ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΜΗΧΑΝΙΚΩΝ Η/Υ & ΔΙΚΤΥΩΝ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΜΣ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΔΙΚΤΥΩΝ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΤΕΧΝΙΚΕΣ ΚΑΙ ΜΕΘΟΔΟΛΟΓΙΕΣ ΓΙΑ ΤΗΝ ΣΧΕΔΙΑΣΗ ΚΑΙ ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΓΙΑ ΜΕΓΑΛΑ ΔΕΔΟΜΕΝΑ

Γιαννάκης Ιωάννης

Επιβλέπων: Στύλιος Χρυσόστομος

Ιωάννινα 9/2/2021

TECHNIQUES AND METHODOLOGIES FOR DESIGNING AND DEVELOPING BIG DATA APPLICATIONS

Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα πτυχιακή εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Γιαννάκης Ιωάννης

Υπογραφή

Περίληψη

Η διπλωματική εργασία αφορά την μελέτη, ανάλυση και σύγκριση τεχνικών που αφορούν την επεξεργασία μεγάλου όγκου δεδομένων (Hadoop, Spark, Kafka, MapReduce) καθώς και τα εργαλεία που χρησιμοποιούνται (Python, Java, Scala) για την υλοποίηση εφαρμογών ανάλυσης και επεξεργασίας μεγάλων δεδομένων. Η πειραματική μελέτη αφορά την ανάλυση, επεξεργασία και κατηγοριοποίηση ενός χαρτοφυλακίου μετοχών, καθώς και τον σχεδιασμό μια διαδικασίας για την πρόβλεψη μελλοντικών τιμών της μετοχής μια συγκεκριμένης εταιρείας.

Abstract

The dissertation studies, analyses and compares existing methodologies and approaches on how to process Big Data (Hadoop, Spark, Kafka, MapReduce) as well as the available tools (Python, Java, Scala) that could be used for analyzing and processing big data. The experimental study concerns the analysis, processing and categorization of a portfolio of shares and it has designed, developed and tested an approach for predicting future share prices of a specific company.

Πίνακας Περιεχόμενων

Πίνακας συντομογραφιών

KDD= Knowledge Discovery from Databases

DIKW = Data, Information, Knowledge, Wisdom

HDFS = Hadoop Distributed File System

YARN = Yet Another Resource Negotiator

ETL = Extract ,Transform , Load

CEP=Complex Event Processing

IDE = Integrated Development Environment

Εισαγωγή

Οι καθημερινές ανάγκες για τη βελτίωση της ζωής του ανθρώπου και την ορθή λήψη αποφάσεων οδηγούν σε νέες μεθόδους έρευνας που αφορούν την επίτευξη καλύτερης ποιότητας ζωής.

Τα αντικείμενα της εξόρυξης και αποθήκευσης δεδομένων καθώς και της μηχανικής μάθησης έχουν συντελέσει στην αυξημένη τα τελευταία χρόνια εξέλιξη της τεχνολογίας και στην εξαγωγή χρήσιμων συμπερασμάτων σε διαφόρους κλάδους όπως η οικονομία, η υγεία, η ναυτιλία δημιουργώντας τεράστια κοινωνική και τεχνολογική πρόοδο.

Ο όγκος των δεδομένων που παράγονται, αποθηκεύονται και αναλύονται καθημερινά αυξάνεται με εξαιρετικά ταχείς ρυθμούς. Η επιστήμη των υπολογιστών έχει ως στόχο την ανάπτυξη και διευκόλυνση των παραπάνω αναγκών.

Τα εργαλεία που θα αναλύσουμε αργότερα και τα οποία συμβάλουν στην διαχείριση δεδομένων είναι το Hadoop, Spark, Kafka καθώς και οι γλώσσες προγραμματισμού όπως η Python, η Java και η Scala, η χρήση των οποίων συνέβαλε στην τεράστια ανάπτυξη του κλάδου του Data science.

Η παρούσα διπλωματική έχει ως σκοπό να αναλύσει σε βάθος τί είναι ο κλάδος των Big Dataπου ακούμε καθημερινά και να μας βοηθήσει να κατανοήσουμε πού χρησιμοποιούνται τα Μεγάλα Δεδομένα. Επίσης, στόχο της παρούσας εργασίας αποτελεί η ανάδειξη συγκεκριμένων εργαλείων τα οποία πρόκειται να συγκριθούν μεταξύ τους ώστε να αναδειχθούν εν τέλει τα όποια θετικά και αρνητικά στοιχεία τους. Στη συνέχεια, θα αναφερθούμε σε τεχνικές και μεθοδολογίες μέσα από μια εφαρμογή για καλύτερη κατανόηση η οποία θα γίνει με μία από τις τρείς γλώσσες προγραμματισμού και ένα από τα τρία προγράμματα συλλογής δεδομένων. Τέλος, θα παρουσιαστούν οι τεχνικές με τις οποίες μπορεί να διδαχθεί το αντικείμενο των Big Data.

1. Εισαγωγή

1.1. Από το χθες στο σήμερα

Από την αρχαιότητα ο άνθρωπος είχε την τάση να παρατηρεί και να αναλύει το καθετί που βλέπει γύρω του φτάνοντας έτσι σε σημαντικές εφευρέσεις-ανακαλύψεις που συνετέλεσαν στην διευκόλυνση της καθημερινής του ζωής. Από τον Βενιαμίν Φραγκλίνο ο οποίος ανακάλυψε τον ηλεκτρισμό μέσω της παρατήρησης του κεραυνού και τον Ισαάκ Νεύτωνα με τον γνωστό σε όλους «νόμο της αδράνειας», φτάσαμε να μιλάμε για την εφεύρεση τουInternet που ανήκει βέβαια στην σύγχρονη ιστορία. Το Internet δημιουργήθηκε κατά τη διάρκεια του Ψυχρού πολέμου το 1965 για να αποστέλλονται κωδικοποιημένα μηνύματα. Έτσι για πρώτη φορά είχαμε τον όρο της «αποστολής πακέτων μηνυμάτων».

Το πρώτο πράγμα που συμπεραίνει κανείς, μελετώντας αυτές τις εφευρέσεις, είναι πως προκειμένου να συμβούν, χρησιμοποιήθηκαν τα κατωτέρω στάδια:

1. Παρατήρηση
2. Καταγραφή της πληροφορίας
3. Ανάλυση της πληροφορίας
4. Υλοποίηση της πληροφορίας

Με τα παραπάνω στάδια καταλαβαίνουμε ότι καταγράφονται δεδομένα χρήσιμα προς την επίλυσή τους.

Φτάνοντας στο σήμερα, στην «εποχή της πληροφορίας» όπως ονομάζεται από πολλούς, καταλήγουμε πως λόγω του τεράστιου όγκου των πληροφοριών και της ανάγκης εξεύρεσης τρόπου διαχείρισης αυτών, τα δεδομένα είναι τόσο πολύτιμα όσο ο μαύρος χρυσός. Λόγω της αξίας που έχουν αποκτήσει τα ίδια, καθώς και η ανάγκη χρηστής διαχείρισής τους, μας οδηγεί στο να τα χαρακτηρίσουμε ως «το νέο πετρέλαιο» της σύγχρονης εποχής που διανύουμε.

Την τελευταία δεκαετία, είναι η έντονη η παρουσία των μεγάλων δεδομένων στη ζωή μας. Μάλιστα, τα εργαλεία με τα οποία γίνεται η ανάλυσή τους έχουν εξελιχθεί κατά πολύ βελτιώνοντας με τον τρόπο αυτό τη δουλειά των ερευνητών. Το Hadoop για παράδειγμα, είναι ένα από τα εργαλεία για τα οποίο θα μιλήσουμε αναλυτικά στην συνέχεια και ένα από τα πιο διαδεδομένα εργαλεία στην διαδικασία της κατανομής της πληροφορίας σε

διάφορους άλλους υπολογιστές (distributed computing). Δεν είναι το μοναδικό βέβαια. Άλλα εργαλεία που κάνουν παρόμοια εργασία είναι το Apache Spark καθώς και το Kafka. Όσον αφορά τις γλώσσες προγραμματισμού όπως ηPython ηR, η Javaακόμα και η Scalaήρθαν στη ζωή μας την τελευταία δεκαετία περίπου για να κάνουν την ανάγνωση και την ανάλυση των δεδομένων πιο γρήγορη.

Για έναν αρχάριο τα παραπάνω εργαλεία- λογισμικά δεν είναι απλά και χρειάζονται ώρες εξοικείωσης ώστε να προσαρμοστεί στα περιβάλλον τους και να μπορεί να τα χρησιμοποιήσει αποτελεσματικά.

1.2 Το αντικείμενο μελέτης

Ο σκοπός της παρούσας μελέτης είναι να γνωρίσουμε τα εργαλεία που συντελούν στην ανάλυση μεγάλων δεδομένων (Big Data),να δούμε πως μπορούν να εισαχθούν στις εταιρείες για την αύξηση της αποτελεσματικότητας τους σε σχέση με το μικρότερο ρίσκο, ενώ στη συνέχεια μέσα από μια εφαρμογή θα δώσουμε ένα παράδειγμα υλοποίησης τους ώστε να δούμε πως μπορούν να λειτουργήσουν στην πράξη. Η αναλυτική παρουσίαση του θέματος θα βοηθήσει τον αναγνώστη να καταλάβει την λειτουργία τους αναδεικνύοντας τη “μαγεία” των συναρπαστικών αποτελεσμάτων. Μέσα από την παρουσίασή τους και τα παραδείγματα που θα δώσουμε θα μιλήσουμε και για τον τρόπο με τον οποίο μπορεί το συγκεκριμένο αντικείμενο να διδαχθεί εφόσον κάποιος έχει τις απαραίτητες γνώσεις για την κατανόησή τους.

1.3 Εξόρυξη δεδομένων

Καθημερινά τα τελευταία χρόνια ακούμε την έκφραση "εξόρυξη δεδομένων" όλο και πιο συχνά. Η συγκεκριμένη ονοματολογία αναφέρεται σε πολλούς κλάδους όπως η ιατρική, τα οικονομικά, στον χώρο των καταναλωτικών αλλά και βιομηχανικών αγαθών, στον χώρο της ναυτιλίας ακόμα και στον πρωτογενή τομέα. Μελέτες που έχουν διεξαχθεί τα τελευταία χρόνια, έχουν δείξει πως επιχειρήσεις που έχουν εισάγει την ανάλυση δεδομένων, έχουν πετύχει με τη σωστή χρήση αύξηση στην αποτελεσματικότητά τους και κατά συνέπεια κέρδη καθώς γνωρίζουν τις αδυναμίες τους και πού πρέπει να εστιάσουν για να έχουν αυξημένη παραγωγικότητα με όσο το δυνατόν αρτιότερο αποτέλεσμα. Το μεγάλο στοίχημα όμως για τις επιχειρήσεις αποτελεί η ψηφιοποίηση των δεδομένων ώστε να μην υπάρχουν αποθήκες για την διατήρηση του αρχείου κάνοντας την δουλειά πιο σύντομη και με μεγαλύτερη ακρίβεια στην εύρεση της πληροφορίας και γενικά των αποτελεσμάτων. Έτσι εισήχθη και ο όρος «*Αποθήκευση και εξόρυξη δεδομένων*».

Η *εξόρυξη δεδομένων* (Data mining) ορίζεται ως η διαδικασία μη προφανούς εξαγωγής πληροφορίας από μεγάλες βάσεις δεδομένων που είναι υπαινισσόμενη άγνωστη εκ των προτέρων και χρήσιμη[1].

Έτσι όπως αντιλαμβανόμαστε οι βάσεις δεδομένων αντικαθιστούν το αρχείο καταγραφής πληροφοριών σε ένα ψηφιακό αποθετήριο - θησαυρό. Η εξόρυξη δεδομένων βέβαια έχει και κάποιους κοινούς στόχους και διαφορές με την μηχανική μάθηση.

Η εξόρυξη δεδομένων έχει ως στόχο να βρεθεί και να συλλεχτεί η πληροφορία η οποία στην συνέχεια θα περάσει στα χέρια των αναλυτών για να τους οδηγήσει σε αποφάσεις κρίσιμες η μη. Σε αντίθετο ρόλο, στην μηχανική μάθηση η εύρεση της πληροφορίας σκοπεύει στην βελτίωση της απόδοσης κάποιας τεχνητής οντότητας όπως ένα ρομπότ η ενός ευφυή πράκτορα κ.α. Για να γίνει πιο κατανοητό, στην εξόρυξη δεδομένων αναλύεται η πληροφορία και το αποτέλεσμα το οποίο εξάγεται έχει πιο θεωρητικό υπόβαθρο σε σχέση με την μηχανική μάθηση στην οποία χρησιμοποιούνται αλγόριθμοι τέτοιοι ώστε να δούμε φυσικά το αποτέλεσμα της οντότητας που έχουμε ορίσει[2].

[1] Αλέξανδρος Νανόπουλος - Ιωάννης Μανωλόπουλος «Εισαγωγή στην εξόρυξη και τις αποθήκες δεδομένων» (΄Έκδοση 2008, σελ.6)
[2] Αλέξανδρος Νανόπουλος - Ιωάννης Μανωλόπουλος «Εισαγωγή στην εξόρυξη και τις αποθήκες δεδομένων» (΄Έκδοση 2008, σελ.7)

1.4 Γιατί χρησιμοποιούμε τώρα τα μεγάλα δεδομένα?

Τα ερωτήματα που τίθενται είναι τα εξής: Αποτελούν τα μεγάλα δεδομένα παροδικό κύμα στον χώρο της επιστήμης; Είναι άλλη μια «μόδα» και θα περάσει; Γιατί δεν υπήρχαν και στο παρελθόν; Απαντήσεις στα παραπάνω ερωτήματα θα δοθούν στην συγκεκριμένη υποενότητα.

Τα μεγάλα δεδομένα, στην πραγματικότητα, δεν διαφέρουν σε τίποτα σε σχέση με το κομμάτι που μιλήσαμε στην προηγούμενη υποενότητα που αφορά τις αποθήκες και την εξόρυξη δεδομένων. Ονομάζονται έτσι, διότι ο όγκος δεδομένων που έχουμε όλα αυτά τα χρόνια είναι πολύ μεγάλος.

Στο παρελθόν, τέτοιου είδους αναλύσεις γίνονταν από στατιστικολόγους, αλλά τα αποτελέσματα ήταν πιο περιορισμένα καθώς η ανάλυσή τους ήταν δύσκολη καθώς η αναλογία δεδομένων- χρόνος ανάλυσης ήταν δυσανάλογη (πολλά δεδομένα- λίγος χρόνος για ανάλυση), με αποτέλεσμα να εισέρχονται καθημερινά δεδομένα, να αυξάνονται και να μην μπορούμε να προβούμε στην αποτελεσματική τους ανάλυση.

Ο κλάδος της πληροφορικής, με τα εργαλεία που έχουν δημιουργηθεί, κατάφερε να αυξήσει την εισαγωγή δεδομένων στα συστήματα σε μειωμένο χρόνο. Έτσι, σύμφωνα με στοιχεία της Google[3], παρατηρούμε πως ο κόσμος που ξεκινά και ασχολείται με τον κλάδο των μεγάλων δεδομένων και συγκεκριμένα με τα analytics έχει αυξηθεί λαμβάνοντας ως σημείο αναφοράς το 2011 και φτάνοντας στο σήμερα.

Όσον αφορά τον καθημερινό όγκο των δεδομένων, η εταιρεία IBM υπολογίζει πως 2,5 εκατομμύρια δεδομένα παράγονται σε καθημερινή βάση. Οι πιο συχνές πηγές δεδομένων προς ανάλυση είναι από αισθητήρες όπως είναι οι μετεωρολογικοί, οι αισθητήρες κίνησης. Άλλες πηγές δεδομένων είναι τα συστήματα παρακολούθησης και καταγραφής σημείων πώλησης (POS), οι δημοσιεύσεις σε social media όπως το Facebook και το Twitter.

Συνεπώς από τα παραπάνω συμπεραίνουμε πως τα μεγάλα δεδομένα δεν αποτελούν άλλη μια μόδα της σύγχρονης εποχής, προϋπήρχαν σαν τομέας μελέτης αλλά η ύπαρξή τους δεν ήταν ευρέως γνωστή. Σήμερα, με την συμβολή της πληροφορικής, ξεκίνησε να γίνεται ένα

[3] Τα μεγάλα δεδομένα από το 2004 μέχρι σήμερα: https://trends.google.com/trends/explore?date=all&q=big%20data

"trend" το οποίο φαίνεται μέχρι στιγμής πως θα έχει διάρκεια στο χρόνο και θα είναι αναγκαίο σε κάθε είδους επιχείρηση ανεξαρτήτως μεγέθους.

2. Η διαδικασία που ακολουθείται

2.1 Τρόποι συλλογής των δεδομένων

Στο τέλος του προηγούμενου κεφαλαίου παρουσιάστηκαν τα μέσα με τη βοήθεια των οποίων μπορούμε να προμηθευτούμε τα δεδομένα. Στο παρόν κεφάλαιο πρόκειται να αναλυθεί ο τρόπος με τον οποίο μπορούν τα συλλεχθούν τα δεδομένα, οι πηγές από τις οποίες επιτυγχάνεται αυτό καθώς και ο τρόπος εισαγωγής των δεδομένων σε ένα δικό μας αποθετήριο (στην περίπτωση που δεν υπάρχουν δικά μας δεδομένα).

Καταρχάς, θα πρέπει να γίνει κατανοητό πως η εξόρυξη δεδομένων δεν λειτουργεί με μαγικό τρόπο. Θα πρέπει να γνωρίζουμε τί εξορύσσουμε (τί δεδομένα ζητάμε). Για παράδειγμα, μπορεί να εξορύξουμε δεδομένα από ένα ψηφιακό αποθετήριο το οποίο να περιλαμβάνει οικονομικά δεδομένα αλλά τα αποτελέσματα που έχει μέσα να μην είναι τα επιθυμητά.

Μπορεί το συγκεκριμένο dataset να περιέχει δεδομένα που είτε είναι άσχετα με τις ιδιότητες που ζητούνται με βάση το σκοπό των αποτελεσμάτων που θέλουμε να εξάγουμε είτε να περιέχουν θόρυβο τα δεδομένα μας ή ακόμα και να επηρεάζουν αρνητικά τα συμπεράσματά μας. Για τον λόγο αυτό, όπως επισημάνθηκε και στην αρχή της εργασίας απαιτείται η γνώση δεδομένων ώστε να μην υπάρξουν αρνητικές συνέπειες.

Για την αντιμετώπιση των συγκεκριμένων αρνητικών συνεπειών στην εξόρυξη δεδομένων ακολουθούμε την διαδικασία που ονομάζεται KDD (Knowledge Discovery from Databases) ή αλλιώς στα ελληνικά «στάδια ανακάλυψης γνώσης από βάσεις δεδομένων» η οποία αναλύεται στις παρακάτω φάσεις:

(α) Ανάπτυξη και κατανόηση της περιοχής εφαρμογής: το συγκεκριμένο στάδιο, περιλαμβάνει την προετοιμασία των δεδομένων καθώς και τον τρόπο με τον οποίο θα γίνει ο μετασχηματισμός τους, τον αλγόριθμο που θα χρησιμοποιηθεί για το πρόβλημά μας καθώς και τον τρόπο αναπαράστασης των δεδομένων. Οι ενέργειες αυτές είναι απαραίτητες ώστε να μπορέσει ο τελικός αποδέκτης που θα λάβει το έργο μας να καταλάβει τους λόγους

χρησιμοποίησης των συγκεκριμένων τεχνικών. Για το συγκεκριμένο βήμα βέβαια υπάρχει περίπτωση να απαιτηθεί επανάληψη στην πορεία.

(β) Δημιουργία του κατάλληλου συνόλου δεδομένων: Στο συγκεκριμένο βήμα, θα πρέπει τα δεδομένα να εντοπιστούν (από κάποια βάση δεδομένων αν δεν τα παράγουμε εμείς με διάφορες τεχνικές) καθώς έχουν οριστεί για την δημιουργία της ανάλυσης της πληροφορίας. Το συγκεκριμένο βήμα είναι πάρα πολύ σημαντικό καθώς στην συγκεκριμένη βάση κατασκευάζονται τα μοντέλα τα οποία θα χρησιμοποιήσουμε για μελέτη. Σε περίπτωση που δημιουργηθούν προβλήματα (απώλεια χαρακτηριστικών) δημιουργούνται σφάλματα κατά την μελέτη. Έτσι τα χαρακτηριστικά πρέπει να είναι όσο το δυνατόν περισσότερα ώστε να παραχθεί μια πιο έγκυρη πληροφορία.

(γ) Προ επεξεργασία και καθαρισμός δεδομένων: Το συγκεκριμένο βήμα είναι ίσως το πιο σημαντικό στην διαδικασία καθώς είναι ιδιαίτερα κρίσιμη η αξιοπιστία των δεδομένων που χρησιμοποιούνται. Έτσι όπως θα πούμε και παρακάτω πρέπει να προχωρήσουμε σε καθαρισμό δεδομένων, σε διαχείριση των τιμών που λείπουν από τα δεδομένα μας, στην απομάκρυνση θορύβου καθώς επίσης και των ακραίων τιμών.

(δ) Μετασχηματισμός των δεδομένων: Προχωρώντας στο συγκεκριμένο βήμα τα δεδομένα μας μετασχηματίζονται σε δεδομένα κατάλληλα για εξόρυξη. Έτσι, οδηγούμαστε στην χρησιμοποίηση μεθόδων όπως η μείωση διαστάσεων και στον μετασχηματισμό χαρακτηριστικών. Με τον συγκεκριμένο τρόπο, πετυχαίνουμε την μείωση του αριθμού δεδομένων, με αποτέλεσμα να μένουν μόνο τα κατάλληλα δεδομένα προς εξέταση.

(ε) Επιλογή της κατάλληλης μεθόδου: Φτάνοντας σε αυτό το σημείο, θα πρέπει να αποφασίσουμε ποιον τύπο από αυτούς της ταξινόμησης, παλινδρόμησης και συσταδοποίησης θα πρέπει να χρησιμοποιήσουμε. Βασική προϋπόθεση στο συγκεκριμένο βήμα, είναι ο αλγόριθμος που χρησιμοποιείται για το μοντέλο εκμάθησης να είναι σε θέση να «μαθαίνει» πιο εύκολα και να μπορεί να χρησιμοποιηθεί με την ίδια ευκολία και στο μέλλον για τον συγκεκριμένο τύπο δεδομένων.

(στ) Επιλογή του κατάλληλου αλγορίθμου και εκτέλεσή του: Στο συγκεκριμένο βήμα πρέπει να πορευθούμε κανείς με βάση τα δεδομένα που θέλουμε να αναλύσουμε. Για να μπορούμε να έχουμε πιο ακριβή δεδομένα θα πρέπει να χρησιμοποιήσουμε νευρωνικά δίκτυα. Επιπλέον, για την καλύτερη κατανόηση της δομής του προβλήματος θα ήταν καλύτερο να

δημιουργηθούν «δέντρα» αποφάσεων. Η απόδοση των δεδομένων εξαρτάται σημαντικά από το ποιόν αλγόριθμο θα χρησιμοποιήσουμε για την εξαγωγή συμπεράσματος, ενώ στην εκτέλεση του αλγορίθμου είναι πιθανό να τρέξουμε τον αλγόριθμο μας πολλές φορές ώστε να "μάθει" και να έχουμε το επιθυμητό αποτέλεσμα.

(ζ) Αξιολόγηση του αποτελέσματος: Φτάνοντας στο τελικό στάδιο, γίνεται η εκτίμηση των δεδομένων που παράχθηκαν από τον αλγόριθμό μας. Αν πρέπει να προστεθούν δεδομένα, τότε θα πρέπει να επαναλάβουμε τα παραπάνω βήματα. Έτσι, όταν φτάσουμε σε ασφαλή συμπεράσματα, προχωράμε στην ανάλυσή τους και τεκμηριώνουμε την πρόβλεψή μας.

2.2 Κατηγοριοποίηση των μεθόδων εξόρυξης δεδομένων

Αναλογικά με το είδος των δεδομένων και το είδος γνώσης που έχουμε προς ανάλυση έχουμε και συγκεκριμένες μεθόδους εξόρυξης δεδομένων. Οι κυριότερες είναι οι παρακάτω :

Α. Κατηγοριοποίηση (classification): Η συγκεκριμένη λειτουργία έχει σκοπό την προγνωστική μέθοδο. Η διαδικασία της μπορεί να αποτυπωθεί με τον παρακάτω τρόπο. Ας υποθέσουμε πως έχουμε ένα σύνολο εγγράφων. Το σύνολο αυτό ονομάζεται σύνολο εκμάθησης (training set).Ο σκοπός μας στην κατηγοριοποίηση είναι η εύρεση ενός μοντέλου πρόβλεψης της τιμής (Forecasting) της ιδιότητας κλάσης από τις τιμές που υπάρχουν στις υπόλοιπες ιδιότητες, για δεδομένα που η τιμή τους δεν είναι γνωστή.

Ας πάρουμε σαν παράδειγμα το κομμάτι των πωλήσεων και ας σκεφτούμε πως έχουμε δύο κατηγορίες πελατών.

Πρώτη ομάδα είναι οι πελάτες που ενδιαφέρονται για την αγορά ενός υπολογιστή.

Δεύτερη ομάδα είναι αυτή που δεν ενδιαφέρεται.

Όπως γίνεται αντιληπτό, οι δύο ομάδες έχουν τα δικά τους χαρακτηριστικά. Μερικά από αυτά μπορεί να είναι τα εξής:

Πρώτη ομάδα:

- Χρόνος παραμονής στο προϊόν και τελική απόφαση
- Χρόνος παραμονής σε ανταγωνιστικά προϊόντα και τελική απόφαση
- Φορές επίσκεψης στο προϊόν και σύγκριση με ανταγωνιστικά προϊόντα

Δεύτερη ομάδα:

- Χρόνος παραμονής και τελική απόφαση.
- Φορές επίσκεψης προϊόντος και τελική απόφαση
- Ενδιαφέρον για ίδια προϊόντα ανταγωνισμού

Έτσι, με βάση τα στοιχεία που έχουμε συλλέξει από τις προηγούμενες εγγραφές μας, μπορούμε να δούμε ποιοι πελάτες ενδιαφέρονται για την αγορά υπολογιστή και οι οποίοι θα αποτελέσουν στόχο των διαφημίσεων με σκοπό την αγορά του προϊόντος. Μπορεί να γίνει μια δυναμική "επίθεση" για αγορά, σε σχέση με την δεύτερη ομάδα που οι διαφημίσεις θα

είναι πιο αραιά εμφανιζόμενες δεδομένου ότι ο πελάτης δεν ενδιαφέρεται και τόσο για την αγορά του προϊόντος.

Β. Ομαδοποίηση: Στην ομαδοποίηση τα πράγματα είναι διαφορετικά. Πολλές φορές, η έννοια της ομαδοποίησης συγχέεται με την κατηγοριοποίηση αλλά στην πραγματικότητα οι δύο έννοιες είναι εκ διαμέτρου αντίθετες. Ο σκοπός της ομαδοποίησης είναι να εντοπίσει ομάδες εγγραφών οι οποίες έχουν μεγάλη ομοιότητα μεταξύ τους, ενώ εγγραφές οι οποίες ανήκουν σε διαφορετικές ομάδες διαφέρουν σημαντικά. Αυτό μπορεί να γίνει κατανοητό, για παράδειγμα, στην επιστήμη της ιατρικής με βάση τα κύτταρα. Μπορούμε να δούμε ποια κύτταρα ανήκουν στην ομάδα των καρκινικών και ποια δεν ανήκουν.

Γ. Συσχέτιση: Στο κομμάτι της συσχέτισης τα πράγματα είναι διαφορετικά.

Στο στάδιο αυτό, ο σκοπός μας είναι να διαπιστώσουμε με γνώμονα τις εγγραφές που έχουμε, ποια στοιχεία συσχετίζονται μεταξύ τους. Πιο απλά, ας υποθέσουμε ότι έχουμε τον πελάτη Α που αγόρασε ένα βιβλίο. Η λειτουργίας της συσχέτισης είναι να «δει» τι αγόρασαν οι υπόλοιποι πελάτες σε σχέση με αυτόν και να ελέγξει ποιοι από αυτούς έχουν κοινά χαρακτηριστικά ώστε να διαπιστωθεί ποιοι θα μπορούσαν να αποτελέσουν δυνητικοί πελάτες παρόμοιων βιβλίων. Για παράδειγμα, αν ο πελάτης Α αγόρασε το βιβλίο Χ που ανήκει στην κατηγορία των αστυνομικών βιβλίων τότε και στον πελάτη Β που βλέπουμε ότι οι προτιμήσεις του είναι στα αστυνομικά βιβλία θα του προταθεί το βιβλίο του πελάτη Α ώστε δυνητικά στο μέλλον να προβεί σε αγορά αυτού.

Δ. Απλή Παλινδρόμηση: Η απλή παλινδρόμηση, είναι η διαδικασία στόχος της οποίας αποτελεί η μάθηση μιας συνάρτησης. Στην εν λόγω συνάρτηση, θα απεικονίζεται ένα αντικείμενο σε μια πραγματική μεταβλητή. Στόχος λοιπόν της απλής παλινδρόμησης είναι η πρόβλεψη των τιμών μιας εξαρτημένης μεταβλητής.

Ε. Άλλες λειτουργίες που χρησιμοποιούνται στην εξόρυξη δεδομένων, είναι ο χαρακτηρισμός (characterization) που αφορά την σύνοψη δεδομένων που έχουν ιδιότητα κλάσης και ο εντοπισμός ψευδών αντικειμένων (outlier detection).

3. Διαχωρισμός και περίληψη δεδομένων

Στην προηγούμενη ενότητα αναλύθηκε η διαδικασία συλλογής δεδομένων καθώς επίσης και τα στάδια που ακολουθούνται μέχρι την υλοποίηση και παρουσίαση του τελικού αποτελέσματος.

Στην συγκεκριμένη υποενότητα, θα παρουσιασθεί η έννοια και το περιεχόμενο των δεδομένων, οι ιδιότητες και τα χαρακτηριστικά τους.

Θα αναλυθεί ο τρόπος με τον οποίο εξετάζονται τα δεδομένα τα δεδομένα εν συνόλω, ενώ στο τέλος θα αναφερθεί το πιο σημαντικό κομμάτι των δεδομένων, που αποτελεί και το μεγαλύτερο κομμάτι της δουλειάς, ήτοι ο καθαρισμός τους.

3.1 Η μορφή και οι ιδιότητες των δεδομένων

Αφού είδαμε των κύκλο παραγωγής και εξόρυξης δεδομένων έως την τελική τους ανάλυση στην προηγούμενη ενότητα έστω και θεωρητικά, έχει έρθει η ώρα να αναφερθούμε στο τι καλούμε ως δεδομένα.

Δεδομένα λοιπόν, είναι η συλλογή αντικειμένων τα οποία έχουν κάποιες ιδιότητες και χαρακτηριστικά από τα οποία μπορούμε να εξάγουμε συμπεράσματα χρήσιμα προς διερεύνηση και μελέτη.

Πιο αναλυτικά αρκεί να σκεφτούμε τα δεδομένα ως ένα έγγραφο excel.Στο έγγραφο του excel έχουμε εγγραφές (δεδομένα) τα οποία φέρουν στην κορυφή τους τίτλους για την κάθε στήλη.

Έτσι, κάθε στήλη αποτελείται από τα χαρακτηριστικά, ενώ κάθε χαρακτηριστικό θεωρείται ως μεταβλητή στην οποία λαμβάνονται τιμές.

Οι μεταβλητές, αναλόγως των τιμές που ενδεχομένως θα λάβουν, διαχωρίζονται στις εξής κατηγορίες:

- Ποσοτικές
- Ποιοτικές

Ποσοτικές, καλούνται οι μεταβλητές οι οποίες μπορούν να μετρηθούν. Παραδείγματα ποσοτικών μεταβλητών, είναι το ύψος μιας μετοχής, το βάρος ενός ανθρώπου, το ύψος ενός αντικειμένου κ.α.

Ποιοτικές, καλούνται οι μεταβλητές οι οποίες είναι περιγραφικές. Ένα παράδειγμα, είναι τα χαρακτηριστικά ενός πληθυσμού τα οποία μεταβάλλονται όχι ως προς το μέγεθος αλλά ως προς την ποιότητα π.χ. το ανθρώπινο φύλο, το χρώμα δέρματος ενός ανθρώπου, το σχήμα των ματιών.

Οι παραπάνω μεταβλητές χωρίζονται σε υποκατηγορίες: Η ποσοτική μεταβλητή χωρίζεται στις διακριτές και στις συνεχείς μεταβλητές ενώ η ποιοτική στην ονομαστική μεταβλητή και στην μεταβλητή διάταξης.

Διακριτή μεταβλητή καλείται το σύνολο τιμών το οποίο είναι υποσύνολο των φυσικών αριθμών. Ένα παράδειγμα διακριτής μεταβλητής, είναι ο αριθμός των πράσινων φαναριών που συναντάμε σε μια διαδρομή.

Συνεχής μεταβλητή καλείται το σύνολο των τιμών είναι ένα συνεχές διάστημα. Ένα παράδειγμα συνεχής μεταβλητής είναι η διάρκεια της διαδρομής.

Ονομαστική μεταβλητή καλούνται οι κατηγορίες οι οποίες δεν μπορούν να συγκριθούν μεταξύ τους. Για παράδειγμα το χρώμα μαλλιών.

Μεταβλητή διάταξης καλείται η σαφής κατάταξη κατηγορίας. Ένα παράδειγμα, είναι η φυσική κατάσταση ενός ατόμου που οι κατηγορίες είναι καλή, μέτρια και άριστη φυσική κατάσταση.[4]

[4] Διαφάνειες μαθήματος «Ανάλυση δεδομένων με κλασσικές τεχνικές» της καθηγήτριας Γεωργία Φουτσιτζή, στο πλαίσιοτου μεταπτυχιακού προγράμματος.

3.2 Η παρουσίαση των δεδομένων

Η ανάλυση δεδομένων πολλές φορές ξεκινά από περιγραφικές αναλύσεις. Όσον αφορά την σύγκριση δύο μεγεθών, οι γραφικές παραστάσεις (για τις οποίες θα γίνει λόγος στη συνέχεια) δεν εξυπηρετούν για το σκοπό αυτό και για αυτό τον λόγο ο τρόπος με τον οποίο προχωράμε είναι αριθμητικός. Συνεπώς, η περιγραφική στατιστική είναι ένας τρόπος με τον οποίο επιτρέπεται η λεπτομερής οργάνωση, μελέτη και παρουσίαση των δεδομένων. Οι μέθοδοι αξιοποίησης είναι δύο. Πρώτη, είναι η αριθμητική μέθοδος ενώ δεύτερη είναι η γραφική μέθοδος.

Στην πρώτη μέθοδο έχουμε τα στατιστικά μέτρα τα οποία με τη σειρά τους είναι:

- Τα μέτρα κεντρικής τάσης (αριθμητικός μέσος, πληθυσμιακός μέσος, διάμεσος, επικρατούσα τιμή, κανονική κατανομή)
- Τα μέτρα απόκλισης (εύρος, διακύμανση, τυπική απόκλιση)
- Τα μέτρα συσχέτισης

Σε αντίθεση με την πρώτη μέθοδο, στην δεύτερη μέθοδο έχουμε τις γραφικές μεθόδους οι οποίες είναι:

- Οι συντελεστές μεταβλητότητας
- Τα ποσοστιαία σημεία (ενδοτεταρτημοριακό εύρος)
- Γραφικές παραστάσεις (Boxplot, ιστόγραμμα).

Όσον αφορά την πρώτη μέθοδο και ειδικότερα τα μέτρα κεντρικής τάσης, παρατηρείται μια εκτενή αναφορά των όρων που περιγράφονται κατωτέρω. Πιο συγκεκριμένα, δίδονται οι παρακάτω ορισμοί:

- **Αριθμητικός μέσος:** Καλείται ο αριθμός ο οποίος χαρακτηρίζει το κέντρο των μετρήσεων και υπολογίζεται διαιρώντας το άθροισμα μετρήσεων που υπάρχει στον δειγματικό μας χώρο.

- **Πληθυσμιακός μέσος:** Αν υποθέσουμε πως έχουμε ένα δείγμα με N μονάδες. Τότε, η μέση τιμή δίνεται από την παρακάτω σχέση.

$$\boldsymbol{\mu} = \frac{\sum \boldsymbol{\chi}}{\boldsymbol{N}}$$

- **Διάμεσος:** Καλείται ο αριθμός αυτός που χαρακτηρίζει το μέσο των διατεταγμένων μετρήσεων στο δείγμα μας, δηλαδή οι μισές μετρήσεις μας βρίσκονται στο αριστερό μέρος ενώ οι υπόλοιπες στο δεξί μέρος. Για να γίνει πιο κατανοητή η συγκεκριμένη μέθοδος αρκεί να δώσουμε ένα παράδειγμα. Έστω, ότι έχουμε τρείς αριθμούς σε ένα διάστημα {0,1,2} η διάμεσος των αριθμών είναι ο αριθμός 1, επειδή n=3 το δείγμα του διαστήματος αποτελείται από τρείς αριθμούς. Σε μια αντίθετη περίπτωση, αν είχαμε ένα διάστημα τι οποίο αποτελούνταν από τους αριθμούς {0,1,2,3,} στην περίπτωση αυτή, η διάμεσος είναι το 1,5 και αυτό επειδή το n = 4.

Συνεπώς αυτά τα οποία μπορούμε να καταλάβουμε για την διάμεσο είναι ότι:

- Μας δείχνει το κέντρο του δείγματος
- Δεν επηρεάζεται από ακραίες τιμές σε αντίθεση με τον αριθμητικό μέσο
- Όταν τα δεδομένα μας είναι μη συμμετρικά τότε η διάμεσος αντιπροσωπεύει καλύτερα το κέντρο. Ένα σύνηθες παράδειγμα είναι τα δεδομένα εισοδήματος τα οποία παρουσιάζουν πιο υψηλές τιμές από δεξιά.

- **Επικρατούσα τιμή:** Σε ένα σύνολο δεδομένων η τιμή η οποία εμφανίζεται πιο συχνά ονομάζεται επικρατούσα τιμή.

- **Κανονική κατανομή:** Στην κανονική κατανομή τα τρία μέτρα που μιλήσαμε παραπάνω ταυτίζονται στην μέση.

Όσον αφορά τη δεύτερη μέθοδο δίδονται οι παρακάτω ορισμοί:

- **Εύρος:** Καλείται η διαφορά που προκύπτει μεταξύ μέγιστης και ελάχιστης μέτρησης. Αυτό μπορούμε να το δούμε αν τοποθετήσουμε τα δεδομένα μας σε μια σειρά μεγέθους από το μικρότερο στο μεγαλύτερο ή και αντίστροφα.

- **Διακύμανση:** Η συγκεκριμένη μέθοδος αφορά τις τετραγωνικές αποστάσεις μετρήσεων από τον αριθμητικό μέσο. Ένα παράδειγμα το οποίο μπορούμε να θέσουμε είναι αν έχουμε δεδομένα πχ $x_1, x_2, x_3 \ldots\ldots x_n$ τότε ο υπολογισμός της διακύμανσης δίνεται από τον παρακάτω τύπο:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{1}{n-1}\left[\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2\right] = \left[\frac{\sum_{i=n}^{n} x_i^2 - n\,\bar{x}^2}{n-1}\right]$$

- **Τυπική απόκλιση:** Η τυπική απόκλιση είναι η τετραγωνική ρίζα της διακύμανσης και η σχέση που μπορούμε να την ορίσουμε είναι: $S = \sqrt{S^2}$. Η τυπική απόκλιση είναι ιδιαίτερα χρήσιμη στην εξαγωγή συμπερασμάτων επειδή μας δίνει τον τρόπο αποκλίσεων στην διεξαγωγή των μετρήσεών μας.

- **Ενδοτεταρτημοριακό εύρος:** Το ενδοτεταρτημοριακό εύρος εξαρτάται από το μέγεθος που έχει η υψηλότερη μέτρηση και η χαμηλότερη μέτρηση. Στην ουσία χωρίζουμε το δείγμα μας σε τρία σημεία στα Q_1, Q_2, Q_3. Έτσι ενδοτεταρτημοριακό εύρος (IR) είναι η διαφορά Q_3-Q_1.

3.3 Τα είδη των διαγραμμάτων στα δεδομένα

Η γνωστή ρήση που λεγόταν πιο παλιά «όσο αξίζει μια εικόνα δεν αξίζουν χίλιες λέξεις» σήμερα αποτελεί κανόνα. Αν συγκρίνουμε την απεικόνιση δεδομένων σε προγενέστερα περιοδικά με την σημερινή εποχή θα δούμε πως τάσεις, συγκρίσεις και μεταβολές παρουσιάζονται σε διαγραμματική απεικόνιση με ένα μόνο πάτημα του ποντικιού.

Η κατασκευή των διαγραμμάτων είναι εύκολη και η απεικόνιση των δεδομένων είναι φανταστική αρκεί να γνωρίζουμε το τι θέλουμε να δείξουμε με το κάθε διάγραμμα καθώς δεν κάνουν όλοι οι τύποι διαγραμμάτων για όλα τα δεδομένα.

Ιστόγραμμα

Η συγκεκριμένη διαγραμματική απεικόνιση χρησιμοποιείται κατά κύριο λόγο στην κατανομή συχνοτήτων. Το ιστόγραμμα είναι ένα διάγραμμα το οποίο έχει κάθετες στήλες που έχουν βάση τα διαστήματα τάξης και ύψος ανάλογο με την συχνότητα παρατήρησης του φαινομένου.

Ένας εναλλακτικός τρόπος απεικόνισης της κατανομής συχνοτήτων είναι η απεικόνιση με την πολυγωνική γραμμή. Η πολυγωνική γραμμή συνδέει τα κεντρικά σημεία των συχνοτήτων. Τα σημεία της πολυγωνικής γραμμής αντιστοιχούν στους κεντρικούς όρους των διαστημάτων τάξης.

Κυκλικό διάγραμμα

Ο συγκεκριμένος τύπος διαγράμματος είναι ευρέως διαδεδομένος στον διαχωρισμό ομάδων σε κατηγορίες.

Ακιδωτό διάγραμμα

Ο συγκεκριμένος τύπος διαγραμμάτων (ακιδωτό διάγραμμα) αποτελεί τον πιο συνηθισμένο τρόπο απεικόνισης κυρίως οικονομικών δεδομένων.

3.4 Καθαρισμός – μετασχηματισμός δεδομένων

Στις περισσότερες εξορύξεις δεδομένων, ενυπάρχουν παράγοντες όπως ο ανθρώπινος παράγοντας, προβλήματα που μπορεί να είχαν συσκευές μέτρησης (αισθητήρες) ο κακός σχεδιασμός διαδικασιών κ.ο.κ των οποίων τα αποτελέσματα συντελούν στην κακή ποιότητα

των δεδομένων. Αυτό σημαίνει πως τα δεδομένα μας μπορεί να επηρεάζονται από τα εξής στοιχεία:

- Θόρυβος στα δεδομένα (από ανθρώπινο λάθος η από λάθος συσκευής μέτρησης)
- "Ανώμαλες" τιμών (λάθη στο σχεδιασμό η λάθη στην εισαγωγή τους)
- Ελλιπείς τιμές (από κακό σχεδιασμό η από λάθη κατά την εισαγωγή τους)

Αν αυτά τα προβλήματα δεν αντιμετωπιστούν προτού προχωρήσουμε στην επεξεργασία και απεικόνιση των δεδομένων, τότε θα υπάρχει σοβαρό πρόβλημα στο αποτέλεσμα μας. Κατά συνέπεια ο καθαρισμός δεδομένων αποτελεί τη σημαντικότερη δουλειά καθώς αποτελεί ένα 70% της όλης διαδικασίας.

Για την αντιμετώπιση αυτού του σοβαρού προβλήματος αρκεί να προχωρήσουμε σε μερικές χρήσιμες τεχνικές. Βέβαια δεν είναι όλες οι τεχνικές οι ίδιες για όλα τα προβλήματα. Βέβαια, μαζί με τον καθαρισμό των δεδομένων μπορεί να χρειαστεί πολλές φορές να μετασχηματίσουμε τα δεδομένα μας. Για παράδειγμα, αν υπάρχουν μεταβλητές με πολύ διαφορετικά διαστήματα τιμών, ή αν ο αλγόριθμος που χρησιμοποιείται απαιτεί τα δεδομένα εισόδου να είναι αποκλειστικά και μόνο σε συγκεκριμένο διάστημα, τότε μπορούμε να εφαρμόσουμε κάποιον μετασχηματισμό στις τιμές για να γίνει εύκολα η δουλειά μας.

Αν προβούμε στις παραπάνω διαδικασίες δεν σημαίνει βέβαια πως πάντοτε τα προβλήματα που έχουμε θα εξαλειφθούν, για τον λόγο αυτό θα πρέπει να χρησιμοποιούνται αλγόριθμοι οι οποίοι θα είναι ανεκτικοί σε δεδομένα που περιέχουν προβλήματα. Κατωτέρω, αναλύονται τα παραπάνω κριτήρια που αναφέρθηκαν σχετικά με τον καθαρισμό δεδομένων.

3.4.1 Θόρυβος

Θόρυβος ονομάζεται είτε η τυχαία αλλοίωση τιμών, είτε η εισαγωγή αντικειμένων με τυχαίες τιμές. Αυτό έχει σαν αποτέλεσμα, τα δεδομένα με θόρυβο να μην ακολουθούν τα πρότυπα των υπολοίπων δεδομένων, κάνοντας έτσι πιο δύσκολο το έργο της εξόρυξής τους.

Οι τεχνικές που μπορούμε να χρησιμοποιήσουμε για την επεξεργασία σήματος, είναι πρώτον (α) ο κατακερματισμός σε διαστήματα και δεύτερον (β) ο στατικός εντοπισμός εξαιρέσεων.

(α) Κατακερματισμός σε διαστήματα. Στην κατηγορία αυτή, οι τιμές ενός χαρακτηριστικού ταξινομούνται και χωρίζονται σε διαστήματα. Στην συνέχεια τα κριτήρια επιλογής είναι:

(i) Η αντικατάσταση των τιμών με τον μέσο όρο του κάθε διαστήματος

(ii) Είτε η αντικατάσταση να γίνει με τις οριακές τιμές. Πιο αναλυτικά αν η τιμή πλησιάζει την μικρότερη τιμή τότε γίνεται αντικατάσταση με αυτή αλλιώς γίνεται αντικατάσταση με την μεγαλύτερη τιμή του διαστήματος.

(β) Στατικός εντοπισμός εξαιρέσεων: Στην συγκεκριμένη περίπτωση γίνεται εντοπισμός των ακραίων τιμών στο σύνολο με στατιστικό τρόπο. Για το κάθε χαρακτηριστικό, γίνεται υπολογισμός του μέσου όρου (μ_χ) και της τυπική απόκλισης (σ_χ). Έτσι, αν υπάρχουν τιμές οι οποίες είναι μικρότερες του διαστήματος $[\mu_\chi - \kappa * \sigma_\chi, \mu_\chi + \kappa * \sigma_\chi]$ τότε οι τιμές αυτές θεωρούνται ακραίες.

3.4.2 "Ανώμαλες" και ασυνεπείς τιμές

Στο συγκεκριμένο πρόβλημα, αν οι τιμές κάποιων ιδιοτήτων σε ένα αντικείμενο δεν ακολουθούν την κατανομή των τιμών των υπόλοιπων αντικειμένων τότε αυτές τις τιμές τις ονομάζουμε "ανώμαλες" τιμές και τα αντικείμενα τα οποία περιέχουν αυτές τις τιμές ονομάζονται "outliers". Βέβαια, οι τιμές αυτές συνήθως προκύπτουν από λάθη κατά την εισαγωγή τιμών.

Έτσι, με βάση τις συντεταγμένες των σημείων που εμφανίζονται, παρατηρούμε αν ακολουθεί την κανονική κατανομή η όχι, δηλαδή αν ένα σημείο απέχει κατά πολύ από τα υπόλοιπα και να διορθώσουμε την τιμή του. Ένας τρόπος είναι να πάρουμε την μέση τιμή του προηγούμενου και επόμενου σημείου ώστε να καλύψουμε την ακραία τιμή.

3.4.3 Ελλιπείς τιμές

Σε ένα σύνολο δεδομένων (Dataset) το οποίο κατεβάζουμε για την επεξεργασία δεδομένων, αποτελεί συχνό φαινόμενο κάποια αντικείμενα να περιέχουν ελλιπείς τιμές, δηλαδή χαρακτηριστικά με άγνωστες τιμές. Μερικές αιτίες, που ενδέχεται να οδηγήσουν στην έλλειψη αυτών των τιμών, είναι για παράδειγμα κάποια στοιχεία να είναι προσωπικά, ή και κάποια άλλα κατά τη διάρκεια ενός ερωτηματολογίου να μην σημειωθούν από τους ερωτηθέντες- χρήστες. Όπως γίνεται αντιληπτό, για την προσέγγιση και αντιμετώπιση του προβλήματος πρέπει να ακολουθηθούν τα εξής βήματα:

1) **Διαγραφή αντικειμένων με ελλιπείς τιμές.**
   Αν δούμε στο δείγμα που έχουμε ότι τα στοιχεία με ελλιπείς τιμές είναι μεγάλο το "χαμένο σύνολο" τότε το δείγμα μας δεν είναι αντιπροσωπευτικό και έτσι πρέπει να το διαγράψουμε ή να μην το χρησιμοποιήσουμε.

2) **Συμπλήρωση ελλιπών τιμών με τη μέθοδο της μέσης τιμής.**
   Το συγκεκριμένο ζήτημα προσεγγίσθηκε και ανωτέρω, ήτοι η συμπλήρωση των τιμών μπορεί να γίνει με τη μέθοδο της μέσης τιμής.

3) **Διατήρηση των ελλιπών τιμών.**
   Αυτό μπορεί να συμβεί, μόνο αν ο αλγόριθμος που χρησιμοποιείται μπορεί να λειτουργήσει με ελλιπείς τιμές. Βέβαια, αυτός δεν είναι ο καλύτερος τρόπος προσέγγισης του προβλήματος διότι όπως επισημάνθηκε ανωτέρω τα αποτελέσματα που θα εμφανισθούν δεν θα τύχουν ιδιαίτερης εγκυρότητας.

3.5 Μείωση αριθμού διαστάσεων

Πολλές φορές, ένα σύνολο δεδομένων το οποίο μπορεί να «κατέβει» από κάποια βάση δεδομένων ενδεχομένως να έχει πολύ μεγάλο αριθμό διαστάσεων. Τα σύνολα δεδομένων που υπάρχουν σε πολυμεσικές εφαρμογές (εφαρμογές που περιέχουν πολλαπλών μορφών περιεχόμενο όπως εικόνες, ήχο, κείμενο) έχουν μεγάλες διαστάσεις. Τέτοιου είδους παραδείγματα αποτελούν τα σύνολα δεδομένων που υπάρχουν στα αεροπλάνα αεροπορικών εταιρειών, σε βιολογικές εφαρμογές και συγκεκριμένα στην ανάλυση μικροστοιχείων, στις εφαρμογές διαχείρισης και ανάλυσης κειμένων, σε εφαρμογές οπτικοποίησης κλπ. Γίνεται λοιπόν αντιληπτό, πως ο αριθμός των όρων αλλά και το πλήθος διαστάσεων μπορεί να ξεπεράσει τις μερικές χιλιάδες.

Για την ορθότερη λειτουργία του αλγορίθμου, κρίνεται αναγκαία η μείωση του όγκου των διαστάσεων. Με τον τρόπο αυτό, τα δεδομένα αναμένεται να είναι πιο πυκνά, γι' αυτό και εκεί έχει λόγο ύπαρξης η έννοια της μείωσης αριθμού διαστάσεων, έτσι ώστε δηλαδή να γίνουν πιο αραιά. Παρατηρείται λοιπόν πως, όσο μεγαλώνει το πλήθος των διαστάσεων, τόσο τα δεδομένα απέχουν το ένα από το άλλο και έτσι είναι περισσότερο αραιά με αποτέλεσμα να καθίσταται δυσχερέστερο το έργο της εξόρυξης δεδομένων, ιδίως σε αλγορίθμους όπως η κατηγοριοποίηση και η ομαδοποίηση.

Για την επίλυση του συγκεκριμένου προβλήματος, ακολουθείται η μείωση του πλήθους των διαστάσεων. Τα πλεονεκτήματα που ανακύπτουν από την συγκεκριμένη μέθοδο είναι τα ακόλουθα:

1. Αφαίρεση άσχετων χαρακτηριστικών και μείωση του θορύβου.
2. Πιο κατανοητά αποτελέσματα λόγω του μικρότερου πλήθους χαρακτηριστικών.
3. Απεικόνιση του αποτελέσματος αν έχουμε αριθμό διαστάσεων μικρότερο ίσο του τρία.
4. Μικρότερος χώρος αναζήτησης άρα και μικρότερος χρόνος εκτέλεσης στον αλγόριθμο.

Ας επισημανθεί πως, αναφορικά με την μείωση του τρόπου διαστάσεων χρησιμοποιούμε την Ανάλυση Πρωταρχικών Συνιστωσών (PCA).

3.5.1 Ανάλυση πρωταρχικών συνιστωσών (PCA)

Στη συγκεκριμένη μέθοδο ακολουθείται μια τεχνική η οποία χρησιμοποιείται για συνεχή χαρακτηριστικά και εξάγει ένα πλήθος χαρακτηριστικών μειωμένο σε συνδυασμό πάντα με τα αρχικά δεδομένα που συλλαμβάνουν την μέγιστη διασπορά. Πιο αναλυτικά, οι τεχνικές που χρησιμοποιούνται είναι οι παρακάτω:

1. Αφαίρεση της μέσης τιμής.

   Στη συγκεκριμένη μέθοδο αυτό που γίνεται για κάθε διάσταση ξεχωριστά είναι ο υπολογισμός της μέσης τιμής και η αφαίρεσή της από τις τιμές της διάστασης.

2. Υπολογισμός του πίνακα συνδιασποράς.

   Ο πίνακας συνδιασποράς είναι ένας συμμετρικός πίνακας. Αυτό που απεικονίζει είναι κατά πόσο τα χαρακτηριστικά που υπάρχουν προς σύγκριση, διαφέρουν σε μεταβλητότητα. Δηλαδή, αν είναι πιο συγκεντρωμένα η όχι. Έτσι όταν είναι πιο κοντά τα σημεία συγκεντρωμένα μεταξύ τους τότε παρατηρείται μικρότερη διασπορά και το αντίθετο.

Αναφορικά με τον πίνακα συνδιασποράς, οι συνηθέστεροι τρόποι μέτρησης είναι πέντε:

- Το εύρος
- Η τεταρτημοριακή απόκλιση
- Η διακύμανση
- Η τυπική απόκλιση
- Ο συντελεστής μεταβλητότητας

3. Υπολογισμός ιδιοδιανυσμάτων και ιδιοτιμών.

Από το προηγούμενο βήμα του πίνακα συνδιασποράς, βρίσκουμε τα ιδιοδιανύσματα και τις ιδιοτιμές τα οποία απεικονίζονται με «*u*» και «*λ*»αντίστοιχα σε πίνακα διαστάσεων n*n. Τα ιδιοδιανύσματα, είναι ορθογώνια μεταξύ τους, κανονικοποιούνται και είναι μοναδιαία δηλαδή μήκος ίσον με 1.Εξετάζοντας τα ιδιοδιανύσματα που έχουν μεγαλύτερη ιδιοτιμή προκύπτουν κατευθύνσεις με μεγαλύτερη διασπορά δεδομένων.

Έτσι, με βάση τον μεγαλύτερο σε άξονα (άξονας με τη μεγαλύτερη τιμή) έχουμε και την μέγιστη δυνατή διασπορά.

4. Επιλογή νέων χαρακτηριστικών.

Γίνεται διάταξη των διανυσμάτων ως προς τις αντίστοιχες τιμές τους. Το πρώτο διάνυσμα ονομάζεται «κύρια συνιστώσα», το δεύτερο διάνυσμα «δεύτερη κύρια συνιστώσα» κ.ο.κ. Γίνεται κατανοητό πως, κάθε κύρια συνιστώσα είναι και ένα νέο χαρακτηριστικό του οποίου κάθε αρχικό σημείο είναι και ένας νέος συνδυασμός συνιστωσών. Κατά συνέπεια, όλες οι νέες συνιστώσες συνθέτουν ένα νέο σύνολο διαστάσεων. Συνθέτοντας όλες τις διαστάσεις σε ένα διάγραμμα διαπιστώνεται αν είναι συσχετισμένες η ασυσχέτιστες μεταξύ τους. Για να γίνει κατανοητό αυτό, εξετάζεται ο πίνακας συνδιασποράς.

5. Μείωση αριθμού διαστάσεων.

Αν οι διαστάσεις που υπάρχουν είναι μεγέθους ntότε θα προβούμε στην εφαρμογή της τεχνικήςPCAμε απώτερο στόχο τη διατήρηση μόνο των πρώτων κύριων συνιστωσών. Αφού σχηματίσουμε τον πίνακά μας και αναπαραστήσουμε τα αποτελέσματα, τότε τα νέα μονοδιάστατα σημεία αναπαριστώνται με το «+» ενώ αν θέλουμε να δείξουμε και τα σημεία των δύο διαστάσεων αυτά αναπαριστώνται με το «ο». Έτσι μπορούμε να προβούμε στη σύγκριση της διασποράς των δεδομένων της νέας διάστασης σε σχέση με τα αρχικά.

3.5.2Η επιλογή των χαρακτηριστικών

Εν συνεχεία της τεχνικής PCA, ένας ακόμα τρόπος μείωσης του πλήθους διαστάσεων είναι η επιλογή χαρακτηριστικών.

Στη συγκεκριμένη τεχνική δεν παράγονται διαστάσεις αλλά γίνεται προσπάθεια ανεύρεσης του βέλτιστου υποσυνόλου. Το υποσύνολο, είναι εκθετική συνάρτηση του πλήθους των αρχικών διαστάσεων.

Οι τρόποι αναζήτησης του υποσυνόλου είναι δύο:

- Φιλτράρισμα
- Μαύρο κουτί

3.6 Μέτρα ομοιότητας και απόστασης

Σχετικά με την συγκεκριμένη μέθοδο, η ομοιότητα είναι μια θεμελιώδης πράξη την οποία χρησιμοποιούμε σε περιπτώσεις όπως η οπτικοποίηση, ομαδοποίηση, κατηγοριοποίηση κ.λ.π. Χρησιμοποιώντας τη συγκεκριμένη μέθοδο, γίνεται προσπάθειας εξεύρεσης κατά πόσο δύο αντικείμενα έχουν όμοια χαρακτηριστικά. Όσο πιο πολύ μοιάζουν τα δύο αντικείμενα μεταξύ τους, τόσο πιο μεγάλη επιβάλλεται να είναι και η τιμή του μέτρου s (y,z). Συχνά, το συγκεκριμένο μέτρο περιορίζεται στο διάστημα [0,1].

Έτσι, στην περίπτωση του συνημίτονου παρατηρούμε το πόσο μεγάλη ή όχι είναι η ομοιότητα δύο αντικειμένων.

Στην περίπτωση της Ευκλείδειας απόστασης, παρατηρείται αν η διαφορά των μεγεθών των διανυσμάτων έχει μεγάλη απόσταση ή όχι.

3.7 Κριτήρια ομοιότητας

Όπως είπαμε και πιο πριν για να μπορέσουμε να αξιολογήσουμε πόσο κοντά είναι τα δεδομένα μας, θα πρέπει να υπάρχει κάποιο κριτήριο που να δείχνει την ομοιότητα των αντικειμένων. Τα πιο γνωστά κριτήρια ομοιότητας είναι τα παρακάτω:

Ευκλείδεια απόσταση: Η συγκεκριμένη μέθοδος είναι από τις πιο γνωστές συναρτήσεις και βασίζεται στην επανάληψη του πυθαγόρειου θεωρήματος και μετράει την απόσταση μεταξύ δύο σημείων στον δισδιάστατο χώρο.

Δίδεται από την εξής συνάρτηση:

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

Σχήμα 1.Ευκλείδια απόσταση

Απόσταση Manhatan:Χρησιμοποιείται κυρίως για προβλήματα λαβυρίνθων και είναι υπεύθυνη για τον υπολογισμό του πλήθους μετακινήσεων χωρίς να υπάρχουν εμπόδια. Δηλαδή, επιλύει το πρόβλημα του λαβύρινθου σαν να μην αντικρίζει εμπόδια μπροστά της.

Η απόσταση Manhatan δίδεται από την συνάρτηση:

$$D(x,y) = \sum_{i=1}^{n} |x_i - y_i|$$

Σχήμα 2.Απόσταση Manhatan

Το κριτήριο της μέγιστης απόστασης: Το συγκεκριμένο κριτήριο βασίζεται στην εύρεση της μέγιστης διαφοράς σε όλες τις διαστάσεις.

Η συνάρτηση από την οποία εκφράζεται είναι η εξής:

$$d(x,y) = max_{i=1}^{n} |x_i - y_i|$$

Σχήμα 3.Κριτήριο μέγιστης απόστασης

4. Γνωστότεροι κατηγοριοποιητές

Στο δεύτερο κεφάλαιο αναφέρθηκε ο όρος κατηγοριοποίηση και μέχρι το παρόν σημείο έγινε λόγος συχνά για την μέθοδο αυτή. Οι πιο γνωστοί κατηγοριοποιητές πάνω στους οποίους θα αναφερθούμε στην συνέχεια είναι οι παρακάτω:

- Δέντρα απόφασης: Ανήκει στους κατηγοριοποιητές του διαμερισμού περιοχών
- Bayesian κατηγοριοποιητής: Ανήκει στον τύπο εξέτασης κατανομών πιθανότητας
- K-Κοντινότεροι γείτονες: Ανήκει στον τύπο πλησιέστερων αντικειμένων.

Άλλοι κατηγοριοποιητές είναι:

- Μηχανές διανυσμάτων υποστήριξης (SVM) που χρησιμοποιούνται στον μη γραμμικό μετασχηματισμό για την απεικόνιση σε πολυδιάστατο χώρο αναζητώντας τον βέλτιστο γραμμικό διαχωρισμό.
- Νευρωνικά δίκτυα που λειτουργούν στα πρότυπα του ανθρώπινου εγκεφάλου.
- Κατηγοριοποιητές κανόνων που κάνουν αναπαράσταση του μοντέλου σε IF-THEN (κατηγοριοποιητές απόφασης).

4.1. Κριτήρια αξιολόγησης κατηγοριοποιητών

Αφού είδαμε ποιοι είναι οι πιο γνωστοί κατηγοριοποιητές ας δούμε ποια είναι τα κριτήρια που πρέπει να ικανοποιούν σε πολύ μεγάλο βαθμό τους κατηγοριοποιητές μας.

1. Ακρίβεια πρόβλεψης μοντέλου
2. Πίνακας σύγχυσης
3. Ευαισθησία
4. Εξειδίκευση
5. Ορθότητα

Ας τους αναλύσουμε έναν έναν ώστε να αποκτήσουμε μια ολοκληρωμένη αντίληψη.

- **Ακρίβεια πρόβλεψης μοντέλου**

Είναι ίσως το σημαντικότερο βήμα σε έναν κατηγοριοποιητή. Για να μετρήσουμε την ακρίβεια ενός μοντέλου θα πρέπει να κάνουμε τα εξής βήματα:

1. Στο σύνολο δεδομένων να είναι γνωστή η κλάση αντικειμένων.

2. Χωρίζουμε το σύνολο σε δύο ομάδες, σε αυτές της εκπαίδευσης (training set) και αυτές του ελέγχου (test set)
3. Το σύνολο εκπαίδευσης παρέχεται στον κατηγοριοποιητή που σχηματίζει το μοντέλο
4. Κάθε αντικείμενο που βρίσκεται στο σύνολο κατηγοριοποιείται με βάση το μοντέλο που χρησιμοποιούμαι. Αν η κλάση πρόβλεψης του μοντέλου είναι πράγματι η κλάση του αντικειμένου τότε έχουμε σωστή πρόβλεψη. Βέβαια η ακρίβεια της πρόβλεψης δίνεται από τη σχέση:

$$Ακρίβεια = \frac{Σωστές\ προβλέψεις}{Συνολικός\ αριθμός\ προβλέψεων}$$

Σχήμα 4.Μέτριση ακρίβειας πρόβλεψης

Η τιμή της ακρίβειας κυμαίνεται στις τιμές 0,1.

- **Ευαισθησία**

Στην ευαισθησία έχουμε δύο κλάσεις στο πρόβλημα μας (θετική-αρνητική).Η ευαισθησία μελετά τις συνέπειες της βέλτιστης λύσης του προβλήματος.Για παράδειγμα, μπορούμε να μελετήσουμε πόσοι άρρωστοι άνθρωποι είναι σε αυτή την κατάσταση, δηλαδή συνεχίζουν να νοσούν.

Το μέτρο της ευαισθησίας ορίζεται ως εξής:

$$sensitivity = \frac{|tpos|}{|pos|}$$

Σχήμα 5.Μέτρο ευαισθησίας

- **Εξειδίκευση**

Στην εξειδίκευση τα πράγματα είναι αντίθετα από την ευαισθησία.

Το μέτρο της εξειδίκευσης είναι:

$$specificity = \frac{|tneg|}{|neg|}$$

Σχήμα 6.Μέτρο εξιδίκευσης

- **Ορθότητα**

Τα κριτήρια αξιολόγησης των κατηγοριοποιητών κλείνουν με την ορθότητα. Η ορθότητα σε συνδυασμό με τα προηγούμενα κριτήρια, αντιπροσωπεύει την ακρίβεια όταν η ανισοκατανομή των κλάσεων που μελετάμε είναι μεγάλη.

Ο τύπος της ορθότητας είναι ο ακόλουθος:

$$precision = \frac{|tpos|}{|tpos| + |fpos|}$$

Σχήμα 7.Τύπος της ορθότητας

4.2. Bayesian κατηγοριοποιητής

Οι συγκεκριμένοι κατηγοριοποιητές βασίζονται στην εξέταση κατανομών πιθανότητας.

Για παράδειγμα, αν έχουμε μια ιδιότητα και μια κλάση π.χ. οικογενειακή κατάσταση (ιδιότητα) και κλάση αν αγόρασε η όχι ένα αυτοκίνητο.

Τότε,μπορούμε να κατανοήσουμε πόσα αντικείμενα έχει το σύνολο εκπαίδευσης και παρουσιάζεται η πιθανότητα μια κλάση να είναι θετική η αρνητική.

4.3. K-Κοντινότεροι γείτονες

Οι κοντινότεροι γείτονες κατηγοριοποιούν ένα αντικείμενο στην κλάση την οποία ανήκει η πλειοψηφία των k γειτόνων κοντά σε αυτό. Η ιδιότητα της κλάσης ακολουθεί την αρχή της τοπικότητας, που σημαίνει ότι παρόμοια αντικείμενα τείνουν να ανήκουν στην ίδια κλάση.

Για να γίνει η κατηγοριοποίηση των πλησιέστερων γειτόνων δίνουμε στην οικογενειακή κατάσταση τιμές 0, 0.5,1οι οποίες αντιστοιχούν στις ιδιότητες.

Για να πραγματοποιήσουμε την κανονικοποίηση, χρησιμοποιούμε μία ιδιότητα από τα δεδομένα όπως η ηλικία. Εφαρμόζοντας μετασχηματισμούς, κάθε αντικείμενο αποτελεί ένα σημείο στον δισδιάστατο Ευκλείδιο χώρο μεγέθους[0,1] * [0,1]. Έπειτα, θέτουμε στο k τιμή τέτοια ώστε να διαπιστωθεί πόσοι γείτονες είναι στην κλάση Ναι και πόσοι στην κλάση Όχι.

5. Ανθρώπινο δυναμικό και μεγάλα δεδομένα

Είναι αντιληπτό πώς τα μεγάλα δεδομένα πλέον έχουν επηρεάσει πολλούς κλάδους στην λήψη ορθολογικών αποφάσεων. Ένας από τους κλάδους εφαρμογής είναι και αυτός του ανθρώπινου δυναμικού.

Το φαινόμενο των μεγάλων δεδομένων ορίζεται ως το «χαρτοφυλάκιο πληροφοριών που χαρακτηρίζεται από υψηλό όγκο, ταχύτητα και ποικιλία και απαιτεί ειδικές μεθόδους για την μετατροπή σε χρήσιμη αξία»[5].

Στο εργασιακό κομμάτι, οι ρόλοι των Big Data jobs παρουσιάζουν μεγάλη ανάπτυξη. Ωστόσο η περιγραφή του ρόλου του υποψηφίου είναι συχνά ασαφής.

Ο ρόλος του data scientist είναι να συγκεντρώνει δεδομένα, να τα οργανώνει (φιλτράρει) σε αυτά που είναι χρήσιμα προς ανάλυση και στην συνέχεια να τα μετουσιώνει σε ιδέες χρήσιμες για την επιχείρηση που να μπορούν να δημιουργήσουν ένα οικονομικό πλεονέκτημα.

Ως προς το ζήτημα αυτό, διεξήχθη σχετική έρευνα από τους Andrea De Mauro , Marco Greco, Michele Grimaldi και Pavo Ritala[6] για να μπορέσουν να κατανοήσουν αλλά μέσω αυτών να κατανοήσουμε και εμείς το χάσμα που υπάρχει μεταξύ των αναγκών των επιχειρήσεων και του συγκεκριμένου αντικειμένου. Οι συγκεκριμένοι, συνέλεξαν από το διαδίκτυο πάνω από 2.700 θέσεις εργασίας από αγγελίες που περιείχαν την λέξη “BigData” είτε στον τίτλο της εργασίας είτε στην περιγραφή της και από εκεί με την μέθοδο του text mining και της ταξινόμησης κατέληξαν στο συμπέρασμα ποια δεδομένα ήταν χρήσιμα και έπρεπε να κρατήσουν.

Η έρευνά τους έδειξε ότι από τις αγγελίες ο όρος “Data scientist” για τους επιχειρηματίες και για το ανθρώπινο δυναμικό είναι ένας όρος πολύ γενικός που συνθέτει το σύνολο διασυνδεδεμένων δεξιοτήτων που απαιτείται από τις επιχειρήσεις για να εκμεταλλεύονται τα μεγάλα δεδομένα.

Ας ξεκαθαρίσουμε όμως μια για πάντα τι είναι τα μεγάλα δεδομένα. Τα «big data analytics»ξεκίνησαν σαν όρος να γίνονται γνωστά το 2011 και δημιουργήθηκαν από τους

[5]De Mauro, Greco, & Grimaldi, 2016, Human resources for Big Data professions: A systematic classification of job roles and required skill sets
[6]De Mauro, Greco, & Grimaldi, 2016, Human resources for Big Data professions: A systematic classification of job roles and required skill sets

όρους «πολύ μεγάλος όγκος δεδομένων» και «εξόρυξη δεδομένων» ωστόσο υπάρχουν κάποια χαρακτηριστικά που δίνουν μια σαφή έννοια για το αντικείμενο. Αυτά είναι:

- Πληροφόρηση
- Τεχνολογία
- Μέθοδοι
- Αντίκτυπο

1. Πληροφόρηση

Κατά τις δύο τελευταίες δεκαετίες παρατηρείται σημαντική αύξηση της ροής των δεδομένων καθώς επίσης και της υπολογιστικής ισχύος. Τα χαρακτηριστικά μεταβάλλονται γρήγορα, τα δεδομένα πλέον είναι περισσότερο "προσωπικά" και αυτό σημαίνει ότι καταναλώνονται περισσότερα δεδομένα από τον ίδιο τον άνθρωπο και σε σχέση με την αλληλεπίδρασή του με άλλους ανθρώπους καθώς επίσης και με μηχανήματα μέσω των προσωπικών τους συσκευών (laptops, smartphones). Πλέον τα δεδομένα έχουν αλλάξει σχετικά με το παρελθόν. Έχουν γίνει μη δομημένα και αυτό σημαίνει πως τα περισσότερα δεδομένα προέρχονται από ήχο/βίντεο, εικόνες καθώς επίσης και από ανθρώπινη ομιλία[7].Αυτό σημαίνει ότι τα δεδομένα βασίζονται στην πυραμίδα DIKW (Data, Information,Knowledge, Wisdom).

2. Τεχνολογία

Η ανάπτυξη ολοένα και πιο φθηνών και ισχυρών τεχνολογιών για την αποθήκευση, μετάδοση και επεξεργασία δεδομένων είναι ένας από τους θεμελιώδης παράγοντες που συντελούν στην αύξηση των μεγάλων δεδομένων. Η ικανότητα αποθήκευσης ολοκληρωμένων κυκλωμάτων έχει αυξηθεί εκθετικά τα τελευταία πενήντα χρόνια καθώς η πυκνότητα των τρανζίστορ διπλασιάζεται κάθε 24 μήνες σύμφωνα με το νόμο του Moore[8]. Έτσι, η επεξεργασία δεδομένων έχει γίνει ταχύτερη και φθηνότερη χάρη στην εξέλιξη της κατανεμημένης υπολογιστικής ισχύος και στα ταχύτερα δίκτυα.

Εργαλείο για το οποίο θα γίνει λόγος στο επόμενο κεφάλαιο και συνδέεται με τα Μεγάλα δεδομένα είναι το Hadoop το οποίο επιτρέπει τη συνεργασία συμπλεγμάτων κατανεμημένων μηχανημάτων με σκοπό την επίτευξη υψηλότερης απόδοσης μέσω παράλληλων

[7]Russom, Big Data Analytics, 2011
[8]Moore,2006

υπολογισμών[9]. Ένα άλλο χαρακτηριστικό της τεχνολογίας των μεγάλων δεδομένων είναι η εμφάνιση του cloud και η αποθήκευση των δεδομένων εκεί. Αυτό έχει ως συνέπεια την μείωση των οικονομικών δαπανών της επιχείρησης.

3. Μέθοδοι:

Για την χρήση των μεγάλων δεδομένων συνεπάγεται η υιοθέτηση νέων αναλυτικών μεθόδων για την μετατροπή μεγάλης σε όγκο ποσότητας πληροφοριών σε αξία για την επιχείρηση. Τα δεδομένα αυτά συνήθως δεν είναι δομημένα σωστά.. Οι τεχνικές επεξεργασίας μεγάλων δεδομένων που συναντάμε συχνά είναι[10]:

a. Η ανάλυση συμπλέγματος
b. Γενετικοί αλγόριθμοι
c. Επεξεργασία φυσικής γλώσσας
d. Αναγνώριση ομιλίας και εικόνας
e. Νευρωνικά δίκτυα
f. Προγνωστικό μοντέλο
g. Πρότυπα παλινδρόμησης
h. Ανάλυση κοινωνικού δικτύου
i. Ανάλυση συναισθημάτων
j. Επεξεργασία σήματος και απεικόνιση δεδομένων

4. Αντίκτυπο

Η άνοδος των μεγάλων δεδομένων έχει επηρεάσει κατά πολύ εκατομμύρια ανθρώπους εκτός της επιστήμης των οικονομικών των πολιτιστικών και κοινωνικών τόσο θετικά όσο και αρνητικά. Οι επιχειρήσεις, έχουν την ευκαιρία να αναπτυχθούν μέσω της αξιοποίησης της ανάλυσης των μεγάλων δεδομένων[11]. Τα μεγάλα δεδομένα μπορούν να οδηγήσουν τις επιχειρήσεις μέσω της μείωσης κόστους και της βέλτιστης λήψης αποφάσεων στην

[9]Davenport& Dyché, 2013
[10] Chen, Chiang, & Storey 2012, Manyika 2011, Marr 2015
[11]Daveport, 2014

βελτίωση των προϊόντων και των υπηρεσιών τους. Γι' αυτόν τον λόγο, τα μεγάλα δεδομένα είναι πιο παραγωγικά σχεδόν σε όλους τους κλάδους[12].

Η μεγαλύτερη ανησυχία με τα μεγάλα δεδομένα είναι η ιδιωτική ζωή. Σύνολα δεδομένων φέρνουν ψηφιακά ίχνη από τη ζωή του ατόμου τα οποία μπορούν να χρησιμοποιηθούν για την αποκάλυψη ιδιωτικών λεπτομερειών ή ακόμα και για πρόβλεψη της μελλοντικής συμπεριφοράς των ατόμων. Αυτό έχει ως αποτέλεσμα, την λήψη αποφάσεων στις επιχειρήσεις για την διατήρηση της ιδιωτικότητας των πελατών τους.

5.1. Μύθοι και αλήθειες των μεγάλων δεδομένων στο εργασιακό πλαίσιο

Η αύξηση της δημοτικότητας των μεγάλων δεδομένων ακολουθήθηκε από τη δημιουργία μιας άλλης έκφρασης που εμφάνισε έντονη αύξηση στα τέλη του 2012 και η ονομασία είναι "Επιστήμη δεδομένων" και "επιστήμονες δεδομένων" σύμφωνα με το Google Trend.

Οι Provostκαι Fawcettπρότειναν δύο λόγους πίσω από την έλλειψη σαφήνειας από την σχετική επιστήμη των δεδομένων. Πρώτον, η επιστήμη δεδομένων είναι συχνά συνδεδεμένη και συγχέεται πολλές φορές με τα μεγάλα δεδομένα και την διαδικασία λήψης αποφάσεων στα μεγάλα δεδομένα. Δεύτερον, το επιστημονικό κομμάτι του αντικειμένου βρίσκεται στο στάδιο πρακτικού και πειραματικού σταδίου έναντι του θεωρητικού και μεθοδολογικού σταδίου.

Η επιστήμη των δεδομένων δύναται να χωριστεί σε δύο ομάδες. Η πρώτη, περιλαμβάνει τις ικανότητες που αφορούν την μετατροπή ακατέργαστων δεδομένων σε ιδέες, δίδοντας έμφαση στην τεχνολογία και τις μεθόδους. Στην συγκεκριμένη ομάδα, περιλαμβάνονται τεχνικές δεξιότητες όπως η ικανότητα χρήσης εργαλείων μεγάλων δεδομένων και επάρκεια σε στατιστικές μεθόδους εκμάθησης μηχανών. Η ομάδα αυτή απαιτεί τεχνικές δεξιότητες («hard skills») αλλά και ειδικές γνώσεις δεδομένων και αναλύσεων.

Η δεύτερη ομάδα αποτελείται από την ικανότητα μετατροπής της γνώσης σε οργανωτική δημιουργία αξίας, στην οποία συμμετέχουν τα κατάλληλα άτομα και οι βασικές επιχειρηματικές διαδικασίες στον οργανισμό. Η ομάδα αυτή, περιλαμβάνει «χαλαρές»

[12]Bughin, 2016

δεξιότητες («soft skills»), κυρίως στον τομέα της επικοινωνίας, γενικές τεχνικές διαχείρισης και γνώσεις ενός συγκεκριμένου επιχειρηματικού τομέα.

Όπως επίσης παρατηρείται, ο μοναδικός ρόλος της επιστήμης των δεδομένων, δεν είναι να υπολογίζει το πλήρες φάσμα των απαιτήσεων των ταλέντων που αντιμετωπίζουν οι εταιρείες στον τομέα των Big Data Analytics.

Τέλος η έρευνα που μπορεί να γίνει εκτυλίσσεται στα εξής στάδια:

Πρώτον, πραγματοποιείται λήψη σημαντικού αριθμού σχετικών θέσεων εργασίας μέσω διαδικτύου, μέσω τεχνικών αποξήλωσης.

Δεύτερον, αναλύονται οι λέξεις που εμφανίζονται στους τίτλους των θέσεων εργασίας προκειμένου έπειτα να κατηγοριοποιηθούν σε ορισμένες οικογένειες. Τρίτον, προσδιορίζονται τα σχετικά σύνολα ικανοτήτων μέσω της εφαρμογής ενός αλγορίθμου μοντελοποίησης θέματος στο περιεχόμενο των θέσεων εργασίας. Τέλος, το τέταρτο βήμα περιλαμβάνει την αξιολόγηση της σχετικής σημασίας των δεξιοτήτων, αναλύοντας το μέσο βαθμό παρουσίας κάθε συνόλου δεξιοτήτων για τις θέσεις εργασίας.

6. Ανάλυση δεδομένων στο cloud

Η εποχή των πληροφοριών επιφέρει την έκρηξη μεγάλων δεδομένων από πολλαπλές πηγές σε κάθε πτυχή της ζωής μας: τα σήματα ανθρώπινης δραστηριότητας, οι φορητοί αισθητήρες, τα πειράματα από την έρευνα εντοπισμού σωματιδίων και τα χρηματιστηριακά δεδομένα είναι μόνο μερικά παραδείγματα. Οι πρόσφατες τάσεις δείχνουν ότι τα επόμενα χρόνια θα συνεχιστεί η εκθετική αύξηση των δεδομένων και υπάρχει μεγάλη ανάγκη να βρεθούν αποτελεσματικές λύσεις για την αντιμετώπιση πτυχών όπως η αποθήκευση δεδομένων, η επεξεργασία σε πραγματικό χρόνο, η εξαγωγή πληροφοριών και η αφηρημένη δημιουργία μοντέλων.

Δύο από τα συνήθως χρησιμοποιούμενα πλαίσια υπολογιστικών συμπλεγμάτων για ανάλυση μεγάλων δεδομένων είναι το Apache Spark που αποτελεί μια ανεπτυγμένη πλατφόρμα από την UC Berkley, που εκμεταλλεύεται υπολογιστικά μέσα στη μνήμη για την επίλυση επαναληπτικών αλγορίθμων και μπορεί να εκτελεστεί σε παραδοσιακά συμπλέγματα όπως το Hadoop και το OpenMP/MPI που αξιοποιεί αποτελεσματικά τις αρχιτεκτονικές πολλαπλών πυρήνων συμπλεγμάτων, όπως Beowulf, και συνδυάζει το

πρότυπο MPI με την πολλαπλή επεξεργασία κοινόχρηστης μνήμης. Οι κύριες διαφορές μεταξύ αυτών των δύο πλαισίων είναι η υποστήριξη ανοχής σφαλμάτων και η αναπαραγωγή δεδομένων. Το Spark, τις αντιμετωπίζει αποτελεσματικά αλλά με σαφή μειονέκτημα στην ταχύτητα. Αντιθέτως, το OpenMP/MPI παρέχει μια λύση που προσανατολίζεται κυρίως σε υπολογιστικές εργασίες υψηλής απόδοσης αλλά είναι πιθανό να υποστεί βλάβες, ιδίως εάν χρησιμοποιείται σε υλικό βασικών προϊόντων. Μέχρι στιγμής, δεν έχει διερευνηθεί η σύγκριση μεταξύ αυτών των δύο πλαισίων.

Μία από τις πρόσφατες λύσεις για ανάλυση μεγάλων δεδομένων είναι η χρήση υπολογιστικού νέφους, η οποία καθιστά διαθέσιμη εκατοντάδες ή χιλιάδες μηχανήματα για την παροχή υπηρεσιών όπως η πληροφορική και η αποθήκευση. Οι παραδοσιακές εσωτερικές λύσεις απαιτούν γενικά μεγάλες επενδύσεις σε υλικό και λογισμικό κα ι πρέπει να αξιοποιηθούν πλήρως προκειμένου να είναι οικονομικά βιώσιμες. Αντ' αυτού, πλατφόρμες cloud, συνήθως φιλοξενούνται από εταιρείες τεχνολογίας πληροφορικής όπως η Google, η Amazon και η Microsoft, σε προσιτές τιμές σε άτομα και οργανισμούς σύμφωνα με τις ανάγκες τους: χρόνος, αριθμός μηχανημάτων, τύποι μηχανημάτων κ.λπ.

Έτσι, συγκρίνονται οι δύο τεχνολογίες παραλληλισμού, το Spark στο Hadoop και το MPI/OpenMP στοBeowulf, εκτελώντας δύο εποπτευόμενους αλγόριθμους μάθησης (KNN και αλγόριθμοι SVM-Pegasos) για τη δοκιμή της κάθετης και οριζόντιας δυνατότητας κλιμάκωσης του συστήματος. Η ανάπτυξη γίνεται σε εικονική μηχανή και συγκεκριμένα στο Google cloud platform.

Εν κατακλείδι, επισημαίνουμε ότι ακόμη και αν και η Spark στο Hadoop με επεξεργασία δεδομένων στη μνήμη μειώνει το χάσμα μεταξύ των Hadoop Map Shrine και HPC για την εκμάθηση μηχανών, απέχουμε πολύ από την επίτευξη σύγχρονης απόδοσης τεχνολογιών HPC. Παρ' όλα αυτά, η Spark στο Hadoop προτιμάται επειδή:

- Προσφέρει ένα κατανεμημένο σύστημα αρχείων με διαχείριση αποτυχιών και αναπαραγωγής δεδομένων
- Επιτρέπει την προσθήκη νέων κόμβων κατά το χρόνο εκτέλεσης.
- Παρέχει ένα σύνολο εργαλείων για ανάλυση και διαχείριση δεδομένων που είναι εύκολο στη χρήση, την υλοποίηση και τη συντήρηση.

Μέχρι στιγμής, δεν έχει προταθεί ενσωμάτωση μεταξύ Hadoop και MPI/OpenMP. Η ιδέα αυτή αποτελεί ένα ενδιαφέρον θέμα έρευνας, διότι θα μπορούσε να βελτιώσει σημαντικά την ταχύτητα.

7. Hadoop και Apache spark

7.1 Εισαγωγή

Ένα κατανεμημένο υπολογιστικό σύστημα είναι αυτό που για την λειτουργία του συντελούν πολλαπλά λογισμικών καταναλώνοντας πολλαπλούς πόρους.

Οι υπολογιστές που αποτελούν ένα κατανεμημένο σύστημα, μπορούν να λειτουργούν ως ένας υπολογιστής που είναι συνδεδεμένοι σε ένα τοπικό δίκτυο, ή να είναι απομακρυσμένα συνδεδεμένοι σε δίκτυο ευρείας περιοχής. Στόχος του συστήματος αυτού είναι η λειτουργία ενός και μόνο υπολογιστή. [IBM(n.d)].

Τα πλεονεκτήματα που προσφέρουν τα κατανεμημένα συστήματα συγκριτικά με τα κεντρικά είναι τα παρακάτω:

- Επεκτασιμότητα (scalability)

  Το σύστημα μπορεί να επεκταθεί - ανάλογα με τις ανάγκες - με την απαίτηση προσθήκης περισσότερων συσκευών.

- Πλεονασμός (redundancy)

  Όλοι οι υπολογιστές παρέχουν τις ίδιες λειτουργίες έτσι ώστε όταν κάποιος υπολογιστής πάθει κάποια βλάβη, η εργασία που εκτελούνταν από αυτόν, θα πρέπει να γίνεται από κάποιον άλλον υπολογιστή. Κατά συνέπεια, ο ένας υπολογιστής πρέπει να είναι σε θέση να αναπληρώνει το κενό του άλλου σε μια τέτοια περίπτωση.

7.2 Hadoop

Η ανάπτυξη του συγκεκριμένου λογισμικού ξεκίνησε το 2005 και αποτελεί ένα πρόγραμμα ανοιχτού κώδικα (open source). Σκοπός του λογισμικού αυτού, είναι η επεκτάσιμη κατανεμημένη υπολογιστική με αξιόπιστο τρόπο. Το συγκεκριμένο πρόγραμμα

ανταπεξέρχεται ικανοποιητικά στην αποθήκευση, στον διαμοιρασμό μεγάλων δεδομένων και στην επεξεργασία τους. Γίνεται λοιπόν αντιληπτό πως με το Hadoop πετυχαίνουμε απόδοση και οικονομία στις λειτουργίες μας.

Η απόδοση στο Hadoop επιτυγχάνεται μέσω της εκτέλεσης παράλληλων διεργασιών.

Ο τρόπος με τον οποίο λειτουργεί είναι πραγματικά εκπληκτικός. Τα δεδομένα δεν χρειάζεται να μετακινούνται μέσω του δικτύου ώστε να φτάσουν στο κεντρικό σημείο επεξεργασίας, αντιθέτως οι διεργασίες χωρίζονται σε μικρότερες και κατανέμονται στους χρήστες ώστε στο τέλος να συνδεθούν ξανά και να εξάγουμε το αποτέλεσμα[13].

7.2.1 Τα εργαλεία του συστήματος.

Όπως έχει προαναφερθεί και σε προηγούμενες ενότητες, ο παγκόσμιος ιστός δημιουργεί κάθε δευτερόλεπτο πολλά δεδομένα χρήσιμα προς διερεύνηση.

Μία από τις εταιρείες κολοσσός στον τεχνολογικό χώρο, η Google, προχώρησε στην δημιουργία και διατήρηση ευρετηρίου. Το συγκεκριμένο εγχείρημα ήταν πολύ δύσκολο καθώς η συνεχόμενη διατήρηση και ενημέρωση του συστήματος ήταν εξαιρετικά δύσκολη και κατά συνέπεια κατανάλωνε πολλούς υπολογιστικούς πόρους. Για τον λόγο αυτό, η συγκεκριμένη εταιρεία σκέφτηκε κάτι καινοτόμο και άκρος αποτελεσματικό: προχώρησε στη δημιουργία του κατάλληλου λογισμικού επεξεργασίας, γνωστό και ως “MapReduce”. Από το όνομα και μόνο καταλαβαίνουμε πως πρόκειται για κάποια ταξινομημένη συρρίκνωση.

Πράγματι, ο προγραμματισμός του συγκεκριμένου προγράμματος και οι λειτουργία του είναι η μετατροπή του συνόλου δεδομένων σε ζεύγη τιμών – κλειδιών, που αποτελεί την ταξινόμηση (map) και φυσικά μια εργασία μείωσης (reduce) που συνδυάζονται αμφότερα ώστε να έχουμε ένα και μόνο αποτέλεσμα. Έτσι ξεκίνησε και η ανάπτυξη του Hadoop που αποτελείται από τρία modules.

1. Hadoop common: Το Hadoop common είναι η βιβλιοθήκη και τα εργαλεία υποστήριξης των παρακάτω modules.

[13]Gilbert E., (2016)

2. MapReduce: Όπως είπαμε και νωρίτερα το συγκεκριμένο module του Hadoop αποτελεί την μέθοδο map που είναι υπεύθυνη για το φιλτράρισμα και την ταξινόμηση των δεδομένων και την μέθοδο reduce που υπολογίζει το συνολικό αποτέλεσμα της διεργασίας. Χρησιμοποιείται κυρίως σε μεγάλο όγκο δεδομένων και δημιουργήθηκε από την Google πριν από το Hadoop το 2004.

3. Hadoop Distributed File System (HDFS). Το συγκεκριμένο module αποτελεί επίσης ένα από τα πιο γνωστά κομμάτια του Hadoop καθώς είναι υπεύθυνο για την κατανεμημένη αποθήκευση μεγάλων δεδομένων. Η λειτουργία του, είναι ο διαχωρισμός δεδομένων και η διανομή σε υπολογιστές (clustering).
   Το καλό της λειτουργίας του συγκεκριμένου moduleείναι η δημιουργία πολλαπλών αντίγραφων για παροχή ασφάλειας. Έτσι, αν για παράδειγμα ένας κόμβος αποτύχει, τα αρχεία δεν θα χαθούν αλλά η συγκεκριμένη εργασία θα ανατεθεί σε άλλο κόμβο. Τα δεδομένα μπορεί να είναι είτε δομημένα είτε μη δομημένα. Επιπλέον είναι δυνατή η υποστήριξη οποιασδήποτε μορφής δεδομένων.

4. Yet Another Resource Negotiator (YARN): Αποτελεί έναν ακόμα διαπραγματευτή πόρων όπως αναφέρει και η ονομασία του. Το YARN, εισήχθη στην έκδοση του Hadoop 2.0 και οι υπηρεσίες που παρέχει αφορούν τον προγραμματισμό διεργασιών και την διαχείριση των υπολογιστικών πόρων στο cluster του Hadoop. Αυτό που κάνει το YARN μοναδικό, είναι ότι μπορεί να «τρέξει» και άλλα frameworks εκτός του MapReduce, έχοντας ως αποτέλεσμα την διαδραστικότητα υπολογισμών σε πραγματικό χρόνο.

   Όλες οι λειτουργίες του Hadoop μπορούν να γίνουν και από έναν μόνο υπολογιστή αλλά δεν θα είναι δυνατή η ακριβής κατανόηση του τρόπου λειτουργίας του. Η καλύτερη επιλογή είναι η λειτουργία σε μεγάλο δίκτυο υπολογιστών, της οποίας τα δεδομένα που διαχειρίζονται από το Hadoop Distributed File System (HDFS) και συνοδεύεται από την ύπαρξη ενός master Name Node (κύριος υπολογιστής)που περιέχει όλο το αρχείο. Τα δεδομένα αποθηκεύονται σε slave Data Nodes και τα clusters διανέμονται σε χιλιάδες κόμβους εργασίας[14].

[14]Gilbert E.,(2016)

7.2.2 Άλλες λειτουργίες του Hadoop.

Η Apache"έχτισε" διάφορα εργαλεία γύρω από το Hadoop ώστε να του παρέχει την μεγαλύτερη δυνατή υποστήριξη, τα οποία συντελούν στην διευκόλυνση του τομέα DataAnalytics. Τα εργαλεία αυτά είναι:

- Oozie: Η λειτουργία του προγράμματος Oozie είναι ο χρονοπρογραμματισμός εργασιών που είναι υπεύθυνος για την διαχείριση εργασιών του Hadoop.

- Pig:Δεν πρόκειται για κάποιο γουρούνι αν και το λογότυπό του αυτό είναι αλλά αποτελεί μια scripting language διαχείρισης δεδομένων με λειτουργίες ETL (extract ,transform , load). Με το pigμπορεί κάποιος να γράψει scripts τα οποία στην συνέχεια μπορούν να μετατραπούν σε εργασίες MapReduce και η γλώσσα με την οποία γράφονται τα scripts ονομάζεται "PigLatin".

- Hive: Το Hive αποτελεί μια γλώσσα ερωτημάτων η οποία μοιάζει με την SQL.Το κοινό στοιχείο αφορά την σύνταξη της γλώσσας (select, fromκ.α.). Το Hive μπορεί να συνδεθεί με μια αποθήκη δεδομένων η οποία βρίσκεται στο Hadoop.

- Sqoop: Τέλος, το Sqoop είναι υπεύθυνο για την μεταφορά των δεδομένων μεταξύ Hadoop και της βάσης δεδομένων που έχουμε στο σύστημά μας.

Βέβαια, εκτός από όλα τα παραπάνω, για την τεχνολογία του Hadoopδεν έχει ενδιαφερθεί μόνο η Apacheπου έχει το προϊόν. Εταιρείες όπως η Amazon,η Google και άλλες έχουν εντάξει το συγκεκριμένο εργαλείο στο σύστημά τους και το παρέχουν σε συνδυασμό με τις δικές τους τεχνολογίες. Ας δούμε μερικούς από τους προμηθευτές και τις υπηρεσίες τους:

- ✓ Cloudera: Η Cloudera αποτελεί έναν από τους συν-δημιουργούς του προϊόντος Hadoop. Παρέχει στις επιχειρήσεις πλήρες περιβάλλον υποστήριξης του Hadoop

- ✓ Google cloud: Η συγκεκριμένη υπηρεσία παρέχεται από την Googlekαι αποτελεί έναcloudχώρο ο οποίος για όποιον ενδιαφέρεται για την πρώτη χρονιά είναι δωρεάν (του παρέχει η google 273€ για υπηρεσίες στην πλατφόρμα) και παρέχει εργαλεία

χρήσιμα όπως το kubernets, sql, υπηρεσίες δικτύου καθώς επίσης και μια μεγάλη σουίτα για μεγάλα δεδομένα καθώς επίσης και ένα cluster Hadoop.

- ✓ Amazon AWS: Είναι ο ανταγωνιστής του Google cloud, κάνει περίπου τα ίδια πράγματα με το προϊόν Google cloud και παρέχει και αυτή στις υπηρεσίες της cluster Hadoop προσθέτοντας πόρους και υποστήριξη.

7.2.3 Πλεονεκτήματα και το μειονέκτημα του Hadoop.

Δίχως αμφιβολία, δε θα χρησιμοποιούσαμε το Hadoop αν δεν είχε πολλά πλεονεκτήματα στην επιστήμη των μεγάλων δεδομένων (Big Data science).

Μερικά από αυτά είναι

- ✓ Είναι επεκτάσιμο: Το Hadoop δεν έχει περιορισμό στην προσθήκη διακομιστών και αυτό διότι όσο περισσότεροι διακομιστές μπουν τόσο αποδοτικότερο θα γίνει το σύστημά μας σε υπολογιστική ταχύτητα και αποθήκευση.

- ✓ Είναι οικονομικά αποδοτικό: Το Hadoop είναι οικονομικά αποδοτικό καθώς χρησιμοποιεί commodity hardware δηλαδή με ένα σχετικά φθηνό υλικό μπορείς να κανείς να εκτελέσει τις εργασίες που επιθυμεί σαν να διαθέτει έναν υπερ-υπολογιστή.

- ✓ Ευελιξία άλλων τεχνολογιών: Παρόλο που χρησιμοποιείται με τα παραπάνω λογισμικά που έχουν φτιαχτεί για αυτό, το Hadoop μπορεί να δουλέψει με αποδοτικό τρόπο και με άλλα εργαλεία. Για παράδειγμα μπορεί να συνεργαστεί με το Sparkκαθώς επίσης όπως είπαμε παραπάνω και οι διεργασίες που κάνει στον τύπο δεδομένων δύναται να είναι δομημένες ή μη.

- ✓ Αποδοτική επίλυση προβλημάτων: Το Hadoop μπορεί να επιλύει προβλήματα μοιράζοντάς τα σε πολλούς κόμβους για την διεξαγωγή των εργασιών και κατά συνέπεια και των αποτελεσμάτων. Με αυτό τον τρόπο, μειώνει τον χρόνο επεξεργασίας και υπολογισμού καθώς επίσης και τον χρόνο μεταφοράς στο δίκτυο.

Ένα μειονέκτημα του Hadoop είναι το εξής:

- ✓ Το συγκεκριμένο σύστημα ίσως δεν είναι κατάλληλο για να αποθηκεύσει μια επιχείρηση η ένας ιδιώτης τα δεδομένα του όσο μεγάλα ή μικρά είναι, καθώς η ασφάλεια που παρέχει η προεπιλεγμένη ρύθμιση παραμέτρων απενεργοποιεί τις δικλείδες ασφαλείας. Έτσι, πρέπει τα δεδομένα να αποθηκευτούν σε μια άλλη βάση δεδομένων στην οποία παρέχεται μεγαλύτερη κρυπτογραφία και ασφάλεια.

7.2.4 Πού χρησιμοποιείται το Hadoop.

Όπως αναφέραμε και ανωτέρω, το Hadoop το χρησιμοποιούν μεγάλοι οργανισμοί όπως η Google, Amazonκ.α.. Το Hadoop, είναι ικανό να επιλύσει μερικά από τα παρακάτω προβλήματα[15]:

i. Μοντελοποίηση κινδύνου (Risk modeling)
ii. Ανάλυση απειλών (Threat analysis)
iii. Ανάλυση φερεγγυότητας πελάτη (Customer churn analysis)
iv. Ανάλυση δεδομένων δικτύου για πρόβλεψη αποτυχίας (Analyzing network data to predict failure)
v. Μηχανή συστάσεων (Recommendation engine)
vi. Στόχευση διαφημίσεων (Ad targeting)
vii. Ανάλυση συναλλαγών (Transaction analysis)

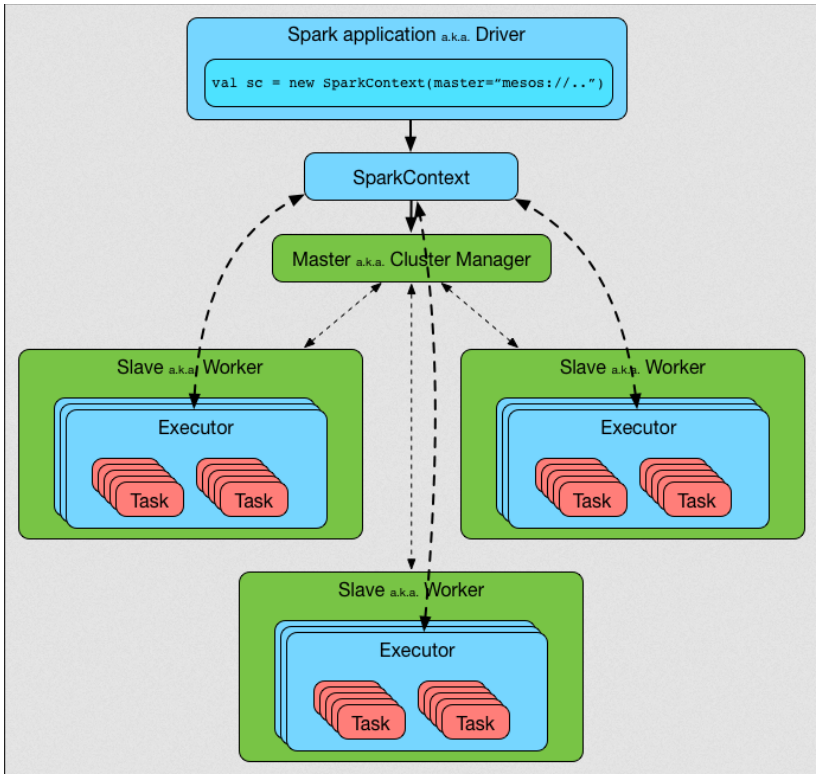
7.3 Apache Spark

Το Apache spark είναι άλλο ένα λογισμικό το οποίο αφορά την υποστήριξη υπολογισμών σε clusters. Το spark είναι ένα ανοιχτό λογισμικό (open source) όπως είναι και το Hadoop και υποστηρίζει τα βασικά εργαλεία με τα οποία δουλεύει ένας αναλυτής δεδομένων με προσανατολισμό στα μεγάλα δεδομένα. Το Apache spark υποστηρίζει πολλές γλώσσες προγραμματισμού όπως Python, R, Scala, και Java. Επίσης, περιλαμβάνει διάφορες εργασίες που κυμαίνονται από την SQL μέχρι και την μηχανική μάθηση και εκτελείται

[15]GilbertE., (2016)

οπουδήποτε από έναν απλό φορητό υπολογιστή μέχρι ένα σύνολο υπολογιστών για την ανάλυση δεδομένων[16].

Η ανάπτυξη του λογισμικού ξεκίνησε το 2009 ως ερευνητικό από την ομάδα εργασίας του Hadoop MapReduce. Στη συνέχεια, καθώς αναπτυσσόταν το λογισμικό, παρατηρήθηκε ότι το MapReduce ήταν αναποτελεσματικό επειδή χρειαζόταν πολλές επαναλήψεις. Για τον λόγο αυτό, οι προγραμματιστές του Apache spark θέλησαν να σχεδιάσουν το συγκεκριμένο πρόγραμμα ώστε να έχει γρήγορη απόκριση στα επαναληπτικά ερωτήματα των αλγορίθμων. Βέβαια, η συγκεκριμένη ιδέα βασίστηκε στην αποθήκευση των ερωτημάτων στην μνήμη RAM και όχι στον σκληρό δίσκο[17]. Το Spark αποτελεί μια προέκταση του μοντέλου MapReduce, σχεδιασμένη έτσι ώστε πέρα από batch processing να υποστηρίζει και διαδραστικά ερωτήματα (interactive queries), επεξεργασία ροών δεδομένων (streaming queries) και machine learning. Η αρχιτεκτονική του Spark φαίνεται στο παρακάτω σχήμα.



Σχήμα 8.Αρχιτεκτονική του Spark

[16]Billchambers&MateiZaharia, 2018
[17]KarauH.etal, 2015

Δομική μονάδα του Spark είναι το Resilient Distributed Dataset (RDD). Το RDD είναι μια συλλογή αντικειμένων δεδομένων μόνο προς ανάγνωση, κατανεμημένο σε ένα cluster υπολογιστών με τη δυνατότητα ανάκτησης σε περίπτωση απώλειας κάποιου τμήματος, ώστε να εξασφαλίζεται η ανοχή σε σφάλματα. Είναι σχεδιασμένο ώστε να τρέχει σε Hadoop cluster και έχει τη δυνατότητα να χρησιμοποιεί δεδομένα από HDFS, HBase, Casssandra, Hive και οποιοδήποτε άλλο Hadoop InputFormat. Το θεμέλιο του Spark είναι το Spark Core. Αποτελεί την γενική μηχανή εκτέλεσης για την πλατφόρμα του Spark. Παρέχει υπολογισμούς In-Memory, βασικές λειτουργικότητες I/O και κατανεμημένο διαμοιρασμό και χρονοδρομολόγηση των tasks. Πάνω στη βάση του Spark ore, έχουν χτιστεί βιβλιοθήκες εκτέλεσης SQL και DataFrames ερωτημάτων (Spark SQL), πεξεργασίας ροών δεδομένων (Spark Streaming), γράφων (GraphX) και machine learning (MLlind).

7.3.1 Εργαλείο του συστήματος

Το module του Spark είναι το Spark SQL. Αποτελεί ένα module για εργασία σε δομημένα δεδομένα. Η λειτουργία του, είναι η ανάκτηση δεδομένων μέσω SQL και η υποστήριξη πολλών πηγών δεδομένων όπως οι πίνακες Hive καθώς και αρχεία JSON. Επιτρέπει σε ερωτήματα Hive να τρέχουν έως και 100 φορές πιο γρήγορα[18].

7.3.2 Βιβλιοθήκη του Spark

Το apache spark παρέχει μια βιβλιοθήκη που ονομάζεται MLlib και η χρήση της αφορά την μηχανική μάθηση. Η συγκεκριμένη βιβλιοθήκη παρέχει πολλούς τύπους αλγορίθμων μηχανικής μάθησης. Πιο συγκεκριμένα, οι αλγόριθμοί της κάνουν classification, collaboration filtering,clustering καθώς και regression ενώ έχουν δυνατότητα αξιολόγησης του μοντέλου διερεύνησης[19].

[18]KarauH.etal,2015
[19]KarauH.etal,2015

7.3.3 Πού χρησιμοποιείται το Apache spark

Το apache spark λόγω του ευρύ φάσματος διερεύνησης δεδομένων, έχεις κυρίως τις εξής χρήσεις:

- Μηχανική μάθηση (machine learning):Όπως αναφερθήκαμε και πριν λίγο στην βιβλιοθήκη MLlib που συνεργάζεται με το apache spark, γίνεται αντιληπτό πως παρέχει ένα ολοκληρωμένο πλαίσιο εκτέλεσης αλγορίθμων μηχανικής μάθησης.
- Δεδομένα ροής (streaming data): Η κύρια χρήση του apache spark και η βασική του ικανότητα είναι η επεξεργασία δεδομένων ροής. Το συγκεκριμένο λογισμικό είναι σε θέση να επεξεργάζεται καθημερινά μεγάλης ροής δεδομένα.
- Διαδραστική ανάλυση (interactive analysis): Toapache spark είναι εξίσου καλό στην διαδραστική ανάλυση δεδομένων. Συγκριτικά με τις υπηρεσίες των προγραμμάτων του Hadoop (Hive&pig) που είναι πιο αργές, το sparkείναι σε θέση να εκτελεί πιο γρήγορα τις διεργασίες μαζί με γλώσσες προγραμματισμού όπως η Pythonή η R[20].

7.4 Apache kafka

Το Apache kafka είναι μια διανομή ανοιχτού λογισμικού και αποτελεί ένα κατανεμημένο σύστημα ανταλλαγής μηνυμάτων (message brooker) για μεταφορά Log αρχείων μεταφέροντάς τα σε πραγματικό χρόνο με ελάχιστη καθυστέρηση.

Οι θεμελιώδεις οντότητες του Kafka είναι οι ροές εγγραφών, οι οποίες αποθηκεύονται ανά κατηγορίες γνωστές ως θέματα – topics. Κάθε εγγραφή αποτελείται από ένα κλειδί, μια τιμή και ένα timestamp. Ανάλογα με το κλειδί της, η εγγραφή μπορεί να κατανεμηθεί περαιτέρω σε ξεχωριστό partition του topic, το οποίο κατ επέκτασιν μπορεί να σημαίνει ξεχωριστό κόμβο Kafka. Με τον server που τρέχει Kafka επικοινωνούν μέσω 4 APIs οι εξής:

- Παραγωγοί: δημοσιεύουν δεδομένα στις ροές εγγραφών ενός ή περισσότερων Kafka topics.
- Καταναλωτές: εγγράφονται σε Kafka topics και στη συνέχεια λαμβάνουν και επεξεργάζονται όπως επιθυμούν τα μηνύματα των ροών της επιλογής τους.

[20]AmsterA., (2016)

- Stream Processors: εφαρμογές επεξεργασίας ροών. Δέχονται ροές δεδομένων ως είσοδο και παράγουν ροές δεδομένων ως έξοδο. Stream Processor είναι και το Spark Streaming.
- Connectors: δημιουργούν επαναχρησιμοποιήσιμους παραγωγούς και καταναλωτές που συνδέουν Kafka topics σε υπάρχουσες εφαρμογές ή συστήματα.

Κατά την εγγραφή πολλαπλών μηνυμάτων σε Kafka server από έναν παραγωγό, έχουμε την εγγύηση πως η σειρά των μηνυμάτων του ίδιου παραγωγού παραμένει σταθερή στην ροή εγγραφών που δημιουργείται και συνεπώς θα αναγνωστεί με τη σωστή σειρά από τους καταναλωτές.

Η ανάγνωση ενός Kafka topic γίνεται με χρήση των offsets. Ο καταναλωτής αρκεί να γνωρίζει μόνο το offset της τελευταίας του ανάγνωσης και το Kafka του εγγυάται πως δε θα διπλοδιαβάσει ούτε θα χάσει κάποιο μήνυμα και πως θα διαβάσει με τη σωστή σειρά όλα τα μηνύματα.

Τα χαρακτηριστικά του Apache kafka είναι τα εξής:

1. Μόνιμη ανταλλαγή μηνυμάτων (Persistent messaging): Για την άντληση πληροφορίας από μεγάλα δεδομένα κάθε είδους, δεν είναι δυνατή η παροχή απώλειας πληροφορίας. Το Apache kafka σχεδιάστηκε με πολυπλοκότητα δομής δίσκου O1 αποδεικνύοντας ότι μπορεί να διαχειριστεί συνεχή ροή δεδομένων που είναι δομημένα κατά σειρά TB.

2. Υψηλή απόδοση (High throughput): Αφού αναφερόμαστε σε μεγάλα δεδομένα, το Apache kafka έχει σχεδιαστεί για την υποστήριξη μηνυμάτων ανά δευτερόλεπτο.

3. Κατανομή (Distributed): Το kafka υποστηρίζει διαχωρισμό μηνυμάτων μέσω των εξυπηρετητών του. Υποστηρίζει επίσης, την κατανομή εργασίας στα clusters.

4. Υποστήριξη πολλών υπολογιστών (Multiple client support): Υποστηρίζει την εύκολη ενσωμάτωση από διαφορετικές πλατφόρμες όπως Java, .NET, PHP, Ruby, Python.

5. Πραγματικός χρόνος (Real time): Τα μηνύματα που παράγονται από τα νήματα πρέπει να είναι ορατά και στον καταναλωτή. Το χαρακτηριστικό αυτό είναι σημαντικό για συστήματα επεξεργασίας σύνθετων συμβάντων (CEP)[21].

[21] GargN., 2013

7.5 MapReduce

Το MapReduce είναι ένα προγραμματιστικό μοντέλο που αναπτύχθηκε από την Google ως σύστημα για τη δημιουργία ευρετηρίων αναζήτησης σε τεράστιο όγκο δεδομένων (πολλά TBs). Το σύστημα αυτό το εμπνεύστηκαν από παλιότερες ιδέες του συναρτησιακού προγραμματισμού καθώς και των περιοχών κατανεμημένου υπολογισμού και βάσεων δεδομένων. Το MapReduce υποστηρίζει την παράλληλη εκτέλεση εφαρμογών σε επεξεργαστές που δεν έχουν ιδιαίτερα χαρακτηριστικά και ισχύ και παρουσιάζει σημαντικότερα οφέλη για εφαρμογές που είναι απαιτητικές σε χρόνο υπολογισμού και μνήμη.

Το προγραμματιστικό μοντέλο MapReduce αναλαμβάνει τη διαμέριση των δεδομένων στο cluster, τη διαχείριση των παράλληλων εργασιών, την εξισορρόπηση των δεδομένων στους επεξεργαστές, την αντιμετώπιση αποτυχιών, καθώς και την επικοινωνία μεταξύ των μηχανημάτων. Ο χρήστης χρειάζεται να γράψει δύο συναρτήσεις, τη map και τη reduce, και το framework φροντίζει ώστε να γίνουν παράλληλα οι υπολογισμοί στα μηχανήματα που αποτελούν το cluster. Με αυτόν τον τρόπο καθίσταται εύκολη η χρήση του προγραμματιστικού μοντέλου ακόμη και χωρίς γνώση παράλληλου προγραμματισμού και κατανεμημένων συστημάτων.

7.6 Προγραμματιστικό μοντέλο

Το βασικό χαρακτηριστικό του MapReduce είναι ότι η λειτουργία του σε κάθε στάδιο βασίζεται σε ζεύγη κλειδιών/τιμών (key/value pairs). Τα δεδομένα εισόδου πρέπει να είναι οργανωμένα σε τέτοια ζεύγη και το framework αναλαμβάνει να εκτελεστεί σε κάθε μηχάνημα η συνάρτηση map που έγραψε ο χρήστης και να παραχθούν ενδιάμεσα ζεύγη κλειδιών/τιμών. Ακολουθεί η εσωτερική ταξινόμηση κατά την οποία συγκεντρώνονται όλες οι τιμές που αντιστοιχούν στο ίδιο κλειδί και τα κλειδιά μαζί με τις αντίστοιχες λίστες τιμών που δημιουργήθηκαν σε αυτή τη φάση περνάνε στη συνάρτηση reduce του χρήστη. Τέλος, η συνάρτηση reduce επεξεργάζεται τα ενδιάμεσα ζεύγη κλειδιών/λίστας τιμών και συνήθως προκύπτουν μικρότερα σύνολα τιμών.

Η συνάρτηση map σπάει τις τιμές, δηλαδή τα κομμάτια κειμένου, σε λέξεις και για κάθε λέξη βγάζει ένα ενδιάμεσο ζεύγος που αντιστοιχεί στην εύρεση της λέξης στο κείμενο μία

φορά. Στη φάση ταξινόμησης, σε κάθε κλειδί-λέξη αντιστοιχίζεται μία λίστα από άσσους με μέγεθος ίσο με το πλήθος των ζευγών που έχουν ως κλειδί τη λέξη. Τέλος, η συνάρτηση reduce διασχίζει τα στοιχεία της λίστας και τα αθροίζει. Για κάθε κλειδί προκύπτει ένα ζεύγος στην έξοδο (λέξη, αριθμός εμφανίσεων).Το MapReduce είναι Turing complete κι επομένως όλα τα προβλήματα μπορούν να γραφτούν σε αυτό το προγραμματιστικό μοντέλο. Ωστόσο, δεν παρουσιάζει οφέλη για όλα τα προβλήματα. Τα προβλήματα που ταιριάζουν στη φιλοσοφία του MapReduce είναι αυτά που:

- Επεξεργάζονται ανεξάρτητα μεταξύ τους δεδομένα,
- Τα δεδομένα εισόδου μπορούν εύκολα να εκφραστούν σε ζεύγη κλειδί/τιμή,
- Χειρίζονται πολύ μεγάλο όγκο δεδομένων, δηλαδή πολλά GBs ή ακόμη και TBs,
- Μπορούν να εκφραστούν ως μία ακολουθία συναρτήσεων map και reduce

Για λίγα δεδομένα δεν αξίζει η χρήση αυτού του προγραμματιστικού μοντέλου, γιατί εισάγονται καθυστερήσεις στον καταμερισμό των εργασιών και το μοίρασμα των δεδομένων, οι οποίες είναι συγκρίσιμες με το χρόνο εκτέλεσης του προγράμματος.

7.7 Πλεονεκτήματα του MapReduce

Ανοχή σε αποτυχίες κόμβων (Fault Tolerance)Το MapReduce μοντέλο χρησιμοποιεί αρχεία για την επικοινωνία μεταξύ των μηχανημάτων του cluster στο διάστημα που μεσολαβεί μεταξύ των φάσεων map και reduce, καθώςκαι για τα ενδιάμεσα αποτελέσματα. Η διαχείριση των αρχείων θα μπορούσε να αποτελεί bottleneck για το MapReduce, γι'αυτό η Google ανέπτυξε το Google File System (GFS),το οποίο παρέχει υψηλό bandwidth αντιγράφοντας και μοιράζοντας τα αρχεία αυτά σε πολλα μηχανήματα (φυσικούς δίσκους). Όταν τα δεδομένα φορτώνονται στους κόμβους, όπως ήδη αναφέρθηκε, χωρίζονται σε κομμάτια τα οποία μάλιστα υπάρχουν σε πολλαπλά αντίγραφα, γεγονός που εξασφαλίζει την αξιοπιστία του συστήματος.
Το MapReduce χειρίζεται αποτελεσματικά τις μερικές αποτυχίες που παρουσιάζονται στους κατανεμημένους υπολογισμούς μετά της κλίμακας. Ο master εντοπίζει map ή reduce εργασίες που έχουν αποτύχει και κανονίζει ώστε να γίνει ανάθεσή τους σε μηχανήματα-slaves που δεν παρουσιάζουν πρόβλημα. Αυτό καθίσταται δυνατό επειδή το MapReduce

έχει shared-nothing αρχιτεκτονική, δηλαδή οι εργασίες είναι ανεξάρτητες. Φυσικά αυτό δεν είναι απόλυτα σωστό, διότι οι reduce εργασίες χρησιμοποιούν τις εξόδους των map εργασιών. Η εξάρτηση αυτή έχει ως αποτέλεσμα να απαιτείται περισσότερη προσοχή όταν επανεκκινείται μία εργασία reduce από ό,τι όταν χρειάζεται να επαναληφθεί μία εργασία map, καθώς τα πρέπει να εξασφαλίζεται ότι η reduce εργασία μπορεί να ανακτήσει τις εξόδους των map εργασιών ή διαφορετικά, να μπορεί να παράξει πάλι τις εξόδους πριν γίνει η επανεκκίνηση της reduce εργασίας. Για να εξασφαλίσει το σύστημα την ανάκαμψή του μετά από αποτυχία του master, κρατάει περιοδικά checkpoints. Βέβαια, επειδή υπάρχει μόνο ένας master στην αρχιτεκτονική αυτή, η αποτυχία του δεν είναι πολύ πιθανή

7.7.1 Τοπικότητα δεδομένων

Στην καρδιά του MapReduce είναι η τοπικότητα των δεδομένων. Το προγραμματιστικό αυτό μοντέλο διαθέτει μηχανισμούς που φροντίζουν, όσο αυτό είναι εφικτό, σε κάθε κόμβο του cluster να ανατίθεται η επεξεργασία δεδομένων που είναι τοπικά. Με αυτόν τον τρόπο η πρόσβαση στα δεδομένα είναι πολύ γρήγορη και τελικά επιτυγχάνεται καλή επίδοση. Μάλιστα επειδή το bandwidth του δικτύου παίζει καθοριστικό ρόλο σε ένα cluster, οι υλοποιήσεις του MapReduce καθορίζουν σαφώς την τοποθεσία του δικτύου.

7.7.2 Write Once, Read Many

Μία βασική ιδιότητα του MapReduce είναι το μοντέλο αποθήκευσης δεδομένων WORM (Write Once, Read Many). Αυτό επιτρέπει υψηλό επίπεδο παραλληλισμού στην πρόσβαση στα δεδομένα νωρίς να απαιτούνται πολύπλοκοι μηχανισμοί συγχρονισμού. Αυτό το μοντέλο εφαρμόζεται άμεσα στις βιολογικές εφαρμογές, καθώς ο μεγάλος όγκος δεδομένων που διαχειρίζονται δεν τροποποιείται κατά τη διάρκεια των εργασιών.

7.7.3 Διαμοιρασμός των εργασιών

Όπως αναφέρθηκε παραπάνω, τα αρχεία εισόδου χωρίζονται σε M τμήματα για τη φάση map, ενώ τα ενδιάμεσα αποτελέσματα σε R για τη φάση reduce. Ιδανικά, τα μεγέθη M και R θα πρέπει να είναι πολύ μεγαλύτερα απο τον αριθμό των κόμβων του cluster για δύο λόγους:
Να είναι δυνατή η δυναμική εξισορρόπηση του φόρτου επεξεργασίας (load balancing),
Να επιταχύνεται η ανάκαμψη του συστήματος όταν κάποιος slave αποτύχει, καθώς οι εργασίες map που είχε περατώσει μπορούν να ανατεθούν παράλληλα σε πολλούς κόμβους.

Βέβαια, τα μεγέθη M και R δεν μπορούν να είναι απεριόριστα μεγάλα, γιατί οι αποφάσεις που πρέπει να παίρνει ο master είναι O (M+R) και οι καταστάσεις που κρατάει στη μνήμη για να ελέγχει αν απέτυχε κάποιος κόμβος είναι O (MxR). Συνήθως η τιμή M διαλέγεται έτσι ώστε κάθε εργασία να είναι μεγέθους μεταξύ 16MB και 64MB και η τιμή R μικρό πολλαπλάσιο του αριθμού των κόμβων στο cluster.

7.7.4 Backup εργασιών

Ο χρόνος εκτέλεσης των προγραμμάτων επηρεάζεται από τους πιο αργούς κόμβους. Για να αντιμετωπιστεί αυτό το πρόβλημα, η Google χρησιμοποιεί τον παρακάτω μηχανισμό: όταν η εκτέλεση ενός MapReduce προγράμματος είναι κοντά στο τέλος, ο master αναθέτει την εκτέλεση των εργασιών που δεν έχουν τερματιστεί και σε άλλους κόμβους. Με αυτόν τον τρόπο, μία εργασία τελειώνει όταν είτε η αρχική είτε η εφεδρική εκτέλεση τελειώσει. Η Google παρατήρησε ως και 44% επιτάχυνση όταν ήταν ενεργοποιημένη η επιβολή backup των εργασιών.

8. Γλώσσες προγραμματισμού για την επεξεργασία δεδομένων

8.1 Η γλώσσα Python

Μία από τις πλέον δημοφιλέστερες γλώσσες προγραμματισμού που χρησιμοποιείται για διερευνητική ανάλυση δεδομένων είναι η Python.

Η συγκεκριμένη γλώσσα δημιουργήθηκε από τον προγραμματιστή GuidoVanRossum ως project για την περίοδο των διακοπών των Χριστουγέννων και κυκλοφόρησε πρώτη φορά το 1991 παίρνοντας το όνομά της από την διάσημη βρετανική κωμωδία «Monty Python Flying Circus»[22].

Πρόκειται για μια γλώσσα προγραμματισμού ανοιχτού κώδικα και η χρήση της εκτός από την ανάλυση δεδομένων μπορεί να είναι και για γενικό προγραμματισμό. Για να διαπιστώσει κάποιος την απλότητά της, αξίζει να παρατηρήσει πως με μια απλή εντολή *"import this"* δύναται να εισάγει οτιδήποτε[23].

Με την Python μπορούμε να λύσουμε από τα πιο απλά μέχρι τα πιο σύνθετα προβλήματα. Η μεγάλη ευκολία που προσφέρει είναι η σύνταξή της. Αν κάποιος γνωρίζει βασικές αρχές προγραμματισμού, είναι σε θέση να προγραμματίσει εύκολα σε python καθώς ακόμα και να διαβάσει κώδικά της. Από την άλλη αν κάποιος δεν γνωρίζει την γλώσσα προγραμματισμού Python μπορεί εύκολα να τη μάθει, να γράφει σε αυτήν και να την διαβάζει[24].

Για να ασχοληθεί κάποιος με το κομμάτι της Python, αρκεί να κατεβάσει από το επίσημο sitetης τον IDE (Integrated Development Environment), ή αλλιώς το πλέον πιο εύχρηστο και δημοφιλές Jupiter στο οποίο η απεικόνιση δεδομένων «τρέχει» απευθείας.

Στον κλάδο της επιστήμης των δεδομένων, η Python αποτελεί ένα από τα πιο ισχυρά εργαλεία, ίσως και το ισχυρότερο. Οι βιβλιοθήκες που διαθέτει είναι απλές και εξαιρετικές στην χρήση τους[25].

Μερικές από τις βιβλιοθήκες της Python είναι οι παρακάτω:

[22] Είναι διαθέσιμη στην επίσημη σελίδα της https://www.python.org/downloads/
[23] NicholasH.Tol., 2015, Python, (n,d).
[24] HoojaS., (2017)
[25] WesMcKin., [2012]

1. NumPy: Αφορά βιβλιοθήκη της python και ασχολείται με πράξεις μεταξύ πινάκων.
2. Matplotlib: Η συγκεκριμένη βιβλιοθήκη αφορά την δημιουργία γραφημάτων για την απεικόνιση των δεδομένων.
3. Scikit-learn: Άλλη μια ισχυρή βιβλιοθήκη της Pythonπου αφορά κυρίως το κομμάτι της μηχανικής μάθησης.
4. Pandas: Η λειτουργία της βιβλιοθήκης είναι η επεξεργασία δεδομένων με δημιουργία πινάκων[26].

8.2 Η γλώσσα Scala

Η Scala, είναι μια γλώσσα προγραμματισμού γενικής χρήσης ,η οποία έχει σχεδιαστεί για να εκφράζει κοινά μοτίβα προγραμματισμού με συνοπτικό και ασφαλή τρόπο.

Η γλώσσα προγραμματισμού Scala σε τεχνικό επίπεδο συνδυάζει τον αντικειμενοστραφή και τον συναρτησιακό προγραμματισμό. Αποτελεί μια κλιμακωτή γλώσσα όπου μπορεί να μεγαλώσει ανάλογα με τις ανάγκες των χρηστών της. Για αυτόν τον λόγο χρησιμοποιείται για το στήσιμο μικρών εφαρμογών έως και ολοκληρωμένων συστημάτων. Είναι βασισμένη στην πλατφόρμα της Java, και είναι ιδιαίτερα εύκολα κατανοητή σε όσους προγραμματιστές γνωρίζουν την εν λόγω γλώσσα. Η κλιμακωτή φύση της προσδίδει και το όνομά της, δηλαδή Scala όπου βγαίνει από το scalable language. Η κλιμακωτή φύση της γλώσσας μπορεί να επηρεαστεί από πολλούς τομείς όπως οι λεπτομέρειες στην σύνταξη ή η δημιουργία abstract κατασκευών. Η δυνατότητα που έχει η Scala να συνδυάζει τον αντικειμενοστραφή με τον συναρτησιακό προγραμματισμό στο επίπεδο που το κάνει, είναι το βασικό χαρακτηριστικό που την καθιστά μία από τις πιο κλιμακωτές γλώσσες προγραμματισμού. Η Scala καταφέρνει να εμβαθύνει σε μεγάλο βαθμό, σε σχέση με άλλες γλώσσες, την διαδικασία συνδυαστικής χρήσης των δύο τρόπων προγραμματισμού που προαναφέρθηκαν. Για παράδειγμα, ενώ άλλες γλώσσες αντιμετωπίζουν τα αντικείμενα και τις συναρτήσεις με τελείως διαφορετικό τρόπο, για την Scala κάθε τιμή συνάρτησης αποτελεί ένα αντικείμενο. Τα είδη των συναρτήσεων αποτελούν κλάσεις οι οποίες μπορούν να κληρονομηθούν από

[26]Μια γρήγορη πρώτη επαφή με τις συγκεκριμένες βιβλιοθήκες μπορεί να βρει κάποιος στο sitewww.datacamp.com και να ανατρέξει σε CheatSheet των βιβλιοθηκών.

άλλες κλάσεις. Αυτές οι δυνατότητες έχουν μεγάλες συνέπειες όσον αφορά την επεκτασιμότητα και την μεγάλη κλιμάκωση της γλώσσας.

Ο Martin Odersky, δημιουργός της Scala, μαζί με την ομάδα του, ξεκίνησαν την υλοποίησή της το 2001 στην ομοσπονδιακή πολυτεχνική σχολή της Λωζάννης (Ecole Polytechnique Federale de Lausanne).Η συγκεκριμένη γλώσσα έκανε την επίσημη εμφάνισή της τον Ιανουάριο του 2004 στην πλατφόρμα Java virtual machines και λίγους μήνες μετά στην πλατφόρμα .NET. Το όνομά της προέρχεται από την φράση scalable language που σημαίνει ότι έχει σχεδιαστεί για να «μεγαλώνει» παράλληλα με τις ανάγκες των χρηστών[27].

Η Scala παρέχει γλωσσική δια λειτουργικότητα με την Java, με σκοπό οι βιβλιοθήκες γραμμένες σε οποιαδήποτε από τις δύο γλώσσες να μπορούν να αναφέρονται απευθείας στον κώδικα Scala ή Java. Όπως και με την Java, η Scala είναι αντικειμενοστραφής και χρησιμοποιεί μια σύντομη σύνταξη που θυμίζει τη γλώσσα προγραμματισμού C. Σε αντίθεση με την Java, η Scala έχει πολλά χαρακτηριστικά λειτουργικών γλωσσών προγραμματισμού όπως το Scheme, το Standard ML και το Haskell. Έχει επίσης ένα προηγμένο σύστημα τύπων που υποστηρίζει αλγεβρικούς τύπους δεδομένων, συν διακύμανση και αντιστοίχιση, τύπους υψηλότερης τάξης (όχι όμως υψηλότερης κατηγορίας) και ανώνυμους τύπους.

Η Scala είναι μια γλώσσα αντικειμενοστραφής όπως είπαμε και τα αντικείμενα στα οποία προσανατολίζεται είναι τα εξής:

- Απόκρυψη πληροφοριών.
- Κληρονομικότητα.
- Πολυμορφισμός - δυναμική σύνδεση.
- Όλοι οι προκαθορισμένοι τύποι είναι αντικείμενα
- Όλες οι λειτουργίες εκτελούνται με την αποστολή μηνυμάτων σε αντικείμενα.
- Όλοι οι τύποι που ορίζονται από τον χρήστη είναι αντικείμενα.

[27]NilanjanR., 2013

8.3 Διαφορές Scala με Java

Η Scala όπως προαναφέρθηκε είναι μία μετεξέλιξη της Java και για αυτόν τον λόγο υπάρχουν πολλές ομοιότητες αλλά και διαφορές. Είναι σημαντική λοιπόν η μελέτη των χαρακτηριστικών κάθε γλώσσας και η σύγκριση μεταξύ τους.

Χαρακτηριστικά της Java :

1. Σχεδιάστηκε για την δημιουργία εφαρμογών που βασίζονται στον αντικειμενοστραφή προγραμματισμό.
2. Μπορεί να χειριστεί πολλά νήματα καθώς επίσης παρέχει λειτουργίες αυτόματης διαχείρισης μνήμης.
3. Παρέχει υψηλή ασφάλεια και απόδοση. Διευκολύνει την δημιουργία κατανεμημένων συστημάτων καθώς είναι δικτυοκεντρική.
4. Επιτρέπει την μεταφορά και εκτέλεσή της σε άλλα συστήματα χωρίς την δημιουργία προβλημάτων.

Χαρακτηριστικά της Scala :

1. Υποστηρίζει τον αντικειμενοστραφή προγραμματισμό, συνδυάζοντας όμως και τον συναρτησιακό προγραμματισμό.
2. Είναι συνοπτική όσον αφορά την σύνταξή της και προσαρμόζεται ανάλογα με τις ανάγκες των χρηστών.
3. Επιτρέπει την εκτέλεση κώδικα γραμμένο σε Java.
4. Έχει στατικό σύστημα τύπων.

Αναλύοντας τα κύρια χαρακτηριστικά κάθε γλώσσας προκύπτουν άμεσα κάποιες βασικές διαφορές.

Τα πλεονεκτήματα που έχει η Java έναντι της Scala αφορούν την πολύχρονη ανάπτυξη της. Διαθέτει εγχειρίδια σχεδιασμένα με μεγάλη προσοχή στην λεπτομέρεια. Αυτό έχει ελκύσει μεγάλο αριθμό προγραμματιστών οι οποίοι την χρησιμοποιούν για μεγάλο εύρος εφαρμογών, ενισχύοντας με αυτόν τον τρόπο την δημοτικότητα της γλώσσας. Λόγω αυτής της δημοτικότητας έχουν αναπτυχθεί πολλές βιβλιοθήκες όλων των ειδών έτσι ώστε να καλυφθούν όλες οι ανάγκες των χρηστών. Επίσης η επίδοσή της έχει αποδειχθεί ότι είναι άριστη σε ένα μεγάλο φάσμα διαφορετικών λειτουργιών.

Η Scala αντίθετα από την Java, διαθέτει μικρότερη κοινότητα χρηστών. Αυτό έχει ως αποτέλεσμα την σχετικά αργή ανάπτυξή της και την μικρή της παρουσία στην κοινότητα των προγραμματιστών. Παρόλα αυτά διαθέτει πολλά χαρακτηριστικά τα οποία την κάνουν να ξεχωρίζει και να προτιμάτε από την Java. Ένα από αυτά είναι η δυνατότητα της να υποστηρίζει και αντικειμενοστραφείς και συναρτησιακές εφαρμογές. Αυτό το χαρακτηριστικό σε συνδυασμό με το σύντομο και περιεκτικό της συντακτικό, επιτρέπει την εύκολη εκμάθηση της γλώσσας από προγραμματιστές που προέρχονται από διαφορετικές γλώσσες προγραμματισμού. Η δυναμικότητα της γλώσσας εξασφαλίζει την προσαρμογή της ανάλογα με την ανάγκη του χρήστη για κάθε πιθανή εφαρμογή. Επιπλέον, ο σχεδιασμός της παρέχει τα εργαλεία για δημιουργία εφαρμογών που βασίζονται στην παραλληλοποίηση και στην ισοκατανομή των δεδομένων, ένα χαρακτηριστικό που την καθιστά εξαιρετικά καλή επιλογή για εφαρμογές διαχείρισης μεγάλου όγκου δεδομένων.

Συνολικά η Scala υστερεί σε περιεχόμενο σε σχέση με την Java, παρόλα αυτά λόγω των χαρακτηριστικών που αναφέρθηκαν παραπάνω, έχει αποκτήσει πολλούς χρήστες ειδικά τα τελευταία χρόνια με τον αριθμό τους συνεχώς να αυξάνεται. Οι μεγάλες εταιρείες λαμβάνουν περισσότερο υπόψιν τις δυνατότητές της και για αυτόν τον λόγο έχει γίνει μία από τις πιο επιθυμητές γλώσσες τις δεκαετίας.

8.4 Scala και μεγάλα δεδομένα

Ένα από τα πιο ισχυρότερα χαρακτηριστικά της Scala που την έχουν κάνει τόσο δημοφιλή, είναι η στενή της σχέση με την cluster-computing πλατφόρμα Spark. Όπως και το Hadoop, το Spark είναι ένα πλαίσιο που χρησιμοποιείται για την επεξεργασία πολύ μεγάλων συνόλων δεδομένων σε παράλληλα και κατανεμημένα σε ένα cluster υπολογιστικών συστημάτων. Ενώ το Hadoop βασίζεται στην τεχνική του MapReduce, το Spark χρησιμοποιεί μία διαφορετική διαδικασία που αντιγράφει τις περισσότερες λειτουργίες στην μνήμη RAM του συστήματος. Με το χειρισμό αυτών των λειτουργιών "στη μνήμη", το Spark μπορεί να επιτύχει πολύ υψηλότερες ταχύτητες από το MapReduce. Αυτό το κέρδος ταχύτητας καθιστά το Spark μοναδικά κατάλληλο για επεξεργασία και ανάλυση δεδομένων.

Το προγραμματιστικό περιβάλλον Spark είναι γραμμένο σε Scala. Παρά το γεγονός ότι υποστηρίζει βιβλιοθήκες γραμμένες σε Java, Python και R, υπάρχουν σαφή πλεονεκτήματα στην χρήση της γλώσσας δημιουργίας του, καθώς επιτρέπει την χρήση νέων λειτουργιών που δεν έχουν μεταφερθεί ακόμα σε άλλες γλώσσες προγραμματισμού. Επιπλέον, η μετάφραση μεταξύ διαφορετικών γλωσσών και περιβαλλόντων μπορεί να οδηγήσει σε σφάλματα και επιβραδύνσεις, γεγονός που δίνει στη Scala ένα πλεονέκτημα σε σχέση με την Python ή την Java. Συνεπώς, επειδή η Scala είναι άμεσα συνδεδεμένη με την πλατφόρμα Spark, η οποία με την σειρά της είναι άμεσα συνδεδεμένη με την διαχείριση Big Data εφαρμογών, καθιστά την Scala πολύ δημοφιλή επιλογή για εφαρμογές τέτοιους είδους.

Σε τεχνικό επίπεδο η Scala διαθέτει μία φιλοσοφία σκέψεις διαφορετική από άλλες γλώσσες προγραμματισμού. Τα χαρακτηριστικά της σε συνδυασμό με την κλιμακωτή φύση της επιτρέπουν στον προγραμματιστή να σκεφτεί με εντελώς νέους τρόπους και να προσεγγίσει πρωτοφανή προβλήματα Big Data με νέες μεθόδους και τεχνικές. Μία σύντομη αλλά εξαιρετικά περιεκτική περιγραφή της φιλοσοφίας της Scala είναι η εξής : Στα προγράμματα Scala στέλνεται ο κώδικας στα δεδομένα και όχι τα δεδομένα στον κώδικα. Η συγκεκριμένη περιγραφή αποτέλεσε εξαιρετική συμβουλή προς εμένα για την υλοποίηση της παρούσας διπλωματικής.

8.5 Η γλώσσα Java

Η γλώσσα Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού, η οποία έχει σχεδιαστεί από την εταιρεία Sun Microsystems. Βασικό πλεονέκτημά της έναντι των υπολοίπων, είναι η ανεξαρτησία του λειτουργικού της συστήματος[28].

8.6 Η γλώσσα R

Η γλώσσα R είναι μια ελεύθερη στατιστική γλώσσα προγραμματισμού, η οποία είναι ιδιαίτερα διαδεδομένη στον κλάδο της επιστήμης των δεδομένων. Η διάθεση του λογισμικού γίνεται από το επίσημο sitetης http://www.r-project.org/ και υποστηρίζει και αυτή (όπως και

[28]Deitel, Paul&Harvey, 2015.

η Python στην οποία αναφερθήκαμε ανωτέρω) πακέτα για την δημιουργία στατιστικών αναλύσεων.

9. Θεωρία χαρτοφυλακίου

Η θεωρία χαρτοφυλακίου, αποτελεί ένα πολύ μεγάλο τμήμα του κλάδου των οικονομικών το οποίο αναφέρεται στον συνδυασμό των περιουσιακών στοιχείων που επενδύει και κατέχει ένας επενδυτής.

Η θεωρία χαρτοφυλακίου οφείλεται στον γνωστό οικονομολόγο και νομπελίστα Harry Markowitz. Ο Markowitz εξετάζει τον τρόπο με τον οποίο πρέπει να συμπεριφέρεται ένας λογικός επενδυτής όταν θέλει να συνθέσει ένα χαρτοφυλάκιο με χρηματοοικονομικούς τίτλους[29].

Αυτό που επεσήμανε ο Markowitz με την θεωρία του είναι πως στόχος κάθε επενδυτή είναι η μεγιστοποίηση της απόδοσης και η ελαχιστοποίηση του κινδύνου. Ο Markowitz, στη θεωρία χαρτοφυλακίου (portfolio theory), έκανε αρκετές υποθέσεις, δύο εκ των οποίων είναι:

1) Οι επενδυτές πρέπει να εξετάζουν την κάθε επένδυση υποθέτοντας ότι αντιπροσωπεύεται από μια κανονική κατανομή πιθανοτήτων των αναμενομένων αποδόσεών της, που θα πραγματοποιηθούν μέσα σε μια περίοδο διακράτησης.

2) Για ορισμένη ποσότητα κινδύνου οι επενδυτές προτιμούν περισσότερη αναμενόμενη απόδοση από λιγότερη[30].

Η αναμενόμενη απόδοση ενός αξιόγραφου ισοδυναμεί με τον αποδεχόμενο κίνδυνο.

Ο κίνδυνος είναι αυτός ο οποίος εκφράζει την αβεβαιότητα πως η απόδοση που πραγματοποιείται δεν είναι ίση με την απόδοση που αναμένουμε.

Τα χαρακτηριστικά του κινδύνου είναι ο χρόνος και η μεταβλητότητα.

Όσο μεγαλύτερο είναι το κεφάλαιο επένδυσης τόσο μεγαλύτερος είναι και ο κίνδυνος το κεφάλαιο να υποστεί ζημία. Αυτό συνεπάγεται πως ποτέ μια επένδυση δεν έχει σταθερή απόδοση, είναι πάντα επικίνδυνη.

[29] Στέφανος Παπαδάμου, Διαχείριση χαρτοφυλακίου μια σύγχρονη προσέγγιση
[30] Βασιλείου, 2008, σελ.145-146

9.1 Αναμενόμενη απόδοση και κίνδυνος χαρτοφυλακίου

Αναμενόμενη απόδοση ενός αξιόγραφου, είναι ο σταθμικός μέσος όρος όλων των δυνητικών αποδόσεων του αξιόγραφου όπου κάθε δυνητική απόδοση σταθμίζεται από την πιθανότητα να συμβεί.

Ο τύπος της είναι ο εξής:

$$E(r) = \sum_{i=1}^{n} P_i r_i$$

Σχήμα 9.Αναμενόμενη απόδοση χαρτοφυλακίου

ΌπουE(r), είναι η αναμενόμενη απόδοση του αξιογράφου, P_iείναι η πιθανότητα που υπάρχει να συμβεί η I δυνητική απόδοση του αξιογράφου και $\sum Pi = 1, r_i$είναι η i δυνητική απόδοση του αξιογράφου και n= ο αριθμός των δυνητικών αποδόσεων.

Η διακύμανση των αναμενόμενων αποδόσεων ενός αξιογράφου είναι η προσδοκώμενη τιμή του τετραγώνου των αποκλίσεων των αποδόσεων του χαρτοφυλακίου από το μέσο όρο της απόδοσης του και δίνεται από την παρακάτω σχέση:

$$Var = \sigma^2$$

Σχήμα 10.Τύπος διακύμανσης

Τον κίνδυνο ενός χαρτοφυλακίου μπορούμε να τον μετρήσουμε με την διακύμανση (για την οποία έγινε λόγος παραπάνω) ή την τυπική απόκλιση των αποδόσεων που έχει το χαρτοφυλάκιο.

Η τυπική απόκλιση ενός χαρτοφυλακίου δίνεται από τον παρακάτω τύπο:

$$\sigma_p = \sqrt{\sum_{i=1}^{n} w_i^2 \sigma_i^2 + \sum_{i=1}^{n} \sum_{i=1}^{n} w_i j_i COV_{ij}}$$

Σχήμα 11.Τύπικη απόκλιση

Η σχέση ανάμεσα στις αποδόσεις των χρεογράφων μπορεί να είναι είτε θετική είτε αρνητική.

Επομένως συνεπάγεται ότι οι τιμές τις οποίες παίρνει η συνδιακύμανση είναι μεταξύ του συν (+) και πλην (-) άπειρο.

Για να αποφευχθεί η δυσκολία στους υπολογισμούς, παίρνουμε έναν δείκτη ο οποίος είναι γνωστός ως συντελεστής συσχέτισης (correlation coefficient) των αποδόσεων των αξιογράφων, ο οποίος διατηρεί τις ιδιότητες της συνδιακύμανσης και μετρά εντός ενός ορίου ± 1 το βαθμό μεταβολής αποδόσεων των μετοχών.

Ο τύπος είναι:

$$COV_{ij} = p_{ij}\sigma_i\sigma_j$$

Σχήμα 12.Τύπος συνδιακύμανσης

9.2 Το υπόδειγμα ενός δείκτη

Το υπόδειγμα του ενός δείκτη (single index model) πήρε το όνομά του από τον οικονομολόγο William Sharpe. Παρουσιάζει μεγάλο ενδιαφέρον, διότι μειώνει σημαντικά τις εκτιμήσεις για τον υπολογισμό του αποτελεσματικού συνόρου. Αναφορικά με το συγκεκριμένο υπόδειγμα, πρέπει όλα τα αξιόγραφα να σχετίζονται μεταξύ τους διότι πρέπει να υπάρχει μια κοινή αντίδραση στις μεταβολές της αγοράς.

Έτσι, πρέπει να υπάρχει μια κοινή απόδοση ενός κοινού δείκτη ο οποίος θα είναι σε θέση να μας δείξει τις αντιδράσεις - μεταβολές της αγοράς. Ως υπόδειγμα συνήθως χρησιμοποιείται ένας χρηματιστηριακός δείκτης (π.χ. γενικός δείκτης τιμών του χρηματιστηρίου του Πεκίνο).

Η μορφή του δείκτη έχει την παραπάνω μορφή:

$$R_I = a_i + b_i * R_m + \varepsilon_i$$

Σχήμα 13.Μορφή ενός δείκτη

Όπου R_I = Η απόδοση του αξιόγραφου

R_m = η απόδοση του χρηματιστηριακού δείκτη της αγοράς και

a_i= η απόδοση ενός αξιόγραφου

b_i = συντελεστής μέτρησης της ευαισθησίας της απόδοσης του αξιογράφου συ τυχών

μεταβολές της απόδοσης του χρηματιστηριακού δείκτη.

ε_i = τυχαίο σφάλμα (πραγματική απόδοση - αναμενόμενη απόδοση)

Δεδομένων των ανωτέρω αναφερθέντων η απόδοση του δείκτη βασίζεται στις παρακάτω

υποθέσεις:

- Οι μεταβλητές R_m είναι τυχαίες.
- Η αναμενόμενη αξία του είναι ίση με το 0. Δηλαδή $\varepsilon_i = 0$
- Η συνδιακύμανση των Cov (R_m και $\varepsilon_i = 0$)

Το ε_i αξιόγραφο είναι ανεξάρτητο από οποιοδήποτε άλλο.Αυτό σημαίνει ότι τααξιόγραφα μεταβάλλονται από κοινού λόγω της κοινής αντίδρασης της οποίας υπάρχει στην αγορά.

Άρα, δεν υπάρχουν παράγοντες που να επηρεάζουν τις αποδόσεις των αξιογράφων, παρά μόνο η απόδοση της συνολικής αγοράς.

Στον δείκτη Sharpe χρησιμοποιούμε γραμμική παλινδρόμηση, της οποίας το αποτέλεσμα απεικονίζεται με μια απλή γραμμή η οποία περιγράφει την σχέση μεταξύ των μεταβολών αποδόσεων του αξιόγραφου και των μεταβολών στις αποδόσεις ενός χρηματιστηριακού δείκτη της αγοράς. Η κλίση της γραμμής αυτής ονομάζεται συντελεστής β και είναι η γωνία η οποία σχηματίζεται στο διάγραμμά μας.

Ο συντελεστής β υπολογίζεται και με τον παρακάτω τύπο:

$$\beta_i = \frac{\sigma_{im}}{\sigma_m^2}$$

Σχήμα 14.Συντελεστής β

Από τα παραπάνω προκύπτει ότι αξιόγραφα με συντελεστή β μεγαλύτερο της μονάδας θεωρούνται επιθετικά και αυτό το καταλαβαίνουμε επειδή αν μεταβάλουμε τον δείκτη κατά 1% θα επιφέρει μεγαλύτερες μεταβολές στις αποδόσεις αξιόγραφων.

Σε αντίθετη περίπτωση αξιόγραφα με συντελεστή β μικρότερα της μονάδας θεωρούνται αμυντικά διότι οι αποδόσεις τους θα έχουν μεγαλύτερη ευαισθησία στις μεταβολές των αποδόσεων του δείκτη της αγοράς.

9.3 Συστημικός και μη συστημικός κίνδυνος

Όπως είναι γνωστό, σε κάθε χαρτοφυλάκιο υπάρχει κίνδυνος άλλοτε σε μικρότερο βαθμό άλλοτε σε μεγαλύτερο βαθμό. Γι' αυτό, όπως επισημάνθηκε ήδη, στόχος κάθε επενδυτή είναι η μεγιστοποίηση της απόδοσης και η ελαχιστοποίηση του κινδύνου.

Ο κίνδυνος ενός αξιογράφου μπορεί να μετρηθεί για την αποφυγή μεγάλου ρίσκου και την σωστή επένδυση των ρευστών διαθέσιμων των πελατών. Ο τύπος μέτρησης είναι ο ακόλουθος:

$$\sigma_i^2 = \beta_\iota \sigma_m^2 + \sigma_{ei}^2$$

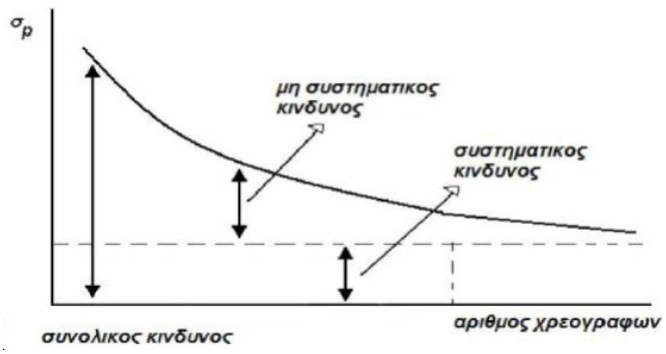
Σχήμα 15.Κίνδυνος αξιογράφου

Στον παραπάνω τύπο, το δεύτερο σκέλος του (σ_{ei}^2)αποτελεί μη συστηματικό κίνδυνο - διαφοροποιήσιμο κίνδυνο και αυτό διότι όταν υπάρχουν πολλά αξιόγραφα στο χαρτοφυλάκιό μας, τότε ο κίνδυνος εξαλείφεται λόγω του ότι προσεγγίζει το μηδέν.

Σε αντίθεση με το δεύτερο σκέλος, στο πρώτο σκέλος $(\beta_\iota \sigma_m^2)$ στην εξίσωση αυτή ναι μεν τοσ_mαποτελεί το ίδιο για όλα τα χαρτοφυλάκια και το πρώτο σκέλοςβ_i αποτελεί το μέτρο του συστηματικού - μη διαφοροποιήσιμου κινδύνου.

Με τον όρο συστηματικό κίνδυνο, ονομάζουμε την μεταβλητότητα των αποδόσεων όλων των περιουσιακών στοιχείων τα οποία είναι εκτεθειμένα στον κίνδυνο για διάφορους λόγους (κυρίως μακροοικονομικούς), όπως η αύξηση της προσφοράς χρήματος. Στον παραπάνω τύπο όμως, ο κίνδυνος μπορεί να εξαλειφθεί με την διακράτηση ενός χαρτοφυλακίου πολλών αξιογράφων.

Μαζί με την λέξη συστηματικός - μη συστηματικός κίνδυνος γίνεται αναφορά της λέξης διαφοροποίηση. Διαφοροποίηση ενός χαρτοφυλακίου ονομάζουμε την επενδυτική στρατηγική στην οποία συγκεντρώνουμε μια ποικιλία χρεογράφων στο χαρτοφυλάκιό μας, των οποίων τόσο οι αποδόσεις όσο και οι συσχετίσεις των αποδόσεων είναι διαφορετικές καθώς επίσης υπάρχουν και διαφορετικά επίπεδα κινδύνου με έναν στόχο φυσικά, την μείωση του συνολικού κινδύνου χωρίς την μείωση της απόδοσης.



Σχήμα 16.Συστηματικός και μη συστηματικός κίνδυνος

Εν συντομία τον συστηματικό και μη συστηματικό κίνδυνο μπορούμε να τον απεικονίσουμε με τον παρακάτω τρόπο:

$$\sigma_i^2 = \text{Συστημικός κίνδυνος} + \text{Μη συστημικός κίνδυνος}$$

9.4 Η θεωρία της κεφαλαιαγοράς

Η θεωρία της κεφαλαιαγοράς βασίζεται στην θεωρία του Markowitz. Επομένως και οι υποθέσεις της εν λόγω θεωρίας είναι κοινές στην βάση τους με την θεωρία του Markowitz με ελάχιστες διαφοροποιήσεις.

1. Όλοι οι επενδυτές λαμβάνουν τις αποφάσεις τους έχοντας ως γνώμονα την θεωρία του Markowitz με αποτέλεσμα να διακρατούν χαρτοφυλάκια που βρίσκονται πάνω στο αποτελεσματικό σύνορο.
2. Οι προσδοκίες όλων των επενδυτών είναι κοινές σχετικά με τις αναμενόμενες αποδόσεις, τυπικές αποκλείσεις καθώς και την συνδιακύμανση των αξιογράφων.
3. Ο επενδυτικός ορίζοντας για όλους τους επενδυτές είναι ίσος με μια περίοδο (μήνα, εξάμηνο, χρόνος κ.α.)
4. Στο χαρτοφυλάκιο υπάρχει ένα στοιχείο χωρίς κίνδυνο στο οποίο μπορούν να επενδύσουν απεριόριστα ποσά και φυσικά να εισπράξουν απόδοση χωρίς κίνδυνο.
5. Δεν υπάρχουν φόροι και κόστη στις συναλλαγές πώλησης και αγοράς περιουσιακών στοιχείων.

6. Οι επενδύσεις είναι όπως και στην πραγματικότητα διαιρετές και διαπραγματεύσιμες.
7. Δεν υπάρχει πληθωρισμός και ο ρυθμός μεταβολής των επιτοκίων έχει προβλεφθεί.
8. Έχουμε πλήρη ανταγωνισμό στην αγορά κεφαλαίου (δεν επηρεάζονται οι τιμές ενός περιουσιακού στοιχείου με υπερβολική αγορά ή πώληση του στοιχείου) και βρίσκεται σε ισορροπία.

9.5 Η γραμμή κεφαλαιαγοράς

Αν υποθέσουμε ότι υπάρχει στο χαρτοφυλάκιό μας ένα περιουσιακό στοιχείο χωρίς κίνδυνο (risk-free asset), του οποίου η απόδοση είναι ίση με μηδέν, τότε από την πλευρά του ο επενδυτής έχει την ευκολία να συνδυάσει μια επένδυση του στοιχείου χωρίς κίνδυνο με μια οποιαδήποτε επένδυση στο χαρτοφυλάκιο.

Για παράδειγμα αν επενδύσει ένα ποσοστό της συνολικής αξίας του χαρτοφυλακίου(w_{rf})στο στοιχείο χωρίς κίνδυνο και το υπόλοιπο ποσοστό που είναι $(1 - w_{rf})$σε ένα υποτιθέμενο χαρτοφυλάκιο Ψ τότε η αναμενόμενη απόδοση του θα είναι ίση με:

$$E(R_P) = [((w_{rf}) * E(R_f) + (1 - w_{rf}) * E(R_\psi)$$

Σχήμα 17.Αναμενόμενη απόδοσης

Από την παραπάνω σχέση παρατηρούμε πως ισχύει διότι η τυπική απόκλιση των αποδόσεων του στοιχείου χωρίς κίνδυνο ισούται με το μηδέν και έτσι προκύπτει ότι $E(R_f) = R_f$.

Ο κίνδυνος του χαρτοφυλακίου του επενδυτή θα είναι:

$$\sigma_P^2 = [(W_{RF})^2 * \sigma_{RF}^2] + [(1 - W_{RF})^2 * \sigma_\chi^2] + [2 * (W_{RF}) * (1 - W_{RF}) * \rho_{RF,X} * \sigma_{RF} * \sigma_\chi]$$

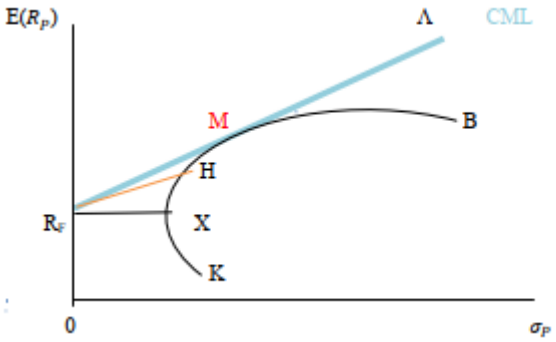
$$=>$$

$$\sigma_\rho = (1 - W_{RF}) * \sigma_\chi \ (2),$$

Ισχύει για $\sigma_{RF} = 0, COV(R_F, R_X) = \sigma_{RF,X} = 0$ *και* $\rho_{RF,X} = 0$

Σχήμα 18.Κίνδυνος χαρτοφυλακίου

Από τις παραπάνω σχέσεις συμπεραίνουμε ότι η αναμενόμενη απόδοση και ο κίνδυνος αυξάνουν γραμμικά, όσο αυξάνεται το ποσοστό των κεφαλαίων το οποίο επενδύεται στο χαρτοφυλάκιο Χ. Συνεπώς, η γραφική απεικόνιση των συνδυασμών αναμενόμενης απόδοσης και του κινδύνου των δύο περιουσιακών στοιχείων θα είναι ευθεία γραμμή, η οποία θα διέρχεται από τα δύο περιουσιακά στοιχεία. Στο Διάγραμμα 1 της γραμμής κεφαλαιαγοράς, απεικονίζονται διάφοροι συνδυασμοί απόδοσης και κινδύνου, από τον συνδυασμό ενός περιουσιακού στοιχείου χωρίς κίνδυνο με χαρτοφυλάκια που ενέχουν κίνδυνο και βρίσκονται στο αποτελεσματικό σύνορο:



Σχήμα 19.Γραμμή κεφαλαιαγοράς

Στο Διάγραμμα 1, παρατηρούμε το χαρτοφυλάκιο Χ και την ευθεία RFX να απεικονίζει τους δυνατούς συνδυασμούς, του στοιχείου χωρίς κίνδυνο και του χαρτοφυλακίου Χ. Στον επενδυτή συνίσταται να επενδύσει σε οποιοδήποτε συνδυασμό που βρίσκεται πάνω στην ευθεία RFX. Όλοι οι υπόλοιποι συνδυασμοί που βρίσκονται πάνω στο αποτελεσματικό σύνορο αλλά κάτω από Χ, είναι λιγότερο αποτελεσματικοί.

Αυτό, ισχύει γιατί για κάθε χαρτοφυλάκιο που βρίσκεται πάνω στο αποτελεσματικό σύνορο (ΚΒ) αλλά κάτω του Χ, υπάρχει κάποιο χαρτοφυλάκιο πάνω στην ευθεία RFX με την ίδια τυπική απόκλιση (κίνδυνο), αλλά μεγαλύτερη αναμενόμενη απόδοση. Το ίδιο ισχύει εάν, ο επενδυτής συνδυάσει το στοιχείο χωρίς κίνδυνο με το χαρτοφυλάκιο Η, το οποίο βρίσκεται πάνω στον αποτελεσματικό κίνδυνο και άνωθεν του Χ. Κάθε σημείο της ευθείας RFH είναι περισσότερο αποτελεσματικό από τα σημεία της ευθείας RFX. Συνεχίζοντας την ίδια διαδικασία η ευθεία γραμμή θα γίνει εφαπτομένη του ΚΒ (αποτελεσματικό σύνορο). Η

ευθεία γραμμή RFM υπερισχύει έναντι των άλλων γραμμών διότι όλα τα χαρτοφυλάκια της είναι πιο αποτελεσματικά εν συγκρίσει με των άλλων ευθειών. Οι γραμμές πάνω από την RFM δεν είναι εφικτές. Η RFM περιλαμβάνει όλα τα αποτελεσματικά χαρτοφυλάκια, όταν ο επενδυτής θα επενδύσει στο χ/φ Μ και στο χ/φ του στοιχείου χωρίς κίνδυνο , δηλαδή όταν ο επενδυτής-πελάτης θα δανείζει και θα εισπράττει την απόδοση χωρίς κίνδυνο. Το αποτελεσματικό σύνορο σ' αυτή την περίπτωση είναι η γραμμή RFMB. Επομένως, η αναμενόμενη απόδοση και ο κίνδυνος του χ/φ αυξάνεται γραμμικά κατά μήκος της ευθείας γραμμής RFM και η προέκταση αυτή υπερτερεί έναντι των άλλων συνδυασμών που βρίσκονται πάνω στο αποτελεσματικό σύνορο ΜΒ και κάτω από τη γραμμή αυτή (για παράδειγμα, του συνδυασμού Β). Άρα, το νέο αποτελεσματικό σύνορο είναι η ευθεία γραμμή RFMΛ που εφάπτεται του παλιού αποτελεσματικού συνόρου ΚΒ στο σημείο Μ. Η γραμμή RFM Λ ονομάζεται γραμμή κεφαλαιαγοράς (Capital Market Line-CML) και απεικονίζει τους όρους ανταλλαγής (trade-off) προσδοκώμενης απόδοσης και κινδύνου για αποτελεσματικά χαρτοφυλάκια, που προσφέρονται σε κατάσταση ισορροπίας.

10. Διαχείριση χαρτοφυλακίου

Αντικείμενο της εμπειρικής μελέτης της εργασίας μας αποτελεί η επίδραση των κεφαλαιακών ελέγχων στις αποδόσεις μετοχών των εταιρειών Pfizer, AbbVie, Lilly, Johnson & Johnson που αποτελούν μέρος του δείκτη S&P 500 και τον χρυσό ως εμπόρευμα μέσω ενός χρηματιστηρίου ανταλλαγής εμπορευμάτων (ETF).

Η βασική ιδέα είναι η διαφοροποίηση του χαρτοφυλακίου ώστε να επιτευχθεί ελάχιστος κίνδυνος χαρτοφυλακίου ή μέγιστες αποδόσεις χαρτοφυλακίου δεδομένου ενός συγκεκριμένου επιπέδου κινδύνου.

Όπως γίνεται αντιληπτό, η κίνηση του παγκόσμιου οικονομικού κύκλου, η πανδημία COVID-19 που ξεκίνησε τον Δεκέμβρη του 2019 από την πόλη Wuhan της Κίνας και εξαπλώθηκε με πολύ μεγάλο ρυθμό σε παγκόσμιο επίπεδο είναι λογικό να επηρεάζει τις τιμές του χαρτοφυλακίου μας και συνολικά τις τιμές των αγορών.

Για τον λόγω αυτό διαλέξαμε και εταιρείες από τον κλάδο της Φαρμακοβιομηχανίας για να ελέγξουμε κατά πόσο η πανδημία έχει επηρεάσει την απόδοση των εταιρειών θετικά η αρνητικά.

Κατά συνέπεια η μελέτη σύγκρισης των προϊόντων μας είναι πολύ σημαντική για να δούμε κατά πόσο ισχύει αυτό.

Τα αποτελέσματα που εξήχθησαν από την έρευνά μας είναι σε βάθος τετραετίας (2017-2020) και παρουσιάζουν τεράστιο ενδιαφέρον ειδικότερα για τον δείκτη μελέτης του χαρτοφυλακίου μας.

Το χαρτοφυλάκιο μας αποτελείται από τους παρακάτω δείκτες:

- PFE (Pfizer)
- ABBV (AbbVie)
- LLY (Lilly)
- JNJ (Johnson & Johnson)
- GLD (Gold)

Η έρευνά παρατίθεται παρακάτω και γίνεται εκτενής αναφορά για τα έτη έρευνας.

Στόχος της συγκεκριμένης έρευνας είναι να δούμε πόσο επηρεάστηκαν οι μεγαλύτερες φαρμακευτικές εταιρείες τόσο τα τελευταία 4 χρόνια όσο και την τελευταία χρονιά με την παγκόσμια πανδημία.

Ας δούμε όμως την έρευνά μας από την αρχή.

10.1 Jupyter

Το Jupyter Notebook είναι μία εφαρμογή ανοιχτού πηγαίου-κώδικα. Χρησιμοποιείται για το διαμοιρασμό εγγράφων τα οποία περιέχουν κώδικα, εξισώσεις, οπτικοποιήσεις και κείμενο. Το Jupyter Notebook προέρχεται από το IPython project και το όνομα προέρχεται από τις προγραμματιστικές γλώσσες τις οποίες υποστηρίζει : Julia, Python και R.

Το Jupyter διανέμεται με τον πυρήνα IPython, ο οποίος επιτρέπει να γραφτούν προγράμματα σε Python. Το Jupyter επιτρέπει να γραφτεί κώδικας σε ξεχωριστά κελιά, τα οποία εκτελούνται αυτόνομα. Αυτό επιτρέπει στον χρήστη να ελέγχει ένα συγκεκριμένο κομμάτι του κώδικα του, χωρίς να είναι απαραίτητο να τρέχει το πρόγραμμα από την αρχή. Αυτό είναι πολύ σημαντικό για την επιστήμη των δεδομένων και τις τεχνικές μηχανικής μάθησης, καθώς επιτρέπει στον χρήστη καλύτερη εκπαίδευση των αλγορίθμων του και την καλύτερη επίβλεψή τους, με αποτέλεσμα την ευκολότερη αποσφαλμάτωση.

10.2 Φόρτωση βιβλιοθηκών

Πριν ξεκινήσουμε να φορτώνουμε τα δεδομένα για την ανάλυση τους θα πρέπει να εισάγουμε ορισμένες βιβλιοθήκες χρήσιμες για την επίτευξη του στόχου μας.

Για την φόρτωση των βιβλιοθηκών ανοίγουμε ένα νέο παράθυρο στο Jupyter και εκεί πληκτρολογούμε: !pip install Numpy, με τον ίδιο τρόπο μπορούμε να φορτώσουμε και την βιβλιοθήκηpandas και ομοίως και τις υπόλοιπες βιβλιοθήκες που θέλουμε να χρησιμοποιήσουμε.

```
In [1]: #Import libraries for stock analysis

        import numpy as np
        import pandas as pd
        from pandas_datareader import data as wb
        import matplotlib.pyplot as plt
        from datetime import datetime
        from dateutil import relativedelta
        from scipy import stats
        import seaborn as sns
        import warnings
        warnings.filterwarnings('ignore', 'The iteration is not making good progress')
        import yfinance as yf
        yf.pdr_override()
```

Σχήμα 20. Βιβλιοθήκες που χρησιμοποιήθηκαν

10.3 Λήψη δεδομένων, φόρτωση και αποτελέσματα

Για την λήψη των δεδομένων χρησιμοποιήσαμε τον ιστότοπο yahoo finance (https://finance.yahoo.com/). Από τον συγκεκριμένο ιστότοπο μπορούμε να λάβουμε οικονομικά δεδομένα που μας ενδιαφέρουν για οποιαδήποτε κλάδο της οικονομίας ενδιαφερόμαστε.

Η φόρτωση των δεδομένων στην περίπτωσή μας έγινε απευθείας από την σελίδα Yahoo finance χωρίς να χρειαστεί να κατεβάσουμε τα δεδομένα σε αρχείο .xls ή .csv . Επιπλέον η χρονιά που ζητήσαμε να ξεκινήσουν να φορτωθούν τα δεδομένα είναι από 01-01-2017 έως και σήμερα 25-4-2020.

Από τους δείκτες που επιλέξαμε θέλουμε να φορτωθεί μόνο το κλείσιμο των δεικτών που είναι η στήλη Adjust Close.

Στη συνέχεια φορτώνουμε τα δεδομένα από την αρχή του έτους 2017 έως και την προηγούμενη ημέρα της έρευνάς μας όπου αποτελεί η τελευταία ημέρα κλεισίματος των δεικτών και αποθήκευσης των δεδομένων[31].

```
In [2]: #Import the tickers, start and end day
        tickers =  ['PFE','ABBV','LLY','JNJ','GLD']
        noa = len(tickers)
        start = '2017-01-01'
        today= datetime.today().strftime ('%Y-%m-%d')
        print (today)

        2020-04-26

In [3]: #Load the data
        mydata = pd.DataFrame()
        for t in tickers:
            mydata[t] = yf.download(t,start,today)['Adj Close']

        [**********100%**********]  1 of 1 completed
        [**********100%**********]  1 of 1 completed
        [**********100%**********]  1 of 1 completed
        [**********100%**********]  1 of 1 completed
        [**********100%**********]  1 of 1 completed
```

Σχήμα 21. Οι δείκτες που χρησιμοποιήθηκαν

Αφού βλέπουμε πως το Dataset φορτώθηκε σωστά το επόμενο βήμα είναι να δούμε τις πέντε (5) πρώτες και τελευταίες χρονοσειρές για τα έτη που επιθυμούμε.

[31] Yves Hilpisch-Python for Finance_ Analyze Big Financial Data-O'Reilly Media (2014) p.323-335

```
In [4]: #Start dataset
        mydata.head(5)
```

Out[4]:

Date	PFE	ABBV	LLY	JNJ	GLD
2017-01-03	29.259394	52.551662	68.996208	106.233673	110.470001
2017-01-04	29.516523	53.292667	69.107185	106.059433	110.860001
2017-01-05	29.800249	53.696846	69.911827	107.169075	112.580002
2017-01-06	29.684984	53.713688	69.985817	106.655518	111.750000
2017-01-09	29.676119	54.067345	70.540756	106.637184	112.669998

```
In [5]: #End dataset
        mydata.tail(5)
```

Out[5]:

Date	PFE	ABBV	LLY	JNJ	GLD
2020-04-20	36.080002	83.989998	157.789993	151.669998	159.699997
2020-04-21	35.619999	80.360001	152.669998	149.679993	158.610001
2020-04-22	36.250000	81.470001	156.710007	152.990005	161.729996
2020-04-23	36.689999	82.040001	159.929993	155.509995	163.339996
2020-04-24	37.380001	83.589996	162.929993	154.860001	162.639999

Σχήμα 22. Οι τιμές των μετοχών

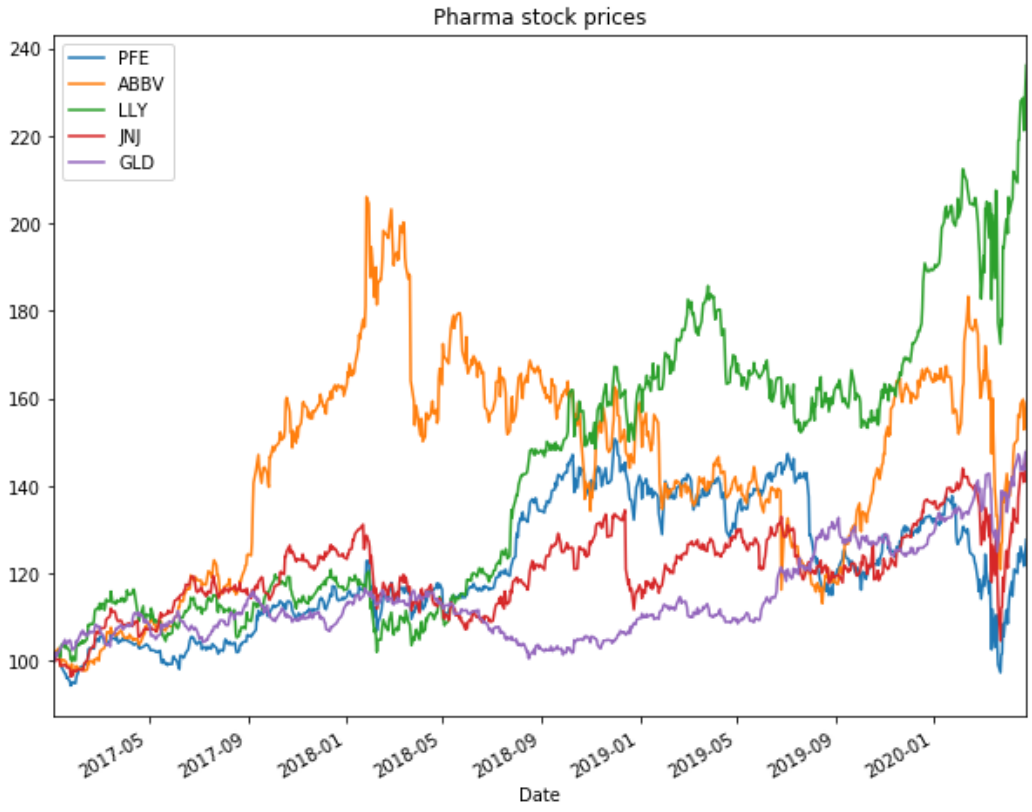
Σε αυτό το βήμα εμφανίζουμε τον αριθμό χαρτοφυλακίων και τα χρόνια επένδυσης.[32]

[32] Yves Hilpisch-Python for Finance_ Analyze Big Financial Data-O'Reilly Media (2014) p.323-335

```
In [7]: #Number of portfolios
        numAssets = len(tickers)
        print ('You have '+ str(numAssets)+' assets in your portfolio')

        You have 5 assets in your portfolio

In [8]: #Years of investment
        d1 = datetime.strptime(start, "%Y-%m-%d")
        d2 = datetime.strptime(today, "%Y-%m-%d")
        delta = relativedelta.relativedelta(d2,d1)
        print ('How many years of investing?')
        print ('%s years'%delta.years)

        How many years of investing?
        3 years
```

Σχήμα 23. Αριθμός των χαρτοφυλακίων και τα χρόνια επένδυσης

Στη συνέχεια ο πίνακάς μας δείχνει τη χρονολογική σειρά των δεδομένων με κανονικό τρόπο.[33]



Σχήμα 23. Χρονολογική σειρά των δεδομένων

[33] Yves Hilpisch-Python for Finance_ Analyze Big Financial Data-O'Reilly Media (2014) p.323-335

Στη συνέχεια μεγάλο ενδιαφέρον παρουσίασαν οι μεγαλύτερες και μικρότερες τιμές στους δείκτες μας. Λόγω ότι στο διάγραμμα δεν μπορούν να φανούν με ακρίβεια προχωρήσαμε στην εμφάνισή τους μέσω του συγκεκριμένου πίνακα.

```
In [10]: #Maximum prices
         for t in tickers:
             print(t + ":",  mydata[t].max())

PFE: 44.124267578125
ABBV: 108.30471801757812
LLY: 162.92999267578125
JNJ: 155.50999450683594
GLD: 163.33999633789062

In [11]: #Minimum prices
         for t in tickers:
             print(t + ":",  mydata[t].min())

PFE: 27.61909294128418
ABBV: 51.032127380371094
LLY: 68.97770690917969
JNJ: 102.49201965332031
GLD: 110.47000122070312
```

Σχήμα 24. Μεγαλύτερες και μικρότερες τιμές

Το επόμενο βήμα είναι η εμφάνιση των ημερήσιων οικονομικών αποδόσεων των δεικτών μας. Σε αυτό το βήμα εμφανίσαμε τις αποδώσεις χωρίς την εμφάνιση ελλείπων τιμών.[34]

```
In [12]: #Returns of prices and drop NaN prices
returns = pd.DataFrame()
for t in tickers:
    returns[t + " Return"] = (np.log(1 + mydata[t].pct_change())).dropna()

returns.head(5)
```

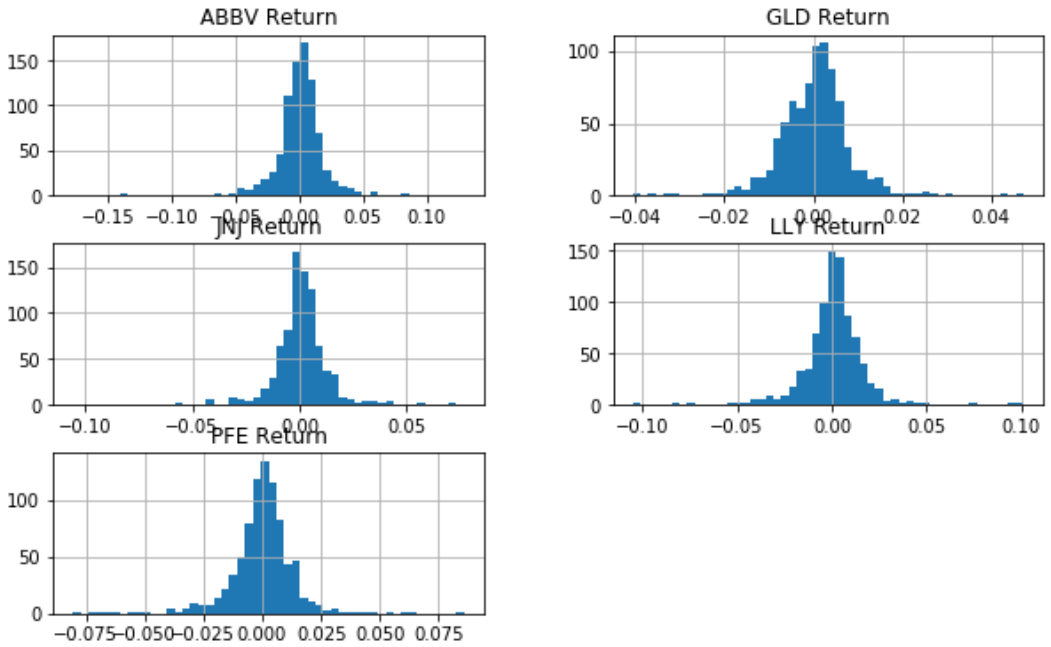
Out[12]:

Date	PFE Return	ABBV Return	LLY Return	JNJ Return	GLD Return
2017-01-04	0.008750	0.014002	0.001607	-0.001642	0.003524
2017-01-05	0.009567	0.007556	0.011576	0.010408	0.015396
2017-01-06	-0.003875	0.000314	0.001058	-0.004804	-0.007400
2017-01-09	-0.000299	0.006563	0.007898	-0.000172	0.008199
2017-01-10	-0.000897	-0.002183	0.000000	-0.001033	0.004251

Σχήμα 25. Ημερήσιες οικονομικές αποδόσεις

[34] Yves Hilpisch-Python for Finance_ Analyze Big Financial Data-O'Reilly Media (2014) p.323-335

Επόμενό μας βήμα είναι η εμφάνιση των ημερήσιων οικονομικών αποδόσεων σε ένα ιστόγραμμα. Από το ιστόγραμμα ότι τα δεδομένα μας αποτελούν κομμάτι της κανονικής κατανομής καθώς είναι συγκεντρωμένα στο διάστημα από -1σ μέχρι 1σ [35]



Σχήμα 26. Ημερήσιες οικονομικές αποδόσεις σε ιστόγραμμα

[35] Yves Hilpisch-Python for Finance_ Analyze Big Financial Data-O'Reilly Media (2014) p.323-335

Στη συνέχεια και με βάση τις αποδόσεις βλέπουμε ημερολογιακά πότε επετεύχθη η καλύτερη και χειρότερη μέρα του κάθε δείκτη. Το βήμα αυτό έχει ως σκοπό να μας δώσει παραπάνω πληροφορίες για τους δείκτες μας.[36]

```
In [14]: #Print best and worst day in stocks
         print('Best Day Returns')
         print('-'*25)
         print(returns.idxmax())
         print('\n')
         print('Worst Day Returns')
         print('-'*25)
         print(returns.idxmin())
```

```
Best Day Returns
-------------------------
PFE Return    2020-03-13
ABBV Return   2018-01-26
LLY Return    2020-03-17
JNJ Return    2020-03-30
GLD Return    2020-03-24
dtype: datetime64[ns]


Worst Day Returns
-------------------------
PFE Return    2020-03-16
ABBV Return   2019-06-25
LLY Return    2020-03-12
JNJ Return    2018-12-14
GLD Return    2020-03-12
dtype: datetime64[ns]
```

Σχήμα 27. Καλύτερη και χειρότερη ημέρα

[36] Yves Hilpisch-Python for Finance_ Analyze Big Financial Data-O'Reilly Media (2014) p.323-335

Κατά τη διάρκεια της χρονολογικής σειράς των δεδομένων, παρατηρούμε σημαντικές διαφορές στην ετήσια απόδοση. Χρησιμοποιούμε συντελεστή 250 ημερών συναλλαγών για να κάνουμε ετήσιες τις ημερήσιες αποδόσεις:

```
In [15]: #Annualy mean
         returns.mean()*250

Out[15]: PFE Return     0.073598
         ABBV Return    0.139461
         LLY Return     0.258194
         JNJ Return     0.113245
         GLD Return     0.116225
         dtype: float64

In [16]: #Annualy Variance
         returns.var()*250

Out[16]: PFE Return     0.047510
         ABBV Return    0.093957
         LLY Return     0.061494
         JNJ Return     0.048091
         GLD Return     0.015225
         dtype: float64
```

Σχήμα 28. Πίνακας μέσης τιμής και διακύμανσης

Ο πίνακας συνδιακύμανσης για τα προς επένδυση περιουσιακά στοιχεία είναι το κεντρικό κομμάτι για ολόκληρη τη διαδικασία επιλογής χαρτοφυλακίου. Ο πίνακας συσχέτισης των αποδόσεων των αξιογράφων διατηρεί τις ιδιότητες της συνδιακύμανσης και μετρά εντός ενός ορίου ±1 το βαθμό μεταβολής των αποδόσεων των μετοχών στον συγκεκριμένο πίνακά μας δεν είναι ±1 οι αποδόσεις γιατί τις έχουμε μετατρέψει σε ετήσιες τις τιμές όπου δεν επηρεάζει σε κάτι το αποτέλεσμα μας καθώς είναι απολύτως ακριβές εάν αφαιρούσαμε το κομμάτι του πολλαπλασιασμού των ημερών ετησίως και παίρναμε την ημερήσια συσχέτιση. Αυτό που παρατηρούμε στην συσχέτιση και για να βεβαιωθούμε πως το αποτέλεσμά μας είναι ακριβές πρέπει διαγώνια τα δεδομένα μας να είναι στο 1. Η βιβλιοθήκη pandas έχει μια μέθοδο ενσωμάτωσης για τη δημιουργία πίνακα συνδιακύμανσης και συσχέτισης.[37]

[37] Yves Hilpisch-Python for Finance_ Analyze Big Financial Data-O'Reilly Media (2014) p.323-335

```
In [17]: #Annualy Covvariance
         returns.cov()*250
```

Out[17]:

	PFE Return	ABBV Return	LLY Return	JNJ Return	GLD Return
PFE Return	0.047510	0.035558	0.038134	0.030858	0.000392
ABBV Return	0.035558	0.093957	0.037266	0.029903	-0.000945
LLY Return	0.038134	0.037266	0.061494	0.034509	0.001097
JNJ Return	0.030858	0.029903	0.034509	0.048091	0.000699
GLD Return	0.000392	-0.000945	0.001097	0.000699	0.015225

```
In [18]: #Annualy Correlation
         returns.corr()*250
```

Out[18]:

	PFE Return	ABBV Return	LLY Return	JNJ Return	GLD Return
PFE Return	250.000000	133.050583	176.377416	161.393423	3.640221
ABBV Return	133.050583	250.000000	122.566565	111.212546	-6.248032
LLY Return	176.377416	122.566565	250.000000	158.643011	8.963978
JNJ Return	161.393423	111.212546	158.643011	250.000000	6.462248
GLD Return	3.640221	-6.248032	8.963978	6.462248	250.000000

Σχήμα 29. Πίνακας συνδιακύμανσης και συσχέτισης

Στη συνέχεια, υποθέτουμε ότι δεν επιτρέπεται σε έναν επενδυτή να θέσει χαμηλές τιμές πώλησης μιας κινητής αξίας. Επιτρέπονται μόνο θετικές θέσεις, πράγμα που σημαίνει ότι το 100% του πλούτου του επενδυτή πρέπει να διαιρεθεί μεταξύ των διαθέσιμων περιουσιακών στοιχείων κατά τρόπο ώστε να είναι όλες οι θέσεις long (θετικό) και ότι οι θέσεις προσθέτουν έως και 100%. Δεδομένων των πέντε αξιών, εσείς θα μπορούσατε, για παράδειγμα, να επενδύσετε ίσα ποσά σε κάθε κινητή αξία (δηλαδή, το 20% του πλούτου σας σε κάθε μία). Ο ακόλουθος κώδικας δημιουργεί πέντε τυχαίους αριθμούς μεταξύ 0 και 1 και στη συνέχεια ομαλοποιεί τις τιμές έτσι ώστε το άθροισμα όλων των τιμών να ισούται με 1.

Τώρα μπορείτε να ελέγξετε ότι τα βάρη περιουσιακών στοιχείων έχουν πράγματι έως 1δηλαδή,$\Sigma w_1 = 1$ όπουI είναι ο αριθμός των περιουσιακών στοιχείων και$w \geq 0$ είναι το βάρος του περιουσιακού στοιχείου i. Η εξίσωση παρέχει τη φόρμουλα για την αναμενόμενη απόδοση χαρτοφυλακίου λαμβάνοντας υπόψη τα βάρη για τους μεμονωμένες κινητές αξίες. Αυτή είναι η αναμενόμενη επιστροφή χαρτοφυλακίου υπό την έννοια ότι η ιστορική μέση απόδοση θεωρείται να είναι ο καλύτερος εκτιμητής για τη μελλοντική (αναμενόμενη)

απόδοση. Εδώ, οι r_i είναι οι δηλωμένες ανεξάρτητες μελλοντικές αποδόσεις (διάνυσμα με τιμές επιστροφής που θεωρείται ότι διανέμονται κανονικά) και μ_i είναι η αναμενόμενη απόδοση για την κινητή αξία i. Τέλος, w^Tείναι η μεταφορά των βαρών του διανύσματος και μ είναι το διάνυσμα των αναμενόμενων επιστροφών των κινητών αξιών.[38]

Στη συνέχεια με τον ίδιο τρόπο βγάλαμε την αναμενόμενη διακύμανση χαρτοφυλακίου. Η συνάρτηση κουκκίδας (dot function) δίνει το προϊόν κουκίδων δύο διανυσμάτων.

Η μέθοδος T (transpose) δίνει τη μεταφορά ενός διανύσματος ή πίνακα:

Η (αναμενόμενη) τυπική απόκλιση χαρτοφυλακίου ή μεταβλητότητα είναι η τετραγωνική ρίζα της διακύμανσης.

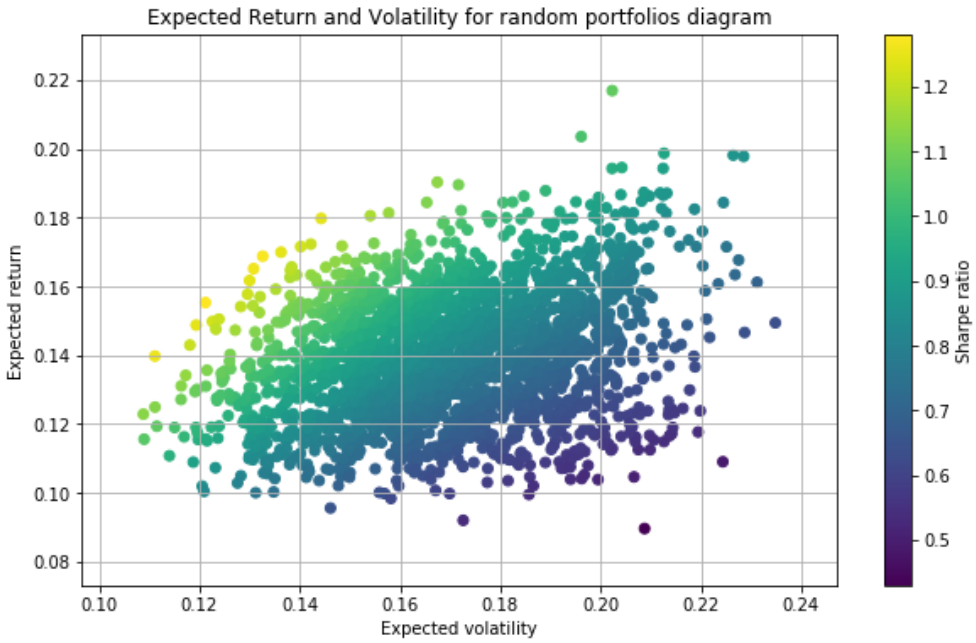
```
In [19]: #Random weights
         weights = np.random.random(noa)
         weights /= np.sum(weights)

In [20]: weights

Out[20]: array([0.03489294, 0.30603331, 0.22071137, 0.1393743 , 0.29898807])

In [21]: #Expected portfolio return
         np.sum(returns.mean()*weights)*250

Out[21]: 0.15276739726666808

In [22]: #Expected portfolio Variance
         np.dot(weights.T,np.dot(returns.cov()*250,weights))

Out[22]: 0.025541640596142934

In [23]: #Expected std/volatility
         np.sqrt(np.dot(weights.T,np.dot(returns.cov()*250,weights)))

Out[23]: 0.15981752280692796
```

Σχήμα 30. Αναμενόμενη απόδοση και τυπική απόκλιση χαρτοφυλακίου

Αυτό ολοκληρώνει κυρίως το σύνολο εργαλείων για την επιλογή μέσης διακύμανσης χαρτοφυλακίου. Υψίστης σημασίας το ενδιαφέρον για τους επενδυτές είναι ποια είναι τα προφίλ απόδοσης κινδύνου για ένα δεδομένο σύνολο τίτλων, και τα στατιστικά χαρακτηριστικά τους. Για το σκοπό αυτό, εφαρμόζουμε μια προσομοίωση Monte Carlo για τη δημιουργία τυχαίων διανυσμάτων βάρους χαρτοφυλακίου σε μεγαλύτερη κλίμακα. Για

[38] Yves Hilpisch-Python for Finance_ Analyze Big Financial Data-O'Reilly Media (2014) p.323-335

κάθε προσομοιωμένη κατανομή, καταγράφουμε την αναμενόμενη απόδοση που προκύπτει και τη διακύμανση χαρτοφυλακίου.



Σχήμα 31. Αναμενόμενη απόδοση που προκύπτει από την διακύμανση χαρτοφυλακίου

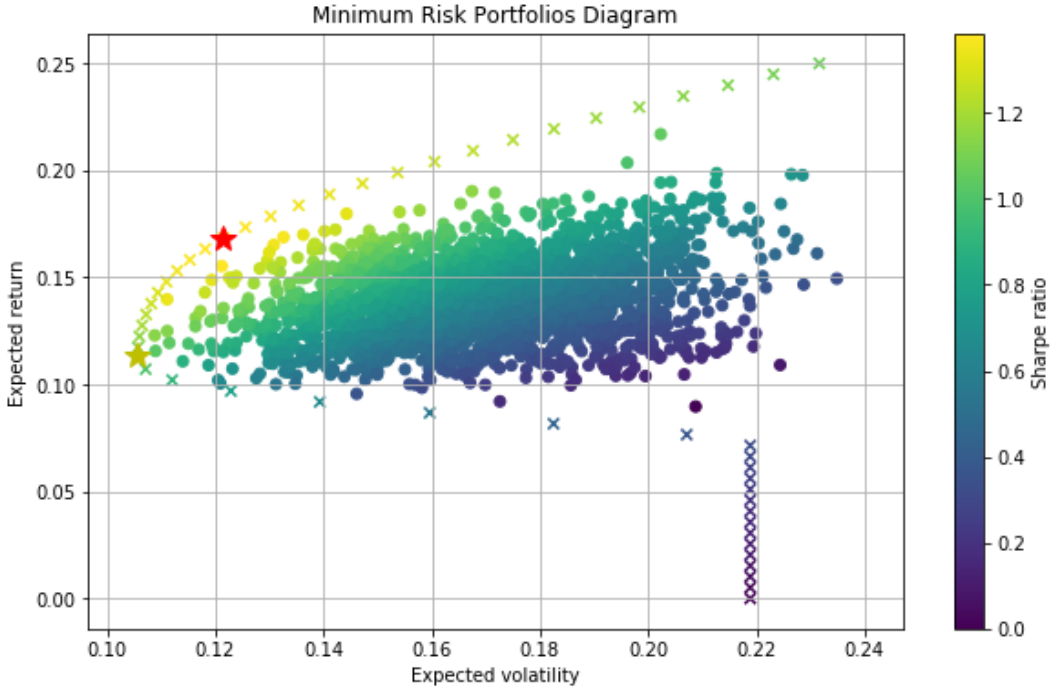
Είναι σαφές από την επιθεώρηση του σχήματος ότι αποδίδουν καλά όλες οι κατανομές βάρους όταν μετρούνται σε όρους μέσης και διακύμανσης. Για παράδειγμα, για ένα σταθερό επίπεδο κινδύνου, ας πούμε, 20%, υπάρχουν πολλά χαρτοφυλάκια που όλα δείχνουν παρόμοιες αποδόσεις. Όπως ένας επενδυτής ενδιαφέρεται γενικά για τη μέγιστη απόδοση δεδομένου ενός σταθερού επιπέδου κινδύνου ή του ελάχιστου κινδύνου δεδομένης μιας σταθερής προσδοκίας απόδοσης. Αυτό το σύνολο χαρτοφυλακίων αποτελεί το λεγόμενο αποτελεσματικό σύνορο.

Η παραγωγή όλων των βέλτιστων χαρτοφυλακίων - δηλαδή, όλων των χαρτοφυλακίων με ελάχιστη μεταβλητότητα για ένα δεδομένο επίπεδο απόδοσης στόχου (ή όλα τα χαρτοφυλάκια με μέγιστη απόδοση για ένα δεδομένο επίπεδο κινδύνου) - είναι παρόμοια με τις προηγούμενες βελτιστοποιήσεις. Η μόνη διαφορά είναι ότι πρέπει να επαναλάβουμε πολλές συνθήκες εκκίνησης. Η προσέγγιση που ακολουθούμε είναι ότι καθορίζουμε ένα επίπεδο απόδοσης στόχου και αντλούμε για κάθε τέτοιο επίπεδο τα βάρη χαρτοφυλακίου που οδηγούν στην ελάχιστη τιμή μεταβλητότητας. Για τη βελτιστοποίηση, αυτό οδηγεί σε δύο προϋποθέσεις: μία για το επίπεδο επιστροφής στόχου και μία για το άθροισμα των

βαρών χαρτοφυλακίου όπως προηγουμένως. Οι οριακές τιμές για κάθε παράμετρο παραμένουν οι ίδιες.[39]Για λόγους σαφήνειας, ορίζουμε μια ειδική συνάρτηση *min_func* για χρήση στη διαδικασία ελαχιστοποίησης. Η συνάρτηση αυτή αποδίδει απλώς την τιμή μεταβλητότητας από τη συνάρτηση στατιστικών στοιχείων:

Κατά την επανάληψη σε διαφορετικά επίπεδα στοχευμένης απόδοσης (trets), αλλάζει μία συνθήκη για την ελαχιστοποίηση. Γι'αυτό το λεξικό συνθηκών ενημερώνεται σε κάθε βρόχο:

Στο σχήμα παρακάτω δείχνει τα αποτελέσματα βελτιστοποίησης. Οι σταυροί υποδεικνύουν τα βέλτιστα χαρτοφυλάκια με δεδομένη συγκεκριμένη απόδοση. Οι τελείες είναι, όπως και πριν, τα τυχαία χαρτοφυλάκια. Επιπλέον, το σχήμα δείχνει δύο μεγαλύτερα αστέρια: ένα για το χαρτοφυλάκιο ελάχιστης μεταβλητότητας / διακύμανσης (το αριστερότερο χαρτοφυλάκιο) και ένα για το χαρτοφυλάκιο με τη μέγιστη αναλογία Sharpe.[40]



Σχήμα 32. Ελάχιστο ρίσκο χαρτοφυλακίου

Το αποδοτικό σύνορο αποτελείται από όλα τα βέλτιστα χαρτοφυλάκια με υψηλότερη απόδοση από το απόλυτο χαρτοφυλάκιο ελάχιστης απόκλισης. Αυτά τα χαρτοφυλάκια

[39] Yves Hilpisch-Python for Finance_ Analyze Big Financial Data-O'Reilly Media (2014) p.323-335
[40] Yves Hilpisch-Python for Finance_ Analyze Big Financial Data-O'Reilly Media (2014) p.323-335

κυριαρχούν σε όλα τα άλλα χαρτοφυλάκια όσον αφορά τις αναμενόμενες αποδόσεις, δεδομένου ενός συγκεκριμένου επιπέδου κινδύνου.

Για να ακολουθήσουν οι υπολογισμοί για την γραμμή κεφαλαιαγοράς, χρειαζόμαστε μια λειτουργική προσέγγιση και το πρώτο παράγωγο για το αποδοτικό όριο. Για το σκοπό αυτό χρησιμοποιούμε παρεμβολή κυβικών σπινθηρισμών.

```
import scipy.interpolate as sci
```

Για την παρεμβολή spline, χρησιμοποιούμε μόνο τα χαρτοφυλάκια από τα αποδοτικά σύνορα. Ο ακόλουθος κώδικας επιλέγει ακριβώς αυτά τα χαρτοφυλάκια από τα σύνολα tvols και τα trets που χρησιμοποιήσαμε στο παρελθόν:

```
ind = np.argmin(tvols)
evols = tvols[ind:]
erets = trets[ind:]
```

Τα νέα αντικείμενα ndarray evols και erets χρησιμοποιούνται για την παρεμβολή:

```
tck = sci.splrep(evols, erets)
```

Μέσω αυτής της αριθμητικής διαδρομής καταλήγουμε σε θέση να ορίσουμε μια συνεχώς διαφοροποιημένη συνάρτηση f (x) για το αποδοτικό όριο και την αντίστοιχη πρώτη παράγωγη συνάρτηση df (x):

```
def f(x):
    ''' Efficient frontier function (splines approximation). '''
    return sci.splev(x, tck, der=0)
def df(x):
    ''' First derivative of efficient frontier function. '''
    return sci.splev(x, tck, der=1)
```

Αυτό που ψάχνουμε είναι μια συνάρτηση t (x) = a + b · x που περιγράφει τη γραμμή που περνά μέσα από το περιουσιακό στοιχείο χωρίς κίνδυνο στον χώρο απόδοσης κινδύνου και είναι εφαπτόμενη στα αποτελεσματικά σύνορα. Η εξίσωση περιγράφει και τις τρεις συνθήκες που πρέπει να πληροί η συνάρτηση t (x).Για το σκοπό αυτό, ορίζουμε μια συνάρτηση που επιστρέφει τις τιμές και των τριών εξισώσεων δεδομένου του συνόλου παραμέτρων p = (a, b, x):

```
def equations(p, rf=0.01):
    eq1 = rf - p[0]
    eq2 = rf + p[1] * p[2] - f(p[2])
    eq3 = p[1] - df(p[2])
    return eq1, eq2, eq3
```

Η συνάρτηση f solve από το scipy.optimize είναι ικανή να επιλύει ένα τέτοιο σύστημα εξισώσεων. Παρέχουμε μια αρχική παραμετροποίηση επιπλέον των εξισώσεων συνάρτησης. Σημειώστε ότι η επιτυχία ή η αποτυχία της βελτιστοποίησης μπορεί να εξαρτάται από την αρχική παραμετροποίηση, η οποία επομένως πρέπει να επιλεγεί προσεκτικά - γενικά με έναν συνδυασμό μορφωμένων εικαστικών με δοκιμή και σφάλμα.[41]

```
opt = sco.fsolve(equations, [0.01, 0.5, 0.15])
```

```
opt = sco.fsolve(equations, [0.01, 0.5, 0.15])
```

Η αριθμητική βελτιστοποίηση αποδίδει τις ακόλουθες τιμές. Όπως είναι επιθυμητό, έχουμε a = rf = 0,01 ενώ οι τρεις εξισώσεις είναι επίσης, όπως είναι επιθυμητό, εμφανίζουν μηδέν.

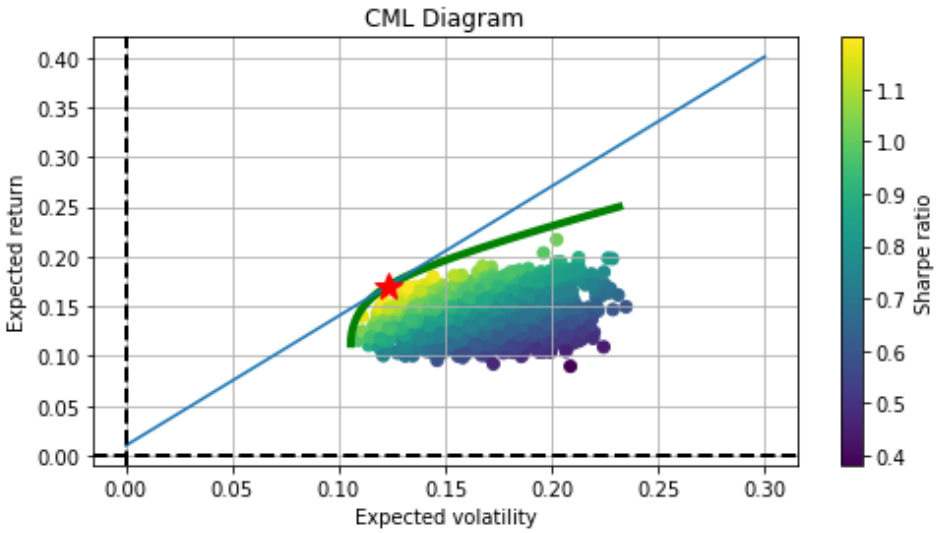
```
In [50]: opt = sco.fsolve(equations, [0.01, 0.5, 0.15])

In [51]: opt
Out[51]: array([0.01      , 1.30324742, 0.12276302])

In [52]: np.round(equations(opt),6)
Out[52]: array([ 0.,  0., -0.])
```

Στο διάγραμμα παρουσιάζονται τα αποτελέσματα γραφικά: το αστέρι αντιπροσωπεύει το βέλτιστο χαρτοφυλάκιο από το αποδοτικό όριο όπου η εφαπτομένη γραμμή περνά μέσω του

[41] Yves Hilpisch-Python for Finance_ Analyze Big Financial Data-O'Reilly Media (2014) p.323-335

σημείου περιουσιακού στοιχείου χωρίς κίνδυνο. Το βέλτιστο χαρτοφυλάκιο έχει αναμενόμενη μεταβλητότητα 10,27% και αναμενόμενη απόδοση 8,27%.

```
plt.figure(figsize=(8, 4))
plt.scatter(pvols, prets,
            c=(prets - 0.01) / pvols, marker='o')
            # random portfolio composition
plt.plot(evols, erets, 'g', lw=4.0)
            # efficient frontier
cx = np.linspace(0.0, 0.3)
plt.plot(cx, opt[0] + opt[1] * cx, lw=1.5)
            # capital market line
plt.plot(opt[2], f(opt[2]), 'r*', markersize=15.0)
plt.grid(True)
plt.axhline(0, color='k', ls='--', lw=2.0)
plt.axvline(0, color='k', ls='--', lw=2.0)
plt.xlabel('expected volatility')
plt.ylabel('expected return')
plt.colorbar(label='Sharpe ratio')
```



Σχήμα 33. Γραμμή κεφαλαιαγοράς

Στο διάγραμμα παρουσιάζονται τα αποτελέσματα γραφικά: το αστέρι αντιπροσωπεύει το βέλτιστο χαρτοφυλάκιο από το αποδοτικό όριο όπου η εφαπτομένη γραμμή περνά μέσω του σημείου περιουσιακού στοιχείου χωρίς κίνδυνο. Το βέλτιστο χαρτοφυλάκιο έχει αναμενόμενη μεταβλητότητα 12% και αναμενόμενη απόδοση 11%.

Το συμπέρασμα που βγαίνει από την πάροδο των ετών είναι ότι το χαρτοφυλάκιό μας είναι κάτω από τη CML επομένως το χαρτοφυλάκιό μας είχε κατώτερη απόδοση ανάλογα με τον συνολικό κίνδυνο και έτσι αποτελεί ένα ασφαλές χαρτοφυλάκιο προς επένδυση.[42]

[42] Yves Hilpisch-Python for Finance_ Analyze Big Financial Data-O'Reilly Media (2014) p.323-335

11. Πρόβλεψη χρονοσειρών

Η έρευνά μας όμως συνεχίζεται και σε αυτό το σημείο θα δούμε πως γίνεται η πρόβλεψη χρονοσειρών καθώς και αλγορίθμους πρόβλεψης σε πραγματικά δεδομένα.

Ας ξεκινήσουμε όμως από το κομμάτι ορισμού της χρονοσειράς.

Το σύνολο των δεδομένων, τα οποία συλλέγονται διαχρονικά και εκφράζουν την εξέλιξη των τιμών μιας μεταβλητής κατά την διάρκεια ίσων διαδοχικών χρονικών περιόδων ονομάζεται χρονοσειρά.

Μια χρονοσειρά αποτελείται από ένα σύνολο παρατηρήσεων όπου οι τιμές αποτελούν κομμάτι ίσων χρονικών περιόδων. Ένα αντιπροσωπευτικό δείγμα εφαρμογής χρονοσειράς αποτελούν τα χρηματιστηριακά δεδομένα τα οποία είναι αποτελούν χρονικές σειρές μήνα, τριμήνου, έτους.

Οι χρονοσειρές απαιτούν μόνο τις παρελθοντικές τιμές της μεταβλητής των διαδοχικών καταστάσεων στο χρόνο. Έτσι, μπορούν να αναλυθούν ώστε να εξάγουμε συμπεράσματα για την συμπεριφορά της μεταβλητής. Με βάση την πληροφορία από το παρελθόν, μας επιτρέπεται να προβλέψουμε και να βγάλουμε κάποια ασφαλή συμπεράσματα σχετικά με τις τιμές στο μέλλον.

Οι χρονοσειρές διακρίνονται σε δύο κατηγορίες, στις συνεχείς και τις διακριτές.

Συνεχείς χρονοσειρές ονομάζονται αυτές που η τιμή του φαινομένου παρατηρείται συνεχώς (συνεχόμενη παρατήρηση). Ένα παράδειγμα συνεχής χρονοσειράς είναι η καταγραφή της σεισμικής δόνησης καθώς οι τιμές είναι συνεχόμενες ή εναλλακτικά η μέτρηση της θερμοκρασίας, όπου σε διαφορετικούς χρόνους παρουσιάζει διαφορετικές ενδείξεις (αύξησης ή μείωσης).

Διακριτές χρονοσειρές ονομάζονται αυτές που η τιμή του φαινομένου καταγράφεται σε ορισμένα χρονικά διαστήματα. Παράδειγμα διακριτών χρονοσειρών είναι η τιμή μιας μετοχής ανά ημέρα όπου υπάρχουν τιμές σε συγκεκριμένα χρονικά διαστήματα.

Οι χρονοσειρές έχουν πεδίο εφαρμογές σε διάφορες επιστήμες όπως στα Οικονομικά, την Ιατρική, κ.ά.

Μερικά παραδείγματα χρονοσειρών είναι η ημερήσια τιμή κλεισίματος μετοχών στο Χρηματιστήριο, οι μηνιαίες πωλήσεις ενός προϊόντος, οι ημερήσιες θερμοκρασίες μιας πόλης.

11.1 Πως εξάγεται μια χρονοσειρά

Για να οριστεί και να μελετηθεί σωστά μια χρονοσειρά πρέπει κανείς να ξεκινήσει με την επισκόπηση του γραφήματός της στο πεδίο του χρόνου, από το οποίο μπορούν να ανιχνευθούν τα βασικά χαρακτηριστικά της όπως η τάση, η κυκλικότητα, η εποχικότητα και οι ακραίες τιμές. Ας δούμε όμως αναλυτικά τα παραπάνω χαρακτηριστικά.

i. Ως τάση (trend) ορίζεται η μακροπρόθεσμη μεταβολή του μέσου επιπέδου των τιμών μιας χρονοσειράς. Η τάση των τιμών μπορεί να έχει τρία χαρακτηριστικά σε ένα συγκεκριμένο χρονικό διάστημα. Να έχει είτε ανοδική είτε πτωτική είτε σταθερή τάση σε ένα συγκεκριμένο χρονικό διάστημα.
   Η διάκρισή διαγραμματικά μπορεί να έχει τη μορφή διάφορων καμπυλών, όπως μια ευθεία γραμμή είτε μια εκθετική καμπύλη. Για να είναι ασφαλή τα συμπεράσματα που θα εξαχθούν για το αν μια σειρά παρουσιάζει τάση ή όχι θα πρέπει να έχουμε ένα ικανό αριθμό παρατηρήσεων και να εκτιμηθεί ένα κατάλληλο χρονικό διάστημα.

ii. Ώς κυκλικότητα (cyclic) μπορεί να οριστεί μια μεταβολή η οποία εμφανίζεται λόγω εξωγενών παραγόντων κατά μεγάλες περιόδους.
   Οι περίοδοι αυτοί συνηθίζεται να είναι μεγαλύτερες του έτους και κυμαίνονται να αποτελούν στοιχεία πενταετίας-δεκαετίας χωρίς όμως αυτό να σημαίνει ότι είναι σταθερού μήκους . Στις γραφικές παραστάσεις των χρονοσειρών παρουσιάζεται ως μια κυματοειδής γραμμή που κινείται ανάμεσα στην υψηλότερη και χαμηλότερη στάθμη. Η κυκλικότητα εμφανίζεται κυρίως σε οικονομικές χρονοσειρές με πιο σύνηθες παράδειγμα το Ακαθάριστο Εθνικό Προϊόν (ΑΕΠ), λόγω των ανόδων και των υφέσεων που παρουσιάζουν οι οικονομίες.

iii. Η εποχικότητα (seasonal) ορίζεται ως μια περιοδική διακύμανση η οποία έχει σταθερό-μικρότερο ή ίσο μήκος ενός έτους. Η διακύμανση αυτή είναι άμεσα

κατανοητή και προβλέψιμη, διότι τα δεδομένα ορισμένων χρονοσειρών επαναλαμβάνονται με τον ίδιο περίπου τρόπο σε σχέση με το χρόνο.

Μερικά παραδείγματα κατανόησης είναι η κατανάλωσης του πετρελαίου θέρμανσης, η οποία είναι μεγαλύτερη κατά τους χειμερινούς μήνες, η μηνιαία κατανάλωση παγωτού όπου είναι μεγαλύτερη κατά την καλοκαιρινή περίοδο. Εφόσον, η εποχική διακύμανση παρουσιάζεται με συστηματικό τρόπο, είναι ένα χαρακτηριστικό που αναγνωρίζεται οπτικά εύκολα και που μπορεί να μετρηθεί και να απομονωθεί εύκολα ώστε να μην επηρεάζει τα δεδομένα μας.

iv. Οι ακραίες τιμές (outliers) χαρακτηρίζονται ως οι απομονωμένες παρατηρήσεις που εμφανίζονται στο γράφημα κάποιας χρονοσειράς ως απότομες αλλαγές στο πρότυπο συμπεριφοράς της. Είναι μη προβλέψιμες και η επίδρασή τους στην χρονοσειρά έχει μικρή χρονική διάρκεια. Η ερμηνεία τέτοιων παρατηρήσεων χρειάζεται ιδιαίτερη προσοχή, διότι απαιτείται θεωρητική γνώση, κριτική ικανότητα και κοινή λογική. Ένα outlier μπορεί να αντιπροσωπεύει μια ασυνήθιστη παρατήρηση που οφείλεται σε κάποιο απρόβλεπτο γεγονός.

12. Αλγόριθμοι Παλινδρόμησης / Regression

12.1 Linear regression

Ένας πολύ διαδεδομένος αλγόριθμος που ανήκει στην κατηγορία μάθησης με επίβλεψη είναι ο Linear Regression (γραμμική παλινδρόμηση) ή εν συντομία LR. Από το όνομά του ο Linear Regression αποτελεί ένα γραμμικό μοντέλο πιο αναλυτικά θεωρεί πως υπάρχει μια γραμμική σχέση μεταξύ των μεταβλητών εισόδου και εξόδου. Η έξοδος όπου συμβολίζεται συνήθως με το λατινικό (y) μπορεί να υπολογιστεί από ένα γραμμικό συνδυασμό της εισόδου (x). Τόσο οι τιμές εισόδου όσο και εξόδου είναι αριθμητικές. Ακόμη, ανατίθεται και ένας συντελεστής παλινδρόμησης (β ή coefficient) σχετικός με το πρόβλημα που καλείται να επιλύσει το μοντέλο σε κάθε τιμή εισόδου και ένα όρο διαταραχής 'ε' στο τέλος που υποδηλώνει πιθανούς άγνωστους παράγοντες που επηρεάζουν την έξοδο. Η διαδικασία αυτή γίνεται με τη μέθοδο Ordinary Least Squares (OLS).

Η OLS είναι ουσιαστικά αυτό που παράγει τους συντελεστές παλινδρόμησης της εξίσωσης. Πιο συγκεκριμένα, βασικότερος στόχος του LR είναι η δημιουργία της βέλτιστης εξίσωσης και αυτό το πετυχαίνει με τη μείωση των λαθών στην εύρεση των καλύτερων συντελεστών παλινδρόμησης.

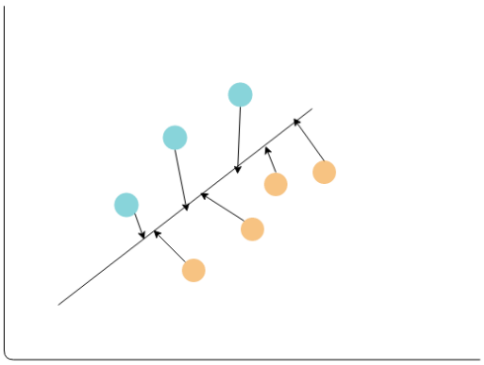
Σαν λάθος (error) ορίζεται η απόσταση από ένα σημείο έως τη γραμμή hyperplane (Σχήμα 36).

Η μέθοδος OLS λειτουργεί επιτυχημένα όταν η εξίσωση που θα δημιουργεί έχει το μικρότερο δυνατό άθροισμα τετραγώνων απόστασης από τα σημεία προς το hyperplane. Όσες περισσότερες διαφορετικές μεταβλητές (διαστάσεις) εισάγουμε τόσο περισσότερο αυξάνεται η δυσκολία εύρεσης των κατάλληλων συντελεστών. Για παράδειγμα, αν φτιάχνουμε ένα προγνωστικό μοντέλο για την ταχύτητα ενός αυτοκινήτου τότε θα έχουμε σαν μεταβλητές το βάρος του, την ιπποδύναμη, τη βαρύτητα, την κλίση του δρόμου, την κατάσταση του οδοστρώματος, την αντίσταση του αέρα και ένα τυχαίο απρόβλεπτο εξωτερικό παράγοντα.

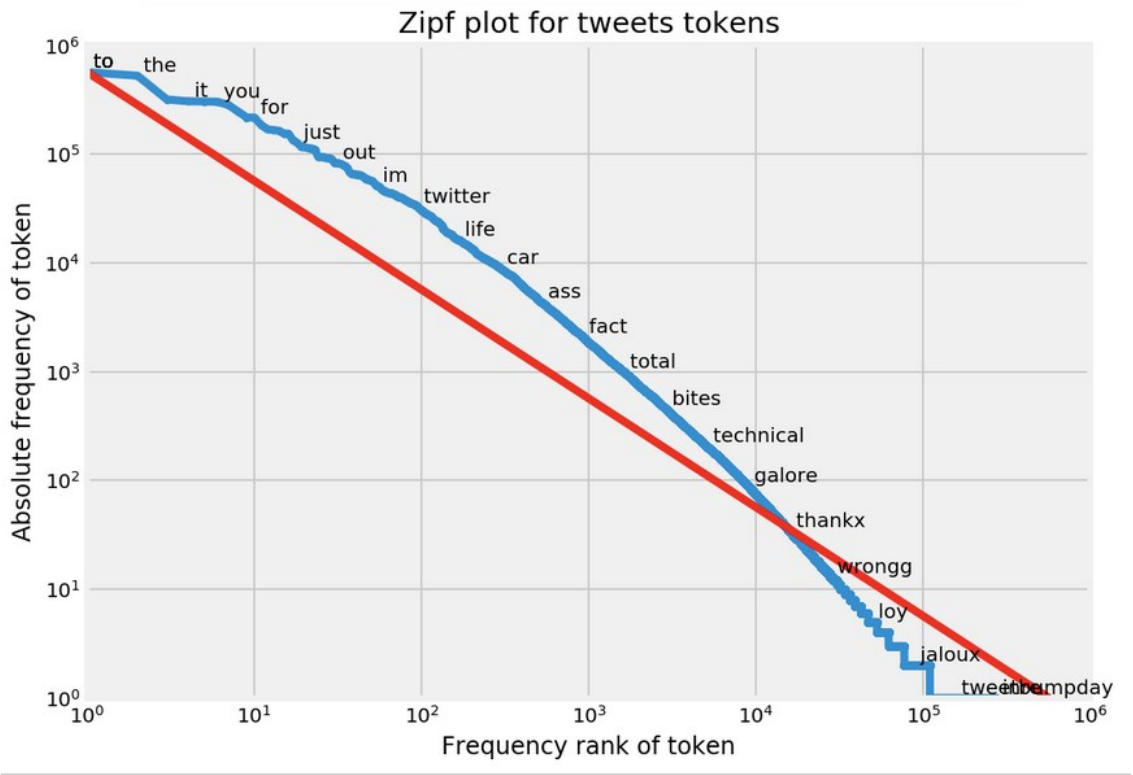
$$y = \beta_0 1 + \beta_1 * \chi_1 + \beta_2 * \chi_2 + \cdots + \beta_n \chi_n + \varepsilon$$

Σχήμα 34. Μέθοδος OLS

Δημιουργώντας ένα σωστό μοντέλο κατά τη διαδικασία της εκμάθησης μπορούμε να επιλέξουμε το βέλτιστο συντελεστή και να λάβουμε τα καλύτερα δυνατά αποτελέσματα σε μελλοντικές προβλέψεις.



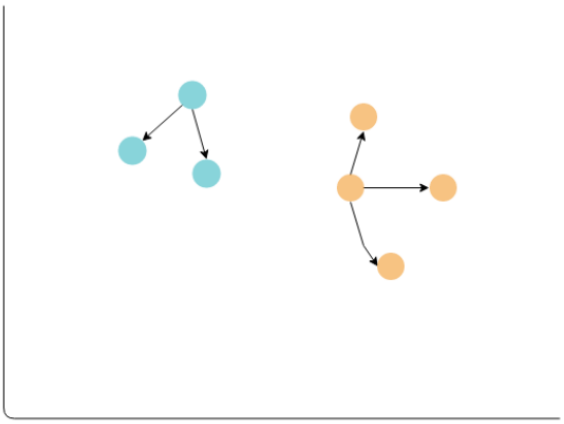
Σχήμα 35. Αναπαράσταση αλγορίθμου Linear Regression



Σχήμα 36. Συχνότητα εμφάνισης λέξεων

12.2 K-Nearest-Neighbor

Ο αλγόριθμος k-nearest-neighbor (KNN) ανήκει στην οικογένεια των μη επιβλεπόμενων αλγορίθμων. Η λειτουργία των KNN είναι σχετικά φτηνή σε κατανάλωση πόρων και εύκολη σε υλοποίηση. Κατά τη διάρκεια της εκμάθησης του μοντέλου τα δεδομένα χωρίζονται σε περιοχές δημιουργώντας ζώνες με συγγενή σε χαρακτηριστικά δεδομένα.



Σχήμα 37. K-Nearest Neighbor

Ανάλογα με το μοντέλο που θέλουμε να δημιουργήσουμε και το πλήθος των δεδομένων που διαθέτουμε ορίζουμε την τιμή που επιθυμούμε στην παράμετρο k. Η τιμή του k καθορίζει το πλήθος των ήδη κατηγοριοποιημένων στοιχείων που θα συγκριθούν με το νεοεισερχόμενο ώστε να τοποθετηθεί αυτό με τη σειρά του στο μοντέλο. Η τιμή του k αλλάζει σημαντικά την ακρίβεια της ταξινόμησης. Πιο συγκεκριμένα: Αν η τιμή του k είναι πολύ μικρή και επομένως το πλήθος των γειτόνων που χρησιμοποιούνται είναι περιορισμένο η απόδοση του k-NN δεν είναι τόσο καλή λόγω των ανεπαρκών trainned δεδομένων. Όσο αυξάνεται το k τόσο η απόδοση βελτιώνεται. Ωστόσο, όταν υπερβούμε κάποιο όριο, το πλήθος των γειτόνων είναι τόσο μεγάλο που κατά τη διάρκεια του υπολογισμού δημιουργείται θόρυβος ο οποίος επηρεάζει την ακρίβεια του μοντέλου.

12.3 Ridge Regression

Ο Ridge Regression είναι μια παραλλαγή του Linear Regression που δημιουργήθηκε ως αποτέλεσμα της ανάγκης για την κατασκευή ενός μοντέλου πρόβλεψης που χρειάζεται να επεξεργαστεί μεγάλο αριθμό μεταβλητών. Όπως έχουμε πει, με την αύξηση των μεταβλητών αυξάνεται και η πολυπλοκότητα και παράλληλα αυξάνονται και τα συνολικά λάθη. Στόχος του Ridge Regression είναι να μεταβάλει αριθμητικά προς τη σωστή κατεύθυνση τους συντελεστές παλινδρόμησης και ταυτόχρονα να μειώσει την απόσταση (error) μεταξύ πρόβλεψης και πραγματικής τιμής. Αρχικά πρέπει να πούμε πως ο Linear Regression συμπεριφέρεται το ίδιο σε όλες τις μεταβλητές, δηλαδή όσο αυξάνεται το πλήθος τους, ταυτόχρονα αυξάνεται και ο συντελεστής κάθε μίας. Όμως ένας υψηλός συντελεστής σημαίνει πως και η μεταβλητή που συνοδεύει είναι σημαντική για την πρόβλεψη, κάτι που δεν είμαστε σίγουροι πως ισχύει πάντα. Την ίδια στιγμή το μοντέλο επηρεάζεται τόσο πολύ (overfitting) από τα δεδομένα εκμάθησης που αδυνατεί να προβλέψει σωστά νέες τιμές, με αποτέλεσμα να τείνει προς το training set. Ο Ridge Regression επιλύει αυτό το πρόβλημα κανονικοποιώντας του συντελεστές παλινδρόμησης ανάλογα με το πόσο υπερβολικό είναι το μοντέλο, με τη χρήση ενός παραμέτρου κανονικοποίησης ‘α’ τον οποίο ορίζουμε εμείς σε τιμές από $0 < \alpha < \infty$.

12.4 Decision Trees

Τα δέντρα αποφάσεων (Conditional decision trees) είναι ένας αλγόριθμος της κατηγορίας των supervised τρόπων μάθησης ο οποίος μπορεί να εκπαιδευτεί ώστε να κάνει προβλέψεις συνεχών ή διακριτών τιμών. Ο τρόπος λειτουργίας του μοντέλου είναι ο εξής: Έστω πως έχουμε κάποια λουλούδια όπως στο γνωστό παράδειγμα με το Dataset Iris,

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
139	6.9	3.1	5.4	2.1	2.0
112	6.8	3.0	5.5	2.1	2.0
9	4.9	3.1	1.5	0.1	0.0
15	5.7	4.4	1.5	0.4	0.0
103	6.3	2.9	5.6	1.8	2.0
92	5.8	2.6	4.0	1.2	1.0
133	6.3	2.8	5.1	1.5	2.0
84	5.4	3.0	4.5	1.5	1.0
69	5.6	2.5	3.9	1.1	1.0
38	4.4	3.0	1.3	0.2	0.0

Σχήμα 38. Iris dataset

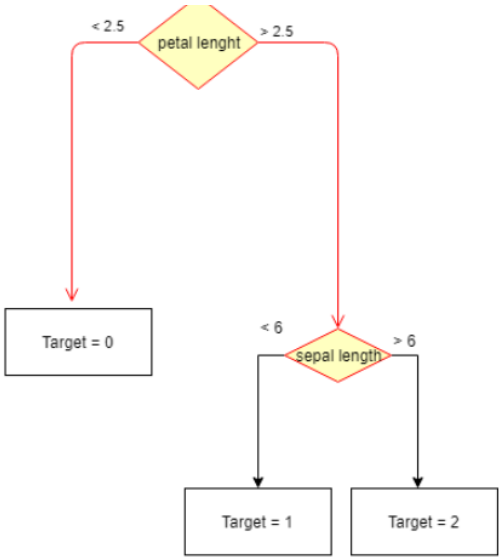
Ο αλγόριθμος ξεκινάει και δημιουργεί υποσύνολα μέσω επαναλήψεων εξετάζοντας τις μεταβλητές κάθε αντικειμένου(π.χ. πλάτος πετάλου) μέχρι τα υποσύνολα να ανήκουν στην ίδια κλάση. Επιλέγει το σημείο διάσπασης υπολογίζοντας με πιο μονοπάτι θα καταλήξει γρηγορότερα σε ‘’καθαρά’’ υποσύνολα.

Επαναλαμβάνεται η ίδια διαδικασία σπάζοντας τα υποσύνολα σε ακόμη μικρότερα υποσύνολα αυτών.

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
92	5.8	2.6	4.0	1.2	1.0
84	5.4	3.0	4.5	1.5	1.0
69	5.6	2.5	3.9	1.1	1.0

Σχήμα 39. Iris dataset υποσύνολα

Μόλις έχουν ολοκληρωθεί οι επαναλήψεις και έχουμε "καθαρά" υποσύνολα το μοντέλο έχει εκπαιδευτεί στο να εντοπίζει τα χαρακτηριστικά που κάνουν κάθε λουλούδι να ανήκει σε μία κατηγορία. Για παράδειγμα, εισάγουμε στο μοντέλο ένα άγνωστο λουλούδι με μήκος ανθού 5cm, πλάτος ανθού 2.4cm, μήκος πετάλου 6cm και πλάτος πετάλου 1cm. Το μοντέλο θα δείξει ότι ανήκει στην κατηγορία με target = 1 αφού όπως φαίνεται σχήμα δεν πληροί τις προϋποθέσεις για την κατηγορία 0 λόγω μεγαλύτερου μήκους πετάλου, ούτε την κατηγορία 2 λόγω μικρότερου μήκους ανθού.



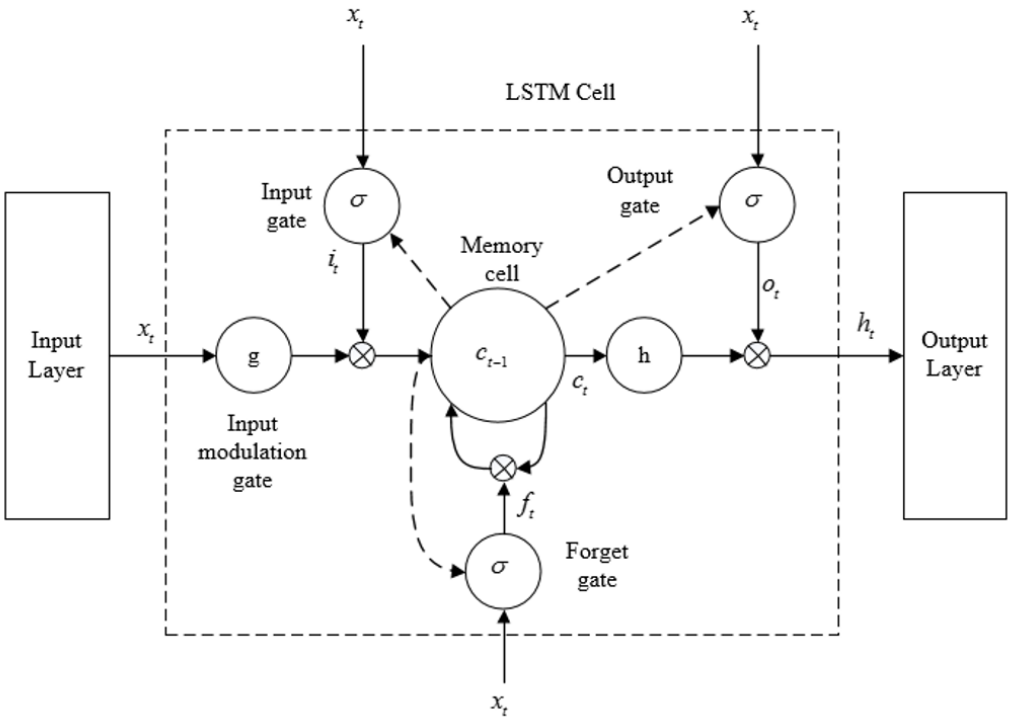
Σχήμα 40. Iris dataset δέντρο απόφασης

Συμπερασματικά, τα δέντρα αποφάσεων μπορούν να χρησιμοποιηθούν σε πληθώρα προβλημάτων πρόβλεψης ανεξάρτητα με το μέγεθος τους ενώ μπορούν να συγκεκριμενοποιηθούν ανάλογα με την περίσταση. Επίσης τα αποτελέσματα τους και ο τρόπος με τον οποίο κατέληξε το μοντέλο σε μια πρόβλεψη μπορούν να παρουσιαστούν γραφικά για ευκολότερη κατανόηση. Στα αρνητικά πρέπει να συμπεριλάβουμε την πιθανότητα overfitting, την αδυναμία παραγωγής ενός αξιόπιστου μοντέλου από μικρό dataset εκμάθησης και την υπερβολική δημιουργία υποσυνόλων κλάσεων λόγω κακής ποιότητας δεδομένων.

12.5 Νευρωνικά δίκτυα LSTM & GRU

Το νευρωνικό δίκτυο LSTM προτάθηκε για πρώτη φορά από τους Sepp Hochreiter και Jürgen Schmidhuber το έτος 1997. Το εν λόγω δίκτυο, έγινε ευρέως γνωστό για την εξαιρετική ικανότητά του να απομνημονεύει μακροπρόθεσμες εξαρτήσεις. Λόγω της αποτελεσματικότητάς τους σε ευρείες πρακτικές εφαρμογές, τα δίκτυα LSTM έχουν λάβει πληθώρα κάλυψης σε επιστημονικά περιοδικά, τεχνικά ιστολόγια και οδηγούς εφαρμογής.Η μακροχρόνια βραχυπρόθεσμη μνήμη («LONG SHORT-TERM MEMORY», εν συντομία «LSTM») είναι μια τεχνητή αρχιτεκτονική επαναλαμβανόμενου νευρωνικού δικτύου («Recurrent Neural Network», εν συντομία «RNN») που χρησιμοποιείται στον τομέα της βαθιάς μάθησης. Σε αντίθεση με τα τυπικά νευρικά δίκτυα feed forward, το δίκτυο LSTM έχει συνδέσεις ανατροφοδότησης. Δεν μπορεί να επεξεργαστεί μόνο μεμονωμένα σημεία δεδομένων (όπως εικόνες), αλλά και ολόκληρες ακολουθίες δεδομένων (όπως ομιλία ή βίντεο). Για παράδειγμα, το LSTM μπορεί να εφαρμοστεί σε εργασίες όπως μη καταχωρισμένη, συνδεδεμένη αναγνώριση γραφής , αναγνώριση ομιλίας και ανίχνευση ανωμαλιών στην κυκλοφορία δικτύου ή IDS (συστήματα εντοπισμού εισβολών).

Η τυπική δομή των κελιών LSTM βρίσκεται στο παρακάτω σχήμα[43].



Σχήμα 41. Δομή των κελιών LSTM

Ένα τυπικό κελί LSTM λοιπόν, όπως διαπιστώνει κανείς παρατηρώντας το ανωτέρω σχήμα, διαμορφώνεται κυρίως από τέσσερις πύλες: πύλη εισόδου (input gate), πύλη διαμόρφωσης εισόδου (input modulation gate), πύλη λήθης (forget gate) και πύλη εξόδου (output gate). Η πύλη εισόδου παίρνει ένα νέο σημείο εισόδου από το εξωτερικό και επεξεργάζεται τα νεοεισαχθέντα δεδομένα. Η πύλη εισόδου κελιού μνήμης λαμβάνει είσοδο από την έξοδο του κελιού LSTM στην τελευταία επανάληψη. Η πύλη λήθης αποφασίζει πότε θα «ξεχάσει» τα αποτελέσματα εξόδου και έτσι επιλέγει το βέλτιστο χρονικό διάστημα για την ακολουθία εισόδου. Η πύλη εξόδου λαμβάνει όλα τα αποτελέσματα που υπολογίζονται και παράγει έξοδο για το κελί LSTM. Σε γλωσσικά μοντέλα, συνήθως προστίθεται ένα στρώμα soft-max για τον προσδιορισμό της τελικής παραγωγής του κελιού LSTM.

Παρακάτω, πρόκειται να αναλυθεί ο τρόπος με τον οποίο μπορεί να χρησιμοποιηθεί ένα δίκτυο LSTM σε ένα μοντέλο πρόβλεψης ροής κυκλοφορίας, προκειμένου να γίνει πιο κατανοητή η λειτουργία του νευρωνικού αυτού δικτύου RNN. Στο εν λόγω μοντέλο εφαρμόζεται ένα στρώμα γραμμικής παλινδρόμησης στο επίπεδο εξόδου του κελιού LSTM.

Ας υποδείξουμε τις χρονοσειρές εισόδου ως $X = (x_1, x_2,\ldots,x_n)$ την κρυφή κατάσταση κελιών μνήμης ως $H = (h_1, h_2,\ldots, h_n)$, τις σειρές χρόνου εξόδου ως $Y = (y_1, y_2,\ldots,y_n)$. Το δίκτυο LSTM κάνει τον υπολογισμό ως εξής:

$$h_t = H(W_{hx}x_t + W_{hh}h_{t-1} + b_h) \quad (1)$$

$$p_t = W_{hy}y_{t-1} + b_y \quad (2)$$

όπου οι πίνακες βάρους υποδηλώνονται ως W, και τα διανύσματα πόλωσης συμβολίζονται ως b. Η κρυφή κατάσταση των κελιών μνήμης υπολογίζεται στους ακόλουθους τύπους:

$$i_t = \sigma\left(W_{ix}x_t + W_{hh}h_{t-1} + W_{ic}c_{t-1} + b_i\right) \quad (3)$$

$$f_t = \sigma\left(W_{fx}x_t + W_{hh}h_{t-1} + W_{fc}c_{t-1} + b_f\right) \quad (4)$$

$$c_t = f_t * c_{t-1} + i_t * g(W_{cx}x_t + W_{hh}h_{t-1} + W_{cc}c_{t-1} + b_c) \quad (5)$$

$$o_t = \sigma\left(W_{ox}x_t + W_{hh}h_{t-1} + W_{oc}c_{t-1} + b_o\right) \quad (6)$$

$$h_t = o_t * h(c_t) \quad (7)$$

Όπου σ σημαίνει την τυπική λειτουργία σιγμοειδούς που ορίζεται στην Εξίσωση 8, * σημαίνει το κλιμακωτό προϊόν δύο διανυσμάτων ή πινάκων, g και h είναι οι επεκτάσεις της λειτουργίας σιγμοειδούς βάσης με το εύρος να αλλάζει σε [-2, 2] και [-1, 1].

$$\sigma(x) = \frac{1}{1+e^x} \qquad (8)$$

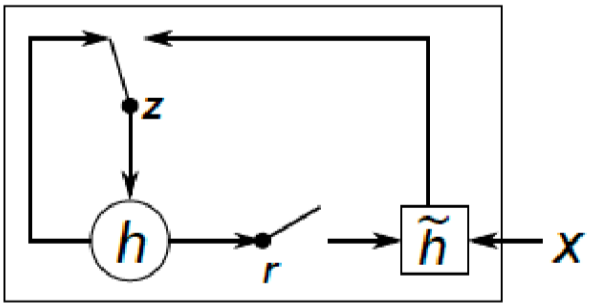
Για αντικειμενική συνάρτηση, χρησιμοποιούμε την συνάρτηση τετραγωνικής απώλειας η οποία αποδίδεται με τον ακόλουθο τύπο:

$$e = \sum_{t=1}^{n} (y_t - p_t)^2 \qquad (9)$$

Το y, αντιπροσωπεύει την πραγματική έξοδο και το p αντιπροσωπεύει την προβλεπόμενη ροή κυκλοφορίας. Προκειμένου να ελαχιστοποιηθεί το σφάλμα εκπαίδευσης και εν τω μεταξύ να αποφευχθούν τοπικά ελάχιστα σημεία, το εργαλείο βελτιστοποίησης Adam, μια τροποποίηση της στοχαστικής βελτίωσης καθόδου κλίσης (SGD) με προσαρμοστικούς ρυθμούς εκμάθησης, εφαρμόζεται για πολλαπλασιασμό πίσω μέσω του χρόνου (BPTT). Τα νευρωνικά δίκτυα είναι γνωστά για τις τεράστιες εκφραστικές τους ικανότητες και είναι ιδιαίτερα εύκολο να ταιριάζουν. Η εκπαίδευση των νευρωνικών δικτύων υπήρξε εδώ και πολύ καιρό ένα δύσκολο πρόβλημα και έχει προταθεί μεγάλη ποσότητα μεθόδων κανονικοποίησης για τη μείωση του υπερβολικού εξοπλισμού. Το 2012, το πρόγραμμα dropout προτάθηκε ως μια πολύ αποτελεσματική μέθοδος εκπαίδευσης νευρωνικών δικτύων προκειμένου να αποκτήσουν καλύτερα χαρακτηριστικά των εικόνων. Ωστόσο, λόγω της επαναλαμβανόμενης ιδιότητας των RNN, το dropout ήταν δύσκολο να εφαρμοστεί σε γλωσσικά μοντέλα RNN. Μέχρι το 2014, οι μέθοδοι dropout έχουν χαρακτηρισθεί ως επιτυχημένοι αναφορικά με την εφαρμογή τους στα νευρωνικά δίκτυα RNN[43].

[43] W. Zaremba, I. Sutskever and O. Vinyals, "Recurrent neural network regularization," arXiv preprint arXiv:1409.2329, 2014.

Όπως προαναφέρθηκε, η LSTM έγινε ευρέως γνωστή για την εξαιρετική ικανότητά της να απομνημονεύει μακροπρόθεσμες εξαρτήσεις. Ωστόσο, λόγω της σύνθετης δομής της, η λύση της διαρκεί συνήθως πολύ χρόνο. Ως εκ τούτου, προκειμένου να επιταχυνθεί η εκπαίδευση, το 2014, προτάθηκε η GRU από τους Cho κ.α.[44], ως τροποποίηση της LSTM για την διενέργεια της αυτόματης μετάφρασης, χάριν της απλούστερης δομής της και της δυνατότητάς της να επιλύει τα προβλήματα που τυχόν εμφανίζονταν κατά την λειτουργία του δικτύου LSTM[45]. Η τυπική δομή των κελιών GRU φαίνεται στο κατωτέρω σχήμα (Σχήμα 2).



Σχήμα 42. Δομή των κελιών GRU

Ένα τυπικό κελί GRU αποτελείται από δύο πύλες: επαναφορά πύλης r και ενημέρωση πύλης z. Παρόμοια με το κελί LSTM, η έξοδος κρυφής κατάστασης στο χρόνο t υπολογίζεται χρησιμοποιώντας την κρυφή κατάσταση του χρόνου t-1 και την τιμή χρονοσειρών εισόδου στο χρόνο t.

$$h_t = f(h_{t-1}, x_t) \qquad (10)$$

Η λειτουργία επαναφοράς των θυρών είναι παρόμοια με τις πύλες λήθης της LSTM. Καθώς τα δίκτυα GRU έχουν πολλές ομοιότητες με τα LSTM, δεν θα διεισδύσουμε περαιτέρω λεπτομερώς στη φόρμουλα. Το μέρος παλινδρόμησης καθώς και η μέθοδος

[44] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, D. Bahdanau, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," arXiv preprint arXiv:1406.1078, 2014
[45] Οπ. παρ. υποσημ. 3

βελτιστοποίησης που χρησιμοποιούμε στην ενυπάρχουσα ερευνητική εργασία για τα δίκτυα GRU είναι ίδια με αυτά των LSTM.

12.6 Αλγόριθμος Min Max Scaler

Πρόκειται για έναν αλγόριθμο ο οποίος μεταμορφώνει ένα dataset διανυσμάτων, ώστε τα δεδομένα να ανήκουν μέσα σε ένα σύνολο τιμών(συχνά[0,1]). Ο αλγόριθμος, λοιπόν, παίρνει ως παραμέτρους τις επιθυμητές τιμές του min και max, υπολογίζει τα συνοπτικά στατιστικά ενός dataset και παράγει ένα μοντέλο Min Max Scaler. Το μοντέλο, έπειτα, είναι σε θέση να μετασχηματίσει κάθε ένα δείγμα ξεχωριστά, ώστε να βρίσκεται μέσα στο επιθυμητό σύνολο τιμών. Η νέα τιμή που ανήκει στην κλίμακα αυτή υπολογίζεται ώς:

$$Rescaled\,(e_i) = \frac{e_i - E_{min}}{E_{max} - E_{min}} * (max - min) + min$$

$$Στην\ περίπτωση\ που\ Emax = Emin: Rescaled\,(e_i) = 0.5 * (max + min)$$

Να τονιστεί ότι εφόσον οι τιμές του αποτελέσματος του αλγορίθμου θα είναι μη-μηδενικές ακόμα και για μηδενικές τιμές, καταλαβαίνει κανείς ότι το αποτέλεσμα θα έχει την μορφή πυκνού διανύσματος ακόμα και για εισόδους που είναι αραιά διανύσματα.

13. Πειραματικά αποτελέσματα

Ύστερα από την παραπάνω ανάλυση σειρά έχει η λειτουργία του μοντέλου μέσω της χρήσης δεδομένων. Οι κατωτέρω πληροφορίες αντλήθηκαν από τον ιστότοπο: "*https://lionbridge.ai/datasets/top-10-stock-market-datasets-for-machine-learning/*" και το dataset που χρησιμοποιήθηκε είναι το: Uniqlo Stock Price Prediction. Το συγκεκριμένο dataset αφορά στοιχεία της εταιρείας Uniqlo. Η Uniqlo αποτελεί μια από τις μεγαλύτερες επιχειρήσεις λιανικής πώλησης ενδυμάτων στην Ιαπωνία και βρίσκεται πάνω από πέντε δεκαετίες στην αγορά. Το συγκεκριμένο σύνολο δεδομένων περιλαμβάνει τις πληροφορίες για τις μετοχές της εταιρείας από το 2012 έως το 2016. Στόχος ήταν η πρόβλεψη της τιμής κλεισίματος της μετοχής.

Όπως έχουμε αναφέρει και από την προηγούμενη μας υλοποίηση το εργαλείο που χρησιμοποιούμε είναι η Python και πιο συγκεκριμένα αυτό του Jupyter.

Τα πακέτα που χρησιμοποιήσαμε είναι πολλά ενδεικτικά χρησιμοποιήθηκαν πακέτα όπως το Numpy, Pandas κ.α (παρατίθενται αναλυτικά παρακάτω).

Πριν την χρήση των δεδομένων θα πρέπει να συλλεχτούν και στη συνέχεια να μορφοποιηθούν κατάλληλα ώστε να είναι δυνατή η επεξεργασία τους μέσω της βιβλιοθήκης Pandas.

Τα δεδομένα μας αποτελούνται από τις ημερήσιες τιμές των 5 ετών της εταιρείας Uniqlo και περιλαμβάνουν τις στήλες:

1. Ημερομηνία
2. Τιμές ανοίγματος
3. Υψηλότερη τιμή
4. Χαμηλότερη τιμή
5. Τιμές κλεισίματος
6. Όγκος

```
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout, GRU, Bidirectional
from keras.optimizers import SGD
import math
from sklearn import metrics
from sklearn.linear_model import LinearRegression
from sklearn.neighbors import KNeighborsRegressor
import seaborn as sns
import keras
from sklearn import linear_model
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from keras.callbacks import EarlyStopping
from keras.utils import to_categorical
import torch
import os
import numpy as np
import pandas as pd
import tqdm as tqdm
import seaborn as sns
from pylab import rcParams
import matplotlib.pyplot as plt
import matplotlib as rc
from sklearn.preprocessing import MinMaxScaler
from pandas.plotting import register_matplotlib_converters
from pandas.plotting import lag_plot
from pandas import datetime
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import r2_score
import mpld3
```

Σχήμα 43. Οι βιβλιοθήκες που χρησιμοποιήθηκαν

Στη συνέχεια με την ενσωμάτωση των βιβλιοθηκών βλέπουμε την χρονική περίοδο μεταφοράς των δεδομένων.

In [2]:
```
Stock_file = "Uniqlo 2012-2016 Completed.csv"

data = pd.read_csv(os.path.join(os.getcwd(), "data", Stock_file))
data.drop(data.columns[[6]], axis=1, inplace=True)##Delete a column
#data.drop(data.columns[[1]], axis=1, inplace=True)##Delete a column
#data.drop(data.columns[[2]], axis=1, inplace=True)##Delete a column

data
```

Out[2]:

	Date	Open	High	Low	Close	Volume
0	2012-01-04	14050	14050	13700	13720	559100
1	2012-01-05	13720	13840	13600	13800	511500
2	2012-01-06	13990	14030	13790	13850	765500
3	2012-01-10	13890	14390	13860	14390	952300
4	2012-01-11	14360	14750	14280	14590	1043400
...	...	...	...	...	...	...
1228	2017-01-06	40500	41030	39720	39720	1435500
1229	2017-01-10	38620	38850	38150	38690	1196900
1230	2017-01-11	38710	38880	38480	38560	545900
1231	2017-01-12	38300	38450	37930	38010	800900
1232	2017-01-13	38900	39380	38240	38430	1321200

1233 rows × 6 columns

Σχήμα 44. Το σύνολο δεδομένων μας 2012-2017

Από την παραπάνω εικόνα βλέπουμε ότι παράγεται ένας πίνακας τιμών μετοχών ανά ημέρα. Οι τιμές που περιλαμβάνει είναι αυτές του ανοίγματος, κλεισίματος, υψηλότερης και χαμηλότερης τιμής καθώς και την τιμή κλεισίματος των αποθεμάτων της μετοχής προσαρμοσμένη στην τιμή της μετοχής για όλες τις εταιρικές πράξεις. Το παραπάνω περιλαμβάνεται επειδή οι τιμές των μετοχών όπως είναι γνωστό καθορίζονται από τους εμπόρους ενώ τα αποθέματα μετοχών και τα μερίσματα επηρεάζουν την τιμή της μετοχής και πρέπει να συνυπολογίζονται.

Στη συνέχεια κάνουμε έναν έλεγχο για να δούμε αν λείπει κάποια τιμή από οποιαδήποτε στήλη του πίνακά μας. Ενώ παρουσιάζουμε τις πέντε πρώτες γραμμές των δεδομένων μας.

```
(data==0).all()
```

```
Date       False
Open       False
High       False
Low        False
Close      False
Volume     False
dtype: bool
```

```
data.head()
```

	Date	Open	High	Low	Close	Volume
0	2012-01-04	14050	14050	13700	13720	559100
1	2012-01-05	13720	13840	13600	13800	511500
2	2012-01-06	13990	14030	13790	13850	765500
3	2012-01-10	13890	14390	13860	14390	952300
4	2012-01-11	14360	14750	14280	14590	1043400

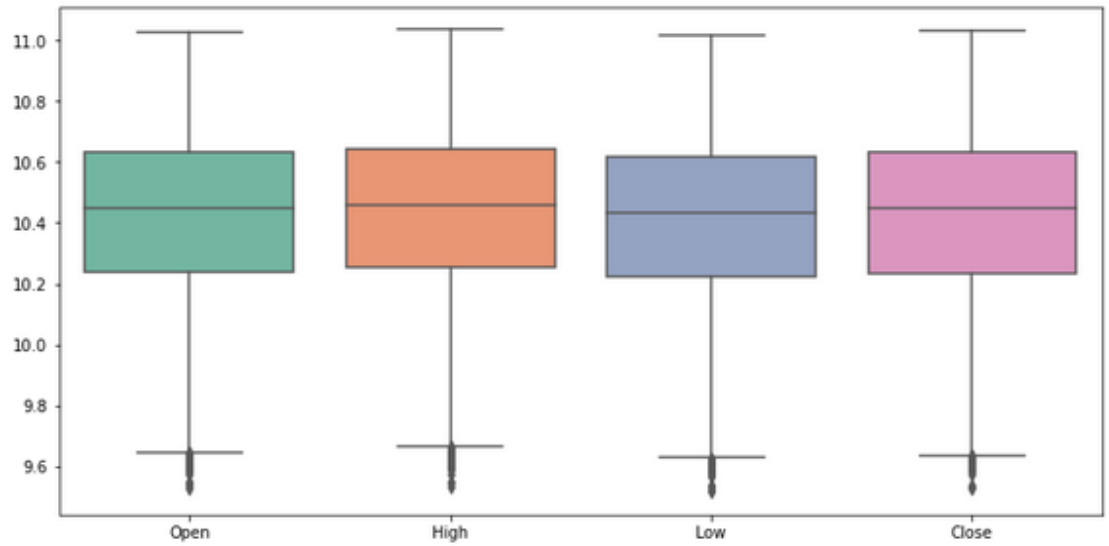
Σχήμα 45. Ελλείπεις τιμές και η απεικόνιση των αρχικών χρονικά δεδομένων

Παρακάτω ελέγχουμε αν υπάρχει κάποια ασυνεπείς τιμή (Outlier), ελλιπείς τιμή, η θόρυβος στα δεδομένα μας η οποία μπορεί να επηρεάσει τα τελικά αποτελέσματά μας. Αφού βεβαιωθήκαμε ότι δεν υπάρχει κάναμε το πρώτο μας διάγραμμα που δείχνει τις ακραίες τιμές των μετοχών.

```
data.isna().sum()
```

```
Date      0
Open      0
High      0
Low       0
Close     0
Volume    0
dtype: int64
```

```
#Akraies times
Scaled_data= np.log(data[['Open','High', 'Low', 'Close']])
plt.figure(figsize = (12,6))
_ = sns.boxplot(data=Scaled_data, palette="Set2")
```



Σχήμα 46. Ελλείπεις τιμές και ακραίες τιμές μετοχών

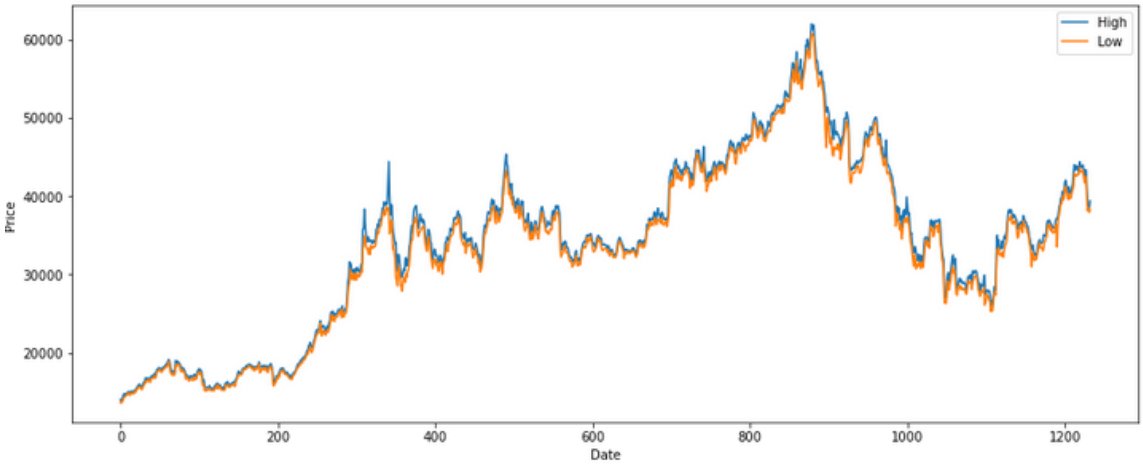
Προχωρώντας τη ροή της ανάλυσης μας παρουσιάζονται συγκεντρωτικά στοιχεία που επεξηγούν τις στήλες του dataset συγκεντρωτικά. Αυτό είναι μια πρώτη εικόνα για να δούμε που είναι τα δεδομένα συγκεντρωμένα, την υψηλότερη χαμηλότερη τιμή, τη μέση τιμή κάθε στήλης κ.α

```
data.describe()
```

	Open	High	Low	Close	Volume
count	1233.000000	1233.000000	1233.000000	1233.000000	1.233000e+03
mean	33790.490673	34214.468775	33383.592863	33795.429846	7.286687e+05
std	10794.171777	10916.449594	10676.623182	10795.797988	4.136818e+05
min	13720.000000	13840.000000	13600.000000	13720.000000	1.391000e+05
25%	27940.000000	28330.000000	27480.000000	27800.000000	4.897000e+05
50%	34500.000000	34895.000000	33950.000000	34480.000000	6.261000e+05
75%	41450.000000	41900.000000	40830.000000	41370.000000	8.270000e+05
max	61550.000000	61970.000000	60740.000000	61930.000000	4.937300e+06

Σχήμα 47. Πληροφορίες του Dataset

Το παρακάτω είναι ένα διάγραμμα που απεικονίζει τις τιμές μεγαλύτερης- μικρότερης τιμής της μετοχής.



Σχήμα 48. Πληροφορίες του Dataset

Αφού έγινε η πρώτη απεικόνιση των δεδομένων μας πρέπει να γίνει η επεξεργασία τους και ο διαχωρισμός τους σε δεδομένα Train και Test. Ο διαχωρισμός τους έγινε σε ποσοστό 80% για το κομμάτι των δεδομένων εκπαίδευσης και 20% για το κομμάτι των δεδομένων ελέγχου.

Παρακάτω φαίνεται ο διαχωρισμός των δύο ο οποίος είναι χρονικά και όχι τυχαία.

```
#spaei to Dataset se train kai test
def split_data(dataframe, Threshold, Feat):
    TrainData=dataframe.loc[:Threshold,Feat]
    TestData=dataframe.loc[Threshold:,Feat]
    return(TrainData,TestData)
PCT=0.8
TrainData,TestData=split_data(data,(int(len(data)*PCT)),["Open","Close"])
```

```
TrainData.head()
```

	Open	Close
0	14050	13720
1	13720	13800
2	13990	13850
3	13890	14390
4	14360	14590

```
TrainData.tail()
```

	Open	Close
982	40360	40430
983	40320	40170
984	40000	39050
985	36700	38140
986	37210	37630

Σχήμα 49. Δεδομένα εκπαίδευσης

```
TestData.head()
```

	Open	Close
986	37210	37630
987	37840	38690
988	37290	37640
989	38120	37000
990	36370	36730

```
TestData.tail()
```

	Open	Close
1228	40500	39720
1229	38620	38690
1230	38710	38560
1231	38300	38010
1232	38900	38430

Σχήμα 50. Δεδομένα ελέγχου

Το μήκος του dataset είναι 1233 εγγραφές. Με τον διαχωρισμό του Dataset σε ποσοστό 80% – 20% τα δεδομένων εκπαίδευσης έγιναν 987 και τα δεδομένων ελέγχου έγιναν 247

```
#plhrofories gia to mhkos tou dataset (Train - Test)
print("Len data is: ", len(data),", len Training set is: ", len(TrainData) ,", len Test set is: ",  len(TestData))
```

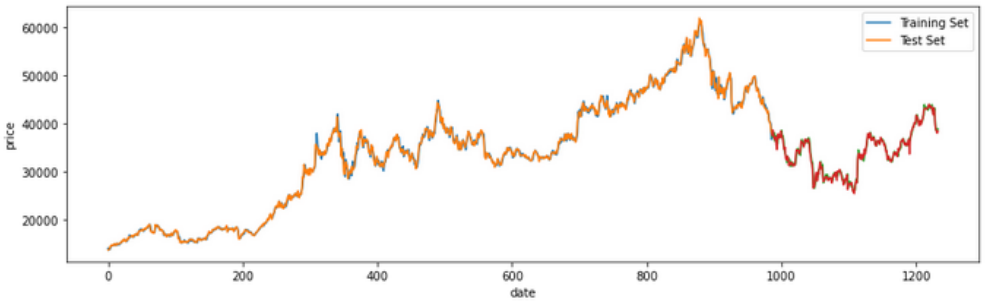
```
Len data is:  1233 , len Training set is:  987 , len Test set is:  247
```

Σχήμα 51. Πληροφορίες Train & Test Data

Επόμενο μας βήμα είναι να γίνει μια γραφική απεικόνιση του παραπάνω.

```
plt.figure(figsize=(14,4))
plt.plot(TrainData)
plt.plot(TestData)
plt.ylabel("price")
plt.xlabel("date")
plt.legend(["Training Set", "Test Set"])
```

```
<matplotlib.legend.Legend at 0x2b5a6c65988>
```



Σχήμα 52. Γραφική απεικόνιση Train & Test Data

Ένα από τα σημαντικότερα βήματα της διαδικασία είναι η κανονικοποίηση των δεδομένων. Αφού προχωρήσαμε σε κανονικοποίηση των δεδομένων και με χρονοσήμανση (Timestamp =5) μας οδηγεί σε να χρησιμοποιούμε τα δεδομένα 5 ημερών πίσω, άρα 10 τιμές άρα 2 δεδομένα επι 5 τιμές πίσω ίσον 10. (2 Features * 5 timestamp).

```
#Data Normalization
SC= MinMaxScaler(feature_range=(0,1))

SC_train= SC.fit_transform(np.array(TrainData))
SC_test= SC.fit_transform(np.array(TestData))

#Gia na apofeux8ei h uperprosarmogh
YSC_train=SC.fit_transform(np.array(TrainData.drop(['Open'],axis=1)))
YSC_test= SC.fit_transform(np.array(TestData.drop(['Open'],axis=1)))
```

```
#Dimiourgoume akolou8ies dedomenwn pou lamvanontai apo to df.Open & df.Close

X_train = []
y_train = []

lengthTr = len(TrainData)
timeStamp=5
for i in range(timeStamp,lengthTr):
    X_train.append(SC_train[i-timeStamp:i])
    y_train.append(YSC_train[i])

X_train=np.array(X_train)
y_train=np.array(y_train)


X_test = []
y_test = []

lengthTe = len(TestData)
for i in range(timeStamp, lengthTe):
    X_test.append(SC_test[i-timeStamp:i])
    y_test.append(YSC_test[i])

X_test=np.array(X_test)
y_test=np.array(y_test)
```

Σχήμα 53. Κανονικοποίηση δεδομένων

Δημιουργείται ένας πίνακας με όνομα Train data όπου έχει μήκος 987 (Γραμμές) και πλάτος 2 (αριθμός στηλών) το ημερήσιο άνοιγμα και κλείσιμο της μετοχής. Οι 987 γραμμές του πίνακα Train Data και οι 247 του πίνακα Test Data αντιστοιχούν στις πρώτες 987 γραμμές του πίνακα Data και στις επόμενες 247 του ιδίου πίνακα. Αντίστοιχα δημιουργείται ένας πίνακας Test Data μήκους 247 x 2.

Στη συνέχεια θα δημιουργήσουμε δύο πίνακες Scale Train και Scale Test οι οποίοι θα περιέχουν τις κανονικοποιημένες τιμές Open και Close των δύο πινάκων Train Data και Test DataO πίνακας Scale Train αποτελεί ένα πίνακα μεγέθους 987 * 2 όπου και σημαίνει 987 το πλήθος των εγγραφών που είναι το 80% από το διαχωρισμό που έγινε νωρίτερα και οι 2

στήλες είναι οι Open & Close. Στην πρώτη ημέρα είναι η ημέρα 0 για εμάς ενώ στην τελευταία γραμμή βλέπουμε την πρόβλεψη ανοίγματος και κλεισίματος για την επόμενη ημέρα.

```
SC_train
```

```
array([[0.00689944, 0.        ],
       [0.        , 0.00165941],
       [0.00564499, 0.00269654],
       ...,
       [0.54944595, 0.52540967],
       [0.4804516 , 0.50653391],
       [0.49111436, 0.4959552 ]])
```

```
len(SC_train)
```

```
987
```

Σχήμα 54. Κανονικοποιημένα δεδομένα εκπαίδευσης

Ο πίνακας Scale Test αποτελεί ένα πίνακα μεγέθους 247 * 2 όπου και σημαίνει 247 το πλήθος των εγγραφών που είναι το 20% από το διαχωρισμό που έγινε νωρίτερα και οι 2 στήλες είναι οι Open & Close. Εδώ έχουμε το καθημερινό άνοιγμα και το κλείσιμο και έχει και σχέση με τον πίνακα X test ο οποίος μέσω του Timestamp μας δίνει τα ανοίγματα και κλεισίματα ανά μπλοκ δηλαδή το πρώτο μπλοκ αποτελεί τις μέρες 0-1-2-3-4. Στο 2ο μπλοκ προστίθεται ένα άνοιγμα και ένα κλείσιμο τη φορά.

```
In [21]: SC_test
```

```
Out[21]: array([[0.62451253, 0.65669631],
                [0.65961003, 0.71382377],
                [0.62896936, 0.65723525],
                [0.67520891, 0.6227432 ],
                [0.57771588, 0.60819186],
                [0.59108635, 0.63459984],
                [0.63899721, 0.54998653],
                [0.54651811, 0.49393694],
                [0.54038997, 0.6453786 ],
                [0.64902507, 0.66963083],
                [0.61281337, 0.62220426],
                [0.62896936, 0.65561843],
                [0.6178273 , 0.62921046],
                [0.63064067, 0.69981137],
                [0.70250696, 0.65885206],
                [0.63342618, 0.63567771],
                [0.63899721, 0.60226354],
                [0.58495822, 0.53651307],
                [0.48746518, 0.49285907],
```

```
In [22]: len(SC_test)
```

```
Out[22]: 247
```

Σχήμα 55. Κανονικοποιημένα δεδομένα ελέγχου

Δημιουργούμε έναν καινούριο πίνακα με όνομα X_train ο οποίος είναι τρισδιάστατος πίνακας η πρώτη διάσταση είναι 982 η δεύτερη διάσταση είναι 5 και η τρίτη διάσταση είναι δύο. Η κάθε μια από τις 982 εγγραφές οι οποίες αντιστοιχούν στις εγγραφές 5 – 987 εγγραφές του πίνακα SC Train περιέχει τις τιμές του ανοίγματος και κλεισίματος (2) των πέντε προηγουμένων ημερών (timestamp = 5).

```
X_train
```

```
array([[[0.00689944, 0.        ],
        [0.        , 0.00165941],
        [0.00564499, 0.00269654],
        [0.00355425, 0.01389753],
        [0.01338072, 0.01804605]],

       [[0.        , 0.00165941],
        [0.00564499, 0.00269654],
        [0.00355425, 0.01389753],
        [0.01338072, 0.01804605],
        [0.0167259 , 0.01659407]],

       [[0.00564499, 0.00269654],
        [0.00355425, 0.01389753],
        [0.01338072, 0.01804605],
        [0.0167259 , 0.01659407],
        [0.02132553, 0.02177971]],

       ...,

       [[0.59627849, 0.5899191 ],
        [0.59753293, 0.59987554],
        [0.59126072, 0.5585978 ],
        [0.55697261, 0.55403443],
        [0.55613632, 0.54864136]],

       [[0.59753293, 0.59987554],
        [0.59126072, 0.5585978 ],
        [0.55697261, 0.55403443],
        [0.55613632, 0.54864136],
        [0.54944595, 0.52540967]],

       [[0.59126072, 0.5585978 ],
        [0.55697261, 0.55403443],
        [0.55613632, 0.54864136],
        [0.54944595, 0.52540967],
        [0.4804516 , 0.50653391]]])
```

```
X_train.shape
```

```
(982, 5, 2)
```

```
len(X_train)
```

```
982
```

Σχήμα 56. Τρισδιάστατος πίνακας X_train και το μήκος του

Δημιουργούμε έναν καινούριο πίνακα με όνομα y_test ο οποίος είναι διδιάστατος πίνακας η πρώτη διάσταση είναι 242 η δεύτερη διάσταση είναι ένα. Η κάθε μια από τις 242 εγγραφές περιέχουν τις τιμές του ανοίγματος και κλεισίματος ανά ημέρα.

```
y_test.shape
```

```
(242, 1)
```

```
y_test
```

```
array([[0.63459984],
       [0.54998653],
       [0.49393694],
       [0.6453786 ],
       [0.66963083],
       [0.62220426],
       [0.65561843],
       [0.62921046],
       [0.69981137],
       [0.65885206],
       [0.63567771],
       [0.60226354],
       [0.53651307],
       [0.49285907],
       [0.51872811],
       [0.40016168],
       [0.38129884],
       [0.37429264],
       [0.41902452],
```

Σχήμα 57. Διδιάστατος y_test και το μήκος του

Στη συνέχεια χρησιμοποιώντας τη βιβλιοθήκη Keras δημιουργούμε ένα σειριακό μοντέλο με το μοντέλο Long Short Term Memory (LSTM).

```
# The LSTM (Long Short Term Memory) architecture
model = Sequential([
                    keras.layers.LSTM(units=50, return_sequences=True, input_shape=(X_train.shape[1],2)),
                    Dropout(0.5),

                    keras.layers.LSTM(units=40, return_sequences=True),
                    Dropout(0.5),

                    keras.layers.LSTM(units=40, return_sequences=True),
                    Dropout(0.5),

                    keras.layers.LSTM(units=40),
                    Dropout(0.5),

                    keras.layers.Dense(units=1),
                   ])
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 5, 50)             10600
_________________________________________________________________
dropout_1 (Dropout)          (None, 5, 50)             0
_________________________________________________________________
lstm_2 (LSTM)                (None, 5, 40)             14560
_________________________________________________________________
dropout_2 (Dropout)          (None, 5, 40)             0
_________________________________________________________________
lstm_3 (LSTM)                (None, 5, 40)             12960
_________________________________________________________________
dropout_3 (Dropout)          (None, 5, 40)             0
_________________________________________________________________
lstm_4 (LSTM)                (None, 40)                12960
_________________________________________________________________
dropout_4 (Dropout)          (None, 40)                0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 41
=================================================================
Total params: 51,121
Trainable params: 51,121
Non-trainable params: 0
_________________________________________________________________
```

Σχήμα 58. Long Short Term Memory αρχιτεκτονική

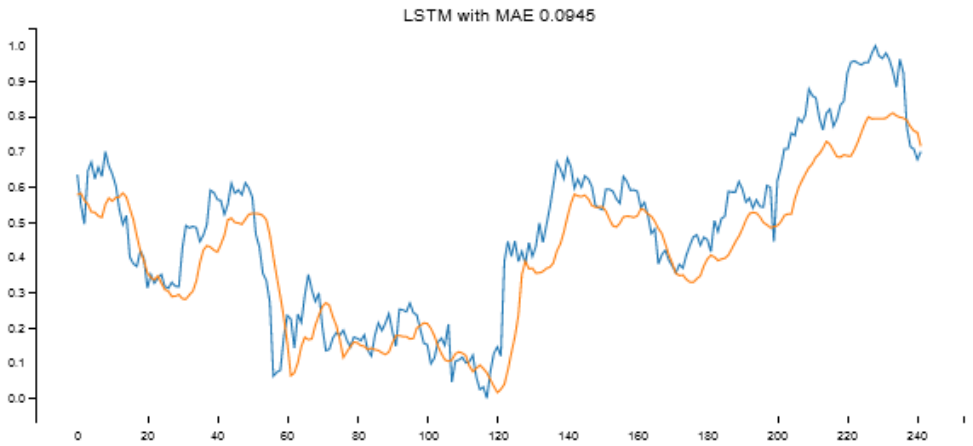
Σε συνέχεια του παραπάνω έχουμε την εκμάθηση του μοντέλου για το Training set γίνεται τόσο με την χρήση του Mean Absolute Error όσο με την χρήση του Mean Squared Error και του Root Mean Square Error.Βλέπουμε ανάλογα με τις “Εποχές” και το ποσοστό σφάλματος. Οι εποχές ανέρχονται στις 50.

Για το νευρωνικό δίκτυο LSTM, το εύρος των τιμών του μέσου όρου του μέσου τετραγωνικού σφάλματος (MSE), για την μετοχή μας απεικονίζεται στα παρακάτω διαγράμματα. Τα αποτελέσματα παρουσιάζονται με τέσσερα δεκαδικά ψηφία. Το μοντέλο δίνει μέση τιμή MSE για την μετοχή Uniqlo για timestamp ίσο με 1 να είναι ίσο με 0,0042 μονάδες, για timestamp ίσο με 2 είναι 0.0057 μονάδες, για timestamp ίσο με 3 είναι ίσο με 0.0063 μονάδες, για timestamp ίσο με 4 είναι 0.0107 μονάδες ενώ τέλος για Timestamp=5 είναι 0.0131 μονάδες.

Επίσης, το εύρος των τιμών του μέσου τετραγωνικού σφάλματος (RMSE), για την μετοχή μας απεικονίζεται στα παρακάτω διαγράμματα. Τα αποτελέσματα παρουσιάζονται με τέσσερα δεκαδικά ψηφία. Το μοντέλο δίνει μέση τιμή RMSE για την μετοχή Uniqlo για timestamp ίσο με 1 να είναι ίσο με 0.0620 μονάδες, για timestamp ίσο με 2 είναι 0.0772 μονάδες, για timestamp ίσο με 3 είναι ίσο με 0.0789 μονάδες, για timestamp ίσο με 4 είναι 0.0930 μονάδες ενώ τέλος για Timestamp=5 είναι 0.0958 μονάδες.

Τέλος το εύρος των τιμών του μέσου όρου (MAE), για την μετοχή μας απεικονίζεται στα παρακάτω διαγράμματα. Τα αποτελέσματα παρουσιάζονται με τέσσερα δεκαδικά ψηφία. Το μοντέλο δίνει μέση τιμή RMSE για την μετοχή Uniqlo για timestamp ίσο με 1 να είναι ίσο με 0.0441 μονάδες, για timestamp ίσο με 2 είναι 0.0605 μονάδες, για timestamp ίσο με 3 είναι ίσο με 0.0683 μονάδες, για timestamp ίσο με 4 είναι 0.0693 μονάδες ενώ τέλος για Timestamp=5 είναι 0.0796 μονάδες.

Παρουσιάζεται και η διαγραμματική απεικόνιση για το Mean Absolute Error.

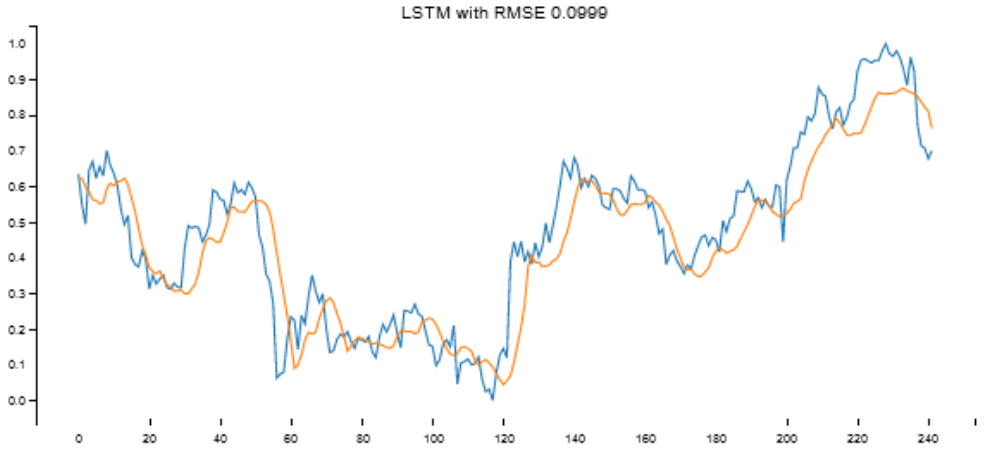


Σχήμα 59. LSTM-MAE

Παρουσιάζεται και η διαγραμματική απεικόνιση του Mean Squared Error

Σχήμα 60. LSTM-MSE

Παρουσιάζεται και η διαγραμματική απεικόνιση του Root Mean Squared Error



Σχήμα 61. LSTM-RMSE

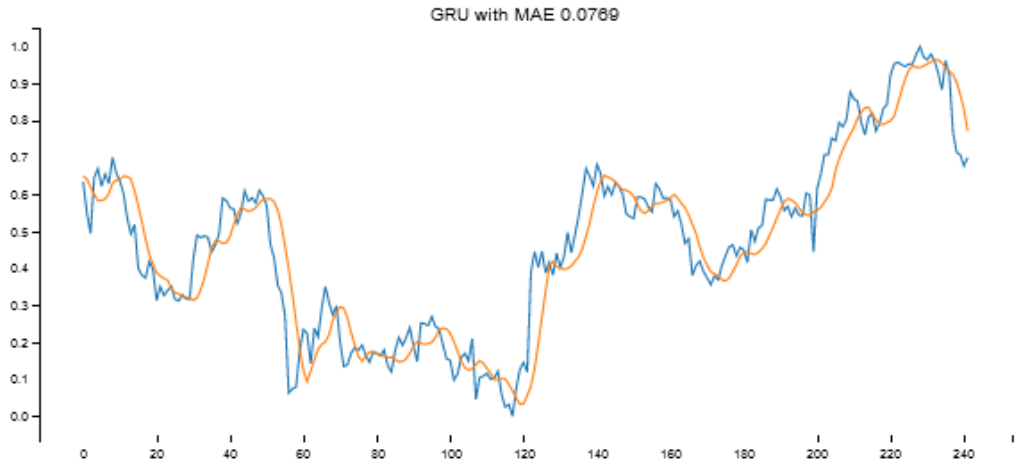
Αντίστοιχη επεξεργασία γίνεται και με την χρήση του νευρωνικού δικτύου GRU τόσο για το Mean Absolute Error όσο και για το Mean Squared Error και το Root Mean Square Error.

Για το νευρωνικό δίκτυο GRU, το εύρος των τιμών του μέσου όρου του μέσου τετραγωνικού σφάλματος (MSE), για την μετοχή μας απεικονίζεται στα παρακάτω διαγράμματα. Τα

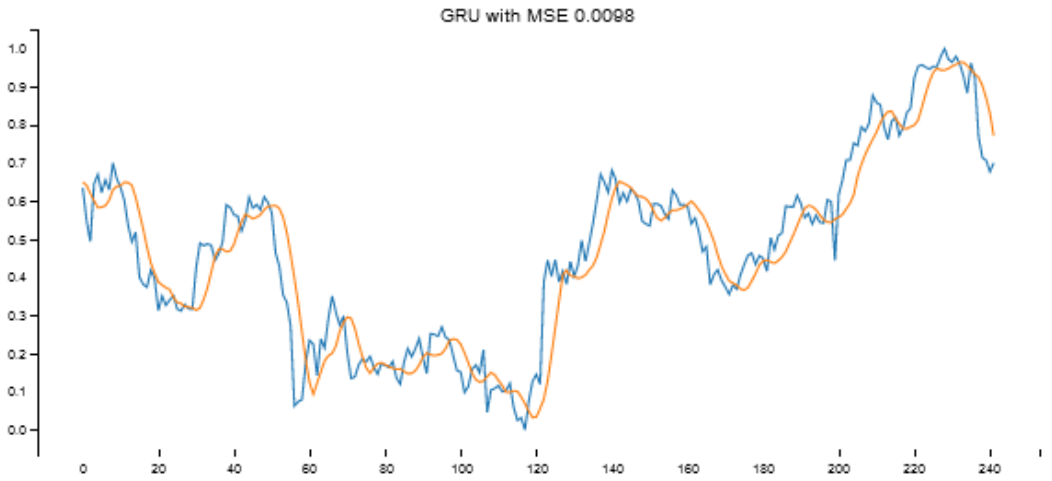
αποτελέσματα παρουσιάζονται με τέσσερα δεκαδικά ψηφία. Το μοντέλο δίνει μέση τιμή MSE για την μετοχή Uniqlo για timestamp ίσο με 1 να είναι ίσο με 0.0038 μονάδες, για timestamp ίσο με 2 είναι 0.0052 μονάδες, για timestamp ίσο με 3 είναι ίσο με 0,0066 μονάδες, για timestamp ίσο με 4 είναι 0,0079 μονάδες ενώ τέλος για Timestamp=5 είναι 0,0092 μονάδες.

Επίσης, το εύρος των τιμών του μέσου τετραγωνικού σφάλματος (RMSE), για την μετοχή μας απεικονίζεται στα παρακάτω διαγράμματα. Τα αποτελέσματα παρουσιάζονται με τέσσερα δεκαδικά ψηφία. Το μοντέλο δίνει μέση τιμή RMSE για την μετοχή Uniqlo για timestamp ίσο με 1 να είναι ίσο με 0.0613 μονάδες, για timestamp ίσο με 2 είναι 0.0722 μονάδες, για timestamp ίσο με 3 είναι ίσο με 0.0812 μονάδες, για timestamp ίσο με 4 είναι 0.0891 μονάδες ενώ τέλος για Timestamp=5 είναι 0.0958 μονάδες.

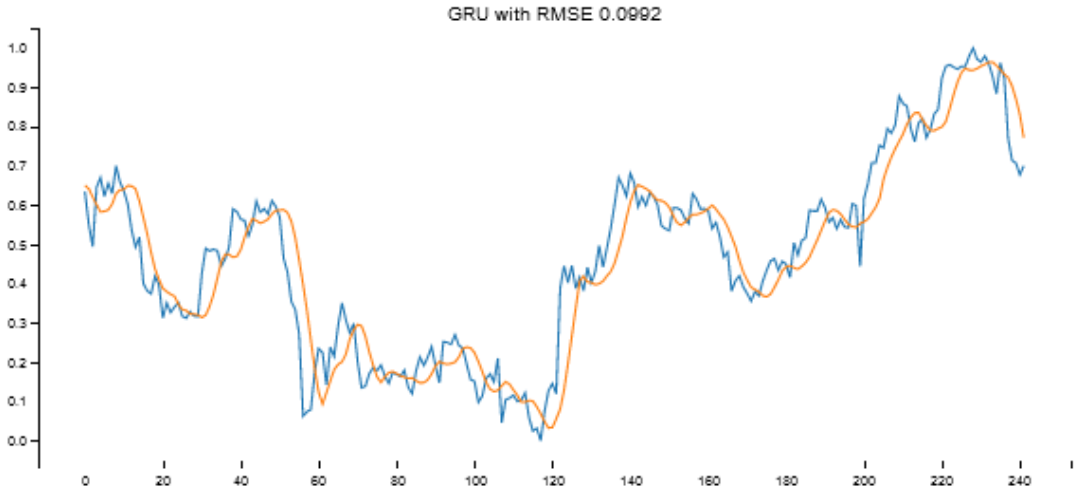
Τέλος το εύρος των τιμών του μέσου όρου (MAE), για την μετοχή μας απεικονίζεται στα παρακάτω διαγράμματα. Τα αποτελέσματα παρουσιάζονται με τέσσερα δεκαδικά ψηφία. Το μοντέλο δίνει μέση τιμή RMSE για την μετοχή Uniqlo για timestamp ίσο με 1 να είναι ίσο με 0.0461 μονάδες, για timestamp ίσο με 2 είναι 0.0554 μονάδες, για timestamp ίσο με 3 είναι ίσο με 0.0633 μονάδες, για timestamp ίσο με 4 είναι 0.0681 μονάδες ενώ τέλος για Timestamp=5 είναι 0.0719 μονάδες.



Σχήμα 62. GRU-MAE



Σχήμα 63. GRU-MSE



Σχήμα 64. GRU-RMSE

Σε όλα αυτά τα παραπάνω διαγράμματα από τα δύο αυτά νευρωνικά δίκτυα βλέπουμε το ποσοστό σφάλματος.

Παρατίθενται οι συγκεντρωτικοί πίνακες από τα παραπάνω μοντέλα με υπολογισμό του r2_score για τα αντίστοιχα ποσοστά.

	LSTM model				
	Timestamp 1	Timestamp 2	Timestamp 3	Timestamp 4	Timestamp 5
MAE	0.9496	0.9191	0.8914	0.8593	0.8306
MSE	0.9484	0.9048	0.8931	0.8631	0.8365
RMSE	0.9396	0.9216	0.8971	0.8650	0.8410

Σχήμα 64. Συγκεντρωτικός πίνακας LSTM με υπολογισμό r2_score

	GRU model				
	Timestamp 1	Timestamp 2	Timestamp 3	Timestamp 4	Timestamp 5
MAE	0.9396	0.9216	0.8971	0.8650	0.8410
MSE	0.9396	0.9216	0.8971	0.8650	0.8410
RMSE	0.9396	0.9216	0.8971	0.8650	0.8410

Σχήμα 65. Συγκεντρωτικός πίνακας GRU

Το συμπέρασμα από τους παραπάνω δύο πίνακες είναι ότι όσο το timestamp ανεβαίνει τόσο ανεβαίνουν και τα σφάλματα, αυτό έχει ως αποτέλεσμα το μικρότερο σφάλμα στα δύο μοντέλα να ανήκει στο timestamp ίσο με ένα (1).

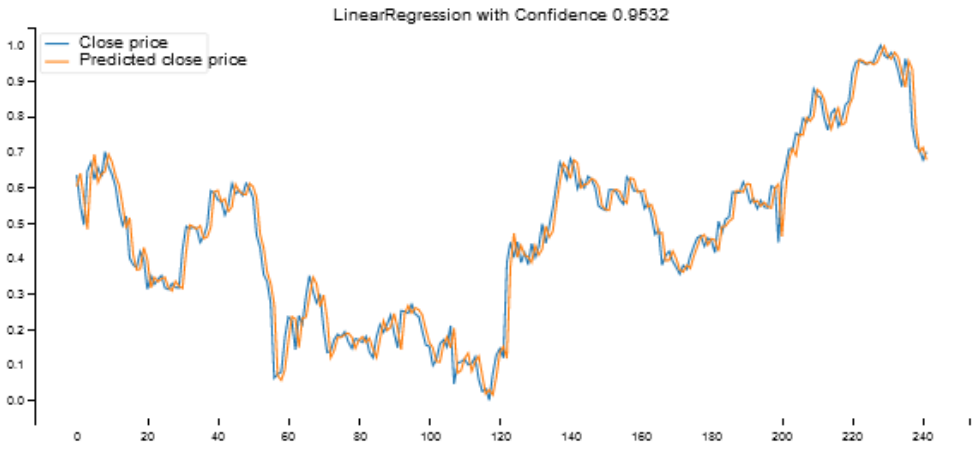
Στη συνέχεια τα παρουσιάζονται τα 4 μοντέλα τα οποία χρησιμοποιούμε για την πρόβλεψη της τιμής κλεισίματος. Τα τέσσερα αυτά μοντέλα είναι :

1. Linear regression
2. KNeighbors
3. BayesianRidge
4. DecisionTree

Στη συνέχεια παρουσιάζεται διαγραμματικά το μοντέλο το οποίο είναι σε θέση να προβλέψει τις τιμές με μεγαλύτερη ακρίβεια για timestamp = 5.

```
mpld3.enable_notebook()
model=LinearRegression(n_jobs=-2)
name="LinearRegression"
model.fit(d2_X_train, y_train)
confidence = model.score(d2_X_test, y_test)
Close= (y_test.reshape(-1,1))
pred_close = (model.predict(d2_X_test))
plt.figure(figsize=(14,6))
plt.title("{} with Confidence {:10.4f}".format(name, confidence))
plt.plot(Close)
plt.plot(pred_close)
plt.legend(["Close price", "Predicted close price"])
```

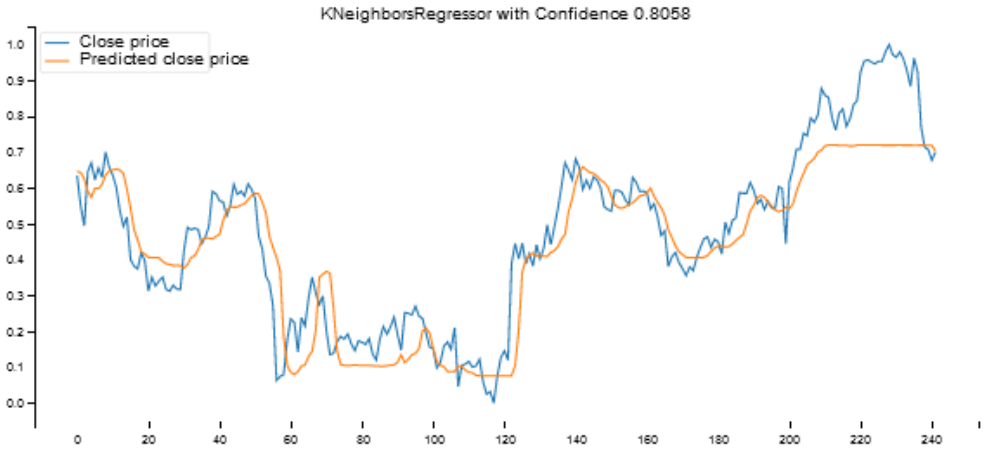
```
<matplotlib.legend.Legend at 0x1a3ce50ccc8>
```



Σχήμα 66. Linear regression

```
mpld3.enable_notebook()
model= KNeighborsRegressor(n_neighbors=250)
name="KNeighborsRegressor"
model.fit(d2_X_train, y_train)
confidence = model.score(d2_X_test, y_test)
Close= (y_test.reshape(-1,1))
pred_close = (model.predict(d2_X_test))
plt.figure(figsize=(14,6))
plt.title("{} with Confidence {:10.4f}".format(name, confidence))
plt.plot(Close)
plt.plot(pred_close)
plt.legend(["Close price", "Predicted close price"])
```

```
<matplotlib.legend.Legend at 0x1a3ce587d88>
```

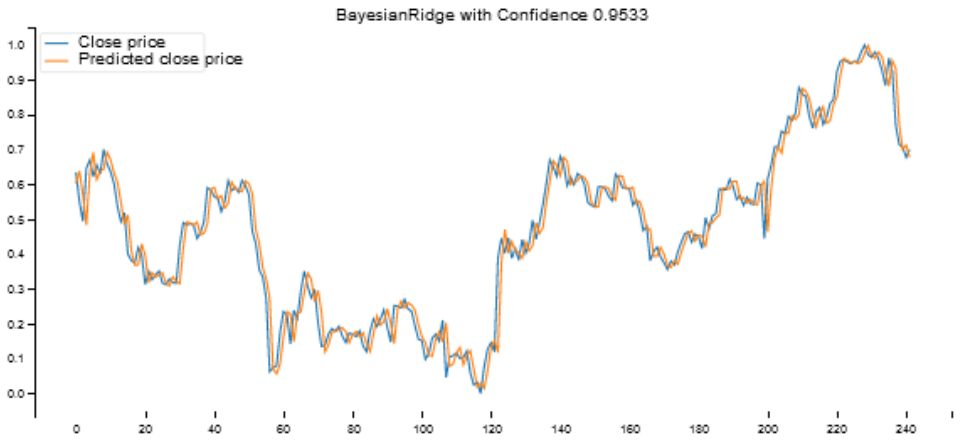


Σχήμα 67. KNeighboors regression

```
mpld3.enable_notebook()
model= linear_model.BayesianRidge()
name="BayesianRidge"
model.fit(d2_X_train, y_train)
confidence = model.score(d2_X_test, y_test)
Close= (y_test.reshape(-1,1))
pred_close = (model.predict(d2_X_test))
plt.figure(figsize=(14,6))
plt.title("{} with Confidence {:10.4f}".format(name, confidence))
plt.plot(Close)
plt.plot(pred_close)
plt.legend(["Close price", "Predicted close price"])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:760: DataConversionWarning: A column-vector
y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel
().
  y = column_or_1d(y, warn=True)
```

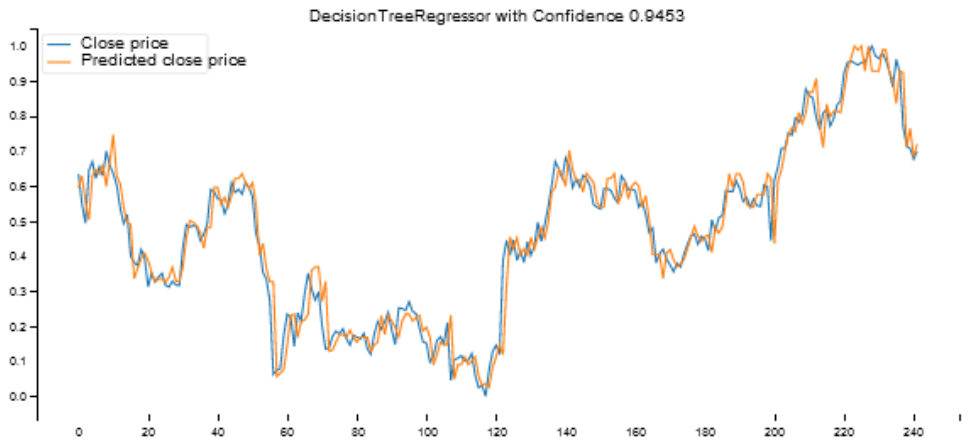
```
<matplotlib.legend.Legend at 0x1a3ce5fe8c8>
```



Σχήμα 68. Ridge regression

```
mpld3.enable_notebook()
model=DecisionTreeRegressor(random_state=5)
name="DecisionTreeRegressor"
model.fit(d2_X_train, y_train)
confidence = model.score(d2_X_test, y_test)
Close= (y_test.reshape(-1,1))
pred_close = (model.predict(d2_X_test))
plt.figure(figsize=(14,6))
plt.title("{} with Confidence {:10.4f}".format(name, confidence))
plt.plot(Close)
plt.plot(pred_close)
plt.legend(["Close price", "Predicted close price"])
```

```
<matplotlib.legend.Legend at 0x1a3ce827e48>
```

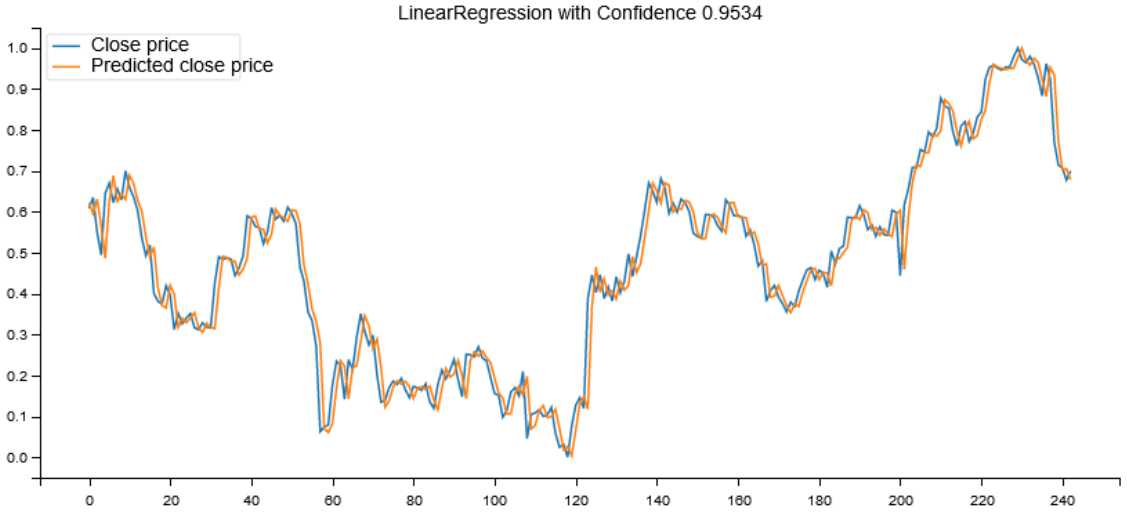


Σχήμα 69. Decision Tree regression

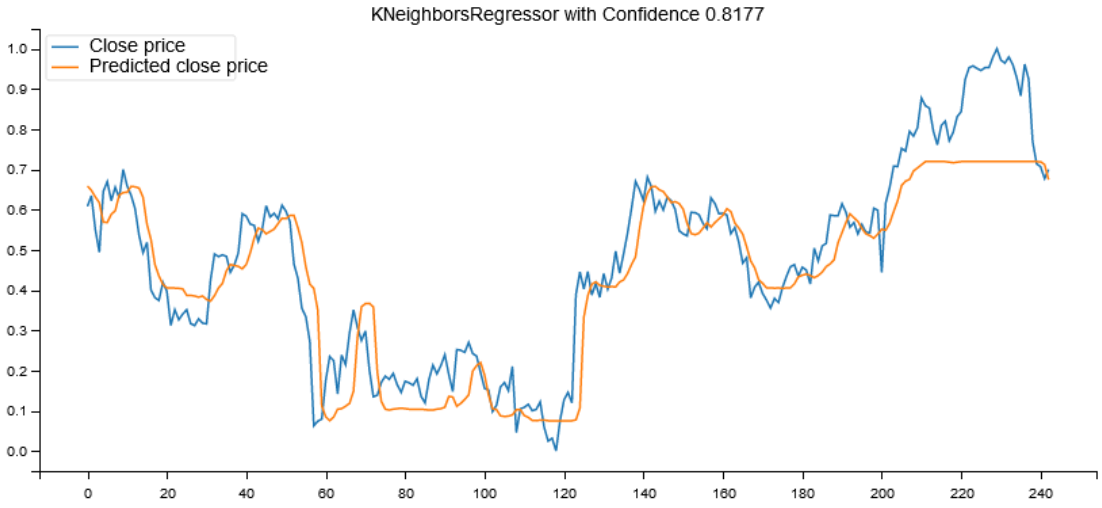
Το συμπέρασμα στο οποίο καταλήγουμε είναι ότι το μοντέλο που είναι σε θέση να προβλέψει με μεγαλύτερη ακρίβεια την μελλοντική πρόβλεψη των τιμών κλεισίματος με ποσοστό 0,32% είναι αυτό του Bayesian Ridge.

Αντιστοίχως γίνεται η αντίστοιχη διαδικασία για timestamp = 4,3,2,1 και το ποσοστό πρόβλεψης το μοντέλων είναι το ακόλουθο.

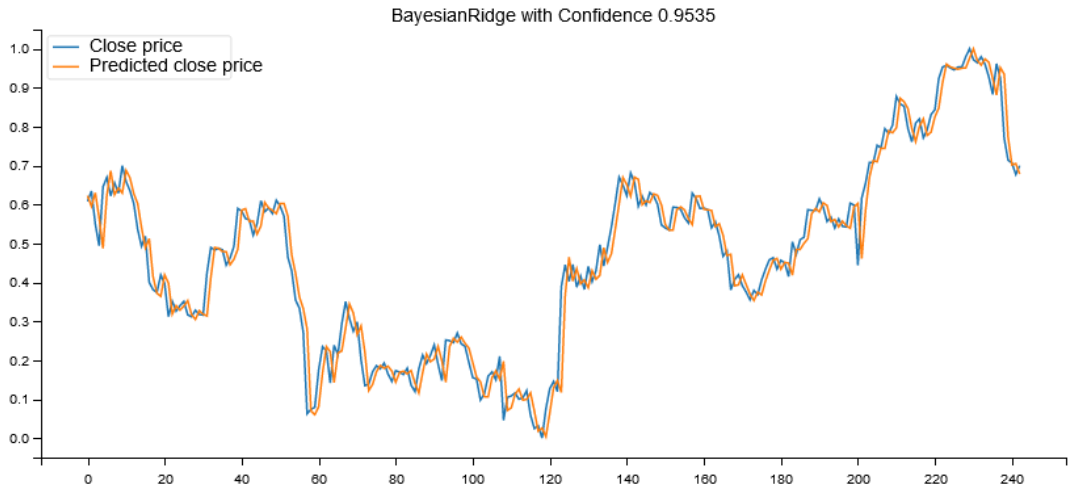
Για timestamp=4



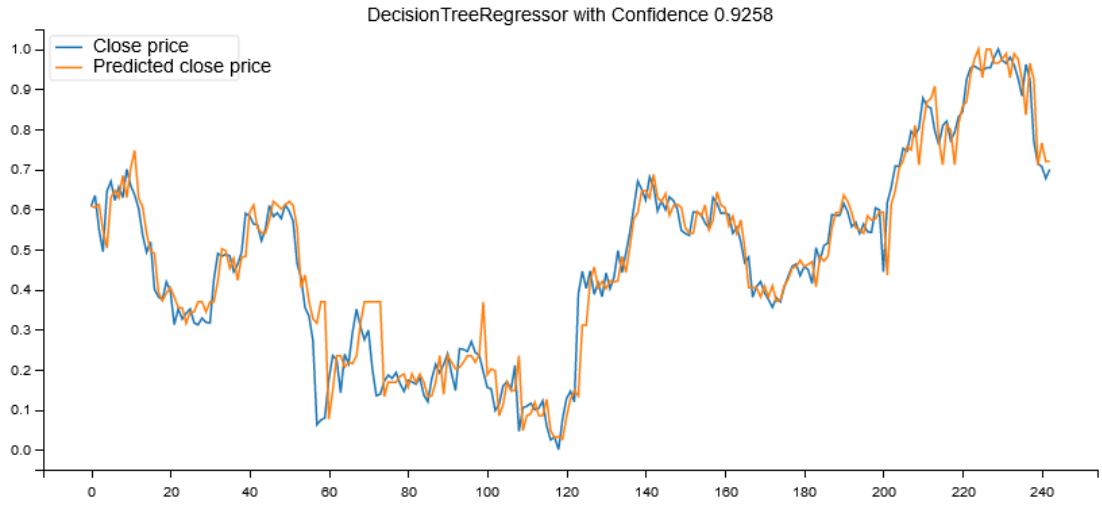
Σχήμα 70. Linear regression



Σχήμα 71. KNeighboors regression



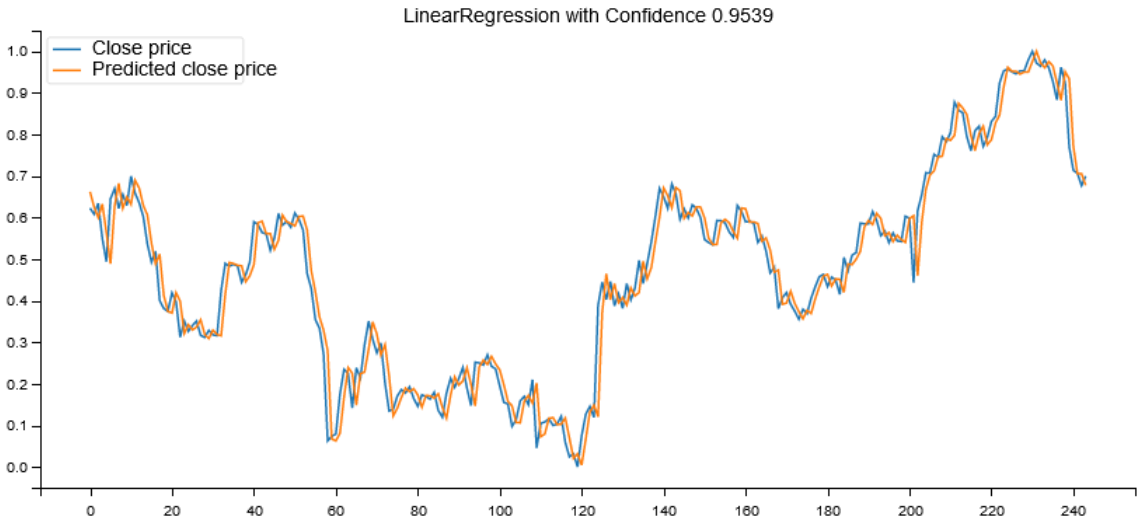
Σχήμα 72. Bayesian Ridge regression



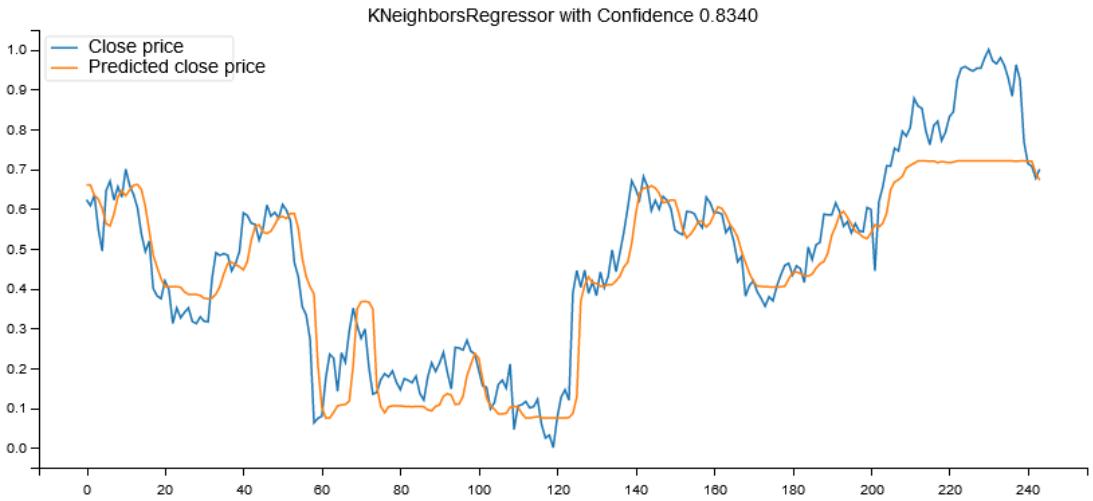
Σχήμα 73. Decision Tree regression

Συνεπώς το καλύτερο μοντέλο στην συγκεκριμένη περίπτωση με ποσοστό 95.35% είναι αυτό του είναι αυτό του Bayesian Ridge παρατηρώντας ότι σε σχέση με τις προηγούμενες προβλέψεις το μοντέλο Linear Regression πρόβλεψε με λίγο μεγαλύτερο ποσοστό ακρίβειας ενώ ανέβηκε το ποσοστό πρόβλεψης τόσο για το kNeighbors όσο και για το Decision tree.

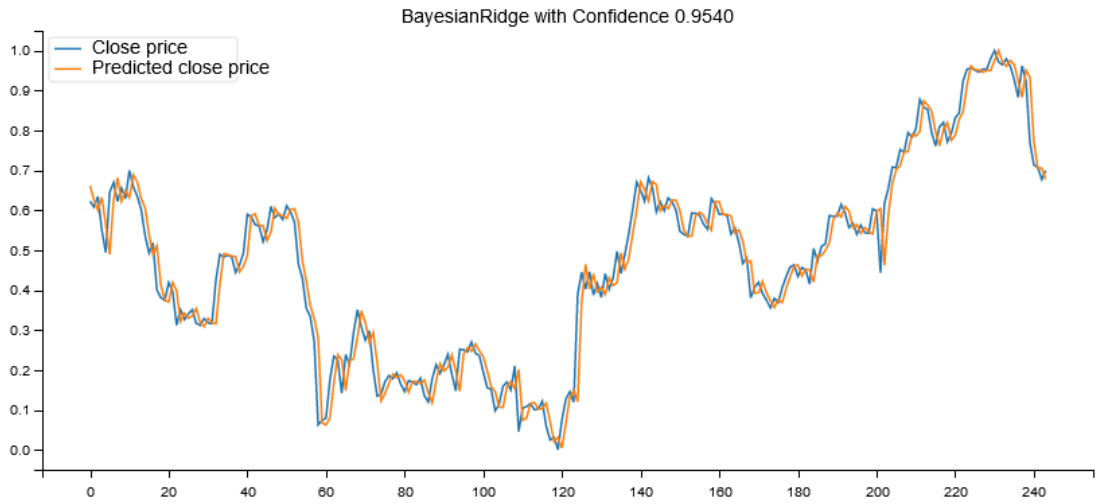
Για Timestamp=3



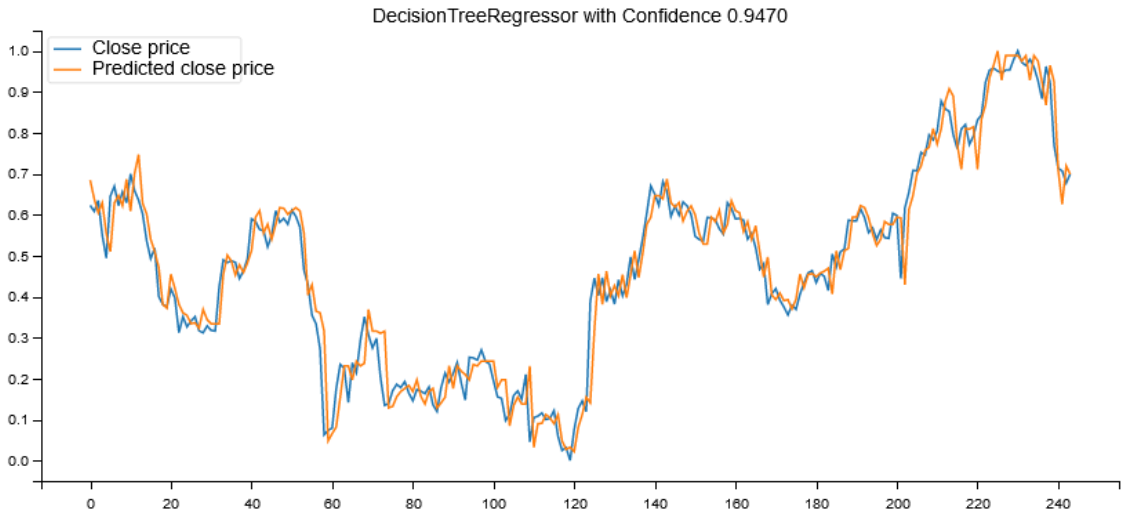
Σχήμα 74. Linear regression



Σχήμα 75. KNeighboors regression



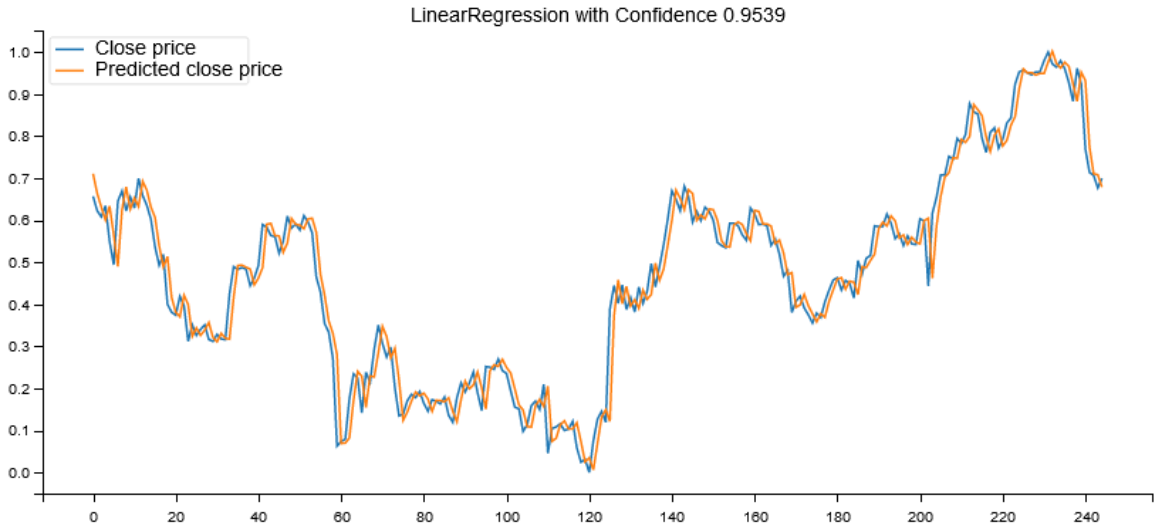
Σχήμα 76. Bayesian Ridge regression



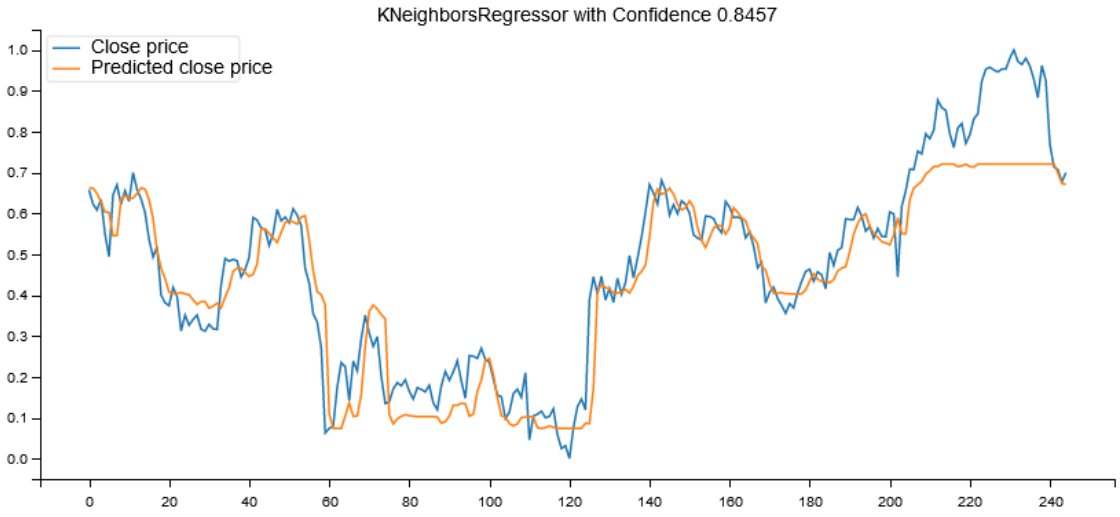
Σχήμα 77. Decision Tree regression

Συνεπώς το καλύτερο μοντέλο στην συγκεκριμένη περίπτωση με ποσοστό 95.40% είναι αυτό του Bayesian Ridge παρατηρώντας ότι σε σχέση με τις προηγούμενες προβλέψεις το μοντέλο Linear Regression πρόβλεψε με λίγο μεγαλύτερο ποσοστό ακρίβειας ενώ ανέβηκε το ποσοστό πρόβλεψης τόσο για το kNeighbors όσο και για το Decision tree.

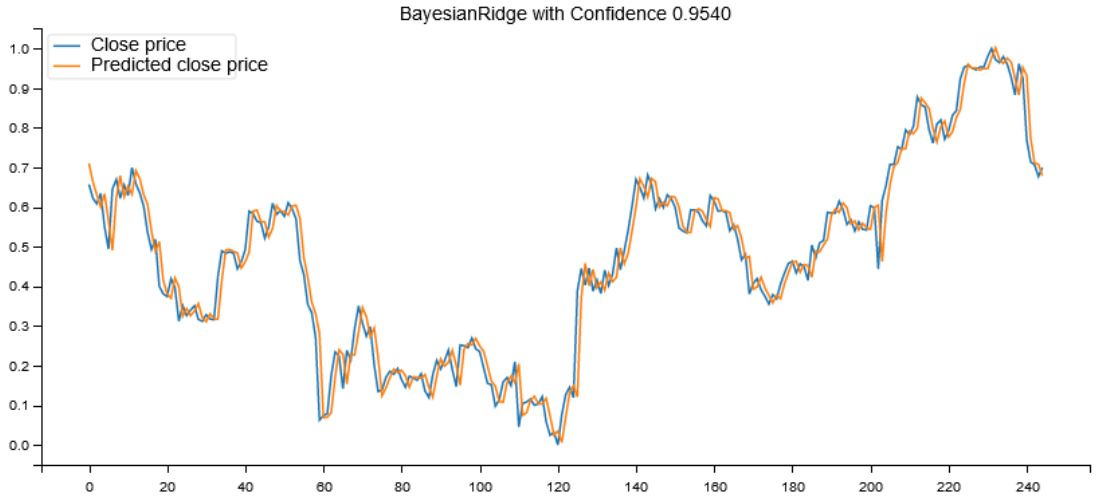
Για Timestamp=2



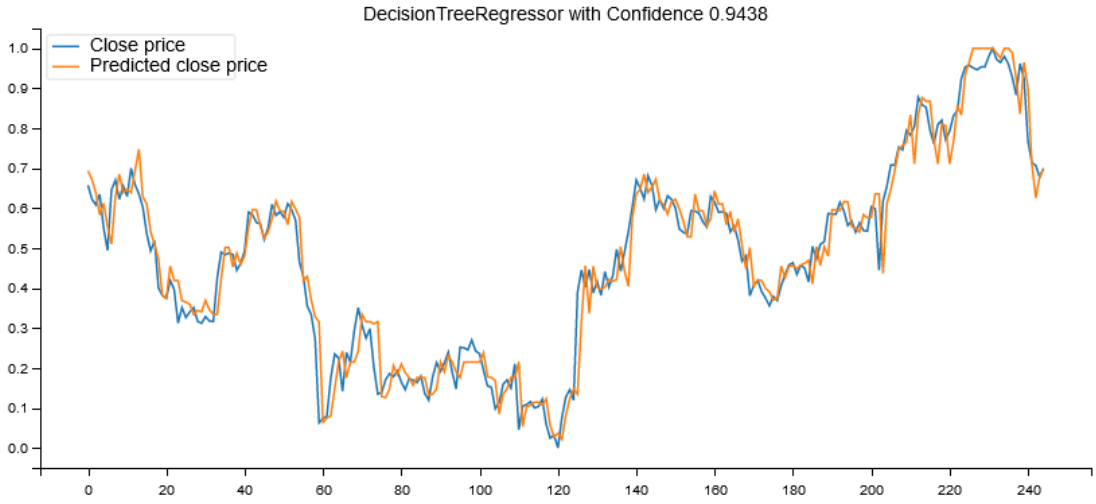
Σχήμα 78. Linear regression



Σχήμα 79. KNeighboors regression



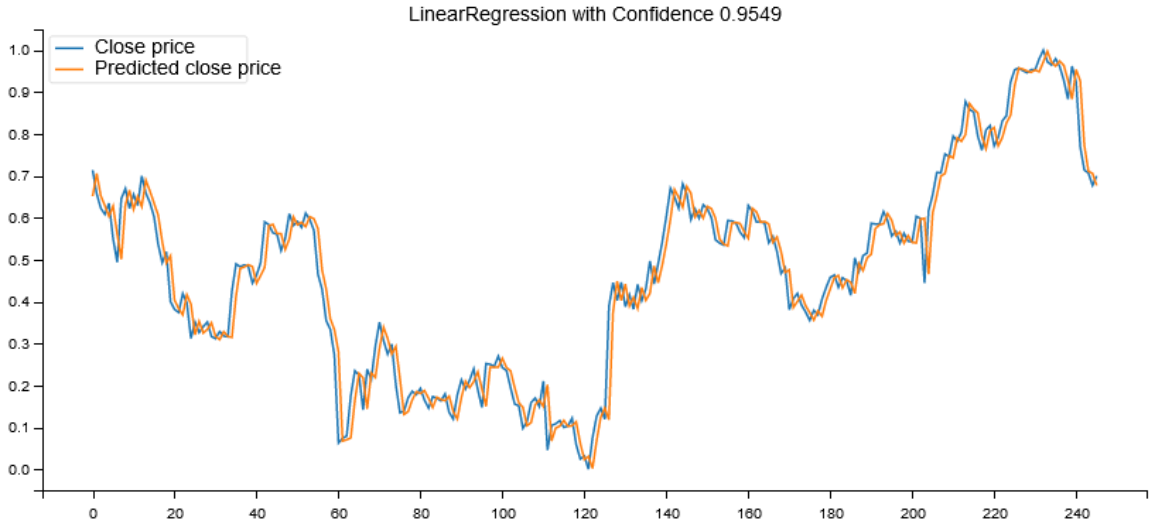
Σχήμα 80. Bayesian Ridge regression



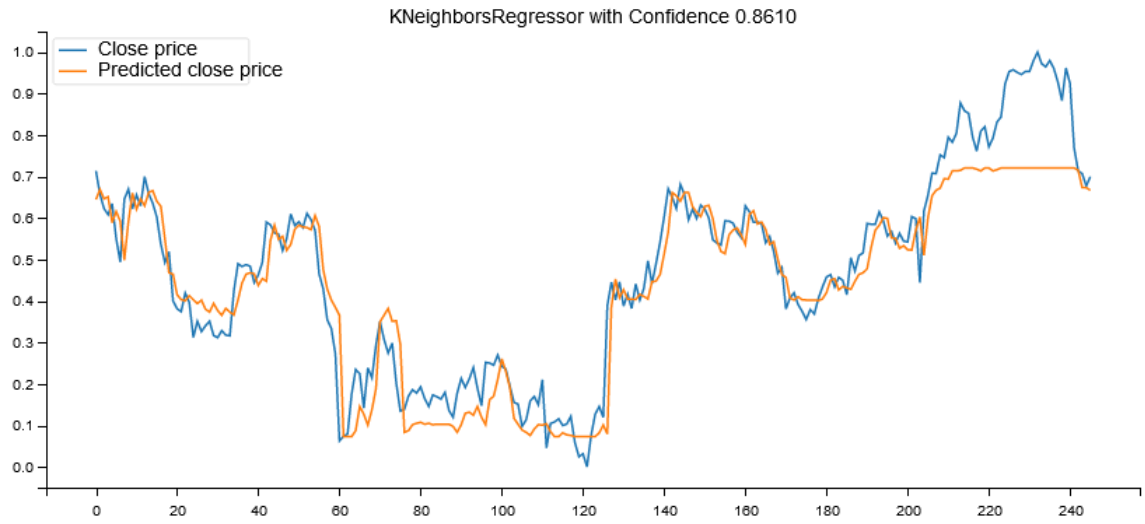
Σχήμα 81. Decision Tree regression

Συνεπώς το καλύτερο μοντέλο στην συγκεκριμένη περίπτωση με ποσοστό 95.40% είναι αυτό του Bayesian Ridge παρατηρώντας ότι σε σχέση με την προηγούμενη πρόβλεψη το μοντέλο Linear Regression πρόβλεψε με το ίδιο ποσοστό ακρίβειας, ανέβηκε το ποσοστό πρόβλεψης τόσο για το kNeighbors ενώ για το Decision tree έπεσε το ποσοστό πρόβλεψης.

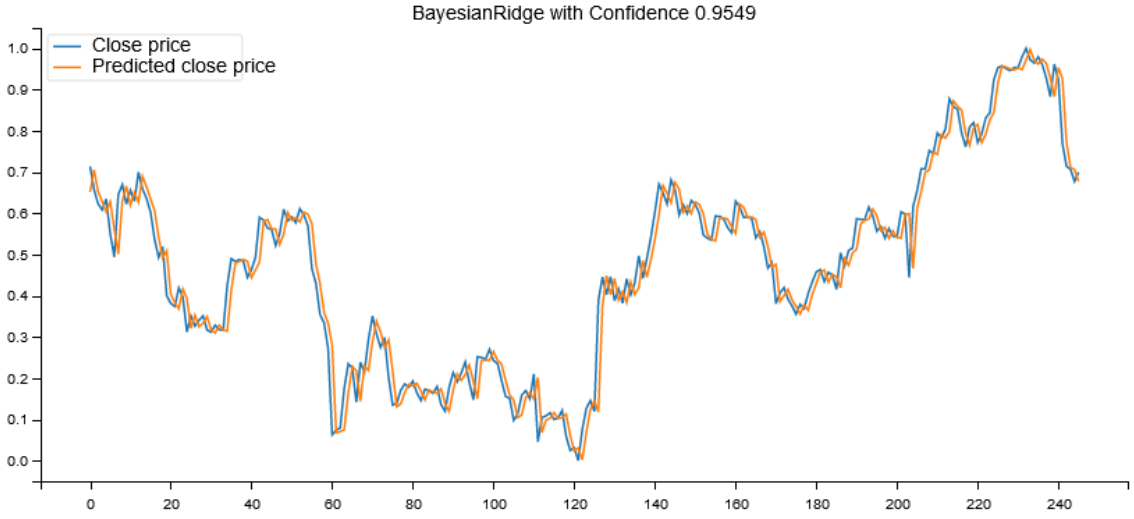
Για Timestamp=1



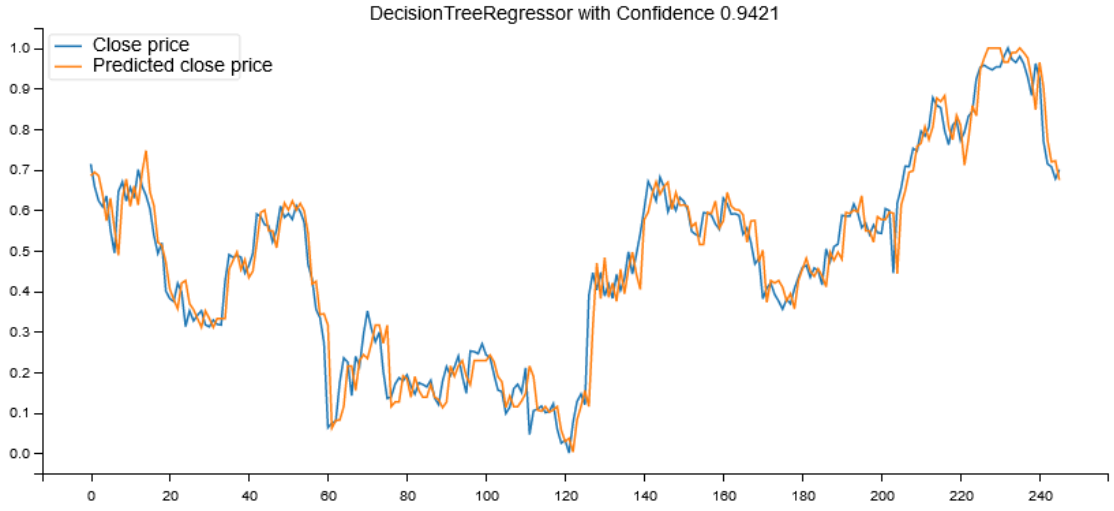
Σχήμα 82. Linear regression



Σχήμα 83. KNeighboors regression



Σχήμα 84. Bayesian Ridge regression



Σχήμα 85. Decision Tree regression

Συνεπώς τα καλύτερα μοντέλο στην συγκεκριμένη περίπτωση με ίδιο ποσοστό 95.49% είναι αυτά του Linear regression και Bayesian Ridge. Σε σχέση με την προηγούμενη πρόβλεψη το μοντέλο kNeighbors πρόβλεψε με μεγαλύτερο ποσοστό ακρίβειας, ενώ το ποσοστό πρόβλεψης για το Decision tree έπεσε ελάχιστα.

Συμπερασματικά από τις εναλλαγές του Timestamp για πρόβλεψη ανάλογα με το βάθος των ημερών παρατηρούμε πως η βραχυχρόνια πρόβλεψη γίνεται με πολύ μεγάλη ακρίβεια σε

σχέση με τη μακροχρόνια πρόβλεψη. Βέβαια οι διαφορές είναι μικρές και τα πιο ‘’δυνατά’’ μας μοντέλα που είναι τόσο το μοντέλο Linear regression όσο και το μοντέλο Bayesian Ridge είναι σε θέση να προβλέψουν με σχετικά ελάχιστες αποκλίσεις.

Παρατίθεται συγκριτικός πίνακας όλων των μοντέλων.

	Machine learning Models				
	Timestamp 1	Timestamp 2	Timestamp 3	Timestamp 4	Timestamp 5
Linear Regression	0,9549	0,9549	0,9539	0,9534	0,9532
KNeighbors	0,8610	0,8457	0,8340	0,8177	0,8058
BayesianRidge	0,9549	0,9540	0,9540	0,9535	0,9533
DecisionTree	0,9421	0,9438	0,9470	0,9258	0,9453

Σχήμα 86. Συγκριτικός πίνακας μοντέλων

	Machine learning Models υπολογισμός με R2_Score				
	Timestamp 1	Timestamp 2	Timestamp 3	Timestamp 4	Timestamp 5
Linear Regression	0,9549	0,9549	0,9539	0,9534	0,9532
KNeighbors	0,8610	0,8457	0,8340	0,8177	0,8058
BayesianRidge	0,9549	0,9540	0,9540	0,9535	0,9533
DecisionTree	0,9421	0,9438	0,9470	0,9258	0,9453
Errors					
LSTM model MAE	0.9496	0.9191	0.8914	0.8593	0.8306
LSTM model MSE	0.9484	0.9048	0.8931	0.8631	0.8365
LSTM model RMSE	0.9396	0.9216	0.8971	0.8650	0.8410
GRU model MAE	0.9396	0.9216	0.8971	0.8650	0.8410
GRU model MSE	0.9396	0.9216	0.8971	0.8650	0.8410
GRU model RMSE	0.9396	0.9216	0.8971	0.8650	0.8410

Σχήμα 87. Συγκεντρωτικός πίνακας αποτελεσμάτων

Από τον παραπάνω πίνακα και τα συγκεντρωτικά αποτελέσματα βλέπουμε πως τα καλύτερα αποτελέσματα είναι αυτά με timestamp = 1 καθώς η συσχέτιση τους είναι ισχυρότερη συγκριτικά με τις υπόλοιπες timestamp τιμές.

13. Μελλοντική εξέλιξη

Η εργασία μας είχε ως σκοπό να αναδείξει τα μεγάλα δεδομένα και τα μέσα τα οποία μπορούμε να χρησιμοποιήσουμε ώστε να επεξεργαστούμε δεδομένα χρήσιμα προς την εξαγωγή συμπερασμάτων.

Στα αρχικά κεφάλαια έγιναν αναφορές στο θεωρητικό κομμάτι ώστε να δημιουργηθεί ένα καλό θεωρητικό υπόβαθρο στην κατανόηση των όρων, ενώ στη συνέχεια προχωρήσαμε πρακτικά στο κομμάτι τόσο της διαχείρισης χαρτοφυλακίου όσο και στην πρόβλεψη μετοχών.

Μελλοντικά το προϊόν θα μπορούσε να απευθυνθεί σε υποψήφιους επενδυτές που ασχολούνται με την αγορά του χρηματιστηρίου.

Πιο συγκεκριμένα θα μπορούσε να κατασκευαστεί ένα σύστημα σχεσιακής βάσης δεδομένων για τη συλλογή τιμών μετοχών και ενός φιλικού User Interface για την απεικόνιση κατασκευάζοντας ένα περιβάλλον προς τον υποψήφιο επενδυτή το οποίο να έχει επιλογές πρόβλεψης χρονοσειρών, διαχείρισης χαρτοφυλακίου και άλλες λειτουργίες οι οποίες θα μπορούσαν να φανούν χρήσιμες στις επενδύσεις τους λαμβάνοντας σωστές αποφάσεις χωρίς μεγάλο ρίσκο.

Βιβλιογραφία

Ακολουθούν βιβλιογραφικές αναφορές (πηγές) της εργασίας

- Αλέξανδρος Νανόπουλος - Ιωάννης Μανωλόπουλος «Εισαγωγή στην εξόρυξη και τις αποθήκες δεδομένων»
- Τα μεγάλα δεδομένα από το 2004 μέχρι σήμερα (https://trends.google.com/trends/explore?date=all&q=big%20data)
- Διαφάνειες μαθήματος «Ανάλυση δεδομένων με κλασσικές τεχνικές» της καθηγήτριας Γεωργία Φουτσιτζή, στο πλαίσιοτου μεταπτυχιακού προγράμματος.
- JAVA ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ, Deitel, Paul & Harvey, 2015
- Στέφανος Παπαδάμου, Διαχείριση χαρτοφυλακίου μια σύγχρονη προσέγγιση.
- Διαχείριση χαρτοφυλακίου Βασιλείου, 2008, σελ.145-146.
- Δημήτρης Ιωαννίδης, Ιωάννης Αθανασιάδης "Στατιστική και μηχανική μάθηση με την R: Θεωρία και εφαρμογές"
- Human resources for Big Data professions: A systematic classification of job roles and required skill sets, Andrea De Mauro, Marco Greco, Michele Grimaldi, Paavo Ritala
- The Role of Big Data Solutions in the Management of Organizations. Review of Selected Practical Examples
- P. Russom, Big Data Analytics (2011)
- Business Intelligence and Analytics: From Big Data to Big ImpactChen, Chiang, & Storey 2012, Manyika 2011, Marr 2015
- How strategists use "big data" to support internal business decisions, discovery and production Daveport, 2014
- Big data, Big bang? Bughin, 2016
- Spark: The Definitive Guide: Big Data Processing Made Simple Billchambers&MateiZaharia, 2018
- Learning spark: lightning-fast big data analysis KarauH.etal, 2015
- Python in education Nicholas H.Tol., 2015, Python, (n,d).
- Python Data Science Handbook: Essential Tools for Working with Data,WesMcKin., [2012]

- Yves Hilpisch-Python for Finance_ Analyze Big Financial Data-O'Reilly Media (2014)
- The Scala Programming Language. Scala. Scala. https://www.scala-lang.org/.
- Scala. Wikipedia. https://en.wikipedia.org/wiki/Scala_(programming_language).
- Guru99. Java vs Scala: What is the Difference? guru99. [Ηλεκτρονικό] 2020.
- https://www.guru99.com/scala-vs-java.html.
- Keenan, Tyler. Scala: A Hybrid Language for Big Data. upwork. 24 January 2017.
- https://www.upwork.com/hiring/data/scala-hybrid-functionalobject-oriented-language-big-data/.
- Apache Spark. Apache Spark-Lightning-fast unified analytics engine. https://spark.apache.org/.
- Apache Spark. Wikipedia. January 2020. https://en.wikipedia.org/wiki/Apache_Spark.
- Jeffrey Dean, Sanjay Ghemawat και Google Inc. Mapreduce: simplified dataprocessing on large clusters. Στο In OSDI'04: Proceedings of the 6th conference onSymposium on Opearting Systems Design & Implementation. USENIX Association,2004