



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Πληροφοριακό Σύστημα για οικονομικά
δεδομένα βασισμένο σε Αλγορίθμους Μηχανικής
Μάθησης**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

των

ΒΟΥΔΟΥΡΗΣ ΑΓΓΕΛΟΣ – ΜΙΧΑΗΛ (ΑΕΜ: 2569)

&

ΚΑΡΓΟΣ ΡΑΦΑΗΛ (ΑΕΜ: 2907)

Επιβλέπων : Νικόλαος Δημόκας
Επίκουρος Καθηγητής

Καστοριά Φεβρουάριος - 2023

Η παρούσα σελίδα σκοπίμως παραμένει λευκή



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Πληροφοριακό Σύστημα για οικονομικά
δεδομένα βασισμένο σε Αλγορίθμους Μηχανικής
Μάθησης**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

των

ΒΟΥΔΟΥΡΗΣ ΑΓΓΕΛΟΣ – ΜΙΧΑΗΛ (ΑΕΜ: 2569)

&

ΚΑΡΓΟΣ ΡΑΦΑΗΛ (ΑΕΜ: 2907)

**Επιβλέπων : Νικόλαος Δημόκας
Επίκουρος Καθηγητής**

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 03/02/2023

.....
Νικόλαος Δημόκας
Επίκουρος Καθηγητής

.....
Δημήτριος Βέργαδος
Επίκουρος Καθηγητής

.....
Σπυρίδων Νικολάου
Λέκτορας

Καστοριά Φεβρουάριος - 2023

Copyright © 2023 – ΒΟΥΔΟΥΡΗΣ ΑΓΓΕΛΟΣ – ΜΙΧΑΗΛ, ΚΑΡΓΟΣ ΡΑΦΑΗΛ

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Δυτικής Μακεδονίας.

Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.

Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε τον επιβλέποντα καθηγητή, κύριο Νικόλαο Δημόκα για την πολύτιμη καθοδήγησή του. Επιπρόσθετα, οφείλουμε να αφιερώσουμε την πτυχιακή μας εργασία στους γονείς μας που μας συμπαραστάθηκαν όλα τα χρόνια της φοίτησης μας στο Τμήμα Πληροφορικής του Πανεπιστημίου Δυτικής Μακεδονίας. Τέλος θα θέλαμε εκφράσουμε ιδιαίτερη ευγνωμοσύνη στον κύριο Κυριάκο Νεοκλέους για τις γνώσεις που πρόσφερε για την δημιουργία των μοντέλων – προβλέψεων.

Περίληψη

Η παρούσα εργασία έχει σαν θέμα ένα Πληροφοριακό Σύστημα για οικονομικά δεδομένα βασισμένο σε Αλγορίθμους Μηχανικής Μάθησης. Βασικός σκοπός της εργασίας αυτής είναι να αναδείξει τη σπουδαιότητα της Μηχανικής Μάθησης, που χρησιμοποιείται ο κάθε κλάδος της Μηχανικής Μάθησης και εμβαθύνει κυρίως στην Μάθηση με Επίβλεψη. Παράλληλα, αναλύονται τα προβλήματα/μειονεκτήματα που παρουσιάζονται στην επιστημονική περιοχή. Τέλος, παρουσιάζεται η εφαρμογή και λειτουργία ενός μεγάλου συνόλου αλγορίθμων σχετικά με το πρόβλημα πρόβλεψης οικονομικών δεδομένων και ποιο συγκεκριμένα της τιμής και τάσης των κρυπτονομισμάτων.

Λέξεις Κλειδιά: Μηχανική Μάθηση, Πρόβλεψη, Στόχος, Συνάρτηση κόστους, Νευρωνικά Δίκτυα, Τεχνητά Νευρωνικά Δίκτυα, Συνελικτικά Νευρωνικά Δίκτυα, Αναδρομικά Νευρωνικά Δίκτυα, Νευρωνικά Δίκτυα Bayesian, Υπερπροσαρμογή, Υποπροσαρμογή, Ταξινόμηση, Παλινδρόμηση, Κρυπτονομίσματα, Python-3, Pandas, Matplotlib, NumPy, Flask, HTML, CSS, JavaScript

Abstract

The subject of the present Thesis is the presentation of an Information System analyzing economic data, based on the algorithms of Machine Learning.

The main idea of this project is not only to show that Machine Learning is a method in which all the aspects of it are used but also to delve deeply into the Supervised Learning and the problems that have occurred recently. It also explains the function of different algorithms and how they work in regard with the problems that have occurred in the field of Supervised Learning.

Key Words: Machine Learning, Forecast, Target, Cost Function, Neural Networks, Convolutional Neural Networks, Recurrent Neural Networks, Bayesian Neural Networks, Overfitting, Underfitting, Classification, Regression, Cryptocurrencies, Python-3, Pandas, Matplotlib, NumPy, Flask, HTML, CSS, JavaScript

Πίνακας Περιεχομένων

Εισαγωγή	1
1. Μηχανική Μάθηση	2
1.1 Τύποι Μηχανικής Μάθησης	2
1.1.1 Μάθηση με Επίβλεψη	2
1.1.2 Μάθηση χωρίς Επίβλεψη	3
1.1.3 Μάθηση με Ενίσχυση	4
1.2 Κριτήρια Επίδοσης	4
2. Μέθοδοι Μάθησης και Πρόβλεψης	9
2.1 Μοντέλα Προβλέψεων Μάθησης	9
2.1.1 Linear Regression	9
2.1.2 Regularized Linear Models	11
2.1.3 Δέντρα	14
2.2 Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Network)	17
2.2.1 Βιολογικοί Νευρώνες	17
2.2.2 Ιστορική Αναδρομή	17
2.2.3 Χρήση των Τεχνητών Νευρωνικών Δικτύων	18
2.2.4 Νευρωνικό Δίκτυο Perceptron	19
2.3 Συνελκτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks)	21
2.3.1 Αρχιτεκτονική δομή CNN	21
2.4 Αναδρομικά Νευρωνικά Δίκτυα (Recurrent Neural Networks)	23
2.4.1 Εκπαίδευση Αναδρομικών Νευρωνικών Δικτύων	25
2.4.2 Μακρά Δίκτυα Βραχείας Μνήμης (Long Sort-Term Memory Networks)	25
2.5 Νευρωνικά Δίκτυα Bayesian	26
2.5.1 Θεώρημα Bayes	27
2.5.2 Ιστορική Αναδρομή δικτύων Bayesian	27
2.5.3 Αβεβαιότητα	27
2.5.4 Πλεονεκτήματα Νευρωνικών Δικτύων Bayes	28
2.5.5 Εφαρμογές στα Bayesian Networks	28
3. Τεχνολογίες και Δεδομένα	29
3.1 Χρήση και τεχνολογίες της Γλώσσας Προγραμματισμού	29
3.2 Περιγραφή back-end API συστήματος εφαρμογής	32

3.2.1 Βασική Λειτουργία Back – End Συστήματος	33
3.3 Εγχειρίδιο χρήσης μοντέλων	34
3.4 Περιγραφή front-end Web συστήματος.....	37
3.4.1 Το Front-End διαχρονικά.....	38
3.4.2 Εργαλεία και τεχνολογίες που χρησιμοποιούνται στην ανάπτυξη Front-End ..	38
3.4.3 Αρχές σχεδίασης και βέλτιστες πρακτικές.....	40
3.4.4 Συμπέρασμα.....	40
3.5 Δεδομένα	41
4. Διαδικτυακή Εφαρμογή	42
5. Αξιολόγηση	46
Συμπεράσματα	52
Βιβλιογραφία.....	55

Λίστα Εικόνων

Εικόνα 1: Η απεικόνιση του Linear Regression δύο χαρακτηριστικών (Πηγή: Géron, 2019) .	10
Εικόνα 2: Η μορφή ενός δέντρου αποφάσεων (Πηγή: Battiti and Brunato, 2013)	15
Εικόνα 3: Η δομή των Random Forest κατά την διάρκεια της εκπαίδευσής τους.....	16
Εικόνα 4: Η μορφή ενός βιολογικού νευρώνα (Πηγή: Géron, 2019).....	17
Εικόνα 5: Απεικόνιση νευρωνικού δικτύου Perceptron που περιέχει μόνο έναν νευρώνα ..	20
Εικόνα 6: Απεικόνιση πολυστρωματικού νευρωνικού δικτύου Perceptron που περιέχει πολλά στρώματα εισόδου, ένα στρώμα εξόδου και ενδιάμεσα κρυφά στρώματα	20
Εικόνα 7: Ένα βαθύ νευρωνικό δίκτυο CNN με ένα κρυφό στρώμα	21
Εικόνα 8: Βασική αρχιτεκτονική δομή CNN	22
Εικόνα 9: Αναπαράσταση ενός απλού Αναδρομικού Νευρωνικού Δικτύου όπου ανατροφοδοτείται ο κόμβος.....	24
Εικόνα 10: Η αρχιτεκτονική δομή μίας μονάδας cell τεχνητού νευρωνικού δικτύου LSTM ..	26
Εικόνα 11: Εξαγωγή δεδομένων.....	29
Εικόνα 12: Επιλογή στόχου	30
Εικόνα 13: Διαχωρισμός πίνακα δεδομένων	30
Εικόνα 14: Επιλογή μοντέλου	31
Εικόνα 15: Δομή API	33
Εικόνα 16: Σετ Δεδομένων	41
Εικόνα 17: Αρχική Σελίδα εφαρμογής.....	42
Εικόνα 18: Σελίδα Εισόδου στο σύστημα για εγγεγραμμένους	43
Εικόνα 19: Σελίδα Αξιολόγησης και επιλογής κρυπτονομίσματος και μοντέλου.....	44
Εικόνα 20: Σελίδα Μοντέλου και τα αποτελέσματά του	44
Εικόνα 21: Σελίδα πρόβλεψης και τα αποτελέσματα του για τα κρυπτονομίσματα	45
Εικόνα 22: Σελίδα Σχετικά με πληροφορίες σχετικά με τους δημιουργούς της εφαρμογής .	45
Εικόνα 23: Σχεδιαγράμματα αποτελεσμάτων Linear Regression	47
Εικόνα 24: Σχεδιαγράμματα αποτελεσμάτων Decision Tree.....	48

Εικόνα 25: Σχεδιαγράμματα αποτελεσμάτων Random Forest Regressor	49
Εικόνα 26: Σχεδιαγράμματα αποτελεσμάτων Gradient Boosting Regressor	50
Εικόνα 27: Σχεδιαγράμματα αποτελεσμάτων LSTM.....	51
Εικόνα 28: Η πρόβλεψη της τάσης του κρυπτονομίσματος Bitcoin	51

Λίστα Πινάκων

Πίνακας 1: Παράδειγμα παραπλανητικής ακρίβειας	5
Πίνακας 2: Παράδειγμα λειτουργίας τύπου ευστοχίας.....	6
Πίνακας 3: Παράδειγμα λειτουργίας τύπου ανάκλησης	6
Πίνακας 4: Παράδειγμα λειτουργίας του τύπου F1-Score / F-measure	7
Πίνακας 5: Παράδειγμα αναπαράστασης φόρμουλας Linear Regression	11
Πίνακας 6: Λειτουργία συνδυασμών των δύο μοντέλων	14
Πίνακας 7: Χαρακτηριστικά υπολογιστή που έτρεξε η εφαρμογή	46
Πίνακας 8: Χαρακτηριστικά της εξαγωγής των δεδομένων	47

Λίστα Εξιτώσεων

Εξίσωση 1: Εξίσωση ακρίβειας (accuracy)	5
Εξίσωση 2: Παράδειγμα εξίσωσης ακρίβειας	5
Εξίσωση 3: Μαθηματικός Τύπος Ευστοχίας	6
Εξίσωση 4: Παράδειγμα του τύπου ευστοχίας	6
Εξίσωση 5: Ο μαθηματικός τύπος της ανάκλησης (Recall)	6
Εξίσωση 6: Παράδειγμα τύπου ανάκλησης	7
Εξίσωση 7: Ο μαθηματικός τύπος F1-Score / F-measure	7
Εξίσωση 8: Παράδειγμα μαθηματικού τύπου F1-Score / F-measure	7
Εξίσωση 9: Ο μαθηματικός τύπος Μέσου Τετραγωνικού Σφάλματος (MSE)	8
Εξίσωση 10: Εξίσωση μοντέλου Linear Regression	10
Εξίσωση 11: Τύπος συνάρτησης κόστους Mean Square Error(MSE)	11
Εξίσωση 12: Τύπος συνάρτησης κόστους Ridge Regression	12
Εξίσωση 13: Τύπος συνάρτησης κόστους Lasso Regression	13
Εξίσωση 14: Τύπος συνάρτησης κόστους Elastic Net	13

Εισαγωγή

Η ακόλουθη εργασία έχει σαν θέμα ένα Πληροφοριακό Σύστημα για οικονομικά δεδομένα βασισμένο σε Αλγορίθμους Μηχανικής Μάθησης. Σκοπός αυτού του θέματος είναι να δείξει πως λειτουργούν και συμπεριφέρονται οι αλγόριθμοι Μηχανικής Μάθησης σε οικονομικά δεδομένα. Επιπλέον γίνεται και η πρακτική εφαρμογή επί του θέματος με την δημιουργία μίας εφαρμογής όπου θα δίνεται η δυνατότητα να αναπαρίστανται πραγματικά δεδομένα και πραγματικές αναλύσεις των αλγορίθμων.

Η πτυχιακή εργασία χωρίζεται σε πέντε κεφάλαια. Στο πρώτο κεφάλαιο γίνεται μία αναφορά για την Μηχανική Μάθηση, από ποιους τύπους αποτελείται, ποια είναι τα προβλήματα και ποια είναι τα κριτήρια επίδοσης. Στο δεύτερο κεφάλαιο αναφέρονται ένα σύνολο από αλγορίθμους Μηχανικής Μάθησης, ο τρόπος λειτουργίας τους και ποια είναι τα πλεονεκτήματα και τα μειονεκτήματα του κάθε αλγόριθμου. Στο τρίτο και τελευταίο κεφάλαιο γίνεται η ανάλυση και η λειτουργία της πρακτικής εφαρμογής όπου περιγράφονται οι διασυνδέσεις των τεχνολογιών που χρησιμοποιούνται μεταξύ του front-end συστήματος και του back-end. Στην συνέχεια στο τέταρτο κεφάλαιο γίνεται γενική περιγραφή του πληροφοριακού συστήματος που υλοποιήθηκε στα πλαίσια της πτυχιακής εργασίας και τέλος στο πέμπτο κεφάλαιο γίνεται η αξιολόγηση των μοντέλων μηχανικής μάθησης που χρησιμοποιήθηκαν στο πληροφοριακό σύστημα, ποιοι είναι ποιο αποδοτικοί, τι δεδομένα χρησιμοποιήκαν για την αξιολόγηση καθώς και το σύστημα στο οποίο δημιουργήθηκε η διαδικτυακή εφαρμογή.

1. Μηχανική Μάθηση

Σε αυτό το κεφάλαιο θα αναφερθούν οι τύποι μηχανικής μάθησης στην τεχνητή νοημοσύνη που υπάρχουν, τι προβλήματα μπορεί να αντιμετωπίσει η Μηχανική Μάθηση, ποια είναι τα κριτήρια επίδοσης και πως μπορούν να χρησιμοποιηθούν.

1.1 Τύποι Μηχανικής Μάθησης

Στο συγκεκριμένο υποκεφάλαιο θα αναλυθούν οι τύποι μηχανικής μάθησης καθώς και ο τρόπος λειτουργίας τους. Με την πάροδο του χρόνου ο άνθρωπος και η τεχνολογία εξελίσσονται και επίσης οι απαιτήσεις αυξάνονται. Αυτό είχε σαν αποτέλεσμα να κάνει τους επιστήμονες να αναζητήσουν νέες διόδους που θα έκαναν την καθημερινή ζωή των ανθρώπων πιο εύκολη. Κάπως έτσι μπήκε η Τεχνητή νοημοσύνη και ο αυτοματισμός γενικότερα στην ζωή των ανθρώπων με τους επιστήμονες να αναπτύσσουν νέες τεχνικές που θα επέλυαν διάφορα ζητήματα και προβλήματα που αφορούν την παγκόσμια κοινότητα. Οι βασικότερες τεχνικές μηχανικής μάθησης που αναπτύχθηκαν είναι η Μάθηση με Επίβλεψη (**supervised learning**), Μάθηση χωρίς Επίβλεψη (**unsupervised learning**), και τέλος η Μάθηση με ενίσχυση (**reinforcement learning**) [1]. Η μηχανική μάθηση μπορεί να εφαρμοστεί σε ένα μεγάλο εύρος προβλημάτων όπως η ασφάλεια, χρηματοοικονομικά, αναγνώριση εικόνας και άλλα.

1.1.1 Μάθηση με Επίβλεψη

Η εποπτευόμενη μάθηση (**supervised learning**) ανήκει σε μία από τις υποκατηγορίες στον κλάδο της μηχανικής μάθησης (**machine learning**). Η εποπτευόμενη μάθηση χρησιμοποιεί αλγορίθμους οι οποίοι προσπαθούν να μάθουν – να εκπαιδευτούν μέσα από ένα σετ εκπαίδευσης το οποίο περιλαμβάνει κάποια χαρακτηριστικά (**features**) εισόδου, μαζί με τα αντίστοιχα χαρακτηριστικά που διαθέτουν τις επιθυμητές λύσεις – ετικέτες (**labels**) με σκοπό την πρόβλεψη μιας τιμής η οποία ονομάζεται συνάρτηση στόχος (**target function**). Η μάθηση με επίβλεψη χωρίζεται σε δύο μεγάλες κατηγορίες προβλημάτων. Τα προβλήματα ταξινόμησης (**classification**) και τα προβλήματα παλινδρόμησης (**regression**) [1] [2]. Ένα από τα προβλήματα που προσπαθεί ο κλάδος της Μηχανικής Μάθησης είναι πως προσπαθεί να προβλέψει καταστάσεις αν και

εφόσον έχει ξανασυμβεί κάτι αντίστοιχο στο παρελθόν. Άρα με λίγα λόγια εάν εκπαιδευτεί ένας αλγόριθμος σε ένα σετ εκπαίδευσης στο οποίο τα δεδομένα είναι ελλιπής τότε το συγκεκριμένο μοντέλο δεν θα μπορέσει να μας προβλέψει μία κατάσταση στην οποία δεν έχει εκπαιδευτεί.

Παρακάτω θα αναλυθούν οι δύο κατηγορίες της μάθησης με επίβλεψη.

- **Πρόβλημα Ταξινόμησης**

Στα προβλήματα ταξινόμησης οι μεταβλητές των στόχων (**targets**) [1] αποτελούνται από λογικές τιμές για παράδειγμα {0,1}. Για παράδειγμα ένα πρόβλημα ταξινόμησης είναι εάν ένα μήνυμα είναι spam ή όχι. Δηλαδή ένας αλγόριθμος ταξινόμησης δέχεται κάποια χαρακτηριστικά σαν είσοδο, μαθαίνει από τα συγκεκριμένα χαρακτηριστικά στα οποία υπάρχουν και οι λύσεις και στην συνέχεια μετά την εκπαίδευση ο αλγόριθμος προσπαθεί να κατηγοριοποιήσει εάν ένα μήνυμα ανήκει στην κατηγορία spam (**0**) ή όχι spam (**1**).

- **Πρόβλημα Παλινδρόμησης**

Τα προβλήματα παλινδρόμησης οι μεταβλητές των στόχων (**targets**) αποτελούνται από συνεχείς τιμές. Δηλαδή ο αλγόριθμος εκπαιδεύεται από ένα σύνολο δεδομένων και προσπαθεί να προβλέψει μία τιμή. Για παράδειγμα ποια θα είναι η τιμή ενός κρυπτονομίσματος τις επόμενες μέρες.

Σύμφωνα με την scikit-learn κάποιοι αλγόριθμοι έχουν καλύτερες επιδόσεις σε προβλήματα ταξινόμησης (**DecisionTreeClassifier**) και κάποιοι άλλοι έχουν καλύτερες επιδόσεις σε προβλήματα παλινδρόμησης (**DecisionTreeRegressor**).

1.1.2 Μάθηση χωρίς Επίβλεψη

Όπως και η μάθηση με επίβλεψη έτσι και η μάθηση χωρίς επίβλεψη (**unsupervised learning**) ανήκει σε μία από τις υποκατηγορίες στον κλάδο της μηχανικής μάθησης. Η διαφορά τους όμως είναι ότι στην μη επιβλεπόμενη μάθηση ο όγκος των δεδομένων δεν περιέχει τις ετικέτες (**labels**) [1] [2] δηλαδή τις επιθυμητές επιθυμητές λύσεις για να εκπαιδευτεί το μοντέλο. Σκοπός της μάθησης χωρίς επίβλεψη είναι να εντοπίσει τις διασυνδέσεις μεταξύ των δεδομένων ώστε να τα χωρίσει κατάλληλα σε ομάδες.

1.1.3 Μάθηση με Ενίσχυση

Όπως και οι δύο προηγούμενες υποκατηγορίες μάθησης έτσι και η μάθηση με ενίσχυση (**reinforcement learning**) [1] [2] ανήκει σε μία από αυτές. Η μάθηση με ενίσχυση έχει σαν στόχο να βγάλει εις πέρας ένα αποτέλεσμα. Στην μάθηση με ενίσχυση το μοντέλο δεν γνωρίζει την επιθυμητή έξοδο όπως συμβαίνει στην επιβλεπόμενη μάθηση αλλά το μοντέλο της μάθησης με ενίσχυση χρησιμοποιεί πολλές ενέργειες προκειμένου να φτάσει σε έναν τελικό στόχο. Η μάθηση με ενίσχυση μπορεί να χρησιμοποιηθεί σε ένα βιντεοπαιχνίδι για την επιτυχία της λήξης ενός παιχνιδιού ή την εκπαίδευση στην κίνηση ενός ρομπότ.) Ένα βασικό χαρακτηριστικό της μάθησης με ενίσχυση είναι πως μπορεί να ανακαλυφθούν νέοι μέθοδοι για κάποια κατάσταση.

1.2 Κριτήρια Επίδοσης

Στο παρόν υποκεφάλαιο θα αναλυθούν μερικά από τα βασικά κριτήρια επίδοσης που υπάρχουν στην μηχανική μάθηση. Τι είναι όμως ένα κριτήριο επίδοσης; Ένα κριτήριο επίδοσης δείχνει την ακρίβεια ενός μοντέλου δηλαδή το πόσο αξιόπιστο μπορεί να είναι. Επίσης όπως και τα μοντέλα χωρίζονται για προβλήματα ταξινόμησης (**classification**) ή προβλήματα παλινδρόμησης (**regression**), έτσι μερικά από τα κριτήρια επίδοσης χρησιμοποιείται για το ένα πρόβλημα ή το άλλο.

- Κριτήρια Επίδοσης Ταξινόμησης

Στα προβλήματα ταξινόμησης οι τιμές που μπορούν να ληφθούν για τα κριτήρια επίδοσης μπορούν να χωριστούν σε τέσσερις κατηγορίες. Πραγματικά Αρνητικά (**True Negative**) τα οποία προβλέφθηκαν αρνητικά και είναι όντως αρνητικά, τα Εσφαλμένα Αρνητικά (**False Negative**) τα οποία προβλέφθηκαν αρνητικά αλλά δεν ανήκουν σε αυτήν την κατηγορία, τα Πραγματικά Θετικά (**True Positive**) τα οποία προβλέφθηκαν θετικά και ανήκουν στην συγκεκριμένη κατηγορία και τέλος τα Εσφαλμένα Θετικά (**False Positive**) τα οποία προβλέφθηκαν θετικά αλλά δεν ανήκουν στην κατηγορία των θετικών τιμών. Με αυτά τέσσερα ξεχωριστά είδη τιμών μπορούν να αναλυθούν τα κριτήρια επίδοσης που ανήκουν στην κατηγορία της ταξινόμησης.

ο **Ακρίβεια (Accuracy):**

Σύμφωνα με το scikit-learn ο μαθηματικός τύπος της ακρίβειας είναι οι καταναμεμημένες σωστές προβλέψεις διά το πλήθος των παρατηρήσεων. Μερικές φορές όμως η ακρίβεια μπορεί να γίνει και παραπλανητική εάν το σύνολο των δεδομένων δεν είναι σωστά διαμορφωμένο. Δηλαδή εάν περιέχει πολλά δεδομένα με ετικέτες της μίας κλάσης και αρκετά λιγότερες από την άλλη. Η ακρίβεια μετράει την επίδοση ενός μοντέλου.

$$accuracy(y, \hat{y}) = \left(\frac{1}{n}\right) \sum_{i=0}^{n-1} 1(\hat{y}_i = y_i)$$

Εξίσωση 1: Εξίσωση ακρίβειας (accuracy) [3]

Πίνακας 1: Παράδειγμα παραπλανητικής ακρίβειας

ΔΕΔΟΜΕΝΑ	ΓΑΤΑ ΚΛΑΣΗ = 0	ΣΚΥΛΟΣ ΚΛΑΣΗ = 1
TN	989	
TP		1
FN	1	
FP		9

$$Accuracy = \left(\frac{989 + 1}{989 + 1 + 9 + 1}\right) = 0.99 = 99\%$$

Εξίσωση 2: Παράδειγμα εξίσωσης ακριβείας

Λόγω αυτής της παραπλάνησης ώστε να υπάρχει μία ποιο ξεκάθαρη εικόνα για τα κριτήρια επίδοσης, παρακάτω θα αναλυθεί η ευστοχία και η ανάκληση που καλύπτουν αυτά τα ζητήματα.

ο **Ευστοχία (Precision) – Ανάκληση (Recall):**

Η ευστοχία (precision) είναι ένας τύπος ο οποίος δείχνει το ποσοστό ακρίβειας των θετικών στοιχείων ενός μοντέλου που αφορά μόνο την κλάση 1. Δηλαδή η ευστοχία προβάλλει τα δείγματα που προβλέφθηκαν σωστά και ανήκουν και ανήκουν στην συγκεκριμένη κλάση. Σύμφωνα με το scikit-learn ο τύπος της ευστοχίας είναι:

$$\frac{tp}{(tp + fp)tpfp}$$

Εξίσωση 3: Μαθηματικός Τύπος Ευστοχίας [4]

Πίνακας 2: Παράδειγμα λειτουργίας τύπου ευστοχίας

ΔΕΔΟΜΕΝΑ	ΓΑΤΑ ΚΛΑΣΗ = 0	ΣΚΥΛΟΣ ΚΛΑΣΗ = 1
TN	4	
TP		3
FN	1	
FP		2

$$Precision = \frac{3}{(3 + 2)} = 0.6 = 60\%$$

Εξίσωση 4: Παράδειγμα του τύπου ευστοχίας

Που σημαίνει ότι το 60% των σκύλων ανήκει όντως στην κλάση 1.

Από την άλλη η **ανάκληση (recall)** έχει την ικανότητα να επιστρέφει το ποσοστό ακριβείας μιας κατηγορίας [1]. Σύμφωνα με την scikit-learn ο τύπος της ανάκλησης είναι:

$$\frac{tp}{(tp + fn)}$$

Εξίσωση 5: Ο μαθηματικός τύπος της ανάκλησης (Recall) [5]

Πίνακας 3: Παράδειγμα λειτουργίας τύπου ανάκλησης

ΔΕΔΟΜΕΝΑ	ΓΑΤΑ ΚΛΑΣΗ = 0	ΣΚΥΛΟΣ ΚΛΑΣΗ = 1
TN	4	
TP		3
FN	1	
FP		2

Που σημαίνει ότι το 25% των σκύλων ανήκει σε λάθος κατηγορία.

$$Recall \frac{3}{(3 + 1)} = 0.75 = 75\%$$

Εξίσωση 6: Παράδειγμα τύπου ανάκλησης

ο **F-measure / F1-Score:**

Το F1-Score δημιουργήθηκε από τον συνδυασμό της ανάκλησης και τις ευστοχίας τα οποία είναι κριτήρια επίδοσης και αναπαριστούν πληροφορίες για την θετική κλάση. Σύμφωνα με το scikit-learn ο τύπος του F-measure ή F1-Score είναι:

$$F1 = 2 * \frac{(Precision * Recall)}{(Precision + Recall)}$$

Εξίσωση 7: Ο μαθηματικός τύπος F1-Score / F-measure [6]

Πίνακας 4: Παράδειγμα λειτουργίας του τύπου F1-Score / F-measure

ΔΕΔΟΜΕΝΑ	ΓΑΤΑ ΚΛΑΣΗ = 0	ΣΚΥΛΟΣ ΚΛΑΣΗ = 1
TN	4	
TP		3
FN	1	
FP		2

$$Recall \frac{3}{(3 + 1)} = 0.75$$

$$Precision \frac{3}{(3 + 2)} = 0.6$$

$$F1 - Score = \left(\frac{(0.6 * 0.75)}{(0.6 + 0.75)} \right) * 2 = 0.66 = 66\%$$

Εξίσωση 8: Παράδειγμα μαθηματικού τύπου F1-Score / F-measure

- **Κριτήρια Επίδοσης Παλινδρόμησης**

Όπως αναφέρθηκε στην αρχή του κεφαλαίου τα κριτήρια ταξινόμησης είναι διαφορετικά από τα κριτήρια παλινδρόμησης, δηλαδή είναι τελείως διαφορετικά διαμορφωμένα. Εδώ θα αναφερθούν μερικά από τα κριτήρια που απασχολούν προβλήματα παλινδρόμησης. Στο συγκεκριμένο υποκεφάλαιο θα αναλυθεί μόνο το Mean Square Error.

- **Μέσο Τετραγωνικό Σφάλμα (Mean Squared Error)**

Μία από τις πιο απλές συναρτήσεις απώλειας είναι το Μέσο Τετραγωνικό σφάλμα. Σύμφωνα με το scikit-learn το Μέσο Τετραγωνικό Σφάλμα ή αλλιώς MSE είναι η διαφορά της προβλεπόμενης τιμής έναντι της πραγματικής τιμής υψωμένο εις την δύναμη δευτέρα. Ο τύπος του MSE είναι:

$$MSE(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} (y_i - \hat{y}_i)^2$$

Εξίσωση 9: Ο μαθηματικός τύπος Μέσου Τετραγωνικού Σφάλματος (MSE) [7]

Με βάση το scikit-learn υπάρχουν και άλλα κριτήρια επίδοσης για προβλήματα παλινδρόμησης (**Regression**) όπως το **Mean Absolute Error** (MAE) και το **Root Mean Square Error** (RMSE).

2. Μέθοδοι Μάθησης και Πρόβλεψης

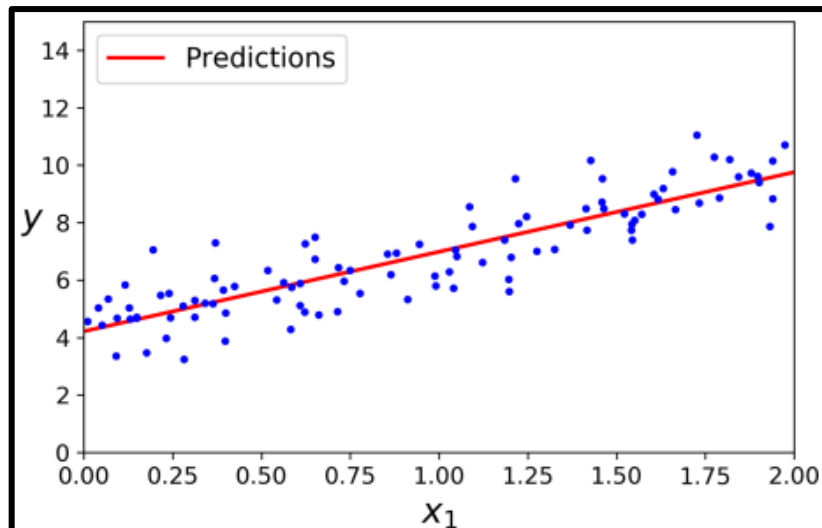
Σε αυτό το κεφάλαιο θα γίνει αναφορά στα μοντέλα μηχανικής μάθησης με επίβλεψη αλλά θα αναλυθούν σε μεγαλύτερο βάθος τα μοντέλα linear regression, regularize linear models και άλλοι αλγόριθμοι. Επιπρόσθετα, η Μηχανική Μάθηση περιλαμβάνει κάποιους αλγορίθμους Μηχανικής Μάθησης οι οποίοι αναλαμβάνουν συγκεκριμένες λειτουργίες αναλόγως τα προβλήματα που υπάρχουν, μερικοί εξ' αυτών ανήκουν στα Τεχνητά Νευρωνικά Δίκτυα που είναι τα πιο βασικά ενώ ταυτόχρονα υπάρχουν και άλλοι αλγόριθμοι που ανήκουν σε άλλες κατηγορίες Τεχνικών Νευρωνικών Δικτύων. Άλλες κατηγορίες για παράδειγμα είναι τα Συνελεκτικά Νευρωνικά Δίκτυα, τα Αναδρομικά, τα Μακρά Δίκτυα Βραχείας Μνήμης και τα Δίκτυα Bayesian. Στην κατηγορία των Τεχνητών Νευρωνικών Δικτύων οι αλγόριθμοι θα αναφερθούν μόνο ονομαστικά και δεν θα αναλυθούν εις βάθος όπως το υποκεφάλαιο «Μοντέλα Εκπαίδευσης».

2.1 Μοντέλα Προβλέψεων Μάθησης

Στο συγκεκριμένο υποκεφάλαιο θα αναπαρασταθούν μερικά από τα μοντέλα προβλέψεων της μάθησης με επίβλεψη και ο τρόπος λειτουργίας τους που προσεγγίζουν για να επιλύσουν προβλήματα ταξινόμησης (**classification**) ή παλινδρόμησης (**regression**). Κάποια από τα μοντέλα εκπαίδευσης είναι σχεδιασμένα για να μπορούν να επιλύουν πιο εύκολα προβλήματα χρονοσειρών (**time – series**) με σκοπό την πρόβλεψη μιας τιμής δηλαδή είναι πιο εξιδεικευμένο σε προβλήματα παλινδρόμησης, και άλλα μοντέλα είναι πιο εύκολο να επιλύσουν προβλήματα ταξινόμησης, δηλαδή για προβλήματα που τα δεδομένα δεν έχουν να κάνουν με βάση τον χρόνο. Για παράδειγμα η πρόβλεψη στο αν ένα άτομο είναι θετικός ή αρνητικός στον κορωνοϊό ή σε μια άλλη αρρώστια.

2.1.1 Linear Regression

Τα γραμμικά μοντέλα χρησιμοποιούνται εδώ και πολλά χρόνια. Έχουν σαν σκοπό την πρόβλεψη μελλοντικών τιμών [8]. Ο τρόπος λειτουργίας του είναι πολύ απλός, προσπαθεί να εφαρμόσει μία ευθεία νοητή γραμμή στα δεδομένα ώστε να έχουνε το λιγότερο δυνατό σφάλμα μεταξύ της γραμμής και των δεδομένων.



Εικόνα 1: Η απεικόνιση του Linear Regression δύο χαρακτηριστικών (Πηγή: Géron, 2019)

Δηλαδή, είναι ένα μοντέλο γραμμικής παλινδρόμησης σύμφωνα την βιβλιοθήκη της scikit-learn. Επίσης η απεικόνιση των σχεδιαγραμμάτων του Linear Regression μπορούν να κατανοηθούν όταν τα γραφήματα είναι δισδιάστατα ή τρισδιάστατα, δηλαδή το μοντέλο δέχεται ως δύο ή τρία χαρακτηριστικά (**features**). Στην περίπτωση που δεχτεί παραπάνω από τρία χαρακτηριστικά (**features**), ποιος μπορεί να κατανοήσει ένα σχεδιάγραμμα που έχει παραπάνω από τρεις διαστάσεις; Για αυτό τον λόγο στην παραπάνω εικόνα η απεικονίζεται σε δισδιάστατη μορφή το Linear Regression για να είναι πιο εύκολη η κατανόησή του. Ένα άλλο θετικό είναι πως ο συγκεκριμένος αλγόριθμος μηχανικής μάθησης είναι πάρα πολύ εύκολος να εκπαιδευτεί [9]. Η δομή του μοντέλου που χρησιμοποιεί το Linear Regression [10] είναι:

$$y = \theta^0 + \theta^1x^1 + \theta^2x^2 + \dots + \theta^nx^n$$

Εξίσωση 10: Εξίσωση μοντέλου Linear Regression

- Το **y** ή αλλιώς ο στόχος (target) είναι η τιμή η οποία προσπαθούμε να προβλέψουμε με βάση την παραπάνω φόρμουλα.
- Τα **x** είναι ο αριθμός των χαρακτηριστικών που θα εκπαιδευτεί το μοντέλο.
- Το **θ^0** είναι το σημείο που τέμνει το **y** - άξονα.
- Το **θ^1** έως το **θ^n** είναι το βάρος των χαρακτηριστικών (**features**) και οι παράμετροι (**parameter**) που θα υπολογίσει το μοντέλο της γραμμικής παλινδρόμησης.

Όπως υποστηρίζει το scikit-learn για (θ_1) έως (θ_n) οι παράμετροι ονομάζονται συντελεστές (**coefficients**) ενώ το (θ_0) ονομάζεται τομή (**intercept**). Επίσης τα x είναι ανεξάρτητες μεταβλητές ενώ τα (θ) είναι οι εξαρτημένες. Ο τρόπος που υπολογίζει τα (θ) αναφέρονται στο κεφάλαιο «Υπολογισμός Παραμέτρων»

Έστω ότι θέλουμε να προβλέψουμε την τιμή ενός σπιτιού και έχει ως χαρακτηριστικά (features) τα δωμάτια και την τιμή.

Πίνακας 5: Παράδειγμα αναπαράστασης φόρμουλας Linear Regression

ΔΩΜΑΤΙΑ	ΤΙΜΗ
4	10.000

Η αναπαράσταση της παραπάνω φόρμουλας είναι: $y = \theta_0 + \theta_1(\Delta\Omega\text{ΜΑΤΙΑ})$

Άρα το Linear Regression προσπαθεί να εντοπίσει τα κατάλληλα θ , ώστε να προσεγγίζουν ποιο κοντά στην τιμή. Πως μπορεί να θεωρηθεί ένα μοντέλο αξιόπιστο; Όπως και στα άλλα μοντέλα, έτσι και αυτό υπάρχουν διάφορες συναρτήσεις κόστους. Στη συγκεκριμένη περίπτωση θα αναλυθεί το Mean Square Error. Με βάση το Mean Square Error (MSE) μετριέται η αξιοπιστία ενός μοντέλου και ιδανικό είναι να προσεγγίζει το δυνατό ελάχιστο αριθμό. Ο τύπος της συνάρτησης κόστους είναι:

$$MSE_{test} = \left(\frac{1}{m}\right) \sum_i ((y^{test}) - (\hat{y}^{test}))^2$$

Εξίσωση 11: Τύπος συνάρτησης κόστους Mean Square Error(MSE) [11]

Η παραπάνω συνάρτηση κόστους είναι ένας μαθηματικός τύπος ο οποίος αφαιρεί τα προβλεπόμενα αποτελέσματα με τα πραγματικά, τα υψώνει εις την δευτέρα και τα αθροίζει. Το αποτέλεσμα συνάρτησης είναι το Mean Square Error.

2.1.2 Regularized Linear Models

Σε αυτό το υποκεφάλαιο θα αναφερθούν μερικά κανονικοποιημένα μοντέλα. Το Ridge Regression, το Lasso Regression και το Elastic Net Regression Τα παραπάνω μοντέλα που αναφέρθηκαν ανήκουν στην γενιά της μηχανικής μάθησης των γραμμικών

μοντέλων, απλά είναι ποιο κανονικοποιημένα. Το βασικό στοιχείο από τα κανονικοποιημένα μοντέλα είναι πως έχουν την ικανότητα να αποφύγουν ποιο εύκολα την υπερπροσαρμογή (**overfitting**), αλλά αυτό δεν σημαίνει ότι δεν μπορεί να συμβεί.

2.1.2.1 Ridge Regression

Κατά την δεκαετία του 1970 οι Hoerl και Kennard εφηύραν μία νέα μέθοδο που θα επίλυε διάφορα προβλήματα παλινδρόμησης η οποία μέθοδος ονομάζεται Ridge Regression και έχει σαν κριτήριο την συνάρτηση κόστους Mean Square Error (MSE) [12]. Όπως το Linear Regression, έτσι και το Ridge Regression ανήκει στα γραμμικά μοντέλα της μηχανικής μάθησης και λειτουργεί περίπου όπως το Linear Regression, δηλαδή προσπαθεί να ελαχιστοποιήσει την συνάρτηση κόστους [13]. Η διαφορά μεταξύ των δύο γραμμικών μοντέλων είναι πως το Ridge Regression αλλάζει ο μαθηματικός τύπος στην συνάρτηση κόστους. Η μία από τις δύο διαφορές είναι πως τα βάρη (**weights**) υψώνονται εις την δευτέρα και η άλλη διαφορά είναι πως στην φόρμουλα προστίθεται μία υπέρ παράμετρος κανονικοποίησης (α) η οποία εάν έχει την τιμή 0 το Ridge Regression λειτουργεί ακριβώς όπως το Linear Regression. Για αυτό και το συγκεκριμένο μοντέλο είναι μία έκδοση κανονικοποιημένης γραμμικής παλινδρόμησης. Αυτό έχει σαν αποτέλεσμα όταν στην υπέρ παράμετρο (α) εισάγεται μία τιμή διάφορη του 0, το μοντέλο μειώνει τον αριθμό των χαρακτηριστικών από τα βάρη (**weights**). Αυτό έχει σαν αποτέλεσμα ο συγκεκριμένος αλγόριθμος να γενικεύεται ποιο εύκολα και η επιρροή των βαρών να είναι ποιο μικρή. Ο λόγος που συμβαίνει αυτό είναι πως αν ένα χαρακτηριστικό έχει υψηλότερο βάρος (**weight**) τότε υπάρχει ενδεχόμενο το αποτέλεσμα της πρόβλεψης να είναι παραπλανητικό. Η συνάρτηση κόστους του Ridge Regression είναι:

$$J(\theta) = MSE(\theta) + \alpha \left(\frac{1}{2}\right) \sum_n^{i=1} = 1\theta_i^2$$

Εξίσωση 12: Τύπος συνάρτησης κόστους Ridge Regression [10]

2.1.2.2 Lasso Regression

Το Least Absolute Shrinkage and Selection Operator Regression ή αλλιώς LASSO είναι και αυτό ένα μοντέλο γραμμικής παλινδρόμησης και χρησιμοποιείται συνήθως σε περιπτώσεις που υπάρχει μεγάλος όγκος δεδομένων [14]. Όπως το Ridge Regression

(παλινδρόμηση κορυφογραμμής) έτσι και το Lasso Regression (Παλινδρόμηση Lasso) ανήκουν και τα δύο στην οικογένεια των Regularized Models (Κανονικοποιημένα Μοντέλα).

Αυτά τα δύο μοντέλα έχουν κάποια κοινά αλλά και κάποιες μικρές διαφορές μεταξύ τους. Το κοινό στοιχείο που έχουν αυτά τα δύο μοντέλα είναι πως η υπέρ παράμετρος του Lasso δηλαδή το (α) πάρει την τιμή 0 λειτουργεί ακριβώς όπως το Linear Regression [15]. Η διαφορά από την άλλη μεταξύ των δύο μοντέλων είναι πως στην συνάρτηση κόστους του Lasso τα βάρη (θ) μπαίνουν σε απόλυτη τιμή, αλλά το Lasso έχει και μία ακόμα ικανότητα. Τα βάρη τα οποία δεν έχουν μεγάλη επιρροή μπορεί να τα διαγράψει. Δηλαδή το Lasso κρατάει το αποτέλεσμα των χαρακτηριστικών όποια είναι σημαντικότερα για αυτό [16]. Η συνάρτηση κόστους του Lasso Regression είναι:

$$J(\theta) = MSE(\theta) + \alpha \sum_{i=1}^n |\theta_i|$$

Εξίσωση 13: Τύπος συνάρτησης κόστους Lasso Regression [10]

2.1.2.3 Elastic Net Regression

Το Elastic Net (ELNET) είναι ο συνδυασμός μεταξύ του Lasso Regression (Tibishirani) και Ridge Regression (Hoerl και Kennard). Το Elastic Net έχει την ικανότητα να μπορεί να επιλέξει αν θα χρησιμοποιήσει τον ένα από τους δύο συνδυασμούς των μοντέλων ή και των δύο μαζί [17]. Η ιδέα της ένωσης των δύο μοντέλων δημιουργήθηκε προκειμένου να γίνουν πιο ακριβείς οι προβλέψεις και να ελαχιστοποιηθεί το σφάλμα από τα πιο απλά γραμμικά μοντέλα [18]. Σύμφωνα με την συνάρτηση κόστους του Elastic Net:

$$J(\theta) = MSE(\theta) + r\alpha \sum_{i=1}^n |\theta_i| + \left(\frac{1-r}{2}\right) \alpha \sum_{i=1}^n \theta_i^2$$

Εξίσωση 14: Τύπος συνάρτησης κόστους Elastic Net

Ο αλγόριθμος δέχεται δύο υπέρ παραμέτρους (r) και (α) τα οποία ρυθμίζει τον συνδυασμό των δύο μοντέλων.

Πίνακας 6: Λειτουργία συνδυασμών των δύο μοντέλων

	r
Lasso Regression	1
Ridge Regression	0
Elastic Net Regression	0.5

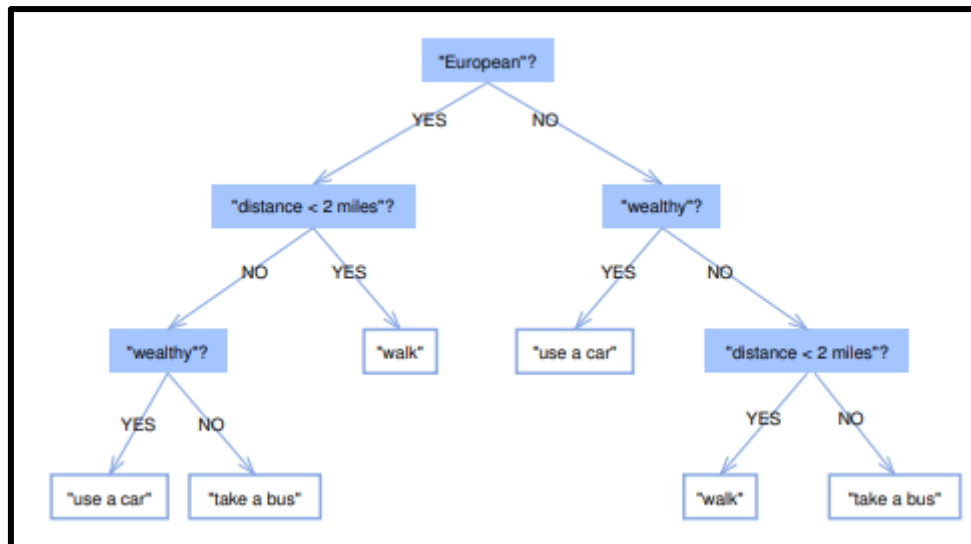
Ο παραπάνω πίνακας απεικονίζει τα κριτήρια που διαλέγει για την επιλογή ενός μοντέλου. Όταν $r = 1$ το Elastic Net χρησιμοποιεί μόνο τον έναν από τους δύο αλγόριθμους (Lasso Regression), ενώ όταν $r = 0$ ο αλγόριθμος χρησιμοποιεί το κομμάτι του Ridge Regression. Για να χρησιμοποιηθούν και οι δύο αλγόριθμοι ταυτόχρονα η τιμή του (r) πρέπει να είναι μεγαλύτερη του 0 και μικρότερη του 1.

2.1.3 Δέντρα

Τα δέντρα έχουν έναν διαφορετικό τρόπο προσέγγισης των προβλημάτων και δεν λειτουργούν όπως τα γραμμικά μοντέλα. Ένα δέντρο μαθαίνει κανόνες απόφασης και με βάση αυτό κάνει τις κατάλληλες προβλέψεις [19]. Με βάση το scikit-learn τα δέντρα είναι μοντέλα μηχανικής μάθησης και χρησιμοποιούνται για προβλήματα ταξινόμησης (**classification**) αλλά και παλινδρόμησης (**regression**).

2.1.3.1 Decision Trees

Τα Decision Tree έκαναν την πρώτη τους εμφάνιση την δεκαετία του 60' από τον Hunt. Η δομή ενός Decision Tree (**Δέντρο Απόφασης**) αρχίζει από πάνω που είναι η ρίζα και ο κορμός που επεκτείνεται προς τα κάτω [16]. Ένα δέντρο απόφασης αποτελείται από κόμβους και φύλλα. Κάθε κόμβος είναι ένα κριτήριο και μπορεί να περιέχει δύο ή περισσότερα παιδιά και κάθε φύλλο είναι μία πιθανή απάντηση [20]. Το δέντρο απόφασης δέχεται ως χαρακτηριστικά (**features**) τα οποία εισέρχονται από την ρίζα και με βάση των κριτηρίων θα καταλήξει σε μία συγκεκριμένη απόφαση (τελική έξοδος) [21]. Τα δέντρα αποφάσεων είναι χρήσιμα σε πολλούς από τους κλάδους της διεθνής κοινότητας αλλά είναι κυρίως αγαπητά επειδή είναι ποιο κατανοήσιμη η αποτύπωση των γραφημάτων τους. Κυρίως δουλεύουν καλά όταν υπάρχει πληθώρα πληροφορία για να εκπαιδευτούν [22].



Εικόνα 2: Η μορφή ενός δέντρου αποφάσεων (Πηγή: Battiti and Brunato, 2013)

Παρακάτω θα αναλυθούν μερικά από τα πλεονεκτήματα και τα μειονεκτήματα που έχουν τα Decision Tree σύμφωνα με το scikit-learn.

- **Πλεονεκτήματα:**

1. Ένα θετικό χαρακτηριστικό των δέντρων είναι πως μπορείς να δεις την αναπαράσταση με τα οποία λαμβάνει υπόψη τα κριτήρια.
2. Δεν χρειάζεται μεγάλη μετατροπή δεδομένων.
3. Έχει την ικανότητα να κάνει ταχύτερες προβλέψεις σε μεγάλο όγκο δεδομένων, δηλαδή δεν έχει μεγάλη πολυπλοκότητα.

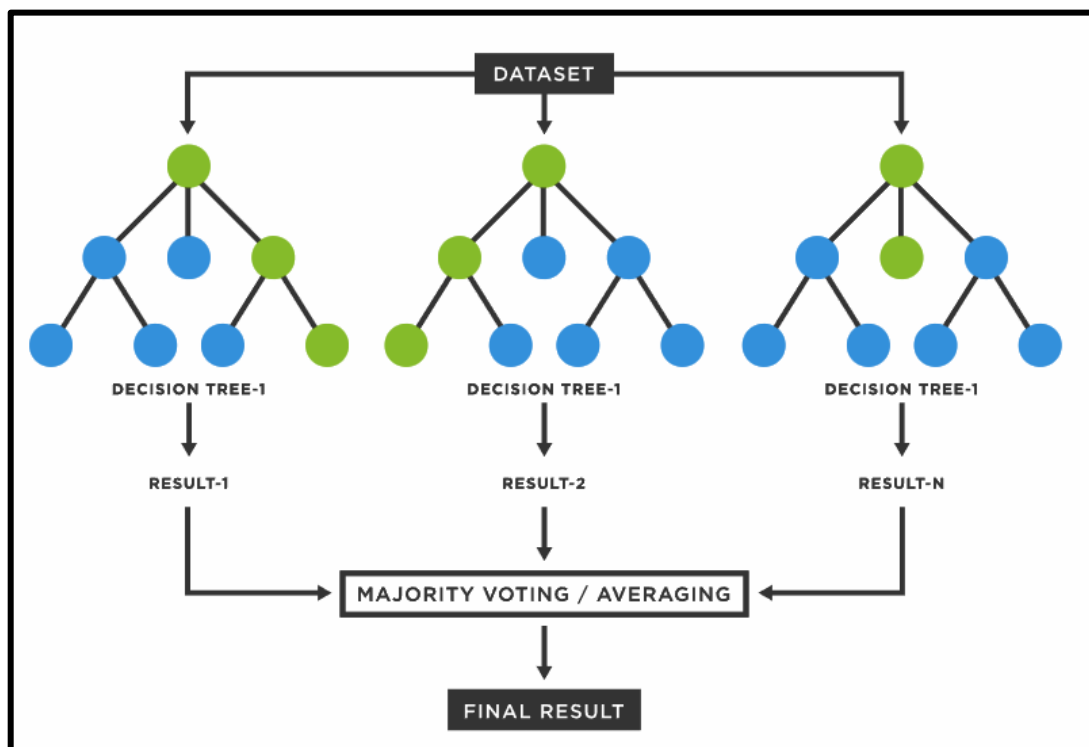
- **Μειονεκτήματα:**

1. Υπάρχει μεγάλος κίνδυνος υπερπροσαρμογής (**overfitting**) με αποτέλεσμα το δέντρο να μη μπορεί να αποδίδει αντικειμενικά.
2. Κάνουν προβλέψεις κατά προσέγγιση με βάση των τιμών που υπάρχουν, δηλαδή αν μία τιμή δεν υπάρχει στα δεδομένα το μοντέλο δεν θα μπορεί να την εντοπίσει για να γίνει η πρόβλεψή της.
3. Τα δέντρα μπορούν εύκολα να ξεγελαστούν από χαρακτηριστικά που έχουν μεγάλη επιρροή.

2.1.3.2 Random Forests

Σύμφωνα με το scikit-learn το Random Forest μπορούν να χρησιμοποιηθούν για προβλήματα ταξινόμησης (**classification**) με τον αλγόριθμο RandomForestClassifier ή

ακόμα και για προβλήματα παλινδρόμησης (**regression**) με τον αλγόριθμο RandomForestRegressor. Το Random Forest (**Αλγόριθμος Τυχαίου Δάσους**) αναφέρθηκε πρώτη φορά από τον Tim Kam Ho το 1995. Ο συγκεκριμένος αλγόριθμος είναι ένα απλό δέντρο σαν το Decision Tree αλλά δεν είναι αυτό που το χαρακτηρίζει. Αυτό που το κάνει ιδιαίτερο είναι πως δημιουργεί πολλά δέντρα και βγάζει μία αντικειμενική άποψη με βάση τον τυχαίων δέντρων που δημιουργήθηκαν σύμφωνα με την μελέτη του Amit και Geman το 1997, για παράδειγμα μπορεί να λάβει σαν κριτήριο τον μέσο όρο των δέντρων [2]. Βέβαια τα δέντρα που δημιουργούνται δέχονται ένα μέρος των χαρακτηριστικών σε κάθε δέντρο που εκπαιδεύεται. Ο σκοπός είναι που δέχεται τυχαία χαρακτηριστικά (**features**) είναι για να αποφύγει τις μεγάλες συσχετίσεις ώστε από πολλά αδύναμα δέντρα να δημιουργηθεί ένα καλό γενικευμένο δέντρο Ένα από τα πλεονεκτήματα του Random Forest είναι ότι υπερπροσαρμόζεται (**overfitting**) δύσκολα, μπορεί να χειριστεί μεγάλο όγκο δεδομένων και είναι αρκετά γρήγορος λόγω της μικρής πολυπλοκότητάς του. Η δημιουργία των Random Forest [23] έφερε μεγάλη εξέλιξη στην αγορά όπου έφερε μεγαλύτερη ακρίβεια στις λύσεις των προβλημάτων.



Εικόνα 3: Η δομή των Random Forest κατά την διάρκεια της εκπαίδευσής τους

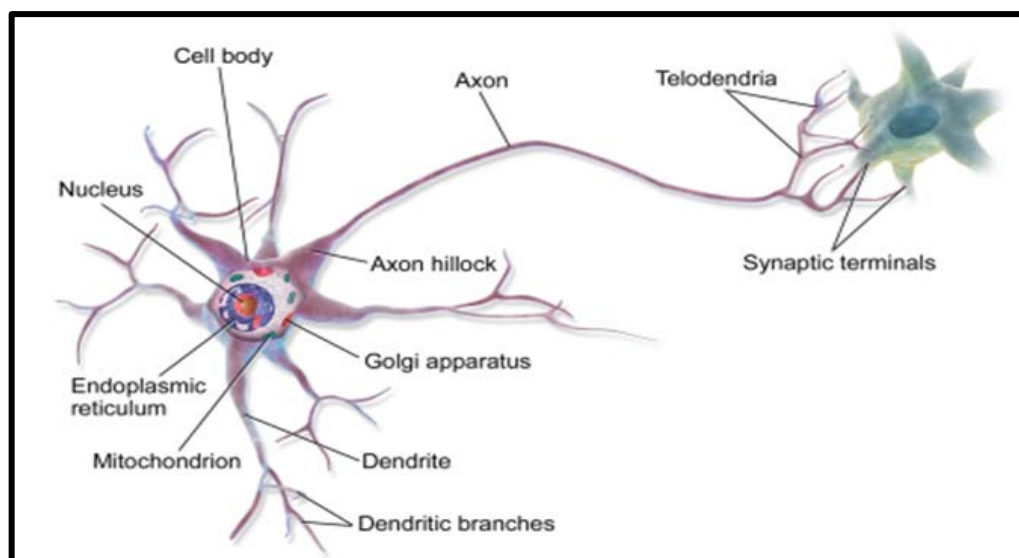
Πηγή: <https://www.tibco.com/reference-center/what-is-a-random-forest>

2.2 Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Network)

Τα τεχνητά νευρωνικά δίκτυα [2] [24] είναι μοντέλα μηχανικής μάθησης, άλλοι υποστηρίζουν πως είναι εμπνευσμένα από την λειτουργία του ανθρώπινου εγκεφάλου και άλλοι επίσης υποστηρίζουν πως η δημιουργία τους είναι εμπνευσμένο από ζωικό εγκέφαλο.

2.2.1 Βιολογικοί Νευρώνες

Πριν αναλυθούν οι νευρώνες της Τεχνικών Νευρωνικών Δικτύων θα αναφερθεί πρώτα στο συγκεκριμένο υποκεφάλαιο η λειτουργία των βιολογικών νευρώνων. Η δομή ενός νευρώνα είναι το σώμα δηλαδή ο πυρήνας του, τους δενδρίτες και έναν άξονα το οποίο αντιπροσωπεύει την έξοδο. Με άλλα λόγια ένας νευρώνας [10] είναι μία μονάδα που δέχεται και αποστέλλει κάποιες πληροφορίες σε άλλους νευρώνες. Οι νευρώνες αλληλοεπιδρούν μεταξύ τους με τους δενδρίτες οι οποίοι είναι προέκταση των νευρώνων, δηλαδή διακλαδώσεις και δημιουργούν ηλεκτρικά σήματα που μεταφέρονται πάνω στους άξονες μεταξύ όλων των νευρώνων.



Εικόνα 4: Η μορφή ενός βιολογικού νευρώνα (Πηγή: Géron, 2019)

2.2.2 Ιστορική Αναδρομή

Το θέμα των Τεχνητών Νευρωνικών Δικτύων άρχισε να θίγεται για πρώτη φορά την δεκαετία του 1940 από τον McCulloch και Pitts όπου παρουσίασαν το πρώτο μοντέλο Τεχνητού Νευρωνικού Δικτύου. Αργότερα στην δεκαετία του 1950, ο Frank Rosenblatt

δημιούργησε το πρώτο νευρωνικό δίκτυο με όνομα Perceptron και αποτελεί μέθοδος της επιβλεπόμενης μάθησης (**supervised learning**) [25] ενώ λίγο καιρό μετά στα τέλη της δεκαετίας του 50' δημιουργήθηκαν τα μοντέλα Τεχνητών Νευρωνικών Δικτύων Adaline και Madaline που εφαρμόστηκε κυρίως σε τηλεφωνικά δίκτυα τα οποία υλοποιήθηκαν από τους Widrow και Hoff [2]. Κατά την διάρκεια του 1960 [10] υπήρχε η ιδέα ότι θα συνομιλούσαν οι άνθρωποι με τις μηχανές αλλά αυτή η ιδέα γρήγορα δεν πήρε μεγάλη έκταση. Το 1980 υπήρξαν νέες εφευρέσεις πάνω στα νευρωνικά δίκτυα αλλά επειδή η πρόοδός τους ήταν αργή για άλλη μια φορά το θέμα δεν επεκτάθηκε. Στη συνέχεια το 1990 αναπτύχθηκαν άλλες τεχνικές μηχανικής μάθησης όπου πρόσφεραν ποιο έγκυρα αποτελέσματα.

2.2.3 Χρήση των Τεχνητών Νευρωνικών Δικτύων

Τα Τεχνητά Νευρωνικά Δίκτυα μπορούν να χρησιμοποιηθούν σε διάφορους τομείς. Μερικοί από αυτούς τους τομείς που χρησιμοποιούνται στα Τεχνητά Νευρωνικά Δίκτυα είναι Προβλέψεις πάνω σε οικονομικά δεδομένα, στον Ιατρικό Τομέα, στην Επεξεργασία Σήματος, σε θέματα Άμυνας και σε άλλους διάφορους τομείς.

- **Προβλέψεις πάνω σε οικονομικά δεδομένα**

Όπως αναφέρεται στο παραπάνω υποκεφάλαιο τα ΤΝΔ [1] [26] μπορούν να φανούν χρήσιμα στον διεθνή οικονομικό τομέα καθώς έχουν την ικανότητα να κάνουν προβλέψεις πάνω σε οικονομικά δεδομένα και έτσι να συνεργήσουν στην οικονομική ευημερία και ανάπτυξη γενικότερα αλλά και να αποτρέψουν καταστάσεις που μπορεί να προκαλέσουν σε ακραίο επίπεδο ένα οικονομικό κραχ. Δηλαδή τα ΤΝΔ μπορούν να χρησιμοποιηθούν σε προβλέψεις τιμών όπως μετοχές, κρυπτονομίσματα, στον χώρο της ενέργειας, στον πρωτογενή τομέα (γεωργία) και σε άλλα. Για να μπορέσει όμως ένα Τεχνητό Νευρωνικό Δίκτυο να δουλέψει σωστά θα χρειαστεί άτομα που έχουν εμπειρία και γνώσεις στρατηγικής πάνω στην οικονομία. Άτομα με γνώσεις στρατηγικής πάνω στην οικονομία χρειάζονται για την χρήση ενός τέτοιου Τεχνητού Νευρωνικού Δικτύου γιατί σε όσο ποιο χρήσιμη πληροφορία εκπαιδευτεί ένα μοντέλο τόσο ποιο ακριβείς θα είναι στις προβλέψεις του.

- **Επεξεργασία και Αναγνώριση Εικόνας**

Τα ΤΝΔ μπορούν να χρησιμοποιηθούν ακόμα για την επεξεργασία και αναγνώριση

μίας εικόνας. Για παράδειγμα ένα μοντέλο ΤΝΔ μπορεί να αναγνωρίσει μία θολή εικόνα, να τροποποιήσει την θολή εικόνα και να της δώσει όσο περισσότερο δυνατό μεγαλύτερη ευκρίνεια. Επίσης τα ΤΝΔ εφαρμόζονται από βιομηχανίες όπου αναγνωρίζουν αντικείμενα και αποφασίζουν σε ποια κατηγορία ανήκουν.

- **Ιατρικός Τομέας**

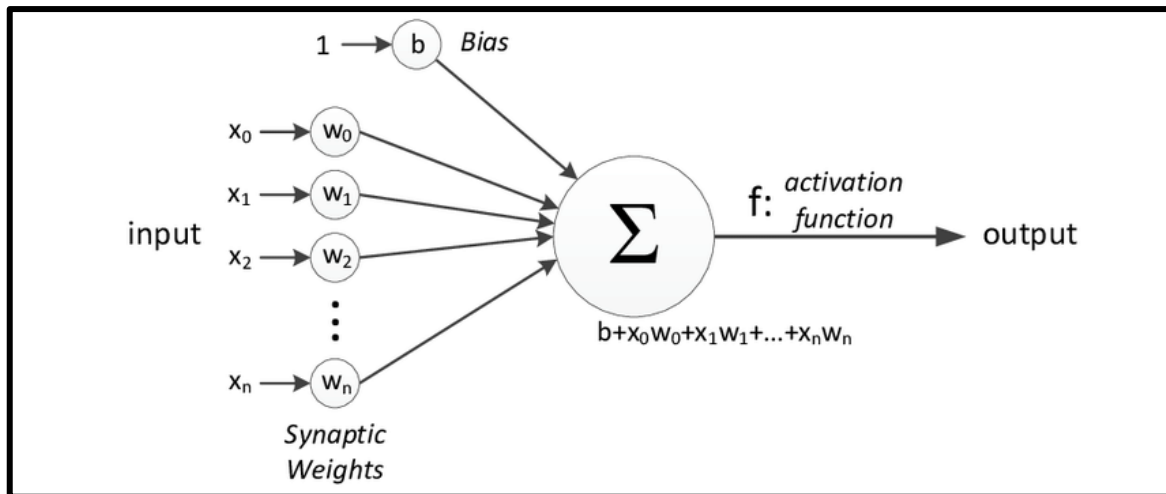
Όπως στην επεξεργασία και αναγνώριση μίας εικόνας έτσι και στον Ιατρικό Τομέα τα ΤΝΔ χρησιμοποιούνται ευρέως [1] [27] [26]. Δηλαδή διάφορα ιατρικά συστήματα χρησιμοποιούν εκπαιδευμένους κατάλληλα αλγόριθμους ΤΝΔ για την διάγνωση και ταξινόμηση μιας αρρώστιας.

- **Θέματα Άμυνας**

Όπως και στους προηγούμενους τομείς, έτσι και στα θέματα άμυνας τα ΤΝΔ έχουν κάνει την εμφάνισή τους. Τα Τεχνητά Νευρωνικά Δίκτυα χρησιμοποιούνται ευρέως και σε θέματα άμυνας προκειμένου να καλύψουν τις διάφορες στρατιωτικές ανάγκες μίας χώρας καθώς επίσης και να συμβάλλουν στην ποιο αποτελεσματική άμυνα της απέναντι σε άλλη χώρα. Δηλαδή με την βοήθεια των ΤΝΔ μπορεί να ενημερωθεί άμεσα η ηγεσία μίας χώρας όταν για ένα γεγονός που συμβαίνει, αν πρόκειται για απειλή ή όχι. Για παράδειγμα σε έναν σύγχρονο στρατιωτικό εξοπλισμό, μία διόπτρα με την χρήση ενός μοντέλου ΤΝΔ να εντοπίζει τους στόχους.

2.2.4 Νευρωνικό Δίκτυο Perceptron

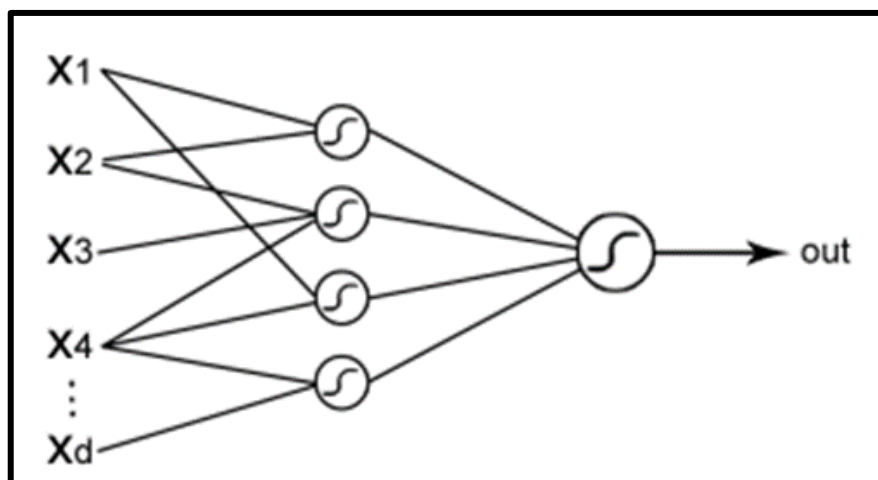
Το Perceptron είναι το πιο απλό νευρωνικό δίκτυο για την εποπτευόμενη μάθηση όπου εφευρέθηκε το 1958 από τον Rosenblatt. Αυτό το μοντέλο περιλαμβάνει μόνο ένα τεχνητό νευρώνα και σύμφωνα με την φόρμουλα κατανοεί γραμμικές σχέσεις. Ο τύπος της φόρμουλας του μοντέλου Perceptron είναι: $z = w_1 x_1 + w_2 x_2 + \dots + w_n x_n$. Το Perceptron είναι μία γραμμική εξίσωση όπου δέχεται χαρακτηριστικά και εάν το αποτέλεσμα της εξόδου είναι μικρότερο από μηδέν τότε βγάζει την αρνητική κλάση διαφορετικά αν το αποτέλεσμα είναι μεγαλύτερο του μηδενός τότε το διαχωρίζει στην θετική κλάση [10]. Κατά την εκπαίδευση του Perceptron προσπαθεί να προσαρμόσει τα βάρη έτσι ώστε να διαχωρίσει όσο καλύτερα μπορεί τα δεδομένα δηλαδή λειτουργεί σαν ένας ταξινομητής.



Εικόνα 5: Απεικόνιση νευρωνικού δικτύου Perceptron που περιέχει μόνο έναν νευρώνα

Πηγή: https://www.researchgate.net/figure/The-simplest-mathematical-model-of-a-neuron-called-the-Perceptron-30_fig2_266485234

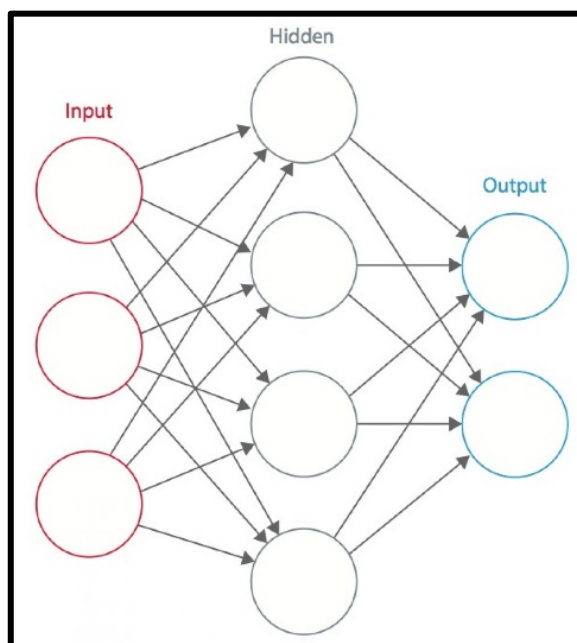
Εκτός από το νευρωνικό δίκτυο Perceptron υπάρχει και το πολυστρωματικό δίκτυο Perceptron (Multilayer Perceptron), δηλαδή νευρωνικά δίκτυα πολλαπλών στρωμάτων τα οποία περιέχουν περισσότερους από ένα νευρώνες. Ένα πολυστρωματικό νευρωνικό δίκτυο συνήθως περιέχει τρία ή και παραπάνω στρώματα. Τα πρώτα στρώματα είναι η είσοδος του νευρωνικού δικτύου, τα τελευταία είναι η έξοδος και τα ενδιάμεσα ονομάζονται κρυφά στρώματα (hidden layers) [16] [10]. Όταν ένα νευρωνικό δίκτυο όπως το Perceptron περιέχει πολλά βαθιά στρώματα, αυτό ονομάζεται βαθύ νευρωνικό δίκτυο (Deep Neural Network). Επίσης στα απλά νευρωνικά δίκτυα η ροή των δεδομένων κινείται μόνο προς μία κατεύθυνση (προς τα εμπρός).



Εικόνα 6: Απεικόνιση πολυστρωματικού νευρωνικού δικτύου Perceptron που περιέχει πολλά στρώματα εισόδου, ένα στρώμα εξόδου και ενδιάμεσα κρυφά στρώματα (Πηγή: Battiti & Brunato, 2013)

2.3 Συνελκτικικά Νευρωνικά Δίκτυα (Convolutional Neural Networks)

Όπως αναφέρθηκαν και στα προηγούμενα κεφάλαια, τα τεχνητά νευρωνικά δίκτυα (ANN) είναι εμπνευσμένα από τον τρόπο λειτουργίας του ανθρώπινου βιολογικού νευρωνικού συστήματος, έτσι και τα CNN αποτελούνται από νευρώνες και είναι εμπνευσμένα από αυτά, αλλά με την ιδιαιτερότητα ότι δίνεται περισσότερη έμφαση στην οπτική απεικόνιση των πραγμάτων. Τα Convolutional Neural Network ή αλλιώς CNN είναι μοντέλα τα οποία έχουν πολλά στρώματα (layers), δηλαδή ανήκουν στην κατηγορία των βαθέων νευρωνικών δικτύων (deep learning) [2] [28]. Σύμφωνα με τον Κωνσταντίνο Διαμαντάρα και Δημήτρη Μπότση τα CNN εμφανίστηκαν την δεκαετία του 80' από τον LeCun ενώ σύμφωνα με το βιβλίο (Τεχνητή Νοημοσύνη Δ' Έκδοση) τα CNN εμφανίστηκαν σαν ιδέα την δεκαετία του 60' από τους Hubel και Wiesel ενώ τα πρώτα μοντέλα CNN υλοποιήθηκαν την δεκαετία του 80' από τον Fukusima.



Εικόνα 7: Ένα βαθύ νευρωνικό δίκτυο CNN με ένα κρυφό στρώμα

Πηγή: <https://www.edge-ai-vision.com/2015/11/using-convolutional-neural-networks-for-image-recognition>

2.3.1 Αρχιτεκτονική δομή CNN

Η δομή των Συνελκτικικών Νευρωνικών Δικτύων (CNN) χωρίζεται σε τρία στάδια επεξεργασίας. Το πρώτο από τα τρία βασικά στάδια είναι τα Επίπεδα Συνέλιξης (Convolution Layers), στην συνέχεια ακολουθεί το Στάδιο Συγκέντρωσης (Pooling) και τέλος εμφανίζονται τα Πλήρως Διασυνδεδεμένα Επίπεδα (Fully Connected Layers).

- **Επίπεδα Συνέλιξης**

Το Επίπεδο Συνέλιξης (Convolution Layer) αναλαμβάνει την συλλογή ιδιαίτερων χαρακτηριστικών σε τοπικές περιοχές που διαθέτει μια εικόνα.

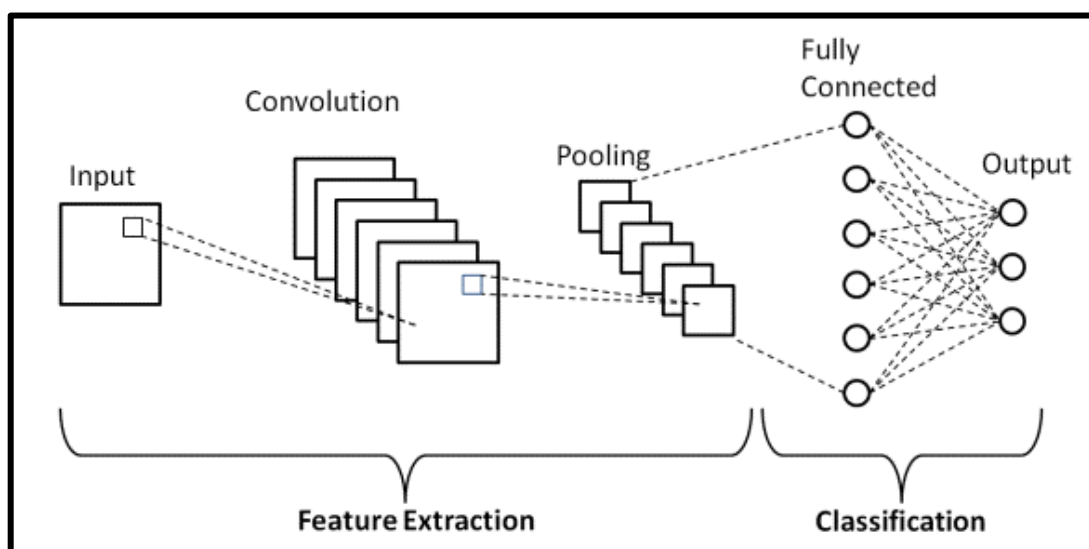
- **Στάδιο Συγκέντρωσης**

Το στάδιο συγκέντρωσης (Pooling) επιτυγχάνει την σμίκρυνση των διαστάσεων μίας εικόνας έτσι ώστε να απαλλαχτεί από τις ασήμαντες πλέον πληροφορίες που είχε προηγουμένως. Αυτό έχει και σαν αποτέλεσμα να εκπαιδεύεται σε ποιο σημαντικές πληροφορίες αλλά και την εξοικονόμηση χρόνου καθώς τα βάρη λιγοστεύουν.

- **Πλήρως Διασυνδεδεμένα Επίπεδα**

«Πλήρως Διασυνδεδεμένα Επίπεδα (Fully Connected Layers), που βρίσκονται στην έξοδο και υλοποιούν τον ταξινομητή.»

Σε μία αρχιτεκτονική δομή Συνελεκτικών Νευρωνικών Δικτύων, μπορεί η δομή να εμφανίζεται με πολλαπλά στάδια συνέλιξης που περιέχει το Επίπεδο Συνέλιξης (Convolutional Layer) συνοδευόμενο από νευρώνα ReLU μαζί με ένα Στάδιο Συγκέντρωσης (Pooling). Όλα αυτά τα στάδια συνέλιξης μπορούν να εμφανίζονται ή όχι πολλές φορές άσχετα αν υπάρχει το Στάδιο Συγκέντρωσης (Pooling) καθώς είναι προαιρετικό επίπεδο. Στο τέλος θα καταλήξουν στο Πλήρως Διασυνδεδεμένο Επίπεδο (Fully Connected Layer) όπου γίνεται η εξαγωγή του αποτελέσματος. Ακολουθεί μία εικόνα που περιγράφει ένα ολοκληρωμένο Συνελεκτικό Νευρωνικό Δίκτυο CNN.



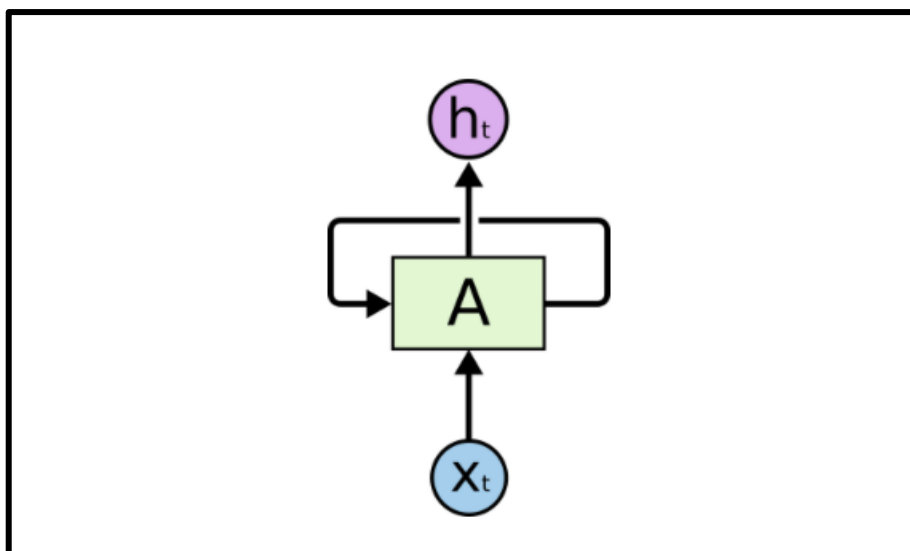
Εικόνα 8: Βασική αρχιτεκτονική δομή CNN

Πηγή: <https://www.upgrad.com/blog/basic-cnn-architecture/>

Όπως αναφέρεται στο βιβλίο (Τεχνητή Νοημοσύνη Δ' Έκδοση) ένα από τα μειονεκτήματα που έχουν τα μοντέλα Συνελεκτικών Νευρωνικών Δικτύων (CNN) είναι πως χρειάζονται μεγάλο όγκο σε δεδομένα για την εκπαίδευσή τους αλλά είναι και αργά στον τρόπο που εκπαιδεύονται. Τα μοντέλα CNN κατά κύριο λόγο χρησιμοποιούνται για την εύρεση αντικειμένων που απεικονίζονται σε εικόνες. Εκτός από την εύρεση αντικειμένων μπορούν να εκπαιδευτούν σε οποιαδήποτε πληροφορία, αρκεί αυτή η πληροφορία να είναι δύο διαστάσεων [2]. Μερικά από τα μοντέλα CNN είναι τα LetNet-5, AlexNet, GoogLeNet αλλά στην παρούσα εργασία δεν θα αναλυθούν τα συγκεκριμένα μοντέλα γιατί θα δοθεί περισσότερη έμφαση σε μοντέλα προβλέψεων πάνω σε οικονομικά δεδομένα.

2.4 Αναδρομικά Νευρωνικά Δίκτυα (Recurrent Neural Networks)

Όπως και τα Συνελεκτικά Νευρωνικά Δίκτυα (CNN), έτσι και τα Αναδρομικά Νευρωνικά Δίκτυα (RNN) ανήκουν στην κατηγορία των βαθέων Τεχνικών Νευρωνικών Δικτύων. Η βασική διαφορά όμως των Αναδρομικών Νευρωνικών Δικτύων με τα πιο απλά νευρωνικά δίκτυα είναι πως ανατροφοδοτούνται δηλαδή οι κόμβοι τους μπορούν να κάνουν κύκλο και έχουν την ικανότητα να αποστέλλουν την πληροφορία πίσω στον εαυτό τους. Ένα μεγάλο πλεονέκτημα που έχουν οι αλγόριθμοι Αναδρομικών Νευρωνικών Δικτύων σε σχέση με άλλα Τεχνικά Νευρωνικά Δίκτυα που έχουν αναφερθεί στα προηγούμενα κεφάλαια είναι πως τα Αναδρομικά Νευρωνικά Δίκτυα επειδή έχουν την ικανότητα να κοιτάνε προς τα πίσω μπορούν να θυμούνται προηγούμενες πληροφορίες. Επίσης τα Αναδρομικά Νευρωνικά Δίκτυα μπορούν να επεξεργάζονται και να κάνουν υπολογισμούς πάνω σε μία σειρά δεδομένων όπου ακολουθεί το ένα μετά το άλλο, δηλαδή σειριακά. Αυτή η σειρά δεδομένων προσαρμόζεται κατά κύριο λόγο με βάση τον χρόνο, αλλά μπορεί και με οποιοδήποτε τρόπο. Για παράδειγμα μπορεί να είναι μία σειρά λέξεων, ηχητικό σήμα κλπ. Με άλλα λόγια τα Αναδρομικά Νευρωνικά Δίκτυα (RNN) όπως συμπεραίνουμε, είναι αλγόριθμοι Μηχανικής Μάθησης που χρησιμοποιούνται κυρίως σε περιπτώσεις που αφορούν τον χρόνο (time-series) ή πάνω σε σειρές δεδομένων [11] [21]. Η παρακάτω εικόνα αναπαριστά πως μοιάζει ένα απλό Αναδρομικό Νευρωνικό Δίκτυο:



Εικόνα 9: Αναπαράσταση ενός απλού Αναδρομικού Νευρωνικού Δικτύου όπου ανατροφοδοτείται ο κόμβος.

Πηγή: <https://camrongodbout.medium.com/recurrent-neural-networks-for-beginners-7aca4e933b82>)

Μερικές από τις εφαρμογές που χρησιμοποιούνται από αλγόριθμους Αναδρομικών Νευρωνικών Δικτύων είναι οι εξής:

- **Εφαρμογές Αναγνώρισης Ομιλίας** όπου με την εισαγωγή κάποιου κείμενου να μπορεί ο αλγόριθμος να καταλάβει τα συναισθήματα ενός ανθρώπου αν είναι χαρούμενος ή λυπημένος.
- **Πρόβλεψη Τιμών Μετοχών** όπου με βάση τον χρόνο (εποχή για παράδειγμα) να μπορεί ο αλγόριθμος να κρίνει πότε μπορεί να γίνει αγορά μιας μετοχής η πώληση μιας μετοχής ανάλογα με το αν οι μετοχές μία περίοδο είναι υψηλές ή χαμηλές. Τα RNN έχουν την ικανότητα να θυμούνται από το περασμένο χρονικό διάστημα και να κατευθύνουν ανάλογα.
- **Εφαρμογές Περιγραφής Εικόνας** όπου με την εισαγωγή μίας εικόνας να μπορεί να παράγεται ένα κείμενο όπου θα περιγράφει την εικόνα.
- **Εφαρμογές Μετάφρασης κειμένου** όπου για παράδειγμα σε μία σειρά χαρακτήρων ένας χαρακτήρας είναι λάθος, έχει την ικανότητα να κάνει συντακτική διόρθωση της συγκεκριμένης λέξης και να το μεταφράσει ταυτόχρονα [2].

Όπως αναφέρθηκε παραπάνω τα Αναδρομικά Νευρωνικά Δίκτυα (RNN) μπορούν να θυμούνται καταστάσεις που έχουν συμβεί στο παρελθόν αλλά έχουν ένα μειονέκτημα με βάση τον χρόνο. Σύμφωνα με τον Hochreiter και Bengio, το «πρόβλημα

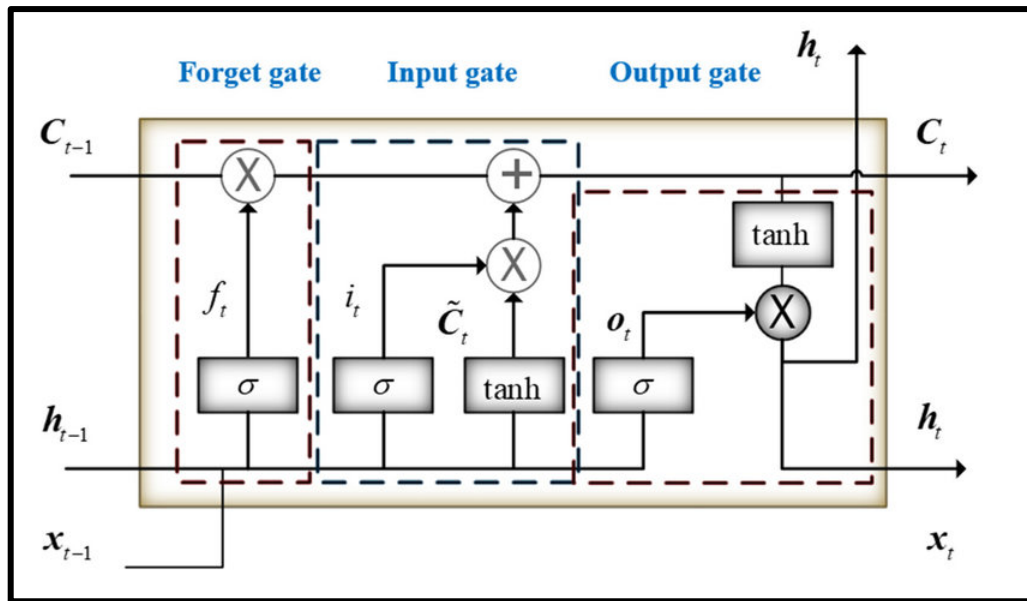
εξασθενούσης παραγώγου» (vanishing gradient problem) εμποδίζει τα μοντέλα RNN να θυμούνται μακροπρόθεσμα δεδομένα. Αυτό το πρόβλημα λύνεται με μία διαφοροποίηση των RNN, δηλαδή με τα Νευρωνικά Δίκτυα LSTM (Long Sort-Term Memory). Μερικά από τα μοντέλα Αναδρομικών Νευρωνικών Δικτύων είναι το μοντέλο Jordan και το μοντέλο του Elman [1].

2.4.1 Εκπαίδευση Αναδρομικών Νευρωνικών Δικτύων

Η εκπαίδευση των μοντέλων Αναδρομικών Νευρωνικών Δικτύων δεν διαφέρει πολύ από τον τρόπο που αναλύθηκε στην ενότητα «3.1.1 Μάθηση με Επίβλεψη» όπου το μοντέλο μαθαίνει από μία αλληλουχία δεδομένων και προσπαθεί να προβλέψει τους στόχους, δηλαδή τα δεδομένα που επιθυμεί ο χρήστης να βγάλει σαν αποτέλεσμα. Μία απλή μέθοδος που χρησιμοποιείται για την εκπαίδευση των μοντέλων των Αναδρομικών Νευρωνικών Δικτύων (RNN) είναι το back-propagation [1]. Η μέθοδος back-propagation είναι ένας αλγόριθμος [2] [21] όπου αναπτύχθηκε για πρώτη φορά την δεκαετία του 1960 και το κύριο χαρακτηριστικό του είναι η διάδοση του σφάλματος σηματοδότησης προς τα πίσω επίπεδα ενός Αναδρομικού Νευρωνικού Δικτύου και όχι μόνο στο ίδιο επίπεδο.

2.4.2 Μακρά Δίκτυα Βραχείας Μνήμης (Long Sort-Term Memory Networks)

Όπως αναφέρεται στην παραπάνω ενότητα, τα Μακρά Δίκτυα Βραχείας Μνήμης (LSTM) είναι μία βελτίωση των Αναδρομικών Νευρωνικών Δικτύων (RNN). Η ιδέα της εξέλιξης των RNN για να λυθεί το πρόβλημα που τα εμπόδιζε με την εκπαίδευση της εξασθενούσης παραγώγου (vanishing gradient problem) συζητήθηκε για πρώτη φορά σαν ιδέα το 1997 από τους Hochreiter και Schmidhuber όπου στην συνέχεια υπήρξε βελτίωση των LSTM από άλλους ερευνητές [2] [10]. Τα Μακρά Δίκτυα Βραχείας Μνήμης (LSTM) έκαναν έντονη την εμφάνισή τους με μεγάλη επιτυχία σε εφαρμογές όπως στην αναγνώριση ομιλίας, στην αυτόματη μετάφραση, στην ανάλυση και σε άλλα [11]. Τα τεχνητά νευρωνικά δίκτυα LSTM δεν διαφέρουν πολύ από τα αναδρομικά νευρωνικά δίκτυα RNN καθώς και στα δίκτυα LSTM υπάρχει η έννοια της αναδρομής. Η βασικής διαφορά που έχουν αυτά τα δύο νευρωνικά δίκτυα βρίσκεται στην αρχιτεκτονική τους.



Εικόνα 10: Η αρχιτεκτονική δομή μίας μονάδας cell τεχνητού νευρωνικού δικτύου LSTM

Πηγή: https://www.researchgate.net/figure/Basic-structure-of-a-long-short-term-memory-LSTM-unit_fig1_330890239

Η αρχιτεκτονική δομή ενός δικτύου LSTM αποτελείται από μία μονάδα c (cell) η οποία αυτή η μονάδα έχει σαν ρόλο να θυμάται κύκλους από παλαιότερες πληροφορίες για να επιλυθεί το πρόβλημα της μνήμης που υπάρχει. Η μονάδα cell αποτελείται από τρεις ξεχωριστές πύλες. Την Πύλη Λήθης (forget gate), την Πύλη Εισόδου (input gate), και τέλος την Πύλη Εξόδου (output gate). Η πύλη λήθης (forget gate) έχει σαν λειτουργία να συγκρίνει προηγούμενες καταστάσεις και θα ελέγχει πως θα επηρεάζονται οι νέες καταστάσεις. Εκτός από την πύλη λήθης, αποτελείται και από μία πύλη εισόδου (input gate) η οποία έχει σαν σκοπό να επεξεργάζεται την πληροφορία που εισέρχεται μέσα σε αυτήν. Τέλος, υπάρχει και μία πύλη εξόδου (output gate) η οποία με την σειρά της παράγει το τελικό αποτέλεσμα από την νέα κατάσταση που δημιουργήθηκε [2] [29]. Τέλος, υπάρχουν και άλλες παραλλαγές των τεχνικών νευρωνικών δικτύων LSTM όπως το GRU (Gated Recurrent Units), τα Multiplicative LSTM και άλλα διάφορα δίκτυα.

2.5 Νευρωνικά Δίκτυα Bayesian

Σε αυτό το κεφάλαιο θα αναλυθούν τα Νευρωνικά Δίκτυα Bayesian και το πως λειτουργούν αλλά για αρχή θα γίνει μία εισαγωγή στο θεώρημα του Bayes και πως λειτουργεί και στην συνέχεια θα ακολουθήσει μία ιστορική αναδρομή των νευρωνικών

δικτύων Bayesian αλλά και στην εμφάνισή τους, ένα υποκεφάλαιο που θα αναλυθεί τι είναι η αβεβαιότητα και ο τρόπος που αντιμετωπίζεται. Επίσης θα αναφερθούν και κάποια πλεονεκτήματα των νευρωνικών δικτύων Bayes.

2.5.1 Θεώρημα Bayes

Το θεώρημα του Bayesian εφευρέθηκε από τον Βρετανό μαθηματικό Thomas Bayes κατά την διάρκεια του 18^{ου} αιώνα. Το θεώρημά του είναι μία μαθηματική φόρμουλα στατιστικών που εφαρμόζεται στην λήψη αποφάσεων με βάση την πιθανότητα που έχει μία κατάσταση, δηλαδή ο νόμος του Bayes «υπολογίζει την πιθανότητα μίας κατάστασης με βάση την σχέση της με ένα άλλο σενάριο» [30] [31]. Επίσης ο νόμος του Bayes εφαρμόστηκε από τον Alan Turing στο Β΄ Παγκόσμιο Πόλεμο για την αποκρυπτογράφηση των μηνυμάτων. Εκτός απ' τον Β΄ Παγκόσμιο Πόλεμο, το θεώρημά του χρησιμοποιήθηκε επίσης και στον τομέα των θετικών επιστημών, για παράδειγμα σε μοντέλα μηχανικής μάθησης [32] όπως Gaussian Naive Bayes, Multinomial Naive Bayes, Complement Naive Bayes αλλά χρησιμοποιήθηκε και σε άλλους αλγόριθμους μηχανικής μάθησης.

2.5.2 Ιστορική Αναδρομή δικτύων Bayesian

Στο τέλος της δεκαετίας του 1980 ο Tishby είχε την ιδέα να χρησιμοποιήσει ένα θεώρημα πιθανοτήτων στα νευρωνικά δίκτυα με σκοπό να γίνουν ποιο αποτελεσματικά. Στις αρχές του 1990 ο Buntine και ο Weigend ανέπτυξαν ένα μοντέλο το οποίο στηριζότανε στην μέθοδο Bayes. Στην συνέχεια και άλλοι ερευνητές χρησιμοποιούσαν μοντέλα νευρωνικών δικτύων Bayesian όπου αποδείχτηκε ότι ήταν πολύ αποδοτικά στην λειτουργία τους. Τέλος στις αρχές του 2000 ο Lampinen και ο Vehtari έδειξαν την αποτελεσματικότητα των νευρωνικών δικτύων Bayesian όπου ήταν ποιο αποτελεσματικά σε σχέση με άλλα νευρωνικά δίκτυα.

2.5.3 Αβεβαιότητα

Στο παραπάνω υποκεφάλαιο αναλύθηκε το θεώρημα του Bayes όπου είναι βασισμένο στις πιθανότητες αλλά όταν γίνεται αναφορά σε πιθανότητες, δηλαδή ποσοστά εννοείται πως κάτι μπορεί να πραγματοποιηθεί ή και όχι. Άρα με τον όρο Αβεβαιότητα εννοείται ή έλλειψη της ακρίβειας σε μία κατάσταση [27]. Για αυτό το λόγο

στο σημείο αυτό γίνεται η εισαγωγή του θεωρήματος του Bayes προκειμένου να γίνουν προβλέψιμα τα μοτίβα αβεβαιότητας από τις πιθανότητες που μπορούν να προκύψουν.

2.5.4 Πλεονεκτήματα Νευρωνικών Δικτύων Bayes

Στα δίκτυα Bayes, η ιδέα της πιθανότητας και ο τρόπος εκπαίδευσής τους δημιουργήθηκαν με σκοπό την αντιμετώπιση αδυναμιών άλλων νευρωνικών δικτύων αλλά και την καλύτερη ακρίβεια αποτελεσμάτων. Μερικά από τα πλεονεκτήματα των δικτύων Bayes είναι:

- Είναι δύσκολο να γίνει υπερπροσαρμογή (**overfitting**) [33] [34] στα δεδομένα.
- Μπορεί να εκπαιδευτεί σε μικρό σύνολο δεδομένων αλλά και σε ημιτελή δεδομένα.
- Γρήγορη ανταπόκριση των μοντέλων Bayesian κατά την διάρκεια της μεταγλώττισης.
- Αντιμετώπιση της Αβεβαιότητας.
- Εύκολη προσαρμογή σε νέα πληροφορία.

2.5.5 Εφαρμογές στα Bayesian Networks

Στα χρόνια που τώρα ζει η ανθρωπότητα είναι απαραίτητα αναγκαίο να υπάρχουν κάποια εργαλεία τα οποία θα χρησιμοποιούνται προκειμένου να εξυπηρετούν τους σκοπούς των ανθρώπων. Τα εργαλεία αυτά θα προβλέπουν διάφορες καταστάσεις στις οποίες η ακρίβειά τους δεν είναι σταθερή και καθημερινός αυτές οι καταστάσεις μεταβάλλονται συνεχώς με νέα δεδομένα. Οι εφαρμογές των Bayesian είναι τα εργαλεία αυτά όπου με μεθόδους στατιστικής μπορούν να βοηθήσουν και σε γραφική μορφή να απεικονίσουν τις πιθανότητες και τα δεδομένα που αλλάζουν διαρκώς για ένα γεγονός ή μία κατάσταση. Μερικές από τις εφαρμογές των Bayesian δικτύων είναι οι ακόλουθες [35]. Τα δίκτυα Bayesian [31] μπορούν να χρησιμοποιηθούν σε προβλέψεις καιρού, για παράδειγμα εάν θα βρέξει ή αν θα είναι ποιο υψηλή η θερμοκρασία την επόμενη ημέρα. Επίσης μπορούν να χρησιμοποιηθούν σε οικονομικά δεδομένα, για παράδειγμα να πάρει κάποιες αποφάσεις μία επιχείρηση, άλλο ένα παράδειγμα που μπορεί να χρησιμοποιηθεί είναι σε αλγόριθμους μηχανικής μάθησης αλλά και σε άλλους τομείς οι οποίοι δεν θα αναλυθούν περαιτέρω.

3. Τεχνολογίες και Δεδομένα

Σε αυτό το κεφάλαιο θα περιγραφούν μερικές από τις τεχνολογίες όπου χρησιμοποιούνται ευρέως για την κατασκευή διαδικτυακών εφαρμογών που δραστηριοποιούνται στην συλλογή και πρόβλεψη οικονομικών δεδομένων. Αρχικά έγινε έρευνα για τα δεδομένα που θα χρησιμοποιηθούν (κρυπτονομίσματα) και από που θα αντληθούν αυτές οι πληροφορίες όπου αποφασίστηκε να αντληθούν από το API της Binance. Δεύτερον, ποια γλώσσα προγραμματισμού και ποιες βιβλιοθήκες θα χρησιμοποιηθούν για την υλοποίηση του προγράμματος. Τέλος, έπρεπε να αποφασιστεί πως θα είναι η δομή και ο σχεδιασμός του Web όπου χρησιμοποιούνται οι τεχνολογίες HTML, CSS και JavaScript καθώς και πως θα γίνει η συνδεσιμότητα και η αλληλεπίδραση με το API.

3.1 Χρήση και τεχνολογίες της Γλώσσας Προγραμματισμού

Αρχικά έπρεπε να αποφασιστεί με ποια γλώσσα προγραμματισμού και ποιες τεχνολογίες θα χρειαστούν για τις λειτουργίες του προγράμματος. Η γλώσσα προγραμματισμού είναι η Python-3 και επιλέχτηκε επειδή μπορεί να χρησιμοποιηθεί πληθώρα βιβλιοθηκών της γλώσσας όπου είναι χρήσιμες για την εξαγωγή δεδομένων, για την απεικόνιση και την ανάλυση των δεδομένων, για το κομμάτι της μηχανικής μάθησης αλλά και για την δημιουργία του API.

```
def Extract_Data(length):  
    ...  
    This Function return a DataFrame about Crypto  
    -----  
    Element: length (int value)  
    ...  
    client = Client()  
  
    date = datetime.datetime.today()  
    date = date.date()  
  
    dataframe = pd.date_range(start='01/01/2020', end=date, freq="1D").to_frame()  
    dataframe.index.name = 'Open time'  
    dataframe.drop([0], axis=1, inplace=True)  
  
    coin_list = ['BTCUSD', 'ETHUSD', 'BNBUSD'] #List of Cryptocurrency  
    columns = ['Open time', 'Open', 'High', 'Low', 'Close', 'Volume', 'Close time', 'Quote asset volume', 'Number of trades', 'TB  
  
    try:  
        for i in range(length):  
            temp = pd.DataFrame(client.get_historical_klines(coin_list[i], Client.KLINE_INTERVAL_1DAY, '1 Jan 2020'), columns=col  
            temp.index = pd.to_datetime(temp['Open time'], unit='ms')  
            temp.drop(['Open time', 'Close time', 'Quote asset volume', 'Number of trades', 'TB Base Vol', 'TB Quot Vol', 'Ignore'],  
            print('Extract: ', i+1, '\n')  
  
            for j in range(len(temp.columns)): #Rename the columns  
                temp.rename(columns={temp.columns[j] : coin_list[i]+'_'+temp.columns[j]}, inplace=True)  
            dataframe = dataframe.join(temp)  
            #print(temp)  
    except:  
        print('Length of coin_list is --->', len(coin_list))  
    return dataframe  
#
```

Εικόνα 11: Εξαγωγή δεδομένων

Στον παραπάνω κώδικα δημιουργείται μέσα στην συνάρτηση μία λίστα με τα ονόματα κρυπτονομισμάτων και μία λίστα με τις λίστες που θα χρησιμοποιηθούν ώστε να γίνει εξαγωγή αλλά και η τροποποίηση των δεδομένων. Η εξαγωγή δεδομένων γίνεται με την εντολή `get_historical_klines`. Από την στιγμή που θα ολοκληρωθεί η εξαγωγή των δεδομένων θα πρέπει να επιλεχτεί ποιος θα είναι ο στόχος (target) που θα πρέπει να προβλεφθεί.

```
def Select_Target(dataframe , name_of_col):
    """
    This function create a Target!
    And drop null rows of the dataframe
    -----
    Element: dataframe (Is a dataframe)
    """
    print('Choice the Target!')
    for i in range(len(dataframe.columns)):
        print(i,': ', dataframe.columns[i])

    target_columns = dataframe[name_of_col].shift(-1) #Select Target
    dataframe.insert(0, 'Target_'+name_of_col, target_columns) #First position
    dataframe.dropna(inplace=True) #Drop the null rows of dataframe

    #print(dataframe.columns)
    #print(dataframe)
    #return dataframe
```

Εικόνα 12: Επιλογή στόχου

Η παραπάνω συνάρτηση επιλέγει τον στόχο από την εξαγωγή δεδομένων που πραγματοποιήθηκε από την συνάρτηση `Extract_Data`. Μετέπειτα, αφού επιλεχτεί ο στόχος θα πρέπει να χωριστούν τα δεδομένα για την εκπαίδευση αλλά και την εγκυρότητα των αλγορίθμων.

```
def train_test_split_df(dataframe, t_size, shuff):
    """
    This function split the data
    -----
    Element: dataframe (Is a dataframe)
    Element: t_size (Is float number <0.2 , 0.3> and split the data)
    Element: shuff (Is Bool, If true the rows is random)
    """

    X = dataframe.iloc[:,1:]
    Y = dataframe.iloc[:,0]

    #global X_train, X_test, y_train, y_test
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=t_size, shuffle=shuff)

    sum_train_test = len(y_train) + len(y_test)
    train_set = 1 - t_size

    fig = plt.figure(figsize=(15,5))
    plt.title('{}'.format(Y.name), color='White', fontsize=20, x=0.19, y=0.85)
    plt.figtext(x=0.168, y=0.7, s='Train {}% of the Data-Set'.format(train_set), color='blue', fontsize=15)
    plt.figtext(x=0.168, y=0.63, s='Test {}% of the Data-Set'.format(t_size), color='orange', fontsize=15)
    plt.plot(np.arange(0,len(y_train),1), y_train, label='Training set', color='blue')
    plt.plot(np.arange(len(y_train), sum_train_test, 1), y_test, label='Test set', color='orange')
    plt.tick_params(axis='x', colors='White')
    plt.tick_params(axis='y', colors='White')
    #plt.grid(True)
    plt.axis('on')
    #plt.legend()

    img_tts = plt.savefig('../Application_Programming_Interface/static/images/img_web/train_test_split_img.png', transparent=True)
    plt.show()
    return X_train, X_test, y_train, y_test
```

Εικόνα 13: Διαχωρισμός πίνακα δεδομένων

Στην παραπάνω συνάρτηση με όνομα `train_test_split_df` γίνεται η επιλογή των δεδομένων που θα πραγματοποιηθεί για να εκπαιδευτεί το μοντέλο (X) αλλά και οι τιμές που πρόκειται να προβλέψει (Y). Επίσης χωρίζονται τα δεδομένα με την συνάρτηση `train_test_split` για να φανεί η εγκυρότητα των μοντέλων.

```
#----- Linear Regression -----#
def Eval_Model_LR(model, X_train, X_test, y_train, y_test):
    """
    This Function return a DataFrame with Real-Value and Prediction values.
    Also print the Model wich used and information about Metrics of the Model and Diagram of the Metrics
    -----
    Element: model (Select the Model)

    --- Train-Test-Split ---
    Element: X_train (X_train)
    Element: X_test (X_test)
    Element: y_train (y_train)
    Element: y_test (y_test)
    """

    #print('\nThe model is: ',model,'\n')

    model.fit(X_train,y_train)

    #Train
    pred_train = model.predict(X_train)
    MSE_train = mean_squared_error(y_train, pred_train)
    RMSE_train = math.sqrt(MSE_train)

    #Test
    pred_test = model.predict(X_test)
    MSE_test = mean_squared_error(y_test, pred_test)
    RMSE_test = math.sqrt(MSE_test)

    fig = plt.figure(figsize=(20,8))
    plt.title('Forecasting of the Test-Set with {}'.format(model), color='Gold', fontsize=20, y=0.93)
    plt.figtext(x=0.335, y=0.79, s='The Metric is Root-Mean-Square-Error', color='White', fontsize=18)
    plt.figtext(x=0.335, y=0.73, s='Train RMSE = {:.2f} --- Real Value'.format(RMSE_train), color='Blue', fontsize=18)
    plt.figtext(x=0.335, y=0.68, s='Train RMSE = {:.2f} --- Prediction' .format(RMSE_test), color='Red', fontsize=18)

    plt.plot(y_test.index, y_test, label='Real Value', color='Blue', linewidth=0.8)
    plt.plot(y_test.index, pred_test, label='Prediction', color='Red', linewidth=1.5)
    plt.tick_params(axis='x', colors='White',direction="in",width=10,labelrotation=-60)
    plt.tick_params(axis='y', colors='White',direction="in",width=10,labelrotation=20)
    plt.axis('on')
    plt.legend()

    #dataframe = pd.DataFrame({'Real_Value': y_test,
    #                          'Forecast': pred_test}, index=y_test.index)

    #img = mpimg.imread()

    img_lr = plt.savefig('../Application_Programming_Interface/static/images/img_web/Model_LR.png',
                        bbox_inches='tight',
                        dpi = 300,
                        transparent=True)

    plt.show()
    #plt.close(fig)#Download Image
    #return dataframe
#----- Linear Regression -----#
```

Εικόνα 14: Επιλογή μοντέλου

Στην παραπάνω συνάρτηση γίνεται η εκπαίδευση του μοντέλου στο σετ εκπαίδευσης και πραγματοποιεί προβλέψεις στο test. Επίσης η μονάδα μέτρησης που χρησιμοποιείται είναι το Root Mean Square Error (RMSE).

3.2 Περιγραφή back-end API συστήματος εφαρμογής

Στο συγκεκριμένο υποκεφάλαιο θα περιγραφούν οι τεχνολογίες που χρησιμοποιήθηκαν για το χτίσιμο του back-end συστήματος, ο τρόπος λειτουργίας του API αλλά και η συνδεσιμότητα που υπάρχει μεταξύ του front-end κομμάτι του συστήματος με το back-end. Το back-end του πληροφοριακού συστήματος για οικονομικά δεδομένα που βασίζεται σε αλγόριθμους μηχανικής μάθησης είναι χτισμένο χρησιμοποιώντας τη γλώσσα προγραμματισμού Python-3. Το back-end είναι υπεύθυνο για την επεξεργασία και την ανάλυση των δεδομένων, την εκπαίδευση των μοντέλων μηχανικής εκμάθησης και την παροχή προβλέψεων στο front-end της εφαρμογής.

Η Python-3 επιλέχθηκε ως η γλώσσα προγραμματισμού για το back-end της εφαρμογής λόγω του τεράστιου οικοσυστήματος βιβλιοθηκών και πλαισίων που είναι κατάλληλα για χειρισμό δεδομένων, μηχανική εκμάθηση και ανάπτυξη ιστού. Το back-end έχει δημιουργηθεί χρησιμοποιώντας πολλές δημοφιλείς βιβλιοθήκες Python-3, όπως Pandas, NumPy και Scikit-learn για αλγόριθμους χειρισμού δεδομένων, ανάλυσης και μηχανικής μάθησης, αντίστοιχα.

Το Pandas είναι μια ισχυρή βιβλιοθήκη για χειρισμό και ανάλυση δεδομένων, χρησιμοποιήθηκε για το χειρισμό των δεδομένων που συλλέγονται από το Binance API, την προεπεξεργασία τους και την προετοιμασία για εκπαίδευση και πρόβλεψη. Το NumPy είναι μια βιβλιοθήκη για μαθηματικούς και επιστημονικούς υπολογισμούς, χρησιμοποιήθηκε για το χειρισμό των συστοιχίων δεδομένων και την εκτέλεση μαθηματικών πράξεων σε αυτούς. Το Scikit-learn είναι μια βιβλιοθήκη μηχανικής μάθησης για την Python-3, που χρησιμοποιήθηκε για την εκπαίδευση και την αξιολόγηση των μοντέλων μηχανικής μάθησης που χρησιμοποιούνται στο σύστημα.

Για να χειριστεί την επικοινωνία μεταξύ του front-end και του back-end, η εφαρμογή χρησιμοποιεί το πλαίσιο web Flask, το οποίο είναι ένα micro web πλαίσιο για την Python-3. Το Flask επιλέχθηκε για την απλότητα και την ευκολία χρήσης του, επιτρέπει την ταχεία ανάπτυξη εφαρμογών web και υποστηρίζει διάφορες λειτουργίες όπως δρομολόγηση, διαχείριση αιτημάτων και απόδοση προτύπων.

Όσον αφορά την επεξεργασία δεδομένων και τη μηχανική χαρακτηριστικών, το back-end χρησιμοποιεί Pandas και NumPy, δύο από τις πιο δημοφιλείς βιβλιοθήκες

χειρισμού δεδομένων για την Python-3. Τα δεδομένα αυξήθηκαν επίσης για να αυξηθεί το μέγεθος του συνόλου δεδομένων και να γίνουν τα μοντέλα πιο ισχυρά.

Εκτός από αυτές τις βιβλιοθήκες, το back-end χρησιμοποιεί επίσης άλλες βιβλιοθήκες και εργαλεία, όπως το Matplotlib για οπτικοποίηση δεδομένων και επίσης χρησιμοποιήθηκε για τη δημιουργία οπτικοποιήσεων των δεδομένων και της απόδοσης του μοντέλου.

3.2.1 Βασική Λειτουργία Back – End Συστήματος

Αρχικά υλοποιήθηκε το front-end κομμάτι του συστήματος για να διαμορφωθούν οι σελίδες και στην συνέχεια μέσω του API καθορίστηκαν οι δρομολογήσεις προκειμένου να πλοηγείται ο χρήστης σε όποια σελίδα θέλει. Η αρχική σελίδα που θα εισέρχεται κατευθείαν ο χρήστης, δηλαδή ο καθορισμός του ριζικού φάκελου και αρχείου διαμορφώνεται με τις εντολές δρομολόγησης `app.route` πάνω από την συνάρτηση `Home` όπου στην συνέχεια θα ανακατευθύνεται μέσα στην σελίδα `home.html` μέσω της εντολής `render_template`.

```
@app.route("/")
@app.route("/Home")
def Home():
    #return render_template('/Home.html')
    return render_template('/Web/Home.html')

@app.route("/Evaluation", methods=['POST', 'GET'])
def Evaluation():
    # #return render_template('/...html')
    if request.method == 'POST':
        targ = request.form['target']
        #df = All_Func.Extract_Data(3)
        All_Func.Select_Target(dataframe=df, name_of_col=targ)
        All_Func.Convert_Type(dataframe=df)
        print('The Target is: ',targ)
        X_train, X_test, y_train, y_test = All_Func.train_test_split_df(dataframe=df, t_size=0.2, shuff=False)
        #All_Func.train_test_split_df(dataframe=df, t_size=0.2, shuff=False)

        # --- Model Img ---#
        All_Func.Eval_Model_LR(model=lr, X_train=X_train, X_test=X_test, y_train=y_train, y_test=y_test)
        All_Func.Eval_Model_DT(model=dt, X_train=X_train, X_test=X_test, y_train=y_train, y_test=y_test)
        All_Func.Eval_Model_GBR(model=gb, X_train=X_train, X_test=X_test, y_train=y_train, y_test=y_test)
        # --- Model Img ---#

        # --- Effect ---#
        All_Func.rmse_effects(model1=lr, model2=dt, model3=gb, X_train=X_train, X_test=X_test, y_train=y_train, y_test=y_test)
        # --- Effect ---#
    return render_template('/Web/evaluation.html')

@app.route("/About")
def About():
    #return render_template('/...html')
    return render_template('/Web/about.html')

@app.route("/Decision_Tree")
def ModelDT():
    #return render_template('/...html')
    return render_template('/Web/decision_tree.html')

@app.route("/Gradient_Boosting_Regressor")
def ModelGBR():
    #return render_template('/...html')
    return render_template('/Web/gradient_boosting_regressor.html')

@app.route("/Linear_Regression")
def ModelLR():
    return render_template('/Web/linear_regression.html')
```

Εικόνα 15: Δομή API

Στη συνέχεια ακολουθεί η δρομολόγηση για το Evaluation όπου μετά μέσα στην συνάρτηση πραγματοποιούνται τα ερωτήματα από τον χρήστη προς το API μέσω της μεθόδου POST. Από τα ερωτήματα επιστρέφονται κάποια δεδομένα από το έγγραφο `evaluation.html` και πραγματοποιούνται οι κατάλληλες συναρτήσεις για την μορφοποίηση και την απεικόνιση των δεδομένων αλλά και των μοντέλων.

Συμπερασματικά, το back-end του συστήματος της ακόλουθης εφαρμογής, έχει δημιουργηθεί χρησιμοποιώντας έναν συνδυασμό δημοφιλών βιβλιοθηκών και πλαισίων Python-3 όπως Pandas, NumPy, Scikit-learn, Flask και Matplotlib, που επιτρέπει τον αποτελεσματικό χειρισμό δεδομένων, ανάλυση, μηχανική εκμάθηση και επικοινωνία με το front-end. Το back-end σχεδιάστηκε για να χειρίζεται και να αναλύει τα δεδομένα, να εκπαιδεύει και να αξιολογεί τα μοντέλα μηχανικής εκμάθησης και να παρέχει τις προβλέψεις στο front-end, όλα με ισχυρό και αποτελεσματικό τρόπο.

3.3 Εγχειρίδιο χρήσης μοντέλων

Το πληροφοριακό σύστημα που αναπτύχθηκε σε αυτή τη διατριβή χρησιμοποιεί πέντε μοντέλα μηχανικής μάθησης για να κάνει προβλέψεις για οικονομικά δεδομένα για τα κρυπτονομίσματα Bitcoin (BTC), Ethereum (ETH) και Binance (BNB). Αυτά τα μοντέλα περιλαμβάνουν:

- **Linear Regression:** Αυτό είναι ένα παραδοσιακό και απλό μοντέλο μηχανικής μάθησης που μοντελοποιεί τη σχέση μεταξύ μιας εξαρτημένης μεταβλητής και μιας ή περισσότερων ανεξάρτητων μεταβλητών. Χρησιμοποιείται για την πρόβλεψη της τιμής της εξαρτημένης μεταβλητής με βάση την τιμή των ανεξάρτητων μεταβλητών. Το μοντέλο γραμμικής παλινδρόμησης που χρησιμοποιείται σε αυτό το πληροφοριακό σύστημα εκπαιδεύεται χρησιμοποιώντας τα ιστορικά οικονομικά δεδομένα για BTC, ETH και BNB. Η διαδικασία εκπαίδευσης περιλαμβάνει τη χρήση της κλάσης `LinearRegression()` που παρέχεται από τη βιβλιοθήκη `scikit-learn`. Στη διαδικασία εκπαίδευσης, τα δεδομένα αρχικά χωρίζονται σε σύνολα εκπαίδευσης και δοκιμών, με το 80% των δεδομένων να χρησιμοποιούνται για εκπαίδευση και το 20% των δεδομένων να χρησιμοποιούνται για δοκιμές. Στη συνέχεια, το μοντέλο

εκπαιδεύεται χρησιμοποιώντας τη συνάρτηση `fit()` που παρέχεται από τη βιβλιοθήκη `scikit-learn`, μεταβιβάζοντας τα δεδομένα εκπαίδευσης ως είσοδο. Οι παράμετροι που χρησιμοποιούνται στη διαδικασία εκπαίδευσης είναι οι προεπιλεγμένες παράμετροι που παρέχονται από τη βιβλιοθήκη `scikit-learn`.

- **Decision Tree:** Το δέντρο αποφάσεων είναι ένας εποπτευόμενος αλγόριθμος μάθησης που χρησιμοποιείται τόσο για εργασίες ταξινόμησης όσο και για εργασίες παλινδρόμησης. Δημιουργεί ένα μοντέλο που μοιάζει με δέντρο για αποφάσεις και τις πιθανές συνέπειές τους, συμπεριλαμβανομένων των αποτελεσμάτων τυχαίων συμβάντων, του κόστους πόρων και της χρησιμότητας. Το μοντέλο δέντρου αποφάσεων που χρησιμοποιείται σε αυτό το σύστημα πληροφοριών εκπαιδεύεται επίσης χρησιμοποιώντας τα ιστορικά οικονομικά δεδομένα για BTC, ETH και BNB. Η διαδικασία εκπαίδευσης περιλαμβάνει τη χρήση της κλάσης `DecisionTreeRegressor()` που παρέχεται από τη βιβλιοθήκη `scikit-learn`. Στη διαδικασία εκπαίδευσης, τα δεδομένα αρχικά χωρίζονται σε σύνολα εκπαίδευσης και δοκιμών, με το 80% των δεδομένων να χρησιμοποιούνται για εκπαίδευση και το 20% των δεδομένων να χρησιμοποιούνται για δοκιμές. Στη συνέχεια, το μοντέλο εκπαιδεύεται χρησιμοποιώντας τη συνάρτηση `fit()` που παρέχεται από τη βιβλιοθήκη `scikit-learn`, μεταβιβάζοντας τα δεδομένα εκπαίδευσης ως είσοδο. Οι παράμετροι που χρησιμοποιούνται στη διαδικασία εκπαίδευσης είναι οι προεπιλεγμένες παράμετροι που παρέχονται από τη βιβλιοθήκη `scikit-learn`.
- **Gradient Boosting Regressor:** Η Gradient Boosting είναι μια μέθοδος εκμάθησης συνόλου για προβλήματα ταξινόμησης και παλινδρόμησης, η οποία παράγει ένα μοντέλο πρόβλεψης με τη μορφή ενός συνόλου αδύναμων μοντέλων πρόβλεψης. Χρησιμοποιεί αλγόριθμο `gradient descent` για να ελαχιστοποιήσει την απώλεια κατά την προσθήκη νέων μοντέλων. Αυτό το μοντέλο εκπαιδεύεται επίσης χρησιμοποιώντας τα ιστορικά οικονομικά δεδομένα για BTC, ETH και BNB. Η διαδικασία εκπαίδευσης περιλαμβάνει τη χρήση της κλάσης `GradientBoostingRegressor()` που παρέχεται από τη βιβλιοθήκη `scikit-learn`. Στη διαδικασία εκπαίδευσης, τα δεδομένα αρχικά χωρίζονται σε σύνολα εκπαίδευσης και δοκιμών, με το 80% των δεδομένων να χρησιμοποιούνται για εκπαίδευση και το 20% των δεδομένων να χρησιμοποιούνται για δοκιμές. Στη συνέχεια, το μοντέλο

εκπαιδύεται χρησιμοποιώντας τη συνάρτηση `fit()` που παρέχεται από τη βιβλιοθήκη `scikit-learn`, μεταβιβάζοντας τα δεδομένα εκπαίδευσης ως είσοδο. Οι παράμετροι που χρησιμοποιούνται στη διαδικασία εκπαίδευσης είναι οι προεπιλεγμένες παράμετροι που παρέχονται από τη βιβλιοθήκη `scikit-learn`.

- **Random Forest:** Το Random Forest είναι ένας εποπτευόμενος αλγόριθμος εκμάθησης που χρησιμοποιείται τόσο για εργασίες ταξινόμησης όσο και για εργασίες παλινδρόμησης. Είναι μια μέθοδος εκμάθησης συνόλου για ταξινόμηση, παλινδρόμηση και άλλες εργασίες, που λειτουργούν κατασκευάζοντας ένα πλήθος δέντρων αποφάσεων κατά το χρόνο εκπαίδευσης και βγάζοντας την κλάση που είναι ο τρόπος των κλάσεων (ταξινόμηση) ή η μέση πρόβλεψη (παλίνδρομος) των μεμονωμένων δέντρων. . Αυτό το μοντέλο εκπαιδύεται επίσης χρησιμοποιώντας τα ιστορικά οικονομικά δεδομένα για BTC, ETH και BNB. Η διαδικασία εκπαίδευσης περιλαμβάνει τη χρήση της κλάσης `RandomForestRegressor()` που παρέχεται από τη βιβλιοθήκη `scikit-learn`. Στη διαδικασία εκπαίδευσης, τα δεδομένα αρχικά χωρίζονται σε σύνολα εκπαίδευσης και δοκιμών, με το 80% των δεδομένων να χρησιμοποιούνται για εκπαίδευση και το 20% των δεδομένων να χρησιμοποιούνται για δοκιμές. Στη συνέχεια, το μοντέλο εκπαιδύεται χρησιμοποιώντας τη συνάρτηση `fit()` που παρέχεται από τη βιβλιοθήκη `scikit-learn`, μεταβιβάζοντας τα δεδομένα εκπαίδευσης ως είσοδο. Οι παράμετροι που χρησιμοποιούνται στη διαδικασία εκπαίδευσης είναι οι προεπιλεγμένες παράμετροι που παρέχονται από τη βιβλιοθήκη `scikit-learn`.
- **LSTM (Long Short-Term Memory):** Η μακροχρόνια βραχυπρόθεσμη μνήμη (LSTM) είναι μια παραλλαγή του αλγόριθμου επαναλαμβανόμενου νευρωνικού δικτύου (RNN), ο οποίος έχει σχεδιαστεί ειδικά για να εμποδίζει την έξοδο του νευρικού δικτύου για μια δεδομένη είσοδο να εξαρτάται μόνο από έναν μικρό αριθμό προηγούμενων επαναλήψεων. Χρησιμοποιείται για την πρόβλεψη δεδομένων χρονοσειρών, όπως οι τιμές των κρυπτονομισμάτων. Το μοντέλο LSTM που χρησιμοποιείται σε αυτό το σύστημα πληροφοριών εκπαιδύεται χρησιμοποιώντας τα ιστορικά οικονομικά δεδομένα για BTC, ETH και BNB. Η εκπαιδευτική διαδικασία περιλαμβάνει τη χρήση της βιβλιοθήκης `Tensorflow`. Στη διαδικασία εκπαίδευσης, τα δεδομένα αρχικά χωρίζονται σε σύνολα εκπαίδευσης και δοκιμών, με το 80% των δεδομένων να χρησιμοποιούνται για εκπαίδευση και το 20% των δεδομένων να

χρησιμοποιούνται για δοκιμές. Στη συνέχεια, το μοντέλο εκπαιδεύεται κατασκευάζοντας την αρχιτεκτονική του μοντέλου και εκπαιδεύοντας το μοντέλο με τα δεδομένα εκπαίδευσης και τις εκπαιδύσιμες παραμέτρους. Οι παράμετροι που χρησιμοποιούνται στη διαδικασία εκπαίδευσης είναι οι προεπιλεγμένες παράμετροι που παρέχονται από τη βιβλιοθήκη Tensorflow.

Στην αρχή τα μοντέλα εκπαιδεύονται σε ένα σετ δοκιμών 80% εκπαίδευση και 20% επικύρωση ώστε να γίνουν προβλέψεις για τις επόμενες 7 ημέρες. Αργότερα τα μοντέλα επανεκπαιδεύονται ξανά στο σετ δοκιμών έως το σετ δοκιμών που προβλέφθηκε και αφαιρεί 7 ημέρες από το σετ εκπαίδευσης, δηλαδή τα μοντέλα εκπαιδεύονται σε σταθερό σετ εκπαίδευσης. Τα διάφορα μοντέλα που χρησιμοποιούνται για την εκπαίδευση των δεδομένων έχουν σαν μέτρο σύγκρισης το RMSE όπου όποιο μοντέλο έχει το χαμηλότερο κάνει και την πρόβλεψη για τα κρυπτονομίσματα.

Αξίζει να σημειωθεί ότι η διαδικασία πρόβλεψης βασίζεται στα προηγούμενα δεδομένα, επομένως τα αποτελέσματα μπορούν να αλλάξουν ανάλογα με τον χρόνο των προβλέψεων και τα αποτελέσματα δεν είναι πάντα ακριβή. Ωστόσο, το σύστημα παρέχει μια γενική άποψη της τάσης της αγοράς, ο χρήστης μπορεί να το χρησιμοποιήσει για να λάβει μια ενημερωμένη απόφαση.

3.4 Περιγραφή front-end Web συστήματος

Το μπροστινό μέρος μιας εφαρμογής αναφέρεται στις πτυχές της διεπαφής χρήστη (UI) και της εμπειρίας χρήστη (UX) της εφαρμογής. Είναι το μέρος της εφαρμογής με το οποίο αλληλοεπιδρούν οι χρήστες και είναι υπεύθυνο για την εμφάνιση, την αίσθηση και τη λειτουργία της εφαρμογής. Η διεπαφή συνήθως υλοποιείται χρησιμοποιώντας έναν συνδυασμό HTML, CSS και JavaScript και έχει σχεδιαστεί για να παρουσιάζει πληροφορίες στον χρήστη και να δέχεται πληροφορίες από τον χρήστη.

Ο σχεδιασμός και η υλοποίηση του μπροστινού άκρου είναι κρίσιμες για την επιτυχία μιας εφαρμογής, καθώς είναι το κύριο σημείο επαφής μεταξύ του χρήστη και της εφαρμογής. Ένα καλά σχεδιασμένο μπροστινό μέρος θα είναι διαισθητικό, εύκολο στη χρήση και οπτικά ελκυστικό, το οποίο μπορεί να βελτιώσει σημαντικά την εμπειρία του

χρήστη και τη συνολική ικανοποίηση του χρήστη από την εφαρμογή. Από την άλλη πλευρά, μια κακώς σχεδιασμένη διεπαφή μπορεί να προκαλέσει σύγχυση, απογοήτευση και τελικά να απομακρύνει τους χρήστες από την εφαρμογή.

Σε αυτό το υποκεφάλαιο, θα διερευνηθούν οι βασικές εκτιμήσεις και τις βέλτιστες πρακτικές για το σχεδιασμό και την υλοποίηση της διεπαφής μιας εφαρμογής. Θα συζητηθεί ο ρόλος των HTML, CSS και JavaScript στο front-end, καθώς και τη σημασία του σχεδιασμού της εμπειρίας χρήστη και των εργαλείων και τεχνολογιών που διατίθενται για τη δημιουργία και τη δοκιμή της διεπαφής.

3.4.1 Το Front-End διαχρονικά

Η ανάπτυξη front-end έχει τις ρίζες της στις πρώτες μέρες του Διαδικτύου, όταν οι ιστότοποι αποτελούνταν κυρίως από στατικές σελίδες HTML. Με την πάροδο του χρόνου, το πεδίο έχει εξελιχθεί ώστε να περιλαμβάνει πιο προηγμένες τεχνολογίες, όπως CSS και JavaScript, που επιτρέπουν μεγαλύτερο έλεγχο στην εμφάνιση και τη συμπεριφορά των ιστοσελίδων.

Σήμερα, η ανάπτυξη front-end περιλαμβάνει ένα ευρύ φάσμα τεχνολογιών και πρακτικών, συμπεριλαμβανομένου του responsive design, της ανάπτυξης για κινητά και του design εμπειρίας χρήστη (UX). Καθώς το Διαδίκτυο και οι εφαρμογές που βασίζονται στον ιστό έχουν γίνει κεντρικά στη σύγχρονη ζωή, η ανάπτυξη του front-end έχει αυξηθεί σε σημασία και πολυπλοκότητα.

3.4.2 Εργαλεία και τεχνολογίες που χρησιμοποιούνται στην ανάπτυξη Front-End

Οι προγραμματιστές front-end χρησιμοποιούν μια ποικιλία εργαλείων και τεχνολογιών για να σχεδιάσουν και να δημιουργήσουν ιστότοπους και εφαρμογές. Η HTML (HyperText Markup Language), η CSS (Cascading Style Sheets) και η JavaScript είναι οι κύριες τεχνολογίες που χρησιμοποιούνται για τη δημιουργία εφαρμογών web στο μπροστινό μέρος. Αυτές οι τεχνολογίες χρησιμοποιήθηκαν στην εφαρμογή που υλοποιήθηκε αλλά και γενικότερα είναι απαραίτητες για τη δόμηση και τη μορφοποίηση περιεχομένου, το στυλ ιστοσελίδων και την προσθήκη διαδραστικότητας και δυναμικής συμπεριφοράς σε εφαρμογές Ιστού.

- **HTML(HyperText Markup Language)**

Η HTML (HyperText Markup Language) είναι η τυπική γλώσσα σήμανσης για τη δημιουργία ιστοσελίδων και άλλων τύπων εγγράφων που μπορούν να εμφανιστούν σε ένα πρόγραμμα περιήγησης Ιστού. Χρησιμοποιείται για τη δομή και την οργάνωση περιεχομένου στον Ιστό, συμπεριλαμβανομένων κειμένου, εικόνων και άλλων μέσων. Η HTML αποτελείται από στοιχεία, τα οποία αντιπροσωπεύονται από ετικέτες, και χαρακτηριστικά, τα οποία παρέχουν πρόσθετες πληροφορίες για το στοιχείο. Για παράδειγμα, το στοιχείο `<p>` αντιπροσωπεύει μια παράγραφο και το χαρακτηριστικό `class` μπορεί να χρησιμοποιηθεί για να εκχωρηθεί ένα όνομα κλάσης στο στοιχείο για λόγους στυλ.

- **CSS (Cascading Style Sheets)**

Το CSS (Cascading Style Sheets) είναι μια γλώσσα φύλλου στυλ που χρησιμοποιείται για την περιγραφή της εμφάνισης και της μορφοποίησης ενός εγγράφου γραμμένου σε HTML. Το CSS επιτρέπει στους προγραμματιστές να ελέγχουν την εμφάνιση μιας ιστοσελίδας ορίζοντας στυλ για στοιχεία όπως γραμματοσειρές, χρώματα και διάταξη. Αυτά τα στυλ μπορούν να εφαρμοστούν σε μεμονωμένα στοιχεία ή ομάδες στοιχείων χρησιμοποιώντας επιλογείς(selectors), τα οποία είναι μοτίβα που ταιριάζουν με στοιχεία με βάση το όνομα, την κλάση ή άλλα χαρακτηριστικά τους.

- **Javascript**

Η JavaScript είναι μια γλώσσα προγραμματισμού που χρησιμοποιείται για να προσθέσει διαδραστικότητα και δυναμική συμπεριφορά σε ιστοσελίδες. Χρησιμοποιείται συνήθως σε συνδυασμό με HTML και CSS για τη δημιουργία διαδραστικών εφαρμογών Ιστού. Ο κώδικας JavaScript εκτελείται στο πρόγραμμα περιήγησης, πράγμα που σημαίνει ότι μπορεί να χειριστεί το DOM (Document Object Model) μιας ιστοσελίδας και να ανταποκρίνεται σε συμβάντα χρήστη, όπως κλικ, τοποθέτηση του δείκτη του ποντικιού και υποβολές φορμών. Η JavaScript είναι μια δυναμική, αντικειμενοστραφή γλώσσα με σύνταξη παρόμοια με αρκετές γλώσσες προγραμματισμού. Είναι μια ισχυρή και ευέλικτη γλώσσα που μπορεί να χρησιμοποιηθεί για τη δημιουργία ενός ευρέος φάσματος εφαρμογών, από απλά σενάρια έως πολύπλοκες εφαρμογές Ιστού.

- **Άλλα εργαλεία:**

Οι προγραμματιστές front-end μπορούν επίσης να χρησιμοποιήσουν άλλα εργαλεία, όπως προγράμματα επεξεργασίας κειμένου – υπερκειμένου, προγράμματα σχεδίασης της σελίδας κλπ.

3.4.3 Αρχές σχεδίασης και βέλτιστες πρακτικές

Οι προγραμματιστές front-end ακολουθούν ένα σύνολο αρχών σχεδίασης και βέλτιστων πρακτικών για να διασφαλίσουν ότι οι ιστότοποι και οι εφαρμογές που δημιουργούν είναι εύχρηστοι, προσβάσιμοι και οπτικά ελκυστικοί. Μερικά από τα βασικά ζητήματα στην ανάπτυξη front-end περιλαμβάνουν:

Ευχρηστία: Οι ιστότοποι και οι εφαρμογές πρέπει να είναι εύχρηστες, με σαφή και διαισθητική πλοήγηση, συνεπή διάταξη και κατάλληλη χρήση χρώματος και τυπογραφίας.

Προσβασιμότητα: Οι ιστότοποι και οι εφαρμογές θα πρέπει να είναι προσβάσιμες σε χρήστες με αναπηρίες, συμπεριλαμβανομένων εκείνων που είναι τυφλοί, κωφοί ή άτομα με κινητικά προβλήματα. Αυτό μπορεί να επιτευχθεί με τη χρήση κατάλληλων ετικετών HTML, εναλλακτικού κειμένου για εικόνες και άλλων τεχνικών.

Responsive design: Οι ιστότοποι και οι εφαρμογές θα πρέπει να σχεδιάζονται για να λειτουργούν καλά σε μια ποικιλία συσκευών, συμπεριλαμβανομένων των επιτραπέζιων υπολογιστών, των tablet και των smartphone. Αυτό απαιτεί τη χρήση τεχνικών σχεδίασης με απόκριση, όπως ευέλικτες διατάξεις και ερωτήματα πολυμέσων.

3.4.4 Συμπέρασμα

Η ανάπτυξη front-end είναι ένας δυναμικός και συνεχώς εξελισσόμενος τομέας, με νέα εργαλεία και τεχνολογίες να αναδύονται συνεχώς. Κατανοώντας την ιστορία, την τρέχουσα κατάσταση και τη μελλοντική κατεύθυνση της ανάπτυξης του front-end, καθώς και των εργαλείων και τεχνολογιών που χρησιμοποιούνται στο πεδίο, μπορούμε να κατανοήσουμε καλύτερα τον ρόλο που παίζει η ανάπτυξη front-end στη δημιουργία σύγχρονων ιστοτόπων και εφαρμογών.

3.5 Δεδομένα

Τα δεδομένα που χρησιμοποιούνται σε αυτό το πληροφοριακό σύστημα είναι ιστορικά οικονομικά δεδομένα για τα κρυπτονομίσματα Bitcoin (BTC), Ethereum (ETH) και Binance(BNB). Τα δεδομένα περιλαμβάνουν διάφορες πληροφορίες όπως η τιμή κλεισίματος, η υψηλότερη τιμή, η χαμηλότερη τιμή, η τιμή ανοίγματος και ο όγκος συναλλαγών κάθε κρυπτονομίσματος. Αυτά τα δεδομένα συλλέχθηκαν από το Binance API, το οποίο είναι μια πλατφόρμα για την πρόσβαση σε οικονομικά δεδομένα διαφορετικών κρυπτονομισμάτων.

Τα δεδομένα συλλέχθηκαν για μια περίοδο 3 ετών, από τον Ιανουάριο του 2020 έως και την τρέχουσα περίοδο. Αυτό το χρονικό πλαίσιο επιλέχθηκε για να διασφαλιστεί ότι τα μοντέλα έχουν αρκετά δεδομένα για να μάθουν και να κάνουν ακριβείς προβλέψεις. Τα δεδομένα συλλέγονταν σε ημερήσια συχνότητα, πράγμα που σημαίνει ότι υπάρχουν περίπου 1000 σημεία δεδομένων για κάθε κρυπτονόμισμα.

Μόλις συλλέχθηκαν τα δεδομένα, φορτώθηκαν σε ένα Pandas DataFrame για περαιτέρω προεπεξεργασία και ανάλυση. Ένα δείγμα των δεδομένων που χρησιμοποιήθηκαν για την εκπαίδευση των μοντέλων φαίνεται στον παρακάτω πίνακα:

Open time	BTCBUSD_Open	BTCBUSD_High	BTCBUSD_Low	BTCBUSD_Close	BTCBUSD_Volume	ETHBUSD_Open	ETHBUSD_High	ETHBUSD_Low	ETHBUSD_Clo
2020-01-01	7169.12	7231.86	7151.58	7176.58	93.548672	128.54	132.51	128.43	130.2
2020-01-02	7177.13	7185.03	6900.00	6945.33	311.705653	129.88	129.88	126.01	126.1
2020-01-03	6944.41	7393.50	6853.79	7329.20	655.633576	127.02	134.63	125.70	133.2
2020-01-04	7330.91	7388.00	7263.55	7346.20	299.060756	134.20	135.52	132.38	134.2
2020-01-05	7346.20	7486.66	7315.00	7351.42	363.286518	134.26	137.99	134.26	134.8
2020-01-06	7349.84	7794.98	7343.29	7755.10	556.563567	135.02	144.23	134.89	143.8
2020-01-07	7757.48	8204.71	7730.00	8154.74	1019.907075	144.35	145.08	138.97	142.8

Εικόνα 16: Σετ Δεδομένων

4. Διαδικτυακή Εφαρμογή

Στο παρόν κεφάλαιο θα παρουσιαστεί και θα αναλυθεί η διαδικτυακή εφαρμογή που αναπτύχθηκε στα πλαίσια της πτυχιακής εργασίας η οποία παρέχει τις πληροφορίες πρόβλεψης για κρυπτονομίσματα. Η διαδικτυακή εφαρμογή αποτελείται από πολλές σελίδες, καθεμία με συγκεκριμένο σκοπό και λειτουργία:

- **Αρχική Σελίδα(home.html)**

Στην πρώτη σελίδα που βλέπουν οι χρήστες όταν επισκέπτονται τον ιστότοπο, εμφανίζει τις τρέχουσες τιμές BTC, ETH και BNB όπου περιλαμβάνει ένα σενάριο JavaScript που ενημερώνει τις τιμές σε πραγματικό χρόνο. Αυτή η σελίδα χρησιμεύει ως εισαγωγή στον ιστότοπο και παρέχει στους χρήστες μια γρήγορη επισκόπηση των τρεχουσών τιμών κρυπτονομισμάτων. Από την αρχική σελίδα, οι χρήστες μπορούν είτε να συνδεθούν (εάν είναι εγγεγραμμένοι χρήστες) είτε να προχωρήσουν στη σελίδα πρόβλεψης (εάν δεν είναι εγγεγραμμένοι χρήστες).

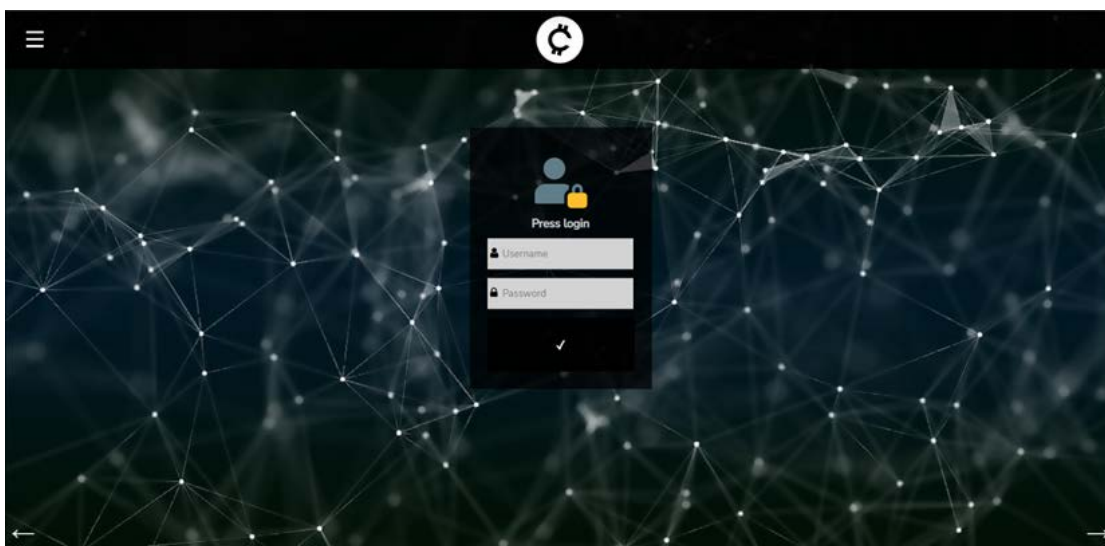


Εικόνα 17: Αρχική Σελίδα εφαρμογής

- **Σελίδα Σύνδεσης(login.html)**

Η σελίδα σύνδεσης είναι προσβάσιμη μόνο σε εγγεγραμμένους χρήστες της εφαρμογής. Περιλαμβάνει μια φόρμα σύνδεσης που επικυρώνεται με χρήση JavaScript για να διασφαλιστεί ότι το όνομα χρήστη και ο κωδικός πρόσβασης που εισήγατε είναι σωστά. Αυτή η σελίδα επιτρέπει στους εγγεγραμμένους χρήστες να

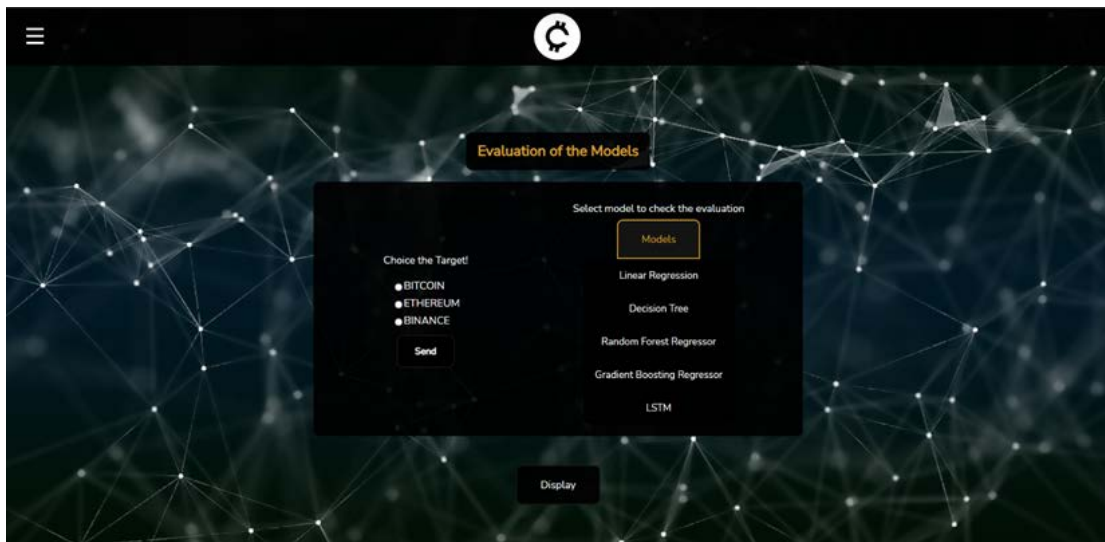
έχουν πρόσβαση στις πληροφορίες πρόβλεψης της εφαρμογής κάνοντας είσοδο με τα μοναδικά τους διαπιστευτήρια. Εάν η σύνδεση είναι επιτυχής, ο χρήστης ανακατευθύνεται στη σελίδα αξιολόγησης. Στην παρούσα φάση ο δυνατότητα πρόσβασης είναι μόνο για τους διαχειριστές της εφαρμογής έχοντας ορίσει σαν Username: Kastoria και Password: 25692907



Εικόνα 18: Σελίδα Εισόδου στο σύστημα για εγγεγραμμένους

- **Σελίδα Αξιολόγησης(evaluation.html)**

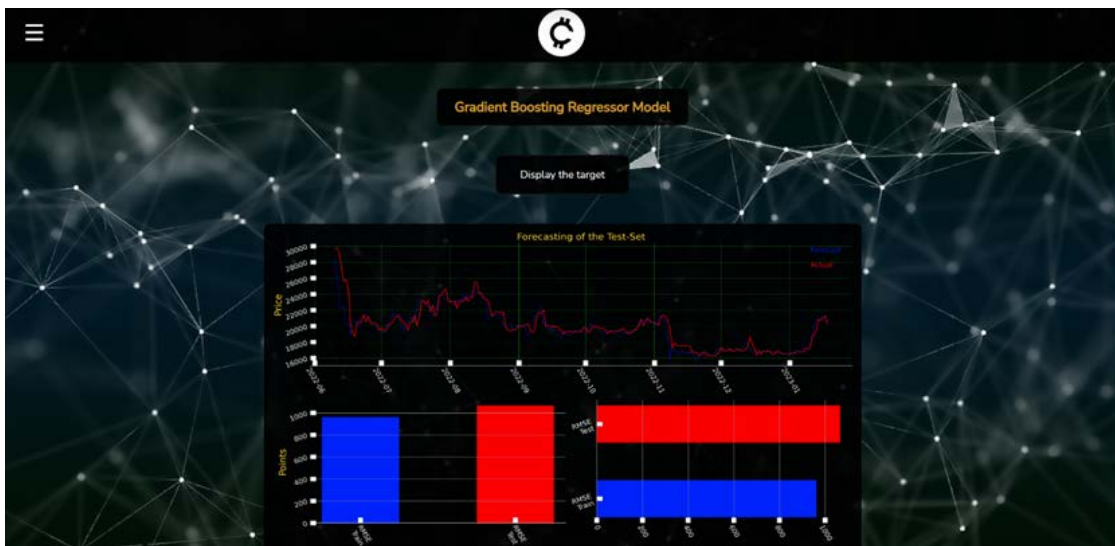
Όταν συνδεθεί στη σελίδα σύνδεσης, μεταφέρεται στην σελίδα αξιολόγησης όπου μπορούν να επιλέξουν το κρυπτονόμισμα αλλά και το μοντέλο για το οποίο θέλουν να δουν περισσότερες πληροφορίες για την επικύρωσή τους και ποιο δίνει τις καλύτερες αποδόσεις. Ο χρήστης μπορεί να επιλέξει από τα κρυπτονομίσματα BTC, ETH και BNB και να υποβάλει τη φόρμα, η οποία στέλνει τα δεδομένα στο backend Python-3 και Flask για επεξεργασία. Στη συνέχεια, το backend δημιουργεί την πρόβλεψη με βάση το επιλεγμένο κρυπτονόμισμα και το επιλεγμένο μοντέλο και ο χρήστης ανακατευθύνεται σε μια σελίδα με επιλογές για προβολή των προβλέψεων χρησιμοποιώντας διαφορετικά μοντέλα μηχανικής εκμάθησης (Linear Regression, Decision Tree, Gradient Boosting Regressor, Random Forest, LSTM). Αυτή η σελίδα επιτρέπει στους εγγεγραμμένους χρήστες να δουν και να επιλέξουν το μοντέλο μηχανικής εκμάθησης που προτιμούν.



Εικόνα 19: Σελίδα Αξιολόγησης και επιλογής κρυπτονομίσματος και μοντέλου

- **Models (Gradient Boosting Regressor.html)**

Σε αυτή τη σελίδα αποτυπώνονται οι πληροφορίες του Gradient Boosting Regressor. Στο πρώτο σχεδιάγραμμα απεικονίζονται οι πραγματικές τιμές και οι προβλεπόμενες τιμές και στο δεύτερο σχεδιάγραμμα απεικονίζονται οι μονάδες μέτρησης που έχει πέσει εκτός το μοντέλο με βάση το RMSE. Εξίσου ο χρήστης μπορεί να περιηγηθεί και σε άλλα μοντέλα και να ελέγξει την εγκυρότητα των μοντέλων.



Εικόνα 20: Σελίδα Μοντέλου και τα αποτελέσματά του

- **Σελίδα Πρόβλεψης(forecasting.html)**

Η σελίδα πρόβλεψης είναι προσβάσιμη σε όλους τους χρήστες του ιστότοπου, ανεξάρτητα από το αν είναι εγγεγραμμένοι ή όχι. Παρέχει εις βάθος προβλέψεις για

κρυπτονομίσματα BTC, ETH και BNB. Αυτή η σελίδα επιτρέπει στους μη εγγεγραμμένους χρήστες να έχουν πρόσβαση σε πληροφορίες προβλέψεων και παρέχει μια γενική επισκόπηση των προβλέψεων για τα επιλεγμένα κρυπτονομίσματα. Τέλος απεικονίζεται ένα σχεδιάγραμμα όπου παρέχει τις τιμές των τριών κρυπτονομισμάτων από το 2020 έως τώρα.



Εικόνα 21: Σελίδα πρόβλεψης και τα αποτελέσματα του για τα κρυπτονομίσματα

- **Σελίδα Σχετικά (about.html)**

Η συγκεκριμένη σελίδα παρέχει πληροφορίες σχετικά με τους προγραμματιστές που υλοποίησαν την εφαρμογή. Περιλαμβάνει λεπτομέρειες όπως τα ονόματα, και τον σκοπό της διαδικτυακής εφαρμογής. Αυτή η σελίδα παρέχει στους χρήστες περισσότερες πληροφορίες σχετικά με την ομάδα πίσω από τον ιστότοπο και τους επιτρέπει να μάθουν περισσότερα για τα άτομα που ανέπτυξαν την εφαρμογή.



Εικόνα 22: Σελίδα Σχετικά με πληροφορίες σχετικά με τους δημιουργούς της εφαρμογής

5. Αξιολόγηση

Στο στάδιο αξιολόγησης αυτού του πληροφοριακού συστήματος, αναλύθηκαν και αξιολογήθηκαν τα αποτελέσματα των προβλέψεων που έγιναν από τους αλγόριθμους μηχανικής μάθησης. Για την παρουσίαση των αποτελεσμάτων χρησιμοποιήθηκε ένας αριθμός γραφικών αναπαραστάσεων, συμπεριλαμβανομένων γραμμικών γραφημάτων και μπάρες. Αυτές οι γραφικές αναπαραστάσεις παρέχουν έναν σαφή και διαισθητικό τρόπο κατανόησης των αποτελεσμάτων των προβλέψεων και σύγκρισης της απόδοσης των διαφορετικών αλγορίθμων. Επιπλέον, για να αξιολογήσουμε την απόδοση των μοντέλων χρησιμοποιήθηκε η μέτρηση της ρίζας του μέσου τετραγωνικού σφάλματος (RMSE), για να αξιολογήσουμε την ακρίβεια της πρόβλεψης και να συγκρίνουμε την απόδοση των διαφορετικών αλγορίθμων. Η παραπάνω εφαρμογή υλοποιήθηκε και έτρεξε σε έναν υπολογιστή με τα εξής χαρακτηριστικά:

Πίνακας 7: Χαρακτηριστικά υπολογιστή που έτρεξε η εφαρμογή

ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ (OS)	Windows 11 Home 64-bit
ΕΠΕΞΕΡΓΑΣΤΗΣ (CPU)	AMD Ryzen 7 5800H, 3.2GHz
ΜΝΗΜΗ RAM	15GB Dual-Channel DDR4, 1575MHz, 22-22-22-52
ΜΗΤΡΙΚΗ ΠΛΑΚΕΤΑ (MOTHERBOARD)	Alienware 0F8CRX (FP6)
ΚΑΡΤΑ ΓΡΑΦΙΚΩΝ (GPU)	NVIDIA GeForce GTX 3060 LAPTOP GPU AMD RADEON GRAPHICS 512MB ATI
ΣΚΛΗΡΟΣ ΔΙΣΚΟΣ (SSD)	1TB

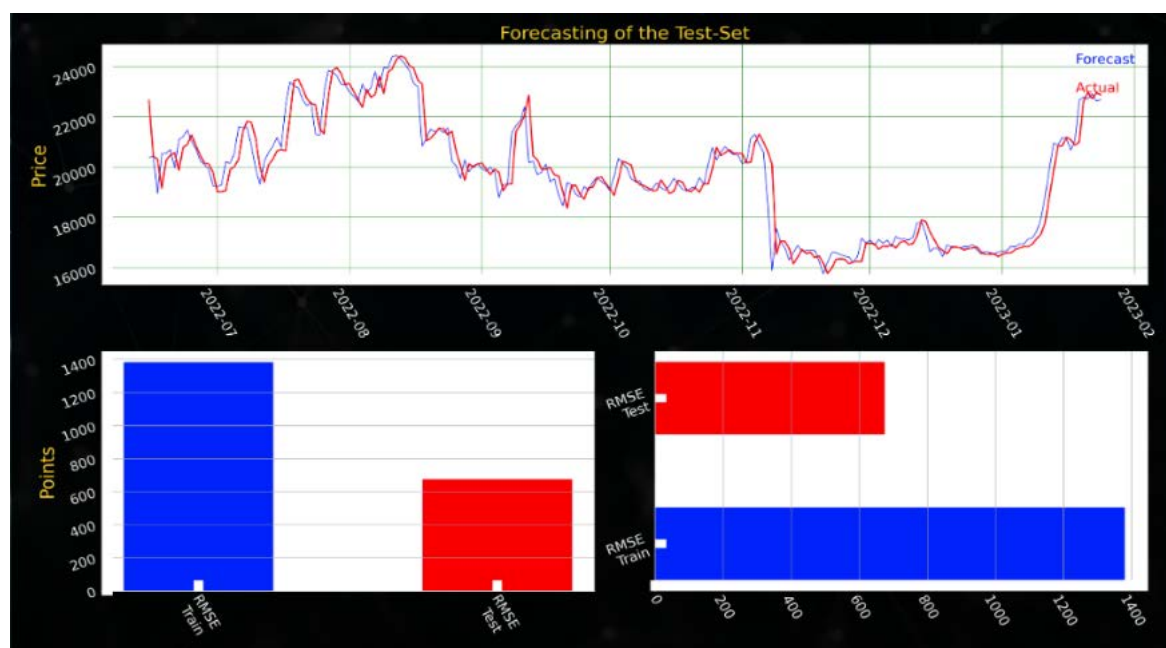
Σχετικά με την προεπεξεργασία (pre-processing) των δεδομένων: Η εξαγωγή των δεδομένων έγινε από το API της Binance και γίνεται η εξαγωγή των κρυπτονομισμάτων του Bitcoin, Ethereum και BNB. Σε αυτά τα κρυπτονομίσματα συμπεριλαμβάνονται χαρακτηριστικά όπως Open time, Open, High, Low, Close, Volume, Close time, Quote asset volume, Number of trades, TB Base Vol, TB Quot Vol και Ignore. Το σεντ δεδομένων περιλαμβάνει μερικές από τις παραπάνω στήλες οι οποίες είναι οι Open time, Open, High, Low, Close, Volume. Τα δεδομένα που εξάγονται αφορούν το χρονικό διάστημα από την 01/01/2020 έως και σήμερα. Αφού έγινε η εξαγωγή των δεδομένων, πραγματοποιήθηκε προ-επεξεργασία των δεδομένων. Αρχικά, χρησιμοποιήθηκε μία

συνάρτηση για να ελέγξει τον τύπο των δεδομένων της κάθε στήλης και αν χρειαστεί να τις μετατρέψει σε δεκαδικό αριθμό. Στη συνέχεια, χρησιμοποιήθηκε μία δεύτερη συνάρτηση προκειμένου εάν κάποια γραμμή μιας στήλης έχει κενή τιμή, να πραγματοποιηθεί γέμισμα προς τα εμπρός (forward fill).

Πίνακας 8: Χαρακτηριστικά της εξαγωγής των δεδομένων

Open time	Ημερομηνία του κρυπτονομίσματος
Open	Η τιμή που άνοιξε το κάθε κρυπτονόμισμα
High	Η υψηλότερη τιμή από το κάθε κρυπτονόμισμα
Low	Η χαμηλότερη τιμή από το κάθε κρυπτονόμισμα
Close	Η τιμή που έκλεισε το κάθε κρυπτονόμισμα
Volume	Ο όγκος συναλλαγών του κάθε κρυπτονομίσματος

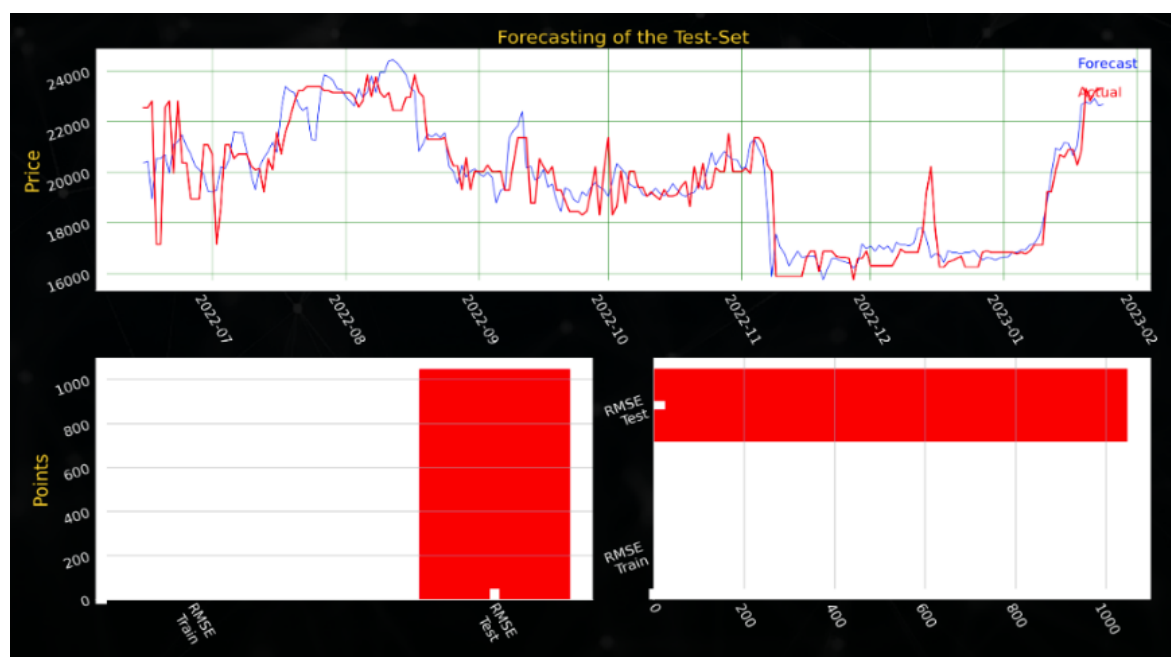
Στα δεδομένα εφαρμόστηκε ο αλγόριθμος Γραμμικής Παλινδρόμησης (**Linear Regression**) και τα αποτελέσματα συγκρίθηκαν με τα πραγματικά δεδομένα. Το μήκος από το σετ δεδομένων είναι από τις 01/01/2020 έως την τρέχουσα περίοδο με στόχο την πρόβλεψη του **Bitcoin**. Ο αλγόριθμος εκπαιδεύτηκε στο 80% του σετ δεδομένων, έπειτα προβλέπει τις επόμενες 7 ημέρες, αφαιρεί 7 ημέρες από την αρχή του σετ δεδομένων ώστε να επανεκπαιδεύεται στο 80% στο σετ δεδομένων. Αυτή η διαδικασία επαναλαμβάνεται συνεχώς μέχρι την τρέχουσα χρονική στιγμή.



Εικόνα 23: Σχεδιαγράμματα αποτελεσμάτων Linear Regression

Ο συγκεκριμένος αλγόριθμος έδωσε ποιο μεγάλη βάση στα χαρακτηριστικά του κρυπτονομίσματος της Binance (BNB) παρά στο Ethereum και Bitcoin. Τα αποτελέσματα του αλγορίθμου γραμμικής παλινδρόμησης έδειξαν ότι οι προβλέψεις ήταν αρκετά ακριβείς, με τιμή RMSE στο σετ εκπαίδευσης λιγότερο από 1400 μονάδες και στο τεστ επικύρωσης λιγότερο από 800 μονάδες.

Στον αλγόριθμο **Decision Tree** τα αποτελέσματα συγκρίθηκαν με τα πραγματικά δεδομένα. Το μήκος από το σετ δεδομένων είναι από τις 01/01/2020 έως την τρέχουσα περίοδο με στόχο την πρόβλεψη του Bitcoin. Ο αλγόριθμος εκπαιδεύτηκε στο 80% του σετ δεδομένων, έπειτα προβλέπει τις επόμενες 7 ημέρες, αφαιρεί 7 ημέρες από την αρχή του σετ δεδομένων ώστε να επανεκπαιδευτεί στο 80% στο σετ δεδομένων. Αυτή η διαδικασία επαναλαμβάνεται συνεχώς μέχρι την τρέχουσα χρονική στιγμή. Ο αλγόριθμος Decision Tree δεν έδωσε μεγάλη βάση στα χαρακτηριστικά που είχε σαν είσοδο. Τα αποτελέσματα του αλγορίθμου Decision Tree έδειξαν ότι οι προβλέψεις δεν ήταν αρκετά ικανοποιητικές, με τιμή RMSE στο σετ εκπαίδευσης λιγότερο από 0 μονάδες και στο τεστ επικύρωσης περισσότερο από 1200 μονάδες.

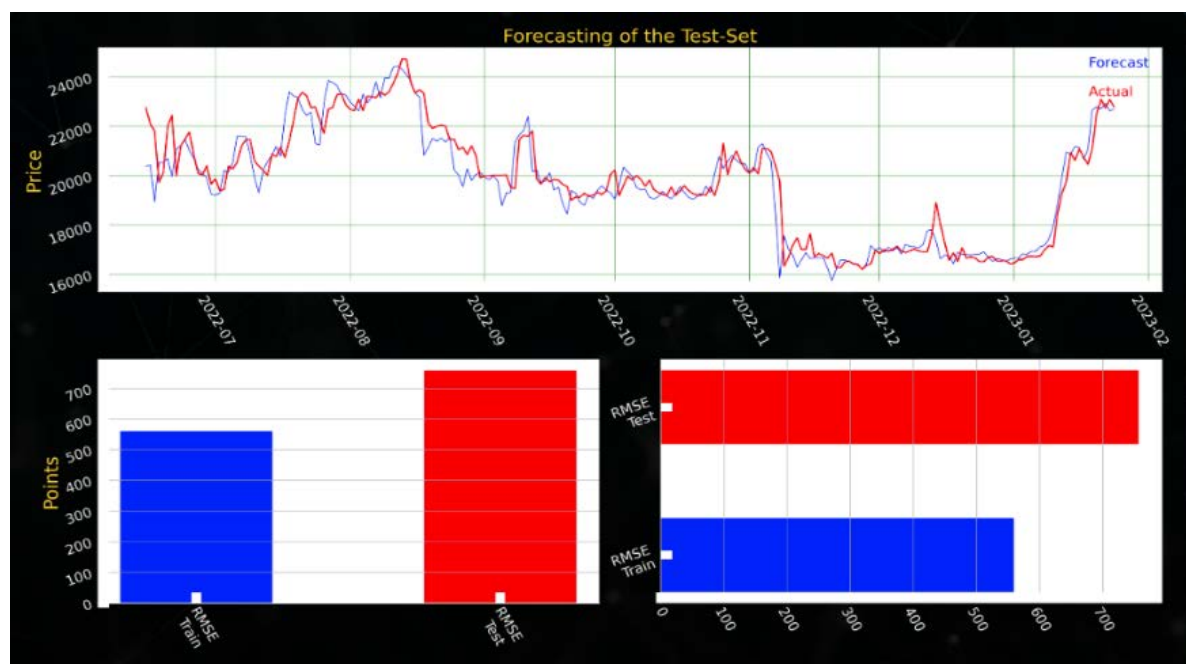


Εικόνα 24: Σχεδιαγράμματα αποτελεσμάτων Decision Tree

Στο παραπάνω σχεδιάγραμμα του Decision Tree ο αλγόριθμος παρουσιάζει υπερπροσαρμογή (overfitting).

Στον αλγόριθμο **Random Forest Regressor** τα αποτελέσματα συγκρίθηκαν με τα πραγματικά δεδομένα. Το μήκος από το σετ δεδομένων είναι από τις 01/01/2020 έως την τρέχουσα περίοδο με στόχο την πρόβλεψη του Bitcoin. Ο αλγόριθμος εκπαιδεύτηκε στο 80% του σετ δεδομένων, έπειτα προβλέπει τις επόμενες 7 ημέρες, αφαιρεί 7 ημέρες από την αρχή του σετ δεδομένων ώστε να επανεκπαιδεύεται στο 80% στο σετ δεδομένων. Αυτή η διαδικασία επαναλαμβάνεται συνεχώς μέχρι την τρέχουσα χρονική στιγμή.

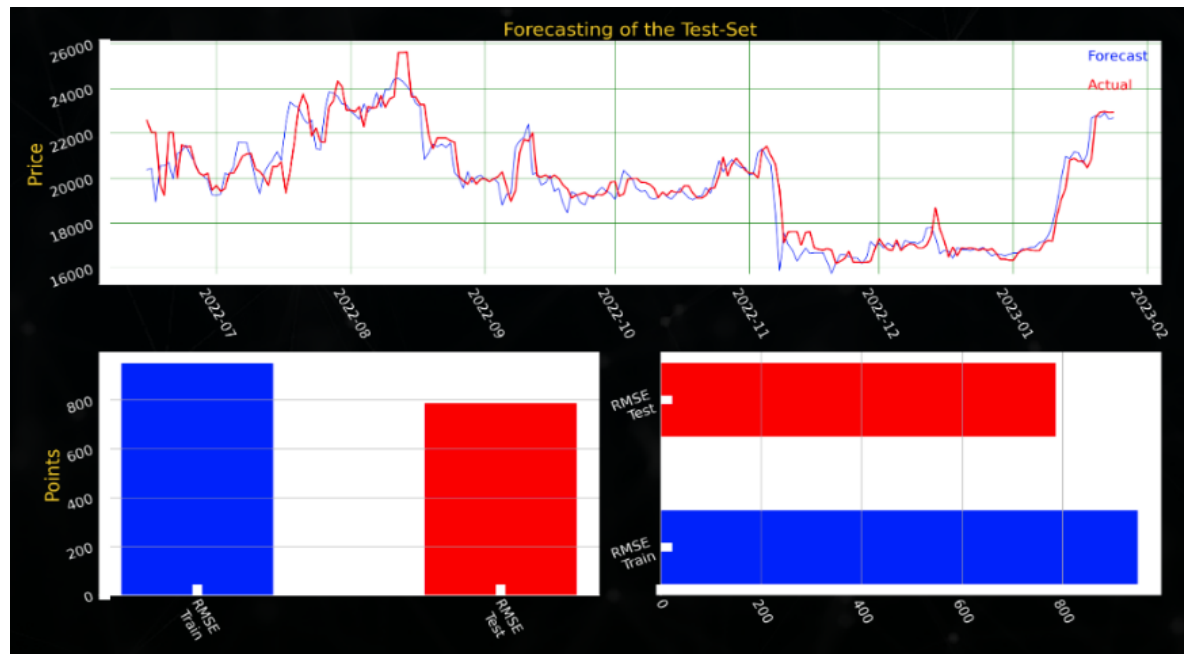
Ο αλγόριθμος Random Forest Regressor δεν έδωσε μεγάλη βάση στα χαρακτηριστικά που είχε σαν είσοδο. Τα αποτελέσματα του αλγορίθμου Random Forest Regressor έδειξαν ότι οι προβλέψεις ήταν ποιο ικανοποιητικές με τον προηγούμενο αλγόριθμο(Decision Tree), με τιμή RMSE στο σετ εκπαίδευσης λιγότερο από 600 μονάδες και στο τεστ επικύρωσης περισσότερο από 1000 μονάδες.



Εικόνα 25: Σχεδιαγράμματα αποτελεσμάτων Random Forest Regressor

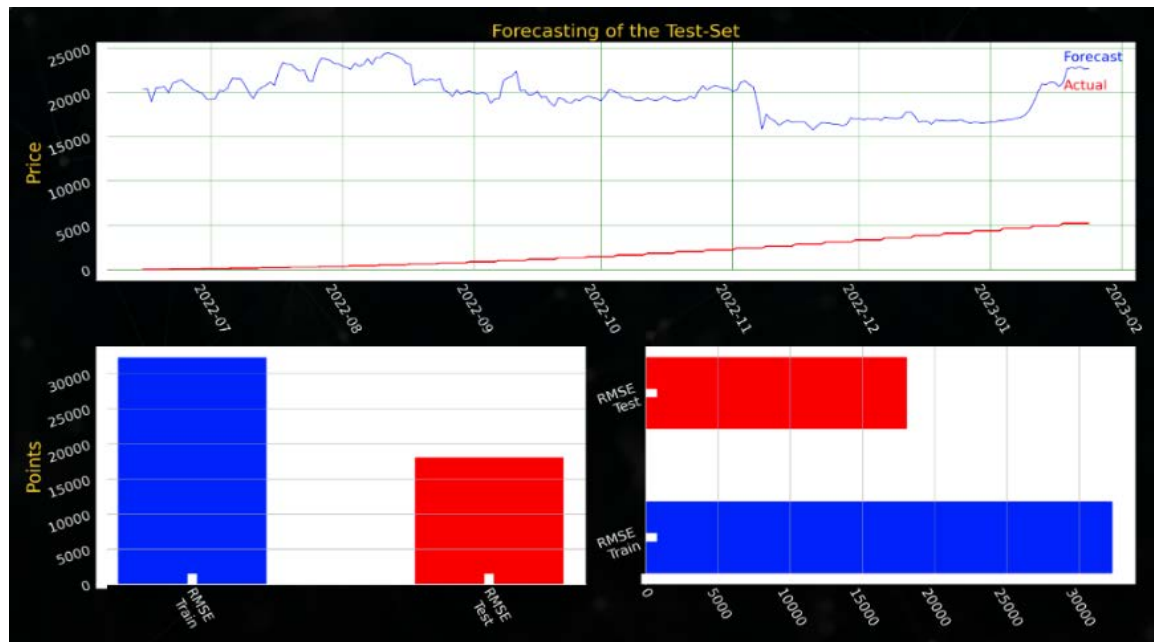
Στον αλγόριθμο **Gradient Boosting Regressor** τα αποτελέσματα συγκρίθηκαν με τα πραγματικά δεδομένα. Το μήκος από το σετ δεδομένων είναι από τις 01/01/2020 έως την τρέχουσα περίοδο με στόχο την πρόβλεψη του Bitcoin. Ο αλγόριθμος εκπαιδεύτηκε στο 80% του σετ δεδομένων, έπειτα προβλέπει τις επόμενες 7 ημέρες, αφαιρεί 7 ημέρες από την αρχή του σετ δεδομένων ώστε να επανεκπαιδεύεται στο 80% στο σετ

δεδομένων. Αυτή η διαδικασία επαναλαμβάνεται συνεχώς μέχρι την τρέχουσα χρονική στιγμή. Ο αλγόριθμος Random Forest δεν έδωσε μεγάλη βάση στα χαρακτηριστικά που είχε σαν είσοδο. Τα αποτελέσματα του αλγορίθμου Gradient Boosting Regressor έδειξαν ότι οι προβλέψεις ήταν αρκετά ακριβείς, με τιμή RMSE στο σετ εκπαίδευσης λιγότερο από 1000 μονάδες και στο τεστ επικύρωσης λιγότερο και πάλι από 1000 μονάδες.



Εικόνα 26: Σχεδιαγράμματα αποτελεσμάτων Gradient Boosting Regressor

Στον αλγόριθμο **LSTM** τα αποτελέσματα συγκρίθηκαν με τα πραγματικά δεδομένα. Το μήκος από το σετ δεδομένων είναι από τις 01/01/2020 έως την τρέχουσα περίοδο με στόχο την πρόβλεψη του Bitcoin. Ο αλγόριθμος εκπαιδεύτηκε στο 80% του σετ δεδομένων, έπειτα προβλέπει τις επόμενες 7 ημέρες, αφαιρεί 7 ημέρες από την αρχή του σετ δεδομένων ώστε να επανεκπαιδεύεται στο 80% στο σετ δεδομένων. Αυτή η διαδικασία επαναλαμβάνεται συνεχώς μέχρι την τρέχουσα χρονική στιγμή. Τα αποτελέσματα του αλγορίθμου LSTM έδειξαν ότι οι προβλέψεις δεν ήταν αρκετά ακριβείς, με τιμή RMSE στο σετ εκπαίδευσης περισσότερο από 30000 μονάδες και στο τεστ επικύρωσης περισσότερο από 15000.



Εικόνα 27: Σχεδιαγράμματα αποτελεσμάτων LSTM

Από τα παραπάνω σχεδιαγράμματα παρατηρήθηκε ότι ο αλγόριθμος Linear Regression εκτός το ότι μπορεί και προβλέπει την τιμή του Bitcoin σχεδόν με μεγάλη ακρίβεια, μπορεί επίσης και προβλέπει συνεχώς την τάση του κρυπτονομίσματος, δηλαδή εάν θα ανέβει ή θα πέσει η τιμή του κρυπτονομίσματος, ο αλγόριθμος το προβλέπει με μεγάλη επιτυχία.



Εικόνα 28: Η πρόβλεψη της τάσης του κρυπτονομίσματος Bitcoin

Συμπερασματικά, τα αποτελέσματα των προβλέψεων που έγιναν από τους αλγόριθμους μηχανικής μάθησης αναλύθηκαν και αξιολογήθηκαν, όλοι οι αλγόριθμοι παρήγαγαν λογικά ακριβείς προβλέψεις, το Gradient Boosting Regressor και το Linear Regression είχαν τα καλύτερα αποτελέσματα. Ωστόσο, πρέπει να σημειωθεί ότι η απόδοση των αλγορίθμων μπορεί να ποικίλλει ανάλογα με το συγκεκριμένο σύνολο δεδομένων και τις παραμέτρους που χρησιμοποιούνται.

Συμπεράσματα

Συμπερασματικά, οι αλγόριθμοι μηχανικής μάθησης που δοκιμάστηκαν στην εφαρμογή Ιστού έχουν δείξει πολλά υποσχόμενα αποτελέσματα για την πρόβλεψη των τιμών των κρυπτονομισμάτων. Από τα αποτελέσματα που προκύπτουν από την εκτέλεση των αλγορίθμων, μπορεί να φανεί ότι κάθε αλγόριθμος έχει τα δικά του πλεονεκτήματα και αδυναμίες και ο καταλληλότερος αλγόριθμος για μια δεδομένη εργασία θα εξαρτηθεί από τους συγκεκριμένους στόχους και περιορισμούς του χρήστη.

Η Γραμμική Παλινδρόμηση (**Linear regression**) βρέθηκε ότι είναι μια απλή και ευρέως χρησιμοποιούμενη μέθοδος που είναι ιδιαίτερα κατάλληλη για την πρόβλεψη συνεχών μεταβλητών, όπως η τιμή. Ωστόσο, ενδέχεται να μην είναι σε θέση να καταγράψει πιο σύνθετες σχέσεις στα δεδομένα. Τα δέντρα αποφάσεων (**Decision trees**) βρέθηκε ότι είναι ένας ευέλικτος αλγόριθμος που μπορεί να χρησιμοποιηθεί για την πρόβλεψη τόσο συνεχών όσο και κατηγορικών μεταβλητών. Είναι σχετικά ακριβείς και γρήγοροι, αλλά είναι επιρρεπείς στην υπερβολική προσαρμογή (overfitting) για μικρότερους χρονικούς ορίζοντες. Η Παλινδρόμηση Ενίσχυσης Κλίσης (**Gradient Boosting Regression**) και το Τυχαίο Δάσος (**Random Forest**) βρέθηκαν να είναι πιο προηγμένοι αλγόριθμοι που ήταν ιδιαίτερα ακριβείς για μικρότερους χρονικούς ορίζοντες, ωστόσο, είναι επίσης αργοί και πολύπλοκοι. Ο **LSTM** είναι ένας ισχυρός αλγόριθμος για τη μοντελοποίηση δεδομένων χρονοσειρών και ήταν σε θέση να συλλάβει χρονικά μοτίβα στα δεδομένα αλλά και αργή επεξεργασία.

Όταν πρόκειται να καθορίσουμε ποιος αλγόριθμος είναι ο πιο διαδεδομένος για το πρόβλημα που προσπαθούμε να λύσουμε, εξαρτάται από τη συγκεκριμένη περίπτωση χρήσης και τις ανταλλαγές που είναι διατεθειμένος να κάνει ο χρήστης. Γενικά, το Gradient Boosting Regression και το Random Forest είναι και τα δύο εξαιρετικά ακριβή και έχουν καλή απόδοση σε σύνολα δεδομένων με μεγάλο αριθμό χαρακτηριστικών. Το LSTM είναι μια εξαιρετική επιλογή όταν το επίκεντρο είναι η μοντελοποίηση δεδομένων χρονοσειρών και είναι ιδιαίτερα χρήσιμο εάν υπάρχουν χρονικά μοτίβα μέσα στα δεδομένα. Τελικά, η απόφαση για το ποιος αλγόριθμος θα χρησιμοποιηθεί θα εξαρτηθεί από τους συγκεκριμένους στόχους, τους πόρους και τους περιορισμούς του χρήστη.

Μερικά θετικά και αρνητικά στοιχεία της διαδικτυακής εφαρμογής που μπορούν να φανούν είναι τα εξής:

Θετικά:

- Η εφαρμογή παρέχει μια βολική και φιλική προς το χρήστη διεπαφή για πρόσβαση σε δεδομένα πρόβλεψης κρυπτονομισμάτων.
- Επιτρέπει στους εγγεγραμμένους χρήστες να δουν περισσότερες πληροφορίες για τα μοντέλα Μηχανικής Μάθησης.
- Ο αποκριτικός σχεδιασμός διασφαλίζει ότι η εφαρμογή είναι προσβάσιμη και εύκολη στη χρήση σε ένα ευρύ φάσμα συσκευών.

Αρνητικά:

- Η ακρίβεια των δεδομένων πρόβλεψης μπορεί να εξαρτάται από την ποιότητα των μοντέλων πρόβλεψης που χρησιμοποιούνται.
- Η εφαρμογή μπορεί να είναι ευάλωτη σε κινδύνους ασφαλείας, όπως μη εξουσιοδοτημένη πρόσβαση ή παραβιάσεις δεδομένων, εάν δεν ληφθούν τα κατάλληλα μέτρα για την προστασία των πληροφοριών του χρήστη.
- Η εφαρμογή μπορεί να απαιτεί πολλούς πόρους από το σύστημα και αυτό έχει σαν αποτέλεσμα την καθυστέρηση στην επεξεργασία των δεδομένων.

Για την **επέκταση και την αναβάθμιση της εφαρμογής** στο μέλλον, ορισμένες πιθανές ιδέες θα μπορούσαν να περιλαμβάνουν:

- Προσθήκη περισσότερων μοντέλων πρόβλεψης ή βελτίωση των υπάρχοντων μοντέλων για αύξηση της ακρίβειας των προβλέψεων.
- Εφαρμογή πρόσθετων μέτρων ασφαλείας, όπως κρυπτογράφηση ή έλεγχος ταυτότητας δύο παραγόντων, για την προστασία των δεδομένων χρήστη.
- Προσθήκη περισσότερων κρυπτονομισμάτων ή άλλων χρηματοοικονομικών μέσων στην εφαρμογή, όπως μετοχές ή εμπορεύματα.
- Ενσωμάτωση με άλλες υπηρεσίες ή πλατφόρμες, όπως ανταλλαγές ή πλατφόρμες συναλλαγών, για να παρέχει στους χρήστες μια πιο ολοκληρωμένη εμπειρία οικονομικής διαχείρισης.

- Ενσωμάτωση παράλληλων νημάτων(threads) ώστε να μειωθεί ο χρόνος επεξεργασίας των δεδομένων.

Η φιλοσοφία πίσω από την εφαρμογή είναι να παρέχει στους χρήστες έναν βολικό και αξιόπιστο τρόπο πρόσβασης σε δεδομένα πρόβλεψης κρυπτονομισμάτων και λήψη τεκμηριωμένων αποφάσεων σχετικά με τις επενδύσεις τους. Χρησιμοποιώντας μοντέλα μηχανικής μάθησης για τη δημιουργία των προβλέψεων, η εφαρμογή στοχεύει να παρέχει στους χρήστες μια πιο ακριβή και αντικειμενική άποψη της αγοράς, αντί να βασίζεται σε υποκειμενικές απόψεις ή ένστικτα.

Σε κάθε περίπτωση, είναι σημαντικό να σημειωθεί ότι πρόκειται για ένα πεδίο με πολλή έρευνα σε εξέλιξη και ενδέχεται να προκύψουν νέες τεχνικές και τεχνολογίες, επομένως είναι σημαντικό να παρακολουθούνται οι εξελίξεις στον τομέα για τη βελτίωση της απόδοσης του αλγορίθμου με την πάροδο του χρόνου.

Βιβλιογραφία

- [1] Κ. ΔΙΑΜΑΝΤΑΡΑΣ και Δ. Α. ΜΠΟΤΣΗΣ, «ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ / ΔΙΑΜΑΝΤΑΡΑΣ ΚΩΝΣΤΑΝΤΙΝΟ , ΜΠΟΤΣΗΣ Α. ΔΗΜΗΤΡΗΣ,» ΒΙΒΛΙΟΠΩΛΕΙΟ Πολιτεία, July 2019. [Ηλεκτρονικό]. Available: <https://www.politeianet.gr/books/9789604619955-diamantaras-konstantinos-kleidarithmos-michaniki-mathisi-310786>. [Πρόσβαση 24 July 2022].
- [2] Ι. Βλαχάβας, Π. Κεφαλάς, Ν. Βασιλειάδης, Φ. Κόκκορας και Η. Σακελλαρίου, «ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ (Δ' ΕΚΔΟΣΗ) / ΣΥΛΛΟΓΙΚΟ,» ΒΙΒΛΙΟΠΩΛΕΙΟ Πολιτεία, 2011. [Ηλεκτρονικό]. Available: <https://www.politeianet.gr/books/9786185196448-sullogiko-panepistimio-makedonias-techniti-noimosuni-d-ekdosi-207432>. [Πρόσβαση 21 June 2022].
- [3] scikit-learn, «3.3. Metrics and scoring: quantifying the quality of predictions,» scikit-learn, 2022. [Ηλεκτρονικό]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html. [Πρόσβαση 28 July 2022].
- [4] scikit-learn, «sklearn.metrics.precision_score,» scikit-learn, 2022. [Ηλεκτρονικό]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html. [Πρόσβαση 27 July 2022].
- [5] scikit-learn, «sklearn.metrics.recall_score,» scikit-learn, 2022. [Ηλεκτρονικό]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html. [Πρόσβαση 27 July 2022].
- [6] scikit-learn, «sklearn.metrics.f1_score,» scikit-learn, 2022. [Ηλεκτρονικό]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html. [Πρόσβαση 28 July 2022].
- [7] scikit-learn, «sklearn.metrics.mean_squared_error,» scikit-learn, 2022. [Ηλεκτρονικό]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html. [Πρόσβαση 21 September 2022].
- [8] T. Hastie, R. Tibshirani και J. Friendman, The elements of statistical learning. 2nd ed. Springer, 2017.
- [9] scikit-learn, «Linear Regression Example,» scikit-learn, 2022. [Ηλεκτρονικό]. Available: https://scikit-learn.org/stable/auto_examples/linear_model/plot_ols.html#sphx-glr-auto-examples-linear-model-plot-ols-py. [Πρόσβαση 18 June 2022].
- [10] A. Géron, Hands-on machine learning with scikit-learn, keras, and tensorflow. 2nd ed., Gravenstein Highway: Sebastopol: O' Reilly Media., 2019.
- [11] I. Goodfellow, Y. Bengio και A. Courville, Deep learning. Cambridge, Mass: The MIT Press, 2017.
- [12] Π. Πανόπουλος, «ΤΟ ΠΡΟΒΛΗΜΑ ΤΗΣ ΠΟΛΥΣΥΓΓΡΑΜΜΙΚΟΤΗΤΑΣ: ΕΚΤΙΜΗΤΕΣ ΕΛΑΧΙΣΤΩΝ ΤΕΤΡΑΓΩΝΩΝ ΣΕ ΠΟΛΛΑΠΛΑ ΣΤΑΔΙΑ. Οικονομικό Πανεπιστήμιο Αθηνών Τμήμα Στατιστικής.»

- [13] Θ. Γαβριηλίδης, «Αλγόριθμοι μηχανικής μάθησης για την αναγνώριση προσώπων. Thesis. Γαβριηλίδης, Θωμάς,» 2021.
- [14] J. Ogotu, T. Schluz-Streeck και H.-P. Piepho, «Genomic selection using regularized linear regression models: Ridge regression, lasso, elastic net and their extensions,» 2012.
- [15] scikit-learn, «sklearn.linear_model.Lasso,» scikit-learn, 2022. [Ηλεκτρονικό]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html. [Πρόσβαση 19 June 2022].
- [16] R. Battiti και M. Brunato , The LION way. Machine Learning plus intelligent Optimization, Los Angeles: Lionsolver Inc..
- [17] Al-Jawarneh, A.S., M. a. Awajan και A.M., «Elastic Net Regression and Empirical Mode Decomposition for Enhancing the Accuracy of the Model Selection,» 2021.
- [18] P. García-Nieto, E. García-Gonzalo και J. Peredes-Sánchez , «Prediction of the critical temperature of a superconductor by using the WOA/MARS, Ridge, Lasso and Elastic-net machine learning techniques,» 2021.
- [19] scikit-learn, «1.10. Decision Trees,» scikit-learn, 2022. [Ηλεκτρονικό]. Available: <https://scikit-learn.org/stable/modules/tree.html>. [Πρόσβαση 19 June 2022].
- [20] J. Quinlan, «Learning decision tree classifiers,» 1996.
- [21] S. Russell και P. Norvig, Artificial intelligence. 3rd ed., Noida, India: Pearson India Education Services Pvt. Ltd., 2018.
- [22] J. Su, «A Fast Decision Tree Learning Algorithm».
- [23] L. Breiman, «'Random Forests', Machine Learning».
- [24] K. Gurney, An Introduction to Neural Networks. 1st edition., London: CRC Press, `997.
- [25] Α. Κωστόπουλος, «Νέοι αλγόριθμοι εκπαίδευσης τεχνητών νευρωνικών δικτύων και εφαρμογές. Πανεπιστήμιο Πατρών Σχολή Θετικών Επιστημών Τμήμα Μαθηματικών.,» 2011.
- [26] Γ. ΡΙΖΟΣ, «Τεχνητά Νευρωνικά Δίκτυα Θεωρία & Εφαρμογές - ΡΙΖΟΣ ΓΙΩΡΓΟΣ» ΕΚΔΟΣΕΙΣ ΑΘ. ΣΤΑΜΟΥΛΗΣ, 1996. [Ηλεκτρονικό]. Available: https://www.stamoulis.gr/%CE%A4%CE%B5%CF%87%CE%BD%CE%B7%CF%84%CE%AC-%CE%9D%CE%B5%CF%85%CF%81%CF%89%CE%BD%CE%B9%CE%BA%CE%AC-%CE%94%CE%AF%CE%BA%CF%84%CF%85%CE%B1-%CE%98%CE%B5%CF%89%CF%81%CE%AF%CE%B1-%CE%95%CF%86%CE%B1%CF%81%CE%BC%CE%BF%CE%B3%CE%AD%CF%82_p-3. [Πρόσβαση 27 June 2022].
- [27] Ι. Βλαχάβας, Π. Κεφαλάς, Ν. Βασιλειάδης, Φ. Κόκκορας και Η. Σακελλαρίου, «Τεχνητή Νοημοσύνη - Γ' Έκδοση,» auth, 2011. [Ηλεκτρονικό]. Available: <http://aibook.csd.auth.gr/>. [Πρόσβαση 19 September 2022].

- [28] K. O'Shea και R. Nash, «An Introduction to Convolutional Neural Networks,» 2015.
- [29] H. Sak, A. Senior και F. Beaufays, «Long short-term memory recurrent neural network architectures for large scale acoustic modeling,» 2014.
- [30] L. Rutten, «Bayes' theorem: scientific assessment of experience,» 2021.
- [31] Y. Youssef, «Bayes Theorem and Real-life Applications,» 2022.
- [32] scikit-learn, «1.9 Naive Bayes,» scikit-learn, 2022. [Ηλεκτρονικό]. Available: https://scikit-learn/stable/modules/naive_bayes.html. [Πρόσβαση 21 October 2022].
- [33] Oppla, «Method Factsheet - Bayesian belief networks (BBN) | Oppla,» Oppla, [Ηλεκτρονικό]. Available: <https://oppla.eu/product/2018>. [Πρόσβαση 22 October 2022].
- [34] L. Uusitalo, «Advantages and challenges of Bayesian networks in environmental modelling,» 2007.
- [35] R. Kenett, «Applications of Bayesian Networks, SSRN Electronic Journal [Preprint],» 2012.