Αυτόματη κατασκευή προφίλ στατιστικών ιδιοτήτων ενός συνόλου δεδομένων

Αλέξανδρος Αλεξίου

Διπλωματική Εργασία

Επιβλέπων: Π. Βασιλειάδης

Ιωάννινα, Μάρτιος 2022

ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNINA

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου Παναγιώτη Βασιλειάδη για την βοήθεια, την κατανόηση και την συνεργασία καθ' όλη τη διάρκεια της εκπόνησης της παρούσας διπλωματικής εργασίας.

Φυσικά, θα ήθελα να ευχαριστήσω τους γονείς μου, που μου παρέχουν τα πάντα απλόχερα και μου δίνουν την δυνατότητα να πάω παρακάτω και τους κοντινούς μου ανθρώπους για την υποστήριξη που μου πρόσφεραν όλα τα χρόνια, καθώς χωρίς αυτούς δεν θα είχα την ίδια δύναμη να προχωρήσω και να βελτιωθώ.

10/03/2022

Αλέξανδρος Αλεξίου

Περίληψη στα ελληνικά

Σκοπός της συγκεκριμένης πτυχιακής εργασίας ήταν η κατασκευή ενός συστήματος, το οποίο θα παρείχε διευκόλυνση στους αναλυτές δεδομένων για γρήγορα αποτελέσματα στατιστικών για σύνολα δεδομένων. Αναλυτικότερα, μιλάμε για ένα σύστημα, το οποίο δέχεται ως είσοδο αρχεία με δεδομένα σε γνωστές μορφές αρχείων, όπως αυτά παράγονται από διάφορα συστήματα σε πολλούς τομείς της επιστήμης των δεδομένων. Έπειτα, επεξεργάζεται την πληροφορία που περιέχουν τα αρχεία αυτά και παρέχει ένα στατιστικό προφίλ για τα δεδομένα εισόδου. Υπάρχει δυνατότητα δημιουργίας νέων (labeled) κολόνων από υπάρχουσες κολόνες, αυτόματη δημιουργία δέντρων πεποιθήσεων, υπολογισμός συσχετίσεων μεταξύ των κολόνων καθώς και περιγραφικά στατιστικά. Αφού μελετήθηκαν τα απαραίτητα μαθηματικά και εργαλεία για την παραγωγή του εργαλείου δημιουργήθηκε το εργαλείο με όνομα Pythia. Ο κώδικας είναι γραμμένος σε γλώσσα Java χρησιμοποιώντας το Apache Spark για τους υπολογισμούς παρέχοντας γρήγορα αποτελέσματα, ενώ ο κώδικας είναι εύκολα επεκτάσιμος και το σύστημα ανθεκτικό σε κλιμάκωση.

Λέξεις Κλειδιά: Επιστήμη Δεδομένων, Ανάλυση δεδομένων, Java, Apache Spark

Abstract

The purpose of this thesis was to build a system that would facilitate data analysts for fast statistical results for datasets. More specifically, we are talking about a system, which accepts as input data files in known file formats, as they are produced by various systems in many areas of Data Science. It then processes the information contained in these files and provides a statistical profile for the input data. It is possible to create new (labeled) columns from existing columns, automatically create Decision trees, calculate correlations between columns as well as descriptive statistics. After studying the necessary mathematics and tools to be able to produce the tool, a tool called Pythia was created. The code is written in Java using Apache Spark for computations, providing fast results, while the code is easily scalable and the system resistant to scaling.

Keywords: Data Science, Data Analysis, Java, Apache Spark

Πίνακας περιεχομένων

Κεφάλαιο 1. Εισαγωγή

Το συγκεκριμένο κεφάλαιο αναφέρεται σε μία σύντομη περιγραφή του αντικειμένου της πτυχιακής, καθώς και στην περιγραφή των επόμενων ενοτήτων.

1.1 Αντικείμενο της διπλωματικής

Τα σύνολα δεδομένων είναι ζωτικής σημασίας και χρησιμοποιούνται πολύ συχνά στο χώρο των Data Science και Data Analytics. Είναι απαραίτητα για την εξαγωγή χρήσιμης πληροφορίας από διαφόρων ειδών δεδομένα έπειτα από την επεξεργασία τους με χρήση αλγορίθμων της επιστήμης των δεδομένων. Η επιλογή ενός συνόλου δεδομένων δεν είναι μια απλή διαδικασία καθώς τα δεδομένα πρέπει να έχουν μια ποιότητα για να αξιοποιηθούν.

Για να εκτιμηθεί η ποιότητα των δεδομένων υπάρχουν εργαλεία τα οποία αξιοποιούνται όπως η οπτικοποίηση των δεδομένων, η εξαγωγή απλών στατιστικών στοιχείων (π.χ. μέση Τιμή, διάμεσος, διακύμανση κ.α.) για την περιγραφή ενός στατιστικού προφίλ το οποίο θα δείξει την ποιότητα των δεδομένων. Τα εργαλεία αυτά συνήθως είναι διαδραστικά, δηλαδή είναι απαραίτητο ο αναλυτής να εξάγει ότι χρειάζεται με το χέρι.(Orange, Tableau κ.ά.) Για παράδειγμα, για τη δημιουργία ενός decision tree σε labeled στήλες, correlation ανάμεσα σε στήλες, μέση τιμή, διακύμανση κ.α. ανάλογα τις στήλες ενδιαφέροντός του.

Η έλλειψη της αυτοματοποίησης αποτελεί μεγάλο πρόβλημα και για το λόγο αυτό στα πλαίσια της εν λόγω διπλωματικής εργασίας κατασκευάστηκε ένα βοηθητικό εργαλείο αυτόματης εξαγωγής στατιστικού προφίλ συνόλων δεδομένων, το Pythia. Το Pythia είναι μια Java εφαρμογή που χρησιμοποιεί το εργαλείο Apache Spark για την επεξεργασία των δεδομένων ώστε να παρέχει αυτόματα τα στοιχεία αναλυτικής επεξεργασίας. Το Apache Spark (Apache Spark, n.d.) είναι μια μηχανή ανάλυσης για επεξεργασία δεδομένων μεγάλης κλίμακας που χρησιμοποιείται εκτενώς στο χώρο των data analytics και είναι

κατάλληλο σε πληθώρα περιπτώσεων που υπάρχει ανάγκη για γρήγορη και επεκτάσιμη επεξεργασία μεγάλων δεδομένων.

Το εργαλείο θα πρέπει να επιτρέπει στον αναλυτή να εγγράψει ένα data set και με τη βοήθεια του συστήματος να δηλώσει τα πεδία του, και τον τύπο τους (int, double, dateTime, Boolean, enum of class labels, κλπ). Πέρα από τα απλά περιγραφικά σταστικά που θα παράγει το σύστημα όπως μέση τιμή, διάμεσος κλπ. για κάθε labeled πεδίο θα δημιουργήσει ένα decision trees για κάθε όλα πεδίο στη βάση όλων των πεδίων του data set, clustering των εγγραφών του dataset και αποτίμηση της ποιότητας του clustering, όπως και Hypothesis testing και chi-squared test. Αν ο αναλυτής έχει δηλώσει μόνο κάποια πεδία να εμπλακούν στην παραγωγή ενός συγκεκριμένου decision tree, τότε εμπλέκονται μόνο αυτά.

Τέλος, το εργαλείο θα παράγει μια αναφορά με τα στατιστικα αποτελέσματα των δεδομένων καθώς και την οπτικοποίηση των αποτελεσμάτων των μεθόδων που εφάρμοσε κατά την επεξεργασία.

1.2 Οργάνωση του τόμου

Τα παρακάτω κεφάλαια πραγματεύονται αναλυτικότερα την δημιουργία της εν λόγω πτυχιακής. Πιο συγκεκριμένα:

Στο δεύτερο κεφάλαιο, γίνεται η περιγραφή των λειτουργιών που θα προσφέρει το σύστημα που δημιουργήθηκε, αναλύθηκε το μαθηματικό υπόβαθρο απαραίτητο για την κατανόηση των υπολογισμών που θα εκτελεί το σύστημα και η ανάλυση των απαραίτητων εννοιών για την χρήση του Apache Spark [Apache Spark, n.d.].

Στο τρίτο κεφάλαιο, περιγράφονται οι λειτουργείες του συστήματος για τη λύση του προβλήματος της αυτόματης εξαγωγής στατιστικών και αναλύεται η ανάπτυξη του συστήματος με διαγράμματα πακέτων και UML των κλάσεων της υλοποίησης.

Το τέταρτο κεφάλαιο, αναλύεται η απόδοση του συστήματος μέσω μετρήσεων των υποσυστημάτων παρέχοντας γραφικές παραστάσεις των αποτελεσμάτων.

Μια σύνοψη της πτυχιακής εργασίας και η παρουσίαση των συμπερασμάτων είναι το περιεχόμενο του πέμπτου κεφαλαίου. Επίσης, αναφέρονται πιθανές μελλοντικές επεκτάσεις του εργαλείου.

Τέλος, γίνεται αναφορά στη βιβλιογραφία, η οποία έχει χρησιμοποιηθεί για τη διεκπεραίωση της πτυχιακής αυτής εργασίας.

Κεφάλαιο 2. Περιγραφή Θέματος

2.1 Στόχος της εργασίας

Ο στόχος του εργαλείου της παρούσας διπλωματικής εργασίας είναι να επιτρέπει στον μελετητή ενός συνόλου δεδομένων να βλέπει με αυτοματοποιημένο τρόπο τα βασικά στοιχεία του στατιστικού προφίλ του συνόλου δεδομένων μέσω μιας παραχθείσας αναφοράς η οποία θα περιέχει ενδιαφέρουσες μετρικές και άλλα στοιχεία που είναι απαραίτητα για την κατανόηση των δεδομένων αυτών.

Συγκεκριμένα, οι απαιτήσεις του συστήματος οργανώνονται ως εξής:

1. Θα πρέπει ο αναλυτής να μπορεί να φορτώσει στο σύστημα τα δεδομένα (csv, tsv, xlsx, json, etc), όπως παράγονται από διάφορα συστήματα (IoT, ιατρικά δεδομένα, κλπ).
2. Ο αναλυτής θα πρέπει να μπορεί να εγγράψει το dataset και με τη βοήθεια του συστήματος να δηλώσει τα πεδία του και τον τύπο των πεδίων αυτών (int, double, dateTime, Boolean, enum of class labels, κλπ.)
3. Ο αναλυτής θα πρέπει να μπορεί να δηλώσει κανόνες labeling για μια στήλη, να χαρακτηρίσει τις τιμές αυτής της στήλης και το σύστημα θα πρέπει να παράγει τη νέα labeled στήλη.
4. Ο αναλυτής θα πρέπει να μπορεί να εκδηλώσει ενδιαφέρον για κάποια πεδία.
5. Ο αναλυτής θα πρέπει να μπορεί, αν το επιθυμεί, να δηλώσει συγκεκριμένα πεδία τα οποία θα εμπλακούν στην δημιουργεί του decision tree.
6. Ο αναλυτής θα πρέπει να λαμβάνει μια αναφορά στο τέλος της επεξεργασίας με στατιστικά στοιχεία και αποτελέσματα των αλγορίθμων όπως Decision tree ή με μελλοντικές προσθήκες, Clustering των εγγραφών του data set, αποτίμηση της ποιότητας του Clustering και Hypothesis testing (π.χ. chi-squared test).

2.2 Υπόβαθρο

Ένα σύνολο δεδομένων είναι μια συλλογή δεδομένων συνήθως στην μορφή πίνακα όπου κάθε στήλη αντιπροσωπεύει μια συγκεκριμένη μεταβλητή και κάθε σειρά αντιστοιχεί σε μια δεδομένη εγγραφή του εν λόγω συνόλου δεδομένων. Τα δεδομένα αυτά μπορεί να προέρχονται από διαφόρων ειδών μετρήσεις και απαιτείται η κατανόηση τους για να χρησιμοποιηθούν αποδοτικά εξάγοντας χρήσιμη πληροφορία.

Περιγραφικά Στατιστικά

Τα περιγραφικά στατιστικά στοιχεία, βοηθούν στην περιγραφή και την κατανόηση των χαρακτηριστικών ενός συγκεκριμένου συνόλου δεδομένων δίνοντας σύντομες περιλήψεις σχετικά με το δείγμα και τα μέτρα των δεδομένων [Bosl12]. Οι πιο γνωστοί τύποι περιγραφικών στατιστικών είναι τα μέτρα κεντρικής τάσης: ο μέσος όρος (mean), η διάμεσος (median) και επικρατούσα τιμή (mode), που χρησιμοποιούνται σχεδόν σε όλα τα επίπεδα των μαθηματικών και της στατιστικής. Ο μέσος όρος, υπολογίζεται προσθέτοντας όλα τα ψηφία στο σύνολο δεδομένων και στη συνέχεια διαιρώντας με τον αριθμό των ψηφίων του συνόλου. Η διάμεσος είναι η τιμή που διαχωρίζει το υψηλότερο μισό από το κάτω μισό ενός δείγματος δεδομένων, ενός πληθυσμού ή μιας κατανομής πιθανοτήτων. Για ένα σύνολο δεδομένων, μπορεί να θεωρηθεί ως "η μεσαία" τιμή. Το βασικό χαρακτηριστικό της διάμεσης τιμής στην περιγραφή δεδομένων σε σύγκριση με τη μέση τιμή (συχνά περιγράφεται απλώς ως "μέσος όρος") είναι ότι δεν παραμορφώνεται από μια μικρή αναλογία εξαιρετικά μεγάλων ή μικρών τιμών και επομένως παρέχει καλύτερη αναπαράσταση μιας "τυπικής" τιμής του συνόλου. Σε μία απόλυτα συμμετρική κατανομή δεδομένων οι παραπάνω τρεις τιμές ταυτίζονται ενώ σε μη συμμετρικές κατανομές διαφέρουν.

Φόρμουλα για τον υπολογισμό της μέσης τιμής:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Φόρμουλα για τον υπολογισμό διάμεσου (όπου n ο αριθμός των στοιχείων ενός συνόλου $X = \{x_1, \ldots x_n\}$):

Αν το n περιττό, $median(x) = x_{\frac{(n+1)}{2}}$

Αν το n άρτιο, $median(x) = \frac{x_{n/2} + x_{(n/2)+1}}{2}$

Ένα ακόμα είδος περιγραφικών στατιστικών είναι οι δείκτες διασποράς. Η διασπορά αναφέρεται στο πόσο μεταβλητή ή "απλωμένη" είναι μια μεταβλητή. Ο πιο απλός δείκτης διασποράς είναι το εύρος (range) και ουσιαστικά είναι η διαφορά μεταξύ της μεγαλύτερης και μικρότερης τιμής μιας μεταβλητής. Στην περίπτωση που υπάρχουν ακραίες τιμές τότε η διαφορά αυτή γίνεται λιγότερη χρήσιμη. Μια ακραία τιμή μπορεί να οφείλεται σε μεταβλητότητα στη μέτρηση ή μπορεί να υποδεικνύει πειραματικό σφάλμα και μερικές φορές εξαιρείται από το σύνολο δεδομένων. Μια ακραία τιμή μπορεί να προκαλέσει σοβαρά προβλήματα στις στατιστικές αναλύσεις. Τον περιορισμό αυτόν λύνει το διατεταρτημοριακό διάστημα (interquartile range -- IQR) το οποίο είναι λιγότερο επηρεαζόμενο από ακραίες τιμές και ορίζεται ως η διαφορά μεταξύ του 75ου και του 25ου εκατοστημορίου των δεδομένων. Για τον υπολογισμό του IQR, το σύνολο δεδομένων χωρίζεται σε τεταρτημόρια, ή σε τέσσερα ζυγά μέρη με σειρά κατάταξης μέσω γραμμικής παρεμβολής (linear interpolation). Αυτά τα τεταρτημόρια συμβολίζονται με Q1 (ονομάζεται επίσης κατώτερο τεταρτημόριο), Q2 (διάμεσος) και Q3 (ονομάζεται επίσης και άνω τεταρτημόριο). Το κάτω τεταρτημόριο αντιστοιχεί στο 25ο εκατοστημόριο και το ανώτερο τεταρτημόριο αντιστοιχεί στο 75ο εκατοστημόριο, άρα IQR = Q3 − Q1. Στη στατιστική, ακραίες τιμές είναι ένα τιμές δεδομένων που διαφέρουν σημαντικά από άλλες παρατηρήσεις.

Οι πιο συχνοί δείκτες διασποράς είναι η διακύμανση (variance) και η τυπική απόκλιση (Standard Deviation). Η διακύμανση είναι το τετράγωνο της τυπικής απόκλισης και οι δύο μετρικές αυτές προσδιορίζουν το πόσο πολύ διαφέρουν οι τιμές του συνόλου δεδομένων από την μέση τιμή και υπολογίζονται ελάχιστα διαφορετικά ανάλογα αν μελετάται ένα δείγμα (sample) ή ένας πληθυσμός (population).

Φόρμουλα για τον υπολογισμό της διακύμανσης δείγματος. Συμβολίζεται με αγγλικό S υψωμένο στο τετράγωνο :

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

Όπου $x_i \in \{x_1, \dots x_n\}$ και $\bar{x}$ η μέση τιμή του δείγματος

Φόρμουλα για τον υπολογισμό της διακύμανσης πληθυσμού. Συμβολίζεται με ελληνικό σίγμα υψωμένο στο τετράγωνο:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

Όπου $x_i \in \{x_1, \dots x_n\}$ και μ η μέση τιμή του πληθυσμού

Αφού οι αριθμοί που είναι υψωμένοι στο τετράγωνο είναι πάντα θετικοί, η διακύμανση είναι πάντα μεγαλύτερη ή ίση του μηδενός. Ωστόσο αυτό επιτυγχάνεται καθώς έχουν αλλάξει οι μονάδες και μπορεί να μην είναι βολικό στην χρήση. Για παράδειγμα, στην περίπτωση μου μετριούνται κιλά πλέον θα μετριούνται τετραγωνικά κιλά. Για να αποφευχθεί αυτό χρησιμοποιείται η τετραγωνική ρίζα της διακύμανσης, που είναι η τυπική απόκλιση.

Φόρμουλα για τον υπολογισμό της τυπικής απόκλισης δείγματος:

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Όπου $x_i \in \{x_1, \dots x_n\}$ και $\bar{x}$ η μέση τιμή του δείγματος

Φόρμουλα για τον υπολογισμό της τυπικής απόκλισης πληθυσμού:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$$

Όπου $x_i \in \{x_1, \dots x_n\}$ και μ η μέση τιμή του πληθυσμού

Γενικότερα, υποστηρίζουμε ότι σε δύο ομάδες ίδιου μεγέθους που μετριούνται στις ίδιες μονάδες η ομάδας με την μεγαλύτερη διακύμανση και τυπική απόκλιση έχει μεγαλύτερη μεταβλητότητα στις τιμές της. Στην περίπτωση που συγκρίνονται μεταβλητές σε διαφορετικές μονάδες τότε η διακύμανση θα επηρεαστεί. Για παράδειγμα, αν γίνει σύγκριση βάρους ομάδων σε κιλά και ουγκιές, τότε το δείγμα που μετράται σε ουγκιές θα έχει μεγαλύτερη διακύμανση. Όταν συγκρίνονται ομάδες σε διαφορετικές μονάδες τότε είναι δύσκολα να εκτιμηθεί η μεταβλητότητα μιας μεταβλητής σε σχέση με μια άλλη. Ο συντελεστή μεταβλητότητας (coefficient of variation) ξεπερνά την δυσκολία αυτή και δίνει την δυνατότητα σύγκρισης μεταβλητότητας μεταβλητών διαφορετικών μονάδων και υπολογίζεται διαιρώντας την τυπική απόκλιση με τη μέση τιμή πολλαπλασιάζοντας με το εκατό.

Φόρμουλα για τον υπολογισμό του συντελεστή μεταβλητότητας:

$$CV = \frac{S}{\bar{x}} \times 100$$

Όπου $x_i \in \{x_1, \dots x_n\}$ και $\underline{x}$ η μέση τιμή του δείγματος

Γραφικές αναπαραστάσεις

Μια άλλη πολύ σημαντική τεχνική κατανόησης δεδομένων είναι οι γραφικές αναπαραστάσεις [Bosl12] . Ενώ οι στατιστικές μέθοδοι και οι διαδικασίες ανάλυσης δεδομένων γενικά δίνουν την έξοδο τους σε αριθμητική ή πινακοειδή μορφή, οι γραφικές τεχνικές επιτρέπουν την εμφάνιση τέτοιων αποτελεσμάτων σε κάποιο είδος εικονογραφικής μορφής. Αυτές περιλαμβάνουν διαγράμματα όπως bar charts, pie charts, pareto charts, box plots, histograms, bivariate charts, scatter plots, line charts κλπ.

Ραβδόγραμμα

Ένα bar chart ή γράφημα ράβδων είναι ένα γράφημα που παρουσιάζει κατηγορικά δεδομένα με ορθογώνιες ράβδους με ύψη ή μήκη ανάλογα με τις τιμές που αντιπροσωπεύουν. Οι ράβδοι μπορούν να σχεδιαστούν κατακόρυφα ή οριζόντια. Οριζόντια αναπαρίστανται δεδομένα ονομαστικών μεταβλητών (nominal variables) και κάθετα τακτικών μεταβλητών (ordinal variables). Ένα κάθετο bar chart ονομάζεται μερικές φορές column chart. Οι ράβδοι σε ένα bar chart είναι ξεχωριστές η μία από τη άλλη και δεν υπονοούν συνέχεια ανεξάρτητα από το αν οι ράβδοι είναι κατηγορίες συνεχούς μεταβλητής.

Ιστόγραμμα

Ένα histogram ή ιστόγραμμα είναι μια κατά προσέγγιση αναπαράσταση της κατανομής αριθμητικών δεδομένων. Εισήχθη για πρώτη φορά από τον Karl Pearson. Για να κατασκευαστεί ένα ιστόγραμμα, το πρώτο βήμα είναι να ομαδοποιηθεί (bin) το εύρος των τιμών - δηλαδή, να διαιρεθεί ολόκληρο το εύρος τιμών σε μια σειρά από διαστήματα (γνωστά και ως «κάδοι») και στη συνέχεια να μετρηθούν πόσες τιμές εμπίπτουν σε κάθε διάστημα. Οι κάδοι συνήθως καθορίζονται ως διαδοχικά, μη επικαλυπτόμενα διαστήματα μιας μεταβλητής. Οι κάδοι (διαστήματα) πρέπει να είναι διπλανοί και συχνά (αλλά δεν απαιτείται να είναι) ίδιου μεγέθους.

Διάγραμμα Διασποράς

Ένα scatter plot (διάγραμμα διασποράς) είναι ένας τύπος γραφικής παράστασης ή μαθηματικού διαγράμματος που χρησιμοποιεί καρτεσιανές συντεταγμένες για την εμφάνιση τιμών για δύο τυπικές μεταβλητές για ένα σύνολο δεδομένων. Εάν τα σημεία είναι κωδικοποιημένα (χρώμα/σχήμα/μέγεθος), μπορεί να εμφανιστεί μία επιπλέον μεταβλητή. Τα δεδομένα εμφανίζονται ως μια συλλογή σημείων, καθένα από τα οποία έχει την τιμή μιας μεταβλητής που καθορίζει τη θέση στον οριζόντιο άξονα και την τιμή

της άλλης μεταβλητής που καθορίζει τη θέση στον κατακόρυφο άξονα. Χρησιμοποιείται πολύ συχνά για την οπτική απεικόνιση πιθανών συσχετίσεων μεταξύ μεταβλητών.

Διάγραμμα Γραμμών

Ένα line chart είναι ένας τύπος γραφήματος που εμφανίζει πληροφορίες ως μια σειρά σημείων δεδομένων που ονομάζονται «δείκτες» που συνδέονται με ευθύγραμμα τμήματα. Είναι ένας βασικός τύπος γραφήματος κοινός σε πολλούς τομείς. Είναι παρόμοιο με ένα scatter plot που αναφέρθηκε παραπάνω, εκτός από το ότι τα σημεία μέτρησης ταξινομούνται (συνήθως με βάση την τιμή του άξονα x) και ενώνονται με ευθύγραμμα τμήματα. Ένα line chart χρησιμοποιείται συχνά για να απεικονίσει μια τάση στα δεδομένα σε χρονικά διαστήματα - μια χρονοσειρά - επομένως η γραμμή συχνά σχεδιάζεται χρονολογικά.

Κατηγορικά Δεδομένα

Οι κατηγορικές μεταβλητές αντιπροσωπεύουν τύπους δεδομένων που μπορούν να χωριστούν σε ομάδες [BOSL12]. Παραδείγματα κατηγορικών μεταβλητών είναι η φυλή, το φύλο, η ηλικιακή ομάδα και το μορφωτικό επίπεδο. Ενώ οι δύο τελευταίες μεταβλητές μπορούν επίσης να ληφθούν υπόψη με αριθμητικό τρόπο χρησιμοποιώντας ακριβείς τιμές για την ηλικία και τον υψηλότερο βαθμό που ολοκληρώθηκε, είναι συχνό να κατηγοριοποιούνται τέτοιες μεταβλητές σε σχετικά μικρό αριθμό ομάδων.

Οι κατηγορικές μεταβλητές μπορεί να μην αναπαρίστανται με αριθμητική κλίμακα και μπορούν να δημιουργηθούν κατηγοριοποιώντας μια συνεχή ή διακριτή μεταβλητή. Για παράδειγμα, η αρτηριακή πίεση είναι ένα μέτρο της πίεσης που ασκείται στα τοιχώματα των αιμοφόρων αγγείων, μετρούμενο σε χιλιοστά υδραργύρου (Hg). Η αρτηριακή πίεση συνήθως μετράται συνεχώς και καταγράφεται με συγκεκριμένες μετρήσεις όπως 120/80 mmHg, αλλά συχνά αναλύεται χρησιμοποιώντας κατηγορίες όπως *χαμηλή, φυσιολογική, προϋπερτασική* και *υπερτασική*. Οι διακριτές μεταβλητές (αυτές που μπορούν να ληφθούν μόνο σε συγκεκριμένες τιμές εντός μιας περιοχής) μπορούν επίσης να ομαδοποιηθούν σε κατηγορικές μεταβλητές. Για παράδειγμα, μπορούμε να συλλέξουμε ακριβείς πληροφορίες για τον αριθμό των παιδιών ανά νοικοκυριό (0 παιδιά, 1 παιδί, 2 παιδιά, 3 παιδιά, κ.λπ.) αλλά να επιλέξουμε να ομαδοποιήσουμε αυτά τα δεδομένα σε κατηγορίες για σκοπούς ανάλυσης, όπως 0 παιδιά, 1– 2 παιδιά και 3 ή περισσότερα παιδιά. Αυτός ο τύπος ομαδοποίησης χρησιμοποιείται συχνά εάν υπάρχουν μεγάλοι αριθμοί κατηγοριών και ορισμένες από αυτές περιέχουν αραιά δεδομένα. Στην περίπτωση του αριθμού των παιδιών σε ένα νοικοκυριό, ένα σύνολο δεδομένων μπορεί

να περιλαμβάνει σχετικά λίγα νοικοκυριά με μεγάλο αριθμό παιδιών και οι χαμηλές συχνότητες σε αυτές τις κατηγορίες μπορεί να επηρεάσουν αρνητικά τη δύναμη της μελέτης ή να καταστήσουν αδύνατη τη χρήση ορισμένων αναλυτικών τεχνικών.

Chi-Squared test

Όταν μια στατιστική ανάλυση αφορά την σχέση μεταξύ δύο κατηγορικών μεταβλητών η κατανομή τους στα δεδομένα συνήθως παρουσιάζεσαι με την μορφή ενός RxC (contingency table). Το R στο R×C αναφέρεται σε σειρά και το C στη στήλη, και ένας πίνακας μπορεί να περιγραφεί από τον αριθμό των γραμμών και στηλών που περιέχει. Αυτό το είδος πίνακα εμφανίζει την κατανομή συχνότητας των μεταβλητών και χρησιμοποιείται σε μεγάλο βαθμό στην έρευνα, την επιχειρηματική ευφυΐα, τη μηχανική και την επιστημονική έρευνα. Αυτό το είδος πίνακα παρέχει μια βασική εικόνα της αλληλεπίδρασης μεταξύ δύο μεταβλητών και βοηθούν στην εύρεση αλληλεπιδράσεων μεταξύ των μεταβλητών αυτών χρησιμοποιώντας τεχνικές, όπως έλεγχο υπόθεσης. Όταν χρησιμοποιούμε έλεγχο υπόθεσης (hypothesis testing) με κατηγορικές μεταβλητές χρειάζεται ένας τρόπος να εκτιμήσουμε αν τα αποτελέσματα είναι στατιστικώς σημαντικά. Για RxC πίνακες η στατιστική τεχνική που συνήθως χρησιμοποιείται είναι το chi-squared τεστ, το οποίο έχει άμεση σχέση με την chi-squared κατανομή. Η δυνατότητα να συσχετιστεί ένα στατιστικό αποτέλεσμα με μία γνωστή κατανομή καθιστά εύκολο τον προσδιορισμό της πιθανότητας του αποτελέσματος. Ένα chi-squared τεστ είναι ένα τεστ στατιστικής υπόθεσης που είναι έγκυρο όταν το στατιστικό τεστ ακολουθεί την chi-squared κατανομή στο null hypothesis. Το Pearson chi-squared τεστ χρησιμοποιείται για να προσδιοριστεί εάν υπάρχει στατιστικά σημαντική διαφορά μεταξύ των αναμενόμενων συχνοτήτων και των παρατηρούμενων συχνοτήτων σε μία ή περισσότερες κατηγορίες ενός RxC πίνακα και αναφέρεται συχνά σε τεστ για τα οποία η κατανομή της στατιστικού τεστ προσεγγίζει την κατανομή chi-squared ασυμπτωτικά, πράγμα που σημαίνει ότι η κατανομή δειγματοληψίας (αν είναι αληθής η μηδενική υπόθεση) της στατιστικής δοκιμής προσεγγίζει μια κατανομή chi-squared όλο και περισσότερο όσο αυξάνεται το μέγεθος του δείγματος.

Φόρμουλα για τον υπολογισμό της τιμής ενός chi-squared test:

$$\chi^2 = \sum_{i=1,j=1}^{i=R,j=C} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

Η τιμή του chi-squared test συμβολίζεται με ελληνικό χ υψωμένο στο τετράγωνο, όπου O_{ij} η παρατηρούμενη τιμή του κελιού ij του πίνακα και E_{ij} η αναμενόμενη τιμή.

Για να ερμηνευτεί το αποτέλεσμα ενός chi-squared τεστ πρέπει να είναι γνωστοί οι βαθμοί ελευθερίας του.

Βαθμοί ελευθερίας για chi-squared τεστ:

$$(r-1)(c-1)$$

Όπου r ο αριθμός των γραμμών του RxC πίνακα και c ο αριθμός των στηλών

Έχοντας υπολογίσει την chi-squared τιμή και τους βαθμούς ελευθερίας μπορούμε να συμβουλευτούμε έναν chi-squared πίνακα για να δούμε αν η τιμή αυτή ξεπερνάει την κρίσιμη τιμή (συνήθως α=0.05) για την συγκεκριμένη κατανομή. Αν την ξεπερνά τότε έχουμε αρκετές αποδείξεις ώστε να απορρίψουμε την μηδενική υπόθεση ότι οι μεταβλητές είναι ανεξάρτητες. Συνήθως τα πακέτα λογισμικού επιστρέψουν και μία p-value τιμή μαζί με τα αποτελέσματα του τεστ και αν η τιμή αυτή είναι μικρότερη από την κρίσιμη τιμή τότε απορρίπτουμε την μηδενική υπόθεση.

Στατιστικά Συσχέτισης

Το πιο συχνό μέτρο συσχέτισης δυο μεταβλητών είναι ο συντελεστής συσχέτισης του Pearson ο οποίος απαιτεί οι μεταβλητές να μετρούνται σε διάστημα τιμών. Ωστόσο, έχουν αναπτυχθεί αρκετά μέτρα συσχέτισης για κατηγορικά (chi-squared test) και τακτικά δεδομένα, και ερμηνεύονται παρόμοια με το συντελεστή συσχέτισης του Pearson. Όπως και με τον συντελεστή συσχέτισης του Pearson, τα στατιστικά συσχέτισης είναι μόνο μέτρα συσχέτισης και δηλώσεις σχετικά με την αιτιότητα δεν μπορούν να υποστηρίζονται μόνο από έναν τέτοιο συντελεστή, ο οποίος είναι ένα μέτρο μιας παρατηρούμενης σχέσης, που δεν μπορεί από μόνος του να αποδείξει αιτιότητα. Πολλές μεταβλητές στον πραγματικό κόσμο έχουν ισχυρή συσχέτιση μεταξύ τους, ωστόσο αυτές οι σχέσεις μπορεί να οφείλονται στην τύχη, στην επιρροή άλλων μεταβλητών ή σε άλλες αιτίες που δεν έχουν ακόμη προσδιοριστεί. Ακόμα κι αν υπάρχει μια αιτιατή σχέση, η αιτιότητα μπορεί να είναι στην αντίθετη κατεύθυνση από αυτό που υποθέτουμε.

Ο συντελεστής συσχέτισης Kendall's tau-a

Ο συντελεστής συσχέτισης Kendall's tau-a (συμβολίζεται με το ελληνικό γράμμα τ), είναι ένα στατιστικό στοιχείο που χρησιμοποιείται για τη μέτρηση της τακτικής συσχέτισης μεταξύ δύο μετρούμενων μεγεθών και είναι ένα μέτρο συσχέτισης κατάταξης: η ομοιότητα των ταξινομήσεων των δεδομένων όταν ταξινομούνται με καθεμία από τις ποσότητες.

Ο συντελεστής συσχέτισης Kendall's tau-a ορίζεται ως:

$$\tau = \frac{(\alpha\rho\iota\theta\mu\acute{o}\varsigma\ \sigma\acute{\upsilon}\mu\phi\omega\nu\omega\nu\ \zeta\epsilon\upsilon\gamma\alpha\rho\iota\acute{\omega}\nu) - (\alpha\rho\iota\theta\mu\acute{\omega}\nu\ \alpha\sigma\acute{\upsilon}\mu\phi\omega\nu\omega\nu\ \zeta\epsilon\upsilon\gamma\alpha\rho\iota\acute{\omega}\nu)}{\binom{n}{2}}$$

Όπου $\binom{n}{2} = \frac{n(n-1)}{2}$ ο αριθμός των τρόπων επιλογής δύο στοιχείων από n στοιχεία.

Ένας πιο σαφής ορισμός για το συντελεστής συσχέτισης Kendall's tau-a είναι:

$$\tau = \frac{2}{n(n-1)} \sum_{i<j} sgn(x_i - x_j) sgn(y_i - y_j)$$

Όπου $\operatorname{sgn}(x) = \begin{cases} -1, & \alpha\nu\ x < 0 \\ 0, & \alpha\nu\ x = 0 \\ 1, & \alpha\nu\ x > 0 \end{cases}$

Για τον υπολογισμό σύμφωνων και ασύμφωνων ζευγών.

Ας υποθέσουμε ότι συγκρίνουμε εγγραφές σε σχέση με ένα συγκεκριμένο ζεύγος πεδίων (X_1, Y_1) και (X_2, Y_2). Σε ένα σύμφωνο ζεύγος, και τα δύο στοιχεία του ενός ζεύγους είναι είτε μεγαλύτερα, είτε ίσα, είτε μικρότερα από τα αντίστοιχα στοιχεία του άλλου ζεύγους. Σε ένα ασύμφωνο ζεύγος, τα δύο στοιχεία του ενός ζεύγους δεν είναι και τα δύο μεγαλύτερα, ίσα ή μικρότερα από τα αντίστοιχα του άλλου ζεύγους . Δηλαδή, σύμφωνο ζεύγος αν $X_i > X_j$ και $Y_i > Y_j$ ή $X_i < X_j$ και $Y_i < Y_j$ και ασύμφωνο αν $Xi > X_j$ και $Y_i < Y_j$ ή $X_i < X_j$ και $Y_i > Y_j$. Ο συντελεστής Kendall tau-a δεν λαμβάνει υπόψιν τις ισοβαθμίες στα ζεύγη. Στη συνέχεια αναφέρεται, ο συντελεστής Kendall tau-b ο οποίος λύνει το πρόβλημα αυτό.

Ο συντελεστής συσχέτισης Kendall's tau-b

Ο συντελεστής συσχέτισης Kendall's tau-b είναι ένα παρόμοιο μέτρο συσχέτισης που βασίζεται σε σύμφωνα και ασύμφωνα ζεύγη, προσαρμοσμένος όμως για τον αριθμό των ισοβαθμιών στις τάξεις.

$$\tau_b = \frac{P - Q}{\sqrt{(P + Q + X_0)(P + Q + Y_0)}}$$

Όπου,

- P = αριθμός σύμφωνων ζευγαριών
- Q = αριθμός ασύμφωνων ζευγαριών
- X_0 = αριθμός ζευγαριών που δεν ισοβαθμούν στο X

Y_0 = αριθμός ζευγαριών που δεν ισοβαθμούν στο Y

Ο συντελεστής συσχέτισης του Pearson

Ο συντελεστής συσχέτισης του Pearson είναι ένα μέτρο γραμμικής συσχέτισης μεταξύ δύο μεταβλητών σε επίπεδο διαστήματος. Οι συσχετίσεις υπολογίζονται συχνά κατά τη διάρκεια του διερευνητικού σταδίου των δεδομένων για να δούμε τι είδους σχέσεις έχουν οι διαφορετικές συνεχείς μεταβλητές μεταξύ τους και συχνά δημιουργούνται διαγράμματα διασποράς (που αναφέρθηκαν παραπάνω) για να εξεταστούν γραφικά αυτές οι σχέσεις. Ο συντελεστής του Pearson ορίζεται ως ο λόγος μεταξύ της συνδιακύμανσης δύο μεταβλητών και του γινόμενου των τυπικών αποκλίσεων τους και είναι ουσιαστικά μια κανονικοποιημένη μέτρηση της συνδιακύμανσης, έτσι ώστε το αποτέλεσμα να έχει πάντα μια τιμή μεταξύ -1 και 1. Όπως και με την ίδια τη συνδιακύμανση, το μέτρο μπορεί να αντικατοπτρίζει μόνο μια γραμμική συσχέτιση μεταβλητών και αγνοεί άλλους τύπους σχέσης ή συσχέτισης.

Ο συντελεστής συσχέτισης του Pearson ορίζεται ως:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - x)^2}\sqrt{\sum_{i=1}^{n}(y_i - \mathrm{y})^2}}$$

Όπου:

- n το δείγμα
- x_i, y_i είναι τα επιμέρους σημεία του δείγματος

- $\bar{x} = \frac{1}{n} x_i$ η μέση τιμή του δείγματος, ομοίως για το $\bar{y}$

Κάνοντας τις πράξεις:

$$r_{xy} = \frac{n \sum_{i=1}^{n} x_i \, y_i \; - \; \sum_{i=1}^{n} x_i \; \sum_{i=1}^{n} y_i}{\sqrt{n \sum_{i=1}^{n} {x_i}^2 \; - \left(\sum_{i=1}^{n} x_i\right)^2} \sqrt{n \sum_{i=1}^{n} {y_i}^2 \; - \left(\sum_{i=1}^{n} y_i\right)^2}}$$

$$r_{xy} = \frac{n \sum_{i=1}^{n} x_i \, y_i \; - \; n\bar{x}\bar{y}}{\sqrt{\sum_{i=1}^{n} {x_i}^2 \; - \; n\bar{x}^2} \sqrt{\sum_{i=1}^{n} {y_i}^2 \; - \; n\bar{y}^2}}$$

$$r_{xy} = \frac{1}{n-1} \sum_{i=1}^{n} \frac{x_i \; - \; \bar{x}}{s_x} \frac{y_i \; - \; \bar{y}}{S_y}$$

Όπου $s_x = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i \; - \; \bar{x})^2}$ η τυπική απόκλιση του δείγματος.

Ομοίως για το s_y.

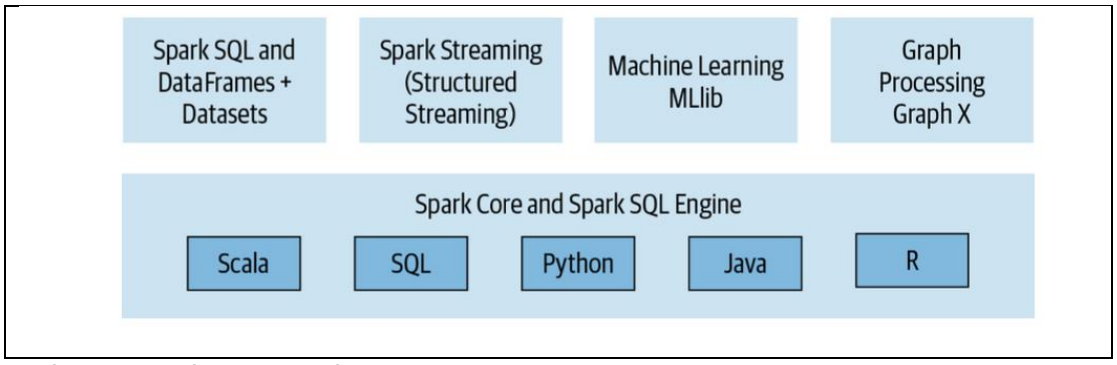
Να σημειωθεί ότι η συσχέτιση είναι μια συμμετρική σχέση, οπότε δεν χρειάζεται να δηλώσουμε ότι η μία μεταβλητή προκαλεί ως αποτέλεσμα την άλλη, μόνο ότι έχουμε παρατηρήσει μια σχέση μεταξύ τους.

Apache Spark

Το Apache Spark [Apache Spark, n.d.] είναι μια ενοποιημένη μηχανή σχεδιασμένη για μεγάλης κλίμακας κατανεμημένη επεξεργασία δεδομένων, σε εγκαταστάσεις σε κέντρα δεδομένων ή στο cloud. Το Spark παρέχει αποθήκευση στη μνήμη για ενδιάμεσους υπολογισμούς, καθιστώντας το πολύ πιο γρήγορο από το Apache Hadoop (Apache Hadoop, n.d.). Ενσωματώνει βιβλιοθήκες για μηχανική μάθηση (MLlib), SQL για διαδραστικά ερωτήματα (Spark SQL), επεξεργασία ροών δεδομένων (Structured Streaming) για αλληλεπίδραση με δεδομένα σε πραγματικό χρόνο και επεξεργασία γραφημάτων (GraphX).

Οι γλώσσες προγραμματισμού που υποστηρίζονται από το Spark είναι οι εξής: Scala, Java, Python και R. Οι προγραμματιστές εφαρμογών και οι επιστήμονες δεδομένων ενσωματώνουν το Spark στις εφαρμογές τους για γρήγορη αναζήτηση, ανάλυση και μετατροπή δεδομένων με κλιμακωσιμότητα. Οι εργασίες που εκπονούνται συχνότερα με το Spark περιλαμβάνουν εργασίες ETL και SQL σε μεγάλα σύνολα δεδομένων,

επεξεργασία δεδομένων ροής από αισθητήρες, IoT ή χρηματοοικονομικά συστήματα και εργασίες μηχανικής εκμάθησης.



Εικόνα 1. Επιμέρους συστήματα του Apache Spark και του API του. [DWTD20]

Spark SQL

Αυτή η βιβλιοθήκη λειτουργεί καλά με δομημένα δεδομένα. Δίνεται η δυνατότητα στον αναλυτή να διαβάσει δεδομένα που είναι αποθηκευμένα σε έναν πίνακα RDBMS ή από μορφές αρχείων με δομημένα δεδομένα (CSV, κείμενο, JSON, Avro, ORC, Parquet κ.λπ.) και στη συνέχεια να δημιουργήσει μόνιμους ή προσωρινούς πίνακες στο Spark. Επίσης, όταν χρησιμοποιούνται τα δομημένα API του Spark σε Java, Python, Scala ή R, μπορούν να υποβληθούν ερωτήματα τύπου SQL στα δεδομένα που μόλις διαβάστηκαν σε ένα Spark DataFrame.

Spark DataFrame

Τα Spark DataFrames είναι κατανεμημένοι πίνακες στη μνήμη με επώνυμες στήλες που ορίζουν ένα schema, όπου κάθε στήλη έχει έναν συγκεκριμένο τύπο δεδομένων: ακέραιος, συμβολοσειρά, πίνακας, χάρτης, πραγματικός, ημερομηνία, χρονική σήμανση κ.λπ. Όταν τα δεδομένα αναπαριστώνται ως ένας δομημένος πίνακας, δεν είναι μόνο εύκολο στην κατανόηση, αλλά και εύκολο να τα επεξεργαστούμε όταν πρόκειται για κοινές λειτουργίες πάνω σε γραμμές και στήλες. Τα DataFrames είναι αμετάβλητα και το Spark διατηρεί μια ιστορία από όλους τους μετασχηματισμούς που έχουν υποβληθεί σε αυτά. Δίνεται η δυνατότητα προσθήκης ή αλλαγής ονομάτων και τύπων των στηλών, δημιουργώντας νέα DataFrames ενώ διατηρούνται οι προηγούμενες εκδόσεις. Μια στήλη με όνομα σε ένα DataFrame και ο σχετικός τύπος δεδομένων του Spark μπορούν να δηλωθούν στο schema του. Ένα schema στο Spark ορίζεται από τα ονόματα των στηλών και των σχετικών τύπων δεδομένων για ένα DataFrame και τις περισσότερες φορές, τα schemata χρησιμοποιούνται όταν διαβάζονται δομημένα δεδομένα από μια εξωτερική πηγή δεδομένων (data sets).

Το Spark επιτρέπει τον ορισμό ενός schema με δύο τρόπους. Ο πρώτος είναι ο ορισμός μέσω προγραμματισμού και ο δεύτερος είναι η χρήση μιας συμβολοσειράς Data Definition Language (DDL), η οποία είναι πολύ πιο απλή και πιο εύκολη στην ανάγνωση. Οποιοσδήποτε από τους δύο τρόπους αυτούς θα παράγει το ίδιο αποτέλεσμα.

```scala
// In Scala
import org.apache.spark.sql.types._
val schema = StructType(Array(StructField("author", StringType, false),
  StructField("title", StringType, false),
  StructField("pages", IntegerType, false)))

# In Python
from pyspark.sql.types import *
schema = StructType([StructField("author", StringType(), False),
  StructField("title", StringType(), False),
  StructField("pages", IntegerType(), False)])
```

Εικόνα 2. Ορισμός σχήματος προγραμματιστικά. [DWTD20]

```scala
// In Scala
val schema = "author STRING, title STRING, pages INT"

# In Python
schema = "author STRING, title STRING, pages INT"
```

Εικόνα 3. Ορισμός σχήματος μέσω συμβολοσειράς (String). [DWTD20]

Spark Column

Ένα Spark Column είναι ένας τύπος στο API του Spark ο οποίος είναι μια κλάση με public μεθόδους που δίνουν την δυνατότητα εκτέλεσης διαφόρων εργασιών στης στήλες ενός DataFrame. Δίνεται η δυνατότητα απαρίθμησης όλων των στηλών με τα ονόματά τους και εκτέλεσης πράξεων στις τιμές τους χρησιμοποιώντας σχεσιακές ή υπολογιστικές εκφράσεις.

Spark Row

Ένα Spark Row είναι ένα γενικό αντικείμενο, που περιέχει ένα ή περισσότερα Spark αντικείμενα τύπου Column. Κάθε στήλη μπορεί να είναι του ίδιου τύπου δεδομένων (π.χ. ακέραιος ή συμβολοσειρά) ή μπορεί να έχει διαφορετικούς τύπους (ακέραιος, συμβολοσειρά, χάρτης, πίνακας, κ.λπ.). Επειδή το Row είναι ένα αντικείμενο στο Spark και μια διατεταγμένη συλλογή πεδίων, υπάρχει η δυνατότητα δημιουργίας μιας σειράς σε καθεμία από τις υποστηριζόμενες γλώσσες του Spark για απόκτηση πρόσβασης στα πεδία της με ένα ευρετήριο που ξεκινά από το 0.

Spark Dataset

Ξεκινώντας από το Spark 2.0, το Dataset αποκτά δύο ξεχωριστά χαρακτηριστικά API: ένα strongly typed API και ένα untyped API. Το DataFrame είναι ένα ψευδώνυμο για μια συλλογή γενικών αντικειμένων Dataset[Row], όπου μια γραμμή είναι ένα γενικό αντικείμενο JVM χωρίς τύπο. Το Dataset, αντίθετα, είναι μια συλλογή strongly typed αντικειμένων JVM, από μια κλάση που ορίζεται στη Scala ή στη Java. Τα DataSets έχουν νόημα μόνο σε Java και Scala, ενώ στην Python και R έχουν νόημα μόνο τα DataFrames. Αυτό οφείλεται στο γεγονός ότι η Python και η R δεν είναι ασφαλείς ως προς τον τύπο (type-safe) καθώς οι τύποι συνάγονται δυναμικά ή δημιουργούνται κατά τη διάρκεια της εκτέλεσης, όχι κατά τη διάρκεια του χρόνου μεταγλώττισης (compile-time). Όπως υπάρχει η δυνατότητα εκτέλεσης μετασχηματισμών και ενεργειών στα DataFrames, έτσι υποστηρίζεται και με τα Datasets.

Resilient Distributed Datasets

Τα RDD ή Resilient Distributed Datasets είναι η θεμελιώδης δομή δεδομένων του Spark. Είναι η συλλογή αντικειμένων που είναι ικανή να αποθηκεύει τα δεδομένα κατανεμημένα στους πολλαπλούς κόμβους της συστάδας υπολογιστών επιτρέποντας την παράλληλη επεξεργασία. Τα RDD είναι ανεκτικά σε σφάλματα αν εκτελεστούν πολλαπλοί μετασχηματισμοί σε αυτά και στη συνέχεια για οποιονδήποτε λόγο αποτύχει οποιοσδήποτε κόμβος. Το RDD, σε αυτήν την περίπτωση, είναι σε θέση να ανακάμψει αυτόματα. Τα Datasets και DataFrames στηρίζονται πάνω στο RDD για την λειτουργία τους παρέχοντας ένα επίπεδο αφαίρεσης στον αναλυτή το οποίο πολλές φορές είναι πολύ χρήσιμο.

Πότε χρησιμοποιούμε DataFrames, πότε Datasets και πότε RDDs

Σε πολλές περιπτώσεις, ανάλογα με τις γλώσσες στις οποίες δουλεύουμε, καθ'ένα από τα Dataset, DataFrame μπορεί να χρησιμοποιηθεί, αλλά υπάρχουν ορισμένες περιπτώσεις όπου το ένα είναι προτιμότερο από το άλλο.

Ακολουθούν κάποια παραδείγματα: [DWTD20]

- Εάν θέλουμε να πούμε στο Spark τι να κάνει, όχι πώς να το κάνει, χρησιμοποιούμε DataFrames ή Datasets αλλιώς χρησιμοποιούμε RDDs.
- Εάν θέλουμε πλούσια σημασιολογία, αφαιρέσεις υψηλού επιπέδου και τελεστές DSL (Domain-Specific Language), χρησιμοποιούμε Data-Frames ή Datasets.

- Εάν θέλουμε αυστηρή ασφάλεια τύπων, χρόνου μεταγλώττισης και δεν μας πειράζει να δημιουργήσουμε πολλαπλές κλάσεις για ένα συγκεκριμένο Dataset[T], χρησιμοποιούμε Datasets.
- Εάν η επεξεργασία απαιτεί εκφράσεις υψηλού επιπέδου, φίλτρα, χάρτες, συναθροίσεις, υπολογιστικούς μέσους όρους ή αθροίσματα, ερωτήματα SQL, πρόσβαση στη στήλη ή χρήση σχεσιακών τελεστών σε ημιδομημένα δεδομένα, χρησιμοποιούμε DataFrames ή Datasets.
- Εάν η επεξεργασία απαιτεί σχεσιακούς μετασχηματισμούς παρόμοιους με ερωτήματα τύπου SQL, χρησιμοποιούμε DataFrames.
- Εάν χρησιμοποιούμε R, χρησιμοποιούμε DataFrames.
- Εάν χρησιμοποιούμε Python, χρησιμοποιούμε DataFrames ενώ χρησιμοποιούμε RDDs εάν χρειάζεται περισσότερος έλεγχος πάνω στα δεδομένα.
- Εάν θέλουμε χώρο και απόδοση ταχύτητας, χρησιμοποιούμε DataFrames.

Αξίζει να σημειωθεί ότι δίνεται η δυνατότητα εύκολης μετακίνησης μεταξύ DataFrames, Datasets και RDDs κατά βούληση χρησιμοποιώντας μια απλή κλήση μεθόδου του API (df.rdd), ωστόσο, αυτό έχει ένα υπολογιστικό κόστος και θα πρέπει να αποφεύγεται εκτός εάν είναι απαραίτητο.

Spark MLlib

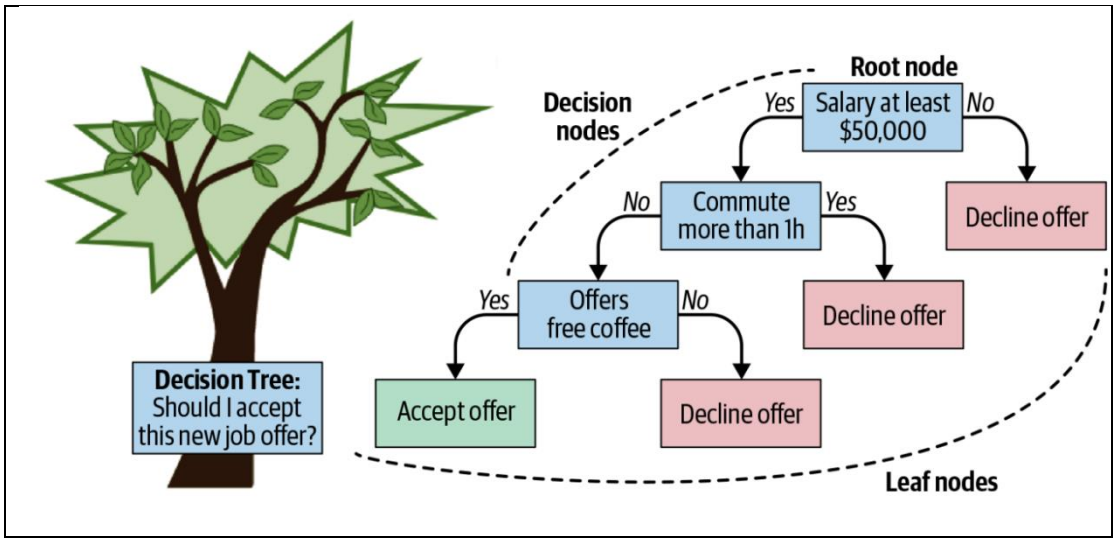
Το Spark συνοδεύεται από μια βιβλιοθήκη που περιέχει κοινούς αλγόριθμους μηχανικής μάθησης (ML) που ονομάζεται MLlib. Από την πρώτη κυκλοφορία του Spark, η απόδοση αυτής της βιβλιοθήκης έχει βελτιωθεί σημαντικά λόγω των υποκείμενων βελτιώσεων της μηχανής του Spark 2.x. Η MLlib παρέχει πολλούς δημοφιλείς αλγόριθμους μηχανικής μάθησης που είναι κατασκευασμένοι πάνω σε API υψηλού επιπέδου που βασίζονται σε DataFrame για τη δημιουργία μοντέλων.

Στην πραγματικότητα, το Spark διαθέτει δύο πακέτα μηχανικής μάθησης: spark.mllib και spark.ml. Το spark.mllib είναι το αρχικό API μηχανικής εκμάθησης, που βασίζεται στο RDD API, ενώ το spark.ml είναι το νεότερο API, που βασίζεται σε DataFrames. Αυτά τα API επιτρέπουν την εξαγωγή ή τον μετασχηματισμό χαρακτηριστικών, τη δημιουργία αγωγών (pipelines) (για εκπαίδευση και αξιολόγηση) και τη διατήρηση μοντέλων (για αποθήκευση και επαναφόρτωσή τους) κατά την ανάπτυξη. Πρόσθετα βοηθητικά προγράμματα περιλαμβάνουν τη χρήση κοινών πράξεων γραμμικής άλγεβρας και στατιστικών.

Decision trees

Ένα δέντρο αποφάσεων είναι μια σειρά κανόνων αν-τότε-αλλιώς που δημιουργούνται από τα δεδομένα με την διαδικασία της εκμάθησης για εργασίες κατηγοριοποίησης ή παλινδρόμησης και χρησιμοποιούνται ευρέως επειδή είναι εύκολο να ερμηνευθούν, χειρίζονται αριθμητικά και κατηγορικά χαρακτηριστικά και δεν απαιτούν κλιμάκωση χαρακτηριστικών.

Αναλυτικότερα, ας υποθέσουμε ότι προσπαθούμε να δημιουργήσουμε ένα μοντέλο για να προβλέψουμε εάν κάποιος θα αποδεχτεί ή όχι μια προσφορά εργασίας με τα χαρακτηριστικά να περιλαμβάνουν μισθό, χρόνο μετακίνησης, δωρεάν καφέ κ.λπ.



Εικόνα 4. Παράδειγμα ενός Δέντρου αποφάσεων. [DWTD20]

Ο κόμβος στην κορυφή του δέντρου ονομάζεται «ρίζα» του δέντρου επειδή είναι το πρώτο χαρακτηριστικό στο οποίο γίνεται ο πρώτος διαχωρισμός των δεδομένων. Αυτή η δυνατότητα θα πρέπει να προσφέρει τον πιο κατατοπιστικό διαχωρισμό — σε αυτήν την περίπτωση, εάν ο μισθός είναι μικρότερος από $50.000, τότε η πλειοψηφία των υποψηφίων θα αρνηθεί την προσφορά εργασίας. Ο κόμβος "Απόρριψη προσφοράς" είναι γνωστός ως "κόμβος φύλλου", καθώς δεν υπάρχουν άλλες διασπάσεις που προέρχονται από αυτόν τον κόμβο και είναι στο τέλος ενός κλάδου. Ωστόσο, αν ο μισθός που προσφέρεται είναι μεγαλύτερος από $50.000, προχωράμε στο επόμενο πιο πληροφοριακό χαρακτηριστικό στο δέντρο αποφάσεων, που σε αυτήν την περίπτωση είναι ο χρόνος μετακίνησης. Ακόμα κι αν ο μισθός είναι πάνω από $50.000, αν η μετακίνηση είναι μεγαλύτερη από μία ώρα, τότε η πλειονότητα των ανθρώπων θα αρνηθεί την προσφορά εργασίας. Το τελευταίο χαρακτηριστικό στο μοντέλο μας είναι ο δωρεάν καφές. Σε αυτήν την περίπτωση, το δέντρο αποφάσεων δείχνει ότι εάν ο μισθός είναι μεγαλύτερος από $50.000, η μετακίνηση είναι μικρότερη από μία ώρα και υπάρχει

δωρεάν καφές, τότε η πλειοψηφία των ανθρώπων θα αποδεχτεί την προσφορά εργασίας.

Η εκμάθηση του δέντρου αποφάσεων είναι μια μέθοδος που χρησιμοποιείται συνήθως στην εξόρυξη δεδομένων. Ο στόχος είναι να δημιουργηθεί ένα μοντέλο που να προβλέπει την τιμή μιας μεταβλητής στόχου με βάση πολλές μεταβλητές εισόδου. Ένα δέντρο αποφάσεων είναι μια απλή αναπαράσταση για την ταξινόμηση παραδειγμάτων. Κάθε στοιχείο της ταξινόμησης μπορεί να ανήκει σε μια κατηγορία που ονομάζεται κλάση. Ένα δέντρο αποφάσεων ή ένα δέντρο ταξινόμησης είναι ένα δέντρο στο οποίο κάθε εσωτερικός (μη φύλλο) κόμβος επισημαίνεται με ένα χαρακτηριστικό εισόδου. Τα βέλη που προέρχονται από έναν κόμβο που έχει επισημανθεί με ένα χαρακτηριστικό εισόδου επισημαίνονται με καθεμία από τις πιθανές τιμές του χαρακτηριστικού στόχου ή το βέλος οδηγεί σε έναν δευτερεύοντα κόμβο απόφασης σε ένα διαφορετικό χαρακτηριστικό εισόδου. Κάθε φύλλο του δέντρου επισημαίνεται με μια κλάση ή μια κατανομή πιθανοτήτων στις κλάσεις, υποδηλώνοντας ότι το σύνολο δεδομένων έχει ταξινομηθεί από το δέντρο είτε σε μια συγκεκριμένη κατηγορία είτε σε μια συγκεκριμένη κατανομή πιθανοτήτων. Ένα δέντρο χτίζεται με το διαχωρισμό του συνόλου δεδομένων, που αποτελεί τον κόμβο ρίζας του δέντρου, σε υποσύνολα — τα οποία αποτελούν τα διαδοχικά παιδιά. Ο διαχωρισμός βασίζεται σε ένα σύνολο κανόνων διαχωρισμού τα οποία βασίζονται σε χαρακτηριστικά κατηγοριοποίησης. Αυτή η διαδικασία επαναλαμβάνεται σε κάθε παραγόμενο υποσύνολο με αναδρομικό τρόπο που ονομάζεται αναδρομική κατάτμηση. Η αναδρομή ολοκληρώνεται όταν το υποσύνολο σε έναν κόμβο έχει όλες τις ίδιες τιμές της μεταβλητής στόχου ή όταν ο διαχωρισμός δεν προσθέτει επιπλέον όφελος ακρίβειας στις προβλέψεις. Αυτή η διαδικασία κατασκευής δέντρων απόφασης από πάνω προς τα κάτω είναι ένα παράδειγμα ενός άπληστου αλγορίθμου και είναι η πιο κοινή στρατηγική για την εκμάθηση δέντρων αποφάσεων.

Στην εξόρυξη δεδομένων, τα δέντρα αποφάσεων μπορούν επίσης να περιγράφουνε ως ο συνδυασμός μαθηματικών και υπολογιστικών τεχνικών που βοηθούν στην περιγραφή, την κατηγοριοποίηση και τη γενίκευση ενός δεδομένου συνόλου δεδομένων.

Τα δεδομένα οργανώνονται σε εγγραφές της μορφής:

$$(x, Y) = (x_1, x_2, x_3, \ldots, x_k, Y)$$

Η εξαρτημένη μεταβλητή Y, είναι η μεταβλητή στόχος που προσπαθούμε να κατανοήσουμε ή να ταξινομήσουμε. Το διάνυσμα x αποτελείται από τα χαρακτηριστικά (features), x_1, x_2, x_3 κ.λπ., που χρησιμοποιούνται για την διαδικασία αυτή.

Οι αλγόριθμοι για την κατασκευή δέντρων αποφάσεων συνήθως λειτουργούν από πάνω προς τα κάτω, επιλέγοντας μια μεταβλητή σε κάθε βήμα που διαχωρίζει καλύτερα το σύνολο των στοιχείων. Διαφορετικοί αλγόριθμοι χρησιμοποιούν διαφορετικές μετρήσεις για τη μέτρηση της "καλύτερης" μεταβλητής και γενικά μετρούν την ομοιογένεια της μεταβλητής στόχου εντός των υποσυνόλων. Αυτές οι μετρήσεις εφαρμόζονται σε κάθε υποψήφιο υποσύνολο και οι τιμές που προκύπτουν συνδυάζονται (π.χ. υπολογίζονται κατά μέσο όρο) για να παρέχουν ένα μέτρο της ποιότητας του διαχωρισμού.

Η πρόσμειξη (impurity) ενός κόμβου είναι ένα μέτρο της ομοιογένειας των ετικετών στον κόμβο, δηλαδή ένα μέτρο που επιτρέπει στο αλγόριθμο να αξιολογεί την ποιότητα του διαχωρισμού των δεδομένων στον κόμβο. Το spark.mllib [Apache Spark, n.d.] υποσύστημα υποστηρίζει δέντρα αποφάσεων για δυαδική και πολλαπλή κατηγοριοποίηση, υποστηρίζοντας τόσο συνεχή όσο και κατηγορικά δεδομένα. Η υλοποίηση χωρίζει τα δεδομένα κατά σειρές, επιτρέποντας κατανεμημένη εκπαίδευση με εκατομμύρια Spark κόμβους για πολύπλοκα μοντέλα. Στο Spark επίσης παρέχονται δύο μέτρα (Gini και Entropy) για τον υπολογισμό της πρόσμειξης για κατηγοριοποίηση, σε κάθε κόμβο. Η εσωτερική λειτουργία και των δύο μεθόδων είναι πολύ παρόμοια καθώς και οι δύο χρησιμοποιούνται για τον υπολογισμό της δυνατότητας/διαίρεσης μετά από κάθε νέο διαχωρισμό σε κάθε κόμβο.

Η πρόσμειξη Gini είναι ένα μέτρο για το πόσο συχνά ένα τυχαία επιλεγμένο στοιχείο από το σύνολο θα χαρακτηριζόταν λανθασμένα εάν είχε κατηγοριοποιηθεί τυχαία σύμφωνα με την κατανομή των ετικετών στο υποσύνολο. Η ελάχιστη τιμή του Gini είναι μηδέν όταν όλες οι περιπτώσεις στον κόμβο εμπίπτουν σε μια και μοναδική κατηγορία στόχου και μεγαλύτερη τιμή το 0.5 όταν όλες οι κατηγορίες στόχου είναι ισοπίθανες με βάση τα δεδομένα του κόμβου.

Συνάρτηση για το Gini impurity:

$$\sum_{i=1}^{C} f_i(1 - f_i) = 1 - \sum_{i=1}^{C} {f_i}^2$$

Όπου f_i, η συχνότητα της ετικέτας i σε έναν κόμβο και το C είναι ο αριθμός των μοναδικών ετικετών.

Αναλυτικότερα και την συνάρτηση της πρόσμειξης Gini, έστω ότι έχουμε δύο ζάρια με πλευρές m και η πιθανότητα να εμφανιστεί η πλευράς i είναι fi, τότε η πιθανότητα διπλής τέτοια ζαριάς είναι $\sum {f_i}^2$.

Έτσι, 1-$\sum {f_i}^2$ είναι η πιθανότητα αποτελεσμάτων διαφορετικών τιμών ζαριών. Διαφορετικά, η πιθανότητα, να έχουμε i ρίψη ακολουθούμενη από j είναι fifj. Αθροίζοντας όλες τις πιθανότητες, με το $i \neq j$, λαμβάνουμε την πιθανότητα των αποτελεσμάτων διαφορετικών τιμών ζαριών: $\sum f_i f_j$, και η συνάρτηση αποδεικνύεται. Όσον αφορά τώρα το πρώτο επιχείρημα, αν γίνει ρίψη ενός ζαριού με m πλευρές, υπάρχει μια πιθανότητα fi η πλευρά i να εμφανιστεί. Ας υποθέσουμε όμως ότι πρέπει να μαντέψουμε την τιμή του ζαριού ρίχνοντας ένα πανομοιότυπο ζάρι. Η πιθανότητα να μαντέψουμε λάθος, υπό τον όρο ότι η τιμή i είναι έγκυρη, είναι $1-fi$. Άρα, η πιθανότητα να μαντέψουμε λάθος, αθροίζοντας τις πιθανές τιμές, είναι $\sum f_i(1 - f_i)$.

Η ελάχιστη τιμή της μετρικής Entropy είναι μηδέν όταν όλες οι περιπτώσεις στον κόμβο εμπίπτουν σε μια και μοναδική κατηγορία στόχου και μεγαλύτερη τιμή τον άσσο όταν όλες οι περιπτώσεις στον κόμβο εμπίπτουν στην ίδια και μοναδική κατηγορία στόχου.

Συνάρτηση για το Entropy impurity:

$$\sum_{i=1}^{C} -f_i log_2(f_i)$$

Όπου f_i, η συχνότητα της ετικέτας i σε έναν κόμβο και το C είναι ο αριθμός των μοναδικών ετικετών.

2.3 Ανάλυση απαιτήσεων

Στην υποενότητα αυτή παρατίθενται τα User Stories για την εργαλείου. Καθώς το εργαλείο λειτουργεί ως επί το πλείστον αυτόματα και για το λόγο αυτό τα User Stories είναι λίγα και ουσιαστικά παρατίθενται οι ενέργειες που πρέπει να κάνει ο αναλυτής για να ξεκινήσει η διαδικασία της αυτόματης παραγωγής της αναφοράς για το σύνολο δεδομένων που θα εισάγει ως είσοδο στο εργαλείο αυτό.

User Stories

- [US1] Ως αναλυτής, θα πρέπει το σύστημα να μου παρέχει την δυνατότητα να εγγράψω ένα νέο σύνολο δεδομένων στο σύστημα δηλώνοντας τα πεδία του αρχείου και τον τύπο τους επιτρέποντας έτσι στο εργαλείο να γνωρίζει το schema του συνόλου δεδομένων για να το επεξεργαστεί.
- [US2] Ως αναλυτής, θα πρέπει το σύστημα να μου παρέχει την δυνατότητα δηλώσω κανόνες labeling για μια κολόνα και το σύστημα να δημιουργήσει την νέα κολόνα επιτρέποντας έτσι την ευκολότερη διαχείριση των δεδομένων και χρήση τους από το εργαλείο.
- [US3] Ως αναλυτής, θα πρέπει το σύστημα να μου παρέχει την δυνατότητα του αυτόματου υπολογισμού του στατιστικού προφίλ του συνόλου δεδομένων για την γρήγορη των λήψη βασικών στατιστικών ιδιοτήτων του.
- [US4] Ως αναλυτής, θα πρέπει το σύστημα να μου παρέχει την δυνατότητα να εγγράψω το σύνολο δεδομένων στο δίσκο εκ νέου μετά από πιθανή εισαγωγή νέων κολόνων ή έπειτα από άλλες τροποποιήσεις του αρχικού συνόλου δεδομένων για πιθανή μετέπειτα επεξεργασία σε άλλα εργαλεία ή εκ νέου επεξεργασία στο εργαλείο αυτό.
- [US5] Ως αναλυτής, θα πρέπει το σύστημα να μου παρέχει την δυνατότητα να εξάγω μια αναφορά με τα ευρήματα που υπολόγισε το σύστημα στο δίσκο για την ανάγνωση των αποτελεσμάτων.

Κεφάλαιο 3. Σχεδίαση & Υλοποίηση

Στο συγκεκριμένο κεφάλαιο περιγράφονται οι λειτουργείες του συστήματος για τη λύση του προβλήματος της αυτόματης εξαγωγής στατιστικών για σύνολα δεδομένων με τους μηχανισμούς του εργαλείου που αναπτύχθηκε χρησιμοποιώντας το Apache Spark, όπως προσδιορίστηκε στα προηγούμενα κεφάλαια, και επίσης δίνονται αναλυτικά κάποιες βασικές λειτουργίες για την καλύτερη κατανόηση του συστήματος. Επιπλέον περιγράφεται η ανάλυση και ο σχεδιασμός του συστήματος με την χρήση UML διαγραμμάτων.

3.1 Ορισμός προβλήματος και επίλυση

Όπως αναφέρθηκε σε προηγούμενη ενότητα, η ανάγκη για ένα σύστημα αυτόματης παραγωγής προφίλ στατιστικών και ανάλυσης δεδομένων είναι μεγάλη καθώς υπάρχοντα εργαλεία απαιτούν διαδραστικότητα με τον αναλυτή ακόμα και για πολύ απλές εργασίες και πολύ συχνά αναλυτές της επιστήμης των Δεδομένων χρησιμοποιούν τέτοια εργαλεία για την εξαγωγή στατιστικών συμπερασμάτων για τα δεδομένα υπό εξέταση, όπως το Orange, Tableau κλπ. ή χρήση εργαλείων όπως Python, Jupyter Notebook κ.α. Το εργαλείο που αναπτύχθηκε και ονομάστηκε Pythia (ήταν το όνομα της αρχιερείας του Ναού του Απόλλωνα στους Δελφούς. Στους Δελφούς υπηρέτησε ως μάντισσα με αποτέλεσμα το μέρος να μείνε γνωστό ως το Μαντείο των Δελφών) προσπαθεί να χτίσει τα θεμέλια ενός συστήματος άκρως επεκτάσιμου και γρήγορου για αυτόματη παραγωγή στατιστικών προφίλ συνόλων δεδομένων που μπορεί να χρησιμοποιηθεί σε υπάρχοντα συστήματα υλοποιημένα σε Java, καθώς το ίδιο είναι υλοποιημένο σε Java και διανέμεται σε ένα πακέτο jar κάνοντας έτσι την εγκατάσταση και τη χρήση του σε άλλα συστήματα πολύ εύκολη.

Προτού ξεκινήσουμε την περιγραφή του συστήματος θα αναφερθούμε στις δυνατότητες του συστήματος.

Περιγραφικά Στατιστικά (Descriptive Statistics): Οποιοδήποτε σύνολο δεδομένων μπορεί να περιγραφεί από αυτά τα στατιστικά και είναι το πρώτο βήμα που συνήθως κοιτάμε σε ένα σύνολο δεδομένων. Το Pythia υπολογίζει απολύτως αυτόματα μέση τιμή,

διάμεσο, τυπική απόκλιση, μέγιστη και ελάχιστη τιμή και πλήθος τιμών για όλες τις κολόνες ακόμα και για μη αριθμητικά δεδομένα (φυσικά τα στατιστικά αυτά ερμηνεύονται διαφορετικά) χρησιμοποιώντας το Spark δίνοντας ταχύτατα αποτελέσματα ακόμα και για πολύ ογκώδη αρχεία συνόλων δεδομένων και δημιουργεί το πρώτο προφίλ στατιστικών.

Υπολογισμός συσχετίσεων κάθε κολόνας με κάθε άλλη (All pairs correlations): Άλλη μια σημαντική μετρική που κοιτάμε στα σύνολα δεδομένων είναι η συσχέτιση των κολόνων μεταξύ τους, καθώς είναι κάτι που σχεδόν πάντα δείχνει θετικά αποτελέσματα συσχέτισης σε κάποιες κολόνες και είναι πολύ χρήσιμο σε περιπτώσεις εφαρμογής αλγορίθμων για προβλέψεις καθώς τα αποτελέσματα των προβλέψεων (ακρίβεια) ενδέχεται να είναι πολύ ικανοποιητικά. Το Pythia χρησιμοποιεί το πακέτο στατιστικών του Spark που παρέχει έναν αλγόριθμο υπολογισμού της συσχέτισης δύο κολόνων με την χρήση της μεθόδου του Pearson ή του Spearman. Το Pythia χρησιμοποιεί τη μέθοδο Pearson που αναλύθηκε παραπάνω, η οποία λαμβάνει τα ονόματα των κολόνων και παράγει το αποτέλεσμα.

Υπολογισμός labeled κολόνας από αριθμητική κολόνα (Labeling system): Αν ο αναλυτής έχει δηλώσει κανόνες labeling για μια στήλη, το σύστημα χαρακτηρίζει τις τιμές αυτής της στήλης και παράγει την νέα labeled στήλη. Αυτό είναι χρήσιμο καθώς δίνεται η δυνατότητα εφαρμογής αλγορίθμων για Δέντρα αποφάσεων, έλεγχο υπόθεσης κ.α. Το Pythia ορίζει με την βοήθεια του αναλυτή, με έναν οργανωμένο τρόπο, ένα σύνολο από κανόνες για την παραγωγή της καινούργιας στήλης και στη συνέχεια απολύτως αυτόματα δημιουργείται η στήλη αυτή και γίνεται εφαρμογή αλγόριθμου για την δημιουργία ενός Δένδρου απόφασης, με την βοήθεια του Spark και συγκεκριμένα με τη βιβλιοθήκη MLlib που αναλύθηκε παραπάνω.

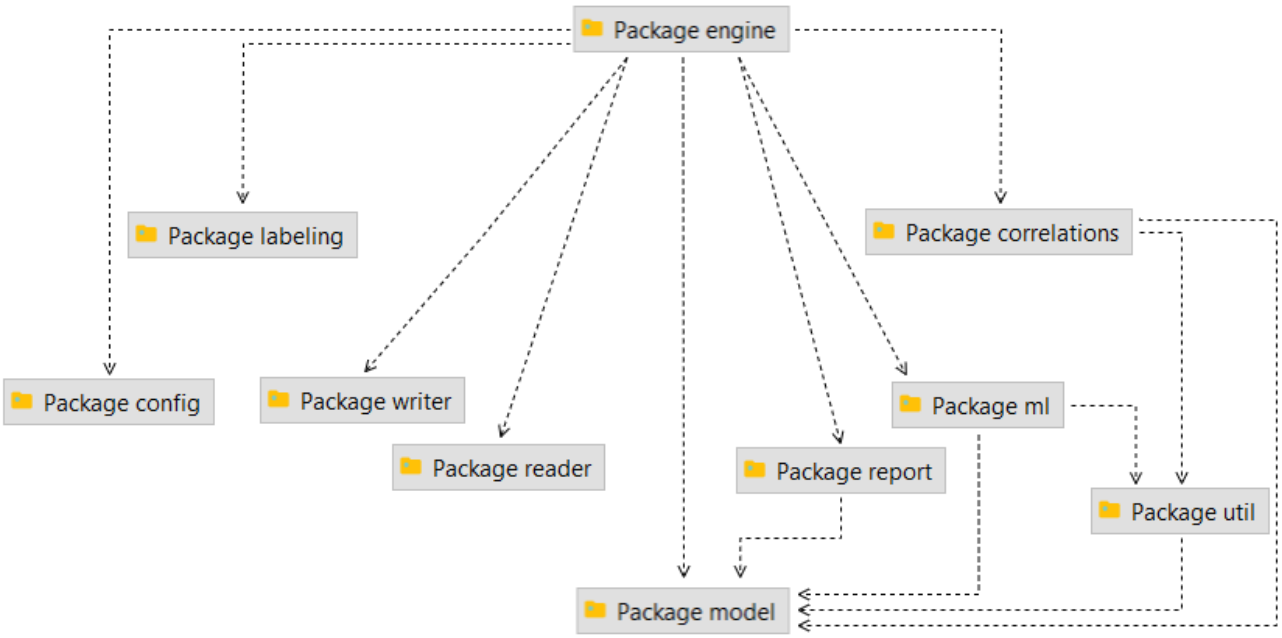
Εξαγωγή ευρημάτων σε αρχείο στο δίσκο (Report system): Αφού ολοκληρωθεί η διαδικασία του υπολογισμού του προφίλ του συνόλου δεδομένων εισόδου, ο αναλυτής έχει την δυνατότητα να εξάγει τα ευρήματα σε μορφή απλού αρχείου κειμένου ή JSON αρχείου. Το JSON (JavaScript Object Notation) είναι ένας τρόπος έκφρασης πληροφοριών που είναι συνήθως εύκολο να κατανοηθεί. Μπορεί να εκφράσει πληροφορίες όπως το XML και βασίζεται στο συντακτικό της γλώσσας προγραμματισμού JavaScript. Ωστόσο, το JSON είναι πιο αυστηρό από το XML και φυσικά από απλό κείμενο και είναι πολύ

εύκολο να χρησιμοποιηθεί στη συνέχεια από έναν αναλυτή για περαιτέρω επεξεργασίας και σε άλλες γλώσσες προγραμματισμού π.χ Python.

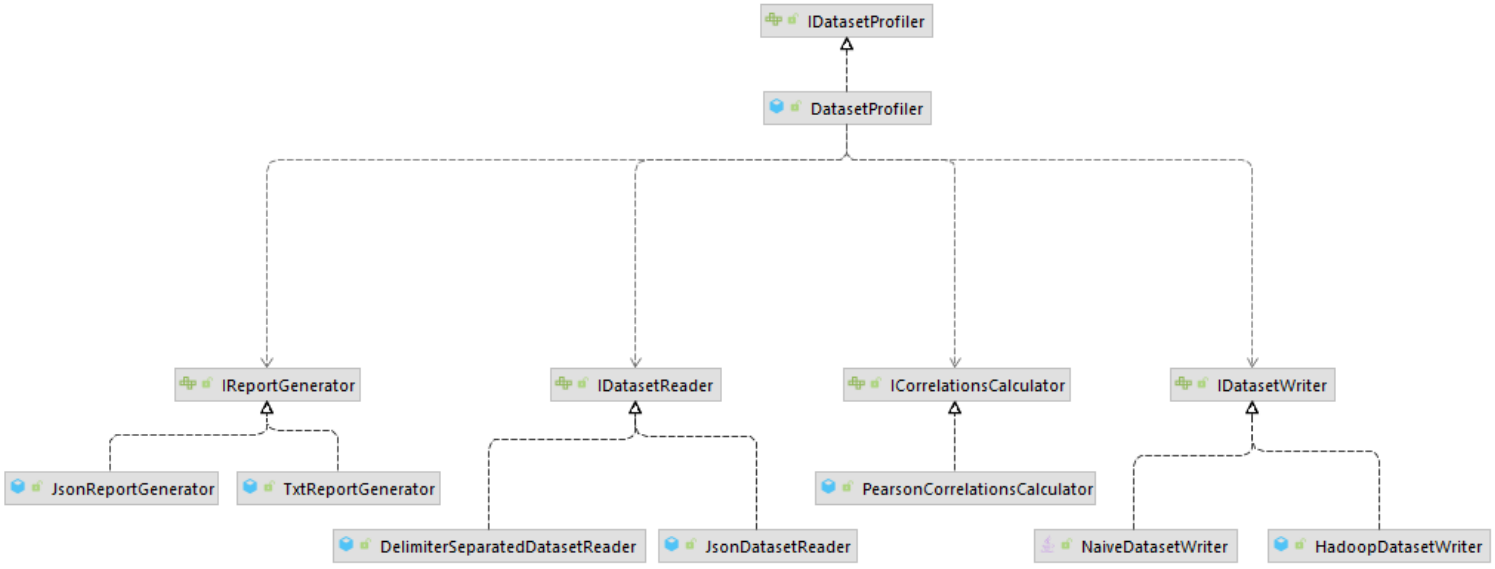
Εξαγωγή συνόλου δεομένων στο δίσκο έπειτα από την τροποποίηση του (Writer system): Τέλος, το σύστημα δίνει την δυνατότητα εξαγωγής του συνόλου δεδομένων εκ νέου στο δίσκο, στην μορφή ενός CSV αρχείου.

3.2 Σχεδίαση και αρχιτεκτονική λογισμικού

Όπως αναφέρθηκε στα προηγούμενα κεφάλαια, ο στόχος του συγκεκριμένου λογισμικού είναι η κάλυψη της συχνής ανάγκης στην επιστήμη των δεδομένων για εξαγωγή στατιστικών ενός συνόλου δεδομένων και η συγκεντρωτική εμφάνιση των στατιστικών αυτών στον αναλυτή με την μορφή μιας αναφοράς. Για την υλοποίηση του λογισμικού σχεδιάστηκαν και υλοποιήθηκαν οι κατάλληλες κλάσεις οι οποίες έχουν χωριστεί σε 10 πακέτα συνολικά: config, correlations, engine, labeling, ml, model, reader, report, util και writer τα οποία αναλύονται παρακάτω.

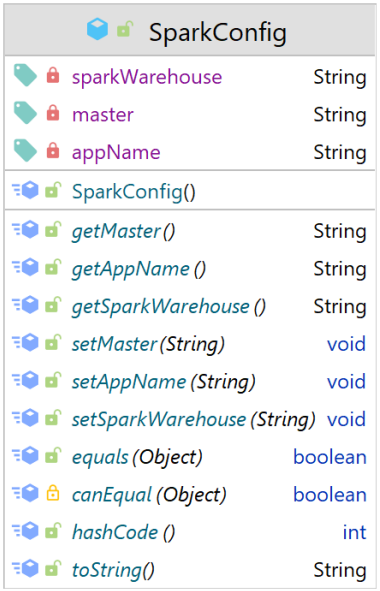


Εικόνα 5. Διάγραμμα πακέτων του Pythia



Εικόνα 6. Διάγραμμα UML των βασικών Interfaces του εργαλείου

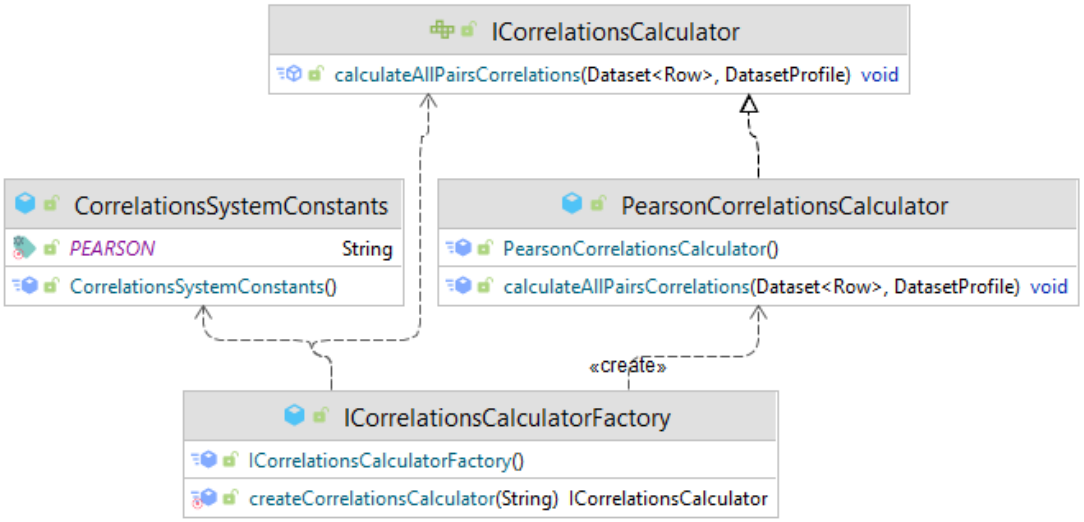
3.2.1 Πακέτο config



Εικόνα 7. Διάγραμμα UML του πακέτου config

Το συγκεκριμένο πακέτο περιέχει μια και μοναδική κλάση υπεύθυνη για την ρύθμιση των παραμέτρων του Spark οι οποίες είναι απαραίτητες για την εκκίνηση ενός Spark Session. Οι παράμετροι αυτές είναι αποθηκευμένες σε ένα αρχείο τύπου properties με όνομα spark.properties στον φάκελο src/main/resources/spark.properties. "Ετσι είναι πιο εύκολη η διαχείριση των παραμέτρων σε περίπτωση που χρειαστεί κάποια αλλαγή στο μέλλον. Στην τρέχουσα υλοποίηση το Spark έχει ρυθμιστεί να τρέχει τοπικά και να χρησιμοποιεί όλους τους πόρους του επεξεργαστή.

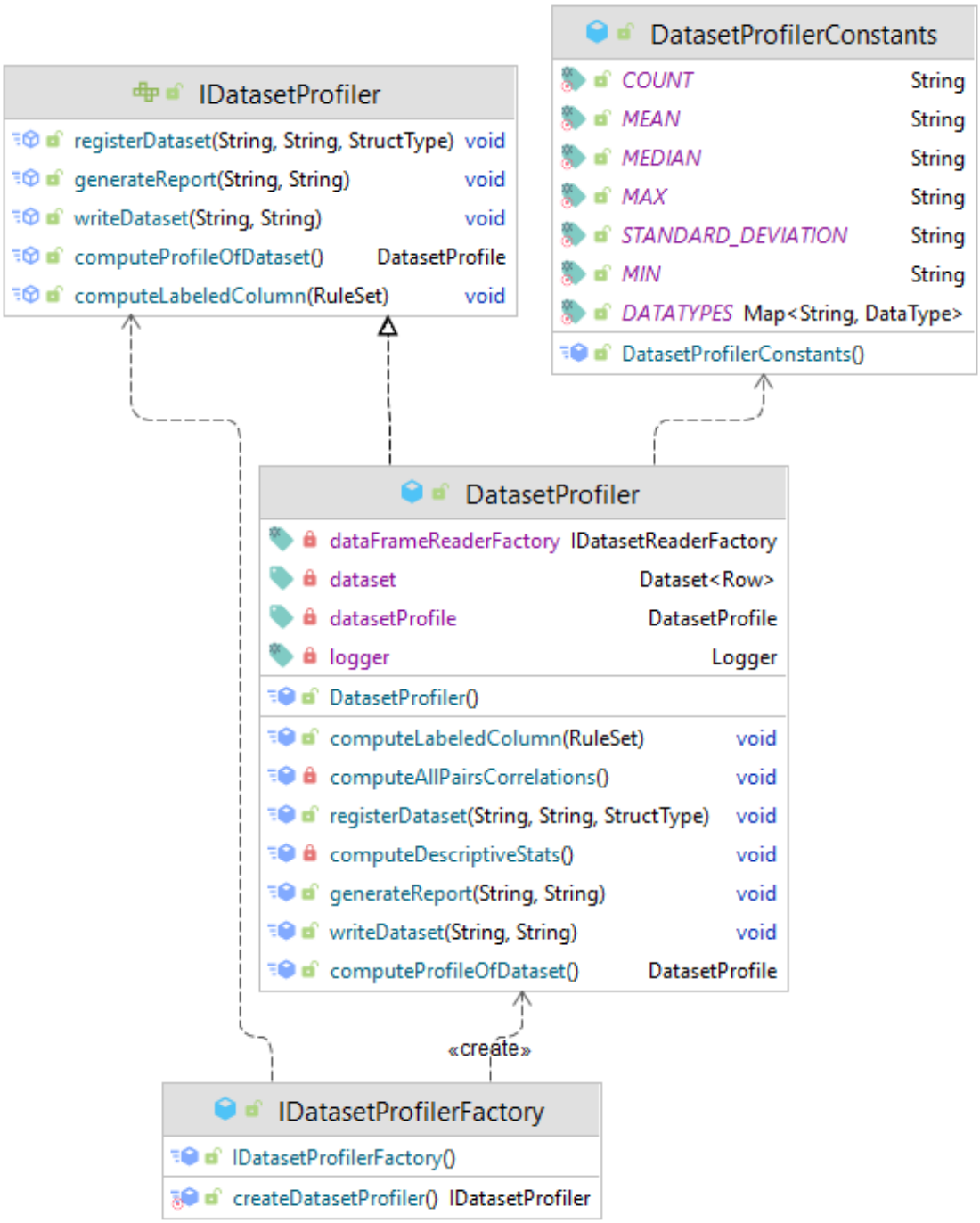
3.2.2 Πακέτο correlations



Εικόνα 8. Διάγραμμα UML του πακέτου correlations

Το συγκεκριμένο πακέτο περιέχει τις κλάσεις οι οποίες είναι υπεύθυνες για τον υπολογισμό των συσχετίσεων μεταξύ όλων των κολόνων χρησιμοποιώντας την μέθοδο Pearson που αναλύθηκε στο Κεφάλαιο 2. Υπάρχει ένα κεντρικό interface με όνομα ICorrelationsCalculator με μία μέθοδο που θα υλοποιεί κάθε παραγόμενη κλάση. Με τον τρόπο αυτό η λειτουργικότητα αυτή είναι επεκτάσιμη με άλλες μεθόδους και αλγόριθμους, καθώς με την υλοποίηση του interface και την χρήση της κλάσης ICorrelationsCalculatorFactory ο κώδικας θα χρειαστεί ελάχιστες αλλαγές για να δουλέψει με την νέα λειτουργία. Στην τρέχουσα υλοποίηση ο αλγόριθμος για τον υπολογισμό των συσχετίσεων βρίσκεται στην κλάση PearsonCorrelationsCalculator και αντικείμενα της κλάσης δημιουργούνται μέσω του ICorrelationsCalculatorFactory χρησιμοποιώντας μία παράμετρο που βρίσκεται στην κλάση CorrelationsSystemConstants που υποδεικνύει το αντικείμενο της κλάσης που θα δημιουργηθεί.

3.2.3 Πακέτο engine

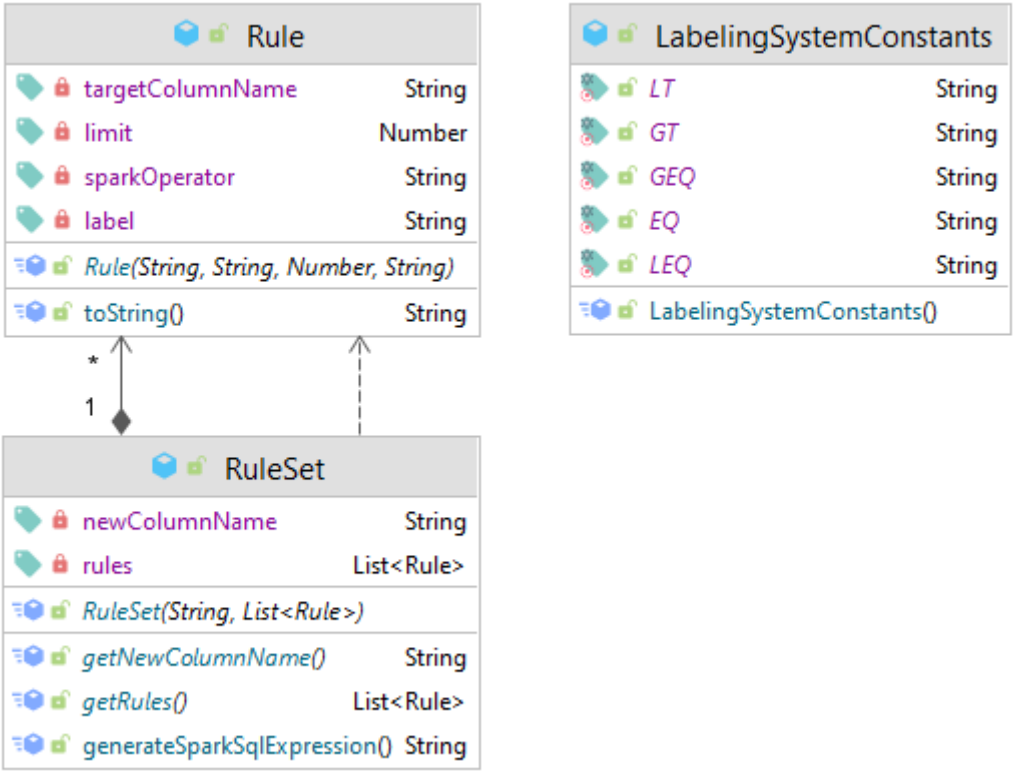


Εικόνα 9. Διάγραμμα UML του πακέτου engine

Το συγκεκριμένο πακέτο περιέχει τις κλάσεις υπεύθυνες για την βασική λειτουργεία του εργαλείου. Συγκεκριμένα υπάρχει ένα κεντρικό interface (IDatasetProfiler) το οποίο ορίζει τις λειτουργίες του API όπως η εγγραφή ενός συνόλου δεδομένων στο σύστημα (registerDataset), η δημιουργία νέων labeled κολόνων από ήδη υπάρχουσες (computeLabeledColumn), ο υπολογισμός του προφίλ του συνόλου δεδομένων μετά την

επιτυχή εγγραφή του στο σύστημα (computeProfileOfDataset), η παραγωγή και εξαγωγή της αναφοράς (generateReport) και η εγγραφή του συνόλου δεδομένων στον δίσκο σε περίπτωση που ο αναλυτής έχει την ανάγκη να αποθηκεύσει το σύνολο αυτό στο δίσκο μετά από την επεξεργασία (π.χ. Εισαγωγή νέων κολόνων). Και στο πακέτο αυτό ακολουθείται η ίδια τακτική, με μία κλάση (DatasetProfiler) η οποία υλοποιεί το κεντρικό interface IDatasetProfiler και υλοποιεί αυτές τις βασικές λειτουργίες που αναφέρθηκαν παραπάνω. Την δημιουργία των αντικειμένων και εδώ αναλαμβάνει ένα factory με όνομα IDatasetFactory. Η κλάση DatasetProfilerConstants καλύπτει την ανάγκη χρήσης σταθερών στις κλάσεις του πακέτου.

3.2.4 Πακέτο labeling

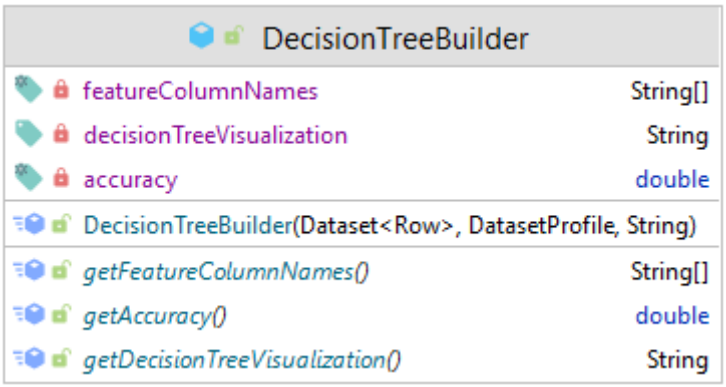


Εικόνα 10. Διάγραμμα UML του πακέτου labeling

Το συγκεκριμένο πακέτο περιέχει τις κλάσεις υπεύθυνες για την οργάνωση του συνόλου κανόνων για την δημιουργία labeled κολόνων. Πιο συγκεκριμένα η κλάση Rule είναι υπεύθυνη για την δημιουργία ενός και μόνο κανόνα που αποτελείται από το όριο (limit) του κανόνα, τον τελεστή (sparkOperator π.χ. >, <, <=, =<, =) και την υπάρχουσα κολόνα που το σύστημα θα εφαρμόσει τον κανόνα (targetColumnName).

Η κλάση RuleSet είναι υπεύθυνη για την οργάνωση του συνόλου των κανόνων που θα ισχύουν για την νέα κολώνα και κρατάει το όνομα της νέα κολόνας αυτής και μια λίστα από αντικείμενα τύπου Rule. Η κλάση Ruleset, είναι υπεύθυνη για την δημιουργία της τελικής έκφρασης η οποία θα εκτελεστεί από το Spark και θα παραχθεί η κολόνα. Η μέθοδος generateSparkSqlExpression κατασκευάζει την έκφραση σύμφωνα με το συντακτικό του Spark SQL και ουσιαστικά καλεί την μέθοδο toString κάθε αντικειμένου Rule η οποία παράγει τμήμα της τελικής έκφρασης. Και σε αυτό το πακέτο υπάρχει η κλάση LabelingSystemConstants με τις σταθερές που χρησιμοποιούνται στο πακέτο αυτό, που στη συγκεκριμένη περίπτωση είναι οι τελεστές ισότητας SQL του Spark (π.χ. >, <, <=, =<, =).

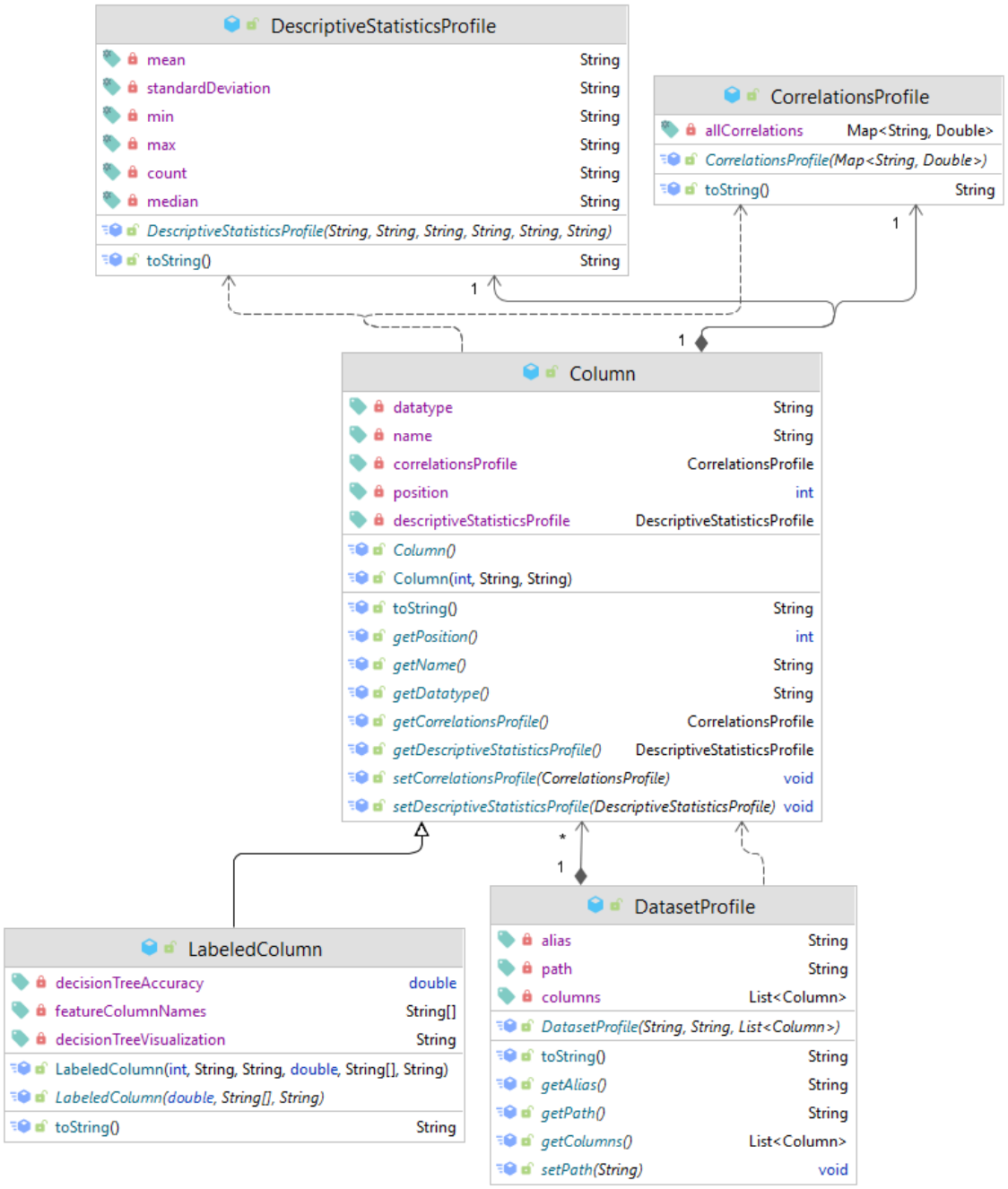
3.2.5 Πακέτο ml



Εικόνα 11. Διάγραμμα UML του πακέτου ml

Το συγκεκριμένο πακέτο περιέχει μόνο μία κλάση υπεύθυνη για την δημιουργία του δέντρου απόφασης μίας labeled κολόνας. Η κλάση αυτή χρησιμοποιεί το πακέτο MLLib του Spark και παράγει μια απλή αναπαράσταση του δέντρου σε If-Else μορφή καθώς και μια μετρική για το πόσο ακριβείς είναι οι προβλέψεις για το δέντρο αυτό.

3.2.6 Πακέτο model

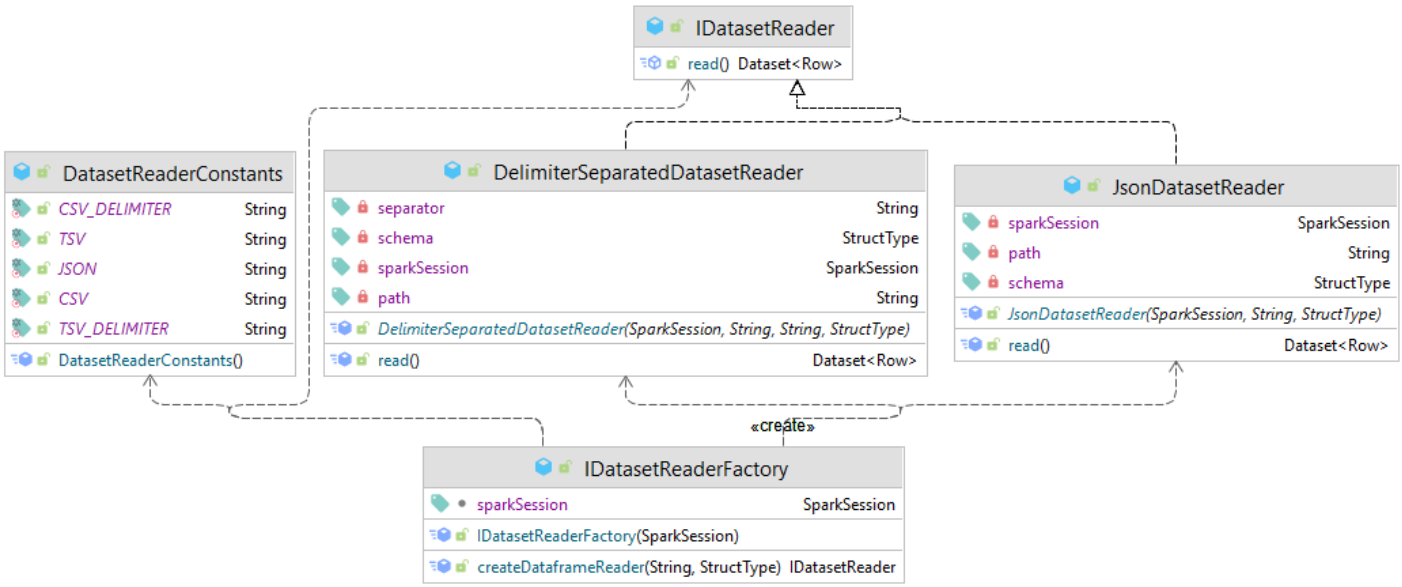


Εικόνα 12. Διάγραμμα UML του πακέτου model

Το πακέτο αυτό περιέχει όλες τις κλάσεις οι οποίες είναι απαραίτητες για την αποθήκευση των δεδομένων που παράγει το εργαλείο. Πιο συγκεκριμένα οι βασικές κλάσεις από τις οποίες αποτελείται το πακέτο είναι οι κλάσεις DatasetProfile, Column, LabeledColumn, DescriptiveStatisticsProfile και CorrelationsProfile. Αυτές οι κλάσεις λοιπόν είναι οι Domain κλάσεις του εργαλείου και το εργαλείο τις χρησιμοποιεί για να

οργανώσει τα παραγόμενα αποτελέσματα έτσι ώστε να τα χρησιμοποιήσει και αργότερα για την εξαγωγή της αναφοράς ευρημάτων.

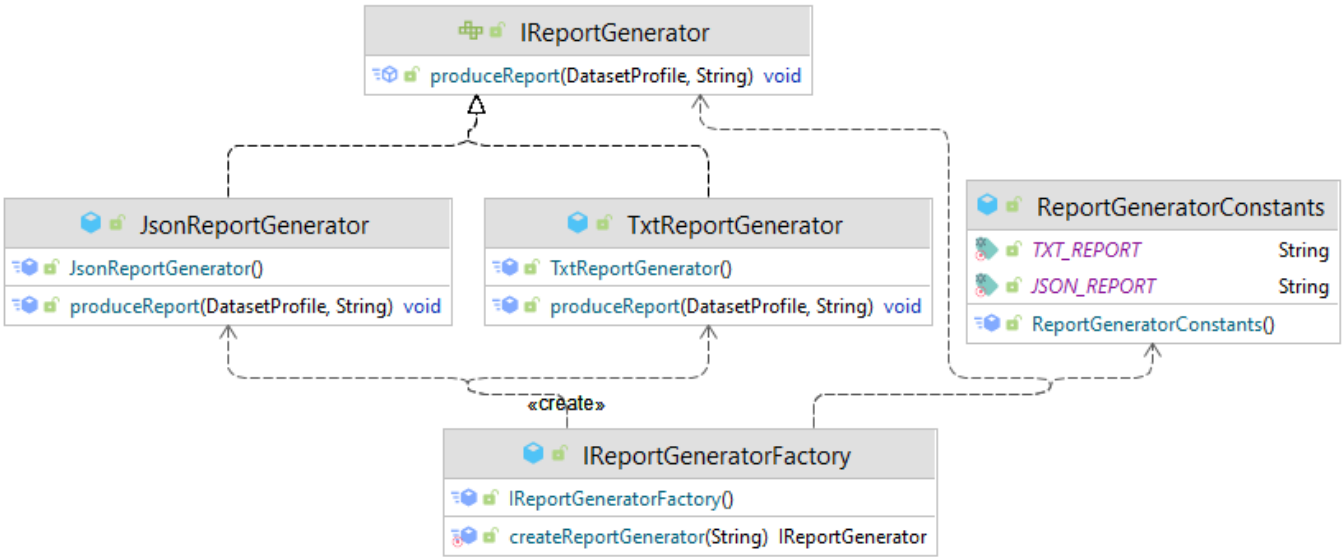
3.2.7 Πακέτο reader



Εικόνα 12. Διάγραμμα UML του πακέτου reader

Το συγκεκριμένο πακέτο περιέχει τις κλάσεις οι οποίες χρησιμοποιούνται για την ανάγνωση των αρχείων εισόδου. Συγκεκριμένα, το εργαλείο υποστηρίζει την είσοδο αρχείων δεδομένων τύπου CSC, TSV και JSON. Υπάρχει ένα κεντρικό interface με όνομα IDatasetReader το οποίο υλοποιούν οι αντίστοιχες κλάσεις για την υποστήριξη της ανάγνωσης των διαφόρων τύπων αρχείων που αναφέρθηκαν παραπάνω. Στην τρέχουσα υλοποίηση υπάρχουν δύο κλάσεις που υλοποιούν το κεντρικό interface, η DelimiterSeparatedDatasetReader και η JsonDatasetReader. Την δημιουργία των αντικειμένων αυτών αναλαμβάνει το factory με όνομα IDatasetReaderFactory λαμβάνοντας ως παράμετρο τον τύπο του Reader που θα χρειαστεί ανάλογα με τον τύπο του αρχείου εισόδου. Έτσι όλη η λειτουργικότητα είναι εύκολα επεκτάσιμη με πιθανή υποστήριξη και άλλων τύπων αρχείων εισόδου στο μέλλον.

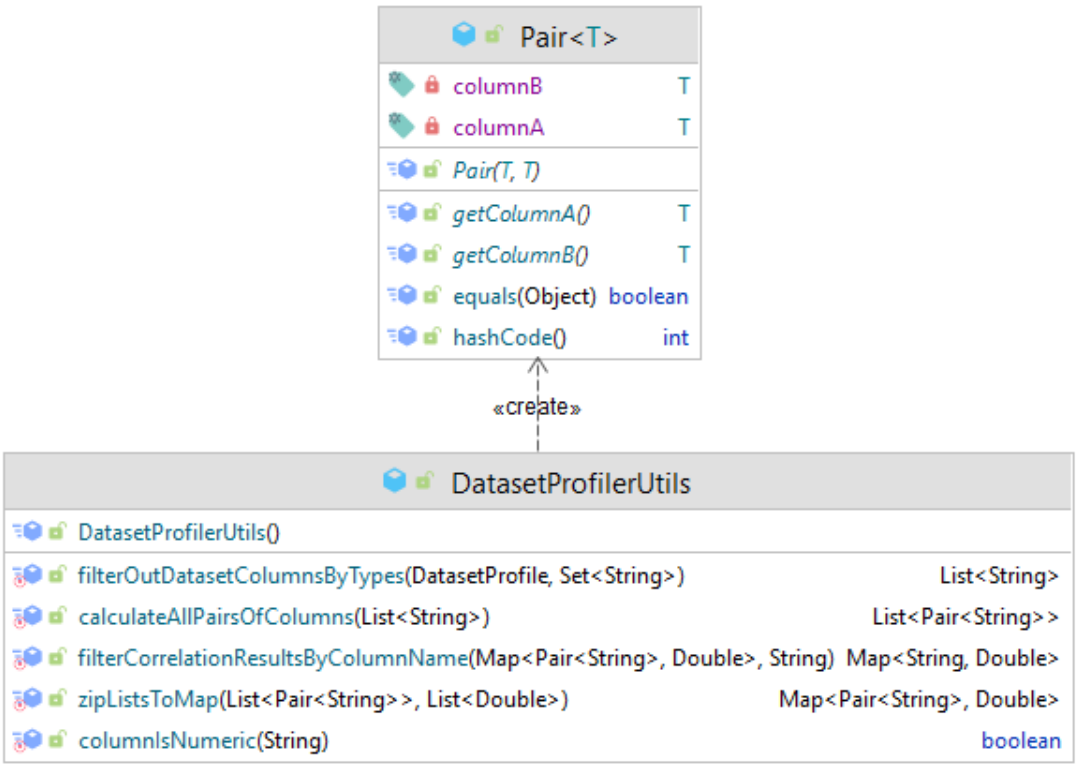
3.2.8 Πακέτο report



Εικόνα 13. Διάγραμμα UML του πακέτου report

Το συγκεκριμένο πακέτο περιέχει τις κλάσεις οι οποίες χρησιμοποιούνται για την δημιουργία της αναφοράς των ευρημάτων. Το σύστημα υποστηρίζει την εξαγωγή αναφοράς σε μορφή απλού αρχείου κειμένου και σε μορφή αρχείου JSON (ευκολότερο στην διαχείρηση για περαιτέρω επεξεργασία με εργαλεία όπως Python, Jupyter Notebook κ.λ.π.). Υπάρχει ένα κεντρικό interface με όνομα IReportGenerator το οποίο υλοποιούν οι κλάσεις TxtReportGenerator και JsonReportGenerator. Την δημιουργία των αντικειμένων αυτών αναλαμβάνει το factory με όνομα IReportGeneratorFactory λαμβάνοντας ως παράμετρο τον τύπο του αρχείου που θα παραχθεί ανάλογα με τον τύπο της αναφοράς που θα επιλέγει ο αναλυτής. Έτσι όλη η λειτουργικότητα είναι εύκολα επεκτάσιμη με πιθανή υποστήριξη και άλλων τύπων αρχείων εξόδου στο μέλλον. (π.χ. PDF)

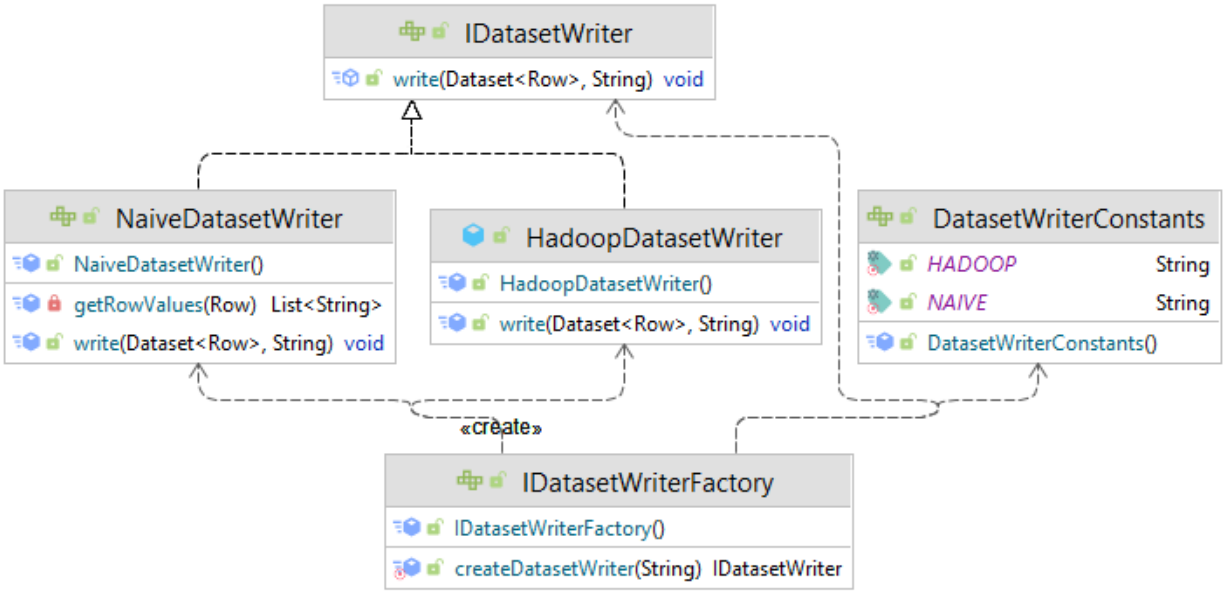
3.2.9 Πακέτο util



Εικόνα 14. Διάγραμμα UML του πακέτου util

Το πακέτο αυτό περιέχει κάποια βασικά εργαλεία-κλάσεις που χρησιμοποιούνται σε διάφορα σημεία του εργαλείου. Συγκεκριμένα, η κλάση DatasetProfileUtils περιέχει μεθόδους για την διαχείριση των δεδομένων που χρησιμοποιούνται από την κλάση DatasetProfiler του πακέτου engine. Η κλάση Pair είναι μια γενική κλάση η οποία χρησιμοποιείται για τα ζευγάρια συσχέτισης (ζευγάρια κολονών) του συστήματος συσχετίσεων (CorrelationsCalculator πακέτο correlations).

3.2.10 Πακέτο writer



Εικόνα 15. Διάγραμμα UML του πακέτου writer

Το συγκεκριμένο πακέτο περιέχει τις κλάσεις οι οποίες χρησιμοποιούνται για την εγγραφή του συνόλου δεδομένων εκ νέου στο δίσκο σε περίπτωση που ο αναλυτής επιθυμεί να κρατήσει το σύνολο αυτό έπειτα από την επεξεργασία μέσω του εργαλείου. Συγκεκριμένα, το εργαλείο υποστηρίζει την εγγραφή του συνόλου με δύο τρόπους. Ο πρώτος τρόπος είναι μια απλοϊκή προσέγγιση διαβάζοντας γραμμή-γραμμή το σύνολο δεδομένων και γράφοντας γραμμή-γραμμή στο δίσκο. Αυτή η προσέγγιση έχει αρνητικά καθώς σε μεγάλα σύνολα δεδομένων η ανάγνωση και εγγραφή ενός συνόλου δεδομένων εκ νέου στο δίσκο, που πιθανώς είναι διαμοιρασμένο σε πολλούς Spark κόμβους δεν είναι αποδοτικό. Για το λόγο αυτό δημιουργήθηκε ο δεύτερος τρόπος ο οποίος χρησιμοποιεί το Hadoop HDFS σύστημα, το οποίο είναι πολύ γρήγορο στην εξαγωγή τεράστιων αρχείων δεδομένων από την μνήμη του Spark στο δίσκο. Φυσικά η δεύτερη αυτή λύση προϋποθέτει την ύπαρξη των εκτελέσιμων του Hadoop στο σύστημα του αναλυτή. Υπάρχει ένα κεντρικό interface με όνομα IDatasetWriter το οποίο υλοποιούν οι αντίστοιχες κλάσεις για την υποστήριξη της εγγραφής με χρήση των δύο μεθόδων που αναφέρθηκαν παραπάνω. Στην τρέχουσα υλοποίηση υπάρχουν δύο κλάσεις που υλοποιούν το κεντρικό interface, η NaiveDatasetWriter που είναι η υλοποίηση της πρώτης μεθόδου και η HadoopDatasetWriter που είναι η υλοποίηση της δεύτερης μεθόδου. Την δημιουργία των αντικειμένων αυτών αναλαμβάνει το factory με όνομα IDatasetWriterFactory λαμβάνοντας ως παράμετρο τον τύπο του writer που θα δημιουργηθεί ανάλογα με την μέθοδο που επιθυμεί να χρησιμοποιήσει ο αναλυτής. Έτσι

όλη η λειτουργικότητα είναι εύκολα επεκτάσιμη με πιθανή υποστήριξη και άλλων τύπων γραφέων στο μέλλον.

3.3 Σχεδίαση και αποτελέσματα ελέγχου του λογισμικού

Για τον έλεγχο του εργαλείου χρησιμοποιήθηκε η μέθοδος του μαύρου κουτιού στην οποία κατασκευάζονται τα δεδομένα εισόδου τα οποία δίνονται σε ένα μαύρο κουτί (τα επιμέρους υποσυστήματα του εργαλείου της διπλωματικής στη συγκεκριμένη περίπτωση) και παράγεται μία έξοδος. Αν η έξοδος είναι ίδια με την αναμενόμενη έξοδο τότε η λειτουργία του μαύρου κουτιού είναι σωστή, ενώ σε αντίθετη περίπτωση είναι λάθος. Αναλυτικότερα, δημιουργήθηκαν έλεγχοι για το σύστημα δημιουργίας labeled στήλης, για την δημιουργία της τελικής αναφοράς, για το σύστημα εγγραφής του συνόλου στο δίσκο και τέλος, δημιουργήθηκαν κάποιοι απλοί έλεγχοι για τα κοινά εργαλεία του πακέτου utils. Φυσικά, δεν έγιναν έλεγχοι για τις λειτουργίες εξωτερικών βιβλιοθηκών (π.χ. Apache Spark API) καθώς οι βιβλιοθήκες αυτές εγγυώνται την σωστή λειτουργία τους με δικούς τους ελέγχους.

3.4 Λεπτομέρειες εγκατάστασης και υλοποίησης

Στην ενότητα αυτή περιγράφονται τα χαρακτηριστικά της συγκεκριμένης υλοποίησης όπως η πλατφόρμα υλοποίησης, το περιβάλλον ανάπτυξης με τα προγραμματιστικά εργαλεία καθώς και οι απαιτήσεις της εφαρμογής από το υλικό.

Το εργαλείο βρίσκεται σε αποθετήριο στο GitHub (https://github.com/DAINTINESS-Group/Pythia) και αναπτύχθηκε στην γλώσσα προγραμματισμού Java (https://www.oracle.com/java) η οποία είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού με πολλά πλεονεκτήματα και ευελιξία. Με την συγκεκριμένη γλώσσα προγραμματισμού είναι εύκολο να δημιουργηθεί επαναχρησιμοποιήσιμος κώδικας ο οποίος είναι εύκολο να εκτελεστεί από ένα υπολογιστικό σύστημα σε κάποιο άλλο, το οποίο έχει την Java εγκατεστημένη. Η ανάπτυξη έγινε σε περιβάλλον Windows 10 με την χρήση του Intellij IDEA για την συγγραφή του κώδικα αλλά υποστηρίζεται και η χρήση του Eclipse. Επίσης χρησιμοποιήθηκε το εργαλείο Maven για την διαχείριση των εξωτερικών πακέτων (dependencies) που χρησιμοποιήθηκαν κατά την ανάπτυξη, όπως και για την διαδικασία του build και του ελέγχου. Αναλυτικότερα, το Maven δίνει την δυνατότητα να αυτοματοποιηθεί η διαδικασία της εκτέλεσης διαφόρων λειτουργιών, όπως του build ώστε να παραχθούν τα απαραίτητα jar αρχεία και test για την εκτέλεση

των ελέγχων αυτόματα κάθε φορά που δίνεται η εντολή για νέο build. Επίσης δίνεται η δυνατότητα εκτέλεσης των ελέγχων χωρίς την εκτέλεση της εντολής του build. Τέλος, η διαχείριση των εξωτερικών πακέτων γίνεται πολύ εύκολη με το Maven καθώς υπάρχει το Maven Repository το οποίο είναι ένα online αποθετήριο για πακέτα Java που μπορεί κανείς να ψάξει εκατομμύρια πακέτα, να επιλέξει το πακέτο που επιθυμεί και μέσα σε πολύ λίγο χρόνο να έχει διαθέσιμο το πακέτο προς χρήση στο περιβάλλον ανάπτυξης. Το Pythia έχει αρκετές εξωτερικές βιβλιοθήκες, και το εργαλείο Maven βοηθάει στην συντήρηση των εκδόσεων των βιβλιοθηκών αυτών καθώς υπάρχουν συγκεντρωτικά σε ένα αρχείο όλες οι δηλώσεις των πακέτων αυτών.

Όσον αφορά τις απαιτήσεις υλικού, το τελικό εργαλείο χρησιμοποιεί το Apache Spark για τους μαθηματικούς υπολογισμούς και για το λόγο αυτό για μεγάλα αρχεία εισόδου με πολλές αριθμητικές κολόνες που κατά συνέπεια αυτό σημαίνει ότι θα εκτελεστούν πολλοί υπολογισμοί ενώ το σύστημα απαιτεί όλους τους πόρους του επεξεργαστή του συστήματος. Βέβαια, η συγκεκριμένη υλοποίηση είναι ρυθμισμένη να χρησιμοποιεί όλους τους πόρους του συστήματος αλλά φυσικά αυτό μπορεί να αλλάξει εύκολα μέσω του αρχείου ρυθμίσεων (spark.properties).

Εγκατάσταση

Στην ενότητα αυτή παρατίθενται τα βήματα για την εγκατάσταση του απαραίτητου λογισμικού για την ανάπτυξη του Pythia.

Βήματα:

1. Εγκατάσταση της Java 8 στο σύστημα ανάπτυξης του εργαλείου. Μετά την εκτέλεση του προγράμματος εγκατάστασης θα πρέπει να ενημερωθεί και το PATH του συστήματος με την νέα μεταβλητή JAVA_HOME η οποία θα δείχνει στον φάκελο με τα εκτελέσιμα αρχεία της Java.
2. Για το Maven δεν υπάρχει διαδικασία εγκατάστασης καθώς έχει συμπεριληφθεί ένα Maven Wrapper το οποίο είναι μια ενσωματωμένη εγκατάσταση του Maven στο project. Να σημειωθεί ότι για να λειτουργήσει αυτή η εγκατάσταση θα πρέπει να υπάρχει η μεταβλητή JAVA_HOME στο μονοπάτι του συστήματος όπως αναφέρθηκε στο πρώτο βήμα.
3. Εγκατάσταση του Eclipse ή του Intellij IDEA (Η Community Edition είναι δωρεάν έκδοση)
4. Σε περίπτωση που έγινε εγκατάσταση του Eclipse θα πρέπει να εγκατασταθεί και η υποστήριξη του Lombok που αναφέρθηκε παραπάνω καθώς δεν υποστηρίζεται αυτόματα από το Eclipse.

5. Για να χρησιμοποιηθεί το HadoopDatasetWriter που αναλύθηκε παραπάνω θα πρέπει να γίνει η εγκατάσταση του Hadoop έκδοση 3.2.2. Αυτό σημαίνει ότι θα πρέπει να γίνει εξαγωγή του tar.gz αρχείου που κατέβηκε από την επίσημη σελίδα του Hadoop και ανανέωση του PATH με την HADOOP_HOME μεταβλητή η οποία θα δείχνει στον φάκελο με τα εκτελέσιμα αρχεία του Hadoop. Στο σύστημα που αναπτύχθηκε το εργαλείο τα εκτελέσιμα αρχεία είναι στον φάκελο C:\Hadoop\hadoop-3.2.2. Σε περίπτωση που η εγκατάσταση του Hadoop γίνει σε Windows απαιτείται και η εγκατάσταση του WinUtils έκδοση 3.2.2. Για την εγκατάσταση θα πρέπει να γίνει αντιγραφή των περιεχομένων του φακέλου hadoop-3.2.2/bin του αποθετηρίου στον φάκελο εγκατάστασης του Hadoop μέσα στον φάκελο bin. Στο σύστημα ανάπτυξης ο φάκελος αυτός βρίσκεται εδώ: C:\Hadoop\hadoop-3.2.2\bin.

Διαδικασία του Build

Μετά την διαδικασία της εγκατάστασης των απαραίτητων εργαλείων, η διαδικασία του build και ελέγχου του εργαλείου είναι απλή.

Windows

Σε συστήματα Windows η εκτέλεση της εντολής **mvnw.cmd clean install** θα ξεκινήσει η διαδικασία εγκατάστασης των πακέτων, η διαδικασία της μεταγλώττισης και του ελέγχου. Μόλις αυτή η διαδικασία ολοκληρωθεί επιτυχώς θα έχει δημιουργηθεί ένας φάκελος με όνομα **target** και εσωτερικά θα υπάρχουν δύο jar αρχεία.

Unix συστήματα

Σε συστήματα τύπου Unix η εντολή είναι ελαφρώς διαφορετική. Με εκτέλεση της εντολής **./mvnw clean install** θα αρχίσει η ίδια διαδικασία όπως περιγράφθηκε παραπάνω.

Παράγωγα jar αρχεία

Όσον αφορά τα 2 jar αρχεία που δημιουργούνται έπειτα της διαδικασίας του build, αυτά έχουν ονόματα **Pythia-x.y.z-all-deps.jar** and **Pythia-x.y.z.jar.** Το πακέτο **all-deps** συμπεριλαμβάνει και τον μεταγλωτισμένο κώδικα των βιβλιοθηκών που χρειάζεται το εργαλείο για να δουλέψει χωρίς να απαιτείται η μέριμνα της εγκατάστασης των βιβλιοθηκών αυτών σε κάποιο άλλο project που θα χρησιμοποιηθεί το εργαλείο. Το άλλο jar αρχείο (**Pythia-x.y.z.jar**) δεν περιέχει τον εκτελέσιμο κώδικα των βιβλιοθηκών πάρα μόνο τον εκτελέσιμο κώδικα του εργαλείου και απαιτείται η μέριμνα της εγκατάστασης

των πακέτων αυτών σε άλλο project στην περίπτωση που θα χρησιμοποιηθεί αυτό το jar αρχείο.

Διαδικασία του Ελέγχου

Windows

Σε συστήματα Windows η εκτέλεση της εντολής **mvnw.cmd test** θα ξεκινήσει η διαδικασία του ελέγχου. Μόλις αυτή η διαδικασία ολοκληρωθεί το Maven θα δώσει τα αποτελέσματα συνοπτικά και το χρόνο που διήρκησε η διαδικασία

Unix συστήματα

Σε συστήματα τύπου Unix η εντολή είναι ελαφρώς διαφορετική. Με εκτέλεση της εντολής ./**mvnw test** θα αρχίσει η ίδια διαδικασία όπως περιγράφθηκε παραπάνω.

3.5 Επεκτασιμότητα του λογισμικού

Όπως αναπτύχθηκε παραπάνω το Pythia έχει δημιουργηθεί διαχωρίζοντας τα αρχεία κώδικα για τις διάφορες λειτουργίες που παρέχει σε Java πακέτα λογισμικού. Κάθε πακέτο από αυτά περιέχει τα αρχεία κώδικα υπεύθυνα για ακριβώς μία από τις λειτουργίες του εργαλείου και τα αρχεία αυτά ακολουθούν αρχιτεκτονική λογισμικού ανεκτική σε επεκτάσεις και τροποποιήσεις καθώς τα περισσότερα πακέτα ακολουθούν ένα Java Interface δηλώνοντας αφηρημένα την λειτουργικότητα του καθενός, το οποίο υλοποιείται αναλόγως για την παροχή των λειτουργιών που αναλαμβάνει και αντικείμενα δημιουργούνται μέσω Factory κλάσεων πετυχαίνοντας έτσι την εύκολη συντήρηση και επέκταση του εργαλείου. Τα πακέτα που δεν λειτουργούν ακολουθώντας κάποιο Interface είναι τα ml, labeling και config. Όπως αναφέρθηκε κατά την ανάλυση των πακέτων, το πακέτο ml περιέχει μία και μοναδική κλάση υπεύθυνη για την δημιουργία ενός δέντρου αποφάσεων, ακολουθώντας μια συγκεκριμένη ακολουθία εντολών χρησιμοποιώντας το API του Spark. Το πακέτο config περιέχει επίσης μια και μοναδική κλάση υπεύθυνη για την αναγνώριση των παραμέτρων λειτουργίας του Spark διαβάζοντας το αρχείο spark.properties από το δίσκο. Το πακέτο labeling περιέχει τις κλάσεις για την υποστήριξη της λειτουργίας εξαγωγής labeled κολόνας χρησιμοποιώντας την γλώσσα SQL μέσω του API του Spark. Συμπερασματικά, τα πακέτα αυτά υλοποιούν πολύ συγκεκριμένες λειτουργίες χρησιμοποιώντας το Spark και για το λόγο αυτό δεν ακολουθούν κάποιο Interface καθώς δεν υπάρχει ανάγκη για επεκτατικές υλοποιήσεις.

Κεφάλαιο 4. Πειραματική Αξιολόγηση

Στο κεφάλαιο αυτό περιγράφεται η πειραματική αξιολόγηση του συστήματος που υλοποιήθηκε. Αναλυτικότερα, έγιναν μετρήσεις για τα διάφορα υποσυστήματα του συστήματος χρησιμοποιώντας ένα σύνολο δεδομένων και μετρώντας τον χρόνο επεξεργασίας για διάφορες τιμές εγγραφών και κολόνων. Στο σύστημα το Spark έχει ρυθμιστεί να δουλεύει χρησιμοποιώντας όλους τους πόρους του συστήματος και οι μετρήσεις εκτελέστηκαν απομονώνοντας το Pythia από άλλες διεργασίες του λειτουργικού συστήματος για πιο ακριβή αποτελέσματα.

4.1 Μεθοδολογία πειραματισμού

Στην υποενότητα αυτή θα αναλυθούν οι διάφορες μετρήσεις που εκτελέστηκαν με σκοπό την πειραματική αξιολόγηση της αποδοτικότητας του συστήματος. Αναλυτικότερα, εκτελέστηκαν χρονικές μετρήσεις για το σύνολο της επεξεργασίας ενός συνόλου δεδομένων για διάφορο πλήθος εγγραφών και αριθμητικών κολόνων, καθώς και αναλυτικές μετρήσεις για τις επιμέρους επεξεργασίες που εκτελεί το σύστημα (εγγραφή του συνόλου στο σύστημα, περιγραφικά στατιστικά, συσχετίσεις κολόνων, εισαγωγή labeled κολόνας και δημιουργία Δέντρου αποφάσεων). Το σύνολο δεδομένων που χρησιμοποιήθηκε είναι ένα αρχείο από το σύνολο δεομένων Yelp (https://www.yelp.com/dataset) και συγκεκριμένα το αρχείο user.json το οποίο έχει περίπου 2.000.000 εγγραφές και 17 αριθμητικές κολόνες (Schema: https://www.yelp.com/dataset/documentation/main).

Μετρήθηκε η συμπεριφορά του συστήματος σε σταθερό πλήθος εγγραφών με διαφορετικό πλήθος αριθμητικών κολόνων όπως και σε διαφορετικό πλήθος εγγραφών και σταθερό πλήθος αριθμητικών κολόνων. Τέλος, δημιουργήθηκαν γραφικές παραστάσεις για την μελέτη των αποτελεσμάτων.

Οι μετρήσεις εκτελέστηκαν σε σύστημα λειτουργικού συστήματος Windows 10 έκδοσης 21H2 με τα ακόλουθα χαρακτηριστικά υλικού.

CPU	AMD Ryzen 7 2700 Eight-Core Processor 3.20 GHz
RAM	16.0 GB DDR4 3200MHz

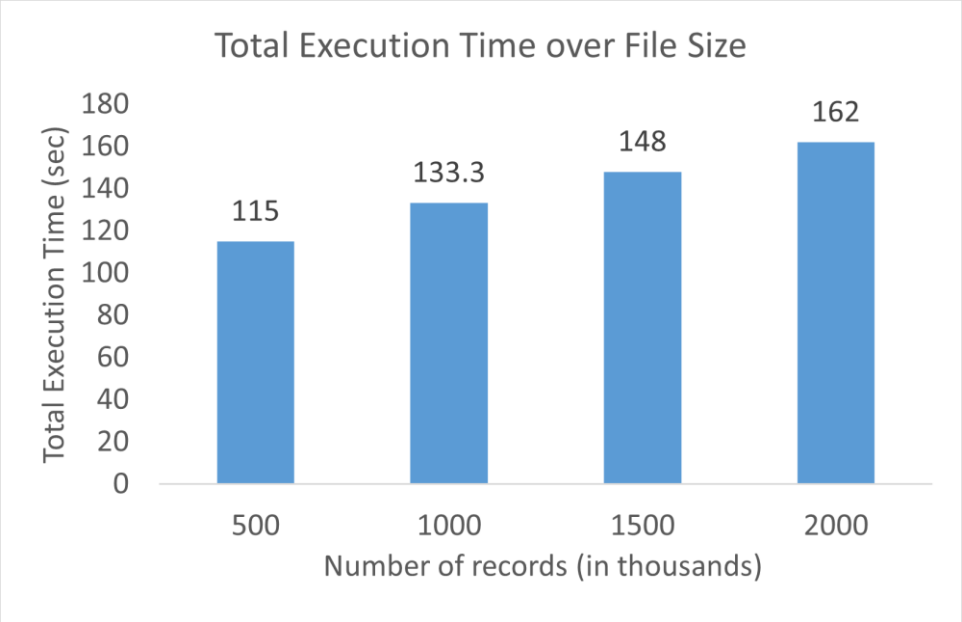
DISK	SSD SAMSUNG MZ-76E250B/EU 860 up to 550 MB/sec Sequential Read up to 520 MB/sec Sequential Write.

4.2 Αναλυτική παρουσίαση αποτελεσμάτων

Στην υποενότητα αυτή παρατίθενται τα αποτελέσματα των μετρήσεων που αναφέρθηκαν παραπάνω. Ο πίνακας 1 περιέχει τους συνολικούς χρόνους εκτέλεσης του συστήματος για την παραγωγή της τελικής αναφοράς αυξάνοντας το πλήθος των εγγραφών κατά 500.000 ξεκινώντας από 500.000 και κρατώντας το πλήθος των αριθμητικών κολόνων σταθερό σε 10 κολόνες.

Αριθμός αριθμητικών στηλών	**Αριθμός συνολικών εγγραφών**			
	500.000	1.000.000	1.500.000	2.000.000
10	115	133.3	148.0	162

Πίνακας 1. Συνολικός χρόνος εκτέλεσης για διαφορετικό πλήθος εγγραφών (σε δευτερόλεπτα)



Γραφική παράσταση 1. Ραβδόγραμμα συνολικού χρόνου εκτέλεσης για διαφορετικό πλήθος εγγραφών για 10 αριθμητικές κολόνες (σε δευτερόλεπτα)

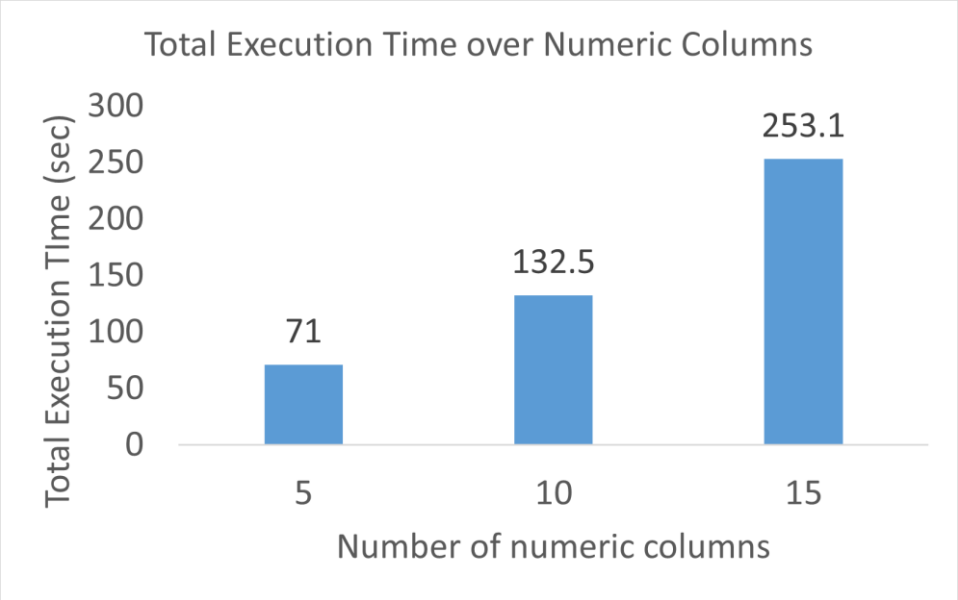
Γραφική παράσταση 2. Διάγραμμα γραμμής συνολικού χρόνου εκτέλεσης για διαφορετικό πλήθος εγγραφών για 10 αριθμητικές κολόνες (σε δευτερόλεπτα)

Στις γραφικές παραστάσεις 1 και 2 παρατηρούμε μια σταθερή αύξηση στον συνολικό χρόνο επεξεργασίας της τάξης των 15 δευτερολέπτων για σταθερή αύξηση των εγγραφών κατά 500.000.

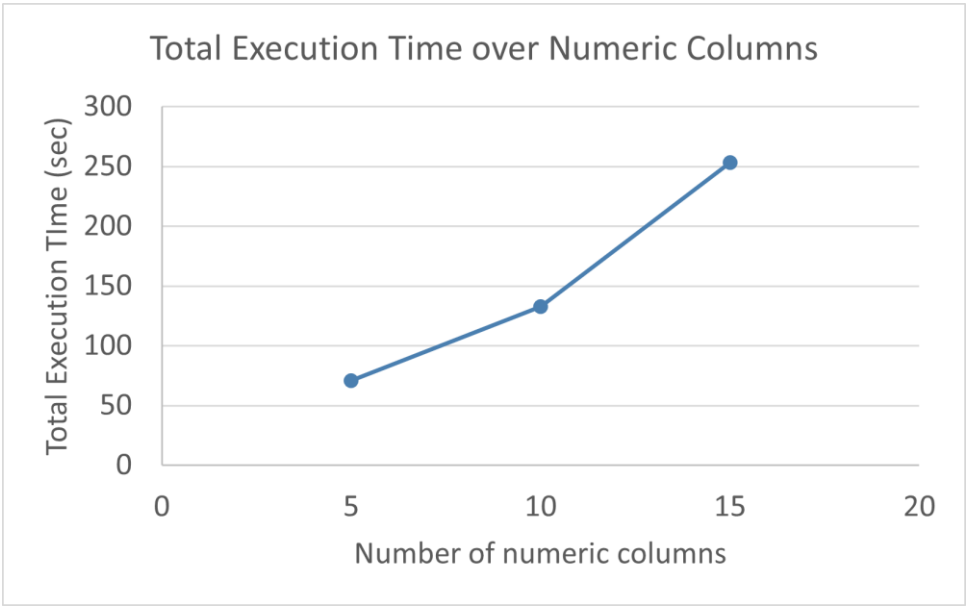
Ο πίνακας 2 περιέχει τους συνολικούς χρόνους εκτέλεσης του συστήματος για την παραγωγή της τελικής αναφοράς αυξάνοντας το πλήθος των αριθμητικών στηλών κατά 5 ξεκινώντας από 5 στήλες και κρατώντας το πλήθος εγγραφών σταθερό στις 1.000.000 εγγραφές.

Αριθμός συνολικών εγγραφών	**Αριθμός αριθμητικών στηλών**		
	5	10	15
1.000.000	71	132.5	253.1

Πίνακας 2. Συνολικός χρόνος εκτέλεσης για διαφορετικό πλήθος κολόνων (σε δευτερόλεπτα)



Γραφική παράσταση 3. Ραβδόγραμμα συνολικού χρόνου εκτέλεσης για διαφορετικό πλήθος αριθμητικών κολόνων για 1.000.000 εγγραφές (σε δευτερόλεπτα)



Γραφική παράσταση 4. Διάγραμμα γραμμής συνολικού χρόνου εκτέλεσης για διαφορετικό πλήθος αριθμητικών κολόνων για 1.000.000 εγγραφές (σε δευτερόλεπτα)

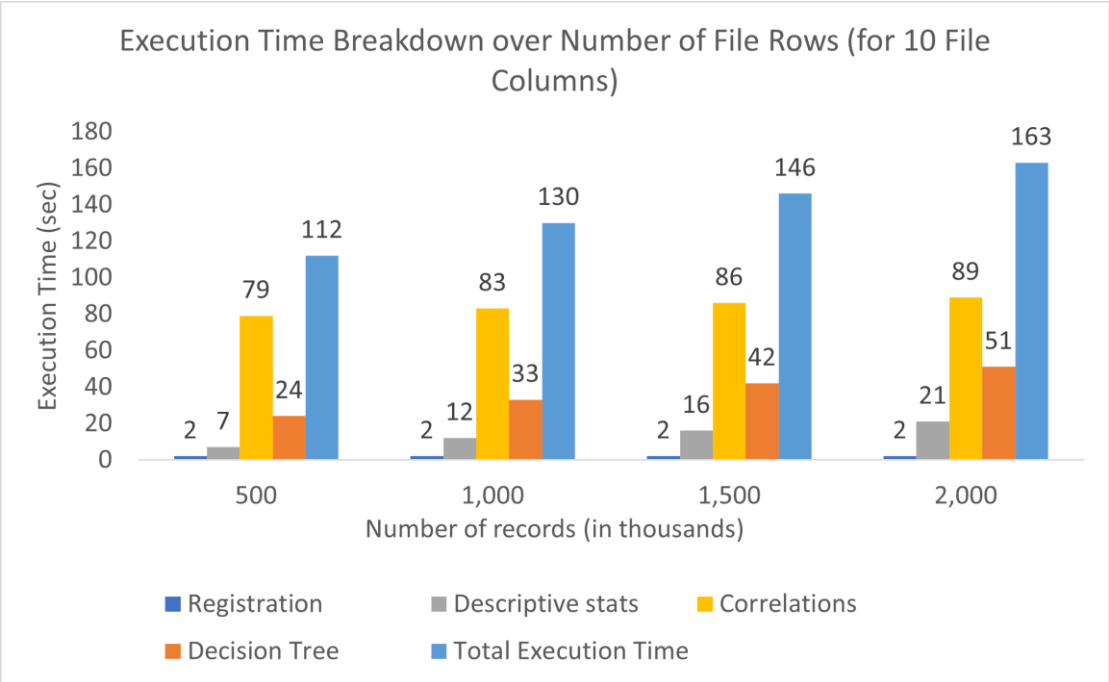
Στις γραφικές παραστάσεις 3 και 4 παρατηρούμε πιο απότομη αύξηση στον συνολικό χρόνο επεξεργασίας για σταθερή αύξηση των αριθμητικών κολόνων κατά 5. Όπως θα φανεί καθαρά στη γραφική παράσταση 6, ο βασικός υπαίτιος είναι οι υπολογισμοί των

συσχετίσεων των ζευγών, αφού όταν αυτά αυξάνονται είναι λογικό να αυξάνονται και οι υπολογισμοί που πρέπει να εκτελέσει το σύστημα.

Ο πίνακας 3 περιέχει τους επιμέρους χρόνους εκτέλεσης των υποσυστημάτων αυξάνοντας το πλήθος των εγγραφών κατά 500.000 ξεκινώντας από 500.000 και κρατώντας το πλήθος των αριθμητικών κολόνων σταθερό στις 10 κολόνες.

Αριθμός συνολικών εγγραφών	**500.000**	**1.000.000**	**1.500.000**	**2.000.000**
Εγγραφή συνόλου	2	2	2	2
Περιγραφικά στατιστικά	7	12	16	21
Συσχετίσεις	79	83	86	89
Δέντρο αποφάσεων	24	33	42	51
Σύνολο	112	130	146	163

Πίνακας 3. Χρόνοι εκτέλεσης των υποσυστημάτων για διαφορετικό αριθμό εγγραφών χρησιμοποιώντας 10 αριθμητικές στήλες (σε δευτερόλεπτα)



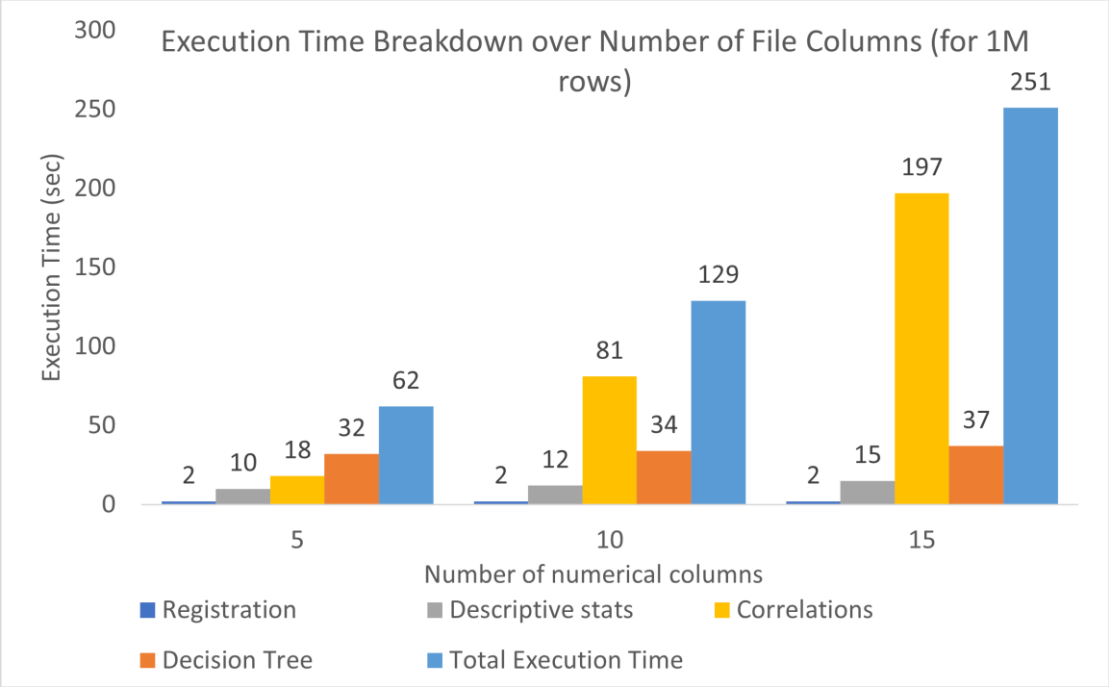
Γραφική παράσταση 5. Ραβδόγραμμα χρόνου εκτέλεσης επιμέρους υποσυστημάτων για διαφορετικό πλήθος εγγραφών για 10 αριθμητικές κολόνες (σε δευτερόλεπτα)

Στη γραφική παράσταση 5 παρατηρούμε ότι αυξάνοντας τις εγγραφές με σταθερό πλήθος αριθμητικών κολόνων αυξάνεται ο χρόνος γενικά σταθερά στην περίπτωση των συσχετίσεων και δεν παρατηρείται κάποια απότομη μεταβολή στον χρόνο. Στην περίπτωση του Δέντρου Απόφασης παρατηρούμε αρκετά σημαντική αύξηση περίπου στα 10 δευτερόλεπτα με αύξηση 500.000 εγγραφών. Όσον αφορά τα περιγραφικά στατιστικά υπάρχει αύξηση περίπου 5 δευτερολέπτων με αύξηση ανά 500.000 εγγραφές. Τέλος στην περίπτωση της εγγραφής του συνόλου στο σύστημα παρατηρούμε ότι αυτή είναι σταθερή στα 2 δευτερόλεπτα.

Ο πίνακας 4 περιέχει τους επιμέρους χρόνους εκτέλεσης των υποσυστημάτων αυξάνοντας το πλήθος των αριθμητικών στηλών κατά 5 ξεκινώντας από 5 στήλες και κρατώντας το πλήθος εγγραφών σταθερό στις 1.000.000 εγγραφές.

Αριθμός αριθμητικών στηλών	**5**	**10**	**15**
Εγγραφή συνόλου	2	2	2
Περιγραφικά στατιστικά	10	12	15
Συσχετίσεις	18	81	197
Δέντρο αποφάσεων	32	34	37
Σύνολο	62	129	251

Πίνακας 4. Χρόνοι εκτέλεσης των υποσυστημάτων για διαφορετικό πλήθος κολόνων χρησιμοποιώντας 1.000.000 εγγραφές (σε δευτερόλεπτα)



Γραφική παράσταση 6. Ραβδόγραμμα χρόνου εκτέλεσης επιμέρους υποσυστημάτων για διαφορετικό πλήθος αριθμητικών κολόνων για 1.000.000 εγγραφές (σε δευτερόλεπτα)

Στη γραφική παράσταση 6 παρατηρούμε ότι αυξάνοντας τις αριθμητικές κολόνες κατά 5 με σταθερό πλήθος εγγραφών στις 1.000.000 η σημαντική αύξηση στον χρόνο υπολογισμού βρίσκεται στις συσχετίσεις κάτι που αναφέρθηκε και παραπάνω καθώς αυξάνονται οι κολόνες και άρα αυξάνονται και τα ζεύγη συσχετίσεων, αυξάνεται και ο χρόνος υπολογισμού τους σημαντικά.

Κεφάλαιο 5. Επίλογος

5.1 Σύνοψη και συμπεράσματα

Τα σύνολα δεδομένων είναι ζωτικής σημασίας και χρησιμοποιούνται πολύ συχνά στο χώρο της Επιστήμης των Δεδομένων (Data Science) και της Ανάλυσης Δεδομένων (Data Analytics). Η επιλογή ενός συνόλου δεδομένων δεν είναι μια απλή διαδικασία καθώς ο αναλυτής θα πρέπει να γνωρίζει κάποια βασικά στατιστικά για αυτό, για να μπορέσει να εκτιμήσει την ποιότητα του.

Για να εκτιμηθεί η ποιότητα των δεδομένων υπάρχουν γνωστά διαδραστικά εργαλεία που αναφέρθηκαν νωρίτερα τα οποία αξιοποιούνται για το λόγο αυτό. Το εργαλείο που αναπτύχθηκε στην εν λόγω διπλωματική έρχεται να λύσει το πρόβλημα της έλλειψης της αυτόματης εξαγωγής στατιστικών για σύνολα δεδομένων ενδιαφέροντος ενός αναλυτή.

Αναλυτικότερα, το εργαλείο επιτρέπει στον αναλυτή να εγγράψει ένα data set και με τη βοήθεια του συστήματος να δηλώσει τα πεδία του, και τον τύπο τους (int, double, Datetime, Boolean, κλπ). Πέρα από τα απλά περιγραφικά σταστικά που υπολογίζει το σύστημα όπως μέση τιμή, διάμεσος κλπ. για κάθε labeled πεδίο δημιουργεί ένα decision tree και υπολογίσει τις συσχετίσεις όλων των ζευγών των αριθμητικών κολόνων του συνόλου. Μετά το τέλος της επεξεργασίας του συνόλου δεδομένων το εργαλείο παράγει μια αναφορά με τα ευρήματα σε απλή μορφή κειμένου ολοκληρώνοντας την αυτόματη εξαγωγή του προφίλ του.

Συμπερασματικά, το εργαλείο κατέστησε δυνατή την αυτόματη και γρήγορη εξαγωγή στατιστικών με διάφορες τεχνικές, δίνοντας στον αναλυτή ευελιξία ως προς την γρήγορη εκτίμηση του στατιστικού προφίλ ενός συνόλου δεδομένων καθώς δεν απαιτεί διαδραστικότητα όπως άλλα εργαλεία στον χώρο της επιστήμης των δεδομένων.

5.2 Μελλοντικές επεκτάσεις

Υπάρχει μία μεγάλη λίστα από πράγματα που έχει ενδιαφέρον να υλοποιηθούν στο μέλλον τα οποία περιγράφονται στις επόμενες παραγράφους και θα ήταν πολύ ενδιαφέρουσα η ενασχόληση με την υλοποίησή τους.

Εισαγωγή επιπλέον αυτόματων λειτουργιών. Αυτή τη στιγμή το Pythia υποστηρίζει αυτόματο υπολογισμό περιγραφικών στατιστικών, συσχετίσεων μεταξύ αριθμητικών κολόνων και αυτόματη δημιουργία δέντρου αποφάσεων για κάθε labeled κολόνα. Φυσικά, υπάρχουν αρκετοί μέθοδοι εξαγωγής συμπερασμάτων για δεδομένα οι οποίοι θα μπορούσαν να εισαχθούν στο Pythia και να βελτιώσουν το εργαλείο. Κάποιες από αυτές τις μεθόδους αναφέρθηκαν στο κεφάλαιο του υποβάθρου όπως η μέθοδος του Chi-squared ελέγχου υπόθεσης που επίσης υποστηρίζεται και από το Spark, και άλλες μέθοδοι υπολογισμού συσχετίσεων μεταξύ κολόνων όπως η μέθοδος Kendall-tau. Τέλος, το Spark υποστηρίζει πολλές μεθόδους μηχανικής μάθησης που θα μπορούσαν να εισαχθούν στο Pythia, όπως η μέθοδος του Clustering των εγγραφών του data set καθώς και η αποτίμηση της ποιότητας του.

Αυτόματη αναγνώριση labeled κολόνων. Όπως αναφέρθηκε παραπάνω το Pythia δίνει την δυνατότητα στον αναλυτή να δηλώσει κανόνες labeling και να δημιουργήσει μια νέα κολόνα με την εφαρμογή των κανόνων αυτών σε μία αριθμητική κολόνα και στην συνέχεια το Pythia θα δημιουργήσει ένα δέντρο αποφάσεων για τη νέα αυτή κολόνα. Μια πιθανή επέκταση θα ήταν η αυτόματη αναγνώριση κολόνων ως labeled υπολογίζοντας τις μοναδικές τιμές μιας μη αριθμητικής κολόνας και αν το πλήθος των τιμών αυτών πληροί ένα κατώφλι που δηλώνει ο αναλυτής τότε το εργαλείο θα τις αντιμετωπίζει ως labeled κολόνες και θα εξάγει και για αυτές ένα δέντρο αποφάσεων.

Υποστήριξη επιπλέον τύπων αρχείων εισόδου. Αυτή τη στιγμή το Pythia υποστηρίζει αρχεία εισόδου τύπου JSON, και αρχεία CSV και TSV. Μια πιθανή επέκταση λοιπόν θα ήταν η υποστήριξη και άλλων τύπων αρχείων όπως PSV, Parquet, EXCEL κ.α.

Δημιουργία αναφορών PDF. Μια άλλη σημαντική επέκταση θα ήταν και η υποστήριξη άλλων μορφών εξαγωγής τελικής αναφοράς μετά την επεξεργασία, όπως η δημιουργία PDF αναφοράς (π.χ. με το εργαλείο Apache PDFBox) ή και πιθανή εισαγωγή γραφικών παραστάσεων και άλλων οπτικοποιήσεων των αποτελεσμάτων στο PDF της αναφοράς.

Δυνατότητα ορισμού κατωφλίων σημαντικών αποτελεσμάτων από το αναλυτή. Η αναφορά που δημιουργείται από το Pythia περιέχει όλα τα αποτελέσματα που υπολόγισε είτε αυτά είναι σημαντικά είτε όχι. Επομένως μια πιθανή επέκταση θα ήταν η

υποστήριξη δήλωσης κατωφλίων από τον αναλυτή για την σημαντικότητα των αποτελεσμάτων και η εξαγωγή στην αναφορά μόνο αυτών που πληρούν τις τιμές κατωφλίων.

Εξαίρεση κολώνων. Μια άλλη επέκταση θα ήταν η δυνατότητα αγνόησης συγκεκριμένων κολόνων από το σύνολο δεδομένων εισόδου καθώς πιθανόν να μην υπάρχει κάποιο ενδιαφέρον από τον αναλυτή για όλες τις κολόνες, αποφεύγοντας έτσι αχρείαστους υπολογισμούς, παρέχοντας γρηγορότερα και πιο στοχευμένα αποτελέσματα

Δημιουργία διεπαφής χρήστη γραμμής εντολών. Όπως αναφέρθηκε, το Pythia είναι μια βιβλιοθήκη που παρέχει τις λειτουργίες που αναλύθηκαν εκτενώς παραπάνω. Αυτό σημαίνει ότι για να χρησιμοποιηθεί πρέπει να το εγκατασταθεί σε κάποιο project και στη συνέχεια να χρησιμοποιηθεί το API που παρέχει τις λειτουργίες αυτές. Επομένως, μια πιθανή επέκταση θα ήταν η δημιουργία μιας διεπαφής χρήστη γραμμής εντολών (command line interface) για την ευκολότερη και γρηγορότερη χρήση του εργαλείου από το τερματικό.

Δημιουργία Web εφαρμογής χρησιμοποιώντας το Pythia ως backend. Ίσως μια από τις σημαντικότερες επεκτάσεις θα ήταν η δημιουργία μιας Web εφαρμογής με χρήση του Spring Framework που θα χρησιμοποιεί το Pythia στο backend κάνοντας τους υπολογισμούς στην πλευρά διακομιστή προσφέροντας τις λειτουργίες αυτές μέσω του φυλλομετρητή. Αυτό συνεπάγεται την εύκολη χρήση Web frontend τεχνολογιών για οπτικοποιήσεις(Chart.js ή D3.js) και κατασκευή σύγχρονων διεπαφών χρήστη μέσω σύγχρονων εργαλείων όπως το React.js.

Βιβλιογραφία

[BOSL12]	Sarah Boslaugh. Statistics in a Nutshell, 2nd Edition Released November 2012, O'Reilly, ISBN: 9781449316822
[Apache Spark, n.d.]	Apache Spark - Unified Engine for large-scale data analytics Apache Software Foundation https://spark.apache.org/ Last visited: 22-03-10
[Apache Hadoop, n.d.]	The Apache™ Hadoop® project develops open-source software for reliable, scalable, distributed computing. https://hadoop.apache.org/ Last visited: 22-03-10
[DWTD20]	Jules S. Damji, Brooke Wenig, Tathagata Das, Denny Lee, Learning Spark - Lightning-Fast Data Analytics, 2nd Edition Released November 2020, O'Reilly, ISBN: 9781492049999