ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

Τμήμα Ψηφιακών Συστημάτων

Π.Μ.Σ: *"Πληροφοριακά Συστήματα & Υπηρεσίες"*

Κατεύθυνση: *"Μεγάλα Δεδομένα και Αναλυτική"*

Μεταπτυχιακή Διατριβή

Τίτλος:

ΟΛΟΚΛΗΡΩΜΕΝΗ ΔΙΑΔΙΚΤΥΑΚΗ ΠΛΑΤΦΟΡΜΑ ΔΥΝΑΜΙΚΗΣ ΕΚΤΕΛΕΣΗΣ ΑΛΓΟΡΙΘΜΩΝ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ

από

ΒΑΣΙΛΑΚΗ ΣΤΑΥΡΟ

ΜΕ1805

Φεβρουάριος, 2021

Επιβλέπων Καθηγητής

ΚΥΡΙΑΖΗΣ ΔΗΜΟΣΘΕΝΗΣ

Εξεταστική Επιτροπή:

Καθηγητης

(Υπογραφή)

Καθηγητης

(Υπογραφή)

Καθηγητής

(Υπογραφή)

Η εργασία αυτή είναι αφιερωμένη στη γυναίκα μου και στους γονείς μου για την στήριξη που πάντα μου δείχνουν.

ΠΕΡΙΛΗΨΗ

Στον σύγχρονο κόσμο μεγάλοι όγκοι δεδομένων καθώς και ποικιλία αυτών παράγονται κάθε ώρα και στιγμή της ημέρας είτε μέσω τεχνολογιών είτε από ανθρώπινους παράγοντες. Αυτά τα δεδομένα παράγονται είτε σε πραγματικό χρόνο (real-time) είτε μαζικά (batches). Η μαζική επεξεργασία δεδομένων είναι εκείνη που πραγματοποιείται σε ένα τμήμα των δεδομένων που έχουν αποθηκευτεί για μια συγκεκριμένη χρονική περίοδο. Η επεξεργασία των πωλήσεων ενός μήνα μιας μεγάλης εμπορικής αλυσίδας είναι ένα παράδειγμα μαζικής επεξεργασίας δεδομένων, οι οποίες θα τεθούν υπό ανάλυση προς όφελος της αλυσίδας.
Ενώ ο αριθμός των τεχνολογιών αυξάνεται, η χρήση τους αυξάνεται, και κατ' επέκταση και τα δεδομένα που παράγουν. Ο πολύ μεγάλος όγκος δεδομένων χαρακτηρίζεται με το όρο '*big data*'. Τα μεγάλα αυτά δεδομένα μπορεί αν είναι δομημένα (structured) ή αδόμητα (unstructured).
Σε αυτή την εργασία θα δημιουργήσουμε μια διαδυκτιακής βάσης (web-based) πλατφόρμα για μηχανική μάθηση και επεξεργασία μεγάλων δεδομένων μαζικά.

Λέξεις-Κλειδιά: *Μεγάλα Δεδομένα, Big Data, επεξεργασία δεδομένων, data processing, εξόρυξη δεδομένων, Data mining, Μηχανική Μάθηση, Machine learning, Spark, Python, Pyspark*

ΠΕΡΙΕΧΟΜΕΝΑ

1. ΕΙΣΑΓΩΓΗ

1.1 ΜΕΓΑΛΑ ΔΕΔΟΜΕΝΑ

Δεδομένα (data) είναι οι ποσότητες, οι χαρακτήρες ή τα σύμβολα στα οποία εκτελούνται λειτουργίες από έναν υπολογιστή, οι οποίοι μπορούν να αποθηκευτούν και να μεταδοθούν με τη μορφή ηλεκτρικών σημάτων και να καταγραφούν σε μαγνητικά, οπτικά ή μηχανικά μέσα εγγραφής. Είναι χαρακτηριστικά ή πληροφορίες, συνήθως αριθμητικά, που συλλέγονται μέσω παρατήρησης. Με πιο τεχνική έννοια, τα δεδομένα είναι ένα σύνολο τιμών ποιοτικών ή ποσοτικών μεταβλητών για ένα ή περισσότερα άτομα ή αντικείμενα, ενώ ένα δεδομένο είναι μια μοναδική τιμή μιας μεμονωμένης μεταβλητής.

Υπάρχουν δύο κατηγορίες δεδομένων, τα ποσοτικά (quantitative) και τα ποιοτικά (qualitative). Τα ποσοτικά δεδομένα, ή αλλιώς και αριθμητικά, αφορούν αριθμούς και πράγματα που μπορούμε να μετρήσουμε αντικειμενικά: διαστάσεις όπως ύψος, πλάτος και μήκος. Άλλα παραδείγματα είναι η θερμοκρασία, η υγρασία και όγκος.
Τα ποσοτικά (quantitative) δεδομένα χωρίζονται σε δύο επιμέρους κατηγορίες τα συνεχή (continuous) και διακριτά (discrete). Ενώ τα ποιοτικά (qualitative) δεδομένα χωρίζονται σε τρεις επιμέρους κατηγορίες, τα δυαδικά (binomial), τα ονομαστικά (nominal) και τα κατηγορικά (ordinal) δεδομένα.

Τα διακριτά (discrete) δεδομένα είναι μια μέτρηση που δεν μπορεί διαιρεθεί περαιτέρω. Συνήθως περιλαμβάνει ακέραιους αριθμούς. Για παράδειγμα, ο αριθμός των παιδιών σε μια οικογένεια είναι διακριτά δεδομένα, επειδή μετράτε ολόκληρες, αδιαίρετες οντότητες: δεν γίνεται να υπάρχουν 2.5 παιδιά ή 1.3 κατοικίδια. Από την άλλη μεριά, τα συνεχή (continuous) δεδομένα θα μπορούσαν να διαιρεθούν σε περαιτέρω επίπεδα. Για παράδειγμα, η μέτρηση του ύψους ενός ανθρώπου μπορεί να γίνει σε ακριβείς κλίμακες - μέτρα, εκατοστά, χιλιοστά κ.α - έτσι το ύψος ανήκει στα συνεχή δεδομένα.

Τα δυαδικά (binomial) χωρίζουν τα δεδομένα σε δύο κατηγορίες που συνήθως είναι αρνητικό/θετικό, σωστό/λάθος και ψευδές/αληθές. Τα ονομαστικά (nominal) δεδομένα δίνουν μεμονωμένα διαφορετικά ονόματα τα οποία δεν μπορούν να καταταχθούν ή να ταξινομηθούν. Για παράδειγμα, το φύλλο ενός ανθρώπου αρσενικό/θηλυκό ή το χρώμα ματιών του ή ακόμα και ο ταχυδρομικός κώδικας της περιοχής στην οποία διαμένει είναι μερικές περιπτώσεις ονομαστικών δεδομένων. Τέλος, τα κατηγορικά δεδομένα (ordinal) τα οποία χωρίζονται σε κατηγορίες οι οποίες μπορούν να συγκριθούν μεταξύ τους. Για παράδειγμα, οι κατηγορίες ύψους όπως κοντός, μεσαίου ύψους και ψηλός ή οι βαθμολογίες από το 1 έως το 10 σε ένα διαγώνισμα.

Μεγάλα Δεδομένα (Big Data) είναι μια συλλογή δεδομένων με τεράστιο όγκο, που αυξάνεται εκθετικά με τον χρόνο. Είναι δεδομένα με τόσο μεγάλο μέγεθος και πολυπλοκότητα που κανένα από τα παραδοσιακά εργαλεία διαχείρισης δεδομένων δεν μπορεί να τα αποθηκεύσει ή να τα επεξεργαστεί αποτελεσματικά. Παραδείγματα όπως το Χρηματιστήριο της Νέας Υόρκης που παράγει περίπου ένα terabyte νέων εμπορικών δεδομένων ανά ημέρα. Επίσης, στατιστικές δείχνουν ότι 500 + terabyte νέων δεδομένων αποθηκεύονται στις βάσεις δεδομένων του ιστότοπου Facebook, κάθε μέρα. Αυτά τα δεδομένα παράγονται κυρίως από μεταφορφώσεις φωτογραφιών και βίντεο, ανταλλαγής μηνυμάτων, τοποθέτησης σχολίων κ.λπ. Τέλος, ένας κινητήρας Jet μπορεί να δημιουργήσει 10 + terabyte δεδομένων σε 30 λεπτά χρόνου πτήσης. Με πολλές χιλιάδες πτήσεις την ημέρα, η παραγωγή δεδομένων μπορεί να φτάσει έως και πολλά Petabytes.

Τύποι Μεγάλων Δεδομένων:

- *Δομημένα (Structured):* Οποιαδήποτε δεδομένα μπορούν να αποθηκευτούν, να γίνονται διαθέσιμα και να επεξεργαστούν με μορφή ενός σταθερού τύπου. Με την πάροδο του χρόνου, το ταλέντο στην επιστήμη των υπολογιστών πέτυχε μεγαλύτερη επιτυχία στην ανάπτυξη τεχνικών εργασίας με τέτοιου είδους δεδομένα (όπου η μορφή είναι γνωστή εκ των προτέρων) και επίσης αποκομίζει αξία από αυτήν. Ωστόσο, στις μέρες μας, προβλέπουμε προβλήματα όταν ένα μέγεθος τέτοιων δεδομένων αυξάνεται σε μεγάλο βαθμό, τα τυπικά μεγέθη είναι στην οργή πολλών zettabytes. Ένας πίνακας που περιέχει πληροφορίες υπαλλήλων αποτελεί παράδειγμα δομημένου τύπου.

- *Μη-Δομημένα (Unstructured):* Οποιαδήποτε δεδομένα με άγνωστη μορφή ή δομή. Εκτός από το τεράστιο μέγεθος, τα μη δομημένα δεδομένα θέτουν πολλές προκλήσεις όσον αφορά την επεξεργασία τους για την απόκτηση αξίας μέσα από αυτά. Ένα τυπικό παράδειγμα μη δομημένων δεδομένων είναι μια ετερογενής πηγή δεδομένων που περιέχει έναν συνδυασμό απλών αρχείων κειμένου, εικόνων, βίντεο κ.λπ. αυτά τα δεδομένα είναι σε ακατέργαστη μορφή ή σε μη δομημένη μορφή. Επίσης τα αποτελέσματα ενός Google search διατελούν παράδειγμα μη δομημένου τύπου.

- *Ημί-Δομιμένα (Semi-Structure):* Μπορούν να περιέχουν και τις δύο μορφές δεδομένων που προ-αναφέρθηκαν. Μπορούμε να τα δούμε αρχικά ως δομημένα αλλά στην πραγματικότητα δεν ορίζονται π.χ σε έναν πίνακα μιας σχεσιακής βάσης δεδομένων. Για παράδειγμα τα δεδομένα που δίνονται μέσα απο ένα αρχείο XML.

Τα Μεγάλα Δεδομένα μπορούν να περιγραφούν με τα παρακάτω χαρακτηριστικά:

- *Όγκος (Volume*): Η ποσότητα των παραγόμενων και αποθηκευμένων δεδομένων. Το μέγεθος των δεδομένων καθορίζει την αξία και τις πιθανές πληροφορίες, και αν μπορεί να θεωρηθεί μεγάλα δεδομένα ή όχι. Το μέγεθος των μεγάλων δεδομένων είναι συνήθως μεγαλύτερο από terabytes και petabytes.

- *Ποικιλία (Variety):* Η ποικιλία αναφέρεται σε ετερογενείς πηγές και στη φύση των δεδομένων, τόσο δομημένων όσο και μη δομημένων. Παλιότερα οι περισσότερες εφαρμογές χρησιμοποιούσαν spreadsheets και σχεσιακές βάσεις δεδομένων ως κύριες πηγές δεδομένων. Στις μέρες μας χρησιμοποιούνται δεδομένα όπως emails, videos, φωτογραφίες, PDF, μηχανές παρακολούθησης κ.α ως πηγές δεδομένων για τις διάφορες εφαρμογές. Έτσι η ποικιλία αυτή από δεδομένα θέτει ζητήματα όπως η εξόρυξη δεδομένων (data mining), αποθήκευση (storage) και ανάλυση αυτών (data analysis).

- *Ταχύτητα (Velocity):* Ο όρος ταχύτητα αναφέρεται στην ταχύτητα παραγωγής δεδομένων. Το πόσο γρήγορα τα δεδομένα δημιουργούνται και υποβάλλονται σε επεξεργασία για την ικανοποίηση των απαιτήσεων, καθορίζει το πραγματικό δυναμικό των δεδομένων. Παραδείγματα ταχύτητας ροής δεδομένων από πηγές όπως επιχειρηματικές διαδικασίες, αρχεία καταγραφής εφαρμογών, δίκτυα και ιστότοπους κοινωνικών μέσων, αισθητήρες, φορητές συσκευές κ.λπ. Η ροή δεδομένων είναι τεράστια και συνεχής.

- *Αξιοπιστία (Veracity):* Η αξιοπιστία σχετίζεται με την ποιότητα και την αξία των δεδομένων. Τα μεγάλα δεδομένα πρέπει αν είναι αξιόπιστα έτσι ώστε να υπάρχει αξία στην ανάλυσή τους. Αν δεν συμβαίνει αυτό υπάρχει η περίπτωση να επηρεάστεί αρνητικά η ακρίβεια στην ανάλυσή τους.

- *Αξία (Value)*: Το όφελος το οποίο μπορούμε να προσκομίσουμε από την επεξεργασία των δεδομένων. Για παράδειγμα, η αξία των δεδομένων μπορεί να αντιπροσωπεύει την κερδοφορία των πληροφοριών που ανακτώνται από την ανάλυση μεγάλων δεδομένων.

- *Μεταβλητότητα (Variability):* Η ασυνέπεια που μπορεί να αποδεικνύεται κατά καιρούς από τα δεδομένα, εμποδίζοντας έτσι τη διαδικασία της ικανότητας χειρισμού και διαχείρισης των δεδομένων αποτελεσματικά.

Η ικανότητα να επεξεργαζόμαστε μεγάλα δεδομένα φέρνει πολλαπλά οφέλη. Οι επιχειρήσεις μπορούν να χρησιμοποιουν τα δεδομένα για συμβουλευτικούς λόγους σε αποφάσεις που παίρνουν. Η πρόσβαση σε δεδομένα από κοινωνικά δίκτυα όπως Facebook και Twitter αλλά και σε μηχανές αναζήτησης επιτρέπει σε οργανισμούς και επιχειρήσεις να προσαρμόζουν τις επιχειρηματικές στρατηγικές τους. Επίσης, υπάρχει συμβολή και στην βελτίωση της εξυπηρέτησης πελατών. Τα παραδοσιακά συστήματα με τα οποία οι επιχειρήσεις έπαιρναν feedback αντικαθίστανται από νέα συστήματα σχεδιασμένα με τεχνολογίες μεγάλων δεδομένων. Σε αυτά τα καινούρια συστήματα αξιοποιούνται τεχνολογίες επεξεργασίας φυσικής γλώσσας (Natural Language Processing ή αλλιώς NLP) για την ανάγνωση και αξιολόγηση απαντήσεων των καταναλωτών. Τέλος, μπορεί να υπάρξει έγκαιρος εντοπισμός κινδύνου για προϊοντα ή υπηρεσίες εάν υπάρξει καθώς και αποτελεσματικότερη απόδοση.

1.2 ΚΑΘΟΡΙΣΜΟΣ ΠΡΟΒΛΗΜΑΤΟΣ

Λόγω του αυξανόμενου όγκου δεδομένων οι λύσεις που θα πρέπει να παρουσιασθούν είναι εκείνες της παράλληλης επεξεργασίας. Το Apache Spark ως analytics engine για μεγάλης κλίμακας δεδομένα λειτουργεί μέσω της παράλληλης επεξεργασίας.

Το Apache Spark παρέχει τη δυνατότητα προγραμματισμού, μέσω του API του, σε διάφορες γλώσσες όπως Scala, Java, Python κ.α όμως χρειάζεται συνδυασμός γνώσεων προγραμματισμού σε ένα intermediate επίπεδο τουλάχιστον και αλγορίθμων μηχανικής μάθησης.

1.3 ΣΚΟΠΟΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Σκοπός της διπλωματικής αυτής εργασίας μέσω της πλατφόρμας για Μεγάλα Δεδομένα που δημιουργήθηκε είναι να παρέχει σε χρήστες τη δυνατότητα να επεξεργαστούν (process) μεγάλο όγκο δεδομένων σε batches αλλά και να εφαρμόσουν αλγορίθμους μηχανικής μάθησης (Machine Learning) με ιδιαίτερη ευκολία μέσω του user interface.

Αφού ο χρήστης ανεβάσει (upload) τα δεδομένα του σε μορφή csv (comma separated values) μπορεί να πάρει πληροφορίες γι αυτά και να τα εξερευνήσει περαιτέρω.

1.4 ΑΝΑΦΟΡΑ ΣΕ ΥΠΑΡΧΟΥΣΕΣ ΔΟΥΛΕΙΕΣ

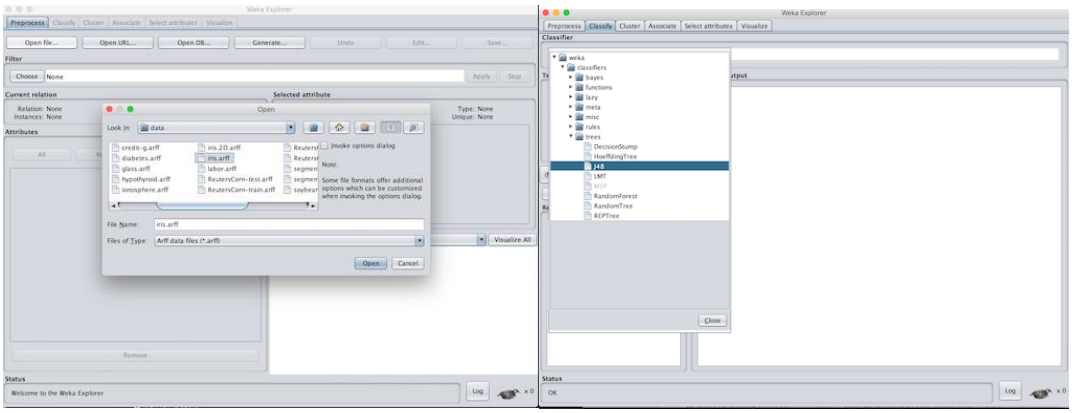
Παραδείγματα από παρόμοιες πλατφόρμες που υπάρχουν είναι οι WEKA και KNIME Analytics.
Το WEKA είναι ένα open source λογισμικό μηχανικής μάθησης στο οποίο υπάρχει πρόσβαση μέσω user interface, standard εφαρμογών τερματικού ή Java API. Χρησιμοποιείται ευρέως για τη διδασκαλία, την έρευνα και τις βιομηχανικές εφαρμογές, περιέχει μια πληθώρα ενσωματωμένων εργαλείων για standard εργασίες μηχανικής μάθησης και επιπλέον παρέχει διαφανή πρόσβαση σε γνωστές βιβλιοθήκες όπως scikit-learn, R και Deeplearning4j.

Το KNIME Analytics Platform είναι και αυτό open source λογισμικό για τη δημιουργία επιστήμης δεδομένων. Διαθέτει εύχρηστο user interface και συνεχώς εξελισσόμενο, το KNIME καθιστά την κατανόηση των δεδομένων και το σχεδιασμό ροών εργασίας της επιστήμης δεδομένων και των επαναχρησιμοποιήσιμων στοιχείων προσβάσιμα σε όλους.

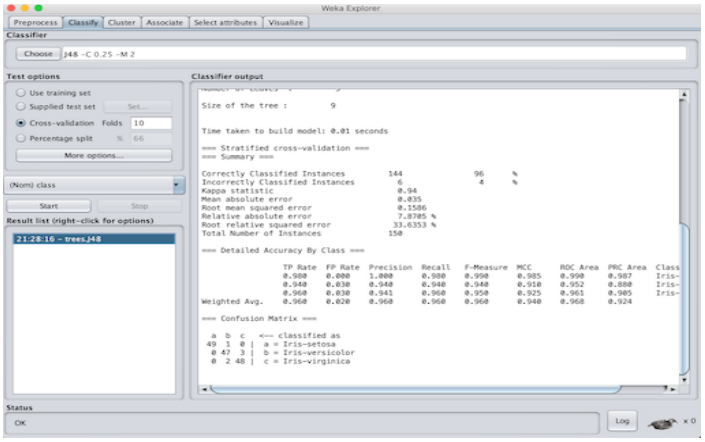
2. ΠΕΡΙΓΡΑΦΗ ΥΠΑΡΧΩΝ ΔΟΥΛΕΙΩΝ

2.1 WEKA

Το WEKA μπορεί να χρησιμοποιηθεί για τη δημιουργία μηχανικής μάθησης (machine learning) pipelines, για την εκπαίδευση classifiers, και να τρέξει αξιολογήσεις χωρίς να χρειαστεί η δημιουργία κώδικα.



Εικόνα 1 Επιλογή Dataset και Classifier



Εικόνα 2 Αποτελέσματα

Στο Figure 1 βλέπουμε ένα παράδειγμα εισαγωγής ενός συνόλου δεδομένων και επιλογής ενός classifier. Ενω στο Figure 2 βλέπουμε την αξιολόγηση της ακρίβειας μια πρόβλεψης. Επίσης παρέχεται η δυνατότητα Deep Learning μέσω του πακέτου WekaDeeplearing4j. Deep neural networks, συμπεριλαμβανομένου των convolutional networks και recurrent networks,

μπορουν να εκπαιδευτούν άμεσα από το interface του χρήστη, παρέχοντας μεθόδους για την ταξινόμηση (classification) εικόνων και κειμένου.

Το WEKA παρέχει τη δυνατότητα σύνδεσης και με το Apache Spark.

2.1.1 WEKA ΚΑΙ SPARK

Το Spark έχει πλεονεκτήματα σε σχέση με την πρώτη γενιά του map-reduce όταν πρόκειται για μηχανική εκμάθηση (machine learning). Έρχεται να αντικαταστήσει το Hadoop/YARN το οποίο ήταν το map-reduce εργαλείο για μεγάλα δεδομένα. Μέσω της αποθήκευσης των δεδομένων στη μνήμη για τον οποιοδήποτε υπολογισμό, βεβαιώνεται η αποτελεσματικότητα για επαναληπτικούς αλγορίθμους.

Εσωτερικά, η διανομήWekaSpark διαχειρίζεται αρχεία CSV μόνο προς το παρόν. Στο μέλλον θα εξεταστεί η υποστήριξη άλλων μορφών δεδομένων, όπως Avro και αρχεία ακολουθίας (sequence files). Ο χειρισμός CSV είναι ο ίδιος με αυτόν του WekaHadoop - λόγω του γεγονότος ότι τα δεδομένα προέλευσης χωρίζονται (split/partitioned) σε πολλούς κόμβους / εργαζόμενους (nodes/workers), δεν μπορεί να υπάρξει γραμμή που θα περιέχει τις επικεφαλίδες των στηλών και τα ονόματα χαρακτηριστικών μπορούν να παρέχονται μέσω ενός αρχείου "names" ή χειροκίνητα ως παράμετροι στις εργασίες. Τα δεδομένα CSV φορτώνονται και αναλύονται μόνο μία φορά, με αποτέλεσμα ένα

RDD <weka.core.Instance> με δομή κατανεμημένων (distributed) δεδομένων. Χάρη στο πλαίσιο επεξεργασίας του MapPartitions () του Apache Spark, η επεξεργασία προχωρά με τον ίδιο τρόπο όπως και στο DistWWekaHadoop, με την εξαίρεση ότι εξαλείφεται το γενικό κόστος της επανεξέτασης των δεδομένων σε κάθε επανάληψη κατά την εκτέλεση επαναληπτικών αλγορίθμων. Η επεξεργασία ενός ολόκληρου διαμερίσματος (partition) κάθε φορά αποτρέπει επίσης τη δημιουργία περιττών αντικειμένων χρησιμοποιώντας τις κλάσεις του διανεμημένου WekaBase.

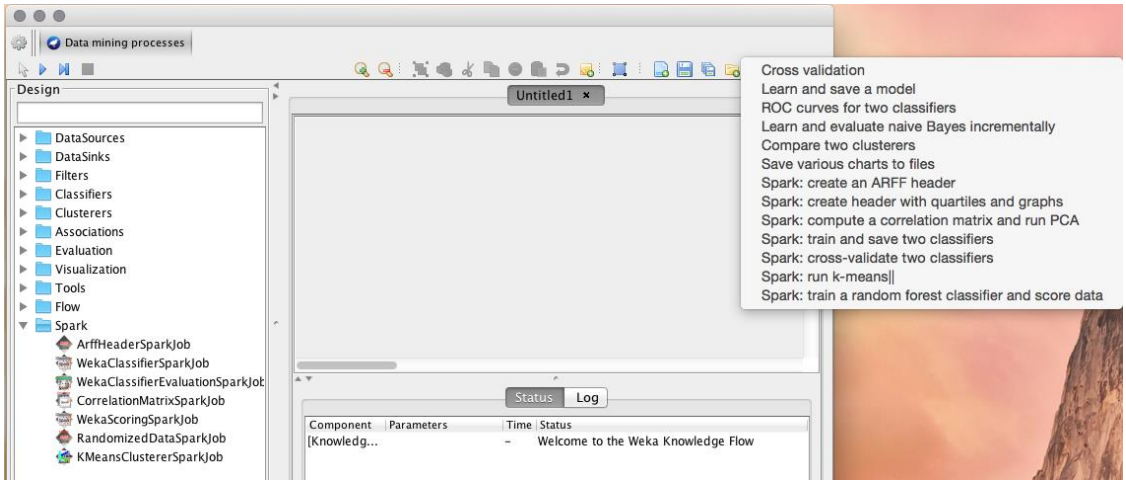
Οι λειτουργίες Reduce συζευγμένες και εναλλακτικές σε ζευγάρια στο Apache Spark και δεν υπάρχει άμεση αναλογία με το Hadoop Reduce, όπου ένας μονός reducer επαναλαμβάνει μια λίστα όλων των στοιχείων με ίδια key-value. Γι αυτό, για να αποφευχθούν πολλές ενέργειες reduce υλοποιούνται μέσω της ταξινόμησης (sorting) και αναδιαμερισμού (repartitioning), ακολουθούμενη από μία φάση διαμερισμού map. Στην πλειοψηφία των περιπτώσεων η προσέγγιση αυτή δουλεύει κανονικά, ενώ στις περιπτώσεις που η εργασία (job) τυχαία ανακατεύει (shuffles) και διαμορφώνει (stratifies) τα δεδομένα εισόδου (input data), το αποτέλεσμα είναι ελαφρά διαφορετικό από την υλοποίηση (implementation) του Hadoop. Η υλοποίηση του Apache Spark έχει ως αποτέλεσμα οι κλάσεις μειοψηφίας να είναι υπερμεγενθυμένες (oversampled).

Έχοντας τη δυνατότητα αναφοράς RDDs καθ’ όλη τη διάρκεια πού το Spark πλαίσιο (context) είναι ζωντανό, δίνεται η δυνατότητα στενότερης σύνδεσης μεταξύ των βημάτων των εργασιών

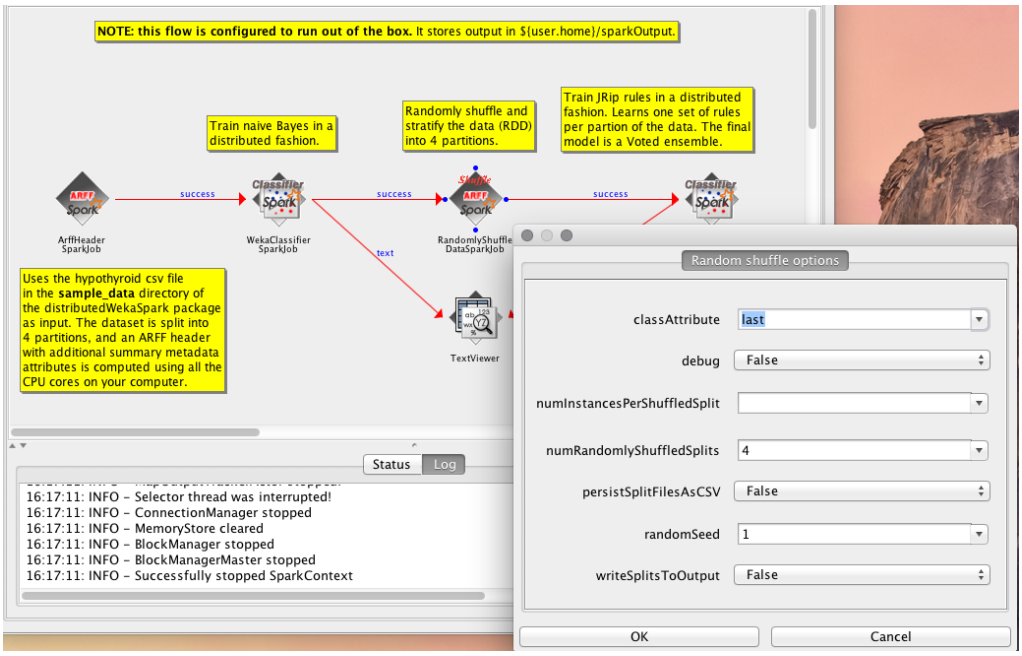
(job) Spark στη ροή της γνώσης (Knowledge flow). Οι τύποι αποτυχίας και επιτυχίας της σύνδεσης που εισάγωνται στο distributedWekaHadoop μπορούν να χρησιμοποιηθούν ως μεταφορείς δεδομένων, όπως το πλαίσιο (context) και αναφορές σε διάφορα σύνολα δεδομένων RDD που είναι σε λειτουργεία. Αυτό επιτρέπει στα βήματα του Spark να ξεκινούν από την αρχή της ροής και να παρουσιάσουν μια απλούστερη εκδοχή του UI για διαμορφώσεις (configurations), π.χ η απόκρυψη στοιχείων σύνδεσης αλλά και CSV.

Το πακέτο distributedWekaSpark συνοδεύεται από βασικές κλάσεις του Spark που είναι επαρκείς για να τρέχουν άμεσα στη λειτουργία local του Spark και να φορτώνουν δεδομένα από το τοπικό σύστημα αρχείων. Αυτή η λειτουργεία επωφελείται όλων των πυρήνων του υπολογιστή με το να ξεκινάει εργάτες (workers) σε διαφορετικά threads. Όλα τα συνδυασμένα παραδείγματα προτύπων για το Spark που διατίθενται από το μενού προτύπων της Γνωσιακής ροής χρησιμοποιούν αυτόν τον τρόπο λειτουργίας. Όλες οι ίδιες εργασίες που είναι διαθέσιμες στο distributedWekaHadoop είναι υλοποιημένες και στο distributedWekaSpark. Όπως το command line που υπάρχει στο Hadoop, το οποίο παρέχεται επίσης και στο Spark αντίστοιχα.

Όλες οι εργασίες (jobs) μπορούν να ως standalone πχ θα επικαλλούνται (invoke) άλλες εργασίες εσωτερικά εάν αυτό είναι απαραίτητο. Κατά τη διάρκεια εκτέλεσης της Γνώσης ροής (Knowledge flow) τα διαφορετικά βήματα εργασίας έχουν γνώση του πλαισίου του Spark και ποια σύνολα δεδομένων υπάρχουν ήδη στη μνήμη του cluster. Αυτό επιτρέπει στη διαμόρφωση των λεπτομερειών των συνδέσεων καθώς και στις επιλογές για διάβασμα αρχείων CSV, να ορίζονται μόνο μια φορά και κατ' επέκταση τα βήματα των εργασιών απλουστεύουν τα UI τους.



Εικόνα 3 Εξόρυξη Δεδομένων

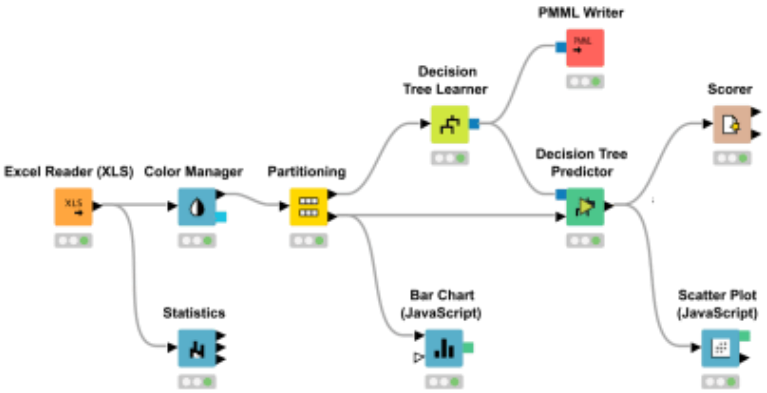


Εικόνα 4 Αρχιτεκτονική με Spark

Οι περιορισμοί που υπάρχουν είναι εκείνοι της επεξεργασίας αποκλειστικά CSV αρχείων και ότι μπορεί να υπάρχει μόνο ένα ενεργό πλαίσιο (context) Spark σε ένα JVM. Έτσι προκείπτουν περιορισμοί στη δομή των διενεργειών της Γνώσης ροής που χρησιμοποιούν βήματα Spark. Επιπλέον, όταν χρησιμοποιείται ένα cluster Spark με YARN ως διαχειρηστή, μόνο το “yarn-client” mode υποστηρίζεται. Τέλος, ένα άλλο ζήτημα που σχετίζεται με το YARN είναι ότι είναι απαραίτητο να έχετε το αρχείο διαμόρφωσης (configuration) του Hadoop στο CLASSPATH. Αυτό συμβαίνει επειδή ο Spark παίρνει τη διεύθυνση του διαχειριστή πόρων και άλλα κομμάτια από τα αρχεία διαμόρφωσης και η ρύθμιση αυτών των πληροφοριών δεν είναι δυνατή μέσω προγραμματισμού.

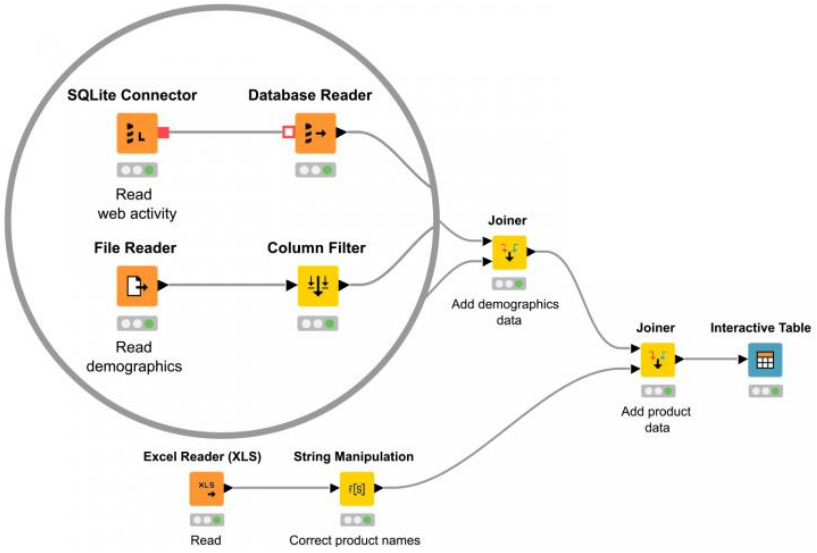
2.2 KNIME

Στην πλατφόρα του KNIME μπορείς να χτήσεις ροές εργασίας για την επιστήμη των δεδομένων. Αρχικά, δίνεται η δυνατότητα στον χρήστη να δημιουργεί οπτικές ροές εργασίας σε ένα διαμορφωμένο διαδραστικό UI για την καλύτερη κατανόηση των δεδομένων χωρίς την ανάγκη εγγραφής κώδικα. Επίσης, ο χρήστης δύναται τη μοντελοποίηση κάθε βήματος της ανάλυσής του καθώς και τον έλεγχο της ροής δεδομένων.



Εικόνα 5 Παράδειγμα ροής εργασίας

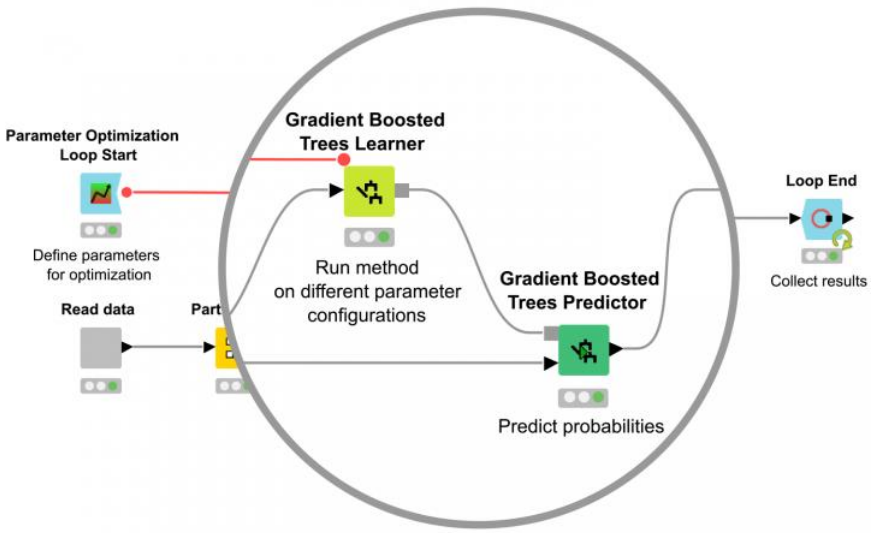
Ένα άλλο σημαντικό πλεονέκτημα είναι η δυνατότητα συνδυασμού δεδομένων από διάφορες πηγές όπως open source, και ο συνδυασμός απλών δεδομένων κειμένου (CSV, PDF, XLS, JSON, XML, κ.α), μη δομημένων τύπου δεδομένων (εικόνες, δίκτυα, κ.α) ή δεδομένα χρονοσειρών. Υπάρχει η δυνατότητα σύνδεσης σε βάσεις αλλά και αποθήκες δεδομένων για την ενσωμάτωση δεδομένων από Oracle, Microsoft SQL, Apache Hive κ.α. Πρόσβαση και ανάκτηση δεδομένων από πηγές όπως Salesforce, SharePoint, SAP Reader (Theobald), Twitter, AWS S3, Google Sheets, Azure και άλλα.



Εικόνα 6 Παράδειγμα ανάμειξης δεδομένων από διάφορες πηγές

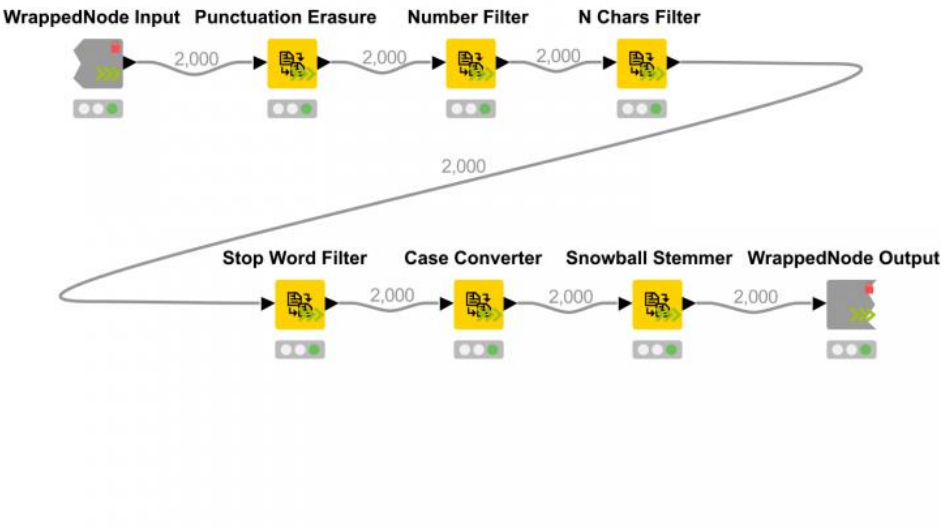
Στη συνέχεια, μπορεί να γίνει διαμόρφωση των δεδομένων, παραγωγή στατιστικών στοιχείων όπως μέσος, τυπική απόκλειση ή και εφαρμογή στατιστικών τεστ για την επικύρωση μιας υπόθεσης. Ακόμα, ενσωμάτωση για μείωση διαστάσεων, ανάλυση συσχέτισης κ.α. Επιπλέον ο χρήστης μπορεί να κανονικοποιήσει τα δεδομένα του, να τα μετατρέψει σε άλλο τύπο και να διαχειριστεί τιμές που λείπουν ή ακόμα και τιμές εκτός εύρους (outliers). Τέλος μπορεί να επιλέξει τις στήλες που θα χρησιμοποιήσει για κάποιον αλγόριθμο μηχανική μάθησης.

Ένα ακόμη βασικό χαρακτηριστικό του KNIME είναι η δημιουργία μοντέλων μηχανικής μάθησης για περιπτώσεις ταξινόμησης (classification), παλινδρόμησης (regression), μείωσης διαστάσεων αλλά και συσταδοποίησης (clustering). Μετ' έπειτα η βελτιστοποίηση της επίδοσης του εκάστοτε μοντέλου με βελτιστοποίηση υπερπαραμέτρων (hyperparameter optimization), ενίσχυση (boosting), συσσώρευση (bagging), στοίβαξη (stacking) ή κατασκευή σύνθετων συνόλων. Η επικύρωση με την εφαρμογή μετρικών επίδοσης όπως ακρίβεια R2, AUC και ROC. Και η πραγματοποίηση προβλέψεων χρησιμοποιώντας τα παραπάνω επικυρωμένα μοντέλα.



Εικόνα 7 Παράδειγμα Μηχανικής Μάθησης

Για την εκτέλεση μεγαλύτερης κλίμακας εργασιών ο χρήστης μπορεί να δημιουργήσει πρότυπα ροών εργασίας με αυξανόμενη αποτελεσματικότητα μέσω της αποθήκευσης στη μνήμη σε ζωντανή ροή καθώς και επεξεργασία με πολλαπλά threads. Ακόμα μπορεί να κάνει επεξεργασία μέσα στη βάση δεδομένων ή με παράλληλο προγραμματισμό μέσω του Apache Spark για περαιτέρω αύξηση απόδοσης του υπολογισμού.

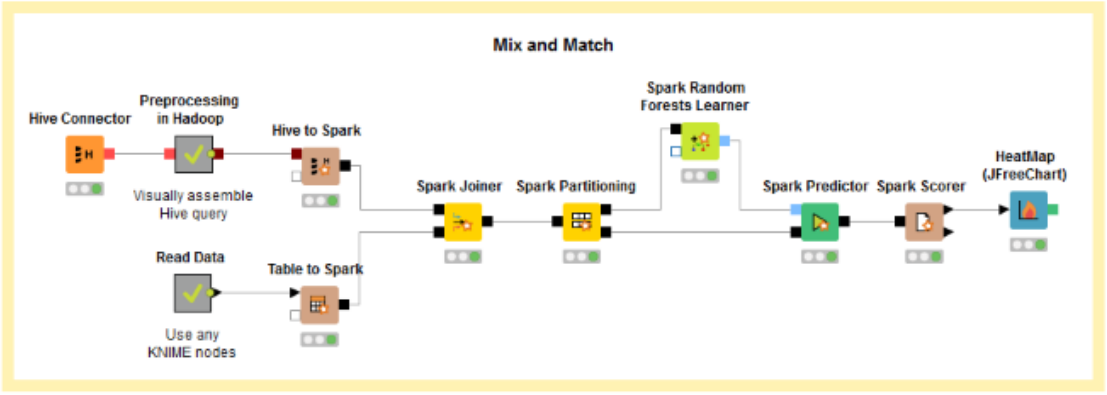


Εικόνα 8 Παράδειγμα κλιμακωτής ανάπτυξης βάσει ζήτησης

2.2.1 KNIME ΚΑΙ SPARK

Η επέκταση KNIME για Apache Spark είναι ένα σύνολο κόμβων (nodes) που χρησιμοποιούνται για τη δημιουργία και την εκτέλεση εφαρμογών Apache Spark με τη πλατφόρμα του KNIME Analytics. Αυτή η βιβλιοθήκη κόμβων δίνει τη δυνατότητα για:

- Ανάμειξη δεδομένων και εργαλείων.
- Εισαγωγή, εξαγωγή και πρόσβαση δεδομένων με Hive και HDFS.
- Αναλυτικές για προβλέψεις (predictive) και βαθμολόγισης (scoring) στο Apache Spark.
- Ενσωμάτωση εφαρμογών Java Apache Spark στις ροές εργασίας του KNIME.
- Ανακατάταξη και αντιστοίχιση τοπικών και Hadoop εκτελέσεων στην ίδια ροή εργασίας.



Εικόνα 9 Παράδειγμα εφαρμογής Spark μέσω του KNIME

Στην βιβλιοθήκη επίσης συμπεριλαμβάνονται κόμβοι που εφαρμόζουν συναρτήσεις στο Apache Spark:

- I/O
- Χειρισμού (Manipulation)
- Μηχανικής Μάθησης
- Στατιστικής
- Βαθμολογίας

Ενσωμάτωση της επεκτάσιμης βιβλιοθήκης μηχανικής εκμάθησης του Apache Spark στις ροές εργασίας σας για να την εκτέλεση:

- Ταξινόμησης (Decision Tree, Naïve Bayes, κ.α)
- Παλινδρόμησης (Logistic Regression, Linear Regression κ.α)
- Συσταδοποίησης (K-Means)
- Συνεργατικό φιλτράρισμα (ALS)
- Μείωσης διαστάσεων (SVD, PCA)

Οι κόμβοι προσφέρουν εύκολη χρήση για εξόρυξη, διαχείριση και εισαγωγή-εξαγωγή δεδομένων αλλά και ενσωμάτωση με την Apache Spark MLlib για περίπλοκες στατιστικές και ισχυρή μηχανική μάθηση στο Apache Spark μέσα από την πλατφόρμα του KNIME Analytics.

3. ΤΕΧΝΟΛΟΓΙΕΣ

Στο κεφάλαιο αυτό θα γίνει αναφορά στις τεχνολογίες που χρησιμοποιήθηκαν για τη δημιουργία της πλατφόρμας. Για το front-end έχουν χρησιμοποιηθεί οι τεχνολογίες JavaScript, CSS, jQuery, HTML και Bootstrap. Ενώ για το back-end έχει χρησιμοποιηθεί η γλώσσα προγραμματισμού Python και το framework Flask. Όσον αφορά το κομμάτι της επεξεργασίας δεδομένων και εφαρμογής αλγορίθμων μηχανικής μάθησης σε αυτά, γίνεται μέσω του Apache Spark. Στη συνέχεια περιγράφεται η κάθε τεχνολογία αναλυτικά.

3.1 SPARK

Το Apache Spark είναι ένα analytics engine για επεξεργασίας μεγάλης κλίμακας δεδομένα. Μερικά από τα βασικά του χαρακτηριστικά είναι:

- Ταχύτητα: Εκτέλεση φόρτου εργασίας έως και 100 φορές γρηγορότερα από άλλα engines. Το Apache Spark επιτυγχάνει υψηλή απόδοση τόσο για batches όσο και για streaming data, χρησιμοποιώντας έναν υπερσύγχρονο DAG scheduler, έναν query optimizer και μια μηχανή φυσικής εκτέλεσης.
- Ευκολία στη χρήση: Παρέχεται η δυνατότητα δημιουργίας εφαρμογών γρήγορα σε Java, Scala, Python, R και SQL.
- Γενικότητα: Συνδυασμός SQL, streaming, και περίπλοκες αναλυτικές. Υπάρχει ένα σύνολο από βιβλιοθήκες συμπεριλαμβανομένου SQL και DataFrames, MLlib για μηχανική μάθηση, GraphX και Spark streaming όπου γίνεται όλα αυτά να συνδυαστούν στην ίδια εφαρμογή.
- Τρέχει παντού: Το Spark τρέχει σε Hadoop, Apache Mesos, Kubernetes, standalone, ή και στο cloud και έχει πρόσβαση σε διάφορες πηγές δεδομένων.

Ξεκίνησε στο AMPLab στο U.C Berkeley το 2009 και έγινε γρήγορα ένα από τα βασικά εργαλεία για την διανεμημένη (distributed) επεξεργασία μεγάλων δεδομένων στον κόσμο. Μπορεί να αναπτυχθεί (deployed) με διάφορους τρόπους, παρέχει εγγενείς συνδέσεις με τις γλώσσες προγραμματισμού Java, Scala, Python και R και υποστηρίζει SQL, ροή δεδομένων (streaming data), μηχανική μάθηση (machine learning) και επεξεργασία γραφημάτων. Χρησιμοποιείται

από τράπεζες, εταιρείες τηλεπικοινωνιών, κυβερνήσεις, εταιρείες παιχνιδιών, και από κολοσσούς όπως Apple, Facebook, IBM και Microsoft.

Σε θεμελιώδες επίπεδο μια Apache Spark εφαρμογή αποτελείται από δύο βασικά στοιχεία: έναν driver, ο οποίος μετατρέπει τον κώδικα του χρήστη σε πολλαπλές εργασίες (tasks) που θα διανεμηθούν στους κόμβους εργατών (worker nodes) και τους executors, που τρέχουν σε αυτούς τους κόμβους και εκτελούν τις εργασίες που τους έχουν ανατεθεί. Κάποιου είδους διαχείρηση του συμπλέγματος (cluster) είναι απαραίτητη για την διαμεσολάβηση μεταξύ αυτών των δύο.

Απευθείας δίνεται η δυνατότητα το Apache Spark να τρέξει σε εκδοχή standalone συμπλέγματος (cluster) το οποίο απλά χρειάζεται το Apache Spark framework και JVM (Java virtual machine) σε κάθε μηχάνημα του συμπλέγματος. Ωστόσο μπορεί ο χρήστης να επωφεληθεί από ένα πιο ισχυρό σύστημα πόρων (robust system) ή κάποιο σύστημα διαχείρισης συμπλέγματος για να διευθετήσει το θέμα κατανομής των εργατών κατά ζήτηση (on demand). Τέτοια συστήματα είναι το Hadoop YARN (Yet Another Resource Negotiator), αλλά και Apache Mesos, Kubernetes και Docker Swarm. Επίσης υπάρχουν διαχειριζόμενες λύσεις (managed solutions) όπως Amazon EMR, Google Cloud Dataproc και Microsoft Azure HDInsight. Η Databricks, επίσης προσφέρει ολοκληρωμένη διαχειριζόμενη λύση, την Databricks Unified Analytics Platform, η οποίο προσφέρει Apache Spark συμπλέγματα, υποστήριξη ροής, ανάπτυξη σε web-based notebooks, και βελτιστοποιημένη απόδοση cloud I/O. Το Apache Spark ενσωματώνει τις εντολές επεξεργασίας δεδομένων του χρήστη σε κατευθυνόμενο (directed) Acyclic Graph ή DAG. Το DAG είναι το επίπεδο προγραμματισμού (scheduling) του Apache Spark, δηλαδή καθορίζει ποιες εργασίες εκτελούνται σε ποιους κόμβους και σε ποια ακολουθία (sequence).

Δύο μεγάλα πλεονεκτήματα είναι αυτά που κάνουν το Apache Spark διάδοχο του παλαιότερου Apache Hadoop στην επεξεργασία μεγάλων δεδομένων. Το πρώτο πλεονέκτημα είναι η ταχύτητα. Το Spark είναι στη-μνήμη (in-memory) μηχανή και αυτό σημαίνει ότι μπορεί να εκτελεί εργασίες ακόμα και 100 φορές γρηγορότερα από το MapReduce, ειδικά όταν συγκρίνεται με εργασία που έχουν πολλαπλά στάδια που χρειάζονται την εγγραφή της κατάστασης στον δίσκο μεταξύ των σταδίων. Το δεύτερο πλεονέκτημα έχει να κάνει με το ότι είναι πιο φιλικό για προγραμματιστές μέσω του Spark API κρύβοντας ένα μεγάλο μέρος της πολυπλοκότητας μιας κατανεμημένης επεξεργασίας μηχανή, πίσω από τις κλήσεις σε απλές μεθόδους σε γλώσσες είτε για ανάλυση δεδομένων με Python και R, είτε σε γλώσσες όπως Java και Scala για υποδομές.

Στην καρδιά του Apache Spark βρίσκεται η έννοια του Resilient Distributed Dataset (RDD), μια έννοια που εκπροσωπεί μια αμετάβλητη συλλογή από αντικείμενα που μπορούν να χωριστούν σε ένα σύμπλεγμα υπολογιστών. Η διεργασίες σε αυτά τα RDD μπορούν επίσης να χωριστούν στο σύμπλεγμα και να εκτελεστούν σε μερίδες (batches) παράλληλα, καταλήγοντας έτσι σε γρήγορη και κλιμακούμενη παράλληλη επεξεργασία. Τα RDD μπορούν να δημιουργηθούν από απλά αρχεία κειμένου, SQL βάσεις δεδομένων, NoSQL βάσεις δεδομένων (πχ MongoDB, Cassandra), Amazon S3 buckets, κ.α. Η πλειοψηφία του Spark Core API είναι χτησμένη πάνω στο

RDD, ενεργοποιώντας έτσι και την παραδοσιακή λειτουργία map-reduce, αλλά επίσης παρέχοντας βοήθεια εξ'αρχής για joining σύνολα δεδομένων, filtering, sampling και aggregation. Το Spark λειτουργεί με κατανεμημένο τρόπο συνδυάζοντας μια διαδικασία πυρήνα προγράμματος οδήγησης (driver core) που χωρίζει μια εφαρμογή Spark σε εργασίες (tasks) και τις διανέμει μεταξύ πολλών διαδικασιών εκτελεστών (executors) που κάνουν τη δουλειά. Αυτοί οι εκτελεστές μπορούν να κλιμακωθούν πάνω-κάτω, ανάλογα με τις ανάγκες της εφαρμογής.

Το Spark SQL αρχικά γνωστό και ως Shark, γίνεται ολοένα και σημαντικότερο στο Apache Spark. Επικεντρώνεται στην επεξεργασία δομημένων δεδομένων, προσεγγίζοντάς τα ως dataframes όπως και στην Python (Pandas). Επίσης παρέχει μια SQL2003-compliant διεπαφή για τη εφαρμογή query στα δεδομένα, φέρνοντας έτσι τη δυνατότητα του Apache Spark σε αναλυτές αλλά και προγραμματιστές. Μαζί με όλα αυτά υπάρχει δυνατότητα ανάγνωσης από πηγές καθώς και εγγραφής σε αυτές, όπως JSON, HDFS, Apache Hive, JDBC, Apache ORC και Apache Parquet. Άλλες διάσημες δομές αποθήκευσης όπως Apache Cassandra, MongoDB, Apache HBase και πολλές άλλες μπορούν να χρησιμοποιηθούν με ξεχωριστούς connectors από το οικοσύστημα των πακέτων Spark. Εσωτερικά το Spark χρησιμοποιεί ένα εργαλείο βελτιστοποίησης query που λέγεται Catalyst, το οποίο εξετάζει δεδομένα και queries έτσι ώστε να παράξει ένα αποδοτικό σχέδιο για την τοποθεσία των δεδομένων και τον υπολογισμό που θα συμβούν στο σύμπλεγμα. Παρά του ότι η διεπαφή RDD είναι διαθέσιμη προτείνεται η χρήση της Spark SQL διεπαφής. Το Spark 2.4 εισήγαγε ένα σύνολο από συναρτήσεις για τον διαχειρισμό arrays και διάφορων υψηλής τάξης τύπων δεδομένων απευθείας.

Το Apache Spark συνδυάζει ακόμα βιβλιοθήκες για την εφαρμογή αλγορίθμων μηχανικής μάθησης και τεχνικών ανάλυσης γραφημάτων σε μεγάλα δεδομένα. Το Spark MLlib περιλαμβάνει ένα framework για τη δημιουργία pipelines από αλγορίθμους μηχανικής μάθησης, επιτρέποντας την εύκολη διεξαγωγή χαρακτηριστικών (features), επιλογών και μετασχηματισμών σε οποιοδήποτε δομημένο σύνολο δεδομένων. Το MLlib υφίστανται με κατανεμημένες υλοποιήσεις με αλγορίθμους ταξινόμησης, παλινδρόμησης και συσταδοποίησης όπως k-means, linear regression καθώς και random forests που μπορούν να εναλλαχθούν μέσα και έξω από pipelines. Οι αλγόριθμοι ή μοντέλα μπορούν να αποθηκευτούν με τη χρήση του MLlib και αργότερα να ανακτηθούν και να εισαχθούν σε Java/Scala-based pipeline για την παραγωγή. Όμως, ακόμα δεν υπάρχει δυνατότητα εκπαίδευσης deep νευρωνικών δικτύων.

Το Spark GraphX συνοδεύεται από μια επιλογή κατανεμημένων αλγορίθμων για την επεξεργασία δομών γραφημάτων, συμπεριλαμβανομένης της εφαρμογής του PageRank της Google. Αυτοί οι αλγόριθμοι χρησιμοποιούν την προσέγγιση RDD του Spark Core για τη μοντελοποίηση δεδομένων. Το πακέτο GraphFrames επιτρέπει την λειτουργία γραφημάτων σε πλαίσια δεδομένων, συμπεριλαμβανομένης της αξιοποίησης του Catalyst optimizer για query γραφημάτων.

Το Spark Streaming ήταν μια προσθήκη στο Apache Spark που το βοήθησε να επεξεργάζεται σε πραγματικό χρόνο ή κοντά σε πραγματικό χρόνο. Παλιότερα οι διαδικασίες, επεξεργασία σε

παρτίδες και σε ροή, ήταν διαφορετικά πράγματα. Θα χρειαζόταν να γραφεί κώδικας ξεχωριστά για MapReduce επεξεργασία παρτίδας δεδομένων και για Apache Storm για ροή δεδομένων. Αυτό προφανώς οδηγεί σε διαφορετικές βάσεις κώδικα που πρέπει να διατηρούνται συγχρονισμένες για την εφαρμογή, παρά το γεγονός ότι βασίζονται σε εντελώς διαφορετικά πλαίσια, που απαιτούν διαφορετικούς πόρους και περιλαμβάνουν διαφορετικές λειτουργικές ανησυχίες για την εκτέλεση τους. Το Spark Streaming επέκτεινε την λειτουργία του Apache Spark μεταφράζοντας τη ροή σε συνεχείς μικρο-παρτίδες οι οποίες μέσω του Apache Spark API γίνονται διαχειρίσιμες. Έτσι η επεξεργασία σε ροή και σε παρτίδες μοιράζονται τον ίδιο κώδικα, και αποφεύγονται περιττές ενέργειες από μεριάς προγραμματιστή αλλά και χειριστή.

Το μέλλον της επεξεργασίας σε ζωντανή ροή είναι αυτό του Structured Streaming το οποίο έιναι ένα API σε ένα πιο αφηρημένο επίπεδο για τη δημιουργία εφαρμογών. Επιτρέπει στον προγραμματιστή να δημιουργήσει μεγάλο αριθμό Dataframes σε ροή και σύνολα δεδομένων. Επίσης, λύνει προβλήματα χρηστών όπως τα event-time aggregations και την αργή παράδοση μηνυμάτων. Όλα τα queries σε δομημένες ροές περνούν από τον βελτιστοποιητή query Catalyst, και μπορούν να τρέξουν ακόμα και με διαδραστικό τρόπο επιτρέποντας στον χρήστη την εφαρμογή SQL query σε δεδομένα ζωντανής ροής.

Εικόνα 10 Αρχιτεκτονική Spark

3.2 PYTHON

Η Python είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου (high-level) και γενικού σκοπού. Η φιλοσοφία σχεδιασμού της Python δίνει έμφαση στην αναγνωσιμότητα του κώδικα με την αξιοσημείωτη χρήση σημαντικού κενού χώρου. Οι γλωσσικές κατασκευές και η αντικειμενοστρεφής (object oriented programming) προσέγγιση στοχεύουν να βοηθήσουν τους προγραμματιστές να γράψουν σαφή, λογικό κώδικα για μικρά και μεγάλα έργα.

Βασικά πλεονεκτήματα της Python:

- Η Python ερμηνεύεται (interpreted) - υποβάλλεται σε επεξεργασία κατά το χρόνο εκτέλεσης από τον διερμηνέα (interpreter). Δεν χρειάζεται να μεταγλωττιστεί (compile) το πρόγραμμά πριν το εκτελεστεί. Αυτό είναι παρόμοιο με τη γλώσσα προγραμμτισμού PERL και PHP.
- Η Python είναι Διαδραστική - υπάρχει η δυνατότητα αλληλεπίδρασης απευθείας με τον διερμηνέα για να γραφτεί κάποιο πρόγραμμα.
- Η Python είναι Object-Oriented - η Python υποστηρίζει αντικειμενοστρεφή στυλ ή τεχνική προγραμματισμού που ενσωματώνει κώδικα μέσα σε αντικείμενα.
- Η Python είναι μια γλώσσα για αρχάριους - η Python είναι μια εξαιρετική γλώσσα για τους προγραμματιστές αρχάριου επιπέδου και υποστηρίζει την ανάπτυξη ενός ευρέος φάσματος εφαρμογών, από απλή επεξεργασία κειμένου έως προγράμματα περιήγησης στο διαδύκτιο έως παιχνίδια.

Σημαντικά χαρακτηριστικά του προγραμματισμού με Python:

- Υποστηρίζει λειτουργικές και δομημένες μεθόδους προγραμματισμού, καθώς και OOP.
- Μπορεί να χρησιμοποιηθεί ως γλώσσα scripting ή μπορεί να μεταγλωττιστεί σε byte-code για την κατασκευή μεγάλων εφαρμογών.
- Παρέχει δυναμικούς τύπους δεδομένων υψηλού επιπέδου και υποστηρίζει δυναμικό έλεγχο τύπων.
- Υποστηρίζει αυτόματη συλλογή απορριμμάτων.
- Μπορεί εύκολα να ενσωματωθεί με C, C ++, COM, ActiveX, CORBA και Java.

3.2.1 FLASK

Το Flask είναι ένα micro web framework γραμμένο σε Python. Κατατάσσεται ως framework επειδή δεν απαιτεί συγκεκριμένα εργαλεία ή βιβλιοθήκες. Δεν έχει επίπεδο αφαίρεσης βάσης δεδομένων, επικύρωση φόρμας (form validation) ή άλλα στοιχεία όπου προϋπάρχουσες βιβλιοθήκες τρίτων παρέχουν κοινές λειτουργίες (functions). Ωστόσο, το Flask υποστηρίζει επεκτάσεις που μπορούν να προσθέσουν λειτουργίες εφαρμογών σαν να εφαρμόστηκαν στο ίδιο το Flask. Υπάρχουν επεκτάσεις για object-relational maps, επικύρωση φόρμας, χειρισμός μεταφορτώσεων (uploading), διάφορες ανοιχτές τεχνολογίες ελέγχου ταυτότητας (authentication) και διάφορα κοινά εργαλεία που σχετίζονται με το framework. Οι εφαρμογές που χρησιμοποιούν το framework Flask περιλαμβάνουν το Pinterest και το LinkedIn.

Το Flask είναι βασισμένο στα έργα Pocoo, Werkzeug και Jinja2. Το Werkzeug είναι μια βιβλιοθήκη της Python για εφαρμογές Web Server Gateway Interface (WSGI), και έχει άδεια χρήσης με άδεια BSD. Το Werkzeug μπορεί να αναγνωρίσει αντικείμενα λογισμικού για αιτήσεις (software requests), απαντήσεις (response) και λειτουργίες χρησιμότητας. Το Jinja όπως φαίνεται παρακάτω.

3.2.2 JINJA

Το Jinja είναι μια μηχανή προτύπου (template engine) Ιστού για τη γλώσσα προγραμματισμού Python. Δημιουργήθηκε από τον Armin Ronacher και έχει άδεια βάσει BSD License. Το Jinja είναι παρόμοιο με τη μηχανή προτύπου Django, αλλά παρέχει εκφράσεις τύπου Python διασφαλίζοντας παράλληλα ότι τα πρότυπα αξιολογούνται σε ένα sandbox. Είναι μια γλώσσα προτύπου που βασίζεται σε κείμενο και έτσι μπορεί να χρησιμοποιηθεί για τη δημιουργία οποιασδήποτε σήμανσης καθώς και πηγαίου κώδικα.

Η μηχανή προτύπου Jinja επιτρέπει την προσαρμογή ετικετών, φίλτρων, δοκιμών και globals. Επίσης, σε αντίθεση με τη μηχανή προτύπων Django, η Jinja επιτρέπει στον σχεδιαστή προτύπων να καλεί συναρτήσεις με επιχειρήματα σε αντικείμενα. Το Jinja είναι η προεπιλεγμένη μηχανή προτύπου του Flask και χρησιμοποιείται επίσης από Ansible και Trac.

3.3 HTML

Το Hypertext Markup Language (HTML) είναι η τυπική γλώσσα σήμανσης για έγγραφα που έχουν σχεδιαστεί για προβολή σε πρόγραμμα περιήγησης στο Web. Μπορεί να υποβοηθηθεί από τεχνολογίες όπως Cascading Style Sheets (CSS) και γλώσσες scripting όπως το JavaScript. Τα προγράμματα περιήγησης στο Web λαμβάνουν έγγραφα HTML από διακομιστή ιστού ή από τοπικό χώρο αποθήκευσης και αποδίδουν τα έγγραφα σε ιστοσελίδες πολυμέσων. Η HTML περιγράφει τη δομή μιας ιστοσελίδας σημασιολογικά και αρχικά περιελάμβανε στοιχεία για την εμφάνιση του εγγράφου.

Τα στοιχεία HTML είναι τα δομικά στοιχεία των σελίδων HTML. Με δομές HTML, εικόνες και άλλα αντικείμενα όπως διαδραστικές φόρμες μπορεί να ενσωματωθούν στη σελίδα που αποδίδεται. Η HTML παρέχει ένα μέσο για τη δημιουργία δομημένων εγγράφων δηλώνοντας τη δομική σημασιολογία για κείμενο όπως επικεφαλίδες, παραγράφους, λίστες, συνδέσμους, εισαγωγικά και άλλα στοιχεία. Η HTML μπορεί να ενσωματώσει προγράμματα γραμμένα σε scripting γλώσσα ενεργειών όπως το JavaScript, το οποίο επηρεάζει τη συμπεριφορά και το περιεχόμενο των ιστοσελίδων. Η συμπερίληψη του CSS καθορίζει την εμφάνιση και τη διάταξη του περιεχομένου. Η Παγκόσμια Κοινοπραξία Ιστού (W3C), πρώην συντηρητής του HTML και τρέχων συντηρητής των προτύπων CSS, ενθάρρυνε τη χρήση του CSS σε συνδυασμό με την HTML σε σχέση με αποκλειστικά HTML.

3.3.1 BOOTSTRAP

Το Bootstrap είναι ένα πλαίσιο ιστού που εστιάζει στην απλοποίηση της ανάπτυξης ενημερωτικών ιστοσελίδων (σε αντίθεση με τις εφαρμογές ιστού). Ο πρωταρχικός σκοπός της προσθήκης σε ένα έργο Ιστού είναι η εφαρμογή των επιλογών χρώματος, μεγέθους, γραμματοσειράς και διάταξης του Bootstrap σε αυτό το έργο. Μόλις προστεθεί, το Bootstrap παρέχει βασικούς ορισμούς στυλ για όλα τα στοιχεία HTML. Το αποτέλεσμα είναι μια ομοιόμορφη εμφάνιση για χαρακτήρες, πίνακες και στοιχεία φόρμας σε προγράμματα περιήγησης ιστού. Επιπλέον, οι προγραμματιστές μπορούν να επωφεληθούν από τις κατηγορίες

CSS που ορίζονται στο Bootstrap για να προσαρμόσουν περαιτέρω την εμφάνιση του περιεχομένου τους. Για παράδειγμα, το Bootstrap έχει προβλέψει πίνακες ανοιχτού και σκούρου χρώματος, επικεφαλίδες σελίδων, πιο εμφανή εισαγωγικά και κείμενο με επισήμανση.

Το Bootstrap έρχεται επίσης με πολλά στοιχεία JavaScript με τη μορφή προσθηκών jQuery. Παρέχουν πρόσθετα στοιχεία διεπαφής χρήστη, όπως παράθυρα διαλόγου, επεξηγήσεις εργαλείων κ.α. Κάθε στοιχείο Bootstrap αποτελείται από δομή HTML, δηλώσεις CSS και σε ορισμένες περιπτώσεις που συνοδεύουν κώδικα JavaScript. Επεκτείνουν επίσης τη λειτουργικότητα ορισμένων υπαρχόντων στοιχείων διεπαφής, όπως για παράδειγμα μια αυτόματη συμπλήρωση λειτουργίας για πεδία εισαγωγής.

3.4 JAVASCRIPT

Η JavaScript η οποία συντομογραφείται συχνά ως JS, είναι μια γλώσσα προγραμματισμού που συμμορφώνεται με τις προδιαγραφές ECMAScript. Η JavaScript είναι γλώσσα υψηλού επιπέδου (high-level), γίνεται compiled πριν τρέξει και πολλαπλών παραδειγμάτων (multi-paradigm) . Έχει σύνταξη που αποτελείται από curly-brackets, δυναμική πληκτρολόγηση, και αντικειμενοστρεφής (object-oriented). Παράλληλα με την HTML και το CSS, η JavaScript είναι μία από τις βασικές τεχνολογίες του World Wide Web. Η JavaScript επιτρέπει διαδραστικές ιστοσελίδες και αποτελεί ουσιαστικό μέρος των εφαρμογών ιστού. Η συντριπτική πλειονότητα των ιστότοπων το χρησιμοποιούν για συμπεριφορά σελίδας από τον client, και όλα τα μεγάλα προγράμματα περιήγησης στο Web διαθέτουν έναν ειδικό μηχανισμό JavaScript για να το εκτελέσουν.

Ως γλώσσα πολλαπλών παραδειγμάτων, το JavaScript υποστηρίζει event-driven και λειτουργικό προγραμματισμό. Διαθέτει application programming interfaces (APIs) για εργασία με κείμενο, ημερομηνίες, κανονικές εκφράσεις, τυπικές δομές δεδομένων και το μοντέλο αντικειμένου εγγράφου (DOM).Το πρότυπο ECMAScript δεν περιλαμβάνει είσοδο / έξοδο (I / O), όπως εγκαταστάσεις δικτύωσης, αποθήκευσης ή γραφικών. Στην πράξη, το πρόγραμμα περιήγησης ιστού ή άλλο σύστημα χρόνου εκτέλεσης παρέχει API JavaScript για I / O.

Οι μηχανές JavaScript χρησιμοποιήθηκαν αρχικά μόνο σε προγράμματα περιήγησης ιστού, αλλά αποτελούν πλέον βασικά στοιχεία άλλων συστημάτων χρόνου εκτέλεσης, όπως το Node.js και το Deno. Αυτά τα συστήματα χρησιμοποιούνται για την κατασκευή διακομιστών και είναι επίσης ενσωματωμένα σε πλαίσια, όπως το Electron και το Cordova, για τη δημιουργία μιας ποικιλίας εφαρμογών. Παρόλο που υπάρχουν ομοιότητες μεταξύ της JavaScript και της Java, συμπεριλαμβανομένου του ονόματος γλώσσας, της σύνταξης και των αντίστοιχων τυπικών βιβλιοθηκών, οι δύο γλώσσες είναι διακριτές και διαφέρουν σε μεγάλο βαθμό στο σχεδιασμό.

3.4.1 JQUERY

Το jQuery είναι μια βιβλιοθήκη JavaScript που έχει σχεδιαστεί για να απλοποιεί τη διασταύρωση και τη διαχείριση δέντρων HTML DOM, καθώς και τον χειρισμό συμβάντων, την κινούμενη εικόνα CSS και το Ajax. Είναι δωρεάν, λογισμικό ανοιχτού κώδικα που χρησιμοποιεί την επιτρεπτή άδεια MIT. Από τον Μάιο του 2019, το jQuery χρησιμοποιείται από το 73% των 10 εκατομμυρίων πιο δημοφιλών ιστότοπων. Η ανάλυση ιστού δείχνει ότι είναι η πιο ευρέως διαδεδομένη βιβλιοθήκη JavaScript με μεγάλο περιθώριο, έχοντας τουλάχιστον 3 έως 4 φορές περισσότερη χρήση από οποιαδήποτε άλλη βιβλιοθήκη JavaScript.

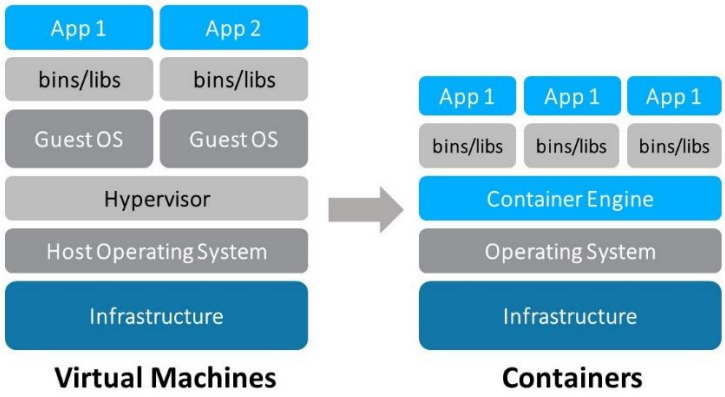
Η σύνταξη του jQuery έχει σχεδιαστεί για να διευκολύνει την πλοήγηση σε ένα έγγραφο, την επιλογή στοιχείων DOM, τη δημιουργία κινούμενων σχεδίων, τη διαχείριση συμβάντων και την ανάπτυξη εφαρμογών Ajax. Το jQuery παρέχει επίσης δυνατότητες στους προγραμματιστές να δημιουργούν προσθήκες πάνω από τη βιβλιοθήκη JavaScript. Αυτό επιτρέπει στους προγραμματιστές να δημιουργούν αφαιρέσεις για χαμηλού επιπέδου αλληλεπίδραση και κινούμενα σχέδια, προηγμένα εφέ και υψηλού επιπέδου, θεματικά widget. Η αρθρωτή (modular) προσέγγιση στη βιβλιοθήκη jQuery επιτρέπει τη δημιουργία ισχυρών δυναμικών ιστοσελίδων και εφαρμογών Ιστού.

Το σύνολο των βασικών δυνατοτήτων του jQuery - επιλογές στοιχείων DOM, διασταύρωση και χειραγώγηση - ενεργοποιήθηκε από τη μηχανή επιλογής, δημιούργησε ένα νέο είδος προγραμματισμού, αλγόριθμους σύντηξης και δομές δεδομένων DOM. Αυτό επηρέασε την αρχιτεκτονική άλλων πλαισίων JavaScript όπως το YUI v3 και το Dojo, ενισχύοντας αργότερα τη δημιουργία του τυπικού API Selectors. Αργότερα, αυτό το στυλ έχει βελτιωθεί με μια βαθύτερη σύντηξη αλγορίθμων-δεδομένων σε έναν κληρονόμο του jQuery, του framework D3.js.

Η Microsoft και το Nokia χρησιμοποιούν jQuery στις πλατφόρμες τους. Η Microsoft τη συμπεριλαμβάνει με το Visual Studio για χρήση στα πλαίσια ASP.NET AJAX και ASP.NET MVC της Microsoft, ενώ η Nokia το έχει ενσωματώσει στην πλατφόρμα ανάπτυξης widget Web Run-Time.

3.5 DOCKER CONTAINER

Το Docker είναι ένα σύνολο προϊόντων πλατφόρμας (PaaS) που χρησιμοποιούν εικονικοποίηση σε επίπεδο λειτουργικού συστήματος για την παράδοση λογισμικού σε πακέτα που ονομάζονται κοντέινερ. Τα εμπορευματοκιβώτια είναι απομονωμένα το ένα από το άλλο και ομαδοποιούν το δικό τους λογισμικό, βιβλιοθήκες και αρχεία διαμόρφωσης. Μπορούν να επικοινωνούν μεταξύ τους μέσω καλά καθορισμένων καναλιών. Όλα τα κοντέινερ εκτελούνται από έναν πυρήνα λειτουργικού συστήματος και επομένως χρησιμοποιούν λιγότερους πόρους από τις εικονικές μηχανές.



Εικόνα 11 Αρχιτεκτονική Docker Container

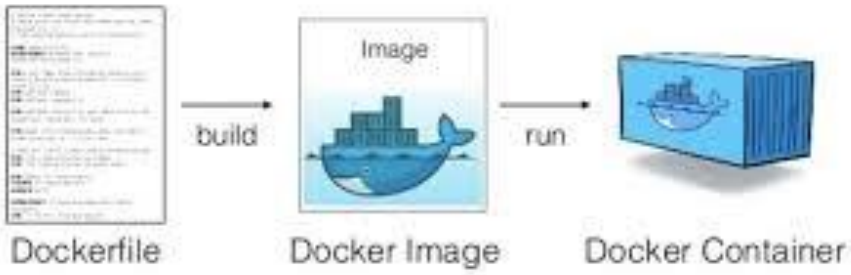
Το λογισμικό που φιλοξενεί τα κοντέινερ ονομάζεται Docker Engine. Ξεκίνησε για πρώτη φορά το 2013 και αναπτύχθηκε από την Docker, Inc.

Το λογισμικό Docker ως προσφορά υπηρεσιών αποτελείται από τρία στοιχεία:

- Λογισμικό: Το daemon Docker, που ονομάζεται dockerd, είναι μια επίμονη διαδικασία που διαχειρίζεται κοντέινερ Docker και χειρίζεται αντικείμενα κοντέινερ. Το daemon ακούει αιτήματα που αποστέλλονται μέσω του Docker Engine API. Το πρόγραμμα πελάτη

Docker, που ονομάζεται docker, παρέχει μια διεπαφή γραμμής εντολών που επιτρέπει στους χρήστες να αλληλεπιδρούν με τους δαίμονες Docker.

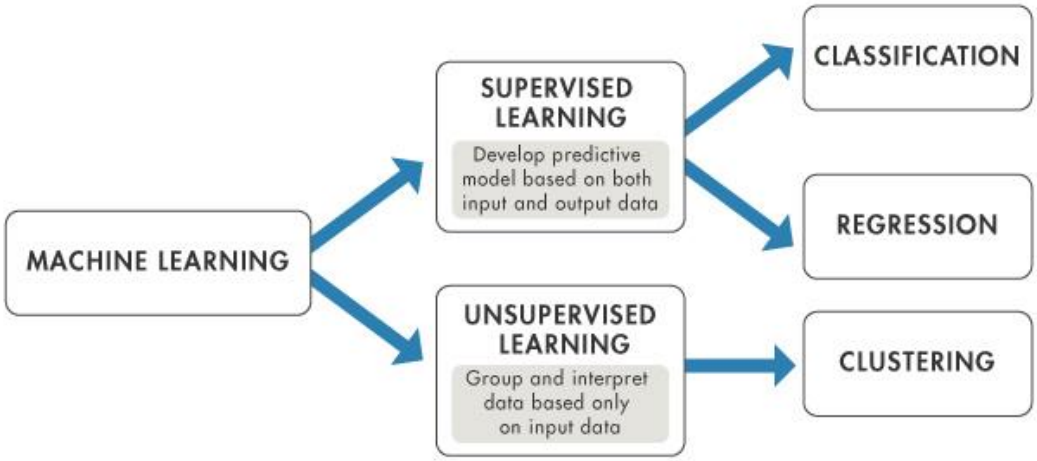
- Αντικείμενα: Τα αντικείμενα Docker είναι διάφορες οντότητες που χρησιμοποιούνται για τη συναρμολόγηση μιας εφαρμογής στο Docker. Οι κύριες κατηγορίες αντικειμένων Docker είναι εικόνες, κοντέινερ και υπηρεσίες.
  Το κοντέινερ Docker είναι ένα τυποποιημένο περιβάλλον με εγκλεισμό που εκτελεί εφαρμογές. Η διαχείριση ενός κοντέινερ χρησιμοποιεί το Docker API ή CLI.
  Η εικόνα Docker είναι ένα πρότυπο μόνο για ανάγνωση που χρησιμοποιείται για την κατασκευή κοντέινερ. Οι εικόνες χρησιμοποιούνται για την αποθήκευση και αποστολή εφαρμογών. Η υπηρεσία Docker επιτρέπει την κλιμάκωση των κοντέινερ σε πολλαπλούς daemon Docker. Το αποτέλεσμα είναι γνωστό ως σμήνος, ένα σύνολο συνεργαζόμενων daemon που επικοινωνούν μέσω του Docker API.
- Μητρώα: Ένα μητρώο Docker είναι ένα αποθετήριο για εικόνες Docker. Οι πελάτες (clients) του Docker συνδέονται σε μητρώα για λήψη ("pull") εικόνων για χρήση ή μεταφόρτωση ("push") εικόνων που έχουν δημιουργήσει. Τα μητρώα μπορούν να είναι δημόσια ή ιδιωτικά. Δύο κύρια δημόσια μητρώα είναι το Docker Hub και το Docker Cloud. Το Docker Hub είναι το προεπιλεγμένο μητρώο όπου το Docker αναζητά εικόνες. Τα μητρώα Docker επιτρέπουν επίσης τη δημιουργία ειδοποιήσεων βάσει συμβάντων.



Εικόνα 12 Παράδειγμα δημιουργίας Docker Container

4. ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ ΚΑΙ ΤΕΧΝΙΚΕΣ

Σε αυτό το κεφάλαιο γίνεται μια εισαγωγή στη μηχανική μάθηση καθώς και στους αλγορίθμους που χρησιμοποιούνται από την πλατφόρμα. Επίσης θα γίνει αναλυτική αναφορά και στις τεχνικές επεξεργασίας δεδομένων που παρέχονται.



Εικόνα 13 Δομή Μηχανικής Μάθησης

Η μηχανική μάθηση (ML) είναι η μελέτη αλγορίθμων υπολογιστών που βελτιώνεται αυτόματα μέσω της εμπειρίας. Θεωρείται ως μέρος της τεχνητής νοημοσύνης. Οι αλγόριθμοι μηχανικής μάθησης δημιουργούν ένα μοντέλο βασισμένο σε δείγματα δεδομένων, γνωστά ως "δεδομένα εκπαίδευσης", προκειμένου να λαμβάνουν προβλέψεις ή αποφάσεις χωρίς να έχουν προγραμματιστεί ρητά να το κάνουν. Οι αλγόριθμοι μηχανικής εκμάθησης χρησιμοποιούνται σε μια ευρεία ποικιλία εφαρμογών, όπως το φιλτράρισμα email και η όραση του υπολογιστή, όπου είναι δύσκολο ή ανέφικτο να αναπτυχθούν συμβατικοί αλγόριθμοι για την εκτέλεση των απαιτούμενων εργασιών.

Ένα υποσύνολο της μηχανικής μάθησης σχετίζεται στενά με την υπολογιστική στατιστική, η οποία επικεντρώνεται στην πραγματοποίηση προβλέψεων χρησιμοποιώντας υπολογιστές, αλλά δεν είναι όλη η μηχανική μάθηση στατιστική μάθηση. Η μελέτη της μαθηματικής βελτιστοποίησης παρέχει μεθόδους, θεωρία και τομείς εφαρμογής στον τομέα της μηχανικής μάθησης. Η εξόρυξη δεδομένων είναι ένα σχετικό πεδίο μελέτης, που εστιάζει στην διερευνητική ανάλυση δεδομένων μέσω της μη εποπτευόμενης (unsupervised) μάθησης. Στην εφαρμογή της σε επιχειρηματικά προβλήματα, η μηχανική μάθηση αναφέρεται επίσης ως προγνωστική ανάλυση. Ο πρωταρχικός στόχος είναι να επιτρέπεται στους υπολογιστές να μαθαίνουν αυτόματα χωρίς ανθρώπινη παρέμβαση ή βοήθεια και να προσαρμόζουν ανάλογα τις ενέργειες.

Οι αλγόριθμοι μηχανικής μάθησης κατηγοριοποιούνται συχνά ως εποπτευόμενοι (supervised) ή μη εποπτευόμενοι (unsupervised).

- Οι εποπτευόμενοι αλγόριθμοι μηχανικής μάθησης μπορούν να εφαρμόσουν ό, τι έχει μάθει στο παρελθόν σε νέα δεδομένα χρησιμοποιώντας επισημασμένα παραδείγματα για την πρόβλεψη μελλοντικών γεγονότων. Ξεκινώντας από την ανάλυση ενός γνωστού συνόλου δεδομένων εκπαίδευσης, ο αλγόριθμος εκμάθησης παράγει μια συνάρτηση για να κάνει προβλέψεις σχετικά με τις τιμές εξόδου. Το σύστημα είναι σε θέση να παρέχει αποτέλεσμα για οποιαδήποτε νέα είσοδο μετά από επαρκή εκπαίδευση. Ο αλγόριθμος εκμάθησης μπορεί επίσης να συγκρίνει την έξοδο του με τη σωστή, προοριζόμενη έξοδο και να βρει σφάλματα προκειμένου να τροποποιήσει ανάλογα το μοντέλο.
- Οι μη εποπτευόμενοι αλγόριθμοι μηχανικής μάθησης χρησιμοποιούνται όταν οι πληροφορίες που υπάρχουν για την εκπαίδευση δεν ταξινομούνται ούτε επισημαίνονται. Η μη εποπτευόμενη μάθηση μελετά πώς τα συστήματα μπορούν να συμπεράνουν μια συνάρτηση για να περιγράψουν μια κρυφή δομή από δεδομένα χωρίς ετικέτα. Το σύστημα δεν καταλαβαίνει τη σωστή έξοδο, αλλά διερευνά τα δεδομένα και μπορεί να εξαγάγει συμπεράσματα από σύνολα δεδομένων για να περιγράψει κρυφές δομές από δεδομένα χωρίς ετικέτα.
- Οι ημι-εποπτευόμενοι αλγόριθμοι μηχανικής μάθησης εμπίπτουν κάπου μεταξύ εποπτευόμενης και μη εποπτευόμενης μάθησης, δεδομένου ότι χρησιμοποιούν δεδομένα με αλλά και χωρίς ετικέτα για εκπαίδευση - συνήθως μια μικρή ποσότητα δεδομένων με ετικέτα και μια μεγάλη ποσότητα δεδομένων χωρίς ετικέτα. Τα συστήματα που χρησιμοποιούν αυτήν τη μέθοδο είναι σε θέση να βελτιώσουν σημαντικά την ακρίβεια της μάθησης. Συνήθως, η ημι-εποπτευόμενη μάθηση επιλέγεται όταν τα ληφθέντα δεδομένα με ετικέτα απαιτούν εξειδικευμένους και σχετικούς πόρους για να την εκπαιδεύσουν / μάθουν από αυτήν. Διαφορετικά, η απόκτηση δεδομένων χωρίς ετικέτα γενικά δεν απαιτεί πρόσθετους πόρους.
- Οι αλγόριθμοι μηχανικής εκμάθησης ενίσχυσης είναι μια μέθοδος μάθησης που αλληλεπιδρά με το περιβάλλον της παράγοντας ενέργειες και ανακαλύπτει λάθη ή ανταμοιβές. Η αναζήτηση δοκιμών και σφαλμάτων και η καθυστερημένη ανταμοιβή είναι τα πιο σχετικά χαρακτηριστικά της μάθησης ενίσχυσης. Αυτή η μέθοδος επιτρέπει σε μηχανές και πράκτορες λογισμικού (software agents) να καθορίζουν αυτόματα την ιδανική συμπεριφορά σε ένα συγκεκριμένο πλαίσιο, προκειμένου να μεγιστοποιηθεί η απόδοσή της. Απαιτείται απλή ανατροφοδότηση ανταμοιβής για να μάθει ο πράκτορας ποια ενέργεια είναι καλύτερη. Αυτό είναι γνωστό ως σήμα ενίσχυσης.

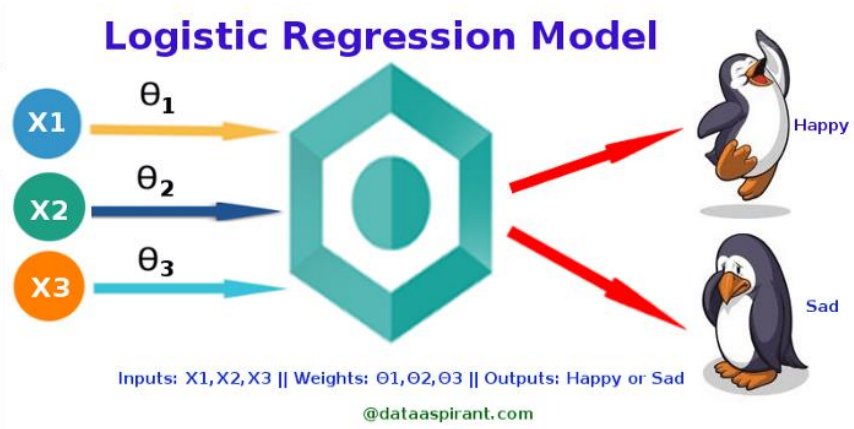
Η μηχανική εκμάθηση επιτρέπει την ανάλυση τεράστιων ποσοτήτων δεδομένων. Αν και γενικά παρέχει ταχύτερα, πιο ακριβή αποτελέσματα για τον εντοπισμό κερδοφόρων ευκαιριών ή επικίνδυνων ρίσκων, μπορεί επίσης να απαιτεί επιπλέον χρόνο και πόρους για να εκπαιδευτεί σωστά. Ο συνδυασμός της μηχανικής μάθησης με την τεχνητή νοημοσύνη και τις γνωστικές

τεχνολογίες μπορεί να την κάνει ακόμη πιο αποτελεσματική στην επεξεργασία μεγάλου όγκου πληροφοριών.

4.1 ΑΛΓΟΡΙΘΜΟΙ ΤΑΞΙΝΟΜΗΣΗΣ (CLASSIFICATION)

Η ταξινόμηση είναι το πρόβλημα του εντοπισμού σε ποιο από ένα σύνολο κατηγοριών ανήκει μια νέα παρατήρηση, βάσει ενός εκπαιδευτικού συνόλου δεδομένων που περιέχει παρατηρήσεις (ή περιπτώσεις) των οποίων η κατηγορία είναι γνωστή. Παραδείγματα είναι η ανάθεση ενός δεδομένου μηνύματος ηλεκτρονικού ταχυδρομείου στην τάξη "ανεπιθύμητη αλληλογραφία" ή "μη ανεπιθύμητη αλληλογραφία" και η εκχώρηση διάγνωσης σε έναν συγκεκριμένο ασθενή με βάση τα παρατηρούμενα χαρακτηριστικά του ασθενούς (φύλο, αρτηριακή πίεση, παρουσία ή απουσία ορισμένων συμπτωμάτων κ.λπ.) . Η ταξινόμηση είναι ένα παράδειγμα αναγνώρισης προτύπων. Στην ορολογία της μηχανικής μάθησης, η ταξινόμηση θεωρείται μια περίπτωση εποπτευόμενης μάθησης, δηλαδή, μάθησης όπου υπάρχει ένα εκπαιδευτικό σύνολο σωστών αναγνωρισμένων παρατηρήσεων. Η αντίστοιχη διαδικασία χωρίς επίβλεψη είναι γνωστή ως ομαδοποίηση, και περιλαμβάνει την ομαδοποίηση δεδομένων σε κατηγορίες με βάση κάποιο μέτρο εγγενούς ομοιότητας ή απόστασης. Ένας αλγόριθμος που χρησιμοποιείται για ταξινόμηση είναι γνωστός ως ταξινομητής. Ο όρος "ταξινομητής" μερικές φορές αναφέρεται επίσης στη μαθηματική συνάρτηση, που εφαρμόζεται από έναν αλγόριθμο ταξινόμησης, ο οποίος κατηγοριοποιεί δεδομένα εισόδου σε μια κατηγορία.

4.1.1 ΛΟΓΙΣΤΙΚΗ ΠΑΛΙΝΔΡΟΜΗΣΗ (LOGISTIC REGRESSION)

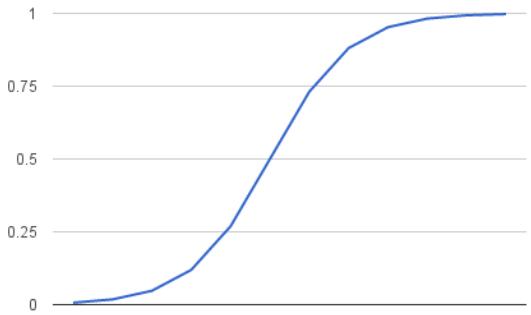


Εικόνα 14 Λογιστική Παλινδρόμηση

Η λογιστική παλινδρόμηση είναι ένα στατιστικό μοντέλο που στη βασική του μορφή χρησιμοποιεί μια λογιστική συνάρτηση για τη μοντελοποίηση μιας δυαδικής εξαρτώμενης μεταβλητής. Στην ανάλυση παλινδρόμησης, η λογιστική παλινδρόμηση υπολογίζει τις παραμέτρους ενός λογιστικού μοντέλου (μια μορφή δυαδικής παλινδρόμησης). Η λογιστική συνάρτηση, που ονομάζεται επίσης σιγμοειδής συνάρτηση, αναπτύχθηκε από στατιστικολόγους για να περιγράψει τις ιδιότητες της αύξησης του πληθυσμού στην οικολογία, αυξάνοντας γρήγορα και μεγιστοποιώντας τη φέρουσα ικανότητα του περιβάλλοντος. Είναι μια καμπύλη σχήματος S που μπορεί να πάρει οποιονδήποτε πραγματικό αριθμό και να τον αντιστοιχίσει σε τιμή μεταξύ 0 και 1, αλλά ποτέ ακριβώς σε αυτά τα όρια.

$$\mathbf{1 \,/\, (1 \,+\, e\text{^} \,-\, value)}$$

Όπου το e είναι η βάση των φυσικών λογάριθμων και η τιμή είναι η πραγματική αριθμητική τιμή που θέλετε να μεταμορφώσετε. Για παράδειγμα, μια γραφική παράσταση των αριθμών μεταξύ -5 και 5 που μετατρέπονται στο εύρος 0 και 1 χρησιμοποιώντας τη λογιστική συνάρτηση.



Εικόνα 15 Σιγμοειδής συνάρτηση

Καθώς ορίστηκε η λογιστική συνάρτηση στη συνέχεια θα δούμε πως χρησιμοποιείται στη λογιστική παλινδρόμηση.

Η λογιστική παλινδρόμηση αναπαριστάται ως μια εξίσωση η οποία μοιάζει πολύ με τη γραμμική παλινδρόμηση. Οι τιμές εισόδου (X) συνδυάζονται γραμμικά χρησιμοποιώντας βάρη ή τιμές συντελεστή για να προβλέψουν μια τιμή εξόδου (P). Μια βασική διαφορά από τη γραμμική παλινδρόμηση είναι ότι η τιμή εξόδου που μοντελοποιείται είναι δυαδικές τιμές (0 ή 1) και όχι αριθμητική τιμή. Ακολουθεί ένα παράδειγμα εξίσωσης λογιστικής παλινδρόμησης:

$$P = \frac{e^{a+bX}}{1+e^{a+bX}}$$

Όπου P είναι η προβλεπόμενη έξοδος, α είναι ο όρος μεροληψίας ή αναχαίτισης και b είναι ο συντελεστής για την τιμή μίας εισόδου (X). Κάθε στήλη στα δεδομένα εισαγωγής σας έχει έναν σχετικό συντελεστή b (μια σταθερή πραγματική τιμή) που προκύπτει από τα δεδομένα εκπαίδευσης. Η πραγματική αναπαράσταση του μοντέλου αποθηκεύεται στη μνήμη ή σε ένα αρχείο είναι οι συντελεστές στην εξίσωση (δηλ. τα α και b).

Η λογιστική παλινδρόμηση διαμορφώνει την πιθανότητα της προεπιλεγμένης κλάσης (π.χ. την πρώτη κλάση). Στην πραγματικότητα, μοντελοποιούμε την πιθανότητα ότι μια είσοδος (X) ανήκει στην προεπιλεγμένη κλάση (Y = 1), μπορούμε να το γράψουμε επίσημα ως:

$$\mathrm{P}\,(\mathrm{X}) \;=\; \mathrm{P}\,(\mathrm{Y} \;=\; 1 \mid \mathrm{X})$$

Η λογιστική παλινδρόμηση είναι μια γραμμική μέθοδος, αλλά οι προβλέψεις μετατρέπονται χρησιμοποιώντας τη λογιστική συνάρτηση. Ο αντίκτυπος αυτού είναι ότι δεν μπορούμε πλέον να κατανοήσουμε τις προβλέψεις ως γραμμικό συνδυασμό των εισόδων όπως μπορούμε με τη γραμμική παλινδρόμηση. Το μοντέλο στην παραπάνω εικόνα *Figure 3* γίνεται:

$$ln\,(p\,(X)\;/\;1\;-\;p\,(X))\;=\;\alpha\;+\;b\;*\;X$$

Έτσι μπορούμε να δούμε ότι ο υπολογισμός της εξόδου στα δεξιά είναι πάλι γραμμικός και η είσοδος στα αριστερά είναι ένα αρχείο καταγραφής της πιθανότητας της προεπιλεγμένης κλάσης. Αυτή η αναλογία στα αριστερά ονομάζεται πιθανότητες (odds) της προεπιλεγμένης κατηγορίας (default class). Οι πιθανότητες υπολογίζονται ως αναλογία της πιθανότητας να συμβεί διαιρούμενη με την πιθανότητα να μη συμβεί το εκάστοτε συμβάν.

$$ln\,(\pi\theta\alpha\nu\acute{o}\tau\eta\tau\varepsilon\varsigma)\;=\;\alpha+\;b\;*\;X$$

Οι συντελεστές (τιμές b) του αλγορίθμου λογιστικής παλινδρόμησης πρέπει να εκτιμηθούν από τα δεδομένα εκπαίδευσης. Αυτό γίνεται χρησιμοποιώντας εκτίμηση μέγιστης πιθανότητας (maximum-likelihood). Η εκτίμηση μέγιστης πιθανότητας είναι ένας κοινός αλγόριθμος μάθησης που χρησιμοποιείται από μια ποικιλία αλγορίθμων μηχανικής μάθησης, αν και κάνει υποθέσεις σχετικά με την κατανομή των δεδομένων. Οι καλύτεροι συντελεστές θα είχαν ως αποτέλεσμα ένα μοντέλο που θα προέβλεπε μια τιμή πολύ κοντά στο 1 για την προεπιλεγμένη κλάση και μια τιμή πολύ κοντά στο 0 για την άλλη κατηγορία. Η διαίσθηση της μέγιστης πιθανότητας για λογιστική παλινδρόμηση είναι ότι μια διαδικασία αναζήτησης αναζητά τιμές για τους συντελεστές (τιμές b) που ελαχιστοποιούν το σφάλμα στις πιθανότητες που προβλέπονται από το μοντέλο σε εκείνα στα δεδομένα.

4.1.2 GRADIENT BOOSTING TREE CLASSIFIER

Η ιδέα πίσω από το "gradient boosting" είναι ότι παίρνει μια αδύναμη υπόθεση ή έναν αδύναμο αλγόριθμο μάθησης και να κάνει μια σειρά τροποποιήσεων σε αυτή/αυτόν που θα βελτιώσουν τη δύναμη της υπόθεσης / αλγορίθμου. Αυτός ο τύπος ενίσχυσης υπόθεσης (hypothesis boosting) βασίζεται στην ιδέα της μάθησης Probability Approximately Correct (PAC). Στην ενίσχυση της υπόθεσης, εξετάζετε όλες τις παρατηρήσεις στις οποίες εκπαιδεύεται ο αλγόριθμος μηχανικής μάθησης και αφήνετε μόνο τις παρατηρήσεις στις οποίες η μέθοδος μηχανικής μάθησης ταξινόμησε με επιτυχία πίσω, αφαιρώντας τις άλλες παρατηρήσεις. Ένας νέος αδύναμος αλγόριθμος μάθησης (learner) δημιουργείται στο σύνολο των δεδομένων που δεν ταξινομήθηκαν σωστά και στη συνέχεια διατηρούνται μόνο τα παραδείγματα που ταξινομήθηκαν με επιτυχία. Αυτή η ιδέα υλοποιήθηκε στον αλγόριθμο Adaptive Boosting (AdaBoost). Για το AdaBoost, δημιουργούνται πολλοί αδύναμοι αλγόριθμοι μάθησης με την προετοιμασία πολλών αλγορίθμων δέντρων αποφάσεων που έχουν μόνο ένα διαχωρισμό. Δίνονται βάρη στις παρατηρήσεις του συνόλου δεδομένων εκπαίδευσης και περισσότερο βάρος αποδίδεται σε περιπτώσεις που είναι δύσκολο να ταξινομηθούν. Περισσότεροι αδύναμοι αλγόριθμοι μάθησης (learners) προστίθενται διαδοχικά στο σύστημα και ανατίθενται στις πιο δύσκολες εκπαιδευτικές περιπτώσεις. Στο AdaBoost, οι προβλέψεις γίνονται με πλειοψηφία, με τις περιπτώσεις να ταξινομούνται σύμφωνα με την όποια τάξη λαμβάνει τις περισσότερες ψήφους από τους αδύναμους αλγορίθμους.

Οι ταξινομητές gradient boosting trees είναι η μέθοδος AdaBoosting σε συνδυασμό με την ελαχιστοποίηση των βαρών, μετά την οποία υπολογίζονται εκ νέου οι ταξινομητές και οι είσοδοι με βάρη. Ο στόχος των ταξινομητών Gradient Boosting είναι να ελαχιστοποιηθεί η απώλεια ή η διαφορά μεταξύ της πραγματικής τιμής τάξης του παραδείγματος εκπαίδευσης και της προβλεπόμενης τιμής της. Έγιναν βελτιώσεις σε αυτή τη διαδικασία και δημιουργήθηκαν οι μηχανές Gradient Boosting. Στην περίπτωση των μηχανών Gradient boosting, κάθε φορά που προστίθεται ένας νέος ασθενής αλγόριθμος μάθησης στο μοντέλο, τα βάρη των προηγούμενων αλγορίθμων μάθησης παγώνουν στη θέση τους, αφήνονται αμετάβλητοι καθώς εισάγονται τα νέα επίπεδα. Αυτό διαφέρει από τις προσεγγίσεις που χρησιμοποιούνται στο AdaBoosting όπου οι τιμές προσαρμόζονται όταν προστίθενται νέοι αλγόριθμοι μάθησης. Η δύναμη των μηχανών gradient boosting προέρχεται από το γεγονός ότι μπορούν να χρησιμοποιηθούν σε δυαδικά προβλήματα ταξινόμησης, σε προβλήματα ταξινόμησης πολλαπλών τάξεων και ακόμη και σε προβλήματα παλινδρόμησης.

Ο Gradient Boosting ταξινομητής εξαρτάται από τη συνάρτηση απώλειας (loss function). Μπορεί να χρησιμοποιηθεί μια προσαρμοσμένη συνάρτηση απώλειας (custom loss function), και πολλές τυποποιημένες (standarized) συναρτήσεις απώλειας υποστηρίζονται από ταξινομητές gradient boosting, αλλά η συνάρτηση απώλειας πρέπει να είναι διαφορίσιμη. Τα συστήματα gradient boosting έχουν δύο απαραίτητα μέρη: έναν ασθενή αλγόριθμο μάθησης και ένα προσθετικό σκέλος (additive component). Χρησιμοποιούν δέντρα αποφάσεων ως αδύναμους αλγορίθμους

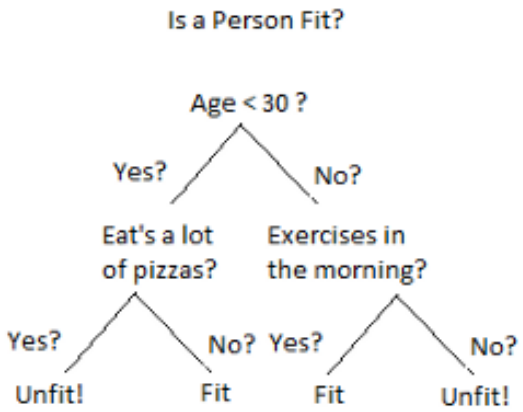
μάθησης. Το προσθετικό σκέλος ενός gradient boosting μοντέλου προέρχεται από το γεγονός ότι τα δέντρα προστίθενται στο μοντέλο με την πάροδο του χρόνου, και όταν συμβεί αυτό, τα υπάρχοντα δέντρα δεν μεταβάλλονται, οι τιμές τους παραμένουν σταθερές. Χρησιμοποιείται μια διαδικασία παρόμοια με την, μείωση βαθμίδας για την ελαχιστοποίηση του σφάλματος μεταξύ των παραμέτρων. Αυτό επιτυγχάνεται λαμβάνοντας την υπολογιζόμενη απώλεια και εκτελώντας μείωση βαθμίδας για να μειώσουμε αυτήν την απώλεια. Στη συνέχεια, οι παράμετροι του δέντρου τροποποιούνται για να μειωθεί η υπολειμματική απώλεια (residual loss). Η έξοδος του νέου δέντρου προσαρτάται στη συνέχεια στην έξοδο των προηγούμενων δέντρων που χρησιμοποιούνται στο μοντέλο. Αυτή η διαδικασία επαναλαμβάνεται έως ότου επιτευχθεί ένας προκαθορισμένος αριθμός δέντρων, ή η απώλεια μειωθεί κάτω από ένα συγκεκριμένο όριο.

4.1.3 ΤΑΞΙΝΟΜΗΤΕΣ ΔΕΝΔΡΩΝ ΑΠΟΦΑΣΗΣ (DECISION TREE CLASSIFIERS)

Η εκμάθηση δέντρων αποφάσεων είναι μία από τις προσεγγιστικές προβλέψεις μοντελοποίησης που χρησιμοποιούνται στην στατιστική, στην εξόρυξη δεδομένων και στη μηχανική μάθηση. Χρησιμοποιεί ένα δέντρο αποφάσεων (ως μοντέλο πρόβλεψης) για να μεταβεί από τις παρατηρήσεις σχετικά με ένα αντικείμενο (που αντιπροσωπεύεται στους κλάδους) σε συμπεράσματα σχετικά με την ετικέτα του αντικειμένου (που απεικονίζεται στα φύλλα). Δενδρικά μοντέλα όπου η μεταβλητή στόχος μπορεί να λάβει ένα διακριτό σύνολο τιμών ονομάζονται δέντρα ταξινόμησης. Σε αυτές τις δομές δέντρων, τα φύλλα αντιπροσωπεύουν ετικέτες κλάσης και τα κλαδιά αντιπροσωπεύουν συνδέσεις χαρακτηριστικών που οδηγούν σε αυτές τις ετικέτες κλάσης. Το δέντρο απόφασης αποτελείται από:

- Κόμβοι: Δοκιμή για την τιμή ενός συγκεκριμένου χαρακτηριστικού.
- Ακμές: Αντιστοιχεί στο αποτέλεσμα μιας δοκιμής και το συνδέει στον επόμενο κόμβο ή φύλλο.
- Κόμβοι φύλλων: Κόμβοι τερματικού που προβλέπουν το αποτέλεσμα (αντιπροσωπεύουν ετικέτες κλάσης ή κατανομή κλάσης)

Για παράδειγμα, ας υποθέσουμε ότι θέλουμε να προβλέψουμε εάν ένα άτομο είναι γυμνασμένο ή αγύμναστο, λαμβάνοντας υπόψη τις πληροφορίες του όπως η ηλικία, οι διατροφικές συνήθειες, η σωματική δραστηριότητα κ.λπ. Οι κόμβοι αποφάσεων είναι οι ερωτήσεις όπως «Ποια είναι η ηλικία;», «Άσκηση;», » Τρώει πολλές πίτσες; Και τα φύλλα αντιπροσωπεύουν αποτελέσματα όπως «γυμνασμένος» ή «αγύμναστος».



Εικόνα 16 Παράδειγμα Δένδρου Απόφασης

Η βασική ιδέα είναι ότι χρησιμοποιήσετε ένα δέντρο αποφάσεων για να χωρίσει τον χώρο δεδομένων σε περιοχές συμπλέγματος (ή πυκνές) και κενές (ή αραιά) περιοχές. Στην απόφαση ταξινόμησης δέντρων ένα νέο παράδειγμα ταξινομείται υποβάλλοντας το σε μια σειρά δοκιμών που καθορίζουν την ετικέτα κατηγορίας του παραδείγματος. Αυτά τα τεστ οργανώνονται σε μια ιεραρχική δομή που ονομάζεται δέντρο αποφάσεων. Τα δέντρα απόφασης ακολουθούν τον αλγόριθμο Divide-and-Conquer. Τα δέντρα αποφάσεων χτίζονται χρησιμοποιώντας ένα ευρετικό που ονομάζεται αναδρομική διαμέριση (recursive partition). Αυτή η προσέγγιση είναι επίσης κοινώς γνωστή ως divide and conquer επειδή διαχωρίζει τα δεδομένα σε υποσύνολα, τα οποία στη συνέχεια χωρίζονται επανειλημμένα σε ακόμη μικρότερα υποσύνολα, και ούτω καθεξής έως ότου η διαδικασία σταματήσει όταν ο αλγόριθμος καθορίσει ότι τα δεδομένα εντός των υποομάδων είναι αρκετά ομοιογενή ή ικανοποιήθηκε άλλο κριτήριο διακοπής.

Βασικός divide and conquer αλγόριθμος:

- Επιλέγετε ένας κόμβος ως ρίζα για δοκιμή και δημιουργείτε κλάδος για κάθε πιθανό αποτέλεσμα του τεστ.
- Διαχωρίστε τις παρουσίες σε υποσύνολα. Ένα για κάθε κλάδο που εκτείνεται από τον κόμβο.
- Επαναλάβετε αναδρομικά για κάθε κλάδο, χρησιμοποιώντας μόνο περιπτώσεις που φτάνουν στον κλάδο.
- Σταματήστε την επανάληψη για ένα υποκατάστημα εάν όλες οι παρουσίες του έχουν την ίδια τάξη.

Ταξινομητές Δένδρων απόφασης:

- Χρησιμοποιώντας τον αλγόριθμο απόφασης, ξεκινάμε από τη ρίζα δέντρου και χωρίζουμε τα δεδομένα σχετικά με τη δυνατότητα που έχει ως αποτέλεσμα το μεγαλύτερο κέρδος πληροφοριών.

- Σε μια επαναληπτική διαδικασία, μπορούμε στη συνέχεια να επαναλάβουμε αυτήν τη διαδικασία διαχωρισμού σε κάθε κόμβο παιδί μέχρι τα φύλλα να είναι καθαρά. Αυτό σημαίνει ότι τα δείγματα σε κάθε κόμβο φύλλων ανήκουν στην ίδια κατηγορία.
- Στην πράξη, ενδέχεται να θέσουμε ένα όριο στο βάθος του δέντρου για να αποφευχθεί το overfitting. Συμβιβάζουμε κάπως εδώ την καθαρότητα, καθώς τα τελικά φύλλα ενδέχεται να έχουν κάποια φασαρία.

Πλεονεκτήματα ταξινομητών δένδρων απόφασης:

- Φθηνό στην κατασκευή.
- Εξαιρετικά γρήγορη ταξινόμηση άγνωστων στοιχείων.
- Εύκολο στην ερμηνεία για μικρού μεγέθους δέντρα.
- Ακρίβεια συγκρίσιμη με άλλες τεχνικές ταξινόμησης για πολλά απλά σύνολα δεδομένων.
- Εξαιρούνται ασήμαντα χαρακτηριστικά.

Μειονεκτήματα ταξινομητών δένδρων απόφασης:

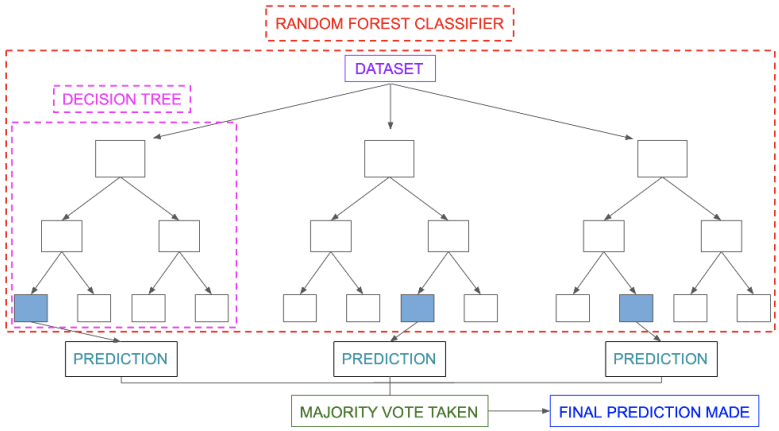
- Επιτυγχάνεται εύκολα overfit.
- Το όριο απόφασης περιορίζεται στο να είναι παράλληλο με τους άξονες χαρακτηριστικών.
- Τα μοντέλα δέντρων αποφάσεων συχνά είναι προκατειλημμένα (biased) σε χαρακτηριστικά που έχουν μεγάλο αριθμό επιπέδων.
- Μικρές αλλαγές στα δεδομένα εκπαίδευσης μπορούν να οδηγήσουν σε μεγάλες αλλαγές στη λογική των αποφάσεων.
- Τα μεγάλα δέντρα μπορεί να είναι δύσκολο να ερμηνευτούν και οι αποφάσεις που λαμβάνουν μπορεί να φαίνονται αντίθετες.

Εφαρμογές των ταξινομητών δένδρων απόφασης στην πραγματική ζωή:

- Βιοϊατρική (δέντρα αποφάσεων για τον προσδιορισμό χαρακτηριστικών που θα χρησιμοποιηθούν σε εμφυτεύσιμες συσκευές).
- Οικονομική ανάλυση (Ικανοποίηση πελατών με ένα προϊόν ή υπηρεσία).
- Αστρονομία (ταξινόμηση γαλαξιών).
- Έλεγχος συστήματος.
- Κατασκευή και παραγωγή (ποιοτικός έλεγχος, κατασκευή ημιαγωγών, κ.λπ.).
- Ιατρική (διάγνωση, καρδιολογία, ψυχιατρική).
- Φυσική (Ανίχνευση σωματιδίων).

4.1.4 ΤΑΞΙΝΟΜΗΤΕΣ ΤΥΧΑΙΟΥ ΔΑΣΟΥΣ (RANDOM FOREST CLASSIFIERS)

Το τυχαίο δάσος, όπως υποδηλώνει το όνομά του, αποτελείται από μεγάλο αριθμό μεμονωμένων δέντρων απόφασης που λειτουργούν ως σύνολο. Κάθε μεμονωμένο δέντρο στο τυχαίο δάσος ξετυλίγει μια πρόβλεψη για την κλάση και η κλάση με τις περισσότερες ψήφους γίνεται η πρόβλεψη του μοντέλου μας. Η θεμελιώδης έννοια πίσω από το τυχαίο δάσος είναι μια απλή αλλά ισχυρή - η σοφία του πλήθους. Ένας μεγάλος αριθμός από σχετικά μη συσχετιζόμενα μοντέλα (δέντρα) που λειτουργούν όλα μαζί για να κάνουν προβλέψεις θα ξεπεράσουν οποιοδήποτε από τα μεμονωμένα μοντέλα και η χαμηλή συσχέτιση (correlation) μεταξύ των μοντέλων είναι το κλειδί.



Εικόνα 17 Παράδειγμα Αρχιτεκτονικής για Ταξινομητές τυχαίου δάσους

Ένα παράδειγμα παρόμοιας συμπεριφοράς είναι οι επενδύσεις με χαμηλούς συσχετισμούς (όπως μετοχές και ομόλογα) όπου ενώνονται για να σχηματίσουν ένα χαρτοφυλάκιο που είναι μεγαλύτερο από το άθροισμα των μερών του, τα μη συσχετισμένα μοντέλα μπορούν να παράγουν σύνολα προβλέψεων που είναι πιο ακριβείς από οποιαδήποτε από τις μεμονωμένες προβλέψεις. Αυτό συμβαίνει διότι τα δέντρα προστατεύουν το ένα το άλλο από τα ατομικά τους λάθη (αρκεί να μην έχουν πάντα λάθος στην ίδια κατεύθυνση). Ενώ ορισμένα δέντρα μπορεί να είναι λάθος, πολλά άλλα δέντρα θα είναι σωστά, έτσι ως ομάδα τα δέντρα μπορούν να κινηθούν προς τη σωστή κατεύθυνση. Επομένως, οι προϋποθέσεις για την καλή απόδοση του τυχαίου δάσους είναι:

- Πρέπει να υπάρχει κάποιο πραγματικό σημείο στα χαρακτηριστικά (features) μας, έτσι ώστε τα μοντέλα που κατασκευάζονται χρησιμοποιώντας τα να είναι καλύτερα από την τυχαία εικασία.
- Οι προβλέψεις (και συνεπώς τα λάθη) που έγιναν από τα μεμονωμένα δέντρα πρέπει να έχουν χαμηλούς συσχετισμούς μεταξύ τους.

Για να διασφαλιστεί ότι η συμπεριφορά κάθε μεμονωμένου δένδρου δεν σχετίζεται με τη συμπεριφορά οποιουδήποτε άλλου δέντρου μέσα στο τυχαίο δάσος (random forest), χρησιμοποιούνται οι παρακάτω μέθοδοι:

- Bagging (Bootstrap Aggregation) - Τα δέντρα αποφάσεων είναι πολύ ευαίσθητα στα δεδομένα στα οποία εκπαιδεύονται - μικρές αλλαγές στο σύνολο εκπαίδευσης μπορεί να οδηγήσουν σε σημαντικά διαφορετικές δομές δέντρων. Το τυχαίο δάσος το εκμεταλλεύεται αυτό επιτρέποντας σε κάθε μεμονωμένο δέντρο να δειγματοληπτεί τυχαία από το σύνολο δεδομένων με αντικατάσταση, με αποτέλεσμα διαφορετικά δέντρα. Αυτή η διαδικασία είναι γνωστή ως bagging. Με το bagging δεν υποκαθιστούμε τα δεδομένα εκπαίδευσης σε μικρότερα κομμάτια και εκπαιδεύουμε κάθε δέντρο σε διαφορετικό κομμάτι. Αντίθετα, εάν έχουμε ένα δείγμα μεγέθους N, εξακολουθούμε να τροφοδοτούμε σε κάθε δέντρο ένα σετ κατάρτισης μεγέθους N (εκτός αν ορίζεται διαφορετικά). Αλλά αντί για τα αρχικά δεδομένα εκπαίδευσης, παίρνουμε ένα τυχαίο δείγμα μεγέθους N με αντικατάσταση. Για παράδειγμα, εάν τα δεδομένα εκπαίδευσης ήταν [1, 2, 3, 4, 5, 6], τότε θα μπορούσαμε να δώσουμε σε ένα από τα δέντρα μας την ακόλουθη λίστα [1, 2, 2, 3, 6, 6]. Παρατηρήστε ότι και οι δύο λίστες έχουν μήκος έξι και ότι τα "2" και "6" επαναλαμβάνονται και τα δύο στα τυχαία επιλεγμένα δεδομένα εκπαίδευσης που δίνουμε στο δέντρο μας (επειδή κάνουμε δείγματα με αντικατάσταση).
- Τυχαιότητα χαρακτηριστικών - Σε ένα κανονικό δέντρο αποφάσεων, όταν είναι ώρα να χωρίσουμε έναν κόμβο, εξετάζουμε κάθε πιθανό χαρακτηριστικό και επιλέγουμε αυτό που παράγει τον περισσότερο διαχωρισμό μεταξύ των παρατηρήσεων στον αριστερό κόμβο έναντι εκείνων του δεξιού κόμβου. Αντίθετα, κάθε δέντρο σε ένα τυχαίο δάσος μπορεί να επιλέξει μόνο από ένα τυχαίο υποσύνολο χαρακτηριστικών. Αυτό επιβάλλει ακόμη μεγαλύτερη διακύμανση μεταξύ των δέντρων στο μοντέλο και τελικά οδηγεί σε χαμηλότερη συσχέτιση μεταξύ των δέντρων και περισσότερη διαφοροποίηση.

4.2 ΑΛΓΟΡΙΘΜΟΙ ΠΑΛΙΝΔΡΟΜΗΣΗΣ (REGRESSION)

Η παλινδρόμηση είναι μια στατιστική μέθοδος που χρησιμοποιείται για την εκτίμηση της σχέσης μεταξύ μιας εξαρτημένης μεταβλητής (ετικέτα) και μιας ή περισσότερων ανεξάρτητων μεταβλητών (χαρακτηριστικά). Οι δύο βασικοί τύποι παλινδρόμησης είναι η απλή γραμμική παλινδρόμηση και η πολλαπλή γραμμική παλινδρόμηση, αν και υπάρχουν μέθοδοι μη γραμμικής παλινδρόμησης για πιο περίπλοκα δεδομένα και ανάλυση. Η απλή γραμμική παλινδρόμηση χρησιμοποιεί μια ανεξάρτητη μεταβλητή για να εξηγήσει ή να προβλέψει το αποτέλεσμα της εξαρτημένης μεταβλητής (ετικέτα), ενώ η πολλαπλή γραμμική παλινδρόμηση χρησιμοποιεί δύο ή περισσότερες ανεξάρτητες μεταβλητές για να προβλέψει το αποτέλεσμα. Παραδείγματα χρήσης της παλινδρόμησης είναι οι επαγγελματίες επενδύσεων, κλάδοι οικονομικών, πρόβλεψεις πωλήσεων και ακόμα και την αύξηση του ΑΕΠ μιας χώρας.

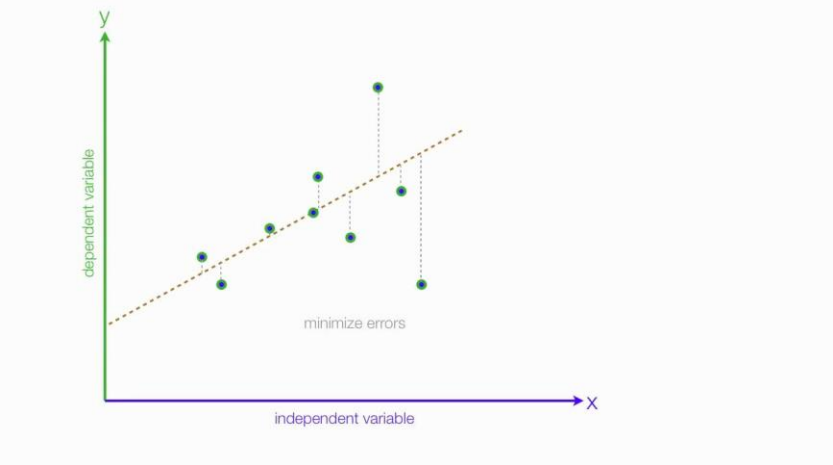
4.2.1 ΓΡΑΜΜΙΚΗ ΠΑΛΙΝΔΡΟΜΗΣΗ (LINEAR REGRESSION)

Η γραμμική παλινδρόμηση αναπτύχθηκε στον τομέα της στατιστικής και μελετήθηκε ως μοντέλο για την κατανόηση της σχέσης μεταξύ των αριθμητικών μεταβλητών εισόδου και εξόδου, αλλά χρησιμοποιείται και στη μηχανική μάθηση. Η γραμμική παλινδρόμηση είναι ένα γραμμικό μοντέλο, π.χ. ένα μοντέλο που αναλαμβάνει μια γραμμική σχέση μεταξύ των μεταβλητών εισόδου (x) και της μεταβλητής μεμονωμένης εξόδου (y). Πιο συγκεκριμένα, αυτό το y μπορεί να υπολογιστεί από έναν γραμμικό συνδυασμό των μεταβλητών εισόδου (x). Όταν υπάρχει μία μεταβλητή εισόδου (x), η μέθοδος αναφέρεται ως απλή γραμμική παλινδρόμηση. Όταν υπάρχουν πολλές μεταβλητές εισόδου τότε η μέθοδος λέγεται πολλαπλή γραμμική παλινδρόμηση. Μπορούν να χρησιμοποιηθούν διαφορετικές τεχνικές για την προετοιμασία ή την εκπαίδευση της εξίσωσης γραμμικής παλινδρόμησης από τα δεδομένα, η πιο συνηθισμένη ονομάζεται κανονικά τετραγωνικά ελάχιστα (Ordinary Least Squares). Είναι συνηθισμένο λοιπόν να αναφέρεται σε ένα μοντέλο που έχει φτιαχτεί με αυτόν τον τρόπο ως Γραμμική Παλινδρόμηση Κανονικών Τετραγώνων ή απλώς Παλινδρόμηση Λιγότερων Τετραγώνων. Τόσο οι τιμές εισόδου όσο και εξόδου είναι αριθμητικές.

Η γραμμική εξίσωση δίνει έναν συντελεστή κλίμακας σε κάθε τιμή εισόδου ή στήλη, που ονομάζεται συντελεστής (coefficient) και αντιπροσωπεύεται από το γράμμα Β. Προστίθεται επίσης ένας ακόμα συντελεστής, δίνοντας στη γραμμή έναν επιπλέον βαθμό ελευθερίας (π.χ. κίνηση προς τα πάνω και προς τα κάτω σε μια δισδιάστατη γραφική παράσταση) και συχνά αποκαλείται τομή ή συντελεστής μεροληψίας (bias coefficient). Για παράδειγμα, σε ένα απλό πρόβλημα παλινδρόμησης (ένα x και ένα y), η μορφή του μοντέλου θα ήταν:

$$y = B0 + B1 * x$$

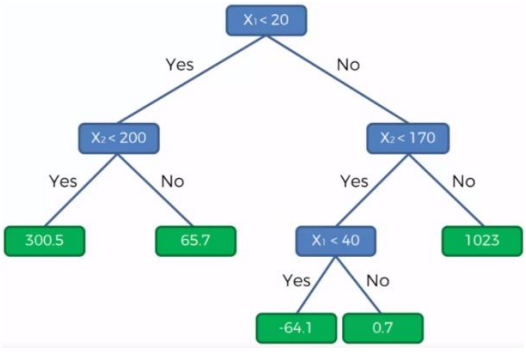
Σε υψηλότερες διαστάσεις όταν έχουμε περισσότερες από μία εισόδους (x), η γραμμή ονομάζεται επίπεδο ή υπερ-επίπεδο. Η αναπαράσταση λοιπόν είναι η μορφή της εξίσωσης και των ειδικών τιμών που χρησιμοποιούνται για τους συντελεστές (π.χ. B0 και B1 στο παραπάνω παράδειγμα). Συνήθως η πολυπλοκότητα ενός μοντέλου γραμμικής παλινδρόμησης καθορίζεται από τον αριθμό των ανεξάρτητων μεταβλητών Β, δηλαδή των συντελεστών που χρησιμοποιούνται στο μοντέλο. Όταν ένας συντελεστής μηδενίζεται, αφαιρεί αποτελεσματικά την επίδραση της μεταβλητής εισόδου στο μοντέλο και συνεπώς από την πρόβλεψη που έγινε από το μοντέλο (0 * x = 0). Κάποιες μέθοδοι κανονικοποίησης αλλάζουν τον αλγόριθμο εκμάθησης ώστε να μειώσουν την πολυπλοκότητα, με το να υπολογίζουν μόνο το απόλυτο μέρος των συντελεστών οδηγώντας έτσι κάποιους από αυτούς στο 0.



Εικόνα 18 Παράδειγμα Γραμμικής Παλινδρόμησης

4.2.2 ΠΑΛΙΝΔΡΟΜΗΣΗ ΜΕ ΔΕΝΤΡΑ ΑΠΟΦΑΣΗΣ (DECISION TREE REGRESSION)

Τα δένδρα απόφασης για ταξινόμηση, που προαναφέρθηκαν παραπάνω, και τα δένδρα απόφασης με παλινδρόμηση λειτουργούν το ίδιο με κύρια διαφορά τους ότι η ετικέτα (label) αυτή τη φορά για την μεταβλητή που θέλουμε να προβλέψουμε μπορεί να είναι συνεχής αριθμός (συνήθως πραγματικός). Για παράδειγμα, η τιμή ενός σπιτιού ή η παραμονή ενός ασθενή στην κλινική ενός νοσοκομείου.



Εικόνα 19 Παλινδρόμηση με Δένδρο απόφασης

4.2.3 GRADIENT BOOSTING TREE REGRESSOR

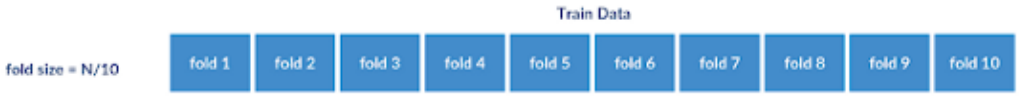
Ο gradient boosting tree regressor στη βασική του δομή λειτουργεί με τον ίδιο τρόπο όπως και ο gradient boosting tree classifier με την διαφορά τους να είναι ότι η μεταβλητή πρόβλεψης είναι συνεχής αριθμός.

4.3 STACKING ΚΑΙ ENSEMBLING ΣΕ ΑΛΓΟΡΙΘΜΟΥΣ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ

Στη στατιστική και τη μηχανική μάθηση, οι μέθοδοι ensemble χρησιμοποιούν πολλαπλούς αλγόριθμους μάθησης για να αποκτήσουν καλύτερη προγνωστική απόδοση από ότι θα μπορούσαν να έχουν από οποιονδήποτε από τους μεμονωμένους αλγόριθμους μάθησης. Σε αντίθεση με ένα στατιστικό ensemble της στατιστικής μηχανικής, το οποίο είναι συνήθως άπειρο, ένα σύνολο μηχανικής μάθησης αποτελείται από ένα συγκεκριμένο πεπερασμένο σύνολο εναλλακτικών μοντέλων, αλλά συνήθως επιτρέπει πολύ πιο ευέλικτη δομή μεταξύ αυτών των εναλλακτικών .

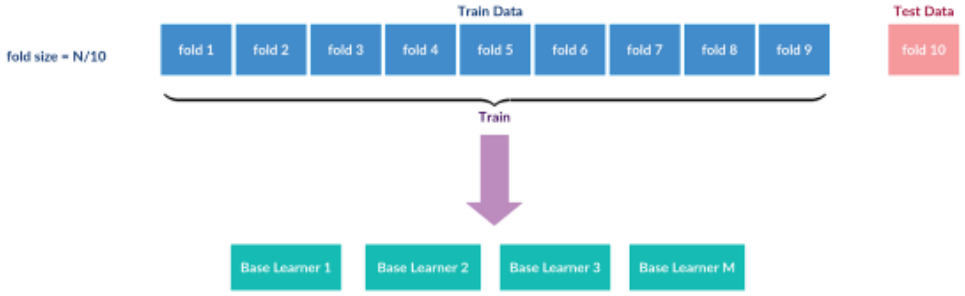
Το Stacking είναι ένας τρόπος συνδυασμού πολλαπλών ταξινομητών ή μοντέλων παλινδρόμησης. Υπάρχουν πολλοί τρόποι για το σύνολο των μοντέλων, τα ευρέως γνωστά μοντέλα είναι Bagging ή Boosting. Το Bagging επιτρέπει πολλά παρόμοια μοντέλα με υψηλή διακύμανση κατά μέσο όρο για μείωση της απόκλισης. Το boosting δημιουργεί πολλαπλά στοιχειώδη μοντέλα για τη μείωση της προκατάληψης, διατηρώντας παράλληλα τη διακύμανση μικρή. Η γενική ιδέα του stacking είναι να εκπαιδευτούν διάφορα μοντέλα, συνήθως με διαφορετικούς τύπους αλγορίθμων (γνωστός και ως βασικος μαθητής ή base learner), στα δεδομένα εκπαίδευσης και, στη συνέχεια, αντί να επιλέξουμε το καλύτερο μοντέλο, όλα τα μοντέλα συγκεντρώνονται χρησιμοποιώντας ένα άλλο μοντέλο τον μετα-μαθητή (meta learner), για να κάνει την τελική πρόβλεψη. Οι είσοδοι για τον μετα-μαθητή είναι τα αποτελέσματα πρόβλεψης των βασικών μαθητών.

Η εκπαίδευση μοντέλων stacking μπορεί να γίνει περίπλοκη. Αυτό που χρειάζεται είναι μερικά βήματα για την εκπαίδευση παρόμοια με εκείνα του k-fold cross-validation. Αρχικά, χωρίζονται τα δεδομένα σε εκείνα εκπαίδευσης (train) και ελέγχου (test). Το σύνολο των δεδομένων για έλεγχο δεν χρησιμοποιείται κατά τη διάρκεια της εκπαίδευσης των stacking μοντέλων. Στη συνέχεια χωρίζεται το σύνολο των δεδομένων εκπαίδευσης σε k-folds. Αν υποθέσουμε ότι το αρχικό σύνολο δεδομένων έχει N παρατηρήσεις, τότε κάθε fold θα έχει N/k παρατηρήσεις αν και δεν είναι απαραίτητο να είναι διασπασμένα σε ίσο αριθμό το κάθε fold. Υποθέτουμε k=10:



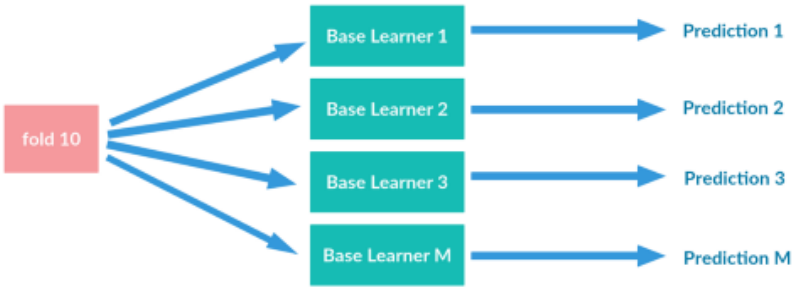
Εικόνα 20 Αρχικός διαχωρισμός δεδομένων

Κρατιέται ένα από τα folds στην άκρη και εκπαιδεύεται ο βασικός μαθητής με τα υπόλοιπα folds. Το εναπομείναν fold χρησιμοποιείται για τον έλεγχο (testing) των δεδομένων σε αυτό το βήμα.



Εικόνα 21 Εκπαίδευση των βασικών μαθητών

Τότε γίνεται η πρόβλεψη πάνω στο τελευταίο fold χρησιμοποιώντας τα μοντέλα M που έχουν εκπαιδευτεί. Έτσι θα προκύψουν M προβλέψεις για κάθε παρατήρηση στο τελευταίο fold. Τώρα έχουμε N/10 σύνολα παρατηρήσεων από προβλέψεις, όπως φαίνεται παρακάτω:



Εικόνα 22 Πρόβλεψεις από τους βασικούς μαθητές

Και έτσι επαναλαμβάνουμε την παραπάνω διαδικασία αλλάζωντας το fold που αφήνουμε στην άκρη από το 1 έως το 10. Στο τέλος των επαναλήψεων, θα έχουμε N σύνολα προβλέψεων που αντιστοιχούν σε κάθε παρατήρηση του αρχικού συνόλου δεδομένων μαζί με την πραγματική τιμή τους.

Data Point #	prediction from base learner 1	prediction from base learner 2	prediction from base learner 3	prediction from base learner M	actual
1	$\hat{y}_{11}$	$\hat{y}_{12}$	$\hat{y}_{13}$	$\hat{y}_{1M}$	y_1
2	$\hat{y}_{21}$	$\hat{y}_{22}$	$\hat{y}_{23}$	$\hat{y}_{2M}$	y_2
...	...	...	...	...	...
N	$\hat{y}_{N1}$	$\hat{y}_{N2}$	$\hat{y}_{N3}$	$\hat{y}_{NM}$	y_N

Εικόνα 23 Προετοιμασία εκπαίδευσης μέτα-μαθητή

Αυτή θα είναι η είσοδος από παρατηρήσεις για τον μετα-μαθητή (meta-learner) με τις οποίες θα εκπαιδευτεί. Τέλος, για να κάνει προβλέψεις ακολουθείται η παραπάνω διαδικασία όμως αυτή τη φορά χωρίς τα k-folds. Δηλαδή, όλοι οι βασικοί μαθητές (base learners) κάνουν προβλέψεις στα καινούρια δεδομένα και περνάν τα αποτελέσματα τους στον μετα-μαθητή για να κάνει την τελική πρόβλεψη.

4.4 ΤΕΧΝΙΚΕΣ ΜΕΤΑΜΟΡΦΩΣΗΣ ΔΕΔΟΜΕΝΩΝ

Κλιμάκωση των δεδομένων (scaling) γενικά σημαίνει αλλαγή του εύρους των τιμών ενώ δεν αλλάζει την κατανομή (distribution) αυτών. Το εύρος ρυθμίζεται συχνά απο 0 έως 1.

Τυποποίηση (standardization) των δεδομένων αναφέρεται στην αλλαγή έτσι ώστε η τυπική απόκλιση της κατανομής από τον μέσο να ισούται με 1. Το αποτέλεσμα είναι κάτι πολύ κοντά στην κανονική κατανομή.

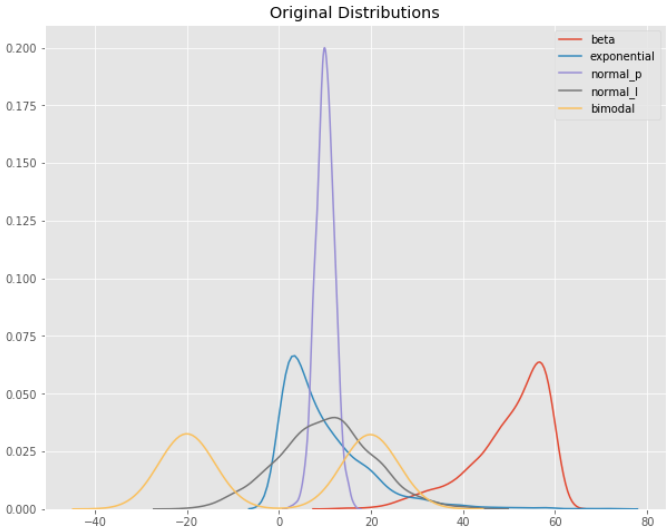
Κανονικοποίηση (normalization) των δεδομένων σημαίνει προσαρμογή των τιμών που χρησιμοποιούνται σε διαφορετικές κλίμακες, σε μία κοινή κλίμακα. Σε πιο περίπλοκες περιπτώσεις η κανονικοποίηση ενσωματώνει ολόκληρη την κατανομή πιθανότητας των προσαρμοσμένων τιμών.

Πολλοί αλγόριθμοι μηχανικής μάθησης αποδίδουν καλύτερα ή συγκλίνουν γρηγορότερα όταν οι λειτουργίες βρίσκονται σε σχετικά παρόμοια κλίμακα ή / και ακολουθούν την κανονική κατανομή. Παραδείγματα τέτοιων οικογενειών αλγορίθμων περιλαμβάνουν: γραμμική και λογιστική παλινδρόμηση, κοντινότερους γείτονες (nearest neighbors), νευρωνικά δίκτυα, υποστήριξη μηχανών διανυσμάτων (support vector machines), ανάλυση βασικών συστατικών (principal components analysis), γραμμική ανάλυση διακρίσεων (linear discriminant analysis). Η κλιμάκωση και η τυποποίηση μπορούν να βοηθήσουν τα χαρακτηριστικά να φτάσουν σε πιο εύπεπτη μορφή για αυτούς τους αλγόριθμους.

Στο παρακάτω παράδειγμα έχουμε 4 κατανομές με διαφορετικά χαρακτηριστικά.

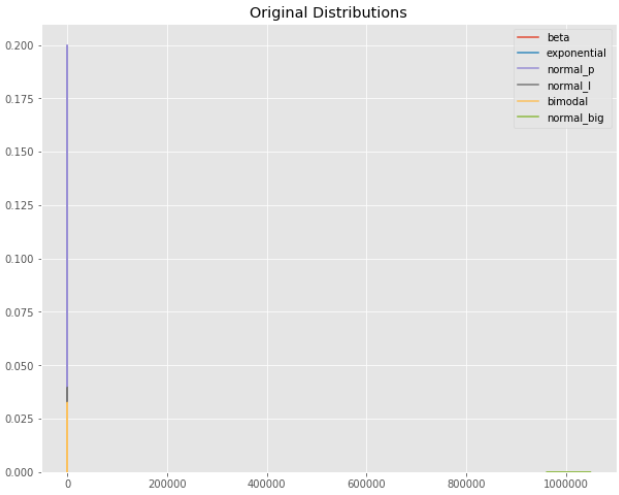
- Beta - με αρνητική κλίση
- Exponential (εκθετική) – με θετική κλίση
- Normal_p - κανονική, υψηλή κυρτότητα
- Normal_l - κανονική, χαμηλή κυρτότητα
- Bimodal – διτροπική

Οι τιμές είναι σχετικά στην ίδια κλίμακα όπως φαίνεται παρακάτω:



Εικόνα 24 Αρχικές Κατανομές

Στη συνέχεια προσθέτουμε μία πέμπτη κατανομή με πολύ μεγαλύτερες τιμές την normal_big η οποία ακολουθεί την κανονική κατανομή και το παραπάνω σχήμα μετατρέπεται σε:



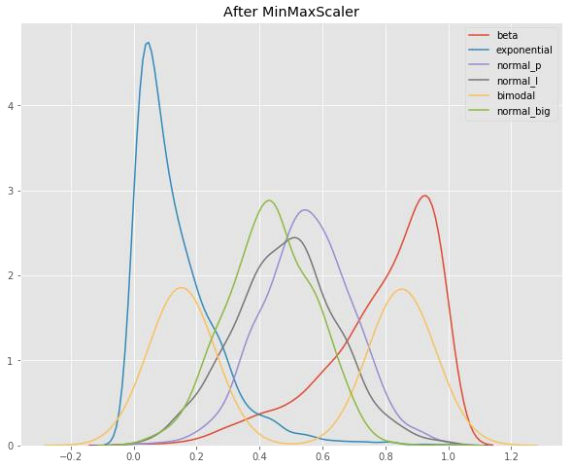
Εικόνα 25 Προσθήκη κατανομής με μεγάλες τιμές normal_big

Παρακάτω θα δούμε την επίδραση που έχουν οι προαναφερθείς τεχνικές στις κατανομές.

4.4.1 MIN-MAX-SCALER

Για κάθε χαρακτηριστικό (feature), ο MinMaxScaler αφαιρεί την ελάχιστη τιμή των χαρακτηριστικών και μετά διαιρεί με το εύρος. Το εύρος είναι η διαφορά μεταξύ της μέγιστης και ελάχιστης τιμής των χαρακτηριστικών. Ο MinMaxScaler διατηρεί το σχήμα της αρχικής διανομής. Δεν αλλάζει ουσιαστικά τις πληροφορίες που είναι ενσωματωμένες στα αρχικά δεδομένα. Επιπλέον, ο MinMaxScaler δεν μειώνει τη σημασία των ακραίων τιμών (outlier). Το

προεπιλεγμένο εύρος για τη δυνατότητα που επιστρέφεται από το MinMaxScaler είναι 0 έως 1. Το παραπάνω σχήμα μετά την εφαρμογή του MinMaxScaler γίνεται:

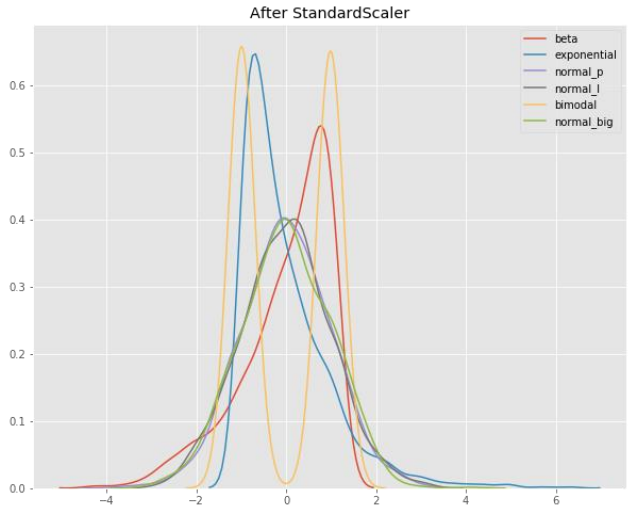


Εικόνα 26 Μετά την εφαρμογή MinMaxScaler

4.4.2 STANDARD SCALER

Ο StandardScaler τυποποιεί ένα χαρακτηριστικό αφαιρώντας το μέσο όρο και, στη συνέχεια, κλιμακώνοντας τη διακύμανση μονάδας. Διακύμανση μονάδας σημαίνει διαίρεση όλων των τιμών με την τυπική απόκλιση. Ο StandardScaler οδηγεί σε κατανομή με τυπική απόκλιση ίση με 1. Η διακύμανση ισούται επίσης με 1. Το StandardScaler κάνει το μέσο όρο της κατανομής 0. Περίπου το 68%, στο παράδειγμά μας, των τιμών θα είναι μεταξύ -1 και 1.

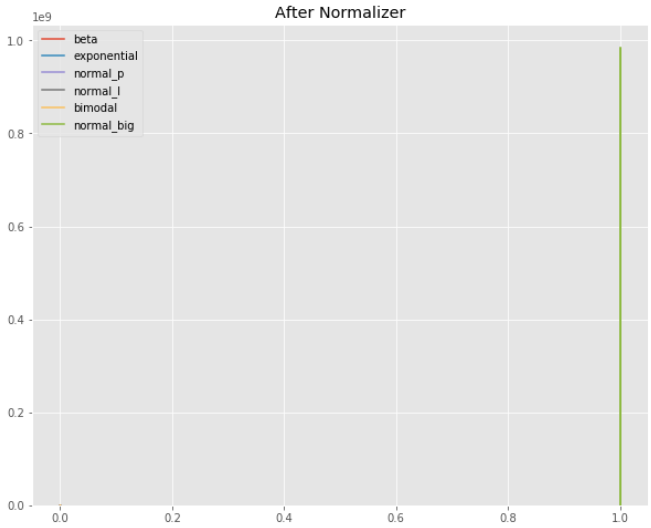
Στο παρακάτω σχήμα φαίνεται ότι όλες οι κατανομές έχουν μέσο κοντά στο μηδέν και μοναδιαία διακύμανση. Οι τιμές είναι σε παρόμοια κλίμακα αλλά το εύρος είναι μεγαλύτερο από εκείνο του MinMaxScaler.



Εικόνα 27 Μετά την εφαρμογή StandardScaler

4.4.3 NORMALIZER

Ο κανονικοποιητής λειτουργεί στις σειρές και όχι στις στήλες των δεδομένων. Εξ' ορισμού, η κανονικοποίηση L2 εφαρμόζεται σε κάθε παρατήρηση έτσι ώστε οι τιμές σε μια σειρά να έχουν έναν κανόνα μονάδας. Κανόνας μονάδας με L2 ορίζεται ως εάν κάθε στοιχείο τετραγωνιστεί και αθροιστεί τότε το σύνολο θα ισούται με 1. Εναλλακτικά η κανονικοποίηση L1 (του ταξιτζή ή Manhattan), μπορεί να εφαρμοστεί αντί της L2. Ο κανονικοποιητής μετατρέπει όλες τις τιμές μεταξύ -1 και 1. Στο συγκεκριμένο παράδειγμα, το normal_big καταλήγει με όλες τις τιμές σε 0.99999. Παρακάτω είναι το σχήμα μετά από τον κανονικοποιητή.



Εικόνα 28 Μετά την εφαρμογή Normalizer

4.5 ΣΥΣΧΕΤΙΣΗ (CORRELATION)

Η συσχέτιση (correlation) είναι ο βαθμός στον οποίο δύο μεταβλητές σχετίζονται γραμμικά. Ο συντελεστής συσχέτισης περιγράφει τόσο τη δύναμη όσο και την κατεύθυνση της σχέσης. Παρακάτω θα αναφερθούμε σε δύο μεθόδους ανάλυσης συσχέτισης: συσχετίσεις Spearman και Pearson.

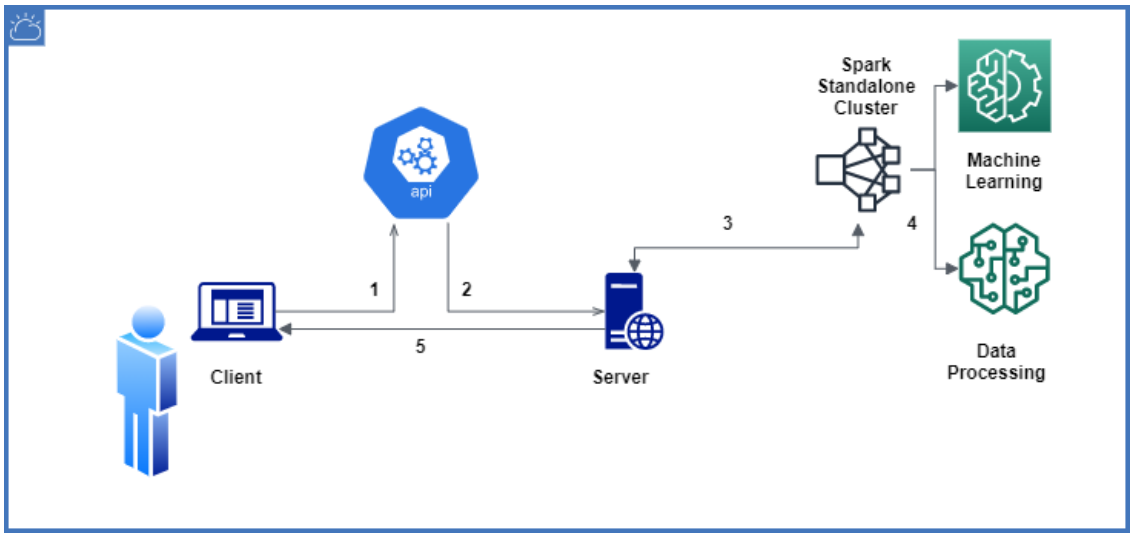
Η συσχέτιση Pearson αξιολογεί τη γραμμική σχέση μεταξύ δύο συνεχών μεταβλητών. Μια σχέση είναι γραμμική όταν μια αλλαγή σε μια μεταβλητή σχετίζεται με μια αναλογική αλλαγή στην άλλη μεταβλητή. Για παράδειγμα, μπορείτε να χρησιμοποιήσετε μια συσχέτιση Pearson για να αξιολογήσετε εάν οι αυξήσεις της θερμοκρασίας στην εγκατάσταση παραγωγής σχετίζονται με τη μείωση του πάχους της επικάλυψης σοκολάτας.

Η συσχέτιση Spearman αξιολογεί τη σχέση μονοτονίας μεταξύ δύο συνεχών ή κατηγορικών μεταβλητών. Σε μια μονοτονική σχέση, οι μεταβλητές τείνουν να αλλάζουν μαζί, αλλά όχι απαραίτητα με σταθερό ρυθμό. Ο συντελεστής συσχέτισης Spearman βασίζεται στις τιμές κατάταξης για κάθε μεταβλητή και όχι στα μη επεξεργασμένα δεδομένα. Για παράδειγμα, για την αξιολόγηση μιας σειράς με την οποία οι εργαζόμενοι ολοκληρώνουν μια δοκιμαστική άσκηση σχετίζεται με τον αριθμό των μηνών που απασχολούνται.

5. ΠΛΑΤΦΟΡΜΑ ΔΥΝΑΜΙΚΗΣ ΕΚΤΕΛΕΣΗΣ ΑΛΓΟΡΙΘΜΩΝ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ

Σε αυτό το κεφάλαιο θα ασχοληθούμε με την περιγραφή της πλατφόρμας αλλά και τη συνολική αρχιτεκτονική της. Αρχικά παρατίθεται μία υψηλού επιπέδου (high-level) μορφή της αρχιτεκτονικής και εξηγούνται τα βήματα που συμβαίνουν σε κάθε στάδιο. Στη συνέχεια περιγράφεται η κάθε σελίδα της πλατφόρμας αλλά και η λειτουργία της. Στόχος της πλατφόρμας είναι να παρέξει στους χρήστες τη δυνατότητα να χειριστούν τα δεδομένα τους σε μεγάλη κλίμακα με ιδιαίτερη ευκολία και να επικεντρωθούν στην αναζήτηση αποτελεσμάτων από αυτά μέσω της εξόρυξης και της εφαρμογής αλγορίθμων μηχανικής μάθησης.

Η αρχιτεκτονική της πλατφόρμας φαίνεται παρακάτω:

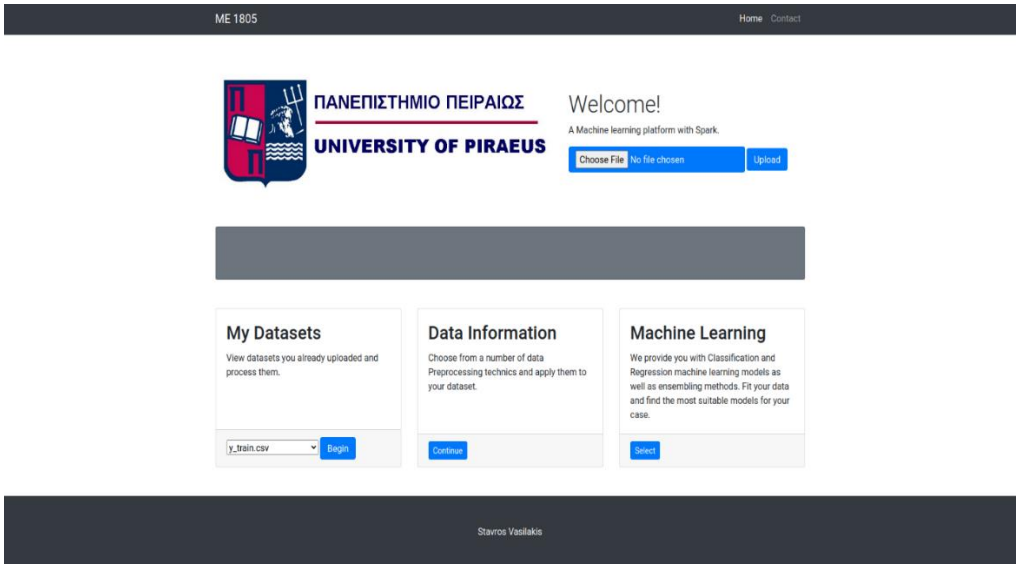


Εικόνα 29 Αρχιτεκτονική Πλατφόρμας

1. Αρχικά ο χρήστης κάνει API κλήση στον server μέσω του client. Για το front-end έχει χρησιμοποιηθεί JavaScript, CSS, HTML, Bootstrap και jQuery.
2. Στη συνέχεια δέχεται το request στο server-side (ή back-end). Για το back-end έχει χρησιμοποιηθεί το framework Flask μαζί με Jinja.
3. Στο επόμενο βήμα το back-end βάσει των επιλογών του χρήστη δημιουργεί ένα Spark Session στο οποίο τρέχει τα απαιτούμενα jobs. Για το Apache Spark έχει γίνει η χρήση της γλώσσας προγραμματισμού Python (PySpark).
4. Αφού δεχθεί τα jobs τα οποία είναι να τρέξουν, βάσει επιλογών του χρήστη, γυρνάει τα αποτελέσματα αυτών πάλι στο back-end. Βασικές βιβλιοθήκες που χρησιμοποιήθηκαν είναι οι *pyspark.ml.classification, pyspark.ml.regression, pyspark.sql, pyspark.ml.feature, pyspark.ml.metrics* και *pypsark.ml.evaluation.*

5. Με τη σειρά του το back-end εμφανίζει τα αποτελέσματα του κάθε request του χρήστη στο UI (User Interface).

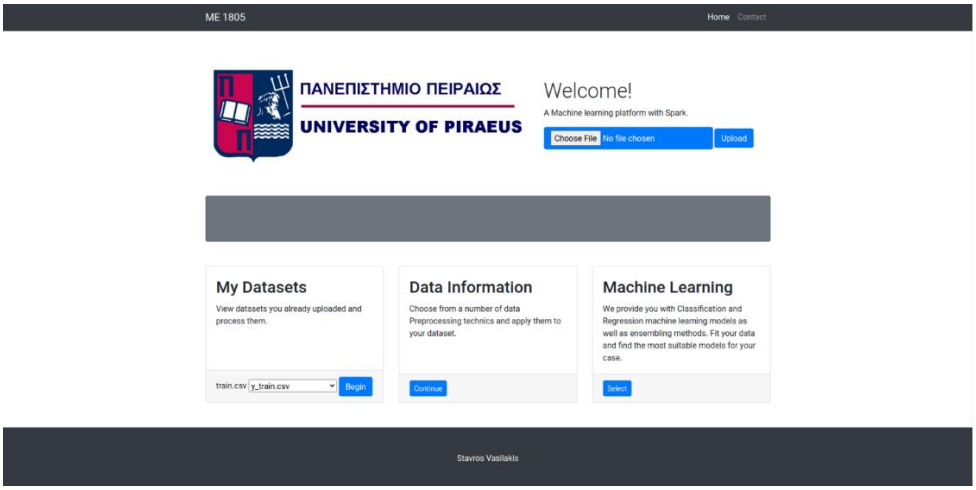
Η πλατφόρμα προορίζεται για εύκολη διαχείριση των δεδομένων καθώς και εφαρμογή αλγορίθμων μηχανικής μάθησης από τον χρήστη. Αρχικά ο χρήστης ανεβάζει το σύνολο δεδομένων (dataset), σε μορφή csv (προς το παρών υποστηρίζεται μόνο η μορφή csv), και αφού προστεθεί στο σύστημα αρχείων (file system) έχει την επιλογή διάφορων ενεργειών για να συνεχίσει.



Εικόνα 30 Αρχική Σελίδα

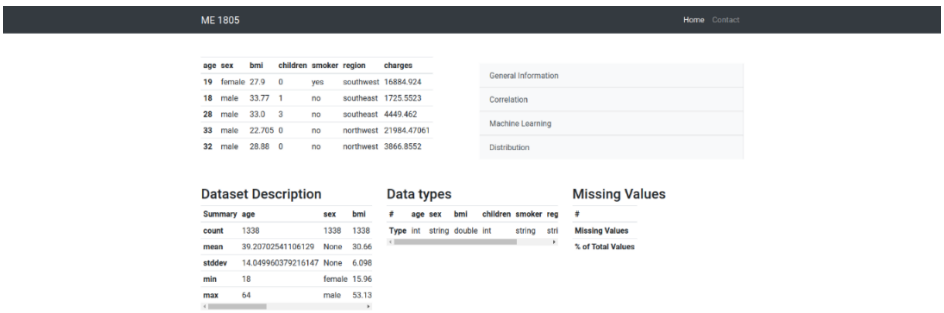
Επιλέγοντας *Choose File, ο χρήστης,* ανοίγει η περιήγηση στο σύστημα των αρχείων και αφού επιλεγεί το επιθυμητό σύνολο δεδομένων κάνει *Upload.* Όταν τελειώσει η διαδικασία θα πρέπει να μπορεί να δει το όνομα του αρχείου .csv στην λίστα των *My Datasets.* Στη συνέχεια, επιλέγει το αρχείο από την λίστα των My Datasets και πατάει *Begin* για να φορτώσει το σύνολο δεδομένων στη μνήμη, δηλαδή να διαβαστεί από το Spark. Καθώς διαβάζει τα δεδομένα η πλατφόρμα υποθέτει ότι η πρώτη γραμμή του αρχείου csv υποδεικνύει το όνομα της κάθε στήλης και διώχνει τις γραμμές οι οποίες δεν ακολουθούν το schema της στήλης (όπως ορίζεται μέσω της επιλογής *“inferSchema” = true*) μέσω της επιλογής *“DROPMALFORMED”*.

Αφού γίνει αυτό θα φανεί το όνομα του συνόλου δεδομένων δίπλα από την λίστα αρχείων, όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 31 Φόρτωση συνόλου δεδομένων

Στο επόμενο βήμα ο χρήστης πατάει *Continue* από το *Data Information* κουτί για να δει γενικές πληροφορίες όπως, απόσπασμα 5 σειρών των δεδομένων, τον τύπο των δεδομένων της κάθε στήλης π.χ *int* για ακέραιο, *double* για πραγματικό και *string* για κείμενο/λέξεις κ.α, την περιγραφή του συνόλου δεδομένων με στοιχεία όπως τον αριθμό των γραμμών, την μέγιστη και ελάχιστη τιμή, την μέση τιμή και την τυπική απόκλιση, και τέλος τον αριθμό των τιμών που λείπουν από την κάθε στήλη, αν λείπουν, και το ποσοστό αυτών επί του συνολικού αριθμού των γραμμών που περιέχει η στήλη.

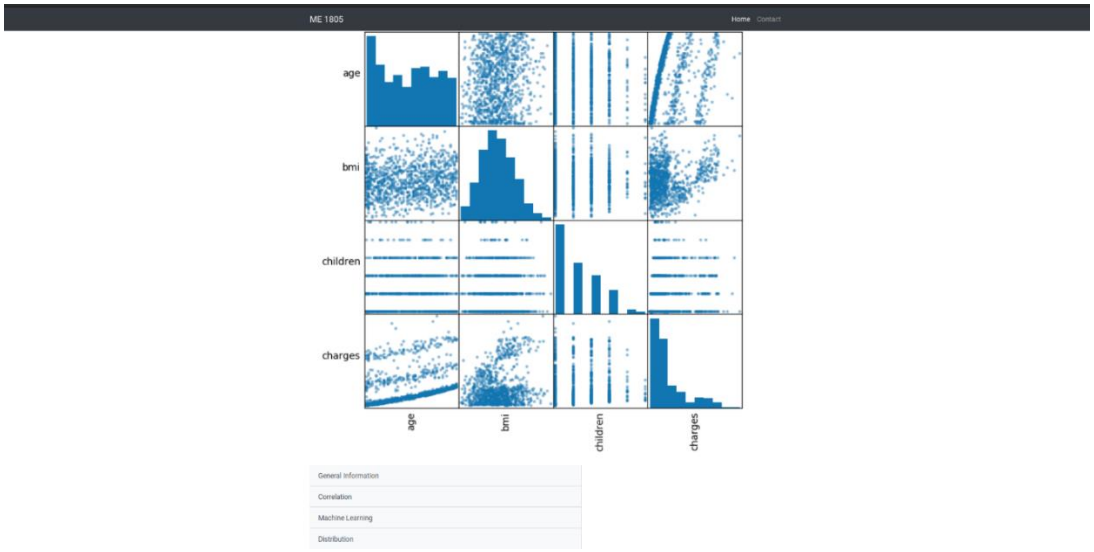


Εικόνα 32 Γενικές Πληροφορίες για τα δεδομένα

Επόμενη επιλογή που δίνεται στον χρήστη είναι να δει τη συσχέτιση (correlation) μεταξύ των στηλών με τον συντελεστή συσχέτισης Pearson. Σε αρχικό στάδιο είναι διαθέσιμος ο συντελεστής συσχέτισης Pearson και σαν επόμενο βήμα θα προστεθεί και η συσχέτιση κατάταξης Spearman.

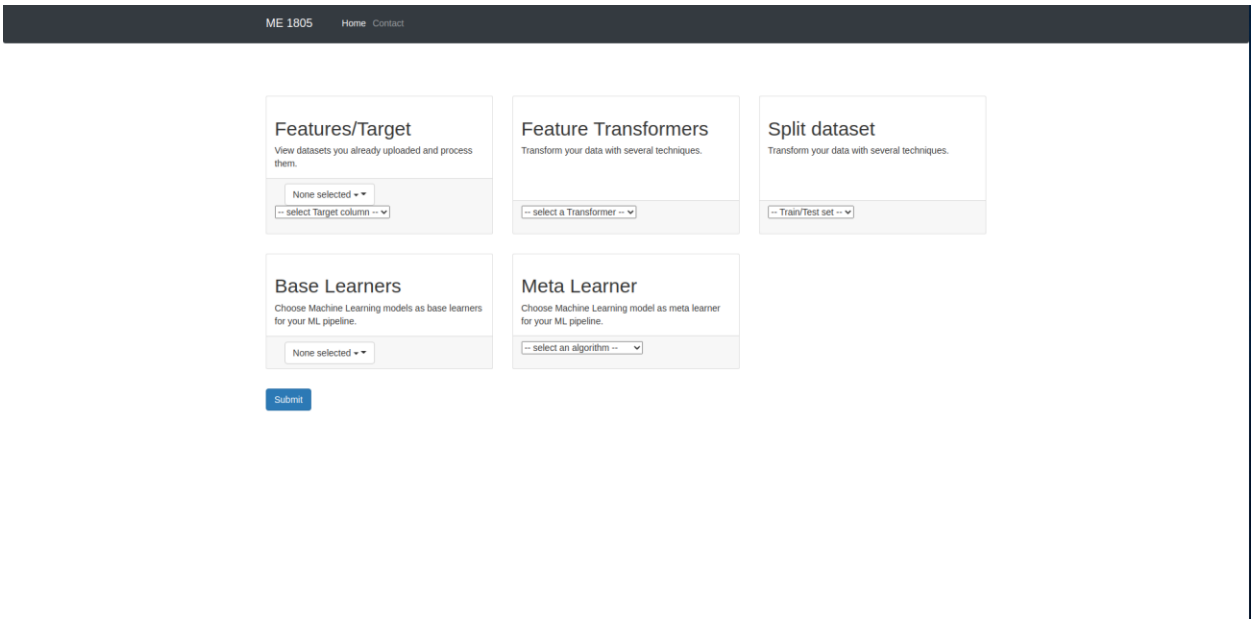
Εικόνα 33 Pearson's συσχέτιση για τα δεδομένα

Μία ακόμη δυνατότητα που έχει ο χρήστης είναι να δει τις κατανομές που μπορεί να ακολουθεί η κάθε στήλη από το σύνολο δεδομένων του καθώς και τη δομή σε σύγκριση με άλλες στήλες.



Εικόνα 34 Κατανομή της κάθε στήλης του συνόλου δεδομένων

Αφού ο χρήστης πάρει τις πληροφορίες που χρειάζεται για τα δεδομένα του συνεχίζει με το τελικό στάδιο το οποίο είναι η προετοιμασία των δεδομένων για τον αλγόριθμο μηχανικής μάθησης ή το stack από βασικούς μαθητές (base-learners) και στη συνέχεια ο μέτα-μαθητής (meta-learner) και η εφαρμογή αυτών. Η σελίδα του Machine Learning φαίνεται παρακάτω:

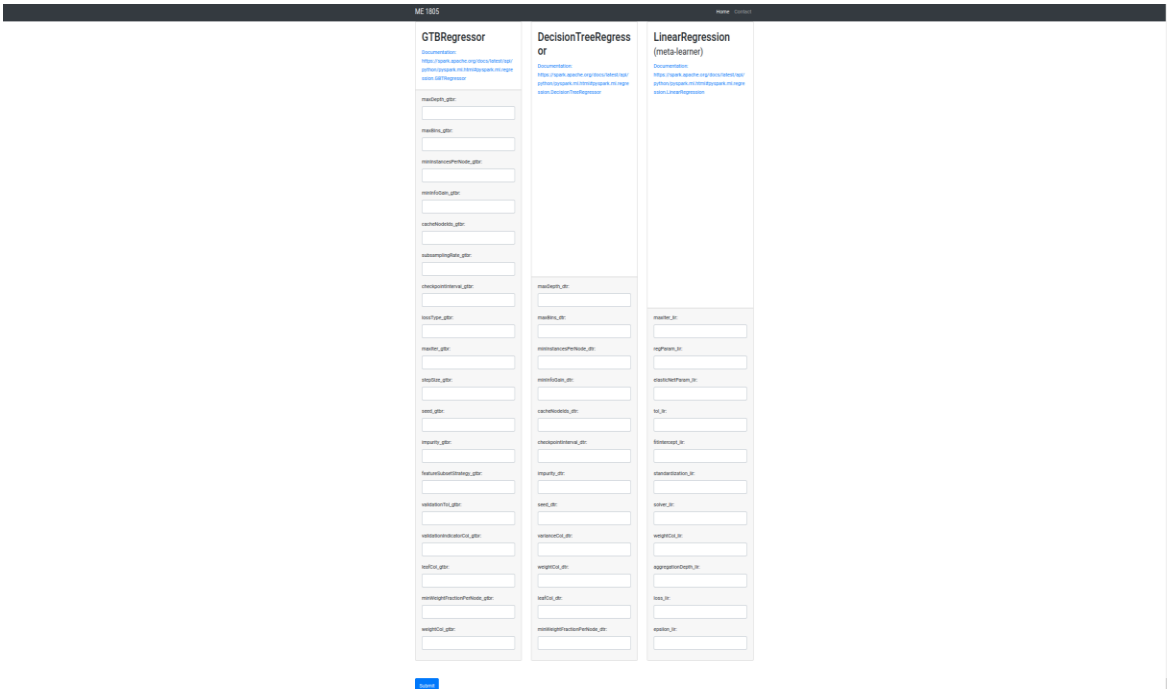


Εικόνα 35 Προετοιμασία και Επιλογή Αλγορίθμων Μηχανικής Μάθησης

Σε πρώτη φάση ο χρήστης ορίζει τις στήλες που θα αποτελέσουν τις ανεξάρτητες μεταβλητές του ή αλλιώς τα χαρακτηριστικά (features) και την εξαρτημένη μεταβλητή ή στόχο (target) από το κουτί *Features/Target*. Μετά, είναι στη ευχέρεια του χρήστη αν θα εφαρμόσει κάποια μετατροπή στα δεδομένα του με τις επιλογές να είναι κανονικοποιητής (*Normalizer*), *StandardScaler* και *MinMaxScaler,* στο *Feature Transformers*. Στο επόμενο στάδιο ορίζεται το ποσοστό στο οποίο θα διασπαστούν τα δεδομένα, μέσω του κουτιού *Split Dataset,* σε εκείνα του συνόλου εκπαίδευσης (train set) και ελέγχου (test set). Οι επιλογές που δίνονται είναι 70%/30%, 75%/25% και 80%/20%. Τέλος, στα κουτιά *Base Learners* και *Meta Learner* ο χρήστης μπορεί είτε να χρησιμοποιήσει έναν αλγόριθμο μηχανικής μάθησης είτε ένα stack από αλγορίθμους βασικούς-μαθητές και έναν μέτα-μαθητή. Οι αλγόριθμοι που είναι διαθέσιμοι είναι:

- Ταξινόμηση (Classification): Λογιστική Παλινδρόμηση (Logistic Regression), GBTClassifier (Gradient Boosting Trees) και Ταξινομητής τυχαίου δάσους (Random Forest Classifier).
- Παλινδρόμηση (Regression): Γραμμική Παλινδρόμηση (Linear Regression), GBTRegressor και παλινδρόμηση με δένδρα απόφασης (Decision Tree Regressor).

Αφού γίνει η επιλογή σε αυτό το στάδιο και επιλεγεί το *submit* τότε ο χρήστης καλείται να δηλώσει τις παραμέτρους για τον κάθε αλγόριθμο μηχανικής μάθησης που επέλεξε. Όπως φαίνεται παρακάτω:



Εικόνα 36 Επιλογή Παραμέτρων για τους Αλγορίθμους επιλογής

Παραπάνω φαίνεται ένα παράδειγμα εισαγωγής παραμέτρων για τους αλγορίθμους που επιλέχθηκαν. Επίσης δίνεται στον χρήστη ο σύνδεσμος με το documentation των αλγορίθμων για να μπορέσει να δει πληροφορίες σχετικά με αυτούς όπως τις default τιμές που δέχονται καθώς και τις επιλογές που έχει συνολικά για τις παραμέτρους. Αφού γίνει submit περνάει στο τελικό στάδιο το οποίο είναι η εκπαίδευση των αλγορίθμων που έχει επιλέξει και τα αποτελέσματα αυτών.

Στο τελικό στάδιο, ακολουθούνται οι παρακάτω ενέργειες μέχρι και την κατάληξη στο αποτέλεσμα:

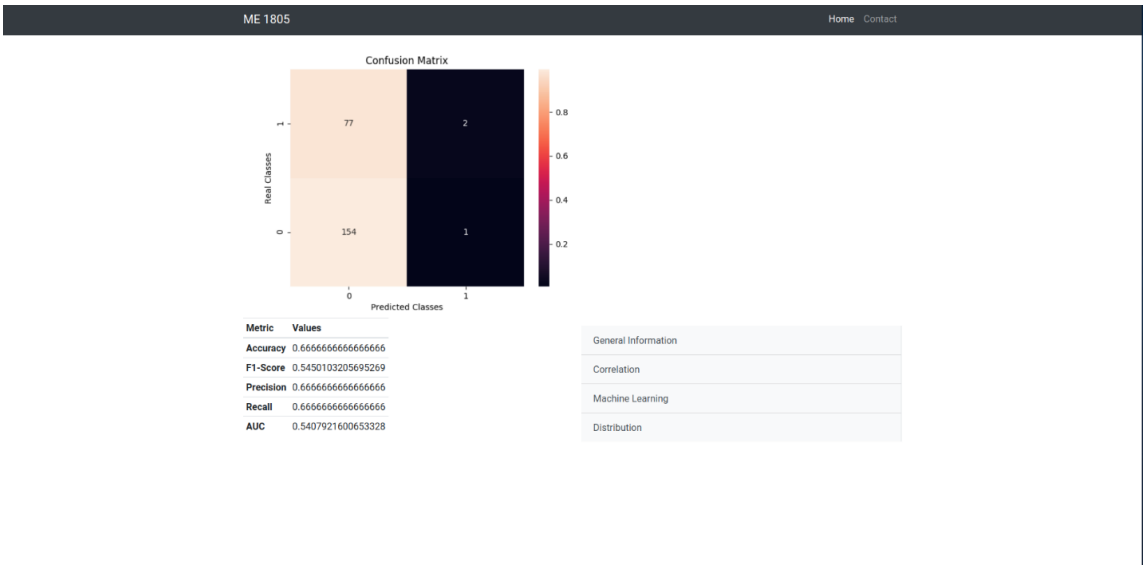
- Αρχικά μετατρέπεται ο τύπος της στήλης στόχου σε δυαδικό σε περίπτωση ταξινόμησης.
- Ελέγχονται οι τιμές που λείπουν στις εκάστοτες στήλες των χαρακτηριστικών και γίνεται η διευθέτιση με τον εξής τρόπο. Αν ο τύπος της μεταβλητής είναι αριθμητικός τότε η τιμή που λείπει αντικαθίσταται από τον μέσο όρο των τιμών της στήλης. Αν ο τύπος της μεταβλητής είναι κατηγορικός τότε η τιμή αντικαθίσταται από την πιο πολυσύχναστη κατηγορία που υπάρχει στην στήλη.
- Εφαρμόζεται ένα pipeline από μετατροπές αναγκαίες για την εισαγωγή των δεδομένων σε αλγορίθμους.
    1. Το StringIndexer κωδικοποιεί μια στήλη συμβολοσειρών ετικετών σε μια στήλη δεικτών ετικετών. Το StringIndexer μπορεί να κωδικοποιήσει πολλές στήλες. Οι δείκτες βρίσκονται σε [0, numLabels] και υποστηρίζονται τέσσερις επιλογές παραγγελίας: "frequencyDesc": φθίνουσα σειρά ανά συχνότητα ετικέτας (η

συχνότερη ετικέτα αντιστοιχεί 0), "frequencyAsc": αύξουσα σειρά ανά συχνότητα ετικέτας (η λιγότερο συχνή ετικέτα αντιστοιχεί 0) , "AlphabetDesc": φθίνουσα αλφαβητική σειρά και "alphabetAsc": αύξουσα αλφαβητική σειρά (προεπιλογή = "frequencyDesc"). Σημειώστε ότι σε περίπτωση ίσης συχνότητας όταν βρίσκεται στο "frequencyDesc" / "frequencyAsc", οι χορδές ταξινομούνται περαιτέρω αλφαβητικά.

2. Το One-hot encoding αντιστοιχεί ένα κατηγορικό χαρακτηριστικό, που αντιπροσωπεύεται ως δείκτες ετικετών, σε ένα δυαδικό διάνυσμα με το πολύ μία μοναδική τιμή που δείχνει την παρουσία μιας συγκεκριμένης τιμής χαρακτηριστικών από ένα σύνολο όλων των τιμών χαρακτηριστικών. Αυτή η κωδικοποίηση επιτρέπει στους αλγόριθμους που αναμένουν συνεχή χαρακτηριστικά να χρησιμοποιούν κατηγορικά χαρακτηριστικά. Για δεδομένα εισαγωγής τύπου συμβολοσειράς, είναι συνηθισμένο να κωδικοποιούνται κατηγορικά χαρακτηριστικά χρησιμοποιώντας το StringIndexer πρώτα. Το OneHotEncoder μπορεί να μετασχηματίσει πολλές στήλες, επιστρέφοντας ένα διάνυσμα one-hot-encoded για κάθε στήλη εισόδου. Είναι συνηθισμένο να συγχωνεύονται αυτά τα διανύσματα σε ένα διάνυσμα χαρακτηριστικών χρησιμοποιώντας το VectorAssembler. Το OneHotEncoder υποστηρίζει την παράμετρο handleInvalid για να επιλέξει τον τρόπο χειρισμού μη έγκυρων εισόδων κατά τη μετατροπή δεδομένων. Στις διαθέσιμες επιλογές περιλαμβάνονται η ένδειξη "keep" (τυχόν μη έγκυρες είσοδοι αντιστοιχίζονται σε έναν επιπλέον κατηγοριοποιημένο δείκτη) και "error" (εμφάνησε σφάλμα).
3. Το VectorAssembler είναι ένας μετασχηματιστής που συνδυάζει μια λίστα στηλών σε ένα διάνυσμα που περιέχει τα στοιχεία όλων των στηλών σε κάθε γραμμή. Είναι χρήσιμο για το συνδυασμό ανεπεξέργαστων (raw) χαρακτηριστικών και χαρακτηριστικών που παράγονται από διαφορετικούς μετασχηματιστές σε ένα διάνυσμα προκειμένου να εκπαιδεύσετε μοντέλα μηχανικής μάθησης. Το VectorAssembler αποδέχεται τους ακόλουθους τύπους στηλών εισαγωγής: όλους τους αριθμητικούς τύπους, τον τύπο boolean και διανύσματα τιμών. Σε κάθε σειρά, οι τιμές των στηλών εισαγωγής θα συνδυαστούν σε ένα διάνυσμα με την καθορισμένη σειρά.

- Τα δεδομένα χωρίζονται σε εκείνα εκπαίδευσης και ελέγχου βάσει των ποσοστών που έχει ορίσει ο χρήστης.
- Εκπαιδεύονται ο αλγόριθμος μηχανικής μάθησης ή το stack από αλγορίθμους και επιστρέφει τα αποτελέσματα όπως φαίνεται παρακάτω.

Για προβλήματα κατάταξης (classification):



Εικόνα 37 Αποτελέσματα Ταξινόμησης

Ο *Confusion Matrix* αποτελείται από τα εξής: πάνω αριστερά False Positive (FP), πάνω δεξιά True Positive (TP), κάτω αριστερά True Negative (TN) και κάτω δεξιά False Negative (FN), με 0=ΟΧΙ και 1=Ναι.

Ακρίβεια (Accuracy) - Η ακρίβεια είναι το πιο διαισθητικό μέτρο απόδοσης και είναι απλώς μια αναλογία σωστά προβλεπόμενης παρατήρησης προς τις συνολικές παρατηρήσεις. Κάποιος μπορεί να πιστεύει ότι, αν έχουμε υψηλή ακρίβεια, τότε το μοντέλο μας είναι το καλύτερο. Ναι, η ακρίβεια είναι ένα εξαιρετικό μέτρο, αλλά μόνο όταν έχετε συμμετρικά σύνολα δεδομένων όπου οι τιμές ψευδώς θετικών και ψευδών αρνητικών είναι σχεδόν ίδιες. Επομένως, πρέπει να εξετάσετε άλλες παραμέτρους για να αξιολογήσετε την απόδοση του μοντέλου σας.

$$Accuracy = TP + TN / TP + FP + FN + TN$$

Precision - Precision είναι ο λόγος των σωστά προβλεπόμενων θετικών παρατηρήσεων προς τις συνολικές προβλεπόμενες θετικές παρατηρήσεις. Η υψηλή ακρίβεια σχετίζεται με το χαμηλό ψευδώς θετικό (FP) ποσοστό.

$$Precision = TP/TP + FP$$

Ανάκληση (Recall) - Ανάκληση είναι ο λόγος των σωστά προβλεπόμενων θετικών παρατηρήσεων προς όλες τις παρατηρήσεις στην πραγματική τάξη - ναι.

$$Recall = \mathrm{TP}/\mathrm{TP} + \mathrm{FN}$$

F1-Score - Το F1-Score είναι ο σταθμισμένος μέσος όρος του Precision και της ανάκλησης. Επομένως, λαμβάνει υπόψη τόσο ψευδώς θετικά όσο και ψευδώς αρνητικά. Διαισθητικά δεν είναι τόσο εύκολο να κατανοηθεί όσο η ακρίβεια, αλλά το F1 είναι συνήθως πιο χρήσιμο από την ακρίβεια, ειδικά αν υπάρχει μια άνιση κατανομή τάξης. Το Precision λειτουργεί καλύτερα αν τα ψευδώς θετικά και τα ψευδώς αρνητικά έχουν παρόμοιο κόστος. Εάν το κόστος των

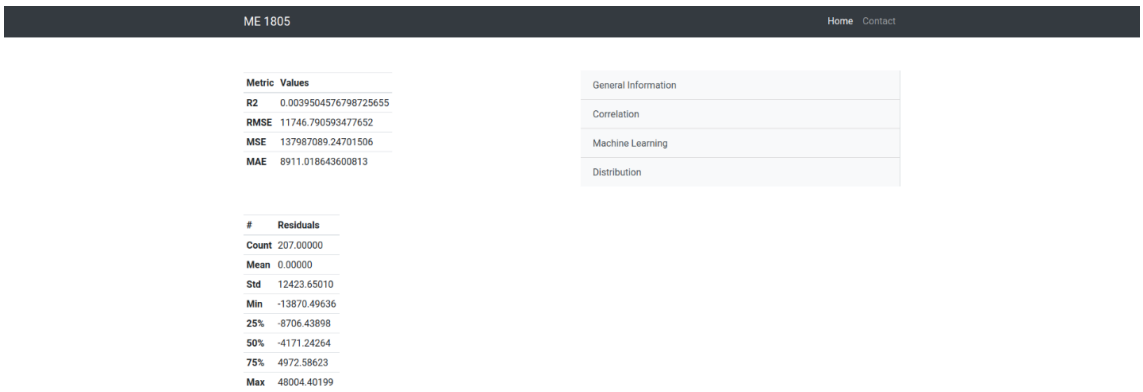
ψευδών θετικών και των ψευδών αρνητικών είναι πολύ διαφορετικό, είναι καλύτερο να κοιτάξετε τόσο την Precision όσο και την Ανάκληση.

$$F1 - score = \ 2 * (\text{Recall} \ * \ \text{Precision}) \ / \ (\text{Recall} \ + \ \text{Precision})$$

Η AUC παρέχει ένα συνολικό μέτρο απόδοσης σε όλα τα πιθανά όρια ταξινόμησης. Ένας τρόπος ερμηνείας της AUC είναι η πιθανότητα ότι το μοντέλο κατατάσσει ένα τυχαίο θετικό παράδειγμα πολύ περισσότερο από ένα τυχαίο αρνητικό παράδειγμα. Η AUC κυμαίνεται από 0 έως 1. Ένα μοντέλο του οποίου οι προβλέψεις είναι 100% λανθασμένες έχει AUC 0. κάποιος του οποίου οι προβλέψεις είναι 100% σωστές έχει AUC 1. Η AUC είναι επιθυμητή για τους ακόλουθους δύο λόγους:

- Η AUC είναι αμετάβλητη σε κλίμακα. Μετρά πόσο καλά ταξινομούνται οι προβλέψεις, παρά τις απόλυτες τιμές τους.
- Η AUC είναι αμετάβλητη ταξινόμηση. Μετρά την ποιότητα των προβλέψεων του μοντέλου ανεξάρτητα από το ποιο όριο ταξινόμησης επιλέγεται.

Στην περίπτωση της παλινδρόμησης (regression) τα αποτελέσματα που επιστρέφονται είναι όπως φαίνεται παρακάτω.



Εικόνα 38 Παράδειγμα αποτελεσμάτων γραμμικής παλινδρόμησης

Τα σημεία δεδομένων συνήθως δεν εμπίπτουν ακριβώς στη γραμμή εξίσωσης παλινδρόμησης. Ένα υπόλοιπο (residual) είναι η κατακόρυφη απόσταση μεταξύ ενός σημείου δεδομένων και της γραμμής παλινδρόμησης. Κάθε σημείο δεδομένων έχει ένα υπόλοιπο. Είναι θετικά εάν βρίσκονται πάνω από τη γραμμή παλινδρόμησης και αρνητικά εάν βρίσκονται κάτω από τη

γραμμή παλινδρόμησης. Εάν η γραμμή παλινδρόμησης πραγματικά διέρχεται από το σημείο, το υπόλοιπο σε αυτό το σημείο είναι μηδέν.

Το μέσο απόλυτο σφάλμα (MAE) αντιπροσωπεύει τον μέσο όρο της απόλυτης διαφοράς μεταξύ των πραγματικών και των προβλεπόμενων τιμών στο σύνολο δεδομένων. Μετρά τον μέσο όρο των υπολοίπων (residuals) στο σύνολο δεδομένων.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}|$$

Το μέσο σφάλμα τετραγώνου (MSE) αντιπροσωπεύει τον μέσο όρο της τετραγωνικής διαφοράς μεταξύ των αρχικών και των προβλεπόμενων τιμών στο σύνολο δεδομένων. Μετρά τη διακύμανση των υπολοίπων (residuals).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

Η τετραγωνική ρίζα του μέσου σφάλματος (RMSE) μετρά την τυπική απόκλιση των υπολοίπων (residuals).

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2}$$

Ο συντελεστής προσδιορισμού ή R-τετράγωνο ή R2 αντιπροσωπεύει το ποσοστό της διακύμανσης στην εξαρτημένη μεταβλητή που εξηγείται από το μοντέλο γραμμικής παλινδρόμησης. Είναι μια βαθμολογία χωρίς κλίμακα, ανεξάρτητα από τις τιμές που είναι μικρές ή μεγάλες, η τιμή του R2 θα είναι μικρότερη από 1.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

Τέλος, αν υπάρξει ανάγκη επικοινωνίας γίνεται μέσω της σελίδας *Contact.*

Εικόνα 39 Σελίδα Επικοινωνίας

6. ΣΥΝΟΨΗ

Για την αντιμετώπιση των προβλημάτων που αναφέρθηκαν στην αρχή της εργασίας δημιουργήθηκε η παραπάνω πλατφόρμα μηχανικής μάθησης. Παρέχοντας έτσι δυνατότητα σε χρήστες να έχουν εύκολη και γρήγορη πρόσβαση σε μεγάλα δεδομένα για διαχείρηση αυτών καθώς και εφαρμογή αλγορίθμων μηχανικής μάθησης μέσω του Apache Spark σε αυτά. Με φιλικό προς τον χρήστη interface και πρακτικό αφού για οποιαδήποτε ενέργεια απαιτούνται μερικά clicks.

Η συγκεκριμένη πλατφόρμα επιδέχεται κάποιες ενέργειες οι οποίες θα βελτιώσουν αρκετά την ποιότητα της καθώς και την εμπειρία του χρήστη. Οι ενέργειες αυτές χωρίζονται σε δύο κύριες κατηγορίες, σε επίπεδο υποδομής (infrastructure) και σε επίπεδο προσθήκης περισσότερων αλγορίθμων μηχανικής μάθησης καθώς και τεχνικών για την περαιτέρω επεξεργασία δεδομένων.

Όσον αφορά την υποδομή στην οποία υπάρχει η πλατφόρμα ένα σημαντικό βήμα θα ήταν να η ανάπτυξη (deploy) της σε κάποιο Kubernetes το οποίο θα παρέχεται από κάποιον cloud πάροχο (π.χ. AWS, Azure, GCP, IBM, κ.α) σαν υπηρεσία. Αυτό θα βοηθήσει στη διαχείριση της εφαρμογής σε γενικό επίπεδο, όπως να δημιουργεί περισσότερα instances της εφαρμογής βάσει ζήτησης, σε περίπτωση προβλήματος ή διακοπής λειτουργίας την αντικατάστασή της κ.α. Επίσης, η αποθήκευση των δεδομένων αφού φορτωθούν θα μπορούσε να γίνεται μέσω κάποιας βάσης δεδομένων (π.χ. PostgreSQL, Cassandra DB, MongoDB κ.α) ή κάποιου S3 Bucket. Αντίστοιχα θα μπορούσε να παρέχεται η δυνατότητα στον χρήστη να φορτώνει τα δεδομένα του από τις παραπάνω μεθόδους αποθήκευσης. Επίσης, μπορούν να προσθεθούν nodes στο Spark cluster διαδικασία η οποία θα επιφέρει εμφανείς διαφορές στην ταχύτητα και την αποδοτικότητα.

Στο επίπεδο της προσθήκης αλγορίθμων μηχανικής μάθησης και τεχνικών θα μπορούσαν να προστεθούν αλγόριθμοι για ταξινόμηση και παλινδρόμηση. Μερικοί εξ αυτών είναι οι Naïve Bayes, Multilayer Perceptron και Linear SVC για ταξινόμηση και Random Forest Regressor, Isotonic Regressor για παλινδρόμηση. Επίσης, πέρα από την συσχέτιση Pearson θα μπορούσε να προστεθεί η Spearman και Kendall συσχέτιση.

ΑΝΑΦΟΡΕΣ

Big Data: Sagiroglu, S., & Sinanc, D. (2013). Big data: A review. 2013 International Conference on Collaboration Technologies and Systems (CTS). doi:10.1109/cts.2013.6567202

https://www.sas.com/en_us/insights/big-data/what-is-big-data.html#:~:text=Big%20data%20is%20a%20term,with%20the%20data%20that%20matters.

https://www.guru99.com/what-is-big-data.html#:~:text=Big%20Data%20is%20a%20collection,data%20but%20with%20huge%20size.

Weka: https://www.cs.waikato.ac.nz/ml/weka/, https://markahall.blogspot.com/2015/03/weka-and-spark.html

KNIME Analytics: https://www.knime.com/knime-analytics-platform, https://www.knime.com/knime-extension-for-apache-spark

Apache Spark: https://spark.apache.org/powered-by.html

https://www.infoworld.com/article/3236869/what-is-apache-spark-the-big-data-platform-that-crushed-hadoop.html

Python: https://www.python.org/ , https://www.tutorialspoint.com/python/index.htm#:~:text=Python%20is%20a%20high%2Dlevel,syntactical%20constructions%20than%20other%20languages.

Flask: https://flask.palletsprojects.com/en/1.1.x/

Jinja: https://jinja.palletsprojects.com/en/2.11.x/

HTML: https://html.com/

Bootstrap: https://getbootstrap.com/

JavaScript: https://www.javascript.com/

jQuery: https://jquery.com/

Docker Container: https://www.docker.com/resources/what-container, https://docs.docker.com/engine/reference/commandline/container/

Machine Learning: https://www.expert.ai/blog/machine-learning-definition/#:~:text=Machine%20learning%20is%20an%20application,it%20to%20learn%20for%20themselves., https://medium.com/@supun.setunga/stacking-in-machine-learning-357db1cfc3a, https://machinelearningmastery.com/stacking-ensemble-machine-learning-with-

python/#:~:text=Stacked%20Generalization%20or%20%E2%80%9CStacking%E2%80%9D%20for,dataset%2C%20like%20bagging%20and%20boosting.

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. Science, 349(6245), 255–260

Logistic Regression: https://machinelearningmastery.com/logistic-regression-for-machine-learning/,

J.C. van Houwelingen & S. le Cessie, 1988. "Logistic Regression, a review," Statistica Neerlandica, Netherlands Society for Statistics and Operations Research, vol. 42(4), pages 215-232, December.

Gradient Boosting Trees: https://towardsdatascience.com/machine-learning-part-18-boosting-algorithms-gradient-boosting-in-python-ef5ae6965be4, https://stackabuse.com/gradient-boosting-classifiers-in-python-with-scikit-learn/, https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/, https://spark.apache.org/docs/latest/ml-classification-regression.html#gradient-boosted-tree-regression, https://spark.apache.org/docs/latest/ml-classification-regression.html#gradient-boosted-tree-classifier,

Cui, H., Huang, D., Fang, Y., Liu, L., & Huang, C. (2018). Webshell Detection Based on Random Forest–Gradient Boosting Decision Tree Algorithm. 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC).

Decision Trees: https://en.wikipedia.org/wiki/Decision_tree_learning, https://medium.com/swlh/decision-tree-classification-de64fc4d5aac, https://spark.apache.org/docs/latest/ml-classification-regression.html#decision-tree-classifier, https://spark.apache.org/docs/latest/ml-classification-regression.html#decision-tree-regression

Random Forest: https://spark.apache.org/docs/latest/ml-classification-regression.html#random-forest-classifier , Cui, H., Huang, D., Fang, Y., Liu, L., & Huang, C. (2018). Webshell Detection Based on Random Forest–Gradient Boosting Decision Tree Algorithm. 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC).

Classification: “Classification, Clustering, and Data Analysis: Recent Advances and Applications”

Regression: https://hbr.org/2015/11/a-refresher-on-regression-analysis

Correlation: https://towardsdatascience.com/normalization-vs-standardization-quantitative-analysis-a91e8a79cebf, https://towardsdatascience.com/clearly-explained-pearson-v-s-spearman-correlation-coefficient-ada2f473b8, https://support.minitab.com/en-us/minitab-express/1/help-and-how-to/modeling-statistics/regression/supporting-topics/basics/a-comparison-of-the-pearson-and-spearman-correlation-methods/#:~:text=The%20Pearson%20correlation%20evaluates%20the%20linear%20relationsh

ip%20between%20two%20continuous%20variables.&text=The%20Spearman%20correlation%20coefficient%20is,evaluate%20relationships%20involving%20ordinal%20variables.,

Pedrycz, W., & Smith, M. H. (1999). Granular correlation analysis in data mining. FUZZ-IEEE'99. 1999 IEEE International Fuzzy Systems. Conference Proceedings (Cat. No.99CH36315).

Standardize / Normalize: https://towardsdatascience.com/scale-standardize-or-normalize-with-scikit-learn-6ccc7d176a02, https://towardsai.net/p/data-science/how-when-and-why-should-you-normalize-standardize-rescale-your-data-3f083def38ff#:~:text=Normalization%3A,when%20features%20have%20different%20ranges., https://spark.apache.org/docs/2.1.0/ml-features.html#minmaxscaler, https://spark.apache.org/docs/2.1.0/ml-features.html#standardscaler, https://spark.apache.org/docs/2.1.0/ml-features.html#normalizer,

Normalization and standardization of electronic health records for high-throughput phenotyping: the SHARPn consortium

Data process techniques: https://spark.apache.org/docs/2.1.0/ml-features.html#stringindexer, https://spark.apache.org/docs/2.1.0/ml-features.html#onehotencoder, https://spark.apache.org/docs/2.1.0/ml-features.html#vectorassembler

Linear Regression: https://spark.apache.org/docs/latest/ml-classification-regression.html#linear-regression,

Wagner, H. M. (1959). Linear Programming Techniques for Regression Analysis. Journal of the American Statistical Association, 54(285), 206–212.

User Interface: https://startbootstrap.com/templates

Metrics: https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e, https://www.displayr.com/what-is-a-correlation-matrix/, https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/#:~:text=80%25%20accurate.&text=Precision%20%2D%20Precision%20is%20the%20ratio,the%20total%20predicted%20positive%20observations.&text=F1%20score%20%2D%20F1%20Score%20is,and%20false%20negatives%20into%20account.

Related GitHub: https://towardsdatascience.com/building-a-linear-regression-with-pyspark-and-mllib-d065c3ba246a, https://spark.apache.org/docs/latest/api/python/pyspark.ml.html, https://github.com/nxs5899/end-to-end-Machine-Learning-model-with-MLlib-in-pySpark/blob/master/MLlib_pySpark.ipynb, https://github.com/gogundur/Pyspark-Logistic-Regression/blob/master/Pyspark/Pyspark%20Classification.ipynb, https://github.com/harunurrashid97/Machine-learning-with-PySpark/blob/master/First%20PySpark%20ml%20model/First%20ML%20models%20using%20PySpark.ipynb