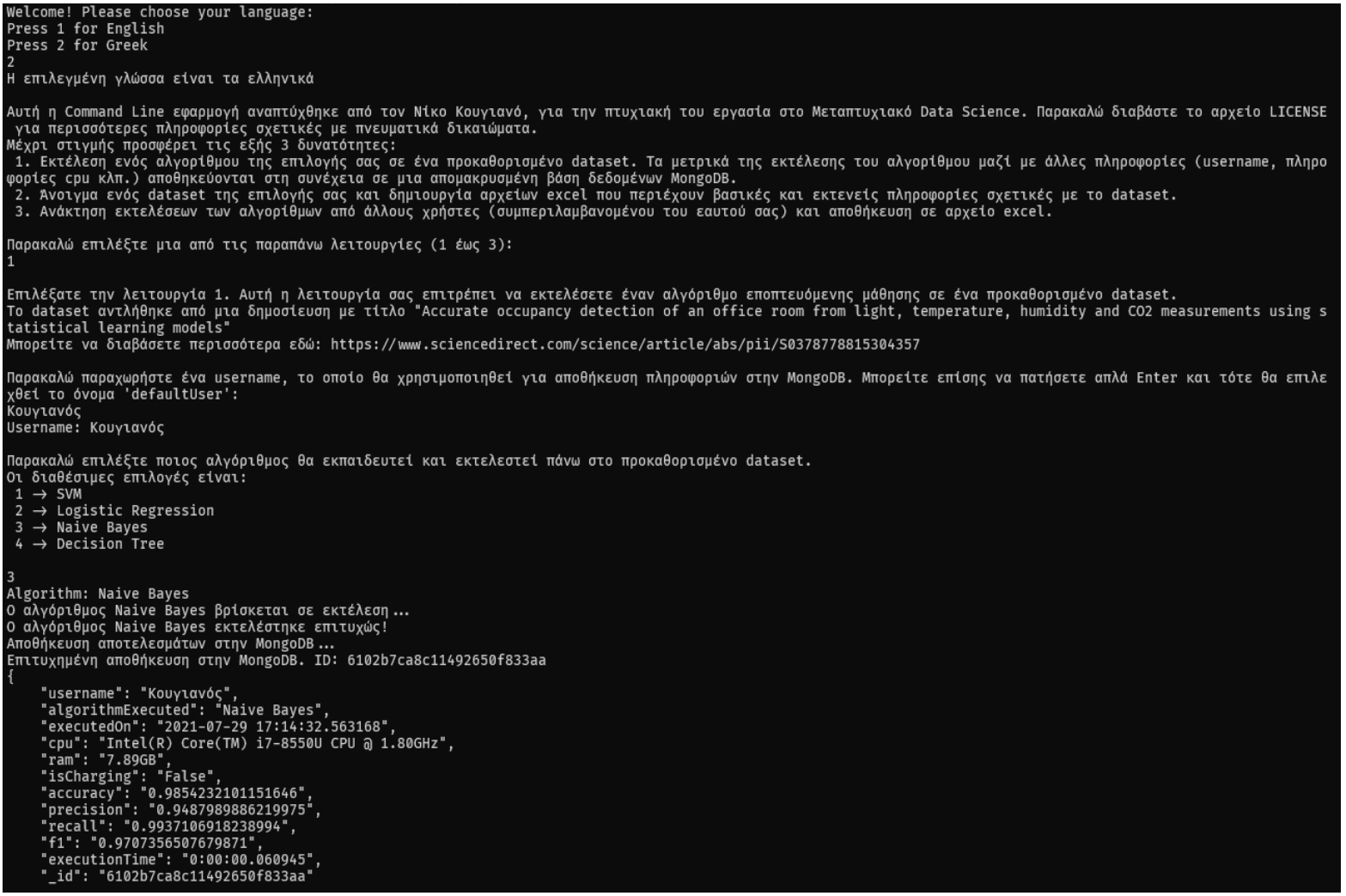
Αφού επιλεχθεί και ο αλγόριθμος, εν συνεχεία φορτώνονται τα δεδομένα στην εφαρμογή, ακολουθεί η εκπαίδευση και η εκτέλεση του αλγορίθμου, και τέλος αποθηκεύονται τα αποτελέσματα στη βάση δεδομένων. Στον παρακάτω πίνακα φαίνεται αναλυτικά η πληροφορία που αποθηκεύεται:

Attribute	**Example**	**Description**
Username	Kougianos	Το όνομα του χρήστη
algorithmExecuted	Logistic Regression	Το όνομα του αλγορίθμου που εκτελέστηκε
executedOn	2021-07-26 16:40:39	Η ακριβής ημερομηνία εκτέλεσης του πειράματος
cpu	Intel Core i7-8550U CPU @ 1.80GHz	Το όνομα και η συχνότητα του επεξεργαστή που χρησιμοποιήθηκε για την εκτέλεση του πειράματος
ram	7.89GB	Μέγεθος της μνήμης του υπολογιστή, σε GB
isCharging	True	Ένδειξη για το αν το λάπτοπ βρίσκεται σε ρεύμα κατά την ώρα της εκτέλεσης[29]

[29] Οι χρόνοι εκτέλεσης ενός προγράμματος επηρεάζονται άμεσα από το αν ένα laptop βρίσκεται στο ρεύμα ή όχι. Ο επεξεργαστής δουλεύει σε υψηλότερες συχνότητες όταν ο υπολογιστής φορτίζεται, με αποτέλεσμα τα προγράμματα να εκτελούνται πιο γρήγορα.

Accuracy	0.9825239	Δείκτης accuracy
precision	0.96485	Δείκτης precision
Recall	0.963257	Δείκτης recall
F1	0.96405	Δείκτης f1
executionTime	0:00:00.140	Συνολικός χρόνος εκπαίδευσης και εκτέλεσης του αλγορίθμου

Πίνακας 7 - Συνολική πληροφορία που αποθηκεύεται στη βάση δεδομένων



Εικόνα 16 - Ενδεικτική εκτέλεση λειτουργίας 1

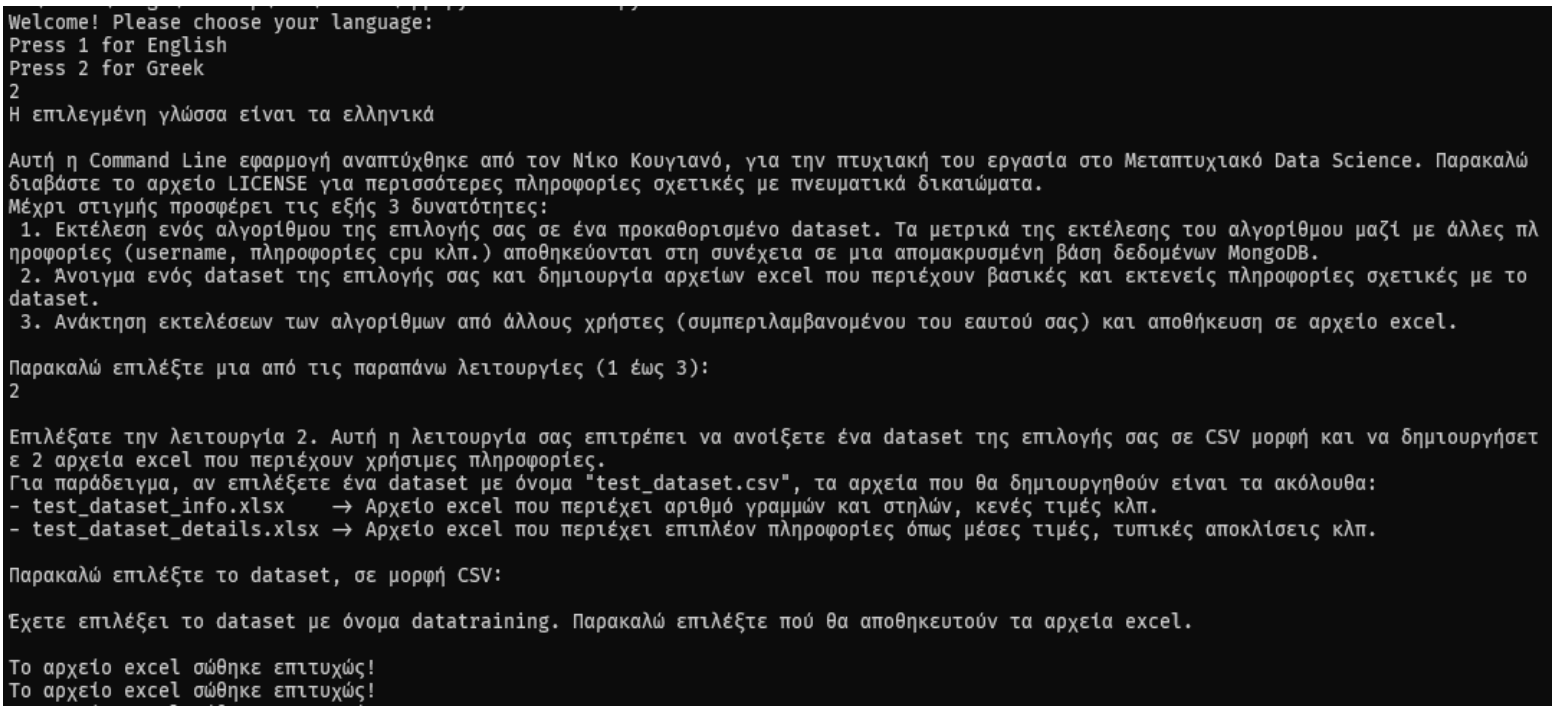
Ακολουθεί ενδεικτική εκτέλεση της εφαρμογής στα ελληνικά:

4.6.2 Περιγραφή λειτουργίας 2

Προχωράμε στην δεύτερη λειτουργία του προγράμματος, η οποία δρα ανεξάρτητα από τις άλλες δυο. Προσφέρει ουσιαστικά έναν εύχρηστο και γρήγορο τρόπο σε έναν χρήστη ο οποίος δεν έχει προγραμματιστικές γνώσεις, να δει χρήσιμες πληροφορίες για ένα σύνολο δεδομένων.

Έχοντας επιλέξει λοιπόν τον αριθμό 2, το πρόγραμμα ζητά από τον χρήστη να φορτώσει μέσα από γραφικό περιβάλλον (δηλαδή από παράθυρο περιήγησης) το dataset που τον ενδιαφέρει. Στη συνέχεια, αφού φορτωθεί το dataset, ανοίγει ένα δεύτερο παράθυρο στο οποίο ο

χρήστης επιλέγει την τοποθεσία που θα αποθηκευτούν τα 2 excels. Όπως αναφέρεται και στο κείμενο που συνοδεύει την εκτέλεση του προγράμματος, τα δημιουργημένα excel αρχεία θα έχουν το όνομα του dataset μαζί με τα λεκτικά _info και _details στο τέλος για καλύτερη κατανόηση. Ακολουθεί ενδεικτική εκτέλεση στα ελληνικά:



Εικόνα 17 - Ενδεικτική εκτέλεση λειτουργίας 2

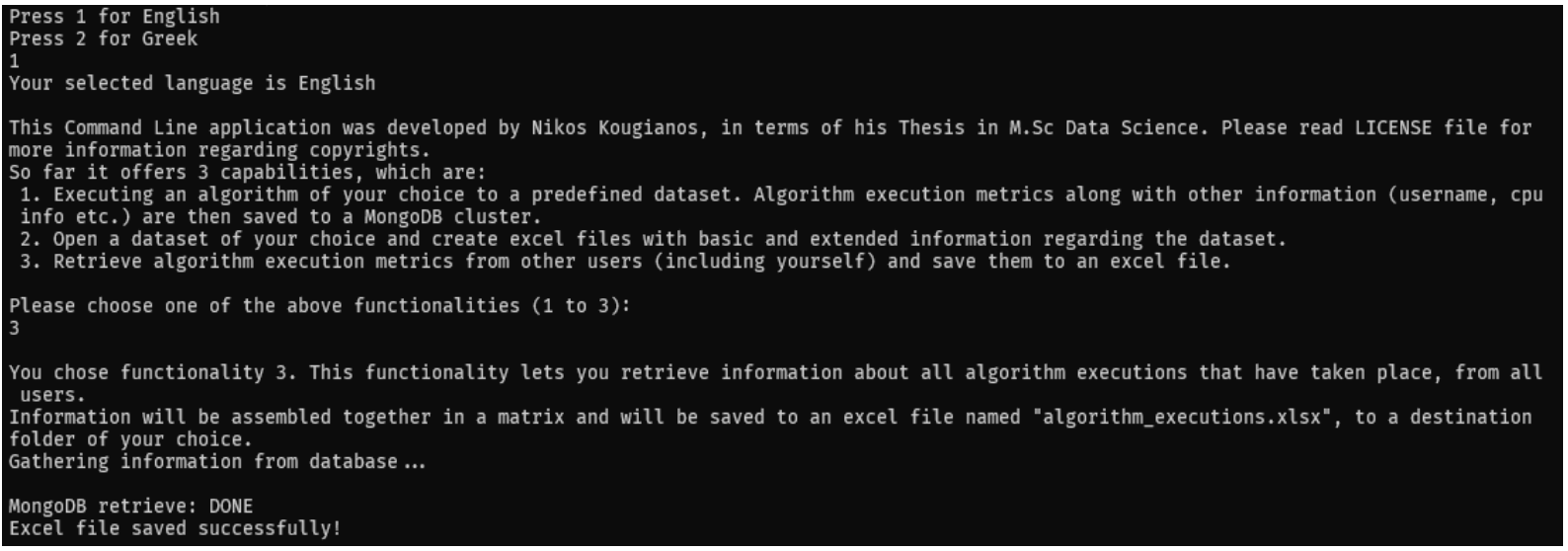
Τα παραγόμενα αρχεία excel στο παραπάνω παράδειγμα ονομάστηκαν datatraining_details.xlsx & datatraining_info.xlsx, και το περιεχόμενο της είναι παρόμοιο με τους πίνακες της ενότητας 4.1.

4.6.3 Περιγραφή λειτουργίας 3

Η τρίτη λειτουργία της εφαρμογής σχετίζεται άμεσα με την πρώτη, και αφορά την ανάκτηση των σχετικών εκτελέσεων που έγιναν από απομακρυσμένη βάση δεδομένων. Επιλέγοντας τον αριθμό 3 λοιπόν, εμφανίζεται ένα ενημερωτικό μήνυμα με λεπτομέρειες της συγκεκριμένης λειτουργικότητας, και στη συνέχεια ζητείται από τον χρήστη να επιλέξει την τοποθεσία όπου θα αποθηκευτεί το excel αρχείο ονόματι algorithm_executions.xlsx.

Στο προγραμματιστικό κομμάτι, αυτό που συμβαίνει είναι το εξής:

1. Σύνδεση στην MongoDB χρησιμοποιώντας τα ορθά credentials που έχουν φτιαχτεί στα πλαίσια της παρούσας πτυχιακής.
2. Σύνδεση στο σωστό MongoDB collection (αντίστοιχο του table σε μια SQL βάση δεδομένων)
3. Εκτέλεση ενός γενικευμένου find query πάνω στο collection το οποίο είναι αντίστοιχο του SELECT * from TABLE σε SQL γλώσσα.
4. Μετατροπή των αποτελεσμάτων σε Pandas Dataframe και αποθήκευση σε αρχείο excel.



Εικόνα 18 - Ενδεικτική εκτέλεση λειτουργίας 3

Ακολουθεί ενδεικτική εκτέλεση στα αγγλικά αυτή τη φορά:

Το αρχείο excel που αποθηκεύεται περιέχει έναν πίνακα της μορφής:

Να σημειωθεί πως η συγκεκριμένη λειτουργία μπορεί να προσδώσει μεγάλη αξία στην σύγκριση αποτελεσμάτων ανάμεσα σε διαφορετικά μηχανήματα, και να εξαχθούν χρήσιμα συμπεράσματα για τις επιδόσεις των αλγορίθμων σε σχέση με την επεξεργαστική ισχύ του εκάστοτε υπολογιστή. Όπως φαίνεται από τις κολώνες cpu & ram, όλες οι δοκιμαστικές εκτελέσεις έχουν γίνει από το λάπτοπ του συγγραφέα του οποίου οι προδιαγραφές αναφέρονται αναλυτικά στην ενότητα 3.2.

username	algorithmExecuted	executedOn	cpu	ram	isCharging	accuracy	precision	recall	f1	executionTime
kougianos	Logistic Regression	2021-07-26 10:51:44.26	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz	7.89GB	True	0,9825	0,9649	0,9633	0,9641	0:00:00.100968
kougi	Decision Tree	2021-07-26 11:02:03.73	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz	7.89GB	True	0,9512	0,9317	0,8626	0,8958	0:00:00.032919
test	Decision Tree	2021-07-26 11:03:00.98	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz	7.89GB	True	0,9311	0,8786	0,8315	0,8544	0:00:00.020077
no charge	Naive Bayes	2021-07-26 11:04:27.43	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz	7.89GB	False	0,9854	0,9488	0,9937	0,9707	0:00:00.028001
test1233	SVM	2021-07-26 11:06:24.32	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz	7.89GB	False	0,9896	0,9641	0,9944	0,9790	0:00:19.904515
kkk	Decision Tree	2021-07-26 11:09:50.01	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz	7.89GB	True	0,9397	0,9300	0,8133	0,8677	0:00:00.028000
1	Naive Bayes	2021-07-26 11:13:22.47	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz	7.89GB	True	0,9854	0,9488	0,9937	0,9707	0:00:00.035266
kougfi	Logistic Regression	2021-07-26 11:13:52.84	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz	7.89GB	True	0,9825	0,9649	0,9633	0,9641	0:00:00.072963
Νίκος Κουγιαν	Naive Bayes	2021-07-26 11:17:54.95	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz	7.89GB	True	0,9854	0,9488	0,9937	0,9707	0:00:00.013035
Νικόλαος Koug	SVM	2021-07-26 11:19:34.44	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz	7.89GB	True	0,9896	0,9641	0,9944	0,9790	0:00:18.600441
Νκακσδ	Logistic Regression	2021-07-26 11:23:28.57	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz	7.89GB	True	0,9825	0,9649	0,9633	0,9641	0:00:00.057947
kjldasjkdlas	Logistic Regression	2021-07-26 11:38:51.52	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz	7.89GB	False	0,9825	0,9649	0,9633	0,9641	0:00:00.124326
defaultUser	Logistic Regression	2021-07-26 16:40:39.68	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz	7.89GB	True	0,9825	0,9649	0,9633	0,9641	0:00:00.140507

Πίνακας 8 - Ενδεικτικός πίνακας αποτελεσμάτων

ΚΕΦΑΛΑΙΟ 5 - Συμπεράσματα και μελλοντικό έργο

5.1 Σύνοψη

Το πέμπτο και τελευταίο κεφάλαιο της εργασίας περιέχει συνοπτικά τα πράγματα που καλύφθηκαν, χρήσιμα συμπεράσματα από τις αναλύσεις που έγιναν καθώς επόμενα βήματα. Ξεκινώντας με μια ιστορική αναδρομή στον κόσμο του Machine Learning, και αναφέροντας εισαγωγικές πληροφορίες για το Data Science, ο αναγνώστης αποκτά μια γενική αλλά απαραίτητη γνώση πάνω στις δύο επιστήμες. Οι επόμενες ενότητες ωστόσο εμβαθύνουν, αναφέροντας περισσότερες λεπτομέρειες για τις κυριότερες κατηγορίες Machine Learning αλγορίθμων (Supervised, Unsupervised, Semi-Supervised) και τους τρόπους λειτουργίας τους. Συνεχίζοντας, παρουσιάστηκαν δημοφιλείς supervised και unsupervised αλγόριθμοι και καταγράφηκαν τα πλεονεκτήματα και μειονεκτήματα του καθένα, με σχετική αναφορά σε επιστημονικά άρθρα και δημοσιεύσεις. Τελειώνοντας το θεωρητικό σκέλος της πτυχιακής εργασίας, πραγματοποιήθηκε έρευνα και αναφέρθηκαν πρόσφατες (2019-2020) ακαδημαϊκές δημοσιεύσεις που σχετίζονται με τον χώρο του Data Science και του Machine Learning.

Στο δεύτερο σκέλος της εργασίας, πραγματοποιήθηκε υλοποίηση και ανάλυση σε πρακτικό επίπεδο, έχοντας ως δεδομένα εισόδου τρία προκαθορισμένα datasets με δεδομένα ενέργειας, που έχουν χρησιμοποιηθεί σε δημοσίευση του 2015. Ακολουθήσαμε το κλασικό μονοπάτι που θα ακολουθούσε και ένας data scientist στην εργασία του, ξεκινώντας με διερευνητική ανάλυση των δεδομένων και στη συνέχεια προχωρώντας με την προεπεξεργασία τους. Στο συγκεκριμένο βήμα να σημειωθεί ότι ήμασταν “τυχεροί” καθώς τα δεδομένα ήταν σε πολύ καλή μορφή, χωρίς ακραίες και κενές τιμές.

Στη συνέχεια, υλοποιήθηκαν αυτόνομα python scripts που περιέχουν την εκπαίδευση και εκτέλεση 4 supervised learning αλγορίθμων και ενός unsupervised (k-means) πάνω στα προαναφερθέντα datasets, καθώς και το αντίστοιχο jupyter notebook. Η περισσότερη προγραμματιστική δουλειά ωστόσο έγινε πάνω στην υλοποίηση Command Line εφαρμογής, η οποία μπορεί να εκτελεστεί από οποιονδήποτε χωρίς να απαιτεί γνώσεις Data Science και προγραμματισμού. Μέσω ενός εύχρηστου interface, ο χρήστης εκτελώντας 3-5 γρήγορες εντολές στο πρόγραμμα μπορεί να εκτελέσει και αυτός τους αλγορίθμους, να αποθηκεύσει τα αποτελέσματα σε βάση δεδομένων και να ανακτήσει πληροφορίες από εκτελέσεις τόσο δικές του

όσο και άλλων χρηστών. Δίνεται η δυνατότητα επίσης, να φορτώσει ένα οποιοδήποτε CSV dataset και να δει χρήσιμες πληροφορίες για αυτό σε αρχεία excel, βοηθώντας έτσι στο κομμάτι της διερευνητικής ανάλυσης.

5.2 Μελλοντικό έργο

Σαν επόμενα βήματα, να αναφερθεί ότι ο μόνος περιορισμός για την επέκταση και τη βελτίωση της CLI εφαρμογής είναι η φαντασία μας. Η εφαρμογή είναι ήδη διαθέσιμη[30] σε δημόσιο αποθετήριο (github) και μπορεί ο οποιοσδήποτε να κατεβάσει τον κώδικα, να δει την υλοποίηση και να πειραματιστεί. Ενδεικτικά επόμενα βήματα και βελτιώσεις θα μπορούσαν να είναι:

- Υποστήριξη ακόμη περισσότερων αλγορίθμων. Στα πλαίσια της πτυχιακής υποστηρίζονται 4 supervised learning αλγόριθμοι και 1 unsupervised.
- Βελτίωση λειτουργίας 2 ώστε να αναγνωρίζει όλα τα text based αρχεία και όχι μόνο CSV.
- Υλοποίηση αντίστοιχου Graphical User Interface που έχει τις ίδιες λειτουργίες με το Command Line Interface.
- Επέκταση λειτουργίας 1 ώστε να υποστηρίζει και άλλα (προκαθορισμένα και μη) datasets πάνω στα οποία μπορούν να εκτελεστούν οι αλγόριθμοι.
- Προσθαφαίρεση χαρακτηριστικών στα datasets, και εκτέλεση των ίδιων αλγορίθμων, για να αποδειχτεί (σε συνδυασμό και με το correlation matrix) κατά πόσο επηρεάζεται η εξαρτώμενη μεταβλητή Occupancy από τις υπόλοιπες.

[30] https://github.com/kougianos/ds-thesis.git

ΒΙΒΛΙΟΓΡΑΦΙΑ

Barone, A. (2020). *Investopedia*. Ανάκτηση από https://www.investopedia.com/terms/c/conditional_probability.asp

Breiman, Friedman, Olshen, & Stone. (1984). *Classification and Regression Trees.* Wadsworth Int. Group.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan†, J., Dhariwal, P., . . . Ramesh, A. (2020). *Language Models are Few-Shot Learners.*

Candanedo, L. M., & Feldheim, V. (2015, December). *researchgate*. Ανάκτηση από https://www.researchgate.net/publication/285627413_Accurate_occupancy_detection_of_an_office_room_from_light_temperature_humidity_and_CO2_measurements_using_statistical_learning_models

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). *End-to-End Object Detection with Transformers.*

Catanzarite, J. (2018, December). *towardsdatascience*. Ανάκτηση από https://towardsdatascience.com/the-naive-bayes-classifier-e92ea9f47523

Chauhan, N. S. (2019, December). *towardsdatascience*. Ανάκτηση από https://towardsdatascience.com/decision-tree-algorithm-explained-83beb6e78ef4

Chen, S.-Y., Su, W., Gao, L., Xia, S., & Fu, H. (2020). DeepFaceDrawing: Deep Generation of Face Images from Sketches.

Dhiraj, K. (2019, May). *Medium*. Ανάκτηση από https://medium.com/@dhiraj8899/top-5-advantages-and-disadvantages-of-decision-tree-algorithm-428ebd199d9a

docs.mongodb. (2021). Ανάκτηση από https://docs.mongodb.com/drivers/pymongo/

Foote, K. D. (2019, March). *dataversity*. Ανάκτηση από https://www.dataversity.net/a-brief-history-of-machine-learning/#

Frankenfield, J. (2020, March). Ανάκτηση από https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp

Friedman, J. H. (1991). *Multivariate adaptive regression splines.* Ανάκτηση από https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.382.970

Fumo, D. (2017, June). *towardsdatascience*. Ανάκτηση από https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861

Gandhi, R. (2018, May). *towardsdatascience*. Ανάκτηση από https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c

Google. (2020). Ανάκτηση από https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages

Greg, H., & Charles, E. (2002). Alternatives to the k-means algorithm that find better clustering.

Hayes, A. (2020). *Investopedia*. Ανάκτηση από https://www.investopedia.com/terms/b/bayes-theorem.asp

Hayes, A. (2021). *Investopedia*. Ανάκτηση από investopedia: https://www.investopedia.com/terms/c/cross-sell.asp

Hazimeh, H., Ponomareva, N., Mol, P., Tan, Z., & Mazumder, R. (2020). The Tree Ensemble Layer: Differentiability meets Conditional Computation.

Hochreiter, S., & Schmidhuber, J. (1997, December). *Long Short0term Memory. Neural computation*. Ανάκτηση από https://www.researchgate.net/publication/13853244_Long_Short-term_Memory

i2tutorials. (2019). *i2tutorials*. Ανάκτηση από https://www.i2tutorials.com/what-are-the-pros-and-cons-of-the-pca/

IBM. (2021). *ibm*. Ανάκτηση από https://www.ibm.com/cloud/learn/unsupervised-learning

Jaadi, Z. (2021). *builtin*. Ανάκτηση από https://builtin.com/data-science/step-step-explanation-principal-component-analysis

Joy, A. (2020). *Pros And Cons Of Supervised Machine Learning*. Ανάκτηση από pythonistaplanet: https://pythonistaplanet.com/pros-and-cons-of-supervised-machine-learning/

Kumar, P. (2020). *towardsdatascience*. Ανάκτηση από https://towardsdatascience.com/3-machine-learning-research-papers-you-should-read-in-2020-9b639bd0b8f0

Kumar, S. (2020). *towardsdatascience*. Ανάκτηση από https://towardsdatascience.com/hierarchical-clustering-agglomerative-and-divisive-explained-342e6b20d710

Lachaux, M.-A., Roziere, B., Chanussot, L., & Lample, G. (2020). *Unsupervised Translation of Programming Languages.*

Liu, Q., & Wu, Y. (2012, January). *Supervised Learning*. Ανάκτηση από https://www.researchgate.net/publication/229031588_Supervised_Learning

Marr, B. (2020). *forbes*. Ανάκτηση από https://www.forbes.com/sites/bernardmarr/2020/10/05/what-is-gpt-3-and-why-is-it-revolutionizing-artificial-intelligence/

McGonagle, J., Geoff, P., & Dobre, A. (2020). *brilliant*. Ανάκτηση από https://brilliant.org/wiki/gaussian-mixture-model/

Pulipaka, D. (2016). *Medium*. Ανάκτηση από https://medium.com/@gp_pulipaka/an-essential-guide-to-classification-and-regression-trees-in-r-language-4ced657d176b

Riggio, C. (2019, November). *towardsdatascience*. Ανάκτηση από https://towardsdatascience.com/whats-the-deal-with-accuracy-precision-recall-and-f1-f5d8b4db1021

SAS. (2020). *SAS*. Ανάκτηση από https://www.sas.com/en_us/insights/analytics/data-mining.html

Seetha, Murty, N., & Saravanan. (2011). On Improving the Generalization of SVM Classifier. Ανάκτηση από https://link.springer.com/chapter/10.1007/978-3-642-22786-8_2

stackoverflow. (2020). Ανάκτηση από https://insights.stackoverflow.com/survey/2020

Steinley, D., & Brusco, M. J. (2007). Initializing K-means batch clustering: A critical evaluation of several techniques.

Sujan, N. I. (2018, June). *Medium*. Ανάκτηση από Medium: https://medium.com/coinmonks/what-is-entropy-and-why-information-gain-is-matter-4e85d46d2f01

Teggi, P. (2020, February). *Medium*. Ανάκτηση από https://medium.com/@pralhad2481/chapter-3-decision-tree-learning-part-1-d0ca2365bb22

Thakur, N. K. (2020). *medium.com*. Ανάκτηση από https://medium.com/analytics-vidhya/comparison-of-initialization-strategies-for-k-means-d5ddd8b0350e

Thanda, A. (2020). *careerfoundry*. Ανάκτηση από https://careerfoundry.com/en/blog/data-analytics/what-is-logistic-regression/

Tripathi, M. (2019). *datascience.foundation*. Ανάκτηση από datascience.foundation: https://datascience.foundation/datatalk/machine-learning-algorithm

Valcheva, S. (2020). *intellspot*. Ανάκτηση από intellspot: http://www.intellspot.com/unsupervised-vs-supervised-learning/

Vashisht, R. (2020). *medium*. Ανάκτηση από medium: https://medium.com/atoti/how-to-solve-the-zero-frequency-problem-in-naive-bayes-cd001cabe211

Wikipedia. (2020). Ανάκτηση από https://en.wikipedia.org/wiki/Pattern_recognition

Wikipedia. (2020). Ανάκτηση από https://en.wikipedia.org/wiki/Use_case

Wikipedia. (2020). Ανάκτηση από https://en.wikipedia.org/wiki/Voronoi_diagram

Wikipedia. (2021). *Wikipedia*. Ανάκτηση από https://en.wikipedia.org/wiki/Vapnik%E2%80%93Chervonenkis_dimension

Xhemali, D., Hinde, C. J., & Stone, R. G. (2009). Naïve Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages . Leicestershire, United Kingdom.

ΠΑΡΑΡΤΗΜΑ

Αναλυτικές οδηγίες εκτέλεσης της Command Line εφαρμογής

Η εφαρμογή μπορεί να εκτελεστεί με 2 διαφορετικούς τρόπους και ο χρήστης είναι ελεύθερος να επιλέξει οποιονδήποτε επιθυμεί, ανάλογα με τις τεχνικές γνώσεις του και το επίπεδο που θέλει να εμβαθύνει. Για παράδειγμα, εάν κάποιος χρήστης είναι και προγραμματιστής και ενδιαφέρεται για τεχνικές λεπτομέρειες και πιθανότατα να θέλει να επεκτείνει ο ίδιος τον κώδικα, θα πρέπει να επιλέξει τον 1^{ο} τρόπο. Από την άλλη, εάν ενδιαφέρεται απλά για την εκτέλεση του προγράμματος και την παρακολούθηση των αποτελεσμάτων, μπορεί αν επιλέξει τον απλούστερο 2^{ο} τρόπο:

1^{ος} τρόπος (download source code)

Ο πρώτος τρόπος απαιτεί βασικές γνώσεις χρήσης ενός terminal και εγκατάστασης λογισμικού στον υπολογιστή. Ο χρήστης μπορεί να κατεβάσει τον πηγαίο κώδικα, να τον δει και να πειραματιστεί όπως επιθυμεί. Για περισσότερες πληροφορίες σχετικά με το πως γίνεται εγκατάσταση σε python & pip μπορείτε να ανατρέξετε εδώ
https://github.com/kougianos/ds-thesis/blob/master/README.md
Προαπαιτούμενο λογισμικό που χρειάζεται στον υπολογιστή:

- Python 3
- Pip
- Git (προαιρετικό)

Βήματα:

1. Εάν έχουμε το git εγκατεστημένο στον υπολογιστή μας (προτείνεται), ανοίγουμε ένα terminal και εκτελούμε την εντολή
   git clone https://github.com/kougianos/ds-thesis.git
2. Εάν δεν έχουμε το git εγκατεστημένο, κατεβάζουμε το πρόγραμμα σε zip αρχείο από το github πατώντας σε αυτό το link
   https://github.com/kougianos/ds-thesis/archive/refs/heads/master.zip
   και στη συνέχεια το κάνουμε unzip.
3. Πηγαίνουμε στο φάκελο app και εκτελούμε την εντολή
   pip install -r requirements.txt
   Η συγκεκριμένη εντολή χρησιμοποιώντας τον dependency manager της python (PIP) θα

εγκαταστήσει αυτόματα όλες τις βιβλιοθήκες που χρειάζονται για την εκτέλεση του προγράμματος.

4. Στον ίδιο φάκελο, εκτελούμε την εντολή

   python mainCLI.py

   για να τρέξει το πρόγραμμα. Πιθανώς να χρειαστεί αντί για python να χρησιμοποιηθεί το python3, αναλόγως του τι λειτουργικό σύστημα έχουμε.
5. Σε περίπτωση που το πρόγραμμα βγάλει κάποιο σφάλμα, πιθανώς να είναι λόγω μη εγκατεστημένης βιβλιοθήκης της python. Θα πρέπει με την εντολή

   pip install package_name

   να γίνουν install όλες οι απαραίτητες βιβλιοθήκες που δεν έγιναν στο βήμα 3.
6. Ακολουθούμε τις οδηγίες στο terminal και εκτελούμε το πρόγραμμα όσες φορές επιθυμούμε.

2ος τρόπος (download executable file)

Ο δεύτερος τρόπος είναι αρκετά πιο φιλικός προς το χρήστη καθώς δεν απαιτεί εγκατάσταση επιπλέον λογισμικού στον υπολογιστή μας. Να σημειωθεί ωστόσο ότι θα χρειαστεί το download ενός αρκετά μεγάλου αρχείου zip (περίπου 350MB) το οποίο περιέχει μέσα το εκτελέσιμο αρχείο του προγράμματος. Να σημειωθεί επίσης ότι το εκτελέσιμο είναι τύπου .exe, συνεπώς δε θα μπορεί να εκτελεστεί σε περιβάλλοντα Mac & Linux.

Βήματα:

1. Κατεβάζουμε το zip αρχείο από εδώ

   https://grizzledwizard.eu/docs/kougianos_thesis.zip

   και το κάνουμε unzip.
2. Ανοίγουμε τον φάκελο, και εντοπίζουμε το αρχείο **mainCLI.exe,** στο οποίο κάνουμε διπλό κλικ.
3. Εάν όλα έχουν πάει καλά, το πρόγραμμα θα ξεκινήσει. Ακολουθούμε τις οδηγίες στο terminal και εκτελούμε το πρόγραμμα όσες φορές επιθυμούμε.