Σχολή Θετικών Επιστημών και Τεχνολογίας

Μεταπτυχιακή Εξειδίκευση στα Πληροφοριακά Συστήματα

Διπλωματική Εργασία

«Αξιολόγηση Μεθόδων Μηχανικής Μάθησης για την Πρόβλεψη Κατεύθυνσης Μεταβλητότητας Κρυπτονομισμάτων»

Γεώργιος Μίχης

Επιβλέπων καθηγητής: Εμμανουήλ Τζαγκαράκης

Πάτρα, Ιούνιος 2022

«Αξιολόγηση Μεθόδων Μηχανικής Μάθησης για την Πρόβλεψη Κατεύθυνσης Μεταβλητότητας Κρυπτονομισμάτων»

Γεώργιος Μίχης

Επιτροπή Επίβλεψης Διπλωματικής Εργασίας

Επιβλέπων Καθηγητής:	Συν-Επιβλέπων Καθηγητής:
Εμμανουήλ Τζαγκαράκης	Δημήτριος Καραπιπέρης
Αναπληρωτής Καθηγητής Πανεπιστημίου Πατρών	Λέκτορας Διεθνούς Πανεπιστημίου Ελλάδος

Πάτρα, Ιούνιος 2022

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κ. Τζαγκαράκη Εμμανουήλ για την υποστήριξη, την καθοδήγηση και την βοήθεια που μου προσέφερε για την πραγματοποίηση αυτής της Διπλωματικής Εργασίας.

Επίσης, θα ήθελα να ευχαριστήσω και στην συνέχεια να αφιερώσω αυτήν την Διπλωματική Εργασία στους γονείς μου και την αδερφή μου.

Περίληψη

Τα κρυπτονομίσματα την τελευταία δεκαετία έχουν αποκτήσει φωνή και έχουν μπει για τα καλά στον χρηματοοικονομικό κόσμο. Παρουσιάζουν μεγάλη μεταβλητότητα, δηλαδή έχουν μεγάλο εύρος αλλά και ταχύτητα διακύμανσης στην ισοτιμία τους. Αυτό είναι κάτι πρωτόγνωρο για τον τομέα των νομισμάτων. Υπάρχουν αρκετές μελέτες και έχουν γίνει πολλές έρευνες που προσπαθούν είτε να εξηγήσουν είτε να προβλέψουν την κατεύθυνση της μεταβλητότητας. Σε αυτές τις προσπάθειες τα τελευταία χρόνια έχουν αρχίσει να συμπεριλαμβάνονται και αλγόριθμοι από την περιοχή της μηχανικής μάθησης. Στόχος αυτής της διπλωματικής εργασίας είναι η αξιολόγηση αλγορίθμων μηχανικής μάθησης σχετικά με την ικανότητά τους να προβλέψουν την κατεύθυνση της μεταβλητότητας του δημοφιλέστερου κρυπτονομίσματος, του Bitcoin, χρησιμοποιώντας πραγματικά δεδομένα που σχετίζονται με αυτό. Στην παρούσα διπλωματική εργασία θα προσεγγίσουμε αυτό το πρόβλημα σαν ένα πρόβλημα κατηγοριοποίησης. Οι αλγόριθμοι που επιλέχθηκαν είναι ο Knn, ο Random Forest, τα Νευρωνικά Δίκτυα και η Λογιστική Παλινδρόμηση. Η σύγκριση και η αξιολόγησή τους έγινε σύμφωνα με το ποσοστό ακρίβειας που συγκέντρωσε ο καθένας. Για τα δεδομένα που χρησιμοποιήσαμε, το μοντέλο του Random Forest είχε την καλύτερη απόδοση με ποσοστό ακρίβειας 66%. Ακολουθεί η Λογιστική Παλινδρόμηση με 62%, τα Νευρωνικά Δίκτυα με 60% και τέλος ο Knn με ποσοστό 54%. Τα αποτελέσματα είναι αρκετά υποσχόμενα και αφήνουν περιθώριο για μελλοντικές έρευνες στην εξέταση τόσο αυτών των αλγορίθμων όσο και άλλων αλγορίθμων της μηχανικής μάθησης.

Λέξεις – Κλειδιά

Μηχανική Μάθηση, κατηγοριοποίηση, μεταβλητότητα, Bitcoin, knn, random forest, νευρωνικά δίκτυα, λογιστική παλινδρόμηση

"Evaluation of Machine Learning Algorithms for Predicting the Direction of Volatility in Cryptocurrencies"

Georgios Michis

Abstract

In the last decade, cryptocurrencies have made some noise and they managed to establish themselves in the financial world. Cryptocurrencies show high volatility, meaning they have a great range and fluctuation in their exchange rate. This is something new for currencies. There are enough papers and studies that try to explain or predict the direction of volatility. In these attempts the last few years, they have started using machine learning algorithms. The purpose of this paper is to evaluate these machine learning algorithms in their ability to predict the direction of volatility of Bitcoin, the most famous crypto, by using real data related to it. In this paper we will approach this problem as a classification problem. The algorithms that were chosen are Knn, Random Forest, Neural Networks and Logistic Regression. The comparison and the evaluation of the algorithms was made through the accuracy of the prediction that each one had. For the data we used, Random Forest model had the best performance with an accuracy of 66%. Logistic Regression is the next to follow with 62%, then Neural Networks with 60% and finally Knn with 54%. The results are quite promising and they leave plenty of room for future studies to examine these and other machine learning algorithms.

Keywords

Machine learning, classification, volatility, Bitcoin, knn, random forest, neural networks, logistic regression

Περιεχόμενα

Κατάλογος Εικόνων / Σχημάτων

Κατάλογος Πινάκων

Συντομογραφίες & Ακρωνύμια

ΔΕ	Διπλωματική Εργασία
ΕΑΠ	Ελληνικό Ανοικτό Πανεπιστήμιο
P2P	Peer-to-peer
ML	Machine Learning
MSCI	Stock market index
S&P 500	Stock market index
VIX	Volatility index for stock market
GARCH	Generalized AutoRegressive Conditional Heteroskedasticity
HAR	Heterogenous AutoRegressive
MAPE	Mean Absolute Percentage Error
MSPE	Mean Squared Prediction Error
MAE	Mean Absolute Error
MSE	Mean Squared Error
RMSE	Root Mean Square Error
QLIKE	Quasi-Likelihood
AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion
D-M	Diebold-Mariano test
MCS	Model Confidence Set
RNN	Recurrent Neural Network
VaR	Value at Risk

RF	Random Forest
SVM	Support Vector Machine
BNN	Bayesian Neural Network
DM	Data Mining
DL	Deep Learning
CNN	Convolutional Neural Network
NN	Neural Network
MLP	Multi-Layer Perceptron
CPU	Central Processing Unit
GPU	Graphics Processing Unit
RAM	Random Access Memory
TPU	Tensor Processing Unit

1. Εισαγωγή

1.1 Αντικείμενο της εργασίας

Βρισκόμαστε σε μία εποχή όπου η τεχνολογία εξελίσσεται συνεχώς. Το διαδίκτυο βρίσκεται παντού και πληθαίνουν συνέχεια τα ηλεκτρονικά μέσα που εξαρτώνται από αυτό. Μέσα από αυτήν την εξέλιξη, τα τελευταία χρόνια επηρεάστηκε και ο κλάδος της οικονομίας. Αυτό είχε σαν αποτέλεσμα την δημιουργία των κρυπτονομισμάτων. Πρόκειται για ψηφιακά νομίσματα και η ιδέα της δημιουργίας τους ήταν για διευκόλυνση των συναλλαγών των χρηστών, χωρίς να υπάρχουν μεσάζοντες ή να υπάρχει εξάρτιση του νομίσματος από κάποια κεντρική αρχή, όπως είναι οι κυβερνήσεις των χωρών. Η δημιουργία τους, βασίστηκε στην κρυπτογραφία και αυτό αποφέρει αξιοπιστία και ασφάλεια. Το κρυπτονόμισμα που τάραξε τα νερά και είναι πλέον το δημοφιλέστερο και το διασημότερο είναι το Bitcoin. Μέσα στα χρόνια παρουσίασε τρομερή αύξηση στην τιμή του. Παρατηρήθηκε στην συνέχεια ότι χαρακτηρίζεται από υψηλή μεταβλητότητα και αυτό σημαίνει επενδυτικές ευκαιρίες. Αυτό είχε σαν αποτέλεσμα να κεντρίσει όλο και περισσότερο το ενδιαφέρον επενδυτών και να τους οδηγήσει σε προσπάθειες πρόβλεψης της τιμής του.

Η συμπεριφορά του Bitcoin μοιάζει αρκετά με αυτή των μετοχών του χρηματιστηρίου. Έτσι για την πρόβλεψη αυτών, πέρα από οικονομικά μοντέλα, πολλοί χρησιμοποίησαν την μηχανική μάθηση. Παρέχουμε δεδομένα (δεδομένα εκπαίδευσης) στους αλγορίθμους της μηχανικής μάθησης και τους εκπαιδεύουμε με σκοπό την δημιουργία μοντέλων. Έπειτα εισάγουμε δεδομένα (δεδομένα ελέγχου) και εξετάζουμε το μοντέλο που δημιουργήσαμε.

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι να γίνει μια προσπάθεια πρόβλεψης της κατεύθυνσης της μεταβλητότητας του Bitcoin. Θα αντιμετωπιστεί σαν ένα πρόβλημα κατηγοριοποίησης. Μέσα από τους αλγορίθμους μηχανικής μάθησης θα προσπαθήσουμε να προβλέψουμε αν θα υπάρχει αύξηση ή μείωση στην τιμή του κρυπτονομίσματος Bitcoin. Στην συνέχεια θα συγκρίνουμε και θα αξιολογήσουμε τα μοντέλα που προέκυψαν από τους αλγορίθμους μηχανικής μάθησης.

1.2 Δομή της εργασίας

Στο πρώτο κεφάλαιο της διπλωματικής εργασίας, γίνεται μια εισαγωγή του αντικειμένου που πραγματεύεται αλλά και της δομής της.

Στο δεύτερο κεφάλαιο γίνεται η βιβλιογραφική ανασκόπηση, που αφορά το Bitcoin, την μεταβλητότητα, την μηχανική μάθηση και τον συνδυασμό αυτών.

Το τρίτο κεφάλαιο αφορά τα δεδομένα της διπλωματικής εργασίας. Γίνεται η ανάκτηση των δεδομένων, η επεξεργασία τους και στην συνέχεια η εφαρμογή των αλγορίθμων μηχανικής μάθησης.

Στο τέταρτο κεφάλαιο παρουσιάζονται τα αποτελέσματα των 4 αλγορίθμων μηχανικής μάθησης που χρησιμοποιήθηκαν.

Στο κεφάλαιο 5 καταγράφονται τα συμπεράσματα των αποτελεσμάτων.

2. Βιβλιογραφική Επισκόπηση

Εδώ θα παρουσιαστούν κάποιες από τις έννοιες που αφορούν και θα εμφανιστούν σε μεγάλο βαθμό μέσα σε αυτήν την διπλωματική εργασία όπως είναι το κρυπτονόμισμα και συγκεκριμένα το Bitcoin, η μεταβλητότητα σαν έννοια όπως παρουσιάζεται στον τομέα των οικονομικών και κατ' επέκταση η μεταβλητότητα του Bitcoin. Ακόμα θα παρουσιαστούν και οι αλγόριθμοι μηχανικής μάθησης που θα χρησιμοποιηθούν για την προσπάθεια υπολογισμού της μεταβλητότητας του κρυπτονομίσματος Bitcoin.

2.1 Τι είναι τα κρυπτονομίσματα;

Όταν μιλάμε για κρυπτονομίσματα μιλάμε για ψηφιακά ή εικονικά νομίσματα. Όπως αναφέρει και το πρώτο συνθετικό του ονόματός τους, η δημιουργία αυτών των ψηφιακών νομισμάτων βασίζεται στην κρυπτογραφία, σε εξειδικευμένους αλγορίθμους και τεχνικές. Εξαιτίας της κρυπτογραφίας, τα κρυπτονομίσματα είναι κατά κάποιον τρόπο μοναδικά και έτσι δεν μπορούν να δημιουργηθούν πλαστά ή να έχουμε διπλή δαπάνη από αυτά (double-spend). Τα κρυπτονομίσματα είναι αποκεντρωμένα δίκτυα και η λειτουργία τους βασίζεται στην τεχνολογία που είναι γνωστή ως blockchain. Αυτό που κάνει τα κρυπτονομίσματα ξεχωριστά είναι το γεγονός ότι δεν υπάρχει κάποιος κεντρικός φορέας που να είναι υπεύθυνος για την έκδοσή τους και έτσι δεν χρειάζεται την στήριξη καμίας κυβέρνησης ή άλλων χρηματοοικονομικών αρχών.

Τα κρυπτονομίσματα παράγονται από την ψηφιακή εξόρυξη (mining) ή μπορούν να αποκτηθούν μέσα από επιχειρήσεις που ειδικεύονται στην ανταλλαγή κρυπτονομισμάτων (cryptocurrency exchanges). Τα κρυπτονομίσματα διασφαλίζουν ασφαλείς πληρωμές για αγορές μέσω διαδικτύου χωρίς να χρειάζονται μεσάζοντες. Παρόλο αυτά, τα κρυπτονομίσματα δεν υποστηρίζονται ακόμα από όλες τις ιστοσελίδες ηλεκτρονικού εμπορίου και ας υπάρχουν κάποια που είναι αρκετά δημοφιλή όπως το Bitcoin ή το Ethereum. Ωστόσο λόγο των υψηλών τιμών που έχουν φτάσει, τα κρυπτονομίσματα χρησιμοποιούνται ευρέως για επενδύσεις και συναλλαγές.

Κεντρικό ρόλο στην λειτουργία του συστήματος των κρυπτονομισμάτων, παίζει η τεχνολογία που είναι γνωστή ως blockchain. Το blockchain είναι μια βάση δεδομένων, στην οποία καταγράφονται όλες οι συναλλαγές που έχουν πραγματοποιηθεί και έχουν

επαληθευτεί από όλα τα μέλη ανεξαρτήτως, που το χρησιμοποιούν. Αποτελεί ουσιαστικά το λογιστικό βιβλίο (online ledger). Το blockchain αποτελείται από συνδεδεμένα blocks και κάθε block καταγράφει ένα σύνολο συναλλαγών. Κάθε καινούριο block που παράγεται πρέπει να επαληθευτεί από κάθε χρήστη πριν να επιβεβαιωθεί. Αυτό δημιουργεί μοναδικότητα στις συναλλαγές και δεν αφήνει να επαναξοδευτούν νομίσματα ή να τροποποιηθούν τα ιστορικά συναλλαγών. Το πρώτο αποκεντρωμένο blockchain μας το παρουσίασε το άτομο ή τα άτομα που είναι γνωστό-ά με το ψευδώνυμο Satoshi Nakamoto. Η τεχνολογία blockchain έχει βρει μεγάλη απήχηση σε πολλούς κλάδους, όπως αυτός της οικονομίας, της εφοδιαστικής αλυσίδας, το crowdfunding και το online voting.

Η εισαγωγή των κρυπτονομισμάτων στον κόσμο της οικονομίας είχε σημαντικές επιδράσεις και μένει να δούμε πως και αν αυτό το αποκεντρωμένο σύστημα μπορεί να έχει και πιο άμεσες πρακτικές εφαρμογές. Όπως είπαμε πρόκειται για ένα αποκεντρωμένο σύστημα που δεν χρειάζεται διαμεσολαβητές όπως τράπεζες ή άλλα χρηματοοικονομικά ινστιτούτα. Διευκολύνεται η μεταφορά κεφαλαίων καθώς χρησιμοποιείται ένα σύστημα δημόσιων και ιδιωτικών κλειδιών και μηχανισμοί όπως το proof of work και το proof of stake. Επιπλέον η μεταφορά είναι πιο εύκολη και πιο γρήγορη σε σχέση με μια μεταφορά κεφαλαίων μέσω τραπέζης. Τέλος, επενδύσεις στα κρυπτονομίσματα μπορούν να αποφέρουν μεγάλο κέρδος, με χαρακτηριστικό παράδειγμα την τιμή του κρυπτονομίσματος bitcoin την τελευταία δεκαετία. Ωστόσο, τα κρυπτονομίσματα έχουν και αρκετούς κινδύνους. Αρχικά τα χαρακτηρίζει μεγάλη μεταβλητότητα. Αυτό μπορεί να φανεί από τα μεγάλα σκαμπανεβάσματα που έχει στην τιμή τους. Εξαιτίας αυτού τους γεγονότος πολλοί πιστεύουν ότι πρόκειται για «φούσκα». Ενώ η τεχνολογία του blockchain θεωρείται ασφαλής, η συναλλαγή κρυπτονομισμάτων ή το ψηφιακό πορτοφόλι (crypto wallet) μπορούν να παραβιαστούν. Ο καθένας μπορεί να κάνει εξόρυξη για να βρει κρυπτονομίσματα, ωστόσο όσο αναφορά τα πιο δημοφιλή, που είναι και αυτά με την υψηλότερη τιμή, απαιτείται πολύ μεγάλη ποσότητα ενέργειας, που ισοδυναμεί με την ενέργεια που μπορεί να χρειάζονται χώρες για να λειτουργήσουν.

Υπάρχουν πλέον πολλών ειδών κρυπτονομίσματα. Η αρχή έγινε το bitcoin που παρουσιάστηκε το 2008 μέσα από μία εργασία από το άτομο ή τα άτομα που είναι γνωστό-ά με το όνομα Satoshi Nakamoto. Είναι το πιο γνωστό και αυτό με την μεγαλύτερη χρηματιστηριακή αξία. Στην συνέχεια δημιουργήθηκαν κι άλλα που

βασίστηκαν στην λειτουργία του bitcoin και αναφέρονται ως altcoins. Κάποια από αυτά είναι το ethereum, το xrp, το tether, το cardano.

2.2 Το κρυπτονόμισμα Bitcoin

Σε αυτήν την διπλωματική εργασία επικεντρωνόμαστε στο κρυπτονόμισμα bitcoin, οπότε θα δούμε μερικά πράγματα γι' αυτό. Το Bitcoin είναι ένα αποκεντρωμένο ψηφιακό νόμισμα και δημιουργήθηκε πρώτη φορά τον Ιανουάριο του 2009. Βασίστηκε στην εργασία και στις ιδέες του Satoshi Nakamoto, όπου αποτελεί ένα ψευδώνυμο και η πραγματική ταυτότητα του ατόμου ή των ατόμων που βρίσκονται πίσω από αυτόν να είναι ακόμα και σήμερα κρυφή. Δημιουργήθηκε με την προϋπόθεση να παρέχει χαμηλότερη προμήθεια συναλλαγής για αγορές online και δεν βασίζεται σε τράπεζες ή κυβερνήσεις για την έκδοσή του.

Το bitcoin είναι ένα άυλο ψηφιακό νόμισμα. Η δημιουργία του βασίζεται στην κρυπτογραφία και χρησιμοποιεί την τεχνολογία peer-to-peer (P2P), δηλαδή στην διαμοίραση πληροφοριών, δεδομένων χωρίς την εμπλοκή κάποιας κεντρικής αρχής. Τα πάντα καταγράφονται σε ένα ψηφιακό λογιστικό βιβλίο και ο καθένας μπορεί να έχει πρόσβαση σε αυτό. Οι συναλλαγές με bitcoin επικυρώνονται μέσω της εξόρυξης (mining). Δεν αποτελεί ακόμα νόμιμο μέσω πληρωμής στις περισσότερες χώρες του κόσμου, ωστόσο είναι αρκετά δημοφιλές και έχει ανεβάσει πολύ την αξία του και έτσι πολλοί είναι αυτοί που επενδύουν σε αυτό και σε άλλα κρυπτονομίσματα που δημιουργήθηκαν εξαιτίας αυτού.

Μέσω της εφαρμογής της τεχνολογίας P2P και του blockchain, διασφαλίζεται η ανωνυμία και η ασφάλεια των συναλλαγών. Το σύστημα του Bitcoin αναφέρεται σε μία συλλογή υπολογιστών, όπου τρέχουν τον κώδικα του Bitcoin και τον αποθηκεύουν στο blockchain. Το blockchain αποτελείται από blocks, και κάθε block περιέχει δοσοληψίες. Όλοι οι υπολογιστές που τρέχουν το blockchain έχουν την ίδια λίστα με blocks και την ίδια λίστα με δοσοληψίες και είναι εμφανείς σε όλους είτε αυτοί κάνουν mining είτε όχι, όπου αυτό το καθιστά δύσκολο να μπορέσει κάποιος να «κλέψει» το σύστημα. Αυτοί που συμμετέχουν στο mining, είναι υπεύθυνοι για την καταγραφή των συναλλαγών και έχουν σαν κίνητρο να συνεχίσουν καθώς η δημιουργία κάθε νέου block προσφέρει bitcoin αλλά και οι επιβεβαιώσεις των συναλλαγών προσφέρουν κάποια αμοιβή. Τέλος σημαντικό ρόλο

παίζουν τα δημόσια και ιδιωτικά «κλειδιά». Πρόκειται για μία μεγάλη σειρά από αριθμούς και γράμματα που δημιουργούνται από αλγόριθμους κρυπτογραφίας. Το δημόσιο κλειδί είναι κάτι αντίστοιχο με τον αριθμό λογαριασμού τραπέζης και αποτελεί την διεύθυνση που κάποιος μπορεί να στείλει bitcoin. Το ιδιωτικό κλειδί είναι σαν το PIN πιστωτικής κάρτας που χρησιμοποιούμε στα ATM και χρησιμοποιείται για να εγκρίνουμε μεταφορές bitcoin.

Όλα ξεκίνησαν στις 31 Οκτωβρίου 2008 με την εργασία του Satoshi Nakamoto. Στις 3 Ιανουαρίου 2009 έχουμε το πρώτο block που δημιουργείται, το Block 0. Είναι γνωστό και ως genesis block και περιέχει την φράση «The Times 03/Jan/2009 Chancellor on brink of second bailout for banks». Στις 8 Ιανουαρίου 2009 ανακοινώνεται η πρώτη έκδοση του λογισμικού Bitcoin και στις 9 Ιανουαρίου 2009 γίνεται η εξόρυξη του Block 1. Σύμφωνα με τον αλγόριθμο του Bitcoin, μπορούν να υπάρξουν μόνο 21 εκατομμύρια bitcoin. Στην αρχή η δημιουργία κάθε νέου block έδινε σαν αμοιβή 50 bitcoin και αυτή η αμοιβή μειώνεται στην μέση κάθε 210000 block. Το Νοέμβριο του 2021 έχει γίνει περίπου η εξόρυξη του 90% των bitcoin, δηλαδή 18,85 εκατομμύρια bitcoin. Υπολογίζεται ότι η εξόρυξη του τελευταίου bitcoin θα γίνει κοντά στο έτος 2140.

Η ανάπτυξη που γνώρισαν τα κρυπτονομίσματα είναι μεγάλη. Το Bitcoin ξεκίνησε με αξία μικρότερη του 1 δολαρίου το 2011 και ξεπέρασε τις 68000 δολάρια το Νοέμβριο 2021. Δεν είναι τυχαίο που αρκετοί επενδυτές την χαρακτηρίζουν ως μία επένδυση με το μεγαλύτερο ρίσκο αλλά και με την υψηλότερη απόδοση.

2.3 Μεταβλητότητα

Η μεταβλητότητα είναι ένας όρος που συναντάμε κυρίως στον τομέα των χρηματοοικονομικών. Συγκεκριμένα, η μεταβλητότητα είναι ένας επενδυτικός όρος που αναφέρεται κυρίως για τις αγορές και τα χρηματιστήρια και περιγράφει την διακύμανση των τιμών για συγκεκριμένη χρονική περίοδο. Αυτή η διακύμανση στις τιμές μπορεί να είναι απρόβλεπτη και απότομη. Επίσης όταν σκεφτόμαστε τον όρο της μεταβλητότητας δεν θα πρέπει να τον συνδέουμε μόνο με την πτώση των τιμών, αλλά και με πιθανές μεγάλες αυξήσεις αυτών. Έτσι μέσω της μεταβλητότητας, οι επενδυτές υπολογίζουν το ρίσκο μιας επένδυσης για συγκεκριμένο χρονικό διάστημα. Καταλαβαίνουμε λοιπόν ότι μία επένδυση που χαρακτηρίζεται από υψηλή μεταβλητότητα, είναι πιθανό να έχει συχνές

και πιθανά μεγάλες αλλαγές αφού η τιμή μπορεί να ανέβει αρκετά ψηλά ή να πέσει αρκετά χαμηλά σε σχέση με μία επένδυση όπου η τιμή της μεταβλητότητας είναι χαμηλή.

Η μεταβλητότητα όπως είπαμε μετράει τις διακυμάνσεις των τιμών για συγκεκριμένο χρονικό διάστημα. Χρησιμοποιώντας όρους στατιστικής, μπορούμε να πούμε ότι η μεταβλητότητα είναι η τυπική απόκλιση των αποδόσεων για ένα συγκεκριμένο χρονικό διάστημα. Δηλαδή μετράμε πόσο αποκλίνουν οι αποδόσεις μίας επένδυσης σε σχέση με τον μέσο όρο των αποδόσεών της. Έτσι, αν οι τιμές κυμαίνονται με γρήγορους ρυθμούς σε καινούρια χαμηλά ή υψηλά σημεία έχουμε μεγάλη μεταβλητότητα και έτσι έχουμε μεγάλη απόκλιση από τον μέσο όρο. Στην αντίθετη περίπτωση όπου έχουμε πιο αργές εναλλαγές στις τιμές και φαίνεται να διατηρείται μία σχετική σταθερότητα έχουμε χαμηλή μεταβλητότητα. Η παρουσίαση της μεταβλητότητας γίνεται με την μορφή ποσοστού.

Συναντάμε δύο είδη μεταβλητότητας. Αυτά είναι η πραγματοποιημένη μεταβλητότητα (realized volatility) και η σιωπηρή μεταβλητότητα (implied volatility). Όταν μιλάμε για την πραγματοποιημένη μεταβλητότητα, συνήθως αναφερόμαστε στην ιστορική μεταβλητότητα, αφού χρησιμοποιούνται τιμές του παρελθόντος για ένα συγκεκριμένο χρονικό διάστημα για να έχουμε τον υπολογισμό της. Την χρησιμοποιούμε για να μπορέσουμε να δούμε, ποια θα μπορούσε να είναι η εξέλιξη της επένδυσης μελλοντικά, αν οι παράγοντες που συνέβαλαν στην δημιουργία της μεταβλητότητας εμφανιστούν και στο μέλλον. Αντίθετα η σιωπηρή μεταβλητότητα αφορά τις προσδοκίες της μεταβλητότητας. Χρησιμοποιούνται τα option price για τον υπολογισμό της για μια συγκεκριμένη χρονική περίοδο. Έτσι, μετά την πάροδο αυτής της περιόδου μπορούμε να καταλάβουμε αν αυτή η προσδοκία της μεταβλητότητας ήταν σωστή ή λανθασμένη.

Η μεταβλητότητα στην αγορά μπορεί να προκύψει από αρκετούς παράγοντες. Το πολιτικό σκηνικό και οι οικονομίες των χωρών μπορούν να παίξουν σημαντικό ρόλο. Τυχόν εκλογές ή κυρώσεις μιας κυβέρνησης σε εταιρίες επηρεάζουν αρκετά όπως και η πορεία της οικονομίας μιας χώρας αν βρίσκεται σε ανοδική πορεία, αν δεν υπάρχει ανεργία και δημιουργούνται νέες θέσεις εργασίας μπορούν να επηρεάσουν τις αγορές. Ακόμα η ατομική πορεία βιομηχανιών και εταιριών μπορεί να επηρεάσει. Η παρουσίαση των οικονομικών τους στοιχείων, η κυκλοφορία νέου προϊόντος ή η ανάκληση κάποιου άλλου, η διαρροή πληροφοριών, η συμπεριφορά αλλά και δηλώσεις υψηλά υφισταμένων μπορούν να επηρεάσουν.

Η μεταβλητότητα αποτελεί σημαντικό κομμάτι του τομέα των χρηματοοικονομικών. Πολλοί είναι αυτοί που προσπάθησαν να προβλέψουν την μεταβλητότητα του χρηματιστηρίου και κατ' επέκταση των αποδόσεών του. Οι Bhowmik και Wang συγκέντρωσαν και παρουσίασαν μεγάλο μέρος της υπάρχουσας βιβλιογραφίας που αφορά στην μεταβλητότητα. Η μεταβλητότητα είναι κάτι το φυσιολογικό και θα υπάρχει πάντα στις επενδύσεις και ιδιαίτερα στις μακροπρόθεσμες. Οφείλουν οι επενδυτές να το περιμένουν και να είναι προετοιμασμένοι να βρεθούν σε τέτοιες καταστάσεις, καθώς σε καταστάσεις υψηλής μεταβλητότητας δημιουργούνται και επενδυτικές ευκαιρίες που μπορεί να αποφέρουν μεγάλο και υψηλό κέρδος.

2.4 Μεταβλητότητα του Bitcoin

Από την στιγμή που το Bitcoin απέκτησε «φωνή» στον χρηματοοικονομικό κόσμο, αμέσως επόμενο ήταν να προσπαθήσουν να εξηγήσουν και στην συνέχεια να προβλέψουν την μεταβλητότητά του. Μία προσπάθεια έγινε με σύγκριση μοντέλων GARCH (Katsiampa, 2017). Τα δεδομένα που χρησιμοποιήθηκαν είναι οι καθημερινές τιμές που «έκλεινε» το Bitcoin από τον Ιούλιο του 2010 μέχρι και τον Οκτώβριο του 2016. Η καλύτερη προσαρμογή των δεδομένων έγινε με το μοντέλο AR-CGARCH, που είχε την πιο κοντινή πρόβλεψη της μεταβλητότητας.

Άλλη μία προσέγγιση για την πρόβλεψη μεταβλητότητας αλλά και της απόδοσης έγινε μέσω του όγκου συναλλαγών του κρυπτονομίσματος Bitcoin (Balcilar & Bouri & Gupta & Roubaud, 2017). Σε αυτήν την προσέγγιση χρησιμοποιήθηκε το τεστ αιτιότητας του Granger. Για την εφαρμογή του τα δεδομένα που χρησιμοποιήθηκαν είναι ο δείκτης του Bitcoin (Bitcoin Index) αλλά και ο όγκος των συναλλαγών του (αγορά-πώληση) από τον Δεκέμβριο του 2011 έως και τον Απρίλιο του 2016. Αυτήν η έρευνα είχε σαν συμπέρασμα ότι ο όγκος των συναλλαγών μπορεί να προβλέψει την απόδοση (εφόσον η αγορά κινείται στα κανονικά επίπεδα, χωρίς πολύ χαμηλές ή υψηλές τιμές) αλλά όχι όμως και την μεταβλητότητα.

Καθώς το Bitcoin έχει κεντρίσει το ενδιαφέρον της αγοράς, συνεχίζονται οι προσπάθειες για την πρόβλεψη της μεταβλητότητας. Στο άρθρο τους οι Aalborg, Molnár και de Vries (2019) προσπαθούν να βρουν τι μπορεί να εξηγήσει την τιμή, την μεταβλητότητα και τον όγκο συναλλαγών του εν λόγω κρυπτονομίσματος. Μελέτησαν πως γίνεται κάποιες

μεταβλητές να εξηγήσουν την μεταβλητότητα, την τιμή και τον όγκο των συναλλαγών. Χρησιμοποιήθηκαν ως δεδομένα για το συγκεκριμένο άρθρο ο δείκτης μεταβλητότητας (VIX), οι αποδόσεις του Bitcoin, οι συναλλαγές του αλλά και οι διευθύνσεις του (Bitcoin addresses). Επίσης χρησιμοποιήθηκε η πραγματική μεταβλητότητα, ο όγκος των συναλλαγών καθώς και την συχνότητα αναζήτησης του όρου «Bitcoin».Τα δεδομένα που χρησιμοποιήθηκαν αφορούσαν την περίοδο Μαρτίου 2012 έως και τον Μάρτιο του 2017. Διαπιστώθηκε ότι τυχόν αλλαγές στην τιμή του Bitcoin δεν μπορούν να προβλεφθούν και ότι η πραγματοποιημένη μεταβλητότητα είναι αρκετά προβλέψιμη βάσει των παλιότερων τιμών.

Στην πραγματική μεταβλητότητα του Bitcoin επικεντρώθηκαν μέσω άρθρου τους και οι Baur, Dimpfl (2017). Πραγματοποιούν μία εις βάθος ανάλυση της πραγματικής μεταβλητότητας του Bitcoin. Τα δεδομένα που χρησιμοποίησαν αφορούν την περίοδο του Ιανουαρίου 2014 έως και τον Ιανουάριο του 2017. Επικεντρώθηκαν στις αγορές που εμπεριέχουν το μεγαλύτερο μερίδιο στις ολικές συναλλαγές με Bitcoin και συμπεριλαμβάνουν το δολάριο (USD), το ευρώ (EUR) και το γιεν (CNY). Κατέληξαν στο γεγονός ότι λόγω της μεγάλης μεταβλητότητας το Bitcoin δεν μπορεί να θεωρηθεί νόμισμα. Ωστόσο μπορεί να σταθεί ως μία επένδυση αν λάβει κανείς υπόψη τις μεγάλες διακυμάνσεις στην τιμή του και τις πιθανές αποδόσεις που αυτό μπορεί να επιφέρει.

Ανάλυση της μεταβλητότητας έκαναν και οι Pichl και Kaizoji (2017). Επικεντρώθηκαν στην ανάλυση της μεταβλητότητας μέσα από την χρονοσειρά τιμών του Bitcoin. Χρησιμοποιήθηκαν οι τιμές του Bitcoin σε σχέση με βασικά νομίσματα όπως το δολάριο, το ευρώ και το γιέν και την μεταβλητότητα αυτών. Τα δεδομένα που μάζεψαν ήταν 88 ημερών, από 17-05-2017 έως και 12-08-2017. Πραγματοποίησαν μελέτη της μεταβλητότητας μέσα από την κατανομή της λογαριθμικής απόδοσης, την κατανομή του όγκου συναλλαγών, τις πιθανές εξισορροπητικές κερδοσκοπίες στις αγορές του Bitcoin, το μοντέλο της πραγματικής μεταβλητότητας και την εφαρμογή μίας μεθόδου νευρωνικών δικτύων για την πρόβλεψη των αποδόσεων χρησιμοποιώντας λογαρίθμους. Κατέληξαν ότι η χρονοσειρά τιμών του Bitcoin είναι πολύ πιο ασταθής σε σχέση με την ισοτιμία δολαρίου ευρώ, μπορεί να υπάρχουν εξισορροπητικές κερδοσκοπίες ωστόσο πρέπει να γίνει μία πιο ενδελεχής μελέτη. Τέλος η εφαρμογή των νευρωνικών δικτύων είχε ικανοποιητικά αποτελέσματα και είναι ικανή για την πρόβλεψη της ημερήσιας απόδοσης αν εφαρμοστούν πιο προχωρημένοι μέθοδοι της μηχανικής μάθησης.

2.5 Πρόβλεψη και μεταβλητότητα

Όπως αναφέρθηκε και παραπάνω η μεταβλητότητα του Bitcoin παίζει σημαντικό ρόλο για τους επενδυτές του συγκεκριμένου κρυπτονομίσματος, καθώς οποιαδήποτε προσπάθεια πρόβλεψής της θα τους οδηγήσει σε καλύτερα αποτελέσματα. Παρατηρούμε όμως ότι οι πρώτες προσεγγίσεις για την πρόβλεψη της μεταβλητότητας έγιναν κυρίως χρησιμοποιώντας οικονομετρικά μοντέλα. Ωστόσο αρκετοί είναι αυτοί που θεώρησαν ότι μπορούν να έχουν καλά αποτελέσματα ως προς την πρόβλεψη της μεταβλητότητας χρησιμοποιώντας την μηχανική μάθηση. Η μηχανική μάθηση ή αλλιώς Machine Learning, αποτελεί κλάδο της επιστήμης των υπολογιστών όπου μέσα από ανάπτυξη αλγορίθμων και εισάγοντας σε αυτούς δεδομένα, μπορεί να μας οδηγήσει σε προβλέψεις σχετικά με αυτά.

Μία απόπειρα, χρησιμοποιώντας Machine Learning (ML), έγινε από τους Jaquart, Dann και Weinhardt (2021). Χρησιμοποίησαν ML για την βραχυπρόθεσμη πρόβλεψη της αγοράς του Bitcoin. Η πρόβλεψη αυτήν αφορά χρονικά διαστήματα της τάξης του 1 λεπτού έως και 60 λεπτά. Τα δεδομένα που χρησιμοποίησαν αποκτήθηκαν από το Bloomberg, το Twitter και το Blockchain.com, και αφορούν την περίοδο από τον Μάρτιο του 2019 έως και τον Δεκέμβριο του 2019. Πήραν τα λεπτομερή δεδομένα των τιμών του bitcoin, του χρυσού, του πετρελαίου αλλά και τις λεπτομερείς αποδόσεις τους από τους δείκτες MSCI World, S&P 500 και VIX. Συμπεριέλαβαν και τις συναλλαγματικές ισοτιμίες του δολαρίου με το ευρώ, το κινέζικο γιέν και το ιαπωνικό γιέν. Αυτά τα δεδομένα πάρθηκαν από το Bloomberg. Από το Twitter συμπεριέλαβαν για το συγκεκριμένο χρονικό διάστημα όλα τα αγγλικά tweets που περιείχαν την «ετικέτα» (hashtag) bitcoin (#bitcoin). Ενώ από το Blockchain.com πήραν τα λεπτομερή δεδομένα για όλες τις συναλλαγές με bitcoin αλλά και την ανάπτυξη του mempool, δηλαδή του χώρου όπου εμπεριέχονται οι συναλλαγές με bitcoin οι οποίες δεν έχουν συμπεριληφθεί ακόμα σε κάποιο block. Χρησιμοποιήθηκαν διάφοροι αλγόριθμοι των Neural Networks, των decision trees καθώς και η μέθοδος ensemble models. Η αξιολόγησή τους έγινε βάσει της ακρίβειας που προβλέφθηκε για κάθε αλγόριθμο χρησιμοποιώντας μετά την διωνυμική κατανομή για να δούμε το ποσοστό επιτυχίας. Σε όλα τα μοντέλα η πρόβλεψη για πιθανή αύξηση ή μείωση φτάνει σε τιμές από 50.9% μέχρι 56% μέσω των οποίων η

πρόβλεψη της ακρίβειας (accuracy) τείνει να αυξάνεται για πρόβλεψη σε μεγαλύτερα χρονικά διαστήματα.

Οι Bergsli, Lind, Molnár και Polasic (2022) , λόγω της υψηλής μεταβλητότητας της τιμής του bitcoin, προσπάθησαν και αυτοί να την προβλέψουν. Θέλησαν να προβλέψουν την μεταβλητότητα για 1, 2, 5, 10 και 15 μέρες πιο μπροστά. Για να το επιτύχουν αυτό χρησιμοποίησαν αρκετά μοντέλα GARCH και 2 μοντέλα HAR και μετέπειτα έκαναν τις μεταξύ τους συγκρίσεις. Ο συνολικός αριθμός των παρατηρήσεων ανέρχεται στις 1720, δηλαδή από τον Ιανουάριο 2014 έως και τον Σεπτέμβριο 2018. Τα δεδομένα αυτά πάρθηκαν από το Bitstamp, μέσω του Bitstamp API και αναφέρονται κυρίως στο δολάριο, όπου η τιμή κλεισίματος του bitcoin είναι η τιμή που έχει πάρει στην τελευταία συναλλαγή κάθε ημέρας, καθώς οι συναλλαγές γίνονται όλο το εικοσιτετράωρο. Εξάγουν αυτές τις τιμές και τις μετατρέπουν σε λογαριθμικές αποδόσεις (log-returns). Για την σύγκριση των μοντέλων χρησιμοποίησαν το κριτήριο πληροφοριών Akiake (AIC), το κριτήριο πληροφοριών Bayesian (BIC), συναρτήσεις απώλειας όπως μέσο τετραγωνικό σφάλμα (MSE), μέσο απόλυτο σφάλμα (MAE), μέσο απόλυτο ποσοστό σφάλματος ή αλλιώς μέση απόλυτη ποσοστιαία απόκλιση (MAPE), την συνάρτηση απώλειας QLIKE που είναι κατάλληλη για την πρόβλεψη της μεταβλητότητας. Χρησιμοποιούν επίσης την διαδικασία του model confidence set (MCS) για να καθορίσουν αν τα διάφορα μοντέλα έχουν στατιστικά διαφορετικές δυνατότητες πρόβλεψης. Συμπέραναν ότι καλύτερη απόδοση πρόβλεψης από τα μοντέλα GARCH έχουν τα EGARCH και APARCH ενώ τα μοντέλα HAR έχουν καλύτερη απόδοση σε σχέση με τα μοντέλα GARCH για δεδομένα υψηλής συχνότητας.

Οι Zhang, He, Wen και Wang επιχείρησαν και αυτοί να προβλέψουν την μεταβλητότητα του bitcoin χρησιμοποιώντας μοντέλα παλινδρόμησης με κατώφλι (threshold regression model). Χρησιμοποίησαν για την πρόβλεψη το μοντέλο HAR-RV, που ειδικεύεται για την πραγματοποιημένη μεταβλητότητα και παραλλαγές αυτού ανάλογα με τα δεδομένα που εισάγουμε κάθε φορά. Τα δεδομένα που χρησιμοποιούνται ξεκινάν από τον Ιούλιο 2013 και τελειώνουν τον Σεπτέμβριο 2020. Αυτά αντλήθηκαν από το BitStamp, όπου περιέχονται οι συναλλαγές του Bitcoin στο νόμισμα του δολαρίου (USD) και από το bitcoincharts.com, όπου παίρνουν τις τιμές που είχε το bitcoin κάθε ημέρα. Οι τελευταίες 1500 παρατηρήσεις του δείγματος χρησιμοποιήθηκαν για την πρόβλεψη των μοντέλων. Η αξιολόγηση τους έγινε με το τεστ των Diebold-Mariano (D-M), όπου χρησιμοποιούνται οι

συναρτήσεις απώλειας QLIKE, MAPE και MSPE (μέσο τετραγωνικό ποσοστό σφάλματος) αλλά και με την διαδικασία του model confidence set (MCS). Συμπέραναν ότι τα μοντέλα HAR-RV είναι ικανά για την πρόβλεψη των αγορών Bitcoin, όπου από τα δεδομένα εκπαίδευσης στα μοντέλα παλινδρόμησης και μέσα από τα lagged returns πάνω και κάτω από το κατώφλι υπάρχουν δυνατότητες πρόβλεψης της μεταβλητότητας όπως και από τα δεδομένα που δοκιμάστηκαν μέσα από τα τεστ των D-M και MCS υπάρχει δυνατότητα για την πρόβλεψη της μελλοντικής μεταβλητότητας.

Ένα ακόμα άρθρο για την πρόβλεψη της μεταβλητότητας του bitcoin παρουσιάστηκε από τους Shen, Wan και Leatham (2021). Στην δικιά τους δημοσίευση συγκρίνουν μοντέλα οικονομετρικών, με κυριότερο το μοντέλο GARCH, με μοντέλα του machine learning όπως είναι τα νευρωνικά δίκτυα και συγκεκριμένα τα επαναλαμβανόμενα νευρωνικά δίκτυα (RNN). Τα δεδομένα που χρησιμοποίησαν τα πήρα από το CoinMarketCap.com και ανήκουν στην περίοδο από τον Ιούλιο 2013 έως και τον Μάιο 2021 και ανέρχονται σε 2944 παρατηρήσεις. Επιλέγουν την χρονοσειρά αποδόσεων του bitcoin και μαζί με τις ημερήσιες τιμές ανοίγματος του καθώς και με τα υψηλά, τα χαμηλά και τις τιμές κλεισίματος υπολογίζουν την πραγματοποιημένη μεταβλητότητα του. Η αξιολόγηση των δεδομένων που χρησιμοποιήθηκαν για να δοκιμαστούν τα μοντέλα έγινε με την ρίζα μέσου τετραγωνικού σφάλματος (RMSE) και με το μέσο απόλυτο σφάλμα (MAE) ενώ για την αποδοτικότητα της διαχείρισης κινδύνου χρησιμοποιούν το VaR (Value-at-Risk / Αξία σε κίνδυνο). Κατέληξαν ότι το μοντέλο RNN είναι ανώτερο για την πρόβλεψη της χρονοσειράς σε σχέση με τα οικονομετρικά μοντέλα και σύμφωνα με το MAE προβλέπει καλύτερα την ακρίβεια. Σε συνδυασμό με το γεγονός ότι δεν τα πήγε καλά στο RMSE σε σχέση με τα οικονομετρικά μοντέλα συμπεραίνουν ότι το μοντέλο RNN είναι λιγότερο αποτελεσματικό στην αντίληψη ακραίων γεγονότων. Ενώ για την αποδοτικότητα στην διαχείριση κινδύνων μέσω του VaR είναι ανώτερα τα οικονομετρικά μοντέλα. Έτσι συμπέραναν ότι τα οικονομετρικά μοντέλα είναι ανώτερα για την ανάλυση ακραίων συνθηκών στην αγορά ενώ το machine learning είναι ιδανικότερο για τις λιγότερο μεταβαλλόμενες συνθήκες στην αγορά.

Έρευνα για την μεταβλητότητα και την συναλλαγή κρυπτονομισμάτων έκαναν και οι Sebastiao και Godinho (2021). Χρησιμοποιώντας αλγορίθμους machine learning επιχειρούν να εξετάσουν την προβλεψιμότητα και την πιθανή κερδοφορία των τριών σημαντικών κρυπτονομισμάτων, του bitcoin, του ethereum και του litecoin. Για την

έρευνά τους χρησιμοποίησαν γραμμικά μοντέλα, μοντέλα τυχαίων δασών (random forests / RFs) και μοντέλα μηχανών διανυσμάτων υποστήριξης (support vector machines / SVMs). Από αυτά τα μοντέλα θέλουν να πάρουν όχι μόνο την πρόβλεψη της απόδοσης των κρυπτονομισμάτων αλλά και πιθανά στοιχεία είτε για αγορά είτε για πώληση. Το χρονικό διάστημα άντλησης του δείγματος των δεδομένων ξεκινάει τον Αύγουστο 2015 και τελειώνει τον Μάρτιο 2019. Το δείγμα αυτό αποτελείται από 1305 παρατηρήσεις. Οι πληροφορίες συναλλαγών, οι τιμές κλεισίματος, τα υψηλά και τα χαμηλά, ο ημερήσιος όγκος συναλλαγών αλλά και η χρηματιστηριακή κεφαλαιοποίηση προέρχονται από την ιστοσελίδα CoinMarketCap. Η αναπαράσταση των τιμών των δεδομένων γίνεται στο αμερικανικό δολάριο. Η αξιολόγηση τους έγινε με το ποσοστό επιτυχίας που έβγαζε το κάθε μοντέλο αλλά και με το μέσο απόλυτο σφάλμα, την ρίζα μέσου τετραγωνικού σφάλματος και το U^2 του Thiel. Χρησιμοποίησαν αρκετά μοντέλα για να βγάλουν συμπεράσματα και μόνο σε μία περίπτωση βρήκαν 5 μοντέλα να ταυτίζουν τα αποτελέσματά τους για τα κρυπτονομίσματα ethereum και litecoin. Έτσι πιστεύουν πως μέσω του machine learning και με σωστά δεδομένα υπάρχει η δυνατότητα πρόβλεψης της κίνησης των κρυπτονομισμάτων για την εύρεση της κατάλληλης στρατηγικής στις συναλλαγές ακόμα και αν οι συνθήκες στην αγορά είναι δυσμενείς.

Ακόμα μία προσέγγιση για την πρόβλεψη της μεταβλητότητας του bitcoin έγινε από τους Guo, Bifet και Antulov-Fantulin (2019). Μελέτησαν το πρόβλημα της βραχυπρόθεσμης πρόβλεψης της μεταβλητότητας του bitcoin μέσα από την μεταβλητότητά του μέσα στον χρόνο αλλά και από το βιβλίο παραγγελιών (order book) των αγορών και των πωλήσεών του. Χρησιμοποιώντας τα δεδομένα της μεταβλητότητας και τα στοιχεία από το βιβλίο παραγγελιών προτείνουν και δημιουργούν temporal mixture μοντέλα και τα συγκρίνουν με βασικά στατιστικά και machine learning μοντέλα. Κάποια από αυτά τα μοντέλα είναι τα EWMA (exponential weighted moving average), GARCH, BEGARCH, STR (structural time series), ARIMA (autoregressive and integrated moving average), RF, XGT (extreme gradient boosting), ENET (elastic-net), GP (gaussian process based regression),LSTMs (long short-term memory recurrent neural network), STRX και ARIMAX. Τα δεδομένα για την μεταβλητότητα και για το βιβλίο παραγγελιών αφορούν την περίοδο του Σεπτεμβρίου 2015 μέχρι και τον Απρίλιο 2017 και εμπεριέχουν 13730 ωριαίες παρατηρήσεις μεταβλητότητας και 701892 στιγμιότυπα του βιβλίου παραγγελιών. Τα δεδομένα αυτά τα πήραν από την ιστοσελίδα OKCoin. Βάσει των δεδομένων που

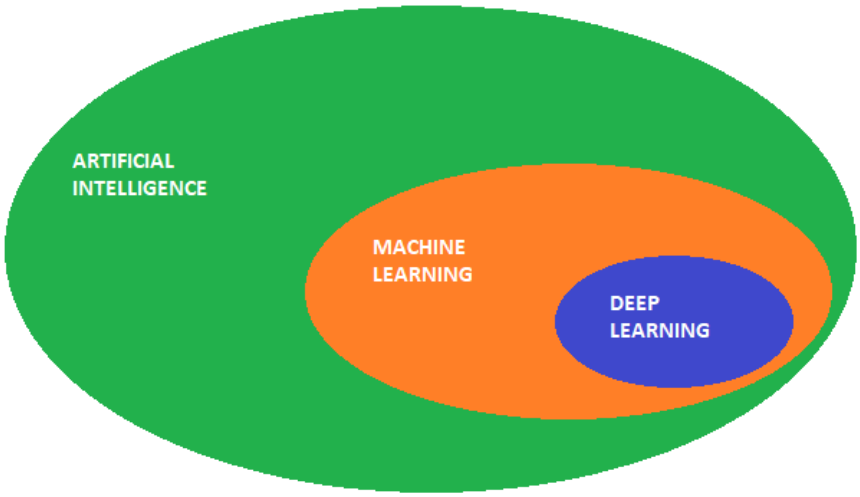
χρησιμοποίησαν για τα temporal mixture μοντέλα που έφτιαξαν και τις συγκρίσεις με τα μοντέλα στατιστικών και machine learning συμπέραναν ότι έχουν μεγαλύτερη ακρίβεια στην εκμάθηση δεδομένων, ερμηνεύουν καλύτερα την επίδραση του βιβλίου παραγγελιών σε σχέση με την μεταβλητότητα, έχουν ευρωστία και προσαρμοστικότητα όσο αναφορά τα χρονομεταβλητά δεδομένα και μπορεί να είναι χρήσιμο σαν ένα ευέλικτο και γενικό πλαίσιο για την πρόβλεψη και την ερμηνεία των δεδομένων bitcoin.

Εξαιτίας της υψηλής μεταβλητότητας των κρυπτονομισμάτων και συγκεκριμένα του bitcoin, οι Cocco, Tonelli και Marchesi (2021) επιχειρούν να προβλέψουν τις τιμές κλεισίματος του bitcoin. Στην προσπάθειά τους αυτήν χρησιμοποιούν μεθόδους machine learning. Επέλεξαν τα νευρωνικά δίκτυα και συγκεκριμένα τα BNN (Bayesian neural network), FFNN (Feed Forward neural network), LSTMNN (Long Short Term Memory neural network). Αρχικά γίνεται αξιολόγηση σε πλαίσιο ενός επιπέδου και στην συνέχεια σε πλαίσιο δύο επιπέδων μέσω του SVR (Support Vector Regression). Χρησιμοποίησαν τις ημερήσιες τιμές κλεισίματος τόσο του bitcoin όσο και του ethereum. Η περίοδος άντλησης των δεδομένων αποτελείται από 1216 τιμές και αφορά την περίοδο Ιανουαρίου 2017 μέχρι και τον Απρίλιο 2020. Η αξιολόγηση των παραπάνω έγινε με την συνάρτηση MAPE. Συμπέραναν ότι υψηλότερη επίδοση έχουν τα μοντέλα των δύο επιπέδων εκτός από το BNN, με καλύτερη απόδοση στο ένα επίπεδο. Η τάξη μεγέθους του μέσου απόλυτου ποσοστιαίου σφάλματος για το BNN έρχεται σε συμφωνία με τιμές που εμφανίστηκαν σε πρόσφατες εργασίες και δημοσιεύσεις.

2.6 Machine Learning

Διεθνώς είναι γνωστή ως Machine Learning, ωστόσο στην ελληνική βιβλιογραφία θα την βρούμε πολλές φορές και ως μηχανική μάθηση. Τι είναι όμως η μηχανική μάθηση; Ξεκινάει από το 1959, όπου ο Άρθουρ Σάμουελ ορίζει την μηχανική μάθηση ως εξής: «Πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μαθαίνουν, χωρίς να έχουν ρητά προγραμματιστεί». Οπότε μηχανική μάθηση είναι ένα υποπεδίο της επιστήμης των υπολογιστών που βασίζεται στην ανάπτυξη και την δημιουργία αλγορίθμων, οι οποίοι να μπορούν να μαθαίνουν από τα δεδομένα και μέσα από αυτά να κάνουν προβλέψεις. Έτσι, μέσα από του αλγορίθμους αυτούς, δημιουργούνται αναλυτικά μοντέλα για να

οδηγηθούμε σε προβλέψεις ή αποφάσεις. Είναι ένα κομμάτι της τεχνητής νοημοσύνης, ίσως και από τα κυριότερα. Αυτό συμβαίνει γιατί βασίζεται στην ιδέα ότι τα συστήματα μπορούν να μαθαίνουν από τα δεδομένα αλλά και να προσδιορίζουν πρότυπα και να λαμβάνουν τυχόν αποφάσεις με την ελάχιστη δυνατή ανθρώπινη παρέμβαση. Μπορούμε να πούμε ότι εφόσον η τεχνητή νοημοσύνη είναι η επιστήμη που αφορά την μίμηση των ανθρώπινων δυνατοτήτων, η μηχανική μάθηση είναι ένα τμήμα της που βοηθά στην εκπαίδευση μιας μηχανής στο να μαθαίνει. Χαρακτηριστικό της εξάρτησης που υπάρχει μεταξύ τους φαίνεται παρακάτω Εικόνα 1.



Εικόνα 1: Σχέση μεταξύ τεχνητής νοημοσύνης, μηχανικής μάθησης, βαθιάς μάθησης

Οι κύριες μέθοδοι μηχανικής μάθησης είναι τρεις. Αυτές είναι η επιβλεπόμενη μάθηση (Supervised Learning), η μη-επιβλεπόμενη μάθηση (Unsupervised Learning) και η ενισχυτική μάθηση (Reinforcement Learning). Οι αλγόριθμοι στην επιβλεπόμενη μάθηση δέχονται κάποια δεδομένα. Ένα δείγμα από αυτά τα δεδομένα χρησιμοποιούνται για την εκπαίδευση των αλγορίθμων. Αυτά τα δεδομένα συνήθως αναφέρονται ως training data και βοηθούν στην δημιουργία και στην εκπαίδευση του κατάλληλου μαθηματικού

μοντέλου κάθε φορά για να έχουμε τις προβλέψεις ή τις αποφάσεις. Τα εναπομείναντα δεδομένα ονομάζονται και testing data. Μετά την εκπαίδευση του, ο αλγόριθμος, χρησιμοποιεί αυτά τα δεδομένα χωρίς τα αποτελέσματά που αναμένονται, για να ελέγξουμε αν έχει διδαχθεί ο αλγόριθμος να βρίσκει τα σωστά αποτελέσματα. Έτσι οι αλγόριθμοι στην αρχή μαθαίνουν ποια είναι τα αποτελέσματα που θέλουμε βάση των εισόδων και των γνωστών εξόδων τους που έγιναν για την εκπαίδευση και στην συνέχεια ελέγχουμε και μετράμε την αποδοτικότητα για τις εισόδους που δίνουμε χωρίς να είναι γνωστές οι έξοδοι που πρέπει να έχουμε. Γνωστές μέθοδοι της επιβλεπόμενης μάθησης είναι η ταξινόμηση και η αναδρομή με σκοπό να κάνουμε προβλέψεις. Συνήθως χρησιμοποιείται για να προβλέπει αν κάποιες συναλλαγές μέσω πιστωτικών καρτών είναι έγκυρες ή αποτελούν απάτη, να κάνει προβλέψεις για τον καιρό, να προβλέψει τιμές προϊόντων ή μετοχών του χρηματιστηρίου, για αναγνώριση εικόνων και αντικειμένων.

Η μη επιβλεπόμενη μάθηση αντίθετα, χρησιμοποιεί δεδομένα που δεν έχουν κάποιο ιστορικό. Ο αλγόριθμος μάθησης πρέπει να βρει την δομή των δεδομένων και ποια μπορεί να είναι η πιθανή τους σχέση. Γνωστές μέθοδοι της μη επιβλεπόμενης μάθησης είναι η ομαδοποίηση των δεδομένων με K-Means, η ιεραρχική ομαδοποίηση (hierarchical clustering) και η πιθανολογική ομαδοποίηση (probabilistic clustering). Χρησιμοποιείται στο targeted marketing και την διαφήμιση, στην ανίχνευση ανωμαλιών αν σύνολο δεδομένων εμφανίζει διαφορετική συμπεριφορά.

Στην ενισχυτική μάθηση οι αλγόριθμοι μαθαίνουν μέσα από τις δοκιμές και τα σφάλματα που θα προκύψουν για να δουν ποιες είναι οι ενέργειες που έχουν τα καλύτερα δυνατά αποτελέσματα. Βασίζεται στον υπεύθυνο λήψης των αποφάσεων, στο περιβάλλον και τις ενέργειες που θα αποφασίσει ο υπεύθυνος. Εφαρμόζεται στην ρομποτική, στο gaming, σε αποφάσεις που αφορούν πραγματικό χρόνο.

Η μηχανική μάθηση συνδέεται τόσο με την εξαγωγή δεδομένων (Data Mining), όσο και με την βαθιά μάθηση (Deep Learning). Όλα ανήκουν στον τομέα του Data Science. Ωστόσο εμπεριέχουν κάποιες διαφορές. Η εξαγωγή δεδομένων ανήκει στην μη επιβλεπόμενη μάθηση. Αφορά την μελέτη, την αποθήκευση και τον χειρισμό μεγάλου όγκου δεδομένων. Στην εξαγωγή δεδομένων χρησιμοποιούνται πολλές και διαφορετικές μέθοδοι για εξαγωγή πληροφοριών από τα δεδομένα. Χρησιμοποιεί μεθόδους στατιστικής αλλά και μηχανική μάθηση για την αναγνώριση διάφορων προτύπων και σχεδίων. Μπορεί να θεωρηθεί ότι είναι ένα εργαλείο το οποίο χρησιμοποιείται από τον άνθρωπο.

Η μηχανική μάθηση αντίθετα προσπαθεί να έχει την ελάχιστη δυνατή παρουσία του ανθρώπινου στοιχείου. Αφορά την συνεχή εύρεση αλγορίθμων και την εκπαίδευση τους μέσα από δεδομένα για να έχουμε τα καλύτερα δυνατά αποτελέσματα. Προσπαθεί δηλαδή να κάνει τις μηχανές «έξυπνες» χωρίς να χρειάζεται να επεμβαίνει συνέχεια ο άνθρωπος.

Και οι δύο μέθοδοι χρησιμοποιούν τους ίδιους αλγορίθμους για την εύρεση προτύπων, ωστόσο τα επιθυμητά αποτελέσματα τους διαφέρουν. Αρχικά το DM προϋπήρχε του ML καθώς το DM εμφανίστηκε περίπου την δεκαετία του ’30 και ήταν γνωστό ως KDD (Knowledge Discovery in Databases). Το ML αντίθετα εμφανίστηκε περίπου την δεκαετία του ’50. Το DM αφορά μεγάλο όγκο δεδομένων για να μπορέσει να βρει πρότυπα και σχέδια να αναζητήσει συγκεκριμένα αποτελέσματα ή προβλέψεις ενώ αντίθετα στο ML έχουμε επεξεργασμένα δεδομένα με σκοπό την κατανόηση και την εκπαίδευση στις δοθείσες παραμέτρους. Τέλος, η εξόρυξη δεδομένων είναι μια διαδικασία που ουσιαστικά αποτελείται από δύο στοιχεία. Αυτά είναι η βάση δεδομένων και η μηχανική μάθηση. Καταλαβαίνουμε λοιπόν ότι ενώ η εξόρυξη δεδομένων χρειάζεται την μηχανική μάθηση, δεν μπορούμε να πούμε ότι το αντίθετό είναι πάντα απαραίτητο, δηλαδή η μηχανική μάθηση να χρειάζεται την εξόρυξη δεδομένων. Αυτό μπορεί να συμβεί αν θέλουμε να βρούμε συνδέσεις και συσχετίσεις στις πληροφορίες που προέκυψαν από την εξόρυξη δεδομένων.

Το Deep Learning είναι μία ειδικευμένη υποκατηγορία του Machine Learning. Το DL χρησιμοποιεί σύνθετους σε δομή αλγορίθμους σε μια προσπάθεια να μοντελοποιηθεί ο ανθρώπινος εγκέφαλος. Για να το πετύχουν αυτό έπρεπε να εξελιχθούν πολύ οι αλγόριθμοι του machine learning. Αυτό γιατί ήθελαν να πετύχουν μια ανάλυση δεδομένων με μία λογική όμοια με αυτήν που θα εφάρμοζε ένας άνθρωπος για να βγάλει συμπεράσματα. Για την επίτευξη αυτού σημαντικό ρόλο έπαιξε τα τελευταία χρόνια τόσο η ανάπτυξη του λογισμικού (software) όσο και των συσκευών (hardware) για την αύξηση της υπολογιστικής δύναμης. Ένας τέτοιος αλγόριθμος είναι και Νευρωνικά Δίκτυα, που θα δούμε αργότερα σε αυτήν την Διπλωματική Εργασία.

Σήμερα οι ποσότητες των δεδομένων αυξάνονται συνεχώς. Με αυτήν την αύξηση οι επιχειρήσεις και οι βιομηχανίες στρέφονται όλο και περισσότερο στην μηχανική μάθηση. Με την μηχανική μάθηση μπορούν και παράγουν γρήγορα, αυτόματα και αποδοτικά μοντέλα για να αναλύουν κάθε φορά μεγαλύτερο όγκο δεδομένων και πιο περίπλοκών κάθε φορά. Έχουμε τρομερά βήματα εξέλιξης και έτσι οι πληροφορίες από αυτά τα

δεδομένα προκύπτουν ταχύτερα και τα αποτελέσματα είναι πιο ακριβή. Αυτό αυξάνει τις πιθανότητες κάθε οργανισμού που έχει στραφεί στην μηχανική μάθηση στο να εντοπίζει επικερδείς ευκαιρίες, να αποφεύγει κινδύνους. Οι επιχειρήσεις ακόμα μπορούν να εργάζονται πιο αποτελεσματικά ή και να αποκτούν πλεονεκτήματα έναντι των ανταγωνιστών τους, καθώς η συλλογή των πληροφοριών από τα δεδομένα που χρησιμοποιούνται, δύναται να γίνει σε πραγματικό χρόνο. Έτσι, είναι εύκολο να καταλάβουμε πως οι εφαρμογές της μηχανικής μάθησης είναι αρκετές. Κάποιες από αυτές είναι η αναγνώριση ομιλίας και γραφικού χαρακτήρα, η διαδικτυακή διαφήμιση, η βιοπληροφορική, ο εντοπισμός διαδικτυακής απάτης και απάτης πιστωτικής κάρτας, σε ιατρικές διαγνώσεις, στην ρομποτική, στο μάρκετινγκ, στις μηχανές αναζήτησης, στην οικονομία, στην υπολογιστική όραση, στην χρηματιστηριακή ανάλυση.

2.6.1 Βιβλιοθήκες για Machine Learning

Τον πρώτο καιρό, όταν κάποιος ήθελε να εφαρμόσει μηχανική μάθηση, έπρεπε να γράψει ο ίδιος τον κώδικα. Στον κώδικα κάθε φορά έπρεπε να γραφτούν όλοι οι αλγόριθμοι αλλά και οι εξισώσεις και οι τύποι μαθηματικών και στατιστικής που ήταν απαραίτητοι για την σωστότερη λειτουργία. Όλο αυτό το γεγονός έκανε την διαδικασία της μηχανικής μάθησης αρκετά χρονοβόρα, κουραστική και πολλές φορές αναποτελεσματική. Σήμερα όμως ισχύει το αντίθετο. Με την εξέλιξη των γλωσσών προγραμματισμού έχουν εξελιχθεί και τα εργαλεία που είναι διαθέσιμα για την κάθε μία. Ένα τέτοιο εργαλείο είναι και οι βιβλιοθήκες. Οι βιβλιοθήκες περιέχουν ένα σύνολο λειτουργιών και διαδικασιών. Αυτές οι λειτουργίες και οι διαδικασίες είναι άμεσα και εύκολα διαθέσιμες για χρήση οποιαδήποτε στιγμή. Αποτελούν αναπόσπαστο κομμάτι των προγραμματιστών καθώς τους βοηθά να συντάξουν πολύπλοκα προγράμματα χωρίς να χρειάζεται ταυτόχρονα να γράψουν πολύ κώδικα. Υπάρχουν πολλών ειδών βιβλιοθήκες και για διάφορα αντικείμενα. Κάποια από αυτά είναι η επεξεργασία κειμένου, βιβλιοθήκες για γραφικά, για χειρισμό και επεξεργασία δεδομένων και για επιστημονικούς υπολογισμούς. Ωστόσο παρακάτω θα δούμε τις βιβλιοθήκες που χρησιμοποιούνται περισσότερο στην μηχανική μάθηση και συγκεκριμένα βιβλιοθήκες της γλώσσας προγραμματισμού Python.

NumPy

Η NumPy είναι μία από τις πιο δημοφιλείς βιβλιοθήκες της γλώσσας προγραμματισμού Python. Μέσα από την βιβλιοθήκη NumPy, μπορούμε να επεξεργαστούμε πολυδιάστατες σειρές και πίνακες είτε δημιουργώντας τους είτε κάνοντας πράξεις. Η NumPy παρέχει την δυνατότητα αποθήκευσης διάφορων τύπων δεδομένων σε πολυδιάστατο χώρο. Αυτό οφείλεται στο γεγονός ότι παρέχει μία μεγάλη συλλογή μαθηματικών συναρτήσεων. Αποτελεί σημαντική βοήθεια για την εφαρμογή του Machine Learning όπου χρειάζονται περίπλοκοι υπολογισμοί. Είναι πολύ χρήσιμη σαν βιβλιοθήκη για γραμμική άλγεβρα, μετασχηματισμούς Fourier και για διάφορες δυνατότητες τυχαίων αριθμών. Δεν είναι τυχαίο άλλωστε που ορισμένες βιβλιοθήκες υψηλού επιπέδου όπως είναι η TensorFlow χρησιμοποιούν εσωτερικά την NumPy.

SciPy

Άλλη μία βιβλιοθήκη που είναι γνωστή σε όσους εφαρμόζουν Machine Learning. Η SciPy εμπεριέχει την ενότητα της NumPy για την επεξεργασία πινάκων. Άλλες ενότητες της βιβλιοθήκης SciPy εμπεριέχουν γρήγορους μετασχηματισμούς Fourier, βελτιστοποίηση εικόνας, ολοκληρώματα, παρεμβολές, γραμμική άλγεβρα, επίλυση συνήθων διαφορικών εξισώσεων, επεξεργασία εικόνας.

Scikit-learn

Αρκετά σημαντική και εξίσου δημοφιλής είναι και η βιβλιοθήκη Scikit-learn. Η δημιουργία της βασίστηκε στις βιβλιοθήκες NumPy, SciPy και Matplotlib. Η Scikit-learn εμπεριέχει και υποστηρίζει την πλειοψηφία των αλγορίθμων μηχανικής μάθησης. Αυτοί οι αλγόριθμοι μπορεί να ανήκουν είτε στην επιβλεπόμενη μάθηση είτε στην μη-επιβλεπόμενη μάθηση. Επιπλέον η βιβλιοθήκη Scikit-learn μπορεί να χρησιμοποιηθεί ακόμα και για εξόρυξη δεδομένων αλλά και για ανάλυση δεδομένων. Αυτό την κάνει να είναι ένα αρκετά σημαντικό εργαλείο για κάποιον που ξεκινάει τώρα να μαθαίνει και να δουλεύει στην μηχανική μάθηση. Ακόμα μέσα από την Scikit-learn κάποιος μπορεί να προεπεξεργαστεί δεδομένα, να διαλέγει τις παραμέτρους των μοντέλων και να τις συγκρίνει και να τις επιβεβαιώνει, να κάνει κατηγοριοποίηση (classification),

συσταδοποίηση (clustering), μείωση διάστασης (dimensionality reduction) και παλινδρόμηση (regression).

Theano

Μία ακόμα βιβλιοθήκη που χρησιμοποιείται στο Machine Learning είναι και η Theano. Είναι ακόμα μία βιβλιοθήκη, η δημιουργία της οποίας βασίστηκε στην NumPy. Η Theano θεωρείται μία από τις ταχύτερες βιβλιοθήκες για Machine Learning. Προσφέρει επίσης αρκετά στον ορισμό, στην αξιολόγηση και στην βελτιστοποίηση των μαθηματικών συναρτήσεων όσον αναφορά τους πολυδιάστατους πίνακες για την καλύτερη δυνατή απόδοσή τους. Η βιβλιοθήκη Theano μπορεί να το επιτυγχάνει αυτό καθώς αξιοποιεί με τον καλύτερο δυνατό τρόπο τον επεξεργαστή (CPU) και την κάρτα γραφικών (GPU) ενός υπολογιστή. Χρησιμοποιείται εκτενώς για δοκιμές μονάδων (unit-testing) και για αυτο-επαλήθευση (self-verification) για να βρίσκει τους διάφορους τύπους λαθών. Έτσι, η Theano είναι μία πολύ ισχυρή βιβλιοθήκη που χρησιμοποιείται για μεγάλης κλίμακας υπολογισμούς σε επιστημονικά project. Ωστόσο είναι απλή και προσβάσιμη στον καθένα που θέλει να την χρησιμοποιήσει για κάποιο δικού του έργο.

TensorFlow

Η TensorFlow είναι μια δημοφιλής βιβλιοθήκη ανοιχτού κώδικα που αναπτύχθηκε από την Google Brain Team της Google. Όπως φαίνεται και από το όνομα η TensorFlow αφορά ένα πλαίσιο στο οποίο ορίζουμε και τρέχουμε υπολογισμούς που αφορούν τανυστές (tensors). Χρησιμοποιείται στην βαθιά μάθηση καθώς εκπαιδεύει και εκτελεί νευρωνικά δίκτυα (deep neural networks). Η TensorFlow διαθέτει και ένα διαδικτυακό εργαλείο που λέγεται TensorBoard. Αυτό βοηθά του προγραμματιστές να οπτικοποιήσουν μοντέλα με τις παραμέτρους, τις κλίσεις και τις αποδόσεις τους. Πέρα από βιβλιοθήκη, η TensorFlow αποτελεί ένα δημοφιλές πλαίσιο για ανάπτυξη ισχυρών μοντέλων μηχανικής μάθησης καθώς προσφέρει υποστήριξη μέσα από ένα μεγάλο εύρος εργαλείων. Τέλος επιδεικνύει την ευελιξία της από το γεγονός ότι μπορεί να τρέξει από μεγάλο εύρος CPUs, GPUs και TPUs (Tensor Processing Units).

Keras

Μία ακόμα βιβλιοθήκη ανοιχτού κώδικα είναι και η Keras. Η Keras χρησιμοποιείται και αυτήν για βαθιά μάθηση. Αποτελεί υψηλού επιπέδου API (Διεπαφή Προγραμματισμού Εφαρμογών) νευρωνικών δικτύων. Μπορεί να τρέξει πάνω από TensorFlow, CNTK (Microsoft Cognitive Toolkit) ή Theano. Η βιβλιοθήκη Keras μπορεί να τρέξει χωρίς προβλήματα τόσο με CPU όσο και με GPU. Ενδείκνυται για αρχάρισους που θέλουν να σχεδιάσουν και να στήσουν το πρώτο τους νευρωνικό δίκτυο. Επίσης η Keras παρέχει υποστήριξη και για τα convolutional neural networks (CNN – Συνελικτικά νευρωνικά δίκτυα) και τα recurrent neural networks (RNN). Τέλος η βιβλιοθήκη Keras διαθέτει χαρακτηριστικά για να δουλέψει κάποιος με εικόνες αλλά και με εικόνες κειμένου (text images).

PyTorch

Η PyTorch είναι μία βιβλιοθήκη machine learning ανοιχτού κώδικα. Βασίστηκε στην ανοιχτού κώδικα βιβλιοθήκη machine learning Torch, που έχει γραφτεί στην γλώσσα προγραμματισμού C και χρησιμοποιείται από την γλώσσα Lua. Η βιβλιοθήκη PyTorch, όπως φαίνεται και από το όνομα χρησιμοποιείται στην Python και αναπτύχθηκε από την ομάδα του Facebook. Δεν είναι ακόμα το ίδιο δημοφιλής όπως η TensorFlow, ωστόσο περιέχει πολλά εργαλεία και επιμέρους βιβλιοθήκες που υποστηρίζουν την μηχανική όραση (computer vision), την επεξεργασία φυσικής γλώσσας (NLP – Natural Language Processing), δυναμικούς γράφους και γραφικές παραστάσεις και πολλά ακόμα προγράμματα machine learning. Επιτρέπει ακόμα στους χρήστες να κάνουν υπολογισμούς σε tensors με επιτάχυνση της GPU (GPU acceleration). Τέλος, η βιβλιοθήκη PyTorch είναι αρκετά φιλική για τους αρχάριους που θέλουν να ξεκινήσουν στο data science και το machine learning.

Pandas

Η ιδανική βιβλιοθήκη για ανάλυση δεδομένων είναι η Pandas. Η βιβλιοθήκη Pandas δεν μπορούμε να πούμε ότι σχετίζεται άμεσα με την μηχανική μάθηση αυτή καθ' αυτή, ωστόσο είναι απαραίτητη για την προετοιμασία και επεξεργασία του σετ δεδομένων. Μπορούμε να πούμε ότι κατά κάποιον τρόπο η Pandas είναι για την Python ότι το

Microsoft Excel για τα Windows. Παρέχει τρομερή διευκόλυνση σε ότι αφορά την εξαγωγή, την χρησιμοποίηση και την χειραγώγηση των δεδομένων. Χρησιμοποιεί είτε μονοδιάστατες (σειρές) είτε δισδιάστατες (dataframes) κατασκευές για να επιτύχει τα παραπάνω. Παρέχει αρκετές μαθηματικές πράξεις για να διευκολύνει τους χρήστες. Η βιβλιοθήκη Pandas εμπεριέχει επίσης μεθόδους για αναζήτηση, συνδυασμό και φιλτράρισμα των δεδομένων. Είναι τόσο μεγάλη η ευκολία χειρισμού των διαφόρων ειδών δεδομένων που μπορεί να επεξεργαστεί, που χρησιμοποιείται για δεδομένα μηχανικής, οικονομικών, επιστημονικών, στατιστικής αλλά και άλλων, είτε αυτά είναι ομοιογενή και ετερογενή δεδομένα είτε ταξινομημένα και αταξινόμητα δεδομένα χρονοσειράς.

Matplotlib

Η βιβλιοθήκη Matplotlib είναι και αυτήν μία βιβλιοθήκη που δεν σχετίζεται άμεσα με την μηχανική μάθηση. Είναι μία από τις δημοφιλέστερες βιβλιοθήκες για απεικόνιση των δεδομένων. Συγκεκριμένα είναι πολύ χρήσιμη αν ο χρήστης θέλει να απεικονίσει τα σχέδια ή τα μοτίβα που προκύπτουν από τα δεδομένα. Με την βιβλιοθήκη Matplotlib μπορούμε να απεικονίσουμε δισδιάστατα γραφήματα και γραφικές παραστάσεις. Διαθέτει ιστογράμματα, γραφήματα σφαλμάτων, ραβδογράμματα, διαγράμματα διασποράς και άλλα. Τέλος η βιβλιοθήκη Matplotlib διαθέτει και ένα κομμάτι που λέγεται pyplot, το οποίο διευκολύνει τους χρήστες να ελέγχουν το στυλ των γραμμών, να επεξεργάζονται τις ιδιότητες των γραμματοσειρών, να μορφοποιούν τους άξονες κ.ά..

2.7 Αλγόριθμοι Μηχανικής Μάθησης και Κατηγοριοποίηση

Η κατηγοριοποίηση είναι μία από τις βασικές τεχνικές της εξόρυξης δεδομένων. Μπορεί να την βρούμε και ως ταξινόμηση στην βιβλιογραφία και αποτελεί κομμάτι της επιβλεπόμενης μάθησης. Είναι μία διαδικασία κατά την οποία προσπαθούμε να αναγνωρίσουμε και να καταλάβουμε τα στοιχεία που μας δίνονται και στην συνέχεια να τα αναθέσουμε σε ένα προκαθορισμένο σύνολο κατηγοριών. Έτσι στόχος αυτής της διαδικασίας είναι η ανάπτυξη ενός μοντέλου το οποίο θα χρησιμοποιηθεί για να

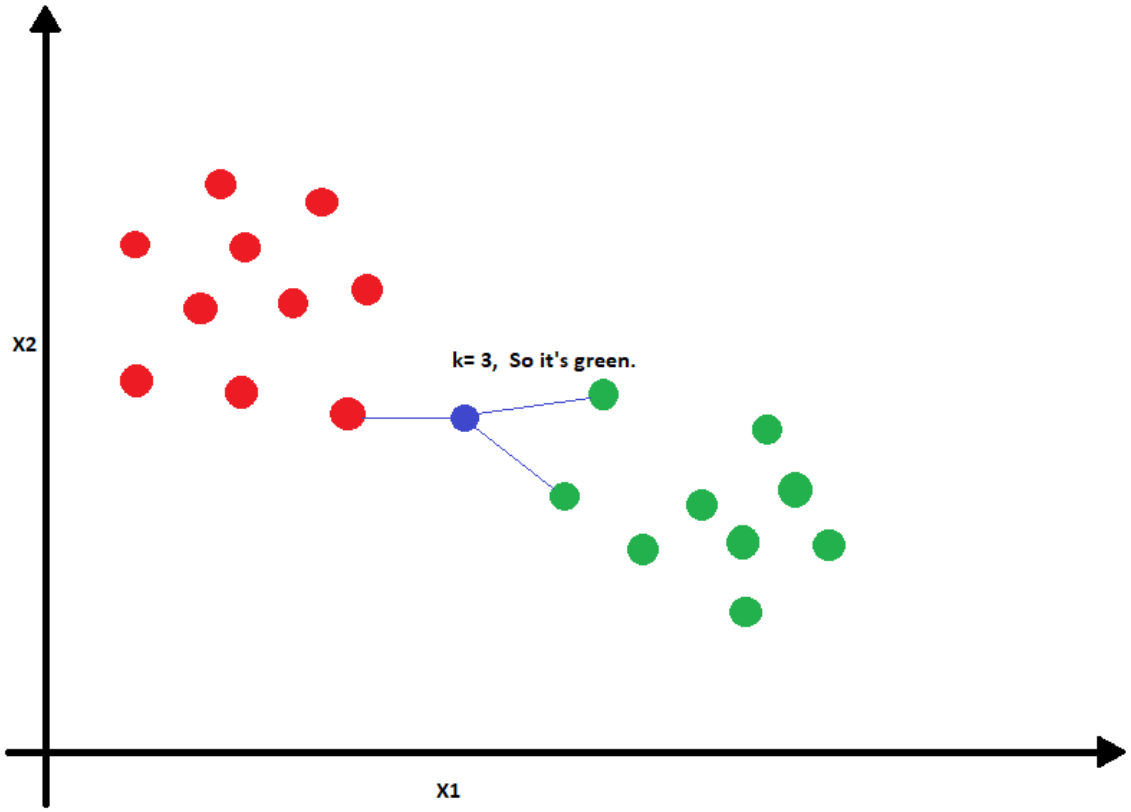
κατηγοριοποιήσουμε μελλοντικά δεδομένα. Έχει μεγάλη απήχηση και χρησιμοποιείται σε πολλούς τομείς, όπως στο να ξεχωρίζουμε τα email σε κατηγορίες spam ή non-spam, στην πρόβλεψη καρκινικών κυττάρων για το γεγονός αν είναι καλοήθη ή κακοήθη, αν οι συναλλαγές μέσω πιστωτικής κάρτας είναι νόμιμες ή όχι κ.ά.. Η κατηγοριοποίηση μπορούμε να πούμε ότι χωρίζεται σε δύο στάδια. Το πρώτο στάδιο αφορά την ανάπτυξη ή την δημιουργία ενός μοντέλου σύμφωνα με τα κατηγοριοποιημένα δεδομένα που ήδη έχουμε. Αυτά τα δεδομένα αποτελούν τα δεδομένα εκπαίδευσης (training data). Χρησιμοποιώντας κάποιον αλγόριθμο κατηγοριοποίησης, αναλύουμε αυτά τα δεδομένα και εκπαιδεύουμε τον αλγόριθμο με σκοπό να σχηματίσουμε ένα μοντέλο. Στην συνέχεια περνάμε στο δεύτερο στάδιο. Αυτό είναι η αξιολόγηση του μοντέλου. Παίρνουμε τα δεδομένα δοκιμής (testing data) και χρησιμοποιούμε το μοντέλο για να τα κατηγοριοποιήσουμε. Έπειτα συγκρίνουμε την κατηγορία που δημιουργήθηκε από το test data με την πρόβλεψη από το training data. Έτσι η ακρίβεια του μοντέλου υπολογίζεται ως το πλήθος των σωστών προβλέψεων προς το συνολικό πλήθος προβλέψεων. Σε αυτό το σημείο θα παρουσιαστούν κάποιοι αλγόριθμοι κατηγοριοποίησης που χρησιμοποιούνται στην μηχανική μάθηση και επιλέχτηκαν να εφαρμοστούν στο πλαίσιο αυτής της Διπλωματικής Εργασίας.

2.7.1 Αλγόριθμος K-NN

Ο K-NN είναι ένας μη-παραμετρικός επιβλεπόμενος αλγόριθμος κατηγοριοποίησης. Οι πρώτες ιδέες πάνω στον K-NN ξεκίνησαν το 1951 από τους Evelyn Fix και Joseph Hodges. Αργότερα ο Thomas Cover το 1967 επέκτεινε τις ιδέες τους. Χρησιμοποιεί αποστάσεις για να ταξινομήσει ή για να κάνει προβλέψεις σε σχέση με την κατηγορία στην οποία ανήκει κάθε μεμονωμένο στοιχείο των δεδομένων.

Ο K-NN μπορεί να χρησιμοποιηθεί τόσο για κατηγοριοποίηση όσο και για παλινδρόμηση. Στην πλειοψηφία του όμως επιλέγεται σαν αλγόριθμος κατηγοριοποίησης, αφού συνήθως λειτουργούμε με την υπόθεση ότι τα παρόμοια στοιχεία θα βρίσκονται κοντά μεταξύ τους. Όσον αφορά την κατηγοριοποίηση, το στοιχείο χωρίς κατηγορία που εξετάζουμε, τοποθετείται σε αυτήν στην οποία ανήκει η πλειοψηφία των γειτόνων του. Στην περίπτωση που θέλουμε να χρησιμοποιήσουμε τον K-NN για προβλήματα παλινδρόμησης τότε το στοιχείο που ψάχνουμε παίρνει την τιμή από τον μέσο όρο των τιμών των

γειτόνων. Χαρακτηριστικό είναι το παράδειγμα κατηγοριοποίησης που φαίνεται στην Εικόνα 2.



Εικόνα 2: Αναπαράσταση αλγορίθμου Knn

Το μπλε σημείο ψάχνει που ανήκει, με τα κόκκινα ή με τα πράσινα σημεία. Εφόσον οι 2 από τους 3 γείτονες ανήκουν στο πράσινο, τότε και αυτό καταλήγει στην πράσινη κατηγορία.

Ωστόσο, πριν ολοκληρωθεί η κατηγοριοποίηση, θα πρέπει να οριστεί και να υπολογιστεί η απόσταση. Υπάρχουν πολλές αποστάσεις που μπορούμε να χρησιμοποιήσουμε, αλλά οι συνήθεις που χρησιμοποιούνται είναι η Ευκλείδεια απόσταση (1), η απόσταση Manhattan (2), η απόσταση Minkowski (3), η απόσταση Hamming (4). Οι εξισώσεις για την καθεμία φαίνονται παρακάτω.

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

$$(2)\ Manhattan\ Distance = \ d(x,y) \ = \left(\sum_{i=1}^{m}|x_i - y_i|\right)$$

$$(3)\ Minkowski\ Distance = \ d(x,y) \ = \left(\sum_{i=1}^{n}|x_i - y_i|\right)^{1/p}$$

$$(4)\ Hamming\ Distance = \ D_H \ = \left(\sum_{i=1}^{k}|x_i - y_i|\right)$$

$$x = y \ \Rightarrow \ \ D = 0$$

$$x \neq y \ \Rightarrow D \neq 1$$

Το k στον αλγόριθμο K-NN συμβολίζει τον αριθμό των γειτόνων που θα ελέγχουμε κάθε φορά για να πραγματοποιήσουμε την κατηγοριοποίηση ενός στοιχείου. Αυτό σημαίνει ότι κάθε φορά το k πρέπει να είναι ένας θετικός ακέραιος αριθμός. Μεγάλη σημασία έχει λοιπόν η τιμή του k που θα επιλέξει ο χρήστης. Χαμηλές τιμές του k μπορεί να οδηγήσουν σε μεγάλη απόκλιση και χαμηλή μεροληψία ενώ υψηλές τιμές του k σε υψηλή μεροληψία και μικρή απόκλιση. Κατά γενική ομολογία η επιλογή τιμής για το k εξαρτάται από τον αριθμό των δεδομένων που εισάγουμε και συνήθως πρέπει να επιλέγουμε περιττό αριθμό για να αποφύγουμε τυχόν «ισοπαλίες» στην κατηγοριοποίηση.

Ο αλγόριθμος K-NN χρησιμοποιείται σε πολλούς και διάφορους τομείς, κυρίως σε θέματα κατηγοριοποίησης. Μπορούμε να το διαπιστώσουμε αρχικά από αρκετούς διαδικτυακούς τόπους τους οποίους επισκεπτόμαστε, ότι έχουν σύστημα συστάσεων ανάλογα με τι αναζητούμε ή «κλικάρουμε» κάθε φορά. Χαρακτηριστικό παράδειγμα είναι το Youtube, το Netflix, η Amazon κ.ά.. Ένα τέτοιο παράδειγμα δόθηκε από τους Adeniyi, Wei, Yongquan. Ακόμα χρησιμοποιείται για την προεπεξεργασία δεδομένων, όταν αυτά έχουν κάποιες τιμές που λείπουν, χρησιμοποιούμε τον K-NN για να τις προσδιορίσουμε. Χρησιμοποιείται επίσης για την αναγνώριση προτύπων δηλαδή για ταξινόμηση ψηφίων ή κειμένου, όπως φαίνεται και από την εργασία των Y. Wang, R.Wang, Li, Adu-Gyamfi, Tian, Zhu. Ακόμα ένας κλάδος που βρήκε χρήση ο αλγόριθμος K-NN είναι και αυτός των οικονομικών. Σε εργασία τους οι Mukid, Widiharih, Rusgiyono, Prahutama χρησιμοποιούν τον K-NN για να προσδιορίσουν οι τράπεζες το ρίσκο ενός δανείου βάσει των πιστωτικών στοιχείων των ατόμων ή των επιχειρήσεων που είναι να το λάβουν. Τέλος οι Nie και Song αναφέρουν για ανάλυση του χρηματιστηρίου που βασίζεται στην δομή του K-NN.

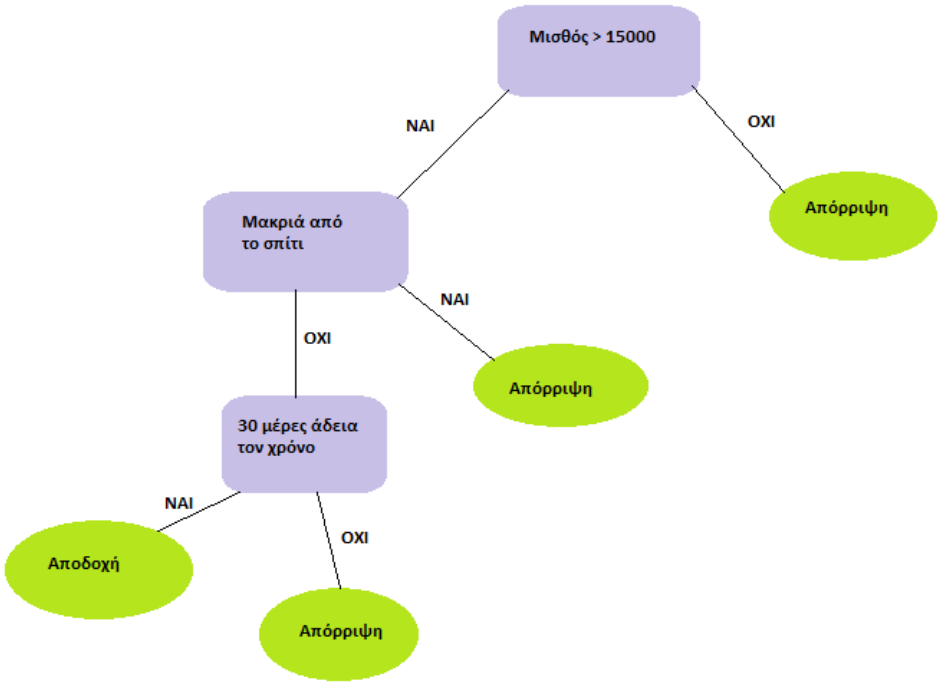
Ο αλγόριθμος K-NN έχει πλεονεκτήματα και μειονεκτήματα. Είναι από τους πρώτους αλγόριθμους που θα μάθει κάποιος αν ξεκινήσει να ασχολείται με το machine learning αφού είναι αρκετά εύκολος στην εφαρμογή του, χρειάζεται μόνο μία τιμή για το k και την επιλογή ποιας απόστασης θα χρησιμοποιηθεί και προσαρμόζεται εύκολα, αφού δεν χρειάζεται κάποια περίοδος εκπαίδευσης των δεδομένων αφού αποθηκεύει τα δεδομένα και μαθαίνει από αυτά την στιγμή που είναι να κάνει τις προβλέψεις και αυτό έχει σαν αποτέλεσμα να μπορούμε να προσθέσουμε σε αυτόν δεδομένα χωρίς να επηρεαστεί η ακρίβεια του. Ωστόσο δεν δουλεύει το ίδιο καλά αν ο αριθμός των δεδομένων είναι

μεγάλος καθώς πρέπει να υπολογίσει μεγάλο αριθμό αποστάσεων και έτσι επηρεάζει την απόδοσή του. Επίσης έχει πρόβλημα αν τα δεδομένα μας είναι σε πολλές διαστάσεις οπότε δυσκολεύεται στον υπολογισμό της απόστασης σε κάθε διάσταση και πολλές φορές αν τα δεδομένα είναι πολλά μπορεί να χρειαστεί κανονικοποίηση των δεδομένων για να έχουμε πιο σωστές προβλέψεις. Τέλος πρέπει κάθε φορά να βρούμε τις σωστές τιμές για το k για να μην εμφανιστούν φαινόμενα overfit και underfit για τα δεδομένα.

2.7.2 Αλγόριθμος Random Forest

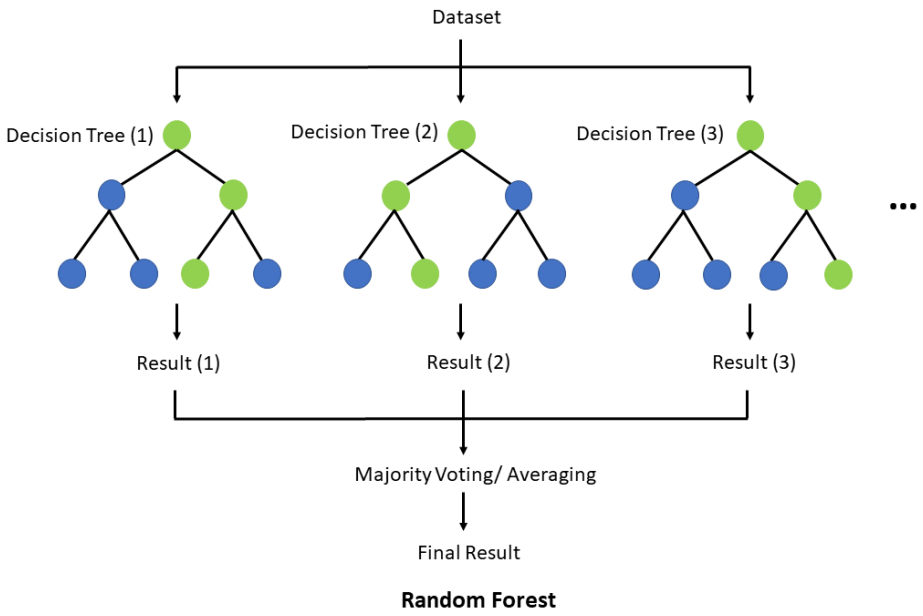
Ο αλγόριθμος Random Forest είναι ένας επιβλεπόμενος αλγόριθμος μηχανικής μάθησης. Είναι ένας από τους πιο συνηθισμένους αλγόριθμους στην μηχανική μάθηση. Προσφέρει ακρίβεια, απλότητα και ευελιξία. Ο Random Forest μπορεί να χρησιμοποιηθεί και για προβλήματα κατηγοριοποίησης αλλά και για προβλήματα παλινδρόμησης, οπότε μπορεί να προσαρμοστεί σε μεγάλο εύρος δεδομένων και καταστάσεων. Το 1995 ο Tim Kam Ho έκανε για πρώτη φορά χρήση του όρου random decision forest. Ενώ ο όρος Random Forest εμφανίστηκε πρώτη φορά από τον Leo Breiman το 2001. Το 2006 οι Breiman και Cutler επέκτειναν τον αλγόριθμο και κατοχύρωσαν το όνομα ‘Random Forests’ το 2006.

Χρησιμοποιείται ο όρος «δάσος» γιατί στην ουσία έχουμε πολλά δέντρα απόφασης (decision trees). Ενώνουμε τα αποτελέσματα των decision trees και παίρνουμε αποτελέσματα με μεγαλύτερη ακρίβεια. Πρέπει αρχικά να δούμε τι είναι όμως τα δέντρα απόφασης. Ένα δέντρο απόφασης διαθέτει μία σειρά από ερωτήσεις τύπου ‘Αλήθεια’ ή ‘Ψέμα’ οι οποίες οδηγούν σε συγκεκριμένη απάντηση. Το δέντρο ξεκινά από την ρίζα. Κάθε ερώτηση αντιπροσωπεύει έναν κόμβο απόφασης. Τα «φύλλα» κόμβοι είναι τα στοιχεία πάνω στα οποία γίνεται η κατηγοριοποίηση. Ένα παράδειγμα με δέντρο απόφασης φαίνεται στην Εικόνα 3. Το δέντρο απόφασης δημιουργείται για το γεγονός αν πρέπει να δεχτούμε μία νέα δουλειά. Οι ελλείψεις αντιστοιχούν στους κόμβους φύλλα ενώ τα μοβ στρογγυλεμένα ορθογώνια στους κόμβους απόφασης. Η ρίζα είναι ο πρώτος κόμβος απόφασης (Μισθός > 15000).



Εικόνα 3: Παράδειγμα δέντρου απόφασης

Παρόλο που ο αλγόριθμος Random Forest αποτελείται από δέντρα απόφασης, υπάρχουν μερικές διαφορές αυτών των δύο. Τα δέντρα απόφασης δημιουργούν κανόνες για να πάρουν αποφάσεις. Ο Random Forest θα διαλέξει τυχαία χαρακτηριστικά και θα κάνει παρατηρήσεις, θα χτίσει το δάσος με τα δέντρα απόφασης και θα βγάλει τον μέσο όρο των αποτελεσμάτων. Αυτό γιατί ο αλγόριθμος Random Forest είναι επέκταση της μεθόδου bagging, οπότε δημιουργεί πολλά ασυσχέτιστα δέντρα. Αυτό οδηγεί σε προβλέψεις με μεγαλύτερη ακρίβεια από ότι θα έκανε ένα δέντρο από μόνο του. Από την μέθοδο bagging τα 2/3 χρησιμοποιούνται για εκπαίδευση ενώ το υπόλοιπο 1/3 συνήθως το χρησιμοποιούμε για δοκιμή. Πριν αρχίσει η εκπαίδευση του Random Forest, έχουν σημασία τρία πράγματα. Αυτά είναι ο αριθμός των κόμβων, ο αριθμός των δέντρων και ο αριθμός των χαρακτηριστικών του δείγματος. Στην Εικόνα 4 βλέπουμε ένα παράδειγμα του Random Forest.



Εικόνα 4: Αναπαράσταση Random Forest

Η λήψη της εικόνας έγινε από https://commons.wikimedia.org/wiki/File:Random_forest_explain.png με άδεια δημοσιοποίησης Attribution-Share Alike 4.0 International

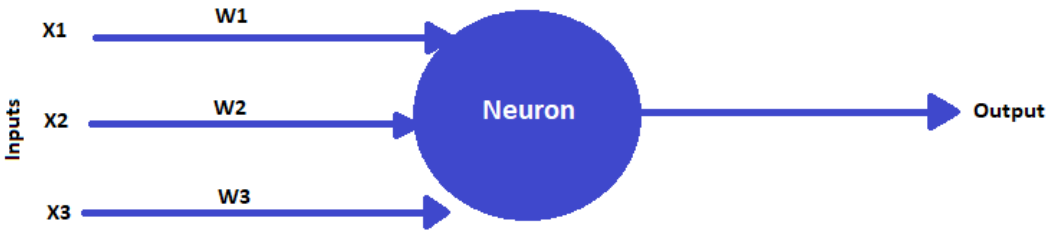
Ο αλγόριθμος Random Forest έχει αρκετές εφαρμογές σε διάφορους κλάδους. Ένας από αυτούς τους κλάδους είναι και αυτός των οικονομικών. Οι Baba και Sevil προσπάθησαν να προβλέψουν την απόδοση μετοχών που τέθηκαν σε δημόσια κυκλοφορία για πρώτη φορά, ενώ οι Tan, Yan και Zhu προσπάθησαν να βρουν τη σωστή στρατηγική στην επιλογή μετοχών στο κινέζικο χρηματιστήριο. Άλλος ένας κλάδος εφαρμογής του αλγορίθμου είναι και αυτός της υγείας. Οι Khalilia, Chakraborty και Popescu προσπάθησαν να προβλέψουν τους κινδύνους ασθένειας ενώ οι Moorthy και Mohamad τον εφάρμοσαν στην επιλογή γονιδίων.

Γενικά ο αλγόριθμος Random Forest έχει αρκετά θετικά στοιχεία. Μπορεί να χρησιμοποιηθεί και για κατηγοριοποίηση αλλά και για παλινδρόμηση. Μπορεί να δουλέψει το ίδιο καλά και με κατηγορικά δεδομένα αλλά και με αριθμητικά χωρίς να χρειάζεται συνήθως κάποια μετατροπή ή κανονικοποίηση των δεδομένων. Έχει χαμηλό ρίσκο να παρουσιάσει το φαινόμενο overfitting, κάτι που μπορεί να παρουσιαστεί στα δέντρα απόφασης, αφού το αποτρέπει χτίζοντας δέντρα διαφορετικού μεγέθους από τα υποσύνολα και συνδυάζει τα αποτελέσματά τους. Παρουσιάζει μεγάλη ακρίβεια στις προβλέψεις αφού χρησιμοποιεί έναν αριθμό δέντρων με σημαντικές διαφορές μεταξύ των

υποομάδων. Ωστόσο ο αλγόριθμος Random Forest δεν είναι τόσο εύκολα ερμηνεύσιμος. Χαρακτηριστικό είναι ότι τον αποκαλούν και «μαύρο κουτί» καθώς δεν μπορούν να εξηγήσουν πως και γιατί κατέληξε σε κάποιο συμπέρασμα. Επίσης επειδή μπορεί να χειριστεί μεγάλο όγκο δεδομένων, απαιτεί και αρκετό χώρο για αποθήκευση αυτών, Επιπλέον ο Random Forest επειδή παρέχει μεγάλη ακρίβεια στις προβλέψεις, μπορεί να γίνει και αρκετά χρονοβόρος άμα ο αριθμός των δεδομένων είναι μεγάλος αφού πρέπει να γίνει επεξεργασία των δεδομένων για κάθε δέντρο απόφασης που υπάρχει. Αυτό μπορεί να τον κάνει αναποτελεσματικό για δεδομένα πραγματικού χρόνου.

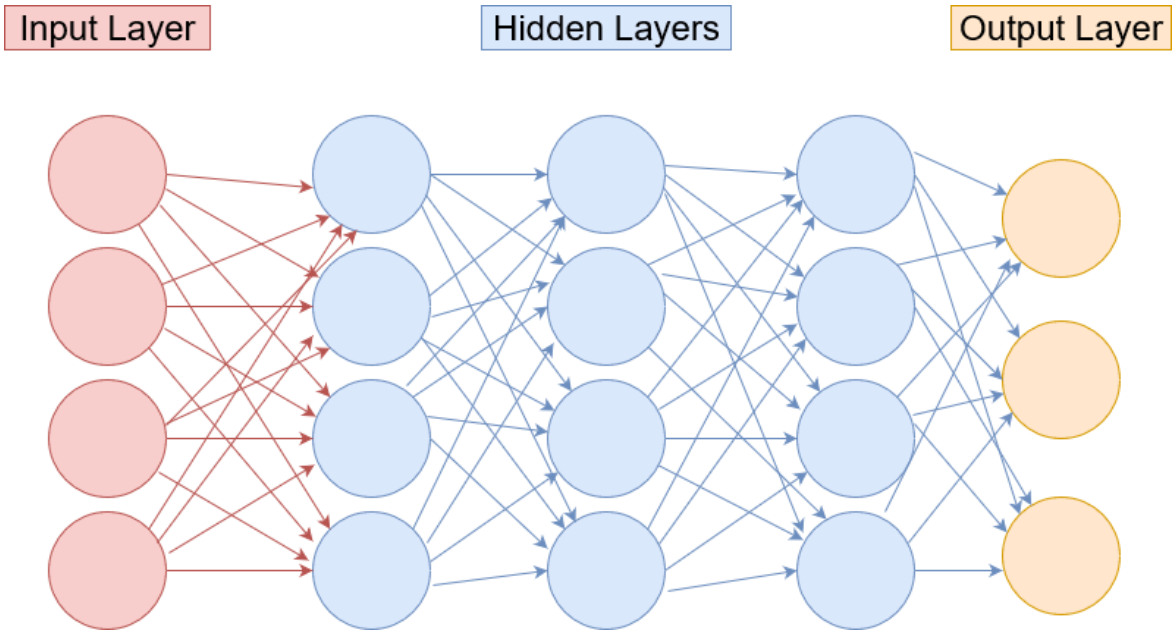
2.7.3 Νευρωνικά Δίκτυα

Όταν μιλάμε για νευρωνικά δίκτυα (NN) αναφερόμαστε σε ένα κύκλωμα διασυνδεδεμένων νευρώνων. Στην περίπτωσή μας αναφερόμαστε στα τεχνητά νευρωνικά δίκτυα (ANN). Πρόκειται για αλγορίθμους που προσπαθούν να μιμηθούν την λειτουργία του ανθρώπινου εγκεφάλου με σκοπό να αναγνωρίζουν μοτίβα και να λύνουν προβλήματα. Χρησιμοποιούνται στον τομέα της τεχνητής νοημοσύνης, της μηχανικής μάθησης και της βαθιάς μάθησης. Πιο συγκεκριμένα τα νευρωνικά δίκτυα είναι ένα υποσύνολο της μηχανικής μάθησης και αποτελούν τον κορμό της βαθιάς μάθησης. Η αρχή των νευρωνικών δικτύων έγινε το 1943 από τους McCulloch και Pitts στην εργασία τους με τίτλο «A logical calculus of the ideas immanent in nervous activity». Προσπάθησαν να καταλάβουν πως ο ανθρώπινος εγκέφαλος μπορούσε να παράγει σύνθετα μοτίβα μέσα από νευρώνες. Μία από τις ιδέες που προέκυψαν από το έργο τους είναι η σύγκριση ενός νευρώνα με λογική του Bool, δηλαδή δυαδικά αποτελέσματα τύπου 0 /1 και True/False. Το 1958 ο Rosenblatt, μέσα από εργασία του μας παρουσίασε τον νευρώνα Perceptron, όπου πήγε το έργο των McCulloch και Pitts ένα βήμα παραπέρα και εισήγαγε τα βάρη στην εξίσωση. Παράδειγμα του Perceptron φαίνεται στην Εικόνα 5.



Εικόνα 5: Ο νευρώνας Perceptron

Τα νευρωνικά δίκτυα αποτελούνται από κόμβους. Κάθε κόμβος έχει το ρόλο του νευρώνα. Συγκεκριμένα τα νευρωνικά δίκτυα αποτελούνται από στρώματα κόμβων που είναι συνδεδεμένα. Έχουν ένα στρώμα εισόδου, ένα ή περισσότερα κρυμμένα στρώματα και ένα στρώμα εξόδου. Ένα απλό νευρωνικό δίκτυο αποτελείται από ένα στρώμα εισόδου, ένα στρώμα εξόδου και ένα κρυμμένο στρώμα μεταξύ τους. Αν το δίκτυο περιέχει περισσότερα από τρία στρώματα, συμπεριλαμβανομένων των στρωμάτων εισόδου και εξόδου τότε έχει μεγάλο «βάθος» και αποτελεί αλγόριθμο της βαθιάς μάθησης. Αν έχει πολλά κρυμμένα στρώματα ένα δίκτυο, έχει την δυνατότητα να αναγνωρίσει πιο σύνθετες πληροφορίες αφού κάθε νευρώνας εκπαιδεύεται στην έξοδο του προηγούμενου. Έτσι, ένας νευρώνας με απλούς υπολογισμούς μπορεί να μας οδηγήσει σε ένα απλό αποτέλεσμα. Πολλοί νευρώνες μαζί μπορούν να αναλύσουν πιο σύνθετα προβλήματα και να μας παρέχουν ακριβείς απαντήσεις. Βλέπουμε την εικόνα ενός νευρωνικού δικτύου στην Εικόνα 6.



Εικόνα 6: Νευρωνικό Δίκτυο με 3 κρυμμένα στρώματα

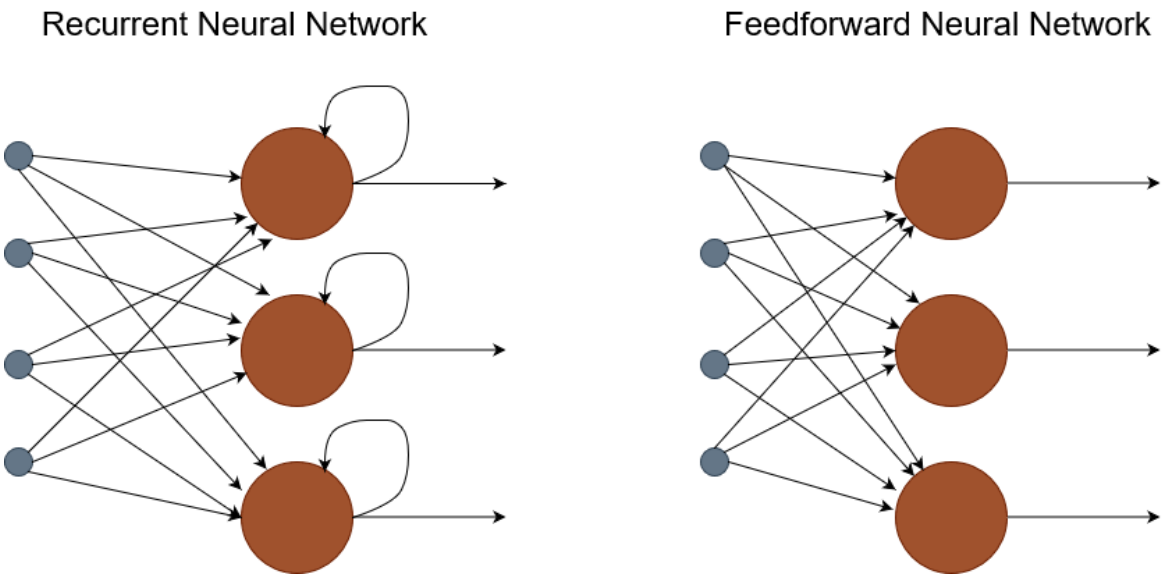
Σε κάθε συνδεδεμένο κόμβο του δικτύου ανατίθεται ένας αριθμός που λέγεται βάρος. Το βάρος αντιπροσωπεύει την «αξία» της πληροφορίας που έχει ανατεθεί στον κάθε κόμβο, δηλαδή πόσο βοηθάει στην κατηγοριοποίηση της πληροφορίας. Όταν ένα νευρώνας λαμβάνει πληροφορίες από άλλους νευρώνες υπολογίζει το συνολικό βάρος ή την «αξία» . Αν ο αριθμός αυτός είναι μεγαλύτερος από ένα συγκεκριμένο όριο, η πληροφορία μεταβαίνει στο επόμενο στρώμα ενώ αν δεν το ξεπερνάει τότε δεν συνεχίζει. Όταν στήνουμε ένα καινούριο νευρωνικό δίκτυο, τα βάρη και τα όρια είναι τυχαίοι αριθμοί. Καθώς όμως εισάγουμε τα δεδομένα εκπαίδευσης στο στρώμα εισόδου, τα βάρη και τα όρια βελτιώνονται και αποδίδουν διαρκώς καλύτερα αποτελέσματα.

Τα περισσότερα νευρωνικά δίκτυα, μεταφέρουν την πληροφορία προς τα μπροστά (feedforward), δηλαδή με κατεύθυνση από το στρώμα εισόδου προς το στρώμα εξόδου. Μπορούμε ωστόσο να εκπαιδεύσουμε το δίκτυό μας με την μέθοδο της οπισθοδιάδοσης (backpropagation). Αυτό σημαίνει ότι έχουμε μεταφορά από το στρώμα εξόδου στο στρώμα εισόδου. Αυτή η μέθοδος μας βοηθάει να υπολογίσουμε και να καταλογίσουμε το σφάλμα που σχετίζεται με κάθε νευρώνα. Στην συνέχεια μπορούμε να προσαρμόσουμε τις παραμέτρους του μοντέλου μας αναλόγως.

Ανάλογα με τον σκοπό που χρησιμοποιούνται τα νευρωνικά δίκτυα, μπορούμε να τα χωρίσουμε σε κατηγορίες:

- Ο νευρώνας Perceptron αποτελεί το πιο παλιό και το πιο απλό νευρωνικό δίκτυο που μπορούμε να συναντήσουμε. Αποτελείται από έναν μόνο νευρώνα.
- Στην συνέχεια έχουμε τα Feedforward neural networks (εμπροσθοτροφοδοτούμενα) ή αλλιώς MLPs (multi-layer perceptrons). Αυτά αποτελούνται από ένα στρώμα εισόδου, ένα στρώμα εξόδου και ενδιάμεσά τους τα κρυφά στρώματα. Εισάγουμε δεδομένα για την εκπαίδευση τους και αποτελούν βασικό θεμέλιο για πεδία όπως το computer vision και το natural language processing.
- Άλλη μία κατηγορία είναι τα Convolutional neural networks (CNN). Είναι παραπλήσια με τα Feedforward neural networks. Χρησιμοποιεί κυρίως στοιχεία της γραμμικής άλγεβρας και συγκεκριμένα τον πολλαπλασιασμό πινάκων για την αναγνώριση μοτίβων και σχεδίων σε εικόνες. Γι' αυτό και χρησιμοποιείται για αναγνώριση εικόνων, αναγνώριση μοτίβων και για computer vision.
- Recurrent neural networks (RNN). Κύριο χαρακτηριστικό των RNN είναι η ανατροφοδότηση. Έχουν την δυνατότητα να αποθηκεύουν τις εξόδους των στρωμάτων και να τις ξαναχρησιμοποιούν για να κάνουν προβλέψεις. Είναι ιδανικά για δεδομένα χρονοσειρών. Χρησιμοποιούνται σε πολλούς τομείς όπως σε μεταφράσεις γλωσσών, αναγνώριση ομιλίας, προβλέψεις μετοχών στο χρηματιστήριο. Χαρακτηριστικά παραδείγματα είναι η Siri της Apple, το Google Translate, η φωνητική αναζήτηση.

Στην Εικόνα 7 βλέπουμε αριστερά το RNN και δεξιά το Feedforward NN.



Εικόνα 7: Recurrent Neural Network vs. Feedforward Neural Network

Τα νευρωνικά δίκτυα παρουσιάζουν ευελιξία. Αυτό τα κάνει χρήσιμα και για κατηγοριοποίηση και για παλινδρόμηση. Επίσης αρκετοί είναι οι κλάδοι στους οποίους έχουν βρει απήχηση και εφαρμογή. Χαρακτηριστική είναι η εφαρμογή τους στον τομέα των οικονομικών. Οι Pang, Zhou, Wang, Lin και Chang με εργασία τους χρησιμοποιούν νευρωνικά δίκτυα για προβλέψεις στο χρηματιστήριο. Στην επεξεργασία εικόνων και σε προσπάθεια αναγνώρισης προσώπων έχουμε αντίστοιχα εργασίες των Cho, Tai, Kweon και των Joseph, Sowmiya, Thomas, Sofia. Έχουμε ακόμα και εφαρμογή στον τομέα της ιατρικής. Οι Janghel, Shukla, Tiwari, Kala χρησιμοποιούν μοντέλα νευρωνικών δικτύων για διάγνωση καρκίνου του μαστού. Ακόμα ένας κλάδος είναι αυτός της ρομποτικής. Οι He, Yan, Sun, Ou, Sun εφάρμοσαν αλγόριθμους νευρωνικών δικτύων σε έναν ρομποτικό βραχίονα.

Τα νευρωνικά δίκτυα έχουν θετικά και αρνητικά στοιχεία. Στα θετικά είναι το γεγονός ότι σε σχέση με τους υπόλοιπους αλγόριθμους μηχανικής μάθησης όπου μετά από ένα σημείο όσα δεδομένα και να χρησιμοποιήσουμε δεν βελτιώνεται η απόδοσή τους, τα νευρωνικά δίκτυα μπορούν να δέχονται συνεχώς και να βελτιώνουν το αποτέλεσμα. Είναι σε μία φάση που είναι αρκετά δημοφιλείς σαν θέμα και αυτό έχει σαν αποτέλεσμα οι αλγόριθμοι που αναπτύσσονται να τρέχουν πιο γρήγορα και αυτό κατ' επέκταση να τους κάνει να δέχονται όλο και περισσότερα δεδομένα. Επίσης έχουν την δυνατότητα να

παράγουν αποτελέσματα και σε περιπτώσεις που τα δεδομένα είναι ελλιπή , ωστόσο βέβαια η απώλεια στην απόδοση να εξαρτάται από την σημασία των δεδομένων που λείπουν. Είναι ιδανικά για μεγάλο αριθμό δεδομένων τα οποία είναι και μη γραμμικά. Τέλος τα νευρωνικά δίκτυα έχουν την ιδιότητα του multitasking. Ενώ το κύριο πλεονέκτημα του είναι το γεγονός ότι μπορεί να αποδώσει καλύτερα από τους υπόλοιπους αλγορίθμους machine learning, τον κάνει να έχει και μερικά μειονεκτήματα. Θεωρείται και αυτός ο αλγόριθμος «μαύρο κουτί». Δεν μπορεί να εξηγηθεί και να κατανοήσουμε πολλές φορές γιατί δίνει τα αποτελέσματα που δίνει. Μπορεί επίσης να θεωρηθεί ότι είναι «υπολογιστικά ακριβός», αφού μπορεί να δεχτεί πάρα πολύ μεγάλο μέγεθος δεδομένων και ακόμα να περιέχει πολλά στρώματα και έτσι να γίνεται ακόμα πιο πολύπλοκο το δίκτυο. Δεν υπάρχει συγκεκριμένος κανόνας κατασκευής ενός νευρωνικού δικτύου. Αυτό επιτυγχάνεται μέσα από δοκιμές, λάθη και εμπειρία. Κάτι ακόμα που βασίζεται στην εμπειρία είναι η πιθανή επίλυση κάποιου προβλήματος, αφού τα δεδομένα που δέχονται τα νευρωνικά δίκτυα είναι αποκλειστικά αριθμοί. Αυτό σημαίνει ότι πρέπει να «μεταφραστούν» τα προβλήματα που θέλουμε να επιλύσουμε για να τα εισάγουμε και στην συνέχεια να ερμηνεύσουμε το αποτέλεσμα.

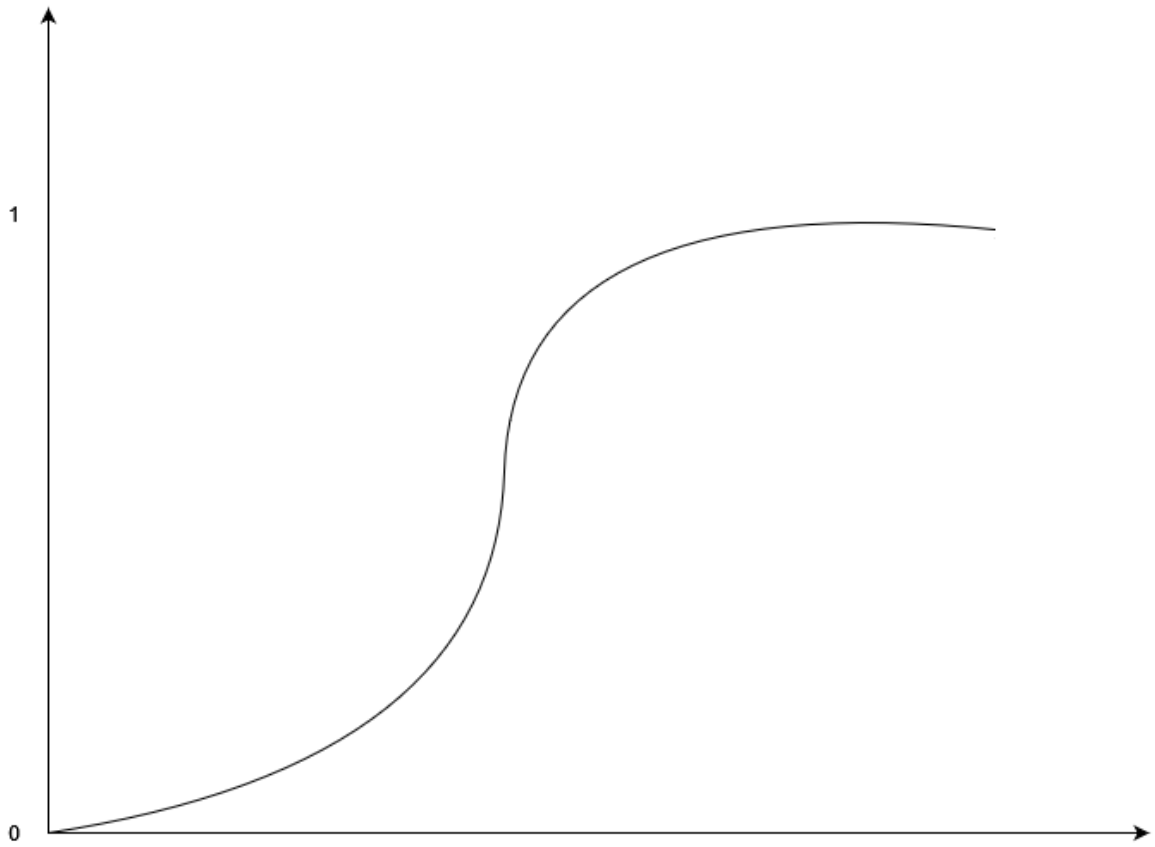
2.7.4 Λογιστική Παλινδρόμηση

Η λογιστική παλινδρόμηση αποτελεί ένα μοντέλο στατιστικής, το οποίο μας βοηθά να υπολογίσουμε την πιθανότητα να συμβεί ένα γεγονός ή να γίνει μία επιλογή για κάτι. Η λογιστική παλινδρόμηση αφορά τις σχέσεις μεταξύ χαρακτηριστικών που έχουμε και έπειτα υπολογίζεται η πιθανότητα για ένα συγκεκριμένο αποτέλεσμα. Καταλαβαίνουμε λοιπόν ότι η λογιστική παλινδρόμηση χρησιμοποιείται για προγνωστική ανάλυση και μοντελοποίηση. Αυτό το γεγονός κάνει χρήσιμη την λογιστική παλινδρόμηση και στην μηχανική μάθηση.

Η λογιστική παλινδρόμηση χρησιμοποιείται για την περιγραφή των δεδομένων και την σχέση μεταξύ μίας εξαρτημένης μεταβλητής και μίας ή περισσοτέρων ανεξάρτητων μεταβλητών. Στον τομέα της μηχανικής μάθησης, η λογιστική παλινδρόμηση μας βοηθά στο να έχουμε ακριβείς προβλέψεις. Είναι δηλαδή ένας είδος δυαδικής κατηγοριοποίησης όπου το αποτέλεσμα θα είναι 0 ή 1. Μπορούμε να συμπεράνουμε ότι ο όρος «λογιστική» οφείλεται στην λογιστική συνάρτηση που είναι γνωστή και ως σιγμοειδής συνάρτηση.

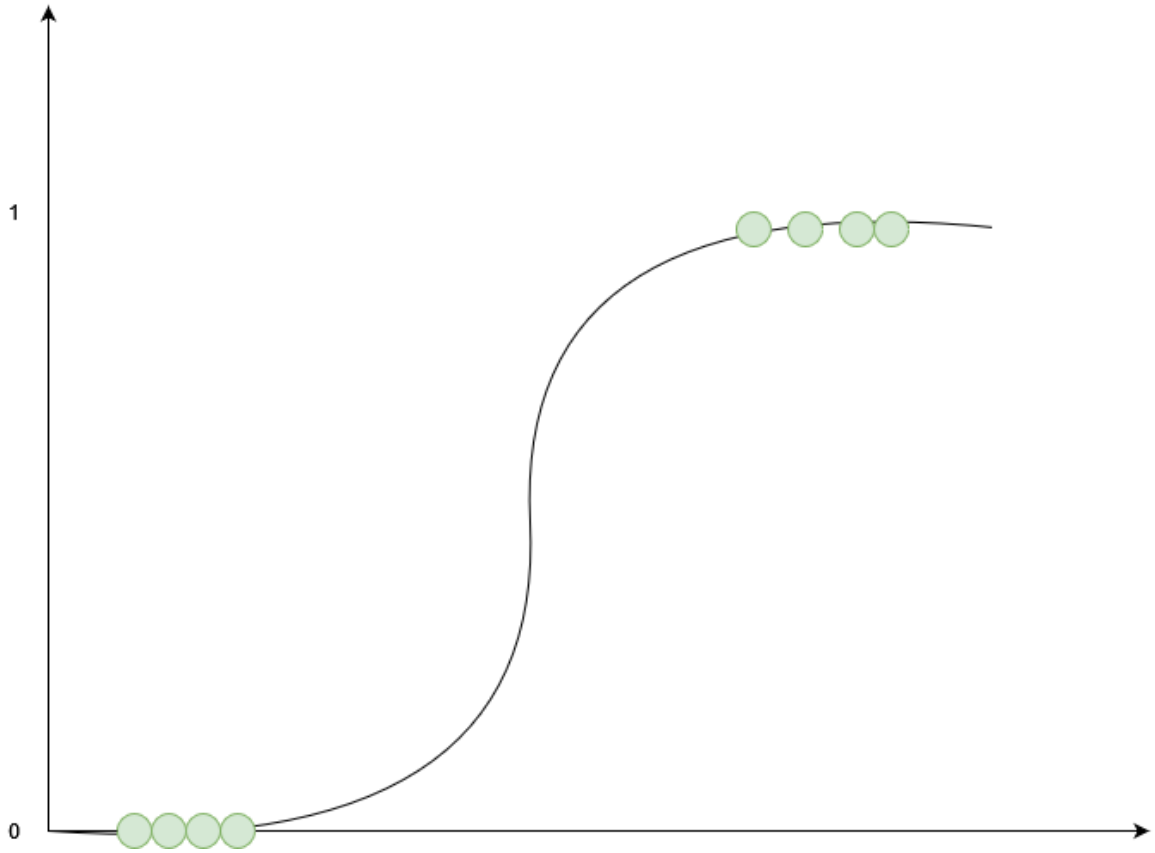
Μπορούμε να δούμε την εξίσωση της σιγμοειδούς συνάρτησης (5) και την γραφική της αναπαράσταση στην Εικόνα 8.

$$(5)\ Logistic\ function = \ S(x) = \frac{1}{1 + e^{-x}}$$



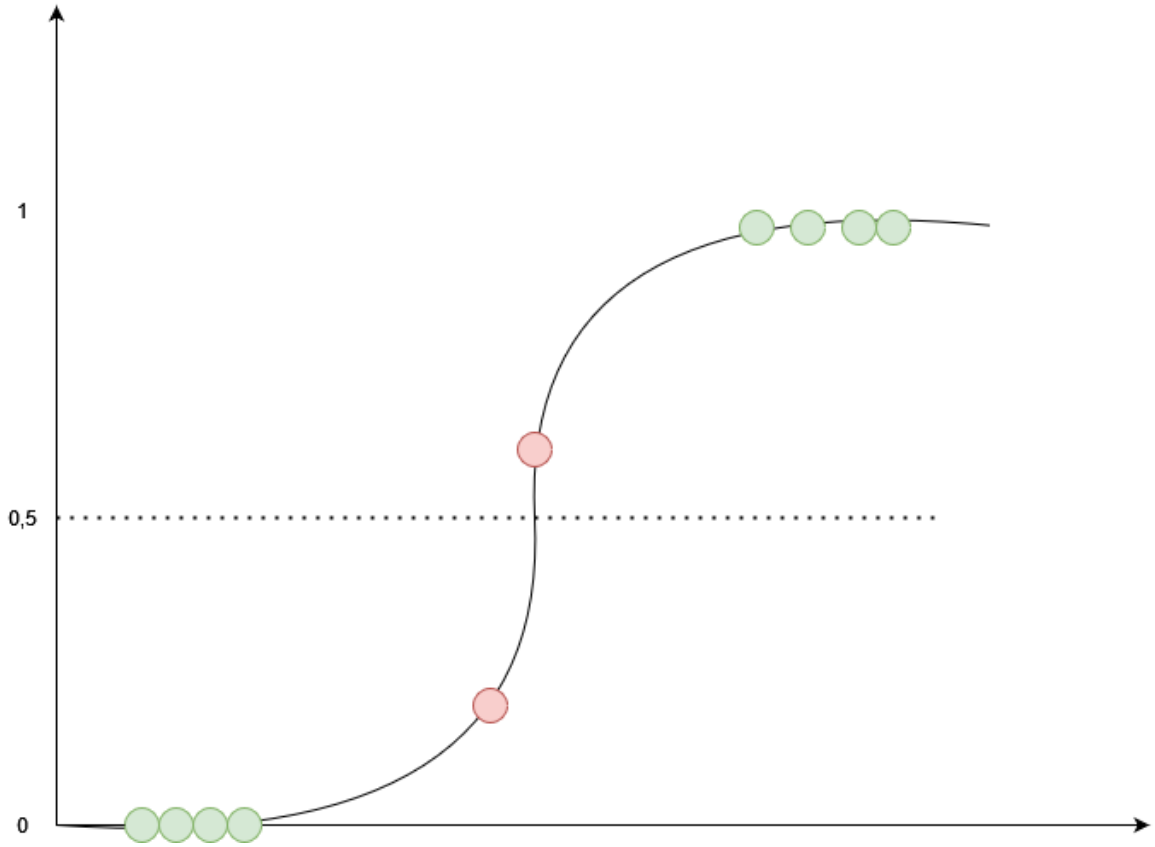
Εικόνα 8: Η σιγμοειδής συνάρτηση

Στην λογιστική παλινδρόμηση τα δεδομένα που εισάγουμε ανήκουν σε κατηγορίες. Αυτό σημαίνει ότι πολλές από τις τιμές εισόδου που έχουμε, αντιστοιχίζονται στις ίδιες τιμές εξόδου. Έτσι μέσω της λογιστικής παλινδρόμησης βρίσκουμε την κατηγορία στην οποία ανήκει η καινούρια τιμή που εισάγεται. Στην Εικόνα 9 βλέπουμε παράδειγμα όπου τα δεδομένα που εισήχθησαν αντιστοιχίζονται σε δύο κατηγορίες, 0 και 1.

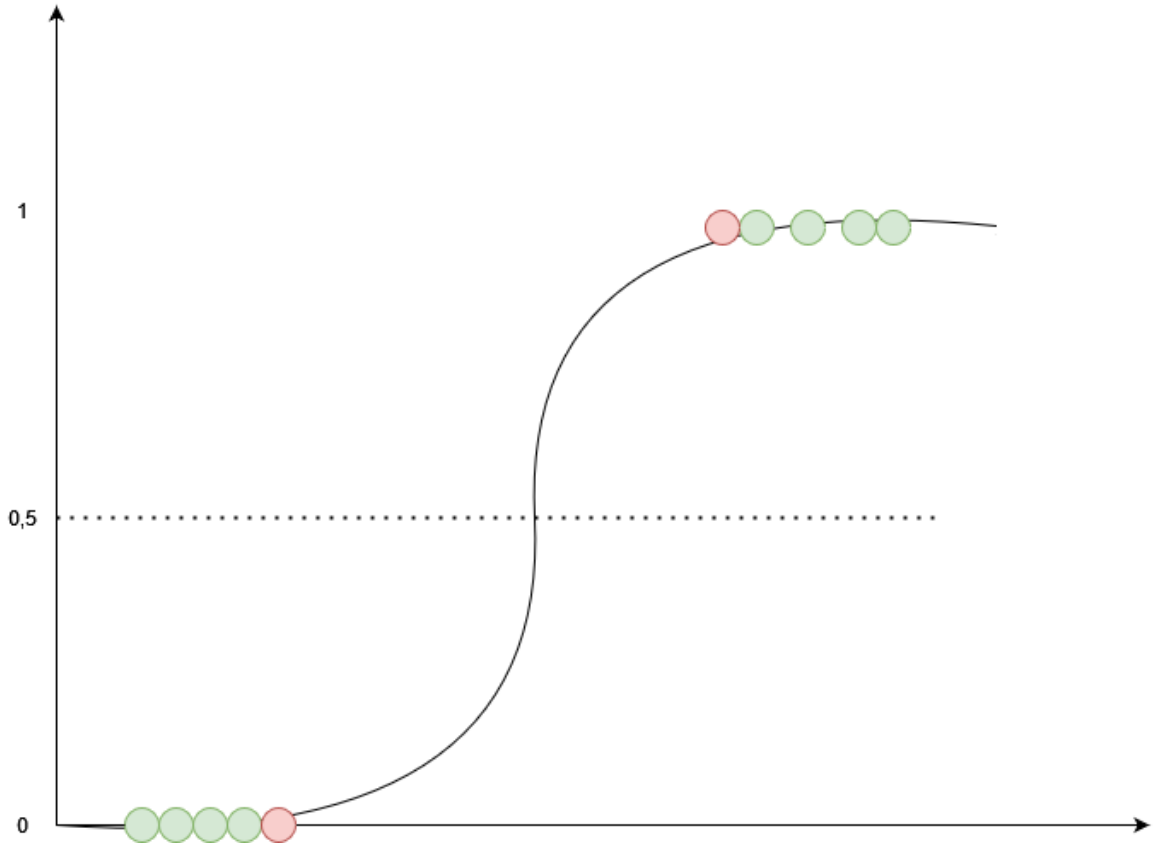


Εικόνα 9: Λογιστική παλινδρόμηση με δεδομένα

Για την κατηγοριοποίηση των τιμών στις δύο κατηγορίες, χρησιμοποιούμε μία τιμή που αποτελεί το κατώφλι. Τιμές που είναι πάνω από αυτήν την τιμή θα συμπεριληφθούν στην κατηγορία 1 ενώ οι τιμές που είναι από κάτω θα συμπεριληφθούν στην κατηγορία 0. Αυτή η διαδικασία φαίνεται στις Εικόνες 10 και 11, όπου θεωρούμε σαν κατώφλι την τιμή 0,5 και έχουμε μία τιμή πάνω από αυτό και μία κάτω από αυτό.



Εικόνα 10: Καινούριες τιμές στο μοντέλο Λογιστικής Παλινδρόμησης



Εικόνα 11: Ταξινόμηση των καινούριων τιμών της Λογιστικής Παλινδρόμησης στις κατηγορίες

Η λογιστική παλινδρόμηση μπορεί να χωριστεί σε τρεις κατηγορίες, σε σχέση με το είδος της εξαρτημένης κατηγορικής μεταβλητής. Αυτές είναι :

- Δυαδική λογιστική παλινδρόμηση (binary logistic regression), όπου ουσιαστικά έχουμε 2 ενδεχόμενα τύπου true/false, επιτυχία/αποτυχία.
- Πολυωνυμική λογιστική παλινδρόμηση (multinomial logistic regression), όπου μπορούμε να έχουμε 3 ή περισσότερες κατηγορίες χωρίς αυτές να είναι σε κάποια σειρά.
- Τακτική λογιστική παλινδρόμηση (ordinal logistic regression), όπου και εδώ έχουμε 3 ή περισσότερες κατηγορίες οι οποίες έχουν μία διαβάθμιση ή απλά μπορούμε να πούμε ότι ισχύει η έννοια της ανισότητας μεταξύ τους.

Η λογιστική παλινδρόμηση έχει βρει εφαρμογή σε αρκετούς κλάδους. Ένας από αυτούς είναι και ο κλάδος της ιατρικής όπως παρουσιάζουν οι Zabor, Reddy, Tendulkar, Patil.

Επίσης στον κλάδο των οικονομικών από τους Bhandari και Johnson-Syder για την πρόβλεψη επιτυχίας ή αποτυχίας ενός οργανισμού. Ακόμα έχει εφαρμογές και στις πολιτικές επιστήμες και πως επηρεάζονται ή πως «στοχεύουν» στους ψηφοφόρους, όπου παρουσιάστηκε από τους Rusch, Lee, Hornik, Jank, Zeileis.

Η λογιστική παλινδρόμηση είναι πολύ αποτελεσματική σαν μέθοδος και γι' αυτό χρησιμοποιείται αρκετά. Εφαρμόζεται εύκολα και είναι αρκετά απλή στην διαδικασία της εκπαίδευσης και αυτό την κάνει να μην χρειάζεται μεγάλη υπολογιστική δύναμη σε σχέση με άλλους αλγορίθμους. Έτσι η λογιστική παλινδρόμηση μπορεί να αποτελέσει ένα βασικό εργαλείο για να μετρήσουμε την απόδοση άλλων πιο σύνθετων αλγορίθμων. Ωστόσο η λογιστική παλινδρόμηση δεν μπορεί να χρησιμοποιηθεί για μη γραμμικά προβλήματα. Επιπλέον δεν είναι από τους πιο ισχυρούς αλγόριθμους, καθώς υπάρχουν άλλοι με καλύτερα αποτελέσματα για πιο σύνθετες προβλέψεις. Μεγάλο ρόλο παίζει και τα δεδομένα που εισάγουμε καθώς η συσχέτιση μεταξύ τους θα πρέπει να είναι ελάχιστη ή καθόλου προκειμένου να έχουμε πιο σωστά αποτελέσματα. Ακόμα η λογιστική παλινδρόμηση μπορεί να παρουσιάσει συχνά το φαινόμενο του overfitting.

3. Δεδομένα και Μεθοδολογία

Σε αυτό το κομμάτι της διπλωματικής εργασίας, θα παρουσιαστεί ότι έχει σχέση με τα δεδομένα. Θα γίνει αναφορά για το πώς έγινε η συλλογή των δεδομένων αλλά και από ποιες πηγές πήραμε τα δεδομένα. Στην συνέχεια θα παρατεθεί ο τρόπος με τον οποίο έγινε η επεξεργασία των δεδομένων ώστε να μπορούν να προσαρμοστούν στους αλγορίθμους και να έχουμε ακρίβεια στα αποτελέσματα. Έπειτα θα δούμε την μεθοδολογία αυτών των αλγορίθμων και πως μας οδηγούν στην πρόβλεψη.

3.1 Περιγραφή και Συλλογή Δεδομένων

Η συλλογή των δεδομένων είναι από τα πιο σημαντικά στάδια μιας έρευνας εφόσον θέλει να οδηγηθεί και να λειτουργήσει σε αυτό που πραγματεύεται. Στην δικιάς μας περίπτωση

επιλέξαμε να μελετήσουμε και να αξιολογήσουμε την μεταβλητότητα του κρυπτονομίσματος bitcoin. Έτσι θα χρειαστούμε δεδομένα που αναφέρονται σε αυτό. Βάση της βιβλιογραφίας που παρουσιάστηκε υπήρχαν αρκετές ιστοσελίδες από τις οποίες μπορούσε κάποιος να πάρει διάφορα δεδομένα για το κρυπτονόμισμα bitcoin. Στην παρούσα διπλωματική εργασία η άντληση των δεδομένων έγινε από 2 πηγές. Η πρώτη πηγή είναι από την ιστοσελίδα finance.yahoo.com. Από εκεί πήραμε καθημερινά δεδομένα που αφορούν το bitcoin. Αυτά τα δεδομένα αναφέρονται στην τιμή ανοίγματος, τιμή κλεισίματος, υψηλότερη και χαμηλότερη τιμή καθημερινά αλλά και τον όγκο συναλλαγών. Τα δεδομένα τα πήραμε σε μορφή αρχείου csv με αρχική ημερομηνία 17/09/2014 και τελική 05/05/2022. Τα υπόλοιπα δεδομένα που χρησιμοποιήθηκαν στην διπλωματική εργασία αντλήθηκαν από την ιστοσελίδα Blockchain.com. Αφορούν κυρίως στοιχεία και δεδομένα που επηρεάζουν την πορεία του κρυπτονομίσματος. Τα δεδομένα από την ιστοσελίδα Blockchain.com είναι και αυτά σε μορφή αρχείου τύπου csv και αφορούν για χρονικό διάστημα 06/05/2019 μέχρι και 04/05/2022. Τέλος, όπου γίνεται αναφορά για χρηματική αξία του κρυτπονομίσματος bitcoin στα δεδομένα, αυτήν αναφέρεται στο νόμισμα του δολαρίου.

Αφού έγινε η συλλογή και η άντληση των δεδομένων, μπορεί πλέον να γίνει μία περιγραφή αυτών. Τα δεδομένα που αντλήθηκαν από την ιστοσελίδα finance.yahoo.com αφορούν την μεταβλητή Open, όπου αναφέρεται στην τιμή με την οποία ανοίγει το κρυπτονόμισμα bitcoin κάθε μέρα. Στην συνέχεια έχουμε το High που αφορά ην υψηλότερη τιμή που θα φτάσει το bitcoin εκείνη την ημέρα. Ακολουθεί αντίστοιχα το Low για την χαμηλότερη τιμή της ημέρας. Ακολουθούν στην συνέχεια οι μεταβλητές Close και Adj Close που αναφέρονται στην τιμή κλεισίματος και στην προσαρμοσμένη τιμή κλεισίματος αντίστοιχα. Έπειτα έχουμε το Volume που αναφέρεται στο όγκο των συναλλαγών.

Τώρα θα ξεκινήσουμε να βλέπουμε τις μεταβλητές που πήραμε από την ιστοσελίδα Blockchain.com. Ξεκινάμε με την μεταβλητή miners-revenue, η οποία περιγράφει τα κέρδη και τις αμοιβές των ατόμων που ασχολούνται με την εξόρυξη (miners). Η μεταβλητή transaction-fees αφορά τα τέλη συναλλαγών που καταβάλλονται στους miners. Στην συνέχεια έχουμε την μεταβλητή n-transactions που αφορά το πλήθος των συναλλαγών την κάθε μέρα, ενώ η μεταβλητή n-transactions-total που αφορά τον συνολικό αριθμό συναλλαγών στο blockchain. Η μεταβλητή cost-per-transaction αφορά

τα έσοδα των miners διαιρούμενα με τον αριθμό των συναλλαγών. Ακολουθεί η μεταβλητή n-unique-addresses όπου συμβολίζει τον αριθμό των διευθύνσεων των χρηστών για την κάθε ημέρα. Τελειώνοντας έχουμε τις μεταβλητές estimated-transaction-volume και estimated-transaction-volume-usd που αφορούν την συνολική εκτιμώμενη αξία σε bitcoin των συναλλαγών που έγιναν και την συνολική εκτιμώμενη αξία σε δολάρια για τις συναλλαγές που έγιναν.

Αυτές είναι οι αρχικές μεταβλητές που χρησιμοποιήσαμε για να πραγματοποιήσουμε την διπλωματική εργασία. Στην συνέχεια θα δούμε την επεξεργασία που έγινε σε αυτές στην γλώσσα προγραμματισμού Python έτσι ώστε να μπορέσουμε να δημιουργήσουμε μία νέα ενιαία βάση δεδομένων όπου εμπεριέχονται όλες αυτές οι μεταβλητές. Βλέπουμε παρακάτω στον Πίνακα 1, τα περιγραφικά στοιχεία των παραπάνω μεταβλητών.

Μεταβλητές	**Περιγραφή**
Open	Ημερήσια τιμή ανοίγματος
High	Υψηλότερη τιμή ημέρας
Low	Χαμηλότερη τιμή ημέρας
Close	Τιμή κλεισίματος
Adj Close	Προσαρμοσμένη τιμή κλεισίματος
Volume	Όγκος συναλλαγών
miners-revenue	Κέρδη των miners
transaction-fees	Τέλη συναλλαγών
n-transactions	Πλήθος συναλλαγών
n-transactions-total	Συνολικό πλήθος συναλλαγών
cost-per-transaction	Έσοδα των miners διαιρούμενα από τον αριθμό συναλλαγών
n-unique-addresses	Μοναδικές διευθύνσεις χρηστών για τις

	συναλλαγές
estimated-transaction-volume	Εκτιμώμενη αξία συναλλαγών σε bitcoin
estimated-transaction-volume-usd	Εκτιμώμενη αξία συναλλαγών σε δολάρια

Πίνακας 1: Περιγραφικά στοιχεία των μεταβλητών

3.2 Τεχνικά χαρακτηριστικά, Λογισμικό, Γλώσσα Προγραμματισμού

Πριν περάσουμε στην επεξεργασία των δεδομένων θεωρήθηκε σκόπιμο να αναφέρουμε κάποια από τα χαρακτηριστικά του τοπικού συστήματος που χρησιμοποιήθηκε για την εκπόνηση της διπλωματικής εργασίας. Αυτά φαίνονται στον Πίνακα 2.

	Χαρακτηριστικά Συστήματος
Λειτουργικό Σύστημα	Windows 10 Pro-64bit
CPU	Intel Core i5 10400F
RAM	4x8 GB DDR4-6400 MHz
GPU	NVIDIA GTX 1060 6GB

Πίνακας 2: Χαρακτηριστικά τοπικού συστήματος

Για την υλοποίηση του κώδικα αρχικά έγινε εγκατάσταση του λογισμικού Anaconda. Παρέχει εύκολη διαχείριση των βιβλιοθηκών τόσο σε αναζήτηση όσο και σε εγκατάσταση. Μέσα από το Anaconda, παρέχεται και το περιβάλλον Jupyter Notebook, όπου έγινε και η συγγραφή του κώδικα στην γλώσσα προγραμματισμού Python. Η έκδοση της γλώσσας προγραμματισμού Python είναι η 3.9.7 ενώ οι βιβλιοθήκες που χρησιμοποιήθηκαν φαίνονται στον Πίνακα 3.

Όνομα βιβλιοθήκης	**Έκδοση**
Numpy	1.20.3

Pandas	1.3.4
Scikit-learn	0.24.2

Πίνακας 3: Βιβλιοθήκες Python που χρησιμοποιήθηκαν στην διπλωματική εργασία

Τέλος ο κώδικας καθώς και τα αρχεία csv που χρησιμοποιηθήκαν υπάρχουν και στην πλατφόρμα του github και συγκεκριμένα στην διεύθυνση: https://github.com/GeorgeMichis/Project1 , που δημιουργήθηκε για την εκπόνηση της παρούσας διπλωματικής εργασίας.

3.3 Επεξεργασία Δεδομένων

Εφόσον έγινε η συλλογή των δεδομένων, επόμενο βήμα που πρέπει να ακολουθήσουμε είναι η επεξεργασία τους. Όπως αναφέραμε, σε αυτήν την διπλωματική εργασία θα πραγματοποιηθεί μέσω της γλώσσας προγραμματισμού Python. Πρώτο βήμα λοιπόν είναι να χρησιμοποιήσουμε τις κατάλληλες βιβλιοθήκες για να μπορέσουμε να επεξεργαστούμε τα δεδομένα. Αυτές είναι η pandas, η numpy, η sklearn. Όλα τα αρχεία που πήραμε από τις ιστοσελίδες finance.yahoo.com και blockchain.com είναι σε μορφή αρχείου τύπου csv. Οπότε αρχικά πρέπει να εισάγουμε την βιβλιοθήκη pandas και μέσω της εντολής read, τα «διαβάζουμε» και πλέον είναι έτοιμα για επεξεργασία μέσω της Python.

Αναφέραμε και παραπάνω τα χρονικά διαστήματα των αρχείων που πήραμε από τις 2 ιστοσελίδες. Βλέπουμε ότι το ένα αρχείο που περιέχει τις μεταβλητές Open, High, Low, Close, Adj Close, Volume αναφέρεται σε καθημερινές μετρήσεις για το διάστημα 17/09/2014 έως και 05/05/2022. Αντίθετα τα αρχεία που περιέχουν τις μεταβλητές miners-revenue, transaction-fees, n-transactions, n-transactions-total, cost-per-transaction, n-unique-addresses, estimated-transaction-volume και estimated-transaction-volume-used αναφέρονται σε καθημερινές μετρήσεις για το διάστημα 06/05/2019 μέχρι και 04/05/2022. Έτσι αποφασίστηκε για την έρευνα το χρονικό διάστημα που θα μελετήσουμε την μεταβλητότητα να είναι από 10/05/2019 μέχρι και 01/05/2022, ένα διάστημα δηλαδή περίπου 3 χρόνων. Οπότε μέσω της εντολής read που είχαμε χρησιμοποιήσει για να διαβάσουμε τα csv αρχεία δημιουργήσαμε κάποια dataframes. Αυτά εμπεριέχουν μία

στήλη είτε με όνομα 'Date' είτε με όνομα 'Timestamp', οπότε μέσω μίας εντολής .loc προσαρμόσαμε της ημερομηνίες που θέλαμε για την έρευνα.

Πριν προχωρήσουμε στην ένωση όλων των μεταβλητών σε ένα κοινό αρχείο για την δημιουργία της βάσης δεδομένων έπρεπε πρώτα να γίνει έλεγχος για το αν υπάρχουν ημερομηνίες που λείπουν για αυτό το συγκεκριμένο χρονικό διάστημα για το κάθε ένα αρχείο ξεχωριστά. Μετά τον ορισμό του συγκεκριμένου χρονικού διαστήματος που ορίστηκε για την έρευνα, είδαμε ότι το πλήθος των στοιχείων στα dataframe που δημιουργήθηκαν ήταν το ίδιο και είναι στις 1087 μετρήσεις, εκτός από τις 3 τελευταίες μεταβλητές δηλαδή τις n-unique-addresses, estimated-transaction-volume και estimated-transaction-volume-usd όπου οι μετρήσεις ήταν 1084. Οπότε πρέπει να γίνει έλεγχος για τις ημερομηνίες που λείπουν. Για να το δούμε αυτό, για κάθε αρχείο που εξετάζαμε, θέταμε σαν δείκτη του dataframe είτε την στήλη 'Date' είτε την στήλη 'Timestamp', ότι είχε δηλαδή το κάθε αρχείο και στην συνέχεια χρησιμοποιούμε την εντολή της βιβλιοθήκης pandas DatetimeIndex.difference(other). Χρησιμοποιώντας την προηγούμενη εντολή για το κάθε αρχείο ξεχωριστά διαπιστώθηκε ότι στο πρώτο αρχείο που εμπεριέχει τις μεταβλητές Open, High, Low, Close, Adj Close, Volume δεν λείπει καμία ημερομηνία. Στις υπόλοιπες μεταβλητές, τις οποίες πήραμε από την ιστοσελίδα blockchain.com (παρόλο που έχουν το ίδιο πλήθος στοιχείων, εκτός από τις 3 τελευταίες μεταβλητές όπως έχουν παρουσιαστεί παραπάνω) λείπουν από όλες τις μεταβλητές οι ημερομηνίες 29/03/20, 28/03/2021 και 27/03/2022. Αυτές είναι οι τελευταίες Κυριακές του Μαρτίου όπου αλλάζει η ώρα. Αυτό είχε σαν αποτέλεσμα να γίνει μία παρατήρηση για το τι συμβαίνει την τελευταία Κυριακή του Οκτωβρίου που αλλάζει η ώρα, όπου εκεί είδαμε αντίστοιχα διπλές ημερομηνίες για τις 27/10/2019, 25/10/2020 και 31/10/2021, όπου παρόλο που είχαμε διπλές ημερομηνίες η τιμή της μεταβλητής ήταν διαφορετική στην κάθε ημερομηνία στα αρχεία και ουσιαστικά ήταν το ίδιο, δηλαδή σαν να μην έλειπε καμία ημερομηνία (έγιναν δοκιμές για τις ημερομηνίες και τον έλεγχο αυτών με την εντολή df.iloc[αριθμός , :]). Τέλος παρατηρήθηκε ότι έλειπαν 3 επιπλέον ημερομηνίες στην μεταβλητή n-unique-addresses και 3 ίδιες ημερομηνίες για τις μεταβλητές estimated-transaction-volume και estimated-transaction-volume-usd, που είναι και ο λόγος που το πλήθος τους ήταν 1084 σε σχέση με το 1087 των υπολοίπων.

Επόμενο βήμα ήταν να ενώσουμε τα dataframe των αρχείων που πήραμε από την ιστοσελίδα blockchain.com. Αυτό έγινε με την μεθοδο reduce() από το functools. Έτσι

ενώσαμε σε ένα μεγάλο dataframe τα αρχεία της ιστοσελίδας blockchain.com. Βλέπουμε ξανά ότι το πλήθος των γραμμών αυτού του dataframe είναι ίδιο με το πλήθος του dataframe που περιέχει τις 6 πρώτες μεταβλητές, δηλαδή 1087. Επομένως τώρα μέσω της df.drop(), αφαιρούμε τις στήλες 'Date' και 'Timestamp' και στην συνέχεια μέσω της pandas.concat() ενώνουμε τα 2 dataframe σε ένα. Στο καινούριο dataframe ελέγχομουμε για NaN τιμές (ξέρουμε από πριν ότι πρέπει να έχουμε 9 στο σύνολο, 3 από την unique-addresses, και από 3 στις estimated-transaction-volume και estimated-transaction-volume-usd) και βρίσκουμε ότι υπάρχουν 9 τιμές NaN στο σύνολό μας. Υπολογίζουμε τον μέσο όρο της κάθε στήλης (μεταβλητής) στην οποία λείπουν οι 3 τιμές και στην συνέχεια αντικαθιστούμε τις τιμές NaN με τον μέσο όρο. Γίνεται έπειτα ξανά έλεγχος και δεν υπάρχουν πλέον κενές τιμές στο dataframe. Τέλος δημιουργούμε ένα αρχείο csv με τον τελικό dataframe και έτσι ολοκληρώνεται το πρώτο στάδιο της επεξεργασίας των δεδομένων. Αξίζει να σημειώσουμε πως έγιναν πολλές δοκιμές μεταξύ του τελικού dataframe και των υπολοίπων για τον έλεγχο και για την επιβεβαίωση αντιστοιχίας των τιμών. Όπως αναφέρθηκε και παραπάνω χρησιμοποιήθηκε η εντολή DataFrame.iloc .

Ξεκινάμε το δεύτερο στάδιο της επεξεργασίας με το διάβασμα του αρχείου csv που περιέχει το τελικό dataframe με τα δεδομένα μας. Πρώτη κίνηση είναι να χρησιμοποιήσουμε την εντολή DataFrame.sort_index() για να αλλάξουμε την ταξινόμηση και να την κάνουμε φθίνουσα. Αυτό το κάνουμε για να μας βοηθήσει στους υπολογισμούς που έχουμε να πραγματοποιήσουμε.

Επόμενο βήμα είναι να υπολογίσουμε την διαφορά στην τιμή κάθε ημέρας. Από την τιμή κλεισίματος αφαιρούμε την τιμή κλεισίματος της προηγούμενης μέρας και βλέπουμε την διαφορά. Αυτήν η διαφορά καταχωρείται σε νέα στήλη με όνομα 'Daily Difference'. Έπειτα, βάση αυτής της στήλης γίνεται έλεγχος των τιμών της για τον αν είναι μεγαλύτερες, μικρότερες ή ίσες με το μηδέν οι ημερήσιες διαφορές και προσθέτουμε τα αποτελέσματα σε μία λίστα. Στην συνέχεια μετατρέπουμε αυτήν τη λίστα σε στήλη του dataframe με όνομα 'Price Status'. Για τον υπολογισμό της στήλη 'Daily Difference' χρησιμοποιήθηκε η εντολή dataframe.shift(-1), οπότε δημιουργήθηκε στην τελευταία γραμμή στις καινούριες στήλες NaN τιμές, οπότε πλέον δεν μας χρειάζεται και γι' αυτό γίνεται η αφαίρεσή της. Οπότε πλέον πρέπει να γίνει εκ νέου χρήση της εντολής dataframe.reset_index(). Έτσι στο τελικό dataframe έχουμε 1086 μετρήσεις.

Η στήλη 'Price Status' είναι αυτήν στην οποία θα βασίσουμε την κατηγοριοποίηση που πρόκειται να κάνουμε. Αυτήν την στιγμή εμπεριέχει τις πιθανές τιμές 'Increased Price', 'Decreased Price' και 'Same Price'. Γίνεται λοιπόν η αλλαγή στις ταξικές ακέραιες τιμές 1, -1, 0 αντίστοιχα.

Επόμενο σημαντικό βήμα πριν την εφαρμογή των αλγορίθμων μηχανικής μάθησης που επιλέχτηκαν είναι η κανονικοποίηση των δεδομένων. Πριν γίνει όμως η κανονικοποίηση, πρέπει πρώτα να αφαιρεθεί η στήλη 'Daily Difference' γιατί ουσιαστικά βοήθησε στο να παραχθεί και να διαμορφωθεί η στήλη 'Price Status'. Αν την αφήσουμε θα επηρεάσει αρνητικά το αποτέλεσμα και δεν θα έχουμε σωστά αποτελέσματα. Αφού πραγματοποιηθεί η αφαίρεση αυτής της στήλης, μπορούμε πλέον να κάνουμε την κανονικοποίηση. Η κανονικοποίηση είναι μία τεχνική που την βλέπουμε να χρησιμοποιείται πολύ συχνά στα δεδομένα όταν πρόκειται να χρησιμοποιηθούν για machine learning. Αυτό γίνεται γιατί οι αριθμητικές τιμές που περιέχει η κάθε στήλη μπορεί να διαφέρουν και μάλιστα κατά πολύ. Αυτό έχει σαν αποτέλεσμα να πρέπει να βρεθεί μία κοινή κλίμακα έτσι ώστε να ομαλοποιηθούν οι τιμές και να βρίσκονται στο ίδιο διάστημα. Η κανονικοποίηση είναι σημαντική και για την σωστή λειτουργία αρκετών αλγορίθμων. Μεγάλες διαφορές στις τιμές θα μπορούσαν να εμφανίσουν προβλήματα στην μοντελοποίηση και κατ' επέκταση στα αποτελέσματα της έρευνας. Στην παρούσα διπλωματική εργασία θα χρησιμοποιήσουμε την μέθοδο MinMaxScaler() από την βιβλιοθήκη scikit-learn. Η MinMaxScaler() χρησιμοποιεί το διάστημα [0,1] με μέγιστη τιμή το 1 και ελάχιστη τιμή το 0. Η μετατροπή των τιμών βασίζεται στην εξίσωση (6):

$$(6)\ x_{scaled} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Η κανονικοποίηση γίνεται σε όλες της μεταβλητές εκτός από την 'Price Status'. Είναι σημαντικό να γίνει κανονικοποίηση, καθώς υπάρχουν πολύ μεγάλες τιμές στα δεδομένα μας και αυτό θα είχε σαν αποτέλεσμα να κυριαρχήσουν και έτσι να μην δημιουργηθεί σωστά το μοντέλο και να είχε και λάθος αποτελέσματα. Αφού γίνει η κανονικοποίηση

δημιουργούμε ένα dataframe με τις κανονικοποιημένες πλέον τιμές και παρουσιάζουμε στον Πίνακα 4 τα περιγραφικά στατιστικά.

Open	**High**	**Low**	**Close**
Count: 1086.000000 Mean: 0.338996 Std: 0.302162 Min: 0.000000 25%: 0.068982 50%: 0.141704 75%: 0.609324 Max: 1.000000	Count: 1086.000000 Mean: 0.339621 Std: 0.305561 Min: 0.000000 25%: 0.064790 50%: 0.138543 75%: 0.609523 Max: 1.000000	Count: 1086.000000 Mean: 0.342867 Std: 0.294202 Min: 0.000000 25%: 0.081784 50%: 0.152508 75%: 0.611145 Max: 1.000000	Count: 1086.000000 Mean: 0.339637 Std: 0.301732 Min: 0.000000 25%: 0.069492 50%: 0.144917 75%: 0.609296 Max: 1.000000
Adj Close	**Volume**	**miners-revenue**	**transaction-fees**
Count: 1086.000000 Mean: 0.339637 Std: 0.301732 Min: 0.000000 25%: 0.069492 50%: 0.144917 75%: 0.609296 Max: 1.000000	Count: 1086.000000 Mean: 0.067686 Std: 0.056674 Min: 0.000000 25%: 0.031623 50%: 0.057180 75%: 0.088112 Max: 1.000000	Count: 1086.000000 Mean: 0.291238 Std: 0.226581 Min: 0.000000 25%: 0.104986 50%: 0.192485 75%: 0.473413 Max: 1.000000	Count: 1086.000000 Mean: 0.177587 Std: 0.179623 Min: 0.000000 25%: 0.037115 50%: 0.101668 75%: 0.275084 Max: 1.000000
n-transactions	**n-transactions-total**	**cost-per-transaction**	**n-unique-addresses**
Count: 1086.000000 Mean: 0.595329 Std: 0.160540	Count: 1086.000000 Mean: 0.527398 Std: 0.289416	Count: 1086.000000 Mean: 0.279510	Count: 1086.000000 Mean: 0.435823 Std: 0.176354

Min: 0.000000 25%: 0.491761 50%: 0.599669 75%: 0.712643 Max: 1.000000	Min: 0.000000 25%: 0.278168 50%: 0.539555 75%: 0.779422 Max: 1.000000	Std: 0.238322 Min: 0.000000 25%: 0.080084 50%: 0.148630 75%: 0.483650 Max: 1.000000	Min: 0.000000 25%: 0.306887 50%: 0.429491 75%: 0.553879 Max: 1.000000
estimated-transaction-volume Count: 1086.000000 Mean: 0.193637 Std: 0.116184 Min: 0.000000 25%: 0.115554 50%: 0.185768 75%: 0.249423 Max: 1.000000		**estimated-transaction-volume-usd** Count: 1086.000000 Mean: 0.193441 Std: 0.168475 Min: 0.000000 25%: 0.066161 50%: 0.124309 75%: 0.290697 Max: 1.000000	

Πίνακας 4: Περιγραφικά στατιστικά μεταβλητών

Τέλος τα περιγραφικά στατιστικά της στήλης 'Price Status', στην οποία θα βασιστούμε για την κατηγοριοποίηση φαίνονται παρακάτω στον Πίνακα 5.

Price Status	
1 (Increased Price):	**564**
-1 (Decreased Price)	**522**

Πίνακας 5: Περιγραφικά στατιστικά μεταβλητής κατηγοριοποίησης

Παρατηρούμε ότι για το διάστημα που επιλέξαμε να κάνουμε την έρευνα, η τιμή αυξήθηκε περισσότερες φορές.

3.4 Μεθοδολογία

Αφού συλλέξαμε τα δεδομένα και ύστερα από κατάλληλη επεξεργασία μπορούμε πλέον να προχωρήσουμε περαιτέρω. Επόμενο βήμα είναι ο διαχωρισμός των δεδομένων. Πρέπει να γίνει διαχωρισμός σε δεδομένα εκπαίδευσης ή αλλιώς training data και σε δεδομένα ελέγχου ή αλλιώς testing data. Χρησιμοποιούμε τα δεδομένα εκπαίδευσης για να «βοηθήσουμε» τον υπολογιστή να μάθει. Από την εκπαίδευση προκύπτει και δημιουργείται κατάλληλο μαθηματικό μοντέλο που θα μας βοηθήσει στις προβλέψεις και στις αποφάσεις. Κατ' επέκταση θα χρησιμοποιηθούν τα δεδομένα ελέγχου για να δούμε κατά πόσο τα μοντέλα που δημιουργήθηκαν από τους αλγορίθμους μηχανικής μάθησης, μπορούν να δώσουν τα επιθυμητά αποτελέσματα.

Ο διαχωρισμός των δεδομένων σε δεδομένα εκπαίδευσης και δεδομένα ελέγχου στην Python επιτυγχάνεται με την βοήθεια της βιβλιοθήκης scikit-learn. Συγκεκριμένα με την train_test_split(). Θα εφαρμόσουμε τυχαίο διαχωρισμό των δεδομένων σε ποσοστά 70:30, δηλαδή το 70% των δεδομένων θα αποτελέσουν δεδομένα εκπαίδευσης και το υπόλοιπο 30% των δεδομένων θα είναι τα δεδομένα ελέγχου. Έτσι από τις 1086 παρατηρήσεις που προέκυψαν μετά το τέλος της επεξεργασίας των δεδομένων, οι 760 παρατηρήσεις αποτελούν τα δεδομένα εκπαίδευσης ενώ οι υπόλοιπες 326 αποτελούν τα δεδομένα ελέγχου. Σε αυτό το σημείο αξίζει να σημειώσουμε ότι μπορούμε να χρησιμοποιήσουμε την παράμετρο random_state που υπάρχει σαν όρισμα στην μέθοδο train_test_split και να έχουμε κάθε φορά τον ίδιο διαχωρισμό στα δεδομένα εκπαίδευσης και ελέγχου για να παίρνουμε τα ίδια αποτελέσματα από τους αλγορίθμους μηχανικής μάθησης. Αυτό είναι κάτι που θέλουμε για αυτήν την Διπλωματική Εργασία, δηλαδή να έχουμε κοινά αποτελέσματα κάθε φορά από τα μοντέλα, οπότε και χρησιμοποιήσαμε την παράμετρο random_state και θέσαμε την τιμή 2.

Εφόσον ολοκληρώθηκε και ο διαχωρισμός των δεδομένων μπορούμε να δούμε τους αλγορίθμους που θα μας βοηθήσουν στην δημιουργία των μοντέλων κατηγοριοποίησης. Οι αλγόριθμοι μηχανικής μάθησης που επιλέχτηκαν για την πρόβλεψη της

μεταβλητότητας του κρυπτονομίσματος Bitcoin είναι ο Knn, ο Random Forest, τα Νευρωνικά Δίκτυα και η Λογιστική Παλινδρόμηση που θα μας βοηθήσει στην σύγκριση των αποτελεσμάτων των αλγορίθμων και να μας δείξει ποιος έχει τα καλύτερα αποτελέσματα. Όπως είδαμε και από την βιβλιογραφία, αυτοί οι αλγόριθμοι είναι αρκετά δημοφιλείς αλγόριθμοι μηχανικής μάθησης. Έτσι έγινε και η επιλογή τους και για την δικιά μας περίπτωση για να μας βοηθήσουν στην πρόβλεψη της μεταβλητότητας.

Στην συνέχεια θα δούμε την λειτουργία και την μεθοδολογία του κάθε αλγορίθμου ξεχωριστά.

3.4.1 Εφαρμογή Αλγορίθμου Knn

Πρώτο βήμα είναι να εισάγουμε εισάγουμε το KNeighborsClassifier από την scikit learn, για την υλοποίηση του Knn. Στις παραμέτρους που δέχεται ο αλγόριθμος δεν αλλάζουμε κάτι και χρησιμοποιούμε τις προκαθορισμένες τιμές, εκτός από τον αριθμό των γειτόνων. Στον αριθμό των γειτόνων (n_neighbors) ξεκινάμε τις δοκιμές με τον αριθμό 33. Επιλέξαμε αυτόν τον αριθμό για να ξεκινήσουμε τις δοκιμές, για το γεγονός ότι η τετραγωνική ρίζα του 1086, που είναι το σύνολο των παρατηρήσεων είναι περίπου 32,9. Έπειτα πραγματοποιήθηκαν δοκιμές στον αριθμό γειτόνων n, τόσο προς τα πάνω όσο και προς τα κάτω. Ωστόσο δεν υπήρχε κάποια σταθερότητα ως προς την αύξηση ή την μείωση του ποσοστού ακρίβειας που έδινε το μοντέλο. Δηλαδή όσο ανεβαίναμε σε αριθμούς γειτόνων δεν αυξανόταν συνέχεια το ποσοστό ακρίβειας ούτε και μειωνόταν. Το ίδιο συνέβαινε και όταν μειώναμε τον αριθμό των γειτόνων, δεν υπήρχε «σταθερή» αύξηση ή μείωση στην ακρίβεια. Για το γεγονός αυτό και επειδή προτείνεται η τιμή των γειτόνων να είναι περιττός αριθμός για τυχόν διαμάχες σε «ισοπαλία», αποφασίστηκε να κρατήσουμε τον αριθμό 33. Δημιουργούμε λοιπόν ένα αντικείμενο KNeighborsClassifier με μόνη παράμετρο των αριθμό των γειτόνων. Επόμενο βήμα είναι η εκπαίδευση του μοντέλου. Χρησιμοποιούμε την εντολή .fit() και σαν παραμέτρους αυτής βάζουμε τα δεδομένα εκπαίδευσης. Αυτό έχει σαν αποτέλεσμα την εκπαίδευση του μοντέλου.

Επόμενο στάδιο είναι η πρόβλεψη. Πρώτα θα γίνει μία πρόβλεψη για τα δεδομένα εκπαίδευσης που χρησιμοποιήσαμε μέσω της εντολής .predict() και όρισμα τα δεδομένα εκπαίδευσης. Η ίδια διαδικασία ακολουθείτε και για τα δεδομένα ελέγχου. Τα

αποτελέσματα της πρόβλεψης και για τα δεδομένα εκπαίδευσης αλλά και για τα δεδομένα ελέγχου, θα τα δούμε μέσα από το confusion matrix και το classification report. Αυτά είναι εργαλεία για τον υπολογισμό της ακρίβειας, της ανάκλησης, του f1-score. Απαραίτητη είναι πάλι η βιβλιοθήκη scikit-learn και η εισαγωγή των classification_report και confusion_matrix.

3.4.2 Εφαρμογή Αλγορίθμου Random Forest

Επόμενος αλγόριθμος που επιλέχθηκε για δημιουργία μοντέλου, είναι ο αλγόριθμος Random Forest. Για την εφαρμογή του αλγορίθμου χρειαζόμαστε την βιβλιοθήκη scikit-learn και καλούμε το RandomForestClassifier. Έπειτα, πρέπει να δημιουργηθεί ένα αντικείμενο τύπου RandomForestClassifier. Σε αυτό το σημείο, μπορούμε να αλλάξουμε τις τιμές στις παραμέτρους που δέχεται ο αλγόριθμος Random Forest. Από τις προκαθορισμένες τιμές των παραμέτρων δεν θα αλλάξουμε κάτι εκτός από τον αριθμό των δέντρων δηλαδή το n_estimators = 100 και την παράμετρο random_state που πρέπει να την θέσουμε στην τιμή 2. Για να είμαστε σίγουροι και να έχουμε ένα καλό ποσοστό ακρίβειας, τρέξαμε σε ένα for loop τον αλγόριθμο Random Forest για την επιλογή του αριθμού δέντρων ξεκινώντας από το 50 μέχρι και το 150. Παρατηρήθηκε ότι το καλύτερο ποσοστό ακρίβειας ήταν στα 145 δέντρα και έτσι επιλέχθηκε το n_estimators να έχει την τιμή 145. Ακολουθεί η εντολή .fit() με παραμέτρους τα δεδομένα εκπαίδευσης, για να πραγματοποιηθεί η εκπαίδευση του μοντέλου.

Μετά πρέπει να κάνουμε την πρόβλεψη. Χρησιμοποιώντας την εντολή .predict() δίνουμε σαν ορίσματα τόσο τα δεδομένα εκπαίδευσης όσο και τα δεδομένα ελέγχου. Αυτά καταχωρούνται σε 2 μεταβλητές αντίστοιχα. Έπειτα πρέπει να εισάγουμε από το sklearn.metrics τα classification_report και confusion matrix. Τοποθετούμε τις κατάλληλες παραμέτρους, δηλαδή μέρος των δεδομένων εκπαίδευσης και την αντίστοιχη μεταβλητή που προέκυψε από την εντολή .predict() και παίρνουμε τα στοιχεία πρόβλεψης. Αυτό το κάνουμε τόσο για τα δεδομένα εκπαίδευσης όσο και για τα δεδομένα ελέγχου.

3.4.3 Εφαρμογή Αλγορίθμου Νευρωνικών Δικτύων

Επόμενος αλγόριθμος που ακολουθεί, είναι ο αλγόριθμος των Νευρωνικών Δικτύων. Υπάρχουν αρκετοί τρόποι υλοποίησης στην γλώσσα προγραμματισμού Python για ένα νευρωνικό δίκτυο. Σε αυτήν την διπλωματική εργασία επιλέχτηκε να χρησιμοποιηθεί ο αλγόριθμος που παρέχετε από την βιβλιοθήκη scikit-learn. Αυτός είναι ο MLPClassifier. Όπως και με τους προηγούμενους αλγορίθμους, πρέπει να δημιουργήσουμε ένα αντικείμενο MLPClassifier. Ο MLPClassifier δέχεται και αυτός παραμέτρους για να καθορίσουμε την λειτουργία του. Ξεκινάμε με την παράμετρο hidden_layer_sizes και θέτουμε την τιμή 14. Αυτό σημαίνει ότι έχουμε ένα στρώμα και αποτελείται από 14 νευρώνες, όσες δηλαδή και οι μεταβλητές/χαρακτηριστικά των δεδομένων που έχουμε. Η επόμενη παράμετρος που θα αλλάξουμε είναι αυτή των επαναλήψεων, όπου θέτουμε το max_iter στις 10000. Επίσης ο MLP δέχεται σαν παράμετρο το random_state. Σε αυτήν την περίπτωση θα θέσουμε το random_state να είσαι ίσο με 2. Σαν συνάρτηση ενεργοποίησης αφήνουμε την προκαθορισμένη συνάρτηση, που είναι η relu, όπως φαίνεται στην εξίσωση (7).

$$(7) \quad f(x) = \max(0, x)$$

Στην συνέχεια μέσω της εντολής .fit() και παραμέτρους τα δεδομένα εκπαίδευσης, δημιουργούμε το μοντέλο των νευρωνικών δικτύων.

Αφού έγινε και η εκπαίδευση του μοντέλου, σειρά έχει η πρόβλεψη. Θα πάρουμε για ακόμα μία φορά τόσο την πρόβλεψη από τα δεδομένα εκπαίδευσης, όσο και από τα δεδομένα ελέγχου. Χρησιμοποιούμε την εντολή .predict() εισάγοντας σε αυτήν τα training data και testing data και καταχωρούνται τα αποτελέσματα σε 2 μεταβλητές αντιστοίχως. Στην συνέχεια και εδώ εμφανίζουμε τα στοιχεία του πίνακα σύγχυσης και της αναφοράς ταξινόμησης που αφορούν τα δεδομένα εκπαίδευσης αλλά και τα δεδομένα ελέγχου.

3.4.4 Εφαρμογή Αλγορίθμου Λογιστικής Παλινδρόμησης

Τελευταίος αλγόριθμος που θα χρησιμοποιήσουμε είναι αυτός της Λογιστικής Παλινδρόμησης. Για την υλοποίηση και αυτού του αλγορίθμου χρειαζόμαστε την βιβλιοθήκη scikit-learn και εισάγουμε την συνάρτηση LogisticRegression. Όπως και οι υπόλοιποι αλγόριθμοι, δέχονται διάφορες τιμές στις παραμέτρους τους, ωστόσο στην συγκεκριμένη περίπτωση, για την Λογιστική Παλινδρόμηση θα χρησιμοποιηθούν οι προκαθορισμένες τιμές. Ξεκινάμε και εδώ δημιουργώντας ένα αντικείμενο της συνάρτησης LogisticRegression(). Στην συνέχεια γίνεται η εκπαίδευση του μοντέλου με την εντολή fit() όπου δέχεται σαν ορίσματα τα δεδομένα εκπαίδευσης.

Ακολουθεί η ίδια διαδικασία στο κομμάτι της πρόβλεψης όπως και με τους υπόλοιπους αλγορίθμους. Αναθέτουμε τα αποτελέσματα της εντολής predict() σε 2 μεταβλητές, μία για τα δεδομένα εκπαίδευσης και μία για τα δεδομένα ελέγχου. Εισάγουμε από το sklearn.metris τα εργαλεία για τον υπολογισμό και την εμφάνιση του πίνακα σύγχυσης και της αναφοράς ταξινόμησης (confusion_matrix() και classification_report()). Έχουμε πλέον και για την Λογιστική Παλινδρόμηση τα απαραίτητα στοιχεία για να κάνουμε τις συγκρίσεις και να δούμε πιο μοντέλο αποδίδει καλύτερα.

3.4.5 Διαδικασία K-fold Cross Validation

Μέχρι στιγμής, έγινε ο διαχωρισμός των δεδομένων μία φορά (η διαδικασία αυτή στο machine learning είναι γνωστή και ως hold-out). Χωρίστηκαν τα δεδομένα με τυχαίο τρόπο σε ποσοστά 70% και 30%, τα οποία δεδομένα θα χρησιμοποιηθούν για εκπαίδευση και για τον έλεγχο αντίστοιχα. Ωστόσο μπορούμε να καταλάβουμε ότι τα ποσοστά ακρίβειας που θα παίρναμε από το κάθε μοντέλο, δεν θα ήταν αξιόπιστα γιατί το τρέξιμο και η εφαρμογή του καθενός έγινε μόνο μία φορά. Αν χωρίζαμε ξανά τα δεδομένα σε δεδομένα εκπαίδευσης και δεδομένα ελέγχου και αλλάζοντας τιμή στην παράμετρο random_state, τα αποτελέσματα όσο αναφορά τα ποσοστά ακρίβειας μπορεί να είχαν σημαντικές διαφορές. Έτσι αποφασίσαμε να εφαρμόσουμε την διαδικασία που είναι γνωστή ως k-fold cross validation.

Σύμφωνα με την διαδικασία k-fold cross validation, εισάγουμε τα δεδομένα που έχουμε και αυτά χωρίζονται σε k τμήματα. Κάθε τμήμα δεδομένων θα χρησιμοποιηθεί κάποια

στιγμή σαν testing data και τα υπόλοιπα k-1 θα αποτελούν το training data. Οι πιο συνηθισμένες τιμές της μεταβλητής k είναι 5 ή 10. Σε αυτήν την διπλωματική εργασία αποφασίσαμε να δώσουμε στο k την τιμή 10. Αυτό σημαίνει ότι τα δεδομένα μας θα χωριστούν σε 10 τμήματα, από τα οποία κάθε ένα ξεχωριστά θα αποτελεί τα δεδομένα ελέγχου ενώ τα υπόλοιπα 9 θα χρησιμοποιηθούν για εκπαίδευση.

Υπάρχουν αρκετά είδη k-fold cross validation. Στην δικιά μας περίπτωση επιλέχθηκε να χρησιμοποιηθεί η μέθοδος Stratified k-fold cross validation. Στο k-fold cross validation ο διαχωρισμός των τμημάτων γίνεται με τελείως τυχαίο τρόπο. Στην περίπτωση του stratified k-fold cross validation έχουμε stratified sampling (στρωματοποιημένη δειγματοληψία), δηλαδή κάθε τμήμα που δημιουργείται είναι αντιπροσωπευτικό των αρχικών δεδομένων.

Για την εφαρμογή του stratified k-fold cross validation δουλέψαμε ως εξής. Αρχικά δημιουργήθηκε μία συνάρτηση με όνομα get_score,η οποία δέχεται σαν παραμέτρους το μοντέλο του αλγορίθμου και τα δεδομένα εκπαίδευσης και ελέγχου. Έπειτα γίνεται εκπαίδευση του μοντέλου με την εντολή fit(). Η συνάρτηση επιστρέφει την ακρίβεια μέσω της εντολής score() και ορίσματα τα δεδομένα ελέγχου.

Επόμενο βήμα είναι να εισάγουμε από το sklearn.model_selection το StratifiedKFold. Δημιουργούμε ένα αντικείμενο StratifiedKFold και βάζουμε σαν όρισμα n_splits= 10 και επειδή θέλουμε τα ίδια αποτελέσματα κάθε φορά που τρέχουμε το πρόγραμμά μας βάζουμε και εδώ την παράμετρο random_state να είσαι ίση με 2. Όμως εφόσον βάλαμε τιμή στην random_state, πρέπει να θέσουμε και την παράμετρο shuffle ως True. Έπειτα δημιουργούμε 4 λίστες, μία για κάθε αλγόριθμο για να αποθηκεύσουμε τις τιμές ακρίβειας για κάθε τμήμα. Κάθε λίστα θα περιέχει 10 τιμές, από τις οποίες στην συνέχεια θα πάρουμε τον μέσο όρο. Μέσω ενός for loop χωρίζουμε κάθε φορά τα δεδομένα για να γίνει ο έλεγχος της ακρίβειας και χρησιμοποιούμε την εντολή append() και εντός αυτής καλούμε την συνάρτηση get_score() και εισάγουμε σε κάθε επανάληψη την τιμή της ακρίβειας του κάθε τμήματος. Στην συνέχεια εκτυπώνουμε την λίστα με τις τιμές του κάθε μοντέλου και υπολογίζουμε με την βοήθεια της βιβλιοθήκης numpy τον μέσο όρο.

4. Αποτελέσματα Μοντέλων

Μέχρι στιγμής έχουμε δει τι πραγματεύεται αυτή η διπλωματική εργασία. Έχει γίνει μία βιβλιογραφική ανασκόπηση πάνω στα αντικείμενα που εξετάζονται, έγινε αναφορά για την συλλογή, την καταγραφή και την επεξεργασία των δεδομένων. Έχουμε δει στοιχεία από την λειτουργία και την εφαρμογή των μοντέλων στην γλώσσα προγραμματισμού Python. Σε αυτό το κομμάτι της διπλωματικής εργασίας, θα γίνει η καταγραφή και η παρουσίαση των αποτελεσμάτων των 4 μοντέλων που επιλέχθηκαν. Στην αρχή έγινε χρήση της διαδικασίας hold-out, χωρίζοντας τα δεδομένα σε ποσοστά 70-30. Το 70% αποτελεί τα δεδομένα εκπαίδευσης και το υπόλοιπο 30% τα δεδομένα ελέγχου. Όμως επειδή δεν θα ήταν απολύτως σωστά τα αποτελέσματα αν γίνει η εφαρμογή τους μόνο μία φορά, επιλέξαμε να χρησιμοποιήσουμε και το stratified k-fold cross validation. Οπότε θα δούμε τα αποτελέσματα για τους αλγορίθμους Knn, Random Forest, Νευρωνικών Δικτύων και Λογιστικής Παλινδρόμησης της διαδικασίας hold-out τόσο για τα δεδομένα εκπαίδευσης όσο και για τα δεδομένα ελέγχου καθώς και τα αποτελέσματα του κάθε ένα από το stratified k-fold cross validation.

4.1 Αποτελέσματα Αλγορίθμου Knn

Ξεκινάμε με τα αποτελέσματα που αφορούν τον αλγόριθμο Knn. Στο πρώτο στάδιο θα δούμε τις προβλέψεις τόσο των δεδομένων εκπαίδευσης όσο και των δεδομένων ελέγχου στον διαχωρισμό που πραγματοποιήσαμε σε ποσοστά 70-30. Έπειτα θα δούμε τα ποσοστά που προέκυψαν από το Stratified k-fold cross validation για k=10. Έχουμε δημιουργήσει αρχικά ένα αντικείμενο του KNeighborsClassifier και στην συνέχεια χρησιμοποιούμε τις εντολές fit() και predict() για εκπαίδευση και πρόβλεψη αντίστοιχα. Στον Πίνακα 6 και στον Πίνακα 7 έχουμε τον πίνακα σύγχυσης και την αναφορά ταξινόμησης αντίστοιχα που αφορούν τα δεδομένα εκπαίδευσης. Σε αυτό το σημείο, να θυμίσουμε ότι σύμφωνα με την στήλη ‘Price Status’, στην οποία βασίσαμε την κατηγοριοποίηση, η τιμή 1 σημαίνει ότι έχουμε αύξηση της τιμής ενώ η τιμή -1 σημαίνει ότι έχουμε μείωση.

	-1	**1**
-1	199	168
1	138	255

Πίνακας 6: Confusion Matrix δεδομένων εκπαίδευσης με Knn

	precision	**recall**	**f1-score**	**support**
-1	0,59	0,54	0,57	367
1	0,60	0,65	0,62	393
accuracy			**0,60**	760
macro avg	0,60	0,60	0,60	760
weighted avg	0,60	0,60	0,60	760

Πίνακας 7: Classification Report δεδομένων εκπαίδευσης με Knn

Μπορούμε να δούμε με μία πρώτη ματιά πως συμπεριφέρθηκε το μοντέλο Knn μετα δεδομένα εκπαίδευσης και να φτάνει σε ποσοστό ακρίβειας 60% και αφορά τα δεδομένα εκπαίδευσης που ήταν 760. Στους Πίνακες 8 και 9 αντίστοιχα θα δούμε τιμές για τα καινούρια δεδομένα, τα δεδομένα ελέγχου που είναι στο σύνολο 326.

	-1	**1**
-1	80	75
1	71	100

Πίνακας 8: Confusion Matrix δεδομένων ελέγχου με Knn

	precision	**recall**	**f1-score**	**support**
-1	0,53	0,52	0,52	155
1	0,57	0,58	0,58	171
accuracy			**0,55**	326
macro avg	0,55	0,55	0,55	326
weighted avg	0,55	0,55	0,55	326

Πίνακας 9: Classification Report δεδομένων ελέγχου με Knn

Μπορούμε να δούμε ότι το ποσοστό ακρίβειας είναι στο 55%..

Στην συνέχεια εφαρμόζουμε το Stratified k-fold cross validation με k= 10 για το μοντέλο Knn και βλέπουμε τα αποτελέσματα στον Πίνακα 10.

	Ακρίβεια
1	0.5963302752293578
2	0.5688073394495413
3	0.5321100917431193
4	0.5871559633027523
5	0.5963302752293578
6	0.5045871559633027
7	0.5277777777777778
8	0.5185185185185185
9	0.5185185185185185
10	0.46296296296296297
Μέσος όρος	**0.5413098878695208**

Πίνακας 10: Αποτελέσματα Stratified k-fold για Knn

Χρησιμοποιώντας την μέθοδο Stratified k-fold για k= 10, δηλαδή την εξέταση 10 διαφορετικών αντιπροσωπευτικών τμημάτων των δεδομένων μας, σαν δεδομένα ελέγχου, ο μέσος όρος των ποσοστών ακρίβειας είναι περίπου στο 54%.

4.2 Αποτελέσματα Αλγορίθμου Random Forest

Επόμενος αλγόριθμος του οποίου θα δούμε τα αποτελέσματα είναι ο Random Forest. Θα ξεκινήσουμε βλέποντας τα αποτελέσματα του διαχωρισμού που πραγματοποιήθηκε σε ποσοστά 70-30. Δημιουργούμε αντικείμενο RandomForestClassifier και στην συνέχεια εκπαιδεύουμε το μοντέλο και προχωράμε στις προβλέψεις. Ξεκινάμε με τα αποτελέσματα των δεδομένων εκπαίδευσης στους Πίνακες 11 και 12.

	-1	**1**
-1	367	0
1	0	393

Πίνακας 11: Confusion Matrix δεδομένων εκπαίδευσης με Random Forest

	precision	**recall**	**f1-score**	**support**
-1	1,00	1,00	1,00	367
1	1,00	1,00	1,00	393
accuracy			**1,00**	760
macro avg	1,00	1,00	1,00	760
weighted avg	1,00	1,00	1,00	760

Πίνακας 12: Classification Report δεδομένων εκπαίδευσης με Random Forest

Το μοντέλο εκπαίδευσης μπορούμε να δούμε ότι έχει απόλυτη ακρίβεια με τα δεδομένα εκπαίδευσης, πάνω στα οποία σχεδιάστηκε. Αυτό όμως δεν θα πρέπει να έχει μεγάλη σημασία. Στους Πίνακες 13 και 14 μπορούμε να δούμε τα αποτελέσματα που πήραμε για τα δεδομένα ελέγχου, που παρουσιάζουν μεγαλύτερο ενδιαφέρον αφού αποτελούν σημαντικό κομμάτι αυτού που πραγματεύεται η διπλωματική εργασία.

	-1	**1**
-1	83	72
1	60	111

Πίνακας 13: Confusion Matrix δεδομένων ελέγχου με Random Forest

	precision	**recall**	**f1-score**	**support**
-1	0,58	0,54	0,56	155
1	0,61	0,65	0,63	171
accuracy			**0,60**	326
macro avg	0,59	0,59	0,59	326
weighted avg	0,59	0,60	0,59	326

Πίνακας 14: Classification Report δεδομένων ελέγχου με Random Forest

Το ποσοστό ακρίβειας για τις 326 μετρήσεις των δεδομένων ελέγχου ανέρχεται στο 60%.

Ακολουθεί η εφαρμογή του Stratified k-fold για k= 10 για τον αλγόριθμο Random Forest με τα αποτελέσματα του να αποτυπώνονται στον Πίνακα 15.

	Ακρίβεια
1	0.6605504587155964

2	0.6697247706422018
3	0.6605504587155964
4	0.6788990825688074
5	0.6697247706422018
6	0.6972477064220184
7	0.6203703703703703
8	0.75
9	0.6759259259259259
10	0.5740740740740741
Μέσος όρος	**0.6657067618076793**

Πίνακας 15: Αποτελέσματα Stratified k-fold για Radom Forest

Μετά και από την χρήση του Stratified k-fold cross validation για τον αλγόριθμο Random Forest, βλέπουμε ότι ο μέσος όρος των ποσοστών ακρίβειας ανέρχεται περίπου στο 66%.

4.3 Αποτελέσματα Αλγορίθμου Νευρωνικών Δικτύων

Στην συνέχεια ακολουθεί ο αλγόριθμος των Νευρωνικών Δικτύων και τα αποτελέσματά του. Δημιουργούμε ένα αντικείμενο MLPClassifier και βάζουμε τις κατάλληλες παραμέτρους. Χρησιμοποιούμε τις εντολες fit() και predict() για την εκπαίδευση και την πρόβλεψη. Πρώτα θα δούμε τα αποτελέσματα του hold-out (διαχωρισμός δεδομένων σε 70-30). Τα confusion matrix και classification report για τα δεδομένα εκπαίδευσης φαίνονται στους Πίνακες 16 και 17.

	-1	**1**
-1	111	256

1	94	299

Πίνακας 16: Confusion Matrix δεδομένων εκπαίδευσης με Νευρωνικά Δίκτυα

	precision	**recall**	**f1-score**	**support**
-1	0,54	0,30	0,39	367
1	0,54	0,76	0,63	393
accuracy			**0,54**	760
macro avg	0,54	0,53	0,51	760
weighted avg	0,54	0,54	0,51	760

Πίνακας 17: Classification Report δεδομένων εκπαίδευσης με Νευρωνικά Δίκτυα

Το ποσοστό ακρίβειας στα δεδομένα εκπαίδευσης έφτασε το 54%. Μεγαλύτερη σημασία έχει όμως ποια θα είναι τα αποτελέσματα για τα δεδομένα ελέγχου. Μπορούμε να τα διακρίνουμε στους Πίνακες 18 και 19.

	-1	**1**
-1	45	110
1	45	126

Πίνακας 18: Confusion Matrix δεδομένων ελέγχου με Νευρωνικά Δίκτυα

	precision	**recall**	**f1-score**	**support**
-1	0,50	0,29	0,37	155
1	0,53	0,74	0,62	171

accuracy			**0,52**	326
macro avg	0,52	0,51	0,49	326
weighted avg	0,52	0,52	0,50	326

Πίνακας 19: Classification Report δεδομένων ελέγχου με Νευρωνικά Δίκτυα

Το ποσοστό για τα δεδομένα ελέγχου στον αλγόριθμο Νευρωνικών Δικτύων ανέρχεται στο 52%. Μένει όμως να δούμε και τα αποτελέσματα με την μέθοδο Stratified k-fold για να έχουμε μια πιο ολοκληρωμένη άποψη. Τα αποτελέσματα αυτής φαίνονται στον Πίνακα 20.

	Ακρίβεια
1	0.5321100917431193
2	0.7614678899082569
3	0.5871559633027523
4	0.5045871559633027
5	0.6146788990825688
6	0.42201834862385323
7	0.8518518518518519
8	0.6388888888888888
9	0.5555555555555556
10	0.5370370370370371
Μέσος όρος	**0.6005351681957186**

Πίνακας 20: Αποτελέσματα Stratified k-fold για Νευρωνικά Δίκτυα

Όπως μπορούμε να δούμε, ο μέσος όρος της ακρίβειας των αποτελεσμάτων από το Stratified k-fold cross validation για τα Νευρωνικά Δίκτυα φτάνει το 60%.

4.4 Αποτελέσματα Αλγορίθμου Λογιστικής Παλινδρόμησης

Τελευταίος αλγόριθμος του οποίου θα δούμε τα αποτελέσματα είναι αυτός της Λογιστικής Παλινδρόμησης. Ξεκινάμε και εδώ παρουσιάζοντας τα αποτελέσματα της μεθόδου hold-out με τον διαχωρισμό των δεδομένων να γίνεται με τυχαίο τρόπο σε ποσοστά 70-30, με το 70% να αποτελεί τα δεδομένα εκπαίδευσης και το 30% τα δεδομένα ελέγχου. Δημιουργούμε αντικείμενο LogisticRegression και έπειτα καλούνται οι εντολές fit() και predict(). Τα πρώτα αποτελέσματα αφορούν τα δεδομένα εκπαίδευσης και φαίνονται στους παρακάτω Πίνακες 21 και 22.

	-1	**1**
-1	196	171
1	99	294

Πίνακας 21: Confusion Matrix δεδομένων εκπαίδευσης με Λογιστική Παλινδρόμηση

	precision	**recall**	**f1-score**	**support**
-1	0,66	0,53	0,59	367
1	0,63	0,75	0,69	393
accuracy			**0,64**	760
macro avg	0,65	0,64	0,64	760
weighted avg	0,65	0,64	0,64	760

Πίνακας 22: Classification Report δεδομένων εκπαίδευσης με Λογιστική Παλινδρόμηση

Το ποσοστό ακρίβειας για τα δεδομένα εκπαίδευσης στην Λογιστική Παλινδρόμηση φτάνει το 64%. Μας ενδιαφέρει όμως περισσότερο τι γίνεται με τα δεδομένα ελέγχου. Αυτά μπορούμε να τα διακρίνουμε στους Πίνακες 23 και 24.

	-1	**1**
-1	86	69
1	52	119

Πίνακας 23: Confusion Matrix δεδομένων ελέγχου με Λογιστική Παλινδρόμηση

	precision	**recall**	**f1-score**	**support**
-1	0,62	0,55	0,59	155
1	0,63	0,70	0,66	171
accuracy			**0,63**	326
macro avg	0,63	0,63	0,62	326
weighted avg	0,63	0,63	0,63	326

Πίνακας 24: Classification Report δεδομένων ελέγχου με Λογιστική Παλινδρόμηση

Η ακρίβεια που προέκυψε από το μοντέλο της Λογιστικής Παλινδρόμησης είναι της τάξεως του 63%. Ακολουθεί παρακάτω στον Πίνακα 25 και τα αποτελέσματα του Stratified k-fold cross validation για τον αλγόριθμο της Λογιστικής Παλινδρόμησης.

	Ακρίβεια
1	0.5963302752293578
2	0.5412844036697247
3	0.6330275229357798

4	0.5871559633027523
5	0.6880733944954128
6	0.5779816513761468
7	0.7037037037037037
8	0.6851851851851852
9	0.6111111111111112
10	0.6203703703703703
Μέσος όρος	**0.6244223581379543**

Πίνακας 25: Αποτελέσματα Stratified k-fold για Λογιστική Παλινδρόμηση

Ο μέσος όρος των ποσοστών που προέκυψε από τα 10 τμήματα που σχηματίστηκαν και εξετάστηκαν από το Stratified k-fold είναι περίπου στο 62%. Είναι σχεδόν της ίδιας τάξης με το ποσοστό της μεθόδου hold-out.

4.5 Σύγκριση αποτελεσμάτων

Έχει ολοκληρωθεί η δημιουργία των μοντέλων και έχουμε πάρει πλέον τις προβλέψεις για τα δεδομένα που έχουμε συγκεντρώσει. Στα αποτελέσματα της μεθόδου hold-out έχουμε καλύτερη απόδοση με την Λογιστική Παλινδρόμηση, που φτάνει το 63%. Μετά έχουμε το μοντέλο από το Random Forest με 60% και ακολουθούν ο Knn με 55% και τα Νευρωνικά Δίκτυα με 52%. Τώρα θα παρουσιάσουμε και τα αποτελέσματα της μεθόδου Stratified k-fold cross validation και τον μέσο όρο των αποτελεσμάτων των 10 δοκιμών για μια πιο ολοκληρωμένη άποψη. Καλύτερη απόδοση, σύμφωνα με την ακρίβεια που προέκυψε από το μοντέλο έχει ο αλγόριθμος Random Forest με την ακρίβεια να είναι περίπου στο 66%. Ακολουθεί το μοντέλο της Λογιστικής Παλινδρόμησης με ποσοστό που φτάνει στο 62%. Συνεχίζουμε με το μοντέλο των Νευρωνικών Δικτύων MLP που συγκεντρώνει ποσοστό ακρίβειας στο 60%. Τελευταίος είναι ο αλγόριθμος Knn, με το ποσοστό του να ανέρχεται στο 54%.

5. Συμπεράσματα

Σκοπός αυτής της διπλωματικής εργασίας ήταν να γίνει μια προσπάθεια πρόβλεψης της κατεύθυνσης της μεταβλητότητας του κρυπτονομίσματος Bitcoin. Αρχικό στάδιο, ήταν η μελέτη της απαραίτητης βιβλιογραφίας. Παρουσιάστηκαν και αναλύθηκαν έννοιες που θα συναντούσαμε αρκετά στα πλαίσια υλοποίησης της παρούσας διπλωματικής εργασίας. Ξεκινήσαμε κάνοντας μια εισαγωγή στον κόσμο των κρυπτονομισμάτων. Τι είναι, τι εξυπηρετούν, πως και γιατί δημιουργήθηκαν. Διαδόθηκε η ιδέα του κρυπτονομίσματος και στην συνέχεια αναπτύχθηκαν αρκετά από αυτά και έφτασαν σε σημείο να έχουν μεγάλη χρηματική αξία με πρώτο το κρυπτονόμισμα Bitcoin.

Αφού επικεντρωθήκαμε στο bitcoin, ασχοληθήκαμε λίγο περισσότερο με αυτό. Γίνεται μια αναφορά στην δημιουργία του και την λειτουργία του μέσα από την τεχνολογία blockchain και πως ήταν η αρχή για την δημιουργία των υπολοίπων κρυπτονομισμάτων.

Αμέσως μετά είδαμε την έννοια της μεταβλητότητας και τα είδη αυτής. Αναφέρεται για συγκεκριμένο χρονικό διάστημα και γίνεται αναφορά ότι αν κάτι παρουσιάζει μεγάλη μεταβλητότητα, τότε παρουσιάζονται και επενδυτικές ευκαιρίες.

Αυτό συνέβη και με το bitcoin. Χαρακτηρίζεται από μεγάλη μεταβλητότητα και αυτό το έχει οδηγήσει σε μεγάλες διακυμάνσεις στην τιμή του. Επόμενο στάδιο ήταν η προσπάθεια εξήγησης αυτής της μεγάλης μεταβλητότητας που παρουσιάζει αυτό το κρυπτονόμισμα. Έτσι παρουσιάστηκαν κάποιες εργασίες και οι προσπάθειες που έγιναν για την εξήγηση της μεταβλητότητας, της τιμής και τι μπορεί να τα επηρεάσει αυτά, κυρίως μέσα από οικονομετρικά μοντέλα.

Ακολούθησαν εργασίες για την πρόβλεψη της μεταβλητότητας και της τιμής. Έγινε συνδυασμός τόσο οικονομετρικών μοντέλων όσο και αλγορίθμων μηχανικής μάθησης.

Έπειτα κάνουμε εισαγωγή στον κόσμο της μηχανικής μάθησης. Γίνεται αναφορά στο τι είναι η μηχανική μάθηση, πότε ξεκίνησε και ποια είναι τα είδη της. Αποτελεί κλάδο του Data Science και συνδέεται με την εξαγωγή/εξόρυξη δεδομένων αλλά και το Deep Learning. Παρουσιάζονται κάποιοι κλάδοι που εφαρμόζουν σήμερα την μηχανική μάθηση και τέλος κάνουμε μία παρουσίαση στις πιο δημοφιλείς βιβλιοθήκες μηχανικής μάθησης

για την γλώσσα προγραμματισμού Python, μιας και είναι η γλώσσα που θα χρησιμοποιηθεί σε αυτήν την εργασία.

Στην συνέχεια γίνεται αναφορά στην κατηγοριοποίηση, που είναι τεχνική της εξόρυξης δεδομένων. Μας ενδιαφέρει καθώς αναφέραμε ότι πρόκειται να δούμε το ζητούμενο της εργασίας σαν πρόβλημα κατηγοριοποίησης. Ακολουθούν οι αλγόριθμοι που θα χρησιμοποιήσουμε. Αυτοί είναι ο Knn, ο Random Forest, τα Νευρωνικά Δίκτυα και η Λογιστική Παλινδρόμηση. Γίνεται μία παρουσίαση του καθενός στο πως δουλεύουν, στα ιδιαίτερα χαρακτηριστικά τους και αναφέρονται κάποιοι κλάδοι στους οποίους εφαρμόζεται ο καθένας μαζί με κάποιες εργασίες.

Μετά την βιβλιογραφική επισκόπηση σειρά είχε η ανάκτηση των δεδομένων. Τα δεδομένα της παρούσας διπλωματικής εργασίας τα πήραμε από τις ιστοσελίδες finance.yahoo.com και blockchain.com. Ακολούθησε μία περιγραφή των μεταβλητών αυτών.

Πριν αρχίσουμε την επεξεργασία δεδομένων, παρουσιάσαμε τα τεχνικά χαρακτηριστικά του τοπικού συστήματος μαζί με το λογισμικό και την γλώσσα προγραμματισμού αλλά και τις βιβλιοθήκες που χρησιμοποιήθηκαν. Αμέσως μετά ακολουθεί μία αναλυτική παρουσίαση της επεξεργασίας των δεδομένων, ξεκινώντας αρχικά με το να διαβάσουμε τα αρχεία csv και στην συνέχεια την επιλογή του χρονικού διαστήματος μέχρι και την δημιουργία της στήλης στην οποία θα βασιστούμε για την κατηγοριοποίηση αλλά και να τελειώσουμε με την κανονικοποίηση των δεδομένων. Έπειτα παρουσιάζονται τα περιγραφικά στατιστικά των μεταβλητών.

Τέλος, ακολουθεί η μεθοδολογία που ακολουθήσαμε για τον διαχωρισμό των δεδομένων και πως έγινε η εφαρμογή των αλγορίθμων σε συνδυασμό με τις παραμέτρους του καθενός. Αυτό ήταν η μέθοδος hold-out, αλλά επειδή δεν θα ήταν αρκετή για τα συμπεράσματα ακολουθήθηκε και η διαδικασία k-fold cross validation και συγκεκριμένα το stratified k-fold cross validation με k=10 όπου θα πάρουμε τον μέσο όρο των 10 μετρήσεων για κάθε αλγόριθμο.

Ακολουθούν τα αποτελέσματα για τον κάθε αλγόριθμο. Κατά γενική ομολογία, πρέπει να θεωρήσουμε τα αποτελέσματα που λάβαμε ικανοποιητικά. Σύμφωνα με την βιβλιογραφία και τις έρευνες όπου παρουσίασαν το ποσοστό ακρίβειας είχαμε περίπου ίδια ή και

βελτιωμένα αποτελέσματα. Ωστόσο αυτό οφείλεται στα δεδομένα και στο πόσο εξειδικευμένη είναι η κάθε έρευνα.

Ξεκινάμε με το μοντέλο του αλγορίθμου Knn. Ο Knn συγκέντρωσε 55% από την μέθοδο hold-out και 54% από το stratified k-fold. Είναι από τους πιο εύχρηστους και από τους πιο παλιούς αλγόριθμους μηχανικής μάθησης. Το ποσοστά των 2 μεθόδων είναι κοντά, όμως δεν είναι και απόλυτα ικανοποιητικά αφού είναι κοντά στο 50% και πιθανόν να χρειάζεται περισσότερη μελέτη ή και αλλαγές στις παραμέτρους και πιθανόν διαφοροποίηση των δεδομένων για να δούμε αν μπορεί να αποδώσει καλύτερα.

Σειρά έχει ο αλγόριθμος Random Forest. Το μοντέλο που προέκυψε από εδώ συγκέντρωσε ποσοστό ακρίβειας 66% για την μέθοδο stratified k-fold και ποσοστό 60% με την μέθοδο hold-out. Βλέπουμε μία διαφορά στα ποσοστά στις 2 μεθόδους. Ήταν το καλύτερο αποτέλεσμα από τους 4 αλγορίθμους για την μέθοδο stratified k-fold, που εξετάστηκαν στην παρούσα Διπλωματική Εργασία. Γνωρίζουμε ότι είναι από τους πιο αξιόπιστους αλγορίθμους για machine learning. Χαρακτηρίζεται από απλότητα και από ποικιλομορφία και γι' αυτό είναι από τους πιο χρησιμοποιημένους αλγορίθμους.

Ακολουθεί το μοντέλο των Νευρωνικών Δικτύων. Το ποσοστό του ανέρχεται στο 60% για την μέθοδο stratified k-fold, ενώ για την μέθοδο hold-out συγκέντρωσε ποσοστό 52%. Βλέπουμε μία μεγάλη διαφορά στα ποσοστά και το αποτέλεσμα από τον διαχωρισμό των δεδομένων σε 70-30 δεν είναι ικανοποιητικό και πιθανόν να χρειάζεται περισσότερη μελέτη. Είναι αρκετά υποσχόμενα τα Νευρωνικά Δίκτυα και περιμέναμε καλύτερη απόδοση και βάση βιβλιογραφίας. Πιθανόν να σχετίζεται με τα δεδομένα και το συγκεκριμένο random_state που χρησιμοποιήθηκε. Υπάρχουν πολλά είδη Νευρωνικών Δικτύων, ωστόσο επιλέχτηκε να γίνει χρήση του MLP. Όπως έχουμε αναφέρει χρησιμοποιούνται και για κατηγοριοποίηση και για παλινδρόμηση. Δίνεται η δυνατότητα στα Νευρωνικά Δίκτυα να μαθαίνουν μέσα από τις επαναλήψεις και αυτός είναι και ο λόγος που επιλέξαμε τόσες πολλές. Χρησιμοποιούνται όλο και σε περισσότερους τομείς. Το μειονέκτημά τους είναι ότι δεν ξέρουμε πως πραγματικά δουλεύουν. Αποτελούν «μαύρο κουτί» και έτσι δεν μπορούμε να ξέρουμε πως και από ποιες μεταβλητές των δεδομένων που χρησιμοποιούμε επηρεάζονται.

Ο τελευταίος αλγόριθμος που εξετάσαμε είναι της Λογιστικής Παλινδρόμησης. Το μοντέλο της Λογιστικής Παλινδρόμησης απέδωσε ακρίβεια 63% από την μέθοδο hold-out

και 62% από το stratified k-fold. Έχει το καλύτερο ποσοστό για την μέθοδο hold-out και τα ποσοστά είναι σχεδόν ίδια για τις 2 μεθόδους. Η Λογιστική Παλινδρόμηση χρησιμοποιείται πολύ στην στατιστική και τον τομέα των οικονομικών επιστημών. Είναι από τους πιο βασικούς αλγορίθμους και συμπεριλαμβάνεται σχεδόν πάντα όταν αναπτύσσονται μοντέλα μηχανικής μάθησης. Για την παρούσα Διπλωματική Εργασία αποτέλεσε και το κριτήριο σύγκρισης των μοντέλων που προέκυψαν από τους αλγορίθμους μηχανικής μάθησης.

Σε αυτό το σημείο είναι αρκετά σημαντικό να τονίσουμε ότι έγινε γίνει χρήση της παραμέτρου random_state. Αυτό το κάναμε για να έχουμε τα ίδια αποτελέσματα από τους αλγορίθμους μηχανικής μάθησης κάθε φορά που τρέχουμε τον κώδικα. Για διαφορετική τιμή στην παράμετρο random_state θα είχαμε διαφορετικά αποτελέσματα, ωστόσο δεν θα ήταν εύκολο να τα παρουσιάζαμε και να βγάζαμε κάποιο συμπέρασμα.

Το Bitcoin έχει μπει για τα καλά στην ζωή μας. Είναι το πιο δημοφιλές κρυπτονόμισμα και έχει καταφέρει να κεντρίσει το ενδιαφέρον στους κλάδους των οικονομικών αλλά και της πληροφορικής. Έχει αποτελέσει το αντικείμενο έρευνας και για τους 2 κλάδους. Χαρακτηρίζεται από μεγάλη μεταβλητότητα και έχουν γίνει προσπάθειες για την πρόβλεψη αυτής τόσο με οικονομικά μοντέλα, όσο και με την μηχανική μάθηση. Αυτό ερευνά και πραγματεύεται η παρούσα Διπλωματική Εργασία. Προσπαθήσαμε να αντιμετωπίσουμε την πρόβλεψη της μεταβλητότητας σαν ένα πρόβλημα κατηγοριοποίησης (1 για αύξηση τιμής και -1 για μείωση τιμής). Χρησιμοποιήθηκαν κάποια μοντέλα μηχανικής μάθησης με σκοπό να επιχειρήσουμε να κάνουμε αυτήν την πρόβλεψη. Επιλέχθηκαν δεδομένα και μεταβλητές που πιστεύουμε πως επηρεάζουν την τιμή του και οδηγηθήκαμε σε κάποιες προβλέψεις. Καταλαβαίνουμε ότι υπάρχουν περιθώρια βελτίωσης αυτών των ποσοστών. Επομένως υπάρχει η δυνατότητα σε μελλοντικές έρευνες και εργασίες τόσο από πλευράς αλγορίθμων και μηχανικής μάθησης, όσο και από πλευράς δεδομένων να έχουμε καλύτερα αποτελέσματα και με μεγαλύτερη ακρίβεια. Αυτό γιατί υπάρχουν πολλά μοντέλα μηχανικής μάθησης που μπορούν να εξεταστούν για να δούμε πιο έχει την καλύτερη απόδοση αλλά και πολλές ακόμα μεταβλητές και δεδομένα να χρησιμοποιηθούν αλλά και να εξεταστούν ποιες επηρεάζουν περισσότερο την μεταβλητότητα και ποιες όχι.

Βιβλιογραφία

Ακολουθούν οι βιβλιογραφικές αναφορές (πηγές) της Εργασίας.

Frankenfield J., 'Cryptocurrency Definition' (2022). Ανακτήθηκε 25 Απριλίου 2022, από την ιστοσελίδα: https://www.investopedia.com/terms/c/cryptocurrency.asp

Nakamoto S. (2008) Bitcoin: A Peer-to-Peer Electronic Cash System (PDF Report: https://bitcoin.org/bitcoin.pdf)

Frankenfield J., 'Bitcoin Definition' (2021). Ανακτήθηκε 25 Απριλίου 2022, από την ιστοσελίδα: https://www.investopedia.com/terms/b/bitcoin.asp

Fidelity, 'Market volatility: defined and explained' (χ.χ.) Ανακτήθηκε 28 Απριλίου 2022, από την ιστοσελίδα: https://www.fidelity.com.sg/beginners/what-is-volatility/market-volatility

Macroption, 'Difference between Implied, Realized and Historical Volatility' (χ.χ.) Ανακτήθηκε 28 Απριλίου 2022, από την ιστοσελίδα: https://www.macroption.com/implied-vs-realized-vs-historical-volatility/

Bhowmik R. & Wang S. (2020) Stock Market Volatility and Return Analysis: A Systematic Literature Review. *Entropy, 2020, 22*(5), 522, doi: 10.3390/e22050522

Katsiampa, P. (2017) Volatility Estimation for Bitcoin: A comparison of GARCH models. *Economics Letters, 2017, 158*, 3-6, doi: 10.1016/j.econlet.2017.06.023

Balsilar, M. & Bouri, E. & Gupta, R. & Roubaud, D. (2017) Can volume predict Bitcoin returns and volatility? A quantiles-based approach. *Economic Modeling, 2017, 64*, 74-81, doi: 10.1016/j.econmod.2017.03.019

Aalborg, H. & Molnár, P. & de Vries, J. (2019) What can explain the price, volatility and trading volume of Bitcoin? *Finance Research Letters, 2019*, *29*, 255-265, doi: 10.1016/j.frl.2018.08.010

Baur, D. & Dimpfl, T. (2017) Realized Bitcoin Volatility. *SSRN Electronic Journal, 2017,* doi: 10.2139/ssrn.2949754

Pichl, L. & Kaizoji, T. (2017) Volatility Analysis of Bitcoin Price Time Series. *Quantiative Finance and Economics, 2017, 1*(4), 474-485, doi: 10.3934/QFE.2017.4.474

Jaquart P. & Dann D. & Weinhardt C. (2021) Short-term bitcoin market prediction via machine learning. *The Journal of Finance and Data Science, 2021*, *7*, 45-66, doi: 10.1016/j.jfds.2021.03.001

Bergsli L. & Lind A. & Molnár, P. & Polasic M. (2022) Forecasting volatility of Bitcoin. *Research in International Business and Finance, 2022, 59*, 101540, doi: 10.1016/j.ribaf.2021.101540

Zhang Y. & He M. & Wen D. & Wang Y. (2022) Forecasting Bitcoin volatility: A new insight from the threshold regression model. *Journal of Forecasting, 2022, 41*(3), 633-652, doi: 10.1002/for.2822

Shen Z. & Wan Q. & Leatham D. (2021) Bitcoin Return Volatility Forecasting: A Comparative Study between GARCH and RNN. *Journal of Risk and Financial Management, 2021, 14*(7), 337; doi: 10.3390/jrfm14070337

Sebastiao H. & Godinho P. (2021) Forecasting and trading cryptocurrencies with machine learning under changing market conditions. *Financial Innovation, 2021, 7,* doi: 10.1186/S40854-020-00217-X

Guo T. & Bifet A. & Antulov-Fantulin N. (2018) Bitcoin Volatility Forecasting with a Glimpse into Buy and Sell Orders. *2018 IEEE International Conference on Data Mining (ICDM)*, *17-20 November 2018* (pp. 989-994). Singapore: IEEE.

Cocco L. & Tonelli R. & Marchesi M. (2021) Predictions of bitcoin prices through machine learning based frameworks. *PeerJ Computer Science, 2021, 7,* e413, doi: 10.7717/peerj-cs.413

Wikipedia, 'Machine Learning' (2022). Ανακτήθηκε 7 Απριλίου 2022, από την ιστοσελίδα: https://en.wikipedia.org/wiki/Machine_learning

Arora S., 'Data Mining Vs. Machine Learning: The Key Difference' (2022). Ανακτήθηκε 8 Απριλίου 2022, από την ιστοσελίδα: https://www.simplilearn.com/data-mining-vs-machine-learning-article

Wolfewicz A., 'Deep Learning vs. machine learning – What's the difference?' (2021). Ανακτήθηκε 8 Απριλίου 2022, από την ιστοσελίδα: https://levity.ai/blog/difference-machine-learning-deep-learning

GeeksforGeeks, 'Best Python Libraries for Machine Learning' (2022). Ανακτήθηκε 9 Απριλίου 2022, από την ιστοσελίδα: https://www.geeksforgeeks.org/best-python-libraries-for-machine-learning/

Bhadwal A., '15 Best Machine Learning Libraries You Should Know in 2022' (2022). Ανακτήθηκε 9 Απριλίου 2022, από την ιστοσελίδα: https://hackr.io/blog/best-machine-learning-libraries

Wikipedia, 'Κατηγοριοποίηση' (2020). Ανακτήθηκε 10 Απριλίου 2022, από την ιστοσελίδα: https://el.wikipedia.org/wiki/%CE%9A%CE%B1%CF%84%CE%B7%CE%B3%CE%BF%CF%81%CE%B9%CE%BF%CF%80%CE%BF%CE%AF%CE%B7%CF%83%CE%B7

Wikipedia, 'k-nearest neighbors algorithm' (2022). Ανακτήθηκε 10 Απριλίου 2022, από την ιστοσελίδα: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

IBM, 'What is the k-nearest neighbors algorithm' (χ.χ.). Ανακτήθηκε 10 Απριλίου 2022, από την ιστοσελίδα: https://www.ibm.com/topics/knn

Fix E. & Hodges J. (1951) Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. *USAF School of Aviation Medicine, Randolph Field, Texas*. (PDF Report : https://apps.dtic.mil/dtic/tr/fulltext/u2/a800276.pdf)

Cover T. & Hart P. (1967) Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory. 13*(1): 21-27. (PDF Report: https://isl.stanford.edu/~cover/papers/transIT/0021cove.pdf)

Adeniyi D.A. & Wei Z. & Yongquan Y. (2016) Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method. *Applied Computer and Informatics, 2016, 12*(1), 90-108, doi: 10.1016/j.aci.2014.10.001

Wang Y. & Wang R. & Li D. & Adu-Gyamfi D. & Tian K. & Zhu Y. (2019) Improved Handwritten Digit Recognition using Quantum K-Nearest Neighbor Algorithm.

International Journal of Theoretical Physics, 2019, *58*, 2331-2340, doi: 10.1007/s10773-019-04124-5

Mukid M.A. & Widiharih T. & Rusgiyono A. Prahutama A. (2018) Credit Scoring analysis using weighted k nearest neighbor. *Journal of Physics: Conference Series*, 1025, 012114, doi:10.1088/1742-6596/1025/1/012114

Nie C. & Song F. (2018) Analyzing the stock market based on the structure of kNN network. *Chaos, Solitons & Fractals, 2018, 113*, 148-159, doi: 10.1016/j.chaos.2018.05.018

TIBCO, 'What is a Random Forest?' (χ.χ.). Ανακτήθηκε 11 Απριλίου 2022, από την ιστοσελίδα: https://www.tibco.com/reference-center/what-is-a-random-forest

IBM, 'Random Forest' (2020). Ανακτήθηκε 11 Απριλίου 2022, από την ιστοσελίδα: https://www.ibm.com/cloud/learn/random-forest

Ho T.K. (1995) Random Decision Forests. *Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995, pp. 278–282.* (PDF Report : https://web.archive.org/web/20160417030218/http://ect.bell-labs.com/who/tkh/publications/papers/odt.pdf)

Breiman L. (2001) Random Forests. *Machine Learning*, 45, 5-32 (2001)

Baba B. & Sevil G. (2020) Predicting IPO initial returns using random forest. *Borsa Istanbul Review, 2020, 20*(1), 13-23, doi: 10.1016/j.bir.2019.08.001

Tan Z. & Yan Z. & Zhu G. (2019) Stock selection with random forest: An exploitation of excess return in the Chinese stock market. *Heliyon, 2019, 5*(8), e02310, doi: 10.1016/j.heliyon.2019.e02310

Khalilia M. & Chakraborty S. & Popescu M. (2011) Predicting disease risks from highly imbalanced data using random forest. *BMC Medical Informatics and Decision Making, 2011, 11*, 51, doi: 10.1186/1472-6947-11-51

Moorthy K. & Mohamad M.S. (2011) Random forest for gene selection and microarray data classification. *Bioinformation, 2011, 7*, 3, 142-146, doi: 10.6026/97320630007142

IBM, 'Neural Networks' (2020). Ανακτήθηκε 12 Απριλίου 2022, από την ιστοσελίδα: https://www.ibm.com/cloud/learn/neural-networks

Master's in Data Science, 'What Is a Neural Network?' (χ.χ.). Ανακτήθηκε 12 Απριλίου 2022, από την ιστοσελίδα: https://www.mastersindatascience.org/learning/what-is-a-neural-network/

McCulloch W.S. & Pitts W. (1943) A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY. *Bulletin of Mathematical Biophysics, 5*, 115-133, (PDF Report: https://home.csulb.edu/~cwallis/382/readings/482/mccolloch.logical.calculus.ideas.1943.pdf)

Rosenblatt F. (1958) THE PERCEPTRON: A PROBABLISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN. *Psychological Review, 65*, 6, 386-408 (PDF Report: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.335.3398&rep=rep1&type=pdf)

Pang X. & Zhou Y. & Wang P. & Lin W. & Chang V. (2018) An innovative neural network approach for stock market prediction. *The Journal of Supercomputing, 2020, 76*, 2098-2118, doi: 10.1007/s11227-017-2228-y

Cho D. & Tai Y. & Kweon I. S. (2018) Deep Convolutional Neural Network for Natural Image Matting Using Initial Alpha Mattes. *IEEE Transactions on Image Processing, 2019, 28*(3), 1054-1067, doi: 10.1109/TIP.2018.2872925

Joseph S. & Sowmiya R. & Thomas R. A. & Sofia X. (2014) Face detection through neural network. *Second International Conference on Current Trends In Engineering and Technology - ICCTET 2014, 8-8 July 2014.* Coimbatore, India: IEEE, doi: 10.1109/ICCTET.2014.6966281

Janghel R.R. & Shukla A. & Tiwari R. & Kala R. (2010) Breast cancer diagnosis using Artificial Neural Network models. *The 3rd International Conference on Information Sciences and Interaction Sciences, 23-25 June 2010,* Chengdu, China: IEEE, doi: 10.1109/ICICIS.2010.5534716

He W. & Yan Z. & Sun Y. & Ou Y. & Sun C. (2018) Neural-Learning-Based Control for a Constrained Robotic Manipulator With Flexible Joints. *IEEE Transactions on Neural Networks and Learning Systems, 2018, 29*(12), 5993-6003, doi: 10.1109/TNNLS.2018.2803167

TIBCO, 'What is Logistic Regression?' (χ.χ.) Ανακτήθηκε 21 Απριλίου 2022, από την ιστοσελίδα: https://www.tibco.com/reference-center/what-is-logistic-regression

Simplilearn, 'An Introduction to Logistic Regression in Python' (2021) Ανακτήθηκε 21 Απριλίου 2022, από την ιστοσελίδα: https://www.simplilearn.com/tutorials/machine-learning-tutorial/logistic-regression-in-python?source=sl_frs_nav_playlist_video_clicked

Zabor E.C. & Reddy C.A. & Tendulkar R.D. & Patil S. (2022) Logistic Regression in Clinical Studies. *International Journal of Radiation Oncology*Biology*Physics, 2022, 112*(2), 271-277, doi: 10.1016/j.ijrobp.2021.08.007

Bhandari S. & Johnson-Syder A.J. (2018) A Generic Model Of Predicting Probability Of Success-Distress Of An Organization: A Logistic Regression Analysis. *Journal of Applied Business Research (JABR), 2018, 34*(1), 169-182, doi: 10.19030/jabr.v34i1.10107

Rusch T. & Lee I. & Hornik K. & Jank W. & Zeileis A. (2013) Influencing elections with statistics: Targeting voters with logistic regression trees. *Annals of Applied Statistics, 2013, 7*(3), 1612-1639, doi: 10.1214/13-AOAS648

Παράρτημα Α: Κώδικας Επεξεργασίας Δεδομένων

```
import pandas as pd

import numpy as np

import sklearn as sl

from functools import reduce



df1 = pd.read_csv('btc-usd.csv')


df2 = pd.read_csv('miners-revenue.csv')


df3 = pd.read_csv('transaction-fees.csv')


df4 = pd.read_csv('n-transactions.csv')


df5 = pd.read_csv('n-transactions-total.csv')


df6 = pd.read_csv('cost-per-transaction.csv')


df7 = pd.read_csv('n-unique-addresses.csv')


df8 = pd.read_csv('estimated-transaction-volume.csv')


df9 = pd.read_csv('estimated-transaction-volume-usd.csv')


# Picking the same time period for all the data

# we are using


data1= df1.loc[(df1['Date'] >= '2019-05-10')

                    & (df1['Date'] < '2022-05-01')]
```

```
data2= df2.loc[(df2['Timestamp'] >= '2019-05-10')
                    & (df2['Timestamp'] < '2022-05-01')]

data3= df3.loc[(df3['Timestamp'] >= '2019-05-10')
                    & (df3['Timestamp'] < '2022-05-01')]

data4= df4.loc[(df4['Timestamp'] >= '2019-05-10')
                    & (df4['Timestamp'] < '2022-05-01')]

data5= df5.loc[(df5['Timestamp'] >= '2019-05-10')
                    & (df5['Timestamp'] < '2022-05-01')]

data6= df6.loc[(df6['Timestamp'] >= '2019-05-10')
                    & (df6['Timestamp'] < '2022-05-01')]

data7= df7.loc[(df7['Timestamp'] >= '2019-05-10')
                    & (df7['Timestamp'] < '2022-05-01')]

data8= df8.loc[(df8['Timestamp'] >= '2019-05-10')
                    & (df8['Timestamp'] < '2022-05-01')]

data9= df9.loc[(df9['Timestamp'] >= '2019-05-10')
                    & (df9['Timestamp'] < '2022-05-01')]



# This part of code was used to find the missing dates
# on each dataframe we created from the csv files.
# We found 3 missing dates on df7,
# 3 dates on df8 and the same 3 from df8 on df9

df5['Timestamp'] = pd.to_datetime(df5['Timestamp']).dt.normalize()
```

```
dftest = df5.set_index('Timestamp')
dftest.index = pd.to_datetime(dftest.index)
print(pd.date_range(
  start="2019-05-10", end="2022-04-30").difference(dftest.index))



# Matching the data by 'Timestamp', so we can merge them and
# get them in one dataframe

data_frames=[data2, data3, data4, data5, data6, data7, data8, data9]

d1 = reduce(lambda  left,right: pd.merge(left,right,on=['Timestamp'],
                                         how='outer'), data_frames)



data1.reset_index(drop=True, inplace=True)

d1



# We need to drop the columns reffering to date
# because its not needed on the final dataframe



final_data1= data1.drop(['Date'], axis = 1)

final_d1= d1.drop(['Timestamp'], axis= 1)



final_df= pd.concat([final_data1, final_d1], axis = 1)
```

```
# We need to check for Na values inside our final dataframe and if we
# find any we must decide how we are going to proceed


final_df.isna().sum().sum()


# We will fill the Na values on the columns that were found
# with the mean of the specific column


mean7= data7['n-unique-addresses'].mean()
mean8= data8['estimated-transaction-volume'].mean()
mean9= data9['estimated-transaction-volume-usd'].mean()

final_df['n-unique-addresses'].fillna(mean7, inplace=True)
final_df['estimated-transaction-volume'].fillna(mean8, inplace=True)
final_df['estimated-transaction-volume-usd'].fillna(mean9, inplace=True)

final_df.isna().sum().sum()



#Converting the final dataframe to a csv file


final_df.to_csv('alldatafinal.csv',index=False)
```

Παράρτημα Β: Κύριος κώδικας ΔΕ

```
import pandas as pd
import numpy as np
import sklearn as sl

df = pd.read_csv('alldatafinal.csv')


print(df)

# We need to sort the dataframe, so it will be easier
# to get some calculations done and get 1 new column

df.sort_index(ascending = False, inplace = True)

# We need to see whether we have an increased price
# or a decreased price each day, with using the closing prices



df['Daily Difference'] = (df.Close - df.Close.shift(-1))
print(df)

inc_dec =[]

for value in df['Daily Difference']:
    if value > 0 :
        inc_dec.append("Increased Price")
    elif value < 0 :
        inc_dec.append("Decreased Price")
    elif value == 0 :
        inc_dec.append("Same Price")
    else:
        inc_dec.append("Invalid")

df["Price Status"] = inc_dec

print(df)
```

```
# We need to remove the final line becuase
# it was used as a small "step" to calculate
# the daily difference on the price


df.drop(index= df.index[-1], axis= 0, inplace= True)

print(df)

# Index must be reseted because of the line removal
# we performed above

df.reset_index(drop=True, inplace=True)
print(df)

print(df['Price Status'].unique())

df['Price Status'].value_counts()


#    Encoding categorical values as ordinal numbers
#    We do here an encoding of categorical values, but with ORDINAL values,
#    i.e. 1 for increase, -1 for decrease and 0 for no change

df.loc[ df[ 'Price Status'] == 'Increased Price', 'Price Status'] = '1'
df.loc[ df[ 'Price Status'] == 'Decreased Price', 'Price Status'] = '-1'
df.loc[ df[ 'Price Status'] == 'Same Price', 'Price Status'] = '0'

# Change datatype of column Price Status from string to integer.
df = df.astype({'Price Status':'int'})


print(df)
```

```
# We need to drop the column
# "Daily Difference" because it was used to create
# the target column "Price Status", where
# we base our classification.

df.drop(df.columns[[14]], axis = 1, inplace = True)



# Values in our dataframe are on a different scale
# so we need to use a scaler to get the values "closer"
# with each other




from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()



scaler.fit(df.drop(['Price Status'], axis= 1))

scaled_features = scaler.transform(df.drop(['Price Status'], axis= 1))




final_features = pd.DataFrame(scaled_features,index = None, columns =
df.columns[:-1])



final_features.head()

final_features.describe()



# Splitting the dataset in order to perform
# train with the data and then test them



from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(scaled_features,
df['Price Status'],
                                                    test_size = 0.30,
random_state= 2)



df['Price Status'].value_counts()



# 1st algorithm to use is KNN


from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors = 33)

knn.fit(X_train, y_train)

pred_train = knn.predict(X_train)
pred_test = knn.predict(X_test)




# Confusion matrix and report for the train set using Knn

from sklearn.metrics import classification_report, confusion_matrix

print("Confusion Matrix and Classification Report of train set in Knn")
print()

print(confusion_matrix(y_train,pred_train))

print(classification_report(y_train,pred_train))
```

```
# Confusion matrix and report for the test set using Knn

from sklearn.metrics import classification_report, confusion_matrix

print("Confusion Matrix and Classification Report of test set in Knn")
print()
print(confusion_matrix(y_test, pred_test))

print(classification_report(y_test, pred_test))


# 2nd algorithm to use is RandomForest


from sklearn.ensemble import RandomForestClassifier

rfc = RandomForestClassifier(n_estimators= 145, random_state= 2)

rfc.fit(X_train, y_train)

rf_pred_train= rfc.predict(X_train)
rf_pred_test= rfc.predict(X_test)


# Confusion matrix and report for the train set using Random Forest

from sklearn.metrics import classification_report, confusion_matrix
print("Confusion Matrix and Classification Report of train set in Random
Forest")
print()

print(confusion_matrix(y_train, rf_pred_train))

print(classification_report(y_train, rf_pred_train))


# Confusion matrix and report for the test set using Random Forest
```

```
from sklearn.metrics import classification_report, confusion_matrix
print("Confusion Matrix and Classification Report of test set in Random Forest")
print()

print(confusion_matrix(y_test, rf_pred_test))

print(classification_report(y_test, rf_pred_test))



# 3rd algorithm is neural netowrks
# This is one way to do it




from sklearn.neural_network import MLPClassifier


mlp = MLPClassifier(hidden_layer_sizes=(14), activation= 'relu',
                    max_iter=10000, random_state= 2)

mlp.fit(X_train,y_train)


predict_train = mlp.predict(X_train)

predict_test = mlp.predict(X_test)




# Confusion matrix and report for the train set
# using MLP Neural Network

print("Confusion Matrix and Classification Report of train set in Neural Network")
print()

print(confusion_matrix(y_train,predict_train))
```

```
print(classification_report(y_train,predict_train))



# Confusion matrix and report for the test set
# using MLP Neural Network

print("Confusion Matrix and Classification Report of test set in Neural
Network")
print()
print(confusion_matrix(y_test,predict_test))

print(classification_report(y_test,predict_test))



# 4rth algorithm we will be using is Logistic Regression

from sklearn.linear_model import LogisticRegression
lr= LogisticRegression()

lr.fit(X_train, y_train)

lr_pred_train = lr.predict(X_train)
lr_pred_test = lr.predict(X_test)

# Confusion matrix and report for the train set
# using Logistic Regression

from sklearn.metrics import classification_report, confusion_matrix

print("Confusion Matrix and Classification Report of train set in
Logistic Regression")
print()

print(confusion_matrix(y_train,lr_pred_train))

print(classification_report(y_train, lr_pred_train))
```

```
# Confusion matrix and report for the test set
# using Logistic Regression

from sklearn.metrics import classification_report, confusion_matrix

print("Confusion Matrix and Classification Report of test set in
Logistic Regression")
print()

print(confusion_matrix(y_test,lr_pred_test))

print(classification_report(y_test, lr_pred_test))




# Creating a function so we can perform
# k-fold-cross validation to evaluate
# the algorithms/models used


def get_score(model, X_train, X_test, y_train, y_test):
    model.fit(X_train, y_train)
    return model.score(X_test, y_test)


from sklearn.model_selection import StratifiedKFold
folds = StratifiedKFold(n_splits= 10, shuffle= True, random_state= 2)


# Creating empty lists, so we can save
# the accuracy score of each algorithm
# after the splits

scores_knn= []
scores_rfc= []
scores_lr= []
scores_mlp= []
for train_index, test_index in folds.split(scaled_features, df['Price
Status']):
```

```
      print("Train:", train_index)
      print("Validation:", test_index)
      X_train, X_test = scaled_features[train_index],
scaled_features[test_index]
      y_train, y_test = df['Price Status'][train_index], df['Price
Status'][test_index]
      scores_knn.append(get_score(KNeighborsClassifier(n_neighbors =
33), X_train, X_test, y_train, y_test))
      scores_rfc.append(get_score(RandomForestClassifier(n_estimators=
145, random_state= 2),
                                  X_train, X_test, y_train, y_test))
      scores_lr.append(get_score(LogisticRegression(), X_train, X_test,
y_train, y_test))
      scores_mlp.append(get_score(MLPClassifier(hidden_layer_sizes=(14),
activation= 'relu',
                                                max_iter=10000,
random_state= 2), X_train, X_test, y_train, y_test))



print("Knn scores: ")
scores_knn


print()
print("The average score for Knn is ", np.mean(scores_knn))


print("Random Forest scores: ")
scores_rfc


print()
print("The average score for Random Forest is ", np.mean(scores_rfc))


print("Logistic Regression scores: ")
scores_lr


print()
print("The average score for Logistic Regression is ",
np.mean(scores_lr))


print("MLP Neural Network scores: ")
scores_mlp


print()
```

```
print("The average score for MLP Neural Network is ",
np.mean(scores_mlp))
```

Υπεύθυνη Δήλωση Συγγραφέα:

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν.1599/1986, η παρούσα εργασία αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης.