

Τμήμα Μηχανικών Βιομηχανικής Σχεδίασης και
Παραγωγής



"Πρόβλεψη Σημάτων με Μεθόδους Μηχανικής Μάθησης"

"Signal Prediction via Machine Learning Methods"

Ροδίτης Δημήτριος
ΑΜ:222017028

Επιβλέπων Καθηγητής : Κάντζος Δημήτριος

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Κάντζος Δημήτριος

Λελίγκου Ελένη Αικατερίνη

Νικολάου Γρηγόρης

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ/ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Ροδίτης Δημήτριος του Δημητρίου, με αριθμό μητρώου 222017028 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών

Δ. Ροδίτης

ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη	6
Abstract	7
Εισαγωγή	8
1. Τεχνητή Νοημοσύνη (Artificial Intelligence)	9
1.1 Μηχανική Μάθηση (Machine Learning)	10
1.1.1 Είδη Μηχανικής Μάθησης.....	13
1.1.2 Εκπαίδευση	13
1.1.3 Συναρτήσεις Απώλειας (Loss Functions)	15
1.2 Βαθιά Μάθηση (Deep Learning)	16
1.2.1 Διαφορές Κλασικής Μηχανικής Μάθησης και Βαθιάς Μάθησης.....	16
2. Αλγόριθμοι Κλασικής Μηχανικής Μάθησης	18
2.1 Γραμμική Παλινδρόμηση (Linear Regression).....	18
2.2 Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines)	19
3. Αλγόριθμοι Βαθιάς Μάθησης.....	23
3.1 Νευρωνικά Δίκτυα	23
3.1.1 Βελτιστοποιητές (Optimizers)	24
3.1.2 Οπισθοδιάδοση (Backpropagation)	28
3.2 Επαναλαμβανόμενα Νευρωνικά Δίκτυα (Recurrent Neural Networks)	29
3.2.1 Long Short Term Memory (LSTM).....	30
4. Βασικοί Χρηματοοικονομικοί Όροι.....	32
4.1 Χρηματιστήριο.....	32
4.1.1 Δημιουργία και Σκοπός του Χρηματιστηρίου	32
4.1.2 Η έννοια της μετοχής	32
4.1.3 Είδη Μετοχών	33
4.1.4 Σημαντικότερα Χρηματιστήρια	34
5. Υλοποίηση	35
5.1 Ανάλυση των δεδομένων	35
5.2 Ανάλυση του Linear Regression.....	36
5.3 Ανάλυση του SVM	37
5.4 Ανάλυση του LSTM	37
5.5 Ανάλυση Αποτελεσμάτων	39
5.6 Συμπεράσματα	55
Βιβλιογραφία	58

Πηγές Εικονών	60
Κατάλογος Πινάκων	61

Περίληψη

Σκοπός της εργασίας είναι η πρόβλεψη χρηματοοικονομικών δεδομένων με μεθόδους μηχανικής μάθησης και ειδικότερα βαθιάς μάθησης. Αρχικά παρουσιάζεται το απαραίτητο θεωρητικό υπόβαθρο που αφορά τη μηχανική μάθηση. Στη συνέχεια περιγράφονται οι αλγόριθμοι κλασικής μηχανικής μάθησης και βαθιάς μάθησης που θα χρησιμοποιηθούν για την πρόβλεψη των δεδομένων. Ακολουθεί η αλγοριθμική υλοποίηση, όπου επιχειρούμε να προβλέψουμε την κίνηση των μετοχών 8 γνωστών εταιρειών με τη χρήση δύο μοντέλων κλασικής μηχανικής μάθησης και ενός βαθιάς μάθησης. Στο τελευταίο κεφάλαιο παρουσιάζονται τα αποτελέσματα της σύγκρισης των μοντέλων όπως και οι προβληματισμοί και προτεινόμενοι τρόποι βελτίωσης για μελλοντική έρευνα.

Abstract

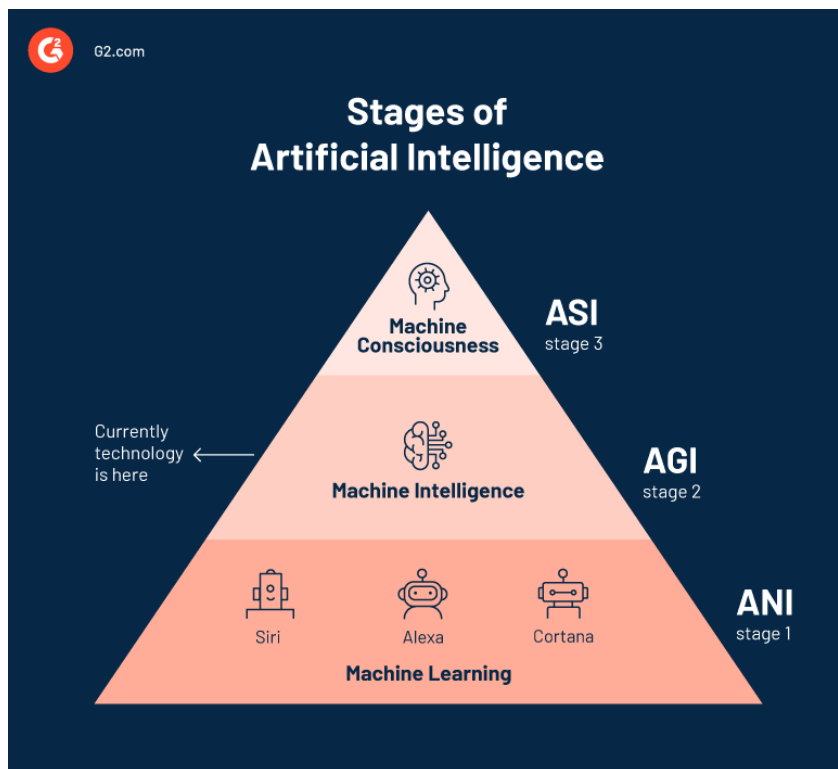
The purpose of this thesis is the prediction of financial data with machine learning methods and particularly deep learning methods. Firstly, the necessary theoretical background on machine learning is presented. Next, the classic machine learning and deep learning algorithms that will be used to predict the data are described. The algorithmic implementation follows where we attempt to predict the stock price of 8 well-known companies using two classic machine learning models and one deep learning model. In the last chapter, the results on the comparison of the models are presented along with the concerns and suggested improvement steps for future research.

Εισαγωγή

Η μηχανική μάθηση είναι ένας επιστημονικός κλάδος που εξελίσσεται σημαντικά, ιδιαίτερα τα τελευταία χρόνια. Η χρήση τεχνικών μηχανικής μάθησης εμφανίζεται ιδιαίτερα στον εταιρικό κόσμο με σκοπό τη διευκόλυνση των ανθρώπινων εργασιών και τη δυνατότητα για μεγαλύτερο κέρδος. Ένας χώρος όπου εφαρμόζεται η μηχανική μάθηση είναι ο κλάδος των χρηματοοικονομικών και συγκεκριμένα το χρηματιστήριο για την πρόβλεψη μετοχών. Πολλές επενδυτικές εταιρείες έχουν ήδη τμήματα που ερευνούν και ασχολούνται με την υλοποίηση τεχνικών για την πρόβλεψη μετοχών. Η χρήση της μηχανικής μάθησης μπορεί να διευκολύνει αυτές τις εταιρείες να λάβουν την απόφαση να αγοραστεί μια συγκεκριμένη μετοχή, τότε θα ήταν καλή ιδέα να αγοραστεί, ακόμα και τι κέρδος θα έχει στο μέλλον βάσει μίας πρόβλεψης. Η εργασία θα ξεκινήσει με την εισαγωγή στο πεδίο της μηχανικής μάθησης και ειδικότερα της βαθιάς μάθησης, θα περιγραφούν οι βασικοί αλγόριθμοι που θα χρησιμοποιηθούν ενώ θα παρουσιαστεί και η βασική χρηματοοικονομική ορολογία που είναι απαραίτητη στην ανάλυση μετοχών. Στη συνέχεια θα υλοποιηθούν δύο μοντέλα κλασικής μηχανικής μάθησης και ένα βαθιάς μάθησης τα οποία δέχονται ως είσοδο τιμές μετοχής και παράγουν ως έξοδο την προβλεπόμενη τιμή της επόμενης μέρας.

1. Τεχνητή Νοημοσύνη (Artificial Intelligence)

Τεχνητή Νοημοσύνη ονομάζεται ο κλάδος της επιστήμης που προσπαθεί να προσομοιώσει την δυνατότητα της ανθρώπινης νοημοσύνης πάνω στην λειτουργία υπολογιστών. Χρησιμοποιεί την ανάπτυξη αλγορίθμων και προγραμμάτων υπολογιστής ισχύς που είναι ικανά να εκτελέσουν εργασίες που απαιτούν σχεδόν πάντα ανθρώπινη νοημοσύνη, όπως οπτική αντίληψη, αναγνώριση ομιλίας, λήψη αποφάσεων και μετάφραση γλώσσας. Το AI είναι μία μεγάλη επιστήμη που περιλαμβάνει πολλά υποπεδία, τα πιο σημαντικά είναι η Μηχανική Μάθηση και η Βαθιά Μάθηση που θα σχολιαστούν αργότερα στην εργασία. Η Μηχανική Μάθηση περιλαμβάνει την ανάπτυξη αλγορίθμων που μπορούν να μάθουν από δεδομένα και να κάνουν προβλέψεις, ενώ η Βαθιά Μάθηση περιλαμβάνει τη χρήση τεχνητών νευρωνικών δικτύων για την ανάλυση μεγάλων και πολύπλοκων συνόλων δεδομένων.



Εικόνα 1.1 : Κατηγορίες Τεχνητής Νοημοσύνης

Το AI κατηγοριοποιείται στις εξής κατηγορίες :

1. Αδύναμη τεχνητή νοημοσύνη (Narrow AI): Το Narrow AI, έχει σχεδιαστεί για να εκτελεί μια συγκεκριμένη εργασία ή ένα σύνολο εργασιών, όπως η αναγνώριση εικόνας. Αυτά τα συστήματα είναι εκπαιδευμένα και βελτιστοποιημένα για να εκτελούν μια συγκεκριμένη εργασία, αλλά δεν μπορούν να εκτελέσουν άλλες εργασίες που απαιτούν γενική ευφυΐα. Τα συστήματα στενής τεχνητής νοημοσύνης είναι επί του παρόντος ο πιο κοινός τύπος τεχνητής νοημοσύνης που χρησιμοποιείται σήμερα.

2. Γενική τεχνητή νοημοσύνη (General AI): Η γενική τεχνητή νοημοσύνη, γνωστή και ως ισχυρή τεχνητή νοημοσύνη, αναφέρεται σε ένα σύστημα τεχνητής

νοημοσύνης που έχει την ικανότητα να εκτελεί ένα ευρύ φάσμα εργασιών που συνήθως απαιτούν ανθρώπινη νοημοσύνη, όπως η κατανόηση της φυσικής γλώσσας, η λήψη αποφάσεων και η επίλυση προβλημάτων. Τα συστήματα γενικής τεχνητής νοημοσύνης έχουν τη δυνατότητα να εκτελέσουν οποιαδήποτε πνευματική εργασία μπορεί ένας άνθρωπος, και δεν περιορίζονται σε μια συγκεκριμένη εργασία ή σύνολο εργασιών.

3. Υπερ-έξυπνη τεχνητή νοημοσύνη (Superintelligent AI): Η υπερέξυπνη τεχνητή νοημοσύνη αναφέρεται σε ένα σύστημα τεχνητής νοημοσύνης που ξεπερνά την ανθρώπινη νοημοσύνη σε πολλούς τομείς και έχει τη δυνατότητα να είναι πολύ πιο ικανό από τους ανθρώπους σε πολλούς τομείς. Ορισμένοι ειδικοί πιστεύουν ότι η υπερέξυπνη τεχνητή νοημοσύνη έχει τη δυνατότητα να είναι μια τεχνολογία μετασχηματισμού που θα μπορούσε να ωφελήσει πολύ την ανθρωπότητα ως σύνολο.

1.1 Μηχανική Μάθηση (Machine Learning)

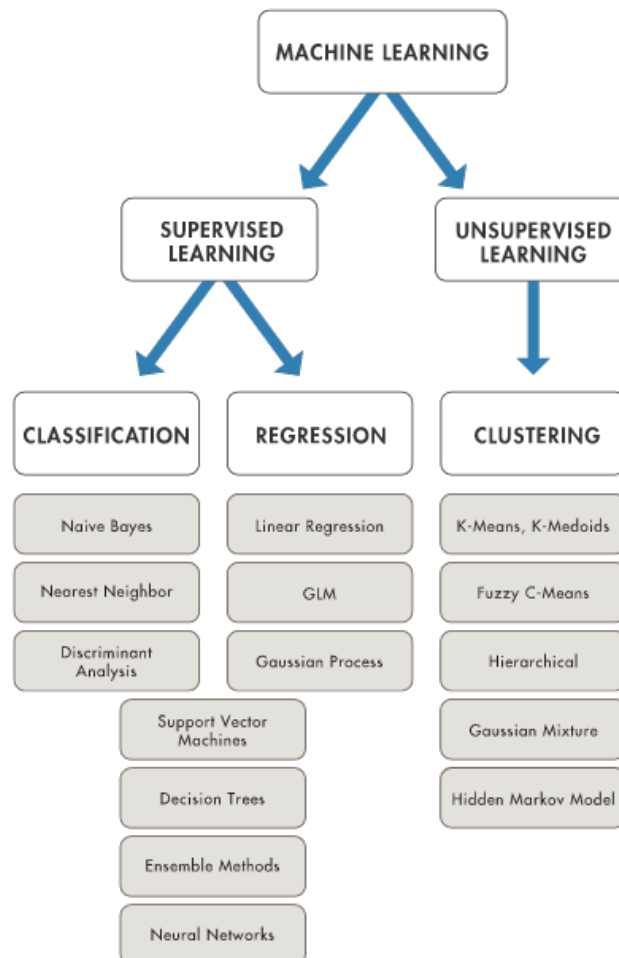
Η μηχανική μάθηση είναι ένας τομέας της επιστήμης των υπολογιστών που χρησιμοποιεί στατιστικές μεθόδους για να επιτρέψει στους υπολογιστές να βελτιώσουν την απόδοσή τους σε μια συγκεκριμένη εργασία μέσω της εμπειρίας. Είναι ένα υποπεδίο της τεχνητής νοημοσύνης και εστιάζει στην ανάπτυξη αλγορίθμων που μπορούν να μάθουν μοτίβα σε δεδομένα, να κάνουν προβλέψεις ή να εκτελέσουν ενέργειες με βάση τις εισροές δεδομένων χωρίς να είναι ρητά προγραμματισμένοι για την εκτέλεση μιας εργασίας. Υπάρχουν τρεις κύριοι τύποι μηχανικής μάθησης: η εποπτευόμενη μάθηση, η μάθηση χωρίς επίβλεψη και η ενισχυτική μάθηση.

Οι αλγόριθμοι μηχανικής μάθησης κατηγοριοποιούνται συχνά με βάση το μαθησιακό τους στυλ, το οποίο αναφέρεται στο ποσό και το είδος της επίβλεψης που λαμβάνουν κατά τη διάρκεια της εκπαιδευτικής διαδικασίας. Μερικοί αλγόριθμοι, όπως τα δέντρα αποφάσεων και οι k-πλησιέστεροι γείτονες, είναι απλοί και εύκολο να κατανοηθούν, ενώ άλλοι, όπως τα βαθιά νευρωνικά δίκτυα, είναι πιο περίπλοκοι και δύσκολο να ερμηνευτούν. Ανεξάρτητα από τον συγκεκριμένο αλγόριθμο που χρησιμοποιείται, ο στόχος της μηχανικής μάθησης είναι η κατασκευή μοντέλων που μπορούν να γενικεύονται καλά σε νέα δεδομένα. Αυτό επιτυγχάνεται με την εκπαίδευση του μοντέλου σε ένα μεγάλο, αντιπροσωπευτικό σύνολο δεδομένων και με τη χρήση τεχνικών όπως η διασταυρούμενη επικύρωση για την αποφυγή υπερπροσαρμογής, όταν το μοντέλο εξειδικεύεται υπερβολικά στα δεδομένα εκπαίδευσης και έχει κακή απόδοση σε νέα δεδομένα.

Γενικότερα, η μηχανική μάθηση είναι ένα ισχυρό εργαλείο για την πραγματοποίηση προβλέψεων, την εύρεση προτύπων σε δεδομένα και την αυτοματοποίηση εργασιών. Η εποπτευόμενη μάθηση είναι ένας τύπος μηχανικής μάθησης όπου ο στόχος είναι η πρόβλεψη μιας μεταβλητής στόχου (γνωστή και ως μεταβλητή εξόδου ή εξαρτημένη μεταβλητή) με βάση ένα σύνολο μεταβλητών εισόδου (γνωστές επίσης ως χαρακτηριστικά ή ανεξάρτητες μεταβλητές). Στην εποπτευόμενη μάθηση, τα δεδομένα εκπαίδευσης αποτελούνται από παραδείγματα με ετικέτα, που σημαίνει ότι η επιθυμητή έξοδος για κάθε παράδειγμα είναι ήδη γνωστή. Το μοντέλο εκπαιδεύεται σε αυτά τα δεδομένα με ετικέτα και ο στόχος είναι να μάθει μια αντιστοίχιση από τις μεταβλητές εισόδου στη μεταβλητή στόχο που μπορεί να γενικευτεί σε νέα, άορατα δεδομένα.

Η εποπτευόμενη μάθηση χρησιμοποιείται σε ένα ευρύ φάσμα εφαρμογών, συμπεριλαμβανομένων ενδεικτικά:

- Ταξινόμηση: Πρόβλεψη μιας κατηγορικής παραγωγής, όπως "σκύλος" ή "γάτα", δεδομένου ενός συνόλου χαρακτηριστικών.
- Παλινδρόμηση: Πρόβλεψη μιας συνεχούς παραγωγής, όπως η τιμή μιας μετοχής, δεδομένου ενός συνόλου χαρακτηριστικών.



Εικόνα 1.2 : Κατηγορίες Μηχανικής Μάθησης

Υπάρχουν πολλοί αλγόριθμοι που χρησιμοποιούνται για την εποπτευόμενη μάθηση, όπως:

- Γραμμική παλινδρόμηση: Ένα απλό γραμμικό μοντέλο που προβλέπει μια συνεχή μεταβλητή στόχο με βάση ένα σύνολο γραμμικών σχέσεων μεταξύ των μεταβλητών εισόδου και της μεταβλητής εξόδου.
- Λογιστική παλινδρόμηση: Ένα μοντέλο που χρησιμοποιείται για προβλήματα δυαδικής ταξινόμησης που προβλέπει την πιθανότητα ενός παραδείγματος να ανήκει σε μία από τις δύο κατηγορίες.

- Δέντρα απόφασης: Ένα μοντέλο που βασίζεται σε δέντρα που χρησιμοποιεί μια σειρά κανόνων εάν-τότε-άλλο για να κάνει προβλέψεις.
- Τυχαία δάση: Ένα σύνολο δέντρων αποφάσεων που κάνει προβλέψεις υπολογίζοντας τον μέσο όρο των αποτελεσμάτων πολλών δέντρων.
- Νευρωνικά δίκτυα: Ένας τύπος μοντέλου που εμπνέεται από τη δομή και τη λειτουργία του ανθρώπινου εγκεφάλου και χρησιμοποιείται τόσο για προβλήματα παλινδρόμησης όσο και για προβλήματα ταξινόμησης.
- Υποστήριξη διανυσματικών μηχανών (SVMs): Ένας τύπος μοντέλου που χρησιμοποιείται για προβλήματα δυαδικής ταξινόμησης που προσπαθεί να βρει ένα υπερεπίπεδο που διαχωρίζει τις δύο κατηγορίες στο χώρο χαρακτηριστικών.
- k-Κοντινότεροι γείτονες (k-NN): Ένα απλό και διαισθητικό μοντέλο που ταξινομεί ένα παράδειγμα με βάση την πλειοψηφία των k πλησιέστερων γειτόνων του στα δεδομένα εκπαίδευσης.

Με την χρήση της εποπτευόμενης μάθησης, είναι σημαντικό να χωρίζονται τα διαθέσιμα δεδομένα σε ένα σύνολο εκπαίδευσης και ένα σύνολο δοκιμών. Το μοντέλο εκπαιδεύεται στο σετ εκπαίδευσης. Το σύνολο δοκιμής χρησιμοποιείται για να ληφθεί μια εκτίμηση της απόδοσης του μοντέλου σε νέα δεδομένα. Η υπερβολική προσαρμογή είναι ένα κοινό πρόβλημα στην εποπτευόμενη μάθηση και εμφανίζεται όταν το μοντέλο εξειδικεύεται υπερβολικά στα δεδομένα εκπαίδευσης και έχει κακή απόδοση σε νέα δεδομένα. Για να αποφευχθεί η υπερβολική προσαρμογή, είναι σύνηθες να χρησιμοποιούνται τεχνικές τακτοποίησης.

Η εποπτευόμενη μάθηση είναι μια ισχυρή και ευρέως χρησιμοποιούμενη τεχνική στη μηχανική μάθηση που μπορεί να χρησιμοποιηθεί για να κάνει προβλέψεις, να ταξινομήσει δεδομένα και να λύσει ένα ευρύ φάσμα άλλων προβλημάτων. Ωστόσο, είναι σημαντικό να επιλέγεται προσεκτικά ο κατάλληλος αλγόριθμος, η ρύθμιση των υπερπαραμέτρων και να αποτρέπεται η υπερπροσαρμογή ώστε να προκαλείται καλύτερη απόδοση σε νέα δεδομένα.

Η μάθηση χωρίς επίβλεψη είναι ένας τύπος μηχανικής μάθησης όπου ο στόχος είναι η ανακάλυψη μοτίβων ή σχέσεων στα δεδομένα χωρίς προηγούμενη γνώση ή καθοδήγηση. Σε αντίθεση με την εποπτευόμενη μάθηση, όπου τα δεδομένα αποτελούνται από επισημασμένα παραδείγματα και ο στόχος είναι η πρόβλεψη της μεταβλητής στόχου με βάση τις μεταβλητές εισόδου, η μη εποπτευόμενη μάθηση περιλαμβάνει εκπαίδευση του μοντέλου σε ένα σύνολο δεδομένων χωρίς ετικέτα, όπου το επιθυμητό αποτέλεσμα είναι άγνωστο. Η ημι-εποπτευόμενη μάθηση είναι ένας τύπος μηχανικής μάθησης που συνδυάζει τόσο την εποπτευόμενη όσο και την μη εποπτευόμενη μάθηση. Ο στόχος της ημι-εποπτευόμενης μάθησης είναι να αξιοποιήσει μεγάλες ποσότητες δεδομένων χωρίς ετικέτα για να βελτιώσει την απόδοση ενός μοντέλου που έχει εκπαιδευτεί σε μικρότερο αριθμό δεδομένων με ετικέτα. Στην ημι-εποπτευόμενη μάθηση, το μοντέλο εκπαιδεύεται σε συνδυασμό δεδομένων

με ετικέτα και χωρίς ετικέτα. Τα δεδομένα με ετικέτα χρησιμοποιούνται για την επίβλεψη του μοντέλου, ενώ τα δεδομένα χωρίς ετικέτα χρησιμοποιούνται για την τακτοποίηση του μοντέλου και τη βελτίωση της ικανότητας γενίκευσής του.

Η ενισχυτική μάθηση (RL) είναι ένας τύπος μηχανικής μάθησης που εστιάζει στο πώς ένας πράκτορας πρέπει να συμπεριφέρεται σε ένα περιβάλλον προκειμένου να μεγιστοποιήσει ένα σήμα ανταμοιβής. Σε αντίθεση με την εποπτευόμενη μάθηση, όπου ο στόχος είναι η πρόβλεψη της μεταβλητής στόχου με βάση τις μεταβλητές εισόδου ή η μάθηση χωρίς επίβλεψη, όπου ο στόχος είναι η ανακάλυψη προτύπων στα δεδομένα, ο στόχος στην ενισχυτική μάθηση είναι η εκμάθηση μιας πολιτικής, η οποία είναι μια χαρτογράφηση από κράτη σε ενέργειες.

1.1.1 Είδη Μηχανικής Μάθησης

Η ταξινόμηση είναι ένας τύπος εποπτευόμενου προβλήματος μηχανικής μάθησης όπου ο στόχος είναι να προβλέψουμε την τάξη ή την κατηγορία ενός σημείου δεδομένων εισόδου. Με άλλα λόγια, δεδομένου ενός συνόλου χαρακτηριστικών εισόδου, η αποστολή ενός αλγορίθμου ταξινόμησης είναι να αντιστοιχίσει την είσοδο σε μία από πολλές προκαθορισμένες κλάσεις. Οι αλγόριθμοι ταξινόμησης χρησιμοποιούνται σε ένα ευρύ φάσμα εφαρμογών, συμπεριλαμβανομένης της ταξινόμησης εικόνων, της ανίχνευσης ανεπιθύμητων μηνυμάτων, της ανάλυσης συναισθημάτων και της ιατρικής διάγνωσης.

Η επιλογή του αλγορίθμου εξαρτάται από τη φύση των δεδομένων και το πρόβλημα που επιλύεται. Κατά την εκπαίδευση ενός μοντέλου ταξινόμησης, ο στόχος είναι να μάθετε μια αντιστοίχιση από τα χαρακτηριστικά εισόδου στις ετικέτες κλάσεων. Αυτό γίνεται συνήθως προσαρμόζοντας τις παραμέτρους του μοντέλου έτσι ώστε να ελαχιστοποιείται η διαφορά μεταξύ των προβλεπόμενων ετικετών κλάσης και των πραγματικών ετικετών κλάσης. Η επιλογή της συνάρτησης απώλειας που χρησιμοποιείται για την αξιολόγηση της απόδοσης του μοντέλου εξαρτάται από τον αλγόριθμο που χρησιμοποιείται.

Μετά την εκπαίδευση του μοντέλου, μπορεί να χρησιμοποιηθεί για να κάνει προβλέψεις σε νέα σημεία δεδομένων, εκχωρώντας την είσοδο στην κλάση με την υψηλότερη προβλεπόμενη πιθανότητα. Η παλινδρόμηση είναι ένας τύπος εποπτευόμενου προβλήματος μηχανικής μάθησης όπου ο στόχος είναι να προβλέψει μια συνεχή τιμή εξόδου δεδομένου ενός συνόλου χαρακτηριστικών εισόδου. Με άλλα λόγια, αλγόριθμοι παλινδρόμησης χρησιμοποιούνται για τη μοντελοποίηση της σχέσης μεταξύ μιας εξαρτημένης μεταβλητής και μιας ή περισσότερων ανεξάρτητων μεταβλητών.

1.1.2 Εκπαίδευση

Η εκπαίδευση ενός αλγορίθμου μηχανικής μάθησης περιλαμβάνει τα ακόλουθα βήματα:

1. Συλλογή δεδομένων: Το πρώτο βήμα στην εκπαίδευση ενός μοντέλου μηχανικής μάθησης είναι η συλλογή ενός συνόλου δεδομένων που αντιπροσωπεύει το πρόβλημα που προσπαθούμε να λύσουμε. Αυτό το σύνολο δεδομένων θα πρέπει να είναι αρκετά μεγάλο ώστε να αντιπροσωπεύει με ακρίβεια τα υποκείμενα μοτίβα στα δεδομένα.

2. Διαχωρισμός των δεδομένων σε σύνολα εκπαίδευσης και δοκιμής: Τα δεδομένα που συλλέγονται συνήθως χωρίζονται σε δύο μέρη: ένα σύνολο εκπαίδευσης, το οποίο χρησιμοποιείται για την εκπαίδευση του μοντέλου και ένα σύνολο δοκιμών και το σύνολο δοκιμής, το οποίο χρησιμοποιείται για την αξιολόγηση της απόδοσης του μοντέλου.
3. Καθορισμός της αρχιτεκτονικής του μοντέλου: Αφού συλλεχθούν και διαιρεθούν τα δεδομένα, το επόμενο βήμα είναι να καθοριστεί η αρχιτεκτονική του μοντέλου. Αυτό περιλαμβάνει την επιλογή του αριθμού και του τύπου των επιπέδων στο μοντέλο, καθώς και των συναρτήσεων ενεργοποίησης και των συναρτήσεων απώλειας που θα χρησιμοποιήσει το μοντέλο.
4. Εκπαίδευση του μοντέλου: Το μοντέλο εκπαιδεύεται χρησιμοποιώντας τα δεδομένα εκπαίδευσης. Κατά τη διάρκεια της εκπαίδευσης, το μοντέλο τροφοδοτείται με τα δεδομένα εισόδου και οι προβλέψεις του συγκρίνονται με την πραγματική έξοδο χρησιμοποιώντας μια συνάρτηση απώλειας. Ο στόχος είναι να ρυθμιστούν τα βάρη του μοντέλου έτσι ώστε η διαφορά μεταξύ της προβλεπόμενης παραγωγής και της πραγματικής παραγωγής να ελαχιστοποιείται. Αυτό γίνεται συνήθως με τη χρήση αλγορίθμων βελτιστοποίησης, όπως το gradient descent ή οι παραλλαγές του.
5. Αξιολόγηση του μοντέλου: Μετά την εκπαίδευση του μοντέλου, η απόδοσή του αξιολογείται χρησιμοποιώντας τα δεδομένα δοκιμών. Οι συνήθεις μετρήσεις που χρησιμοποιούνται για την αξιολόγηση της απόδοσης ενός μοντέλου περιλαμβάνουν την ακρίβεια, την ανάκληση, τη βαθμολογία F1 και την AUC.
6. Βελτίωση το μοντέλου: Εάν η απόδοση του μοντέλου δεν είναι ικανοποιητική, η αρχιτεκτονική του μοντέλου ή η διαδικασία εκπαίδευσης μπορούν να προσαρμοστούν και το μοντέλο μπορεί να επανεκπαιδευτεί. Αυτή η διαδικασία επαναλαμβάνεται μέχρι να επιτευχθεί το επιθυμητό επίπεδο απόδοσης.
7. Ανάπτυξη του μοντέλου: Αφού το μοντέλο έχει εκπαιδευτεί και αξιολογηθεί, μπορεί να αναπτυχθεί σε πραγματικό σενάριο για να γίνουν προβλέψεις για νέα δεδομένα.

Η εκπαίδευση ενός αλγόριθμου μηχανικής μάθησης περιλαμβάνει τη συλλογή ενός συνόλου δεδομένων, τη διαίρεση των δεδομένων σε σύνολα εκπαίδευσης και δοκιμής, τον καθορισμό της αρχιτεκτονικής του μοντέλου, την εκπαίδευση του μοντέλου, την αξιολόγηση του μοντέλου, τη λεπτομέρεια του μοντέλου εάν χρειάζεται και την ανάπτυξη του μοντέλου. Ο στόχος της εκπαίδευσης είναι να προσαρμόσει τις παραμέτρους του μοντέλου έτσι ώστε να μπορεί να κάνει ακριβείς προβλέψεις για νέα δεδομένα.

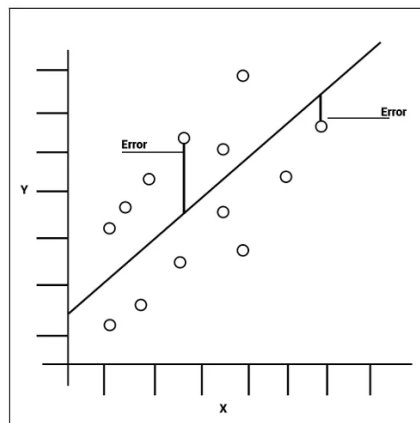
1.1.3 Συναρτήσεις Απώλειας (Loss Functions)

Μέσο Τετραγωνικό σφάλμα (Mean Squared Error)

Το μέσο τετραγωνικό σφάλμα (MSE) είναι μια συνάρτηση απώλειας που χρησιμοποιείται συνήθως για προβλήματα παλινδρόμησης. Μετρά τη διαφορά μεταξύ της προβλεπόμενης παραγωγής και της πραγματικής παραγωγής σε ένα μοντέλο παλινδρόμησης. Το MSE υπολογίζεται ως ο μέσος όρος των τετραγωνικών διαφορών μεταξύ της προβλεπόμενης παραγωγής και της πραγματικής παραγωγής, σε όλα τα δείγματα του συνόλου δεδομένων. Μαθηματικά, εκφράζεται ως:

$$MSE = (1/n) * \Sigma(y_i - \hat{y}_i)^2$$

όπου n , είναι ο αριθμός των δειγμάτων y_i , είναι η πραγματική έξοδος, \hat{y} είναι η προβλεπόμενη έξοδος και i το δείγμα.

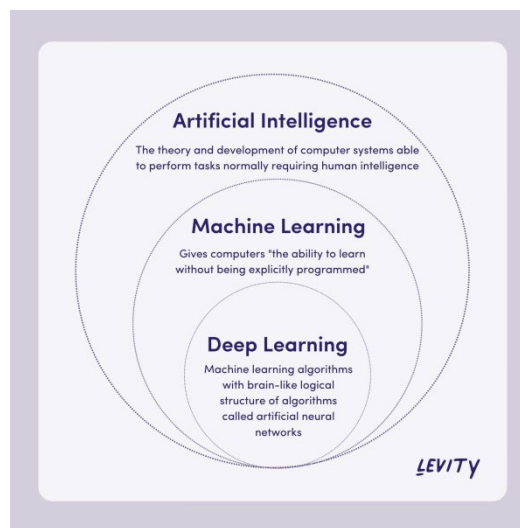


Εικόνα 1.3 : Παράδειγμα έρευνας σφάλματος με την χρήση του μέσου τετραγωνικού σφάλματος

Η συνάρτηση απώλειας MSE είναι ένα μέτρο του πόσο καλά ταιριάζει το μοντέλο στα δεδομένα. Ένα χαμηλό MSE υποδηλώνει καλή προσαρμογή, ενώ ένα υψηλό MSE δείχνει κακή προσαρμογή. Ο στόχος της εκπαίδευσης ενός μοντέλου παλινδρόμησης είναι η ελαχιστοποίηση του MSE. Το MSE είναι μια διαφοροποιήσιμη συνάρτηση απώλειας, που σημαίνει ότι η κλίση της σε σχέση με τις παραμέτρους του μοντέλου μπορεί να υπολογιστεί, κάτι που είναι χρήσιμο για την εκπαίδευση του μοντέλου χρησιμοποιώντας αλγόριθμους βελτιστοποίησης. Γενικότερα, το μέσο τετραγωνικό σφάλμα μετρά τη διαφορά μεταξύ της προβλεπόμενης παραγωγής και της πραγματικής παραγωγής σε ένα μοντέλο παλινδρόμησης και υπολογίζεται ως ο μέσος όρος των τετραγωνικών διαφορών μεταξύ της προβλεπόμενης παραγωγής και της πραγματικής παραγωγής. Ο στόχος της εκπαίδευσης ενός μοντέλου παλινδρόμησης είναι η ελαχιστοποίηση του MSE, που είναι μια διαφοροποιήσιμη συνάρτηση απώλειας που μπορεί να χρησιμοποιηθεί για βελτιστοποίηση.

1.2 Βαθιά Μάθηση (Deep Learning)

Η βαθιά μάθηση είναι ένα υποπεδίο της μηχανικής μάθησης που εμπνέεται από τη δομή και τη λειτουργία του ανθρώπινου εγκεφάλου, γνωστά ως τεχνητά νευρωνικά δίκτυα. Χρησιμοποιεί πολλαπλά στρώματα τεχνητών νευρώνων, που ονομάζονται κόμβοι, για την επεξεργασία και ανάλυση πολύπλοκων δεδομένων. Κάθε κόμβος σε ένα δίκτυο βαθιάς εκμάθησης συνδέεται με πολλούς άλλους κόμβους και οι συνδέσεις μεταξύ των κόμβων έχουν σχετικά βάρη που καθορίζουν την ισχύ της σύνδεσης. Κατά τη διάρκεια της εκπαίδευσης, τα βάρη των συνδέσεων προσαρμόζονται έτσι ώστε το δίκτυο να μπορεί να μάθει να προβλέπει με ακρίβεια την έξοδο για μια δεδομένη είσοδο.



Εικόνα 1.4 : Έννοιες Τεχνητής Νοημοσύνης

Υπάρχουν διάφοροι τύποι δικτύων βαθιάς μάθησης, συμπεριλαμβανομένων των επαναλαμβανόμενων νευρωνικών δικτύων (RNN) και των συνελκτικών νευρωνικών δικτύων (CNN). Τα δίκτυα βαθιάς μάθησης εκπαιδεύονται χρησιμοποιώντας μεγάλες ποσότητες δεδομένων με ετικέτα και είναι σε θέση να μάθουν πολύπλοκα μοτίβα στα δεδομένα μέσω πολλαπλών επαναλήψεων διάδοσης προς τα εμπρός και προς τα πίσω. Κατά τη διάδοση προς τα εμπρός, η είσοδος διέρχεται μέσω του δικτύου και η έξοδος υπολογίζεται. Κατά τη διάδοση προς τα πίσω, το σφάλμα μεταξύ της προβλεπόμενης εξόδου και της πραγματικής εξόδου χρησιμοποιείται για την προσαρμογή των βαρών των συνδέσεων μεταξύ των κόμβων.

1.2.1 Διαφορές Κλασικής Μηχανικής Μάθησης και Βαθιάς Μάθησης

Όπως και προαναφέρθηκε, η βαθιά μάθηση είναι μέρος της μηχανικής μάθησης, αναφέρθηκαν επίσης κάποια θετικά και αρνητικά χαρακτηριστικά και για τις δύο περιπτώσεις. Τώρα θα δούμε κάποιες διαφορές μεταξύ των δύο.

- Οι αλγόριθμοι βαθιάς μάθησης είναι πιο περίπλοκοι από αυτούς της μηχανικής μάθησης και δεν έχουν ανάγκη την ανθρώπινη παρέμβαση.
- Η ανάγκη για δυνατότερο υπολογιστικό σύστημα στα μοντέλα βαθιάς μάθησης. Αυτό ισχύει λόγω του πιο περίπλοκου κώδικα, όπως και τον μεγαλύτερο όγκο δεδομένων.
- Οι αλγόριθμοι κλασικής μηχανικής μάθησης χρειάζονται λιγότερο χρόνο για να εκτελεστούν .

Πλεονεκτήματα Βαθιάς Μάθησης

1. Τα μοντέλα βαθιάς μάθησης μπορούν να εκτελέσουν χιλιάδες επαναλαμβανόμενες δραστηριότητες σε μικρό χρονικό διάστημα, εφόσον δεν λειτουργούν με ακατέργαστα προϊόντα. Επίσης όσες και να είναι οι ενέργειες που θα εκτελέσει, η ποιότητα του αποτελέσματος δεν θα υποβαθμιστεί.
2. Η βαθιά μάθηση έχει το πλεονέκτημα ότι μπορεί να αναλύει μεγάλους όγκους δεδομένων και να πραγματοποιεί μεγάλους υπολογισμούς με μικρό κόστος. Επίσης μπορεί να εντοπίσει μικρές διαφορές στα δεδομένα και να τα διαφοροποιήσει, με αποτέλεσμα την θετική επίδραση στην παραγωγικότητα του μοντέλου.
3. Τα ακατέργαστα δεδομένα προκαλούν συνήθως πρόβλημα στα μοντέλα μηχανικής μάθησης. Αυτό λύνεται με την χρήση του Feature Engineering, το οποίο χαρακτηρίζει, ή βάζει ταμπέλες θα μπορούσαμε να πούμε στα δεδομένα μας. Η βαθιά μάθηση όμως έχει την δυνατότητα να αναλύει και να συσχετίζει τα δεδομένα που θα προσφέρουμε μόνη της.

Μειονεκτήματα Βαθιάς Μάθησης

1. Ο όγκος δεδομένων που χρειάζεται το μοντέλο είναι κάτι που θα συζητηθεί και στην υλοποίηση του πειράματος. Για να λειτουργήσει ο αλγόριθμος στο 100% του χρειάζεται έναν τεράστιο όγκο δεδομένων για την εκπαίδευση του. Αυτό δυσκολεύει την κατάσταση για την αποθήκευση αυτών των δεδομένων. Στην περίπτωση μας, οι τιμές των μετοχών αποθηκεύονται σε αρχεία μικρού όγκου, διότι είναι απλά πίνακες με νούμερα. Σε έναν αλγόριθμο για αναγνώριση φωνής όμως θα υπάρχει πρόβλημα λόγω του όγκου των αρχείων.
2. Ένα εξίσου σημαντικό ζήτημα είναι ότι τα μοντέλα αυτά απαιτούν μεγάλη υπολογιστική ισχύ. Αυτό σημαίνει ότι ένα καθημερινό σύστημα θα δυσκολευτεί να ανταποδώσει σε έναν πολύπλοκο αλγόριθμο καθυστερώντας την εκτέλεση του.

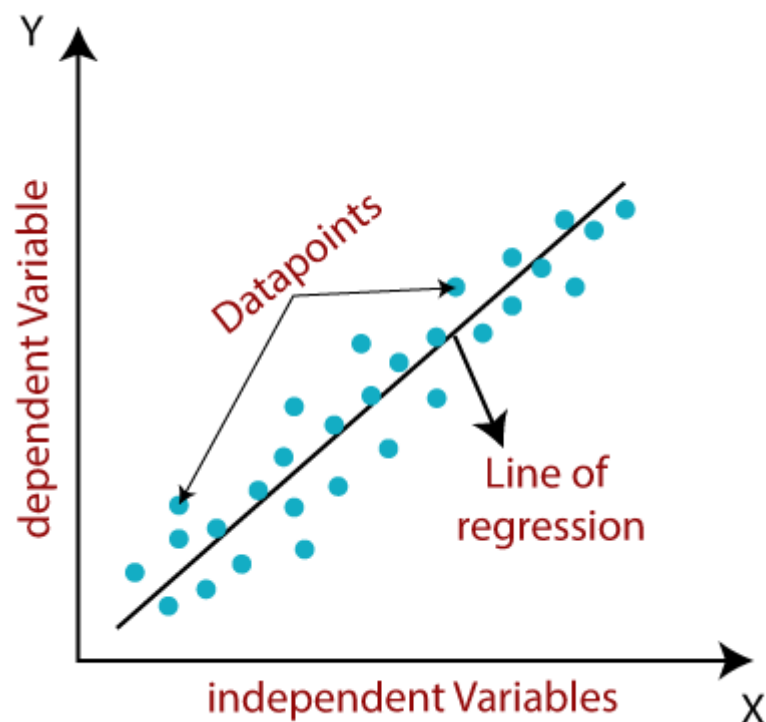
2. Αλγόριθμοι Κλασικής Μηχανικής Μάθησης

2.1 Γραμμική Παλινδρόμηση (Linear Regression)

Η Γραμμική Παλινδρόμηση είναι μια δημοφιλής και απλή στατιστική μέθοδος που χρησιμοποιείται για τη μοντελοποίηση της σχέσης μεταξύ μιας εξαρτημένης μεταβλητής (επίσης γνωστή ως μεταβλητή απόκρισης ή αποτελέσματος) και μιας ή περισσότερων ανεξάρτητων μεταβλητών (γνωστές και ως προγνωστικές ή επεξηγηματικές μεταβλητές). Ο στόχος της γραμμικής παλινδρόμησης είναι να χωρέσει μια γραμμή που αντιπροσωπεύει καλύτερα τη σχέση μεταξύ των μεταβλητών. Η γραμμική παλινδρόμηση βασίζεται στην υπόθεση ότι η σχέση μεταξύ των ανεξάρτητων και των εξαρτημένων μεταβλητών είναι γραμμική. Αυτό σημαίνει ότι μια αλλαγή στις ανεξάρτητες μεταβλητές συνδέεται με μια σταθερή αλλαγή στην εξαρτημένη μεταβλητή. Η γραμμή που ταιριάζει στα σημεία δεδομένων αντιπροσωπεύεται από μια εξίσωση που έχει τη μορφή:

$$y = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n$$

y είναι η εξαρτημένη μεταβλητή, x_1, x_2, \dots, x_n είναι οι ανεξάρτητες μεταβλητές, b_0 είναι η τομή, b_1, b_2, \dots, b_n είναι οι συντελεστές που σχετίζονται με τις ανεξάρτητες μεταβλητές.



Εικόνα 2.1 : Παράδειγμα συνάρτησης της Γραμμικής Παλινδρόμησης

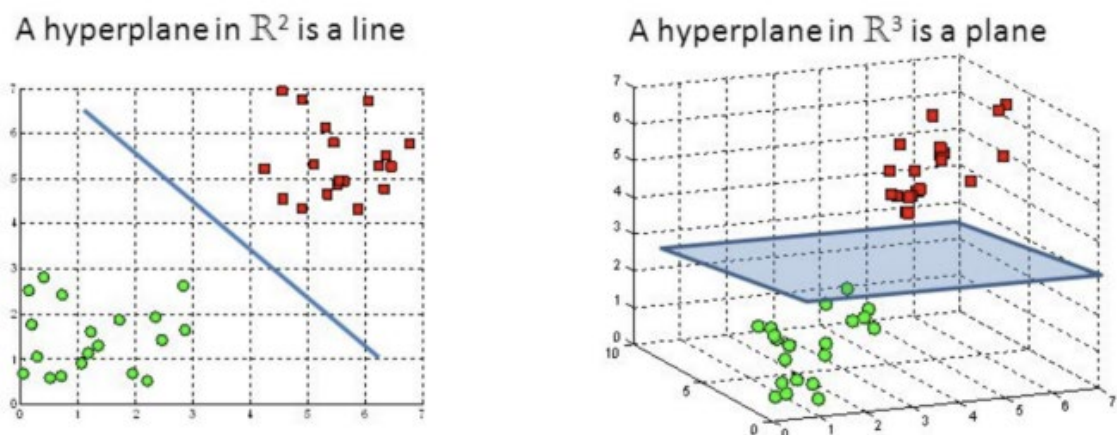
Αυτοί οι συντελεστές αντιπροσωπεύουν τη μεταβολή της εξαρτημένης μεταβλητής που σχετίζεται με μια μεταβολή μιας μονάδας στην ανεξάρτητη μεταβλητή. Οι συντελεστές υπολογίζονται χρησιμοποιώντας μια στατιστική μέθοδο που ονομάζεται μέθοδος ελάχιστων τετραγώνων, η οποία εντοπίζει τη γραμμή που ελαχιστοποιεί το άθροισμα των τετραγωνικών

διαφορών μεταξύ των παρατηρούμενων και των προβλεπόμενων τιμών της εξαρτημένης μεταβλητής. Μόλις υπολογιστούν οι συντελεστές, το μοντέλο γραμμικής παλινδρόμησης μπορεί να χρησιμοποιηθεί για να γίνουν προβλέψεις σχετικά με την εξαρτημένη μεταβλητή με βάση τις νέες τιμές των ανεξάρτητων μεταβλητών.

Η γραμμική παλινδρόμηση είναι μια απλή και ισχυρή μέθοδος για τη μοντελοποίηση σχέσεων μεταξύ μεταβλητών. Ωστόσο, έχει κάποιους περιορισμούς. Για παράδειγμα, υποθέτει ότι η σχέση μεταξύ των μεταβλητών είναι γραμμική, κάτι που μπορεί να μην συμβαίνει πάντα. Επιπλέον, υποθέτει ότι οι ανεξάρτητες μεταβλητές δεν σχετίζονται μεταξύ τους, κάτι που μπορεί να μην ισχύει στην πράξη. Για να ξεπεραστούν αυτοί οι περιορισμοί, μπορούν να χρησιμοποιηθούν πιο προηγμένες μέθοδοι, όπως η πολυωνυμική παλινδρόμηση και η πολλαπλή γραμμική παλινδρόμηση.

2.2 Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines)

Οι μηχανές διανυσμάτων υποστήριξης (SVM) είναι ένας τύπος αλγόριθμου μηχανικής μάθησης που χρησιμοποιείται για εργασίες ταξινόμησης και παλινδρόμησης. Η θεμελιώδης ιδέα πίσω από τα SVM είναι να βρεθεί ένα υπερεπίπεδο σε ένα χώρο υψηλών διαστάσεων που διαχωρίζει τα σημεία δεδομένων σε διαφορετικές κατηγορίες όσο το δυνατόν καθαρότερα.



Εικόνα 2.2 : Υπερεπίπεδο στον δισδιάστατο και στον τρισδιάστατο χώρο

Το υπερεπίπεδο επιλέγεται με τέτοιο τρόπο ώστε να μεγιστοποιεί το περιθώριο μεταξύ των δύο πλησιέστερων σημείων από διαφορετικές κλάσεις. Αυτό το περιθώριο αντιπροσωπεύει την απόσταση μεταξύ του υπερεπίπεδου και των πλησιέστερων σημείων δεδομένων από κάθε κατηγορία. Τα σημεία που βρίσκονται πιο κοντά στο υπερεπίπεδο ονομάζονται διανύσματα υποστήριξης και είναι τα μόνα που έχουν σημασία στη διαδικασία βελτιστοποίησης. Η διαδικασία βελτιστοποίησης στα SVM περιλαμβάνει την εύρεση του υπερεπίπεδου που όχι μόνο διαχωρίζει τα σημεία δεδομένων αλλά και μεγιστοποιεί το περιθώριο. Αυτό επιτυγχάνεται με την επίλυση ενός προβλήματος τετραγωνικού προγραμματισμού, το οποίο περιλαμβάνει την ελαχιστοποίηση μιας συνάρτησης κόστους

που υπόκειται σε ορισμένους περιορισμούς. Η συνάρτηση κόστους τιμωρεί τις εσφαλμένες ταξινομήσεις και ενθαρρύνει ένα μεγαλύτερο περιθώριο μεταξύ των κατηγοριών.

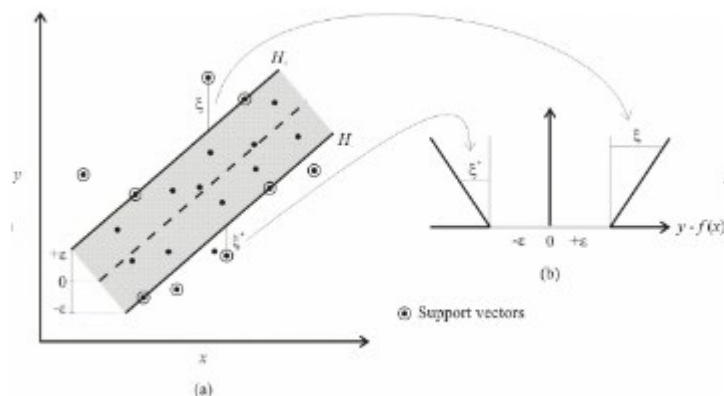
Τα SVM είναι ιδιαίτερα χρήσιμα όταν ασχολούμαστε με δεδομένα υψηλών διαστάσεων, όπου άλλοι αλγόριθμοι ταξινόμησης μπορεί να μην λειτουργούν καλά. Είναι επίσης αποτελεσματικά σε περιπτώσεις όπου τα δεδομένα δεν διαχωρίζονται γραμμικά. Σε τέτοιες περιπτώσεις, τα SVM χρησιμοποιούν μια τεχνική που ονομάζεται kernel trick, η οποία χαρτογραφεί τα σημεία δεδομένων σε έναν χώρο υψηλότερης διάστασης όπου γίνονται γραμμικά διαχωρίσιμα.

Συνάρτηση Απώλειας (Loss Function)

Η συνάρτηση απώλειας που χρησιμοποιείται για τις μηχανές υποστήριξης διανυσμάτων (SVM) σε εργασίες παλινδρόμησης είναι συνήθως η συνάρτηση απώλειας, γνωστή ως απώλεια Vapnik ή απώλεια χωρίς ευαισθησία. Αυτή η λειτουργία απώλειας έχει σχεδιαστεί για να χειρίζεται θορυβώδη δεδομένα και ακραίες τιμές, που είναι κοινά σε προβλήματα παλινδρόμησης. Η συνάρτηση απώλειας Vapnik ορίζεται ως εξής :

$$L(y, f(x)) = \max(0, |y - f(x)| - \epsilon)$$

όπου y είναι η πραγματική τιμή στόχου, $f(x)$ είναι η προβλεπόμενη τιμή στόχος και το ϵ είναι μια μη αρνητική σταθερά που ονομάζεται ανοχή.



Εικόνα 2.3 : Σχηματική εφαρμογή της συνάρτησης Vapnik

Η ιδέα πίσω από τη συνάρτηση απώλειας είναι να τιμωρούνται μόνο εκείνες οι προβλέψεις που βρίσκονται εκτός της μη ευαίσθητης ζώνης. Οι προβλέψεις εντός της μη ευαίσθητης ζώνης δεν τιμωρούνται, καθώς θεωρούνται αρκετά ακριβείς. Με αυτόν τον τρόπο, ο αλγόριθμος εστιάζει στην εύρεση ενός υπερεπίπεδου που μπορεί να ταιριάζει καλύτερα στα δεδομένα, ενώ αγνοεί μικρές αποκλίσεις. Στην παλινδρόμηση SVM, ο στόχος είναι να βρεθεί το υπερεπίπεδο που ελαχιστοποιεί το άθροισμα της συνάρτησης Vapnik και ένας όρος τακτοποίησης που τιμωρεί την πολυπλοκότητα του μοντέλου.

Συναρτήσεις Πυρήνα (Kernel Functions)

- Συνάρτηση Ακτινικής Βάσης (Radial Basis Function RBF): Ο RBF είναι ένας από τους πιο συχνά χρησιμοποιούμενους kernels στα SVM. Υπολογίζει την ομοιότητα μεταξύ δύο σημείων δεδομένων με βάση την απόστασή τους το ένα από το άλλο. Ο RBF μπορεί να μετατρέψει τα δεδομένα εισόδου σε έναν απεριόριστο χώρο χαρακτηριστικών, ο οποίος επιτρέπει στον αλγόριθμο να προσαρμόζει σύνθετα μη γραμμικά όρια απόφασης. Η συνάρτηση ορίζεται ως:

$$K(x, x') = \exp(-\gamma \|x - x'\|^2)$$

όπου x και x' είναι δύο σημεία δεδομένων, $\|...\|$ υποδηλώνει την Ευκλείδεια απόσταση και το γ είναι μια υπερπαράμετρος που ελέγχει το πλάτος της συνάρτησης. Μια μικρότερη τιμή του γ έχει ως αποτέλεσμα έναν ευρύτερο πυρήνα, που σημαίνει ότι περισσότερα σημεία δεδομένων θεωρούνται παρόμοια, ενώ μια μεγαλύτερη τιμή του γ οδηγεί σε έναν στενότερο kernel, που σημαίνει ότι μόνο τα κοντινά σημεία δεδομένων θεωρούνται παρόμοια.

- Πολυωνυμικός πυρήνας (Polynomial Kernel): Ο πολυωνυμικός πυρήνας είναι ένας άλλος συχνά χρησιμοποιούμενος kernel στα SVM. Μετατρέπει τα δεδομένα εισόδου σε έναν χώρο χαρακτηριστικών υψηλότερης διάστασης υπολογίζοντας όλους τους πιθανούς συνδυασμούς των χαρακτηριστικών εισόδου μέχρι ένα ορισμένο βαθμό. Η συνάρτηση ορίζεται ως:

$$K(x, x') = (\gamma \langle x, x' \rangle + r)^d$$

όπου x και x' είναι δύο σημεία δεδομένων, γ είναι ένας παράγοντας κλιμάκωσης, r είναι μια σταθερά, d είναι ο βαθμός του πολυωνύμου και $\langle x, x' \rangle$ δηλώνει το γινόμενο κουκίδων των δύο σημείων δεδομένων. Ο πολυωνυμικός πυρήνας μπορεί να χειριστεί μη γραμμικά όρια απόφασης αυξάνοντας τον βαθμό του πολυωνύμου.

- Γραμμικός πυρήνας (Linear Kernel): Ο γραμμικός πυρήνας είναι ο απλούστερος πυρήνας στα SVM. Δεν μετασχηματίζει τα δεδομένα εισόδου, αλλά υπολογίζει το γινόμενο κουκίδων των χαρακτηριστικών εισόδου. Η συνάρτηση πυρήνα ορίζεται ως:

$$K(x, x') = \langle x, x' \rangle$$

όπου x και x' είναι δύο σημεία δεδομένων και $\langle x, x' \rangle$ δηλώνει το γινόμενο κουκίδων των δύο σημείων δεδομένων. Ο γραμμικός kernel λειτουργεί καλά για γραμμικά διαχωρίσιμα δεδομένα, αλλά δεν μπορεί να χειριστεί πολύπλοκα μη γραμμικά δεδομένα.

- Σιγμοειδής πυρήνας (Sigmoid Kernel): Ο σιγμοειδής πυρήνας είναι ένας μη γραμμικός πυρήνας που μετασχηματίζει τα δεδομένα εισόδου χρησιμοποιώντας τη σιγμοειδή συνάρτηση (Sigmoid Function). Η σιγμοειδής συνάρτηση αντιστοιχίζει τα

δεδομένα εισόδου σε μια τιμή μεταξύ 0 και 1, η οποία μπορεί να ερμηνευθεί ως πιθανότητα. Η συνάρτηση ορίζεται ως:

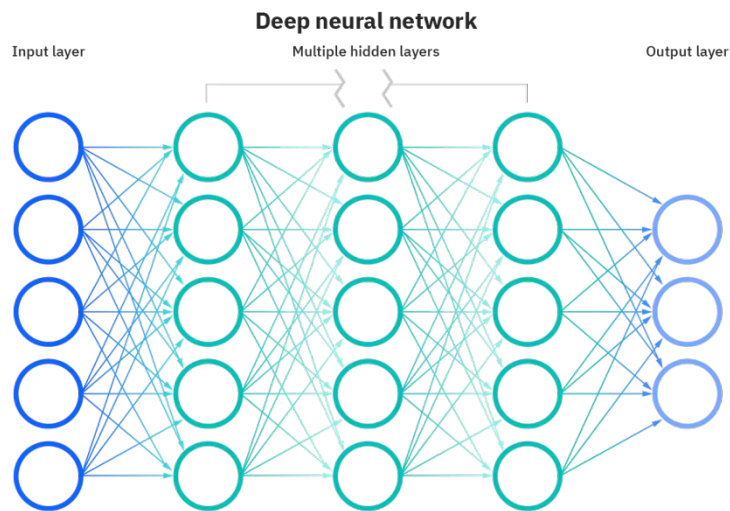
$$K(x, x') = \tanh(\gamma \langle x, x' \rangle + r)$$

όπου x και x' είναι δύο σημεία δεδομένων, το γ είναι ένας παράγοντας κλιμάκωσης, το r είναι μια σταθερά και το \tanh υποδηλώνει την υπερβολική συνάρτηση εφαπτομένης (Hyperbolic Tangent Function). Ο σιγμοειδής πυρήνας μπορεί να χρησιμοποιηθεί για να χειριστεί μη γραμμικά όρια, αλλά είναι ευαίσθητος στην επιλογή των υπερπαραμέτρων.

3. Αλγόριθμοι Βαθιάς Μάθησης

3.1 Νευρωνικά Δίκτυα

Τα νευρωνικά δίκτυα είναι ένας τύπος μοντέλου μηχανικής μάθησης που έχει πάρει την δομή του από τη δομή και τη λειτουργία του ανθρώπινου εγκεφάλου. Η κεντρική ιδέα πίσω από τα νευρωνικά δίκτυα είναι η κατασκευή ενός μοντέλου που μπορεί να αναγνωρίσει μοτίβα και σχέσεις μέσα στα δεδομένα, όπως ο ανθρώπινος εγκέφαλος μπορεί να αναγνωρίσει μοτίβα και σχέσεις στις πληροφορίες. Ένα νευρωνικό δίκτυο αποτελείται από διασυνδεδεμένους κόμβους, οι οποίοι ονομάζονται τεχνητοί νευρώνες και είναι οργανωμένοι σε στρώματα.



Εικόνα 3.1 : Νευρωνικό δίκτυο με 5 layers

Το επίπεδο εισόδου δέχεται τα ακατέργαστα δεδομένα εισόδου, ενώ το επίπεδο εξόδου παράγει την τελική πρόβλεψη. Τα επίπεδα μεταξύ των επιπέδων εισόδου και εξόδου ονομάζονται κρυφά επίπεδα και σε αυτά γίνονται οι πιο σημαντικοί υπολογισμοί. Κάθε τεχνητός νευρώνας λαμβάνει εισόδους από άλλους νευρώνες, εκτελεί έναν απλό υπολογισμό σε αυτές τις εισόδους και στη συνέχεια εξάγει ένα σήμα σε άλλους νευρώνες. Ο υπολογισμός που εκτελείται από κάθε νευρώνα ονομάζεται συνάρτηση ενεργοποίησης, η οποία μπορεί να είναι μια απλή μαθηματική πράξη, όπως μια σιγμοειδής ή βηματική συνάρτηση. Οι συναρτήσεις ενεργοποίησης βοηθούν το δίκτυο να εισάγει την μη γραμμικότητα στο μοντέλο, επιτρέποντάς του να καταγράφει σύνθετες σχέσεις μεταξύ των εισόδων και των εξόδων. Τα βάρη στις συνδέσεις μεταξύ των νευρώνων είναι οι πιο σημαντικές παράμετροι του νευρωνικού δικτύου, καθώς καθορίζουν τη σχετική σημασία κάθε εισόδου στην έξοδο.

Κατά την εκπαίδευση ενός νευρωνικού δικτύου, τα βάρη προσαρμόζονται έτσι ώστε να ελαχιστοποιείται το σφάλμα μεταξύ των προβλέψεων του μοντέλου και των πραγματικών τιμών εξόδου. Αυτή η διαδικασία προσαρμογής των βαρών είναι γνωστή ως οπισθοδιάδοση και είναι η πιο ευρέως χρησιμοποιούμενη μέθοδος για την εκπαίδευση των νευρωνικών δικτύων. Υπάρχουν πολλοί διαφορετικοί τύποι νευρωνικών δικτύων, το καθένα με τα δικά του πλεονεκτήματα και αδυναμίες. Μερικοί από τους πιο συνηθισμένους τύπους περιλαμβάνουν τα επαναλαμβανόμενα δίκτυα και τα συνελκτικά δίκτυα.

Τα επαναλαμβανόμενα δίκτυα έχουν σχεδιαστεί για να χειρίζονται διαδοχικά δεδομένα, όπως χρονοσειρές ή κείμενο φυσικής γλώσσας. Σε αυτά τα δίκτυα, εισάγονται συνδέσεις ανάδρασης, επιτρέποντας στο δίκτυο να χρησιμοποιεί πληροφορίες από προηγούμενα χρονικά βήματα όταν κάνει προβλέψεις. Τα συνελκτικά δίκτυα είναι εξειδικευμένα δίκτυα προώθησης που χρησιμοποιούνται για εργασίες αναγνώρισης εικόνας και βίντεο. Αυτά τα δίκτυα χρησιμοποιούν συνελκτικά και ομαδικά επίπεδα για να εξαγάγουν χαρακτηριστικά από τα δεδομένα εισόδου και στη συνέχεια χρησιμοποιούν πλήρως συνδεδεμένα επίπεδα για να κάνουν προβλέψεις.

Τα νευρωνικά δίκτυα έχουν γίνει μια από τις πιο ευρέως χρησιμοποιούμενες μεθόδους μηχανικής μάθησης και έχουν εφαρμοστεί σε ένα ευρύ φάσμα προβλημάτων, όπως ταξινόμηση εικόνων, αναγνώριση ομιλίας, επεξεργασία φυσικής γλώσσας και πρόβλεψη χρηματοοικονομικών δεδομένων.

3.1.1 Βελτιστοποιητές (Optimizers)

Οι βελτιστοποιητές είναι αλγόριθμοι που χρησιμοποιούνται στη μηχανική εκμάθηση για την προσαρμογή των παραμέτρων ενός μοντέλου προκειμένου να ελαχιστοποιηθεί η συνάρτηση απώλειας. Η συνάρτηση απώλειας μετρά τη διαφορά μεταξύ των προβλεπόμενων τιμών εξόδου και των πραγματικών τιμών εξόδου και ο στόχος της εκπαίδευσης είναι να βρεθεί το σύνολο των παραμέτρων του μοντέλου που έχει ως αποτέλεσμα τη μικρότερη δυνατή απώλεια. Υπάρχουν πολλοί διαφορετικοί τύποι βελτιστοποιητών, όπως:

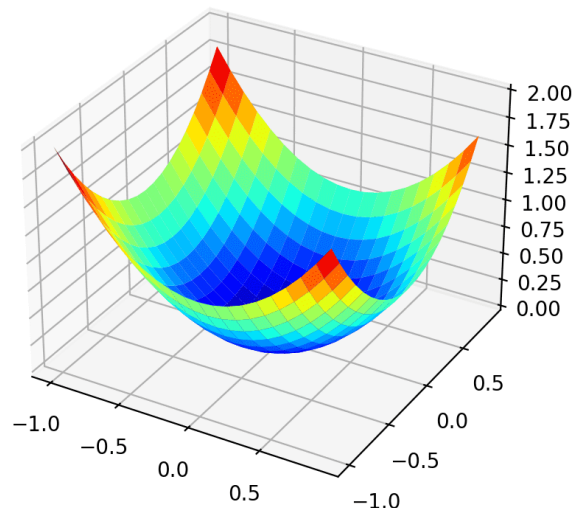
- **Stochastic Gradient Descent (SGD):** Πρόκειται για έναν απλό και ευρέως χρησιμοποιούμενο αλγόριθμο βελτιστοποίησης που χρησιμοποιεί τη διαβάθμιση της συνάρτησης απώλειας σε σχέση με τις παραμέτρους του μοντέλου για να ενημερώσει τις παραμέτρους προς την κατεύθυνση της ελάχιστης απώλειας.
- **Mini-batch Gradient Descent:** Αυτή είναι μια παραλλαγή του SGD που ενημερώνει τις παραμέτρους του μοντέλου με βάση ένα μικρό τυχαίο δείγμα των δεδομένων εκπαίδευσης, αντί για ολόκληρο το σύνολο δεδομένων. Αυτό επιτρέπει στο εργαλείο βελτιστοποίησης να κλιμακωθεί σε μεγάλα σύνολα δεδομένων και μπορεί να οδηγήσει σε ταχύτερη σύγκλιση.
- **Momentum:** Αυτό το εργαλείο βελτιστοποίησης προσθέτει έναν όρο “momentum” στον κανόνα ενημέρωσης, ο οποίος βοηθά το εργαλείο βελτιστοποίησης να ξεπεράσει τα τοπικά ελάχιστα και να συγκλίνει πιο γρήγορα.
- **Adagrad:** Αυτό το εργαλείο βελτιστοποίησης προσαρμόζει τον ρυθμό εκμάθησης με βάση την παράμετρο, γεγονός που του επιτρέπει να χειρίζεται αραιές κλίσεις και να βελτιώνει τη σύγκλιση για ορισμένα προβλήματα.

- Adadelta: Αυτό το εργαλείο βελτιστοποίησης είναι παρόμοιο με το Adagrad, αλλά αντί να προσαρμόζει τον ρυθμό εκμάθησης με βάση την παράμετρο, προσαρμόζει την κλίμακα της ενημέρωσης κλίσης.
- Adam: Ο Adam (Adaptive Moment Estimation) είναι ένας ευρέως χρησιμοποιούμενος αλγόριθμος βελτιστοποίησης στη βαθιά μάθηση. Είναι ένας συνδυασμός του αλγόριθμου gradient descent και του αλγόριθμου momentum και έχει σχεδιαστεί για να είναι ένας αποτελεσματικός και ισχυρός βελτιστοποιητής για την εκπαίδευση σε βαθιά νευρωνικά δίκτυα.

Αλγόριθμος βελτιστοποίησης Adam

Η βασική ιδέα πίσω από τον Adam είναι να διατηρηθεί ένας εκθετικός κινητός μέσος όρος της κλίσης και του τετραγώνου της κλίσης και να χρησιμοποιηθούν αυτοί οι κινητοί μέσοι όροι για να προσαρμόσετε δυναμικά τον ρυθμό εκμάθησης κατά τη διάρκεια της προπόνησης. Ο αλγόριθμος υπολογίζει προσαρμοστικούς ρυθμούς εκμάθησης για κάθε παράμετρο στο μοντέλο και ενημερώνει τις παραμέτρους χρησιμοποιώντας τον ακόλουθο κανόνα ενημέρωσης:

$$w_t + 1 = w_t - m_t * \frac{m_t}{(\sqrt{v_t} + \epsilon)}$$



Εικόνα 3.2 : Εφαρμογή της Gradient Descent με τον Adam

Ο Adam έχει πολλά πλεονεκτήματα σε σχέση με άλλους αλγόριθμους βελτιστοποίησης, όπως ότι είναι υπολογιστικά αποδοτικός, έχει καλές ιδιότητες σύγκλισης και είναι σε θέση να χειρίζεται αραιές κλίσεις. Είναι συχνά η προεπιλεγμένη επιλογή βελτιστοποιητή για την εκπαίδευση σε βαθιά νευρωνικά δίκτυα και έχει χρησιμοποιηθεί για την επίτευξη αποτελεσμάτων αιχμής σε ένα ευρύ φάσμα εργασιών μηχανικής μάθησης. Ωστόσο, όπως όλοι οι αλγόριθμοι βελτιστοποίησης, ο Adam δεν είναι τέλειος και μπορεί ακόμα να κολλήσει σε κακά τοπικά ελάχιστα ή να συγκλίνει πολύ αργά για ορισμένα προβλήματα. Σε αυτές τις περιπτώσεις, μπορεί να χρειαστεί να δοκιμαστούν διαφορετικοί βελτιστοποιητές για να επιτευχθούν καλύτερα αποτελέσματα. Κάθε βελτιστοποιητής έχει τα δικά του δυνατά και αδύνατα σημεία και η επιλογή του βελτιστοποιητή εξαρτάται από το συγκεκριμένο πρόβλημα και τη φύση των δεδομένων. Γενικά, διαφορετικοί βελτιστοποιητές μπορούν να παράγουν διαφορετικά αποτελέσματα και συχνά είναι καλή ιδέα να δοκιμάζονται πολλοί διαφορετικοί για να φανεί ποιος λειτουργεί καλύτερα για ένα συγκεκριμένο πρόβλημα.

Κατάβαση Κλίσης (Gradient Descent)

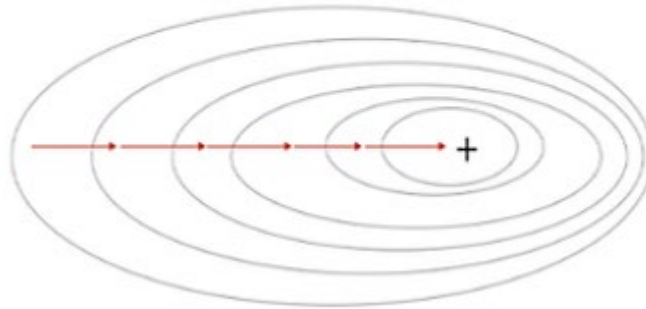
Η κατάβαση κλίσης είναι ένας αλγόριθμος βελτιστοποίησης που χρησιμοποιείται για την εκπαίδευση μοντέλων μηχανικής μάθησης. Είναι μια μέθοδος βελτιστοποίησης πρώτης τάξης που στοχεύει στην εύρεση των βέλτιστων τιμών των παραμέτρων του μοντέλου που ελαχιστοποιούν τη συνάρτηση κόστους. Στον Gradient Descent, ο αλγόριθμος ξεκινά αρχικοποιώντας τυχαία τις παραμέτρους του μοντέλου και στη συνέχεια τις ενημερώνει επαναληπτικά για να ελαχιστοποιήσει τη συνάρτηση κόστους. Σε κάθε επανάληψη, ο αλγόριθμος υπολογίζει τη διαβάθμιση της συνάρτησης κόστους σε σχέση με τις παραμέτρους του μοντέλου και τις ενημερώνει προς την αντίθετη κατεύθυνση της κλίσης. Ο κανόνας ενημέρωσης για την κατάβαση κλίσης είναι ο εξής:

$$\theta = \theta - \alpha * \nabla_{\theta} J(\theta)$$

Το θ είναι το διάνυσμα των παραμέτρων του μοντέλου, α είναι ο ρυθμός εκμάθησης (μέγεθος βήματος) και $\nabla_{\theta} J(\theta)$ είναι η κλίση της συνάρτησης κόστους σε σχέση με τις παραμέτρους του μοντέλου. Ο ρυθμός εκμάθησης καθορίζει το μέγεθος των βημάτων που γίνονται προς την κατεύθυνση της αρνητικής κλίσης. Είναι μια υπερπαραμέτρος που πρέπει να συντονιστεί για να βρεθεί η βέλτιστη τιμή για το συγκεκριμένο πρόβλημα. Εάν ο ρυθμός εκμάθησης είναι πολύ μικρός, ο αλγόριθμος μπορεί να χρειαστεί πολύ χρόνο για να συγκλίνει, ενώ εάν είναι πολύ μεγάλος, ο αλγόριθμος μπορεί να μην συγκλίνει ή ακόμη και να αποκλίνει.

Το Gradient Descent μπορεί να χρησιμοποιηθεί με διαφορετικές παραλλαγές της συνάρτησης κόστους, όπως το μέσο τετραγωνικό σφάλμα, ανάλογα με το συγκεκριμένο πρόβλημα που αντιμετωπίζεται. Η συνάρτηση κόστους μετρά πόσο καλά ταιριάζει το μοντέλο στα δεδομένα εκπαίδευσης και χρησιμεύει ως ο στόχος που ο αλγόριθμος προσπαθεί να ελαχιστοποιήσει. Ένα από τα πλεονεκτήματα της κατάβασης κλίσης είναι ότι είναι ένας απλός και επεκτάσιμος αλγόριθμος που μπορεί να εφαρμοστεί σε ένα ευρύ φάσμα προβλημάτων μηχανικής μάθησης. Ωστόσο, μπορεί να υποφέρει από αργή σύγκλιση, ειδικά σε χώρους υψηλών διαστάσεων με πολύπλοκες συναρτήσεις κόστους. Για να μετριαστεί αυτό, μπορούν να χρησιμοποιηθούν παραλλαγές της κατάβασης κλίσης, όπως η στοχαστική κατάβαση κλίσης, η οποία ενημερώνει τις παραμέτρους του μοντέλου χρησιμοποιώντας μικρότερες παρτίδες δεδομένων εκπαίδευσης και όχι ολόκληρο το σύνολο δεδομένων.

Gradient Descent



Εικόνα 3.3 : Παράδειγμα εκπαίδευσης της κατάβασης κλίσης

Στοχαστική Κατάβαση Κλίσης (Stochastic Gradient Descent)

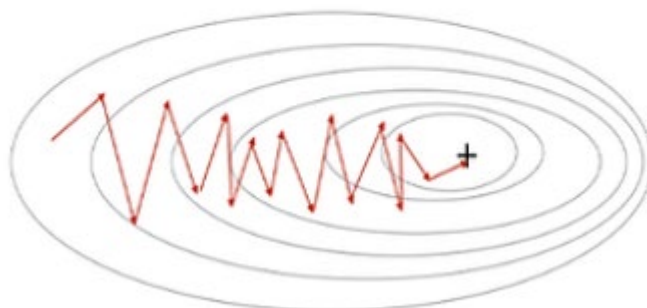
Η στοχαστική κατάβαση κλίσης (SGD) είναι ένας αλγόριθμος βελτιστοποίησης που χρησιμοποιείται ευρέως στη μηχανική εκμάθηση για την εκπαίδευση μεγάλων συνόλων δεδομένων. Πρόκειται για μία παραλλαγή του αλγόριθμου κατάβασης κλίσης, ο οποίος ενημερώνει τις παραμέτρους του μοντέλου σε μικρές παρτίδες αντί να ενημερώνει όλες ταυτόχρονα. Στον SGD, οι παράμετροι του μοντέλου ενημερώνονται χρησιμοποιώντας ένα μόνο τυχαία επιλεγμένο παράδειγμα εκπαίδευσης, αντί να χρησιμοποιούν ολόκληρο το σετ εκπαίδευσης. Αυτή η προσέγγιση μειώνει το υπολογιστικό κόστος του υπολογισμού της κλίσης, το οποίο μπορεί να είναι σημαντικό σε μεγάλα σύνολα δεδομένων. Επιπλέον, η χρήση τυχαίων παρτίδων μπορεί να εισάγει θόρυβο που μπορεί να βοηθήσει τον αλγόριθμο να ξεφύγει από τα τοπικά ελάχιστα, με αποτέλεσμα την ταχύτερη σύγκλιση. Ο κανόνας ενημέρωσης για το SGD είναι παρόμοιος με αυτόν της κατάβασης κλίσης, αλλά αντί να υπολογίζει τη διαβάθμιση της συνάρτησης κόστους σε σχέση με όλα τα δεδομένα εκπαίδευσης, υπολογίζει τη διαβάθμιση σε σχέση με ένα τυχαία επιλεγμένο υποσύνολο των δεδομένων. Ο κανόνας ενημέρωσης για το SGD έχει ως εξής:

$$\theta = \theta - \alpha * \nabla_{\theta} J(\theta; x(i); y(i))$$

Το θ είναι το διάνυσμα των παραμέτρων του μοντέλου, α είναι ο ρυθμός εκμάθησης (μέγεθος βήματος) και $\nabla_{\theta} J(\theta; x(i); y(i))$ είναι η κλίση της συνάρτησης κόστους σε σχέση με τις παραμέτρους του μοντέλου, που υπολογίζεται χρησιμοποιώντας ένα τυχαία επιλεγμένο παράδειγμα εκπαίδευσης $(x(i), y(i))$. Η διαδικασία επιλογής τυχαίων παραδειγμάτων από το σετ εκπαίδευσης επαναλαμβάνεται πολλές φορές μέχρι να επιτευχθεί σύγκλιση. Σε κάθε επανάληψη, ο αλγόριθμος ενημερώνει τις παραμέτρους του μοντέλου με βάση τη διαβάθμιση που υπολογίζεται από τα τυχαία επιλεγμένα παραδείγματα.

Ένα από τα κύρια πλεονεκτήματα του SGD είναι η ικανότητά του να χειρίζεται μεγάλα σύνολα δεδομένων αποτελεσματικά. Χρησιμοποιώντας μικρές παρτίδες δεδομένων, ο αλγόριθμος μπορεί να επεξεργαστεί μεγάλο όγκο δεδομένων με μικρότερο υπολογιστικό κόστος, γεγονός που καθιστά πιο εφικτή τη χρήση του σε εργασίες μηχανικής μάθησης μεγάλης κλίμακας. Επιπλέον, η χρήση τυχαίων παρτίδων μπορεί επίσης να βοηθήσει τον

Stochastic Gradient Descent



Εικόνα 3.4 : Παράδειγμα εκπαίδευσης της στοχαστικής κατάβασης κλίσης

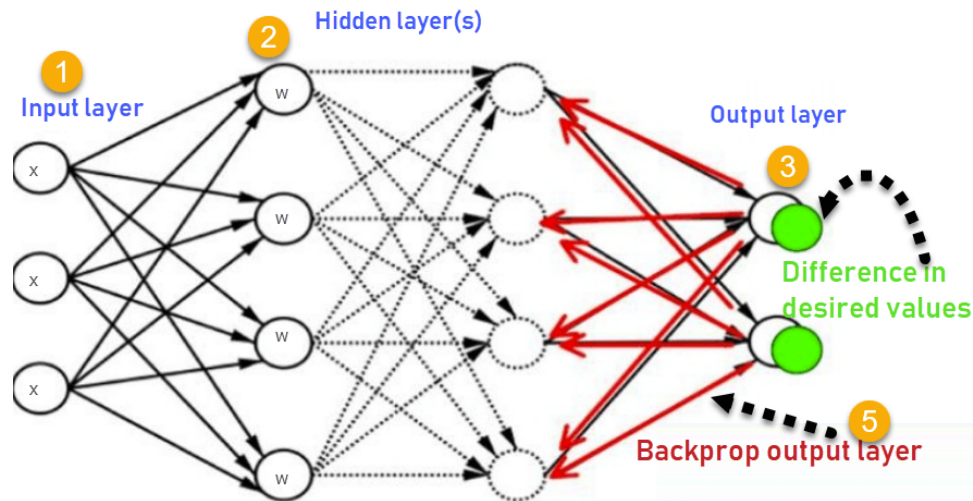
αλγόριθμο να ξεφύγει από τα τοπικά ελάχιστα, γεγονός που μπορεί να οδηγήσει σε καλύτερη απόδοση. Ωστόσο, η χρήση τυχαίων παρτίδων μπορεί επίσης να εισάγει υψηλή διακύμανση στις εκτιμήσεις της κλίσης, η οποία μπορεί να επιβραδύνει τη σύγκλιση του αλγορίθμου. Για να μετριαστεί αυτό, μπορούν να χρησιμοποιηθούν παραλλαγές του SGD, όπως το Mini-Batch Gradient Descent, οι οποίες υπολογίζουν τη διαβάθμιση χρησιμοποιώντας μια μικρή, σταθερού μεγέθους παρτίδα παραδειγμάτων εκπαίδευσης και όχι μόνο ενός. Επιπλέον, ο ρυθμός εκμάθησης πρέπει να ρυθμιστεί προσεκτικά για να εξισορροπηθεί η αντιστάθμιση μεταξύ της ταχύτητας σύγκλισης και της ακρίβειας.

3.1.2 Οπισθοδιάδοση (Backpropagation)

Η οπισθοδιάδοση είναι ένας αλγόριθμος που χρησιμοποιείται για την εκπαίδευση τεχνητών νευρωνικών δικτύων, ειδικά των "βαθιών" νευρωνικών δικτύων. Είναι ένας αλγόριθμος εκμάθησης που χρησιμοποιείται για την ενημέρωση των βαρών του δικτύου προκειμένου να ελαχιστοποιηθεί η συνάρτηση απώλειας. Η συνάρτηση απώλειας υπολογίζει τη διαφορά μεταξύ των προβλέψεων του δικτύου και της πραγματικής εξόδου και ο στόχος της οπισθοδιάδοσης είναι η προσαρμογή των βαρών με τέτοιο τρόπο ώστε να ελαχιστοποιείται αυτή η διαφορά.

Η οπισθοδιάδοση είναι ένας αλγόριθμος βελτιστοποίησης που βασίζεται σε κλίση, που σημαίνει ότι χρησιμοποιεί τη διαβάθμιση της συνάρτησης απώλειας σε σχέση με τα βάρη του δικτύου για να καθορίσει την κατεύθυνση στην οποία πρέπει να ενημερωθούν τα βάρη. Ο αλγόριθμος λειτουργεί κάνοντας πρώτα ένα πέρασμα προς τα εμπρός μέσω του δικτύου, στο οποίο η είσοδος μετατρέπεται σε πρόβλεψη εξόδου. Στη συνέχεια, στο πίσω πέρασμα, υπολογίζεται η κλίση της συνάρτησης απώλειας σε σχέση με κάθε βάρος στο δίκτυο. Αυτή η κλίση χρησιμοποιείται στη συνέχεια για την ενημέρωση των βαρών προς την αντίθετη κατεύθυνση προς τη διαβάθμιση, έτσι ώστε να ελαχιστοποιηθεί η συνάρτηση απώλειας.

Η βασική ιδέα πίσω από την αντίστροφη διάδοση είναι η χρήση του κανόνα της αλυσίδας της διαφοροποίησης για τον υπολογισμό των κλίσεων της συνάρτησης απώλειας σε σχέση με κάθε βάρος στο δίκτυο. Ο κανόνας της αλυσίδας επιτρέπει τον υπολογισμό των διαβαθμίσεων



Εικόνα 3.5 Εξήγηση οπισθοδιάδοσης σε σύστημα τεσσάρων layers

εφαρμόζοντας αναδρομικά τις παραγώγους της συνάρτησης ενεργοποίησης κάθε επιπέδου στις διαβαθμίσεις της απώλειας σε σχέση με τις εξόδους αυτού του επιπέδου.

Η οπισθοδιάδοση είναι ένας ισχυρός και ευρέως χρησιμοποιούμενος αλγόριθμος στη βαθιά μάθηση και έχει χρησιμοποιηθεί για την επίτευξη αποτελεσμάτων αιχμής σε ένα ευρύ φάσμα εργασιών, όπως ταξινόμηση εικόνων, αναγνώριση ομιλίας και επεξεργασία φυσικής γλώσσας. Ωστόσο, όπως όλοι οι αλγόριθμοι βελτιστοποίησης, η οπισθοδιάδοση έχει ορισμένους περιορισμούς, όπως η δυσκολία αποφυγής κακών τοπικών ελάχιστων και η ευαισθησία του αλγορίθμου στην επιλογή του ρυθμού εκμάθησης και άλλων υπερπαραμέτρων.

3.2 Επαναλαμβανόμενα Νευρωνικά Δίκτυα (Recurrent Neural Networks)

Τα επαναλαμβανόμενα νευρωνικά δίκτυα (RNN) είναι ένας τύπος δικτύου βαθιάς εκμάθησης που έχει σχεδιαστεί για την επεξεργασία αλληλουχιών δεδομένων, όπως δεδομένα χρονοσειρών, σήματα ομιλίας ή εργασίες επεξεργασίας φυσικής γλώσσας. Τα RNN επεξεργάζονται δεδομένα σε ακολουθίες, όπου η έξοδος σε ένα δεδομένο χρονικό βήμα εξαρτάται από την είσοδο σε προηγούμενα χρονικά βήματα. Τα RNN αποτελούνται από επαναλαμβανόμενες μονάδες, οι οποίες είναι κόμβοι που λαμβάνουν ως είσοδο τόσο την τρέχουσα είσοδο όσο και την κρυφή κατάσταση από το προηγούμενο χρονικό βήμα.

Η κρυφή κατάσταση είναι ένα διάνυσμα που συνοψίζει τις πληροφορίες που επεξεργάζεται το δίκτυο μέχρι εκείνο το σημείο της ακολουθίας. Η κρυφή κατάσταση υπολογίζεται χρησιμοποιώντας μια συνάρτηση ενεργοποίησης, όπως μια σιγμοειδή συνάρτηση (sigmoid) ή την tanh συνάρτηση και στη συνέχεια περνά στο επόμενο χρονικό βήμα, επιτρέποντας στο δίκτυο να διατηρεί πληροφορίες από προηγούμενα χρονικά βήματα. Υπάρχουν διάφορες παραλλαγές των RNN, συμπεριλαμβανομένων των απλών RNN, των δικτύων μακράς βραχυπρόθεσμης μνήμης (LSTM) και των περιφραγμένων επαναλαμβανόμενων μονάδων (Gated Recurrent Units GRU). Τα απλά RNN έχουν απλή δομή και είναι καλά στην καταγραφή βασικών μοτίβων σε ακολουθίες, αλλά δεν είναι κατάλληλα για την επεξεργασία μεγάλων ακολουθιών λόγω του προβλήματος της κλίσης εξαφάνισης.

Τα LSTM και τα GRU έχουν πιο πολύπλοκες δομές και έχουν σχεδιαστεί για να ξεπερνούν αυτό το πρόβλημα, καθιστώντας τα καλύτερα κατάλληλα για την επεξεργασία μεγάλων ακολουθιών δεδομένων. Τα RNN εκπαιδεύονται χρησιμοποιώντας μια παραλλαγή του αλγορίθμου της οπισθοδιάδοσης που είναι γνωστή ως οπισθοδιάδοση μέσω του χρόνου (BPTT). Κατά τη διάρκεια της εκπαίδευσης, η ακολουθία εισόδου επεξεργάζεται ένα βήμα τη φορά και το σφάλμα μεταξύ της προβλεπόμενης εξόδου και της πραγματικής εξόδου χρησιμοποιείται για την προσαρμογή των βαρών των συνδέσεων μεταξύ των κόμβων.

3.2.1 Long Short Term Memory (LSTM)

Η Μακροχρόνια Βραχυπρόθεσμη Μνήμη (LSTM) είναι ένας τύπος επαναλαμβανόμενου νευρωνικού δικτύου (RNN) που έχει σχεδιαστεί για να χειρίζεται το πρόβλημα της εξαφάνισης των κλίσεων στα παραδοσιακά RNN. Τα LSTM χρησιμοποιούνται συνήθως για εργασίες όπως η επεξεργασία φυσικής γλώσσας, η αναγνώριση ομιλίας και η διαδοχική πρόβλεψη. Ένα δίκτυο LSTM αποτελείται από μια σειρά κυψελών μνήμης, πυλών και συναρτήσεων ενεργοποίησης. Τα κύτταρα μνήμης διατηρούν πληροφορίες για μεγαλύτερο χρονικό διάστημα και οι πύλες ελέγχουν τη ροή πληροφοριών προς και έξω από τα κύτταρα μνήμης. Οι λειτουργίες ενεργοποίησης καθορίζουν την έξοδο του δικτύου.

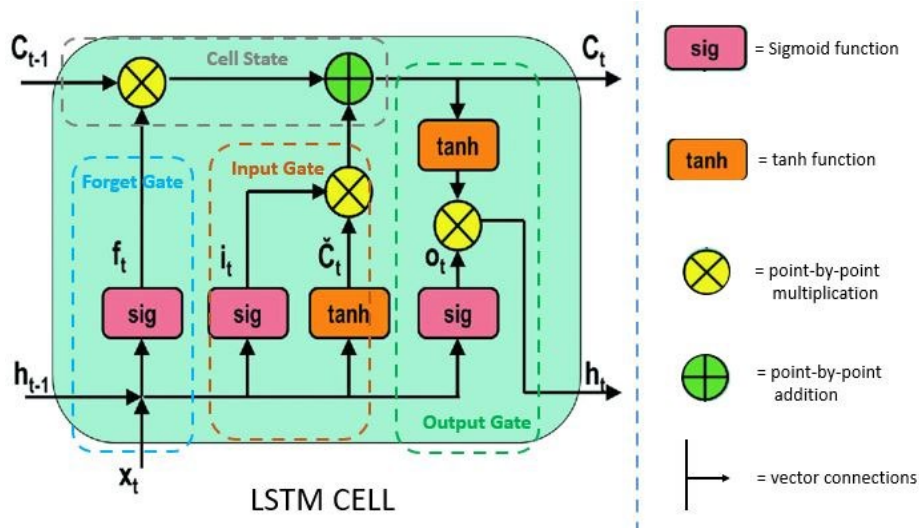
Πριν μπούμε στην επεξήγηση των πυλών θα σχολιάσουμε κάποιες ακόμα έννοιες. Τα LSTM επεξεργάζονται δεδομένα εισόδου ένα βήμα τη φορά. Τα δεδομένα εισόδου διαβιβάζονται μέσω του δικτύου, μαζί με την προηγούμενη κρυφή κατάσταση και κατάσταση κυψέλης. Οι πληροφορίες στα κελιά μνήμης ενημερώνονται πολλαπλασιάζοντας πρώτα την προηγούμενη κατάσταση κελιού με την πύλη forget και προσθέτοντας την τρέχουσα είσοδο πολλαπλασιασμένη με την πύλη εισόδου. Αυτές οι νέες πληροφορίες αποθηκεύονται στα κελιά μνήμης. Η κρυφή κατάσταση υπολογίζεται πολλαπλασιάζοντας την έξοδο από τα κελιά μνήμης με την πύλη εξόδου και περνώντας την από μια συνάρτηση ενεργοποίησης tanh. Η κρυφή κατάσταση χρησιμοποιείται ως είσοδος για το επόμενο χρονικό βήμα.

Υπάρχουν τρεις τύποι πυλών σε ένα δίκτυο LSTM:

- Πύλη εισόδου: Η πύλη εισόδου καθορίζει πόσο από την τρέχουσα είσοδο πρέπει να αποθηκευτεί στα κελιά μνήμης. Υπολογίζεται με βάση την τρέχουσα είσοδο και την προηγούμενη κρυφή κατάσταση. Η πύλη εισόδου χρησιμοποιεί μια συνάρτηση ενεργοποίησης σιγμοειδούς για να παράγει μια τιμή μεταξύ 0 και 1, η οποία αντιπροσωπεύει την ποσότητα πληροφοριών από την τρέχουσα είσοδο που πρέπει να αποθηκευτεί στα κελιά μνήμης.
- Forget gate: Η πύλη forget καθορίζει ποιες πληροφορίες από το προηγούμενο χρονικό βήμα πρέπει να ξεχαστούν. Υπολογίζεται επίσης με βάση την τρέχουσα είσοδο και την προηγούμενη κρυφή κατάσταση. Η πύλη forget χρησιμοποιεί μια συνάρτηση ενεργοποίησης σιγμοειδούς για να παράγει μια τιμή μεταξύ 0 και 1, η οποία αντιπροσωπεύει την ποσότητα πληροφοριών από το προηγούμενο χρονικό βήμα που πρέπει να ξεχαστεί.
- Πύλη εξόδου: Η πύλη εξόδου καθορίζει ποιες πληροφορίες από τα κελιά μνήμης πρέπει να χρησιμοποιηθούν για τον υπολογισμό της εξόδου του δικτύου.

Υπολογίζεται με βάση την τρέχουσα είσοδο και την προηγούμενη κρυφή κατάσταση. Η πύλη εξόδου χρησιμοποιεί μια συνάρτηση ενεργοποίησης σιγμοειδούς για να παράγει μια τιμή μεταξύ 0 και 1, η οποία αντιπροσωπεύει την ποσότητα πληροφοριών από τα κελιά μνήμης που πρέπει να χρησιμοποιηθούν για τον υπολογισμό της εξόδου.

Οι πληροφορίες που είναι αποθηκευμένες στα κελιά μνήμης ενημερώνονται σε κάθε χρονικό βήμα και οι πύλες ελέγχουν τη ροή των πληροφοριών μέσα και έξω από τα κελιά μνήμης. Αυτό επιτρέπει στο δίκτυο να διατηρεί πληροφορίες για μεγαλύτερο χρονικό διάστημα και να λαμβάνει αποφάσεις με βάση τόσο την τρέχουσα είσοδο όσο και τις πληροφορίες που είναι αποθηκευμένες στα κελιά μνήμης. Τα LSTM έχουν χρησιμοποιηθεί για την επίτευξη αποτελεσμάτων αιχμής σε μια ποικιλία εργασιών, συμπεριλαμβανομένης της αυτόματης μετάφρασης και της αναγνώρισης ομιλίας. Είναι ιδιαίτερα κατάλληλα για εργασίες όπου οι ακολουθίες εισόδου είναι μεγάλες και έχουν πολύπλοκες εξαρτήσεις μεταξύ των στοιχείων.



Εικόνα 3.6 : Λειτουργία κελιού LSTM με χρήση των πυλών

4. Βασικοί Χρηματοοικονομικοί Όροι

4.1 Χρηματιστήριο

Το χρηματοδοτικό σύστημα και οι τράπεζες ως σύνολο αποτελούνται από πολλά μικρότερα παραρτήματα. Ένα από τα σημαντικότερα αυτά παραρτήματα είναι το χρηματιστήριο. Το χρηματιστήριο είναι στην πραγματικότητα ο χώρος στον οποίον οι επενδυτές δίνουν τα χρήματά τους στις επιχειρήσεις, και εκείνες με την σειρά τους εκμεταλλεύονται αυτά τα χρήματα για τυχόν εργασίες τους και επενδυτικές δυνατότητες. Προφανώς ο λόγος για τον οποίο ένας επενδυτής διαλέγει μεταφέρει και να εμπιστευτεί το κεφάλαιό του σε μια επιχείρηση, είναι διότι πιστεύει μέσω της δικιάς του έρευνας ως προς την επιχείρηση, ότι πρόκειται να αναπτυχθεί έτσι ώστε να του επιφέρει κέρδος στην επένδυση του. Έτσι καταφέρνει η κάθε εισηγμένη στο χρηματιστήριο εταιρεία να πάρει χρήματα που μπορεί να χρησιμοποιήσει για να εξελιχθεί, και ο επενδυτής να βγάλει προφανώς κέρδος από την επένδυσή του στην εταιρεία. Οι συναλλαγές που γίνονται στον χώρο του χρηματιστηρίου είναι μετοχές, εμπορικές, ομόλογα, ομολογίες (κοινές και μετατρέψιμες) όπως και δικαιώματα προτίμησης.

4.1.1 Δημιουργία και Σκοπός του Χρηματιστηρίου

Αρχικά, το χρηματιστήριο άτυπα δημιουργήθηκε από την ανάγκη ανταλλαγής αγαθών και εμπορευμάτων κυρίως για αγροτικούς σκοπούς. Οι πρώτες κοινωνίες που το χρησιμοποίησαν για τέτοιους λόγους ήταν στον χώρο της Ευρώπης, ενώ στην ήπειρο της Αμερικής δημιουργήθηκε για την εμπορία των μετάλλων. Υπάρχουν και δεδομένα που τείνουν πως η πρώτη που εφευρέθηκε ποτέ χρηματιστηριακή αγορά στην Γαλλία υπήρχε και λειτουργούσε από τα μέσα του 12ου αιώνα, προφανώς άτυπα και χωρίς τον όρο : "Χρηματιστήριο". Τυπικά όμως το χρηματιστήριο άρχισε να λειτουργεί οργανωμένα τις αρχές του 17ου αιώνα με βάση το Άμστερνταμ. Ο όρος που δόθηκε τότε ήταν ο χώρος / αγορά που γίνονται συναλλαγές χρηματοοικονομικών προϊόντων.

Οι πρώτες μετοχές που διαπραγματεύτηκαν ήταν της εταιρείας Dutch East India Company. Το αμέσως επόμενο χρηματιστήριο που δημιουργήθηκε ήταν στην Βιέννη το 1711 και στην Αγγλία το 1773 , όπου και εκεί είχαν συναλλαγές με τις μετοχές της Dutch East India Company. Αυτές οι κινήσεις που "χρηματοδότησαν" ξένες επιχειρήσεις βοήθησαν επίσης και τις διπλωματικές ενέργειες μεταξύ των χωρών . Από τότε ήταν ευρέως γνωστό πως για την σωστή διαπραγμάτευση και για να εκπληρωθούν έτσι όπως πρέπει οι αγορές και οι πωλήσεις των μετοχών, χρειάζεται να έχει καλή πληροφόρηση ο διαχειριστής των κάθε μετοχών. Το χρηματιστήριο που πλέον χρησιμοποιείται από τον περισσότερο κόσμο είναι το χρηματιστήριο αξιών. Η συγκεκριμένη αγορά αφορά αντικείμενα και μετοχές πρωτογενής και δευτερογενής αγοράς, όπου θα εξηγηθούν στην συνέχεια της εργασίας για το τι πρόκειται και τι τομέας συμπεριλαμβάνεται στην κάθε αγορά\.

4.1.2 Η έννοια της μετοχής

Ένα σημαντικό κομμάτι που χρειάζεται για την κατανόηση της λειτουργίας του χρηματιστηρίου, είναι να καταλάβουμε πως λειτουργούν οι μετοχές. Ο κάτοχος μιας μετοχής, στην πραγματικότητα είναι και συνιδιοκτήτης αυτής της εταιρείας. Οι επιχειρήσεις συνήθως

εκδίδουν μετοχές, με σκοπό να αυξήσουν την ρευστότητα τους, διότι τα χρήματα που πληρώνονται για την αγορά μετοχών εισέρχονται στην εταιρεία. Η τιμή και η αξία της μετοχής, βασίζεται μεταξύ άλλων, στις βασικές οικονομικές αρχές της προσφοράς και της ζήτησης. Με λίγα λόγια, αν υπάρχει μεγάλη ζήτηση στην αγορά των μετοχών μιας συγκεκριμένης επιχείρησης, θα αυξάνεται και η τιμή της. Η τιμή και η αξία όμως, πολλές φορές μπορεί να μεταβληθεί απρόοπτα και πολλές φορές ραγδαία μέσα σε λίγο χρόνο με την εμφάνιση ενός τυχαίου συμβάντος.

Πρόσφατο παράδειγμα είναι η Corona Extra. Με την έκρηξη του κορονοϊού (Corona Virus) ο κόσμος συσχέτισε τα προϊόντα της Corona με το ιό, με αποτέλεσμα να μην αγοράζονται τα αγαθά της. Αυτό έφερε την εταιρεία σε δύσκολη θέση και στην συνέχεια έπεσε και η τιμή της μετοχής της. Από αυτό το γεγονός βλέπουμε ότι όταν ζημιώνεται η επιχείρηση, πέφτει η αξία της μετοχής. Αν η τιμή έχει πέσει χαμηλότερα από όταν αγοράστηκε η μετοχή, χάνει λεφτά ο αγοραστής. Σε περιπτώσεις που υπάρχει ξαφνικά μεγάλη μεταβολή στην τιμή, συγκεκριμένα αρνητική, δημιουργείται ένα μεγάλο πρόβλημα. Δεν υπάρχει ζήτηση. Αυτό σημαίνει ότι όσο πέφτει η αξία της μετοχής δεν υπάρχει λόγος να την αγοράσει κάποιος, άρα αναγκαστικά ένας μέτοχος δεν μπορεί να πράξει ώστε να σώσει έστω ένα μερίδιο των χρημάτων του.

4.1.3 Είδη Μετοχών

Οι μετοχές διαχωρίζονται σε πολλά διαφορετικά είδη. Αρχικά, τις κοινές και τις προνομιούχες. Μία από τις διαφορές μεταξύ τους είναι στο δικαίωμα ψήφου που δίνεται στον κάτοχο. Το κυρίως όφελος που δίνεται στις κοινές, είναι το δικαίωμα ψήφου, δηλαδή οι κάτοχοι προνομιούχων μετοχών δεν έχουν το δικαίωμα να αποφασίσουν για την κατεύθυνση της εταιρείας. Η διαδικασία της εκλογής γίνεται κάθε χρόνο στην ετήσια γενική συνέλευση της επιχείρησης. Άλλη διάκριση είναι μεταξύ σε ονομαστικές και ανώνυμες. Ο λόγος που υπάρχει αυτή η διαφορά είναι για νομικούς λόγους. Οι ονομαστικές μετοχές, όπως και χαρακτηρίζει το όνομά τους, έχουν πάνω τους το όνομα της εταιρείας όπως και του μετόχου. Οι ανώνυμες αντίθετα έχουν μόνο το όνομα της εταιρείας. Οι επιχειρήσεις που εκδίδουν ονομαστικές μετοχές αποκλειστικά, είναι οι τράπεζες, οι κοινής ωφέλειας κλπ. Τέτοιου είδους επιχειρήσεις είναι υποχρεωμένες από τον νόμο να έχουν ονομαστικές μετοχές ενώ στα υπόλοιπα είδη εταιρειών συνηθίζεται να υπάρχουν ανώνυμες μετοχές.

Η τελευταία διαφοροποίηση που υπάρχει είναι μεταξύ δημόσιων και ιδιωτικών μετοχών. Αρχικά να ξεκαθαρίσουμε ότι οι δημόσιες μετοχές έχουν να κάνουν με δημόσιες εταιρείες και αντίστοιχα οι ιδιωτικές με ιδιωτικές εταιρείας. Οι περισσότερες εταιρείες που ξεκινάνε ως μικρές, είναι και ιδιωτικές. Το είδος των μετοχών που εκδίδουν είναι ιδιωτικές. Τώρα σε τέτοιου μήκους επιχειρήσεις συνήθως οι ανταλλαγές των μετοχών είναι συχνές και το σκέλος των μετοχών είναι δικαιολογημένα μικρό. Όσο όμως μεγαλώνει μια επιχείρηση έχει περισσότερο κέρδος, όπως και μεγαλύτερη ανάγκη για ρευστοποίηση. Αυτό σημαίνει ότι έχει ανάγκη για περισσότερους μετόχους, ώστε να μπορέσει να αναπτυχθεί. Εδώ έρχεται η ιδέα να εισαχθεί η εταιρεία στο χρηματιστήριο. Το πρώτο πράγμα που γίνεται είναι ότι η εταιρεία γίνεται δημόσια, όπως και οι μετοχές. Αυτή η διαδικασία ονομάζεται ως αρχική δημόσια προσφορά. Με λίγα λόγια εκφέρει το γεγονός ότι για πρώτη φορά μια επιχείρηση εκδίδει δημόσιες μετοχές αντί για ιδιωτικές. Αυτή τη διαδικασία, την χρησιμοποιούν και μικρότερες εταιρείες πολλές φορές, για τον μοναδικό λόγο ότι τους λείπουν χρήματα, άρα

ένας τρόπος να μαθευτεί από τον κόσμο και να της "δώσουν" λεφτά είναι να δημοσιοποιηθεί με την χρήση του χρηματιστηρίου.

4.1.4 Σημαντικότερα Χρηματιστήρια

Μία από τις σημαντικότερες και πιο γνωστές χρηματιστηριακές αγορές στον κόσμο είναι το New York Stock Exchange (NYSE). Το NYSE βρίσκεται στο Manhattan στην Νέα Υόρκη. Είναι το μεγαλύτερο χρηματιστήριο στον κόσμο όσο αναφορά την κεφαλοποίηση εισηγμένων επιχειρήσεων. Το 2022 το NYSE έχει καταγράψει market capitalisation (κεφαλοποίηση αγοράς) 25.8 Τρισεκατομμύρια Αμερικάνικα δολάρια. Το NYSE λειτουργεί και ως ένα πλήρως ηλεκτρονικό χρηματιστήριο, ονόματι ως "Euronext". Η πρώτη μορφή που είχε το NYSE ήταν ένα δωμάτιο που νοικιαζόταν για 200 δολάρια τον μήνα από τις 17 Μαΐου το 1792. Επίσης το συγκεκριμένο χρηματιστήριο περιλαμβάνει συνολικά περισσότερες από τρεις χιλιάδες επιχειρήσεις, μικρές, μεσαίες και μεγάλες. Να διευκρινιστεί ότι οι εισηγμένες αυτές επιχειρήσεις δεν βρίσκονται απαραίτητα στις Ηνωμένες Πολιτείες της Αμερικής, αλλά και σε άλλες χώρες. Σίγουρα όμως οι πληθώρα των εταιρειών έχει ως βάση τις Η.Π.Α. Η εταιρεία στην οποία ανήκει το NYSE, ονομάζεται Intercontinental Exchange με βάση τα γραφεία στην Wall Street όπως αναφέρθηκε και προηγουμένως.

Για να εισαχθεί μία επιχείρηση στην αγορά του χρηματιστηρίου της Νέας Υόρκης, πρέπει να πληρεί και τα απαραίτητα κριτήρια και να έχει μία σημαντική θέση στον κλάδο τον οποίο δραστηριοποιείται. Άλλο σημαντικό χρηματιστήριο είναι το NASDAQ (National Association of Securities Dealers Automated Quotations Stock Market) με κεφαλαιοποίηση αγοράς 17.36 τρισεκατομμύρια δολάρια το 2022, την δεύτερη μεγαλύτερη στον κόσμο, ακριβώς μετά του NYSE. Το NASDAQ βρίσκεται επίσης στις Η.Π.Α και συγκεκριμένα έχει τα κεντρικά του γραφεία στην Νέα Υόρκη. Η επιχείρηση στην οποία ανήκει το NASDAQ είναι η Nasdaq Inc όπου έχει υπό την κατοχή της και το Nasdaq Nordic, που είναι ένα σκανδιναβικό χρηματιστήριο με βάση την Στοκχόλμη της Σουηδίας και ιδρύθηκε το 2003. Αντίστοιχα το 1971 χρηματοδοτήθηκε το NASDAQ από την πλέον Financial Industry Regulatory Authority (FINRA). Το συγκεκριμένο χρηματιστήριο χωρίζεται σε δύο διαφορετικές αγορές αναλόγως το μέγεθος και την "ισχύ" των επιχειρήσεων. Η πρώτη αγορά, είναι η λεγόμενη εθνική αγορά που περιλαμβάνει μεγάλες και μεσαίες μετοχές που δραστηριοποιούνται καθημερινά.

Ο δεύτερος τομέας είναι η αγορά χαμηλής κεφαλαιοποίησης. Σε αυτήν την αγορά υπάρχουν μικρότερες επιχειρήσεις που βρίσκονται στα πρόθυρα της ανάπτυξης, για αυτό και είναι πιο "χαλαρά" τα κριτήρια για να εισαχθεί ένας μετοχικός τίτλος. Μία σημαντική διευκρίνηση όσο αναφορά το NASDAQ, είναι ότι ως χρηματιστήριο λειτουργεί απολύτως ηλεκτρονικά και δεν έχει πουθενά στον κόσμο μία φυσική υπόσταση. Αυτό σημαίνει ότι όλες οι συναλλαγές που κάνουν οι μέτοχοι, γίνονται αυστηρά και μόνο μέσω του δικτύου ηλεκτρονικών υπολογιστών, δίνοντας σε αυτό το χρηματιστήριο ένα προτέρημα έναντι άλλων χρηματιστηρίων που λειτουργούν με τον αναλογικό τρόπο συναλλαγών.

5. Υλοποίηση

Σε αυτό το κεφάλαιο θα εφαρμοστούν τα τρία μοντέλα Linear Regression, SVM και LSTM σε πραγματικά δεδομένα χρηματιστηρίου 8 εταιρειών. Θα ξεκινήσουμε με την παρουσίαση των αρχείων των δεδομένων και των σχετικών features τους, στην συνέχεια θα αναλύσουμε κάθε τμήμα του κώδικα των μοντέλων και την απόδοσή τους. Τα αποτελέσματα απαρτίζονται από συγκρίσεις των μοντέλων χρησιμοποιώντας την συνάρτηση απώλειας MSE (Mean Squared Error) και γραφικές παραστάσεις που απεικονίζουν τις τιμές πραγματικές τιμές των μετοχών σε σχέση με τις προβλεπόμενες τιμές. Θα εφαρμοστούν δοκιμές διαφορετικών δεδομένων εκπαίδευσης, βελτιστοποιητών και συναρτήσεων ενεργοποίησης με σκοπό να λειτουργήσουμε τα μοντέλα με όσο το δυνατόν καλύτερη απόδοση. Τέλος, θα παρουσιαστούν τα συμπεράσματα και θα σχολιαστούν οι προβληματισμοί και τρόποι βελτίωσης.

5.1 Ανάλυση των δεδομένων

Τα δεδομένα που χρησιμοποιήθηκαν για την ανάπτυξη των αλγορίθμων είναι από 8 γνωστές εταιρείες. Οι επιχειρήσεις είναι οι εξής : Coca Cola, AMD, Microsoft, Toyota Motors, BMW, Amazon, Apple, Nvidia. Ο λόγος που επιλέχτηκαν αυτές οι εταιρείες συγκεκριμένα, είναι διότι βρίσκονται πολλά χρόνια στο χρηματιστήριο και έχουν διαθέσιμα δεδομένα πολλών χρόνων. Όπως αναφέρθηκε και σε προηγούμενες ενότητες, το Deep Learning γενικότερα απαιτεί έναν μεγάλο αριθμό δεδομένων, άρα οι εταιρείες που είναι εισηγμένες περισσότερα χρόνια βοηθούν περισσότερο στο να φανεί η αξία του. Τα αρχεία της κάθε επιχείρησης είναι σε μορφή CSV και είναι διαθέσιμα μέσω του yahoo finance. Παρακάτω θα γίνει η διευκρίνηση ενός αρχείου για την κατανόηση των διαφορετικών features που θα χρησιμοποιηθούν κατά την υλοποίηση.

Date	Open	High	Low	Close	Volume	Adj Close	
6/29/2010		19	25	17.540001	23.889999	18766300	23.889999
6/30/2010	25.790001		30.42	23.299999	23.83	17187100	23.83
7/1/2010	25	25.92	20.27	21.959999	8218800	21.959999	21.959999
7/2/2010	23	23.1	18.709999	19.200001	5139800	19.200001	19.200001
7/6/2010	20	20	15.83	16.110001	6866900	16.110001	16.110001
7/7/2010	16.4	16.629999	14.98	15.8	6921700	15.8	15.8
7/8/2010	16.139999	17.52	15.57	17.459999	7711400	17.459999	17.459999
7/9/2010	17.58	17.9	16.549999	17.4	4050600	17.4	17.4
7/12/2010	17.950001	18.07	17	17.049999	2202500	17.049999	17.049999
7/13/2010	17.389999	18.639999	16.9	18.139999	2680100	18.139999	18.139999
7/14/2010	17.940001	20.15	17.76	19.84	4195200	19.84	19.84
7/15/2010	19.940001	21.5	19	19.889999	3739800	19.889999	19.889999
7/16/2010	20.700001	21.299999	20.049999	20.639999	2621300	20.639999	20.639999
7/19/2010	21.370001	22.25	20.92	21.91	2486500	21.91	21.91
7/20/2010	21.85	21.85	20.049999	20.299999	1825300	20.299999	20.299999
7/21/2010	20.66	20.9	19.5	20.219999	1252500	20.219999	20.219999
7/22/2010	20.5	21.25	20.370001	21	957800	21	21
7/23/2010	21.190001	21.559999	21.059999	21.290001	653600	21.290001	21.290001
7/26/2010	21.5	21.5	20.299999	20.950001	922200	20.950001	20.950001
7/27/2010	20.91	21.18	20.26	20.549999	619700	20.549999	20.549999
7/28/2010	20.549999	20.9	20.51	20.719999	467200	20.719999	20.719999
7/29/2010	20.77	20.879999	20	20.35	616000	20.35	20.35

Πίνακας 5.1: Παράδειγμα αρχείου CSV

Εδώ βλέπουμε ένα παράδειγμα ενός μέρος του αρχείου των μετοχών της Tesla.

- I) Αρχικά στην πρώτη στήλη έχουμε την ημερομηνία (Date) για κάθε δεδομένο που παρουσιάζεται σε κάθε γραμμή.
- II) Στην στήλη Open βρίσκεται η αξία της μετοχής όταν θα ανοίξει το χρηματιστήριο εκείνη την ημέρα.
- III) High είναι η μέγιστη τιμή της μετοχής στην διάρκεια της ημέρας.
- IV) Low είναι η ελάχιστη τιμή της μετοχής στην διάρκεια της ημέρας.
- V) Στην στήλη Close βρίσκεται η αξία που θα έχει η μετοχή την στιγμή που θα κλείσει το χρηματιστήριο εκείνη την ημέρα.
- VI) Volume : Ο συνολικός αριθμός μετοχών που διαπραγματεύτηκαν κατά τη διάρκεια της ημέρας
- VII) Adj Close : Η τιμή κλεισίματος προσαρμοσμένη για παράγοντες όπως μερίσματα και διασπάσεις μετοχών

5.2 Ανάλυση του Linear Regression

```
import numpy as np
import pandas as pd
df = pd.read_csv("stock_data.csv")
X = df.drop(["Close", "Date"], axis=1)
y = df["Close"]

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y_test, y_pred)
print(f'{mse:.4f}')
```

Σε αυτήν την ενότητα πραγματοποιείται η δημιουργία του μοντέλου Γραμμικής Παλινδρόμησης χρησιμοποιώντας την βιβλιοθήκη Sklearn. Η Γραμμική Παλινδρόμηση είναι μια μέθοδος κλασικής Μηχανικής Μάθησης και όπως φαίνεται παρακάτω, ο κώδικας είναι σχετικά απλός. Φορτώνουμε τα δεδομένα του αρχείου CSV σε ένα dataframe ονόματι df μαζί με όλα τα features του. Στην συνέχεια φτιάχνουμε 2 μεταβλητές X και y, όπου το X είναι τα features μας και το y ο στόχος μας (Close). Μετά χωρίζουμε τα δεδομένα μας, 80% για την εκπαίδευση και 20% για το test, δημιουργούμε το μοντέλο και ξεκινάμε τη διαδικασία της εκπαίδευσης. Τέλος χρησιμοποιούμε το εκπαιδευμένο πλέον μοντέλο για να προβλέψουμε τις τιμές και υπολογίζουμε το μέσο τετραγωνικό σφάλμα μεταξύ των τιμών των προβλέψεων και των πραγματικών τιμών.

5.3 Ανάλυση του SVM

```
import numpy as np
import pandas as pd
df = pd.read_csv('stock_data.csv')
X = df.drop(['Date', 'Close'], axis=1)
y = df['Close']

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

from sklearn.svm import SVR
model = SVR(kernel='linear', C=100, gamma='auto', epsilon=.1)
model.fit(X_train, y_train)
y_train_pred = model.predict(X_train)
y_test_pred = model.predict(X_test)

from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y_test, y_test_pred)
print(f" {mse:.4f} ")
```

Σε αυτό το κεφάλαιο πραγματοποιείται η δημιουργία του μοντέλου SVM χρησιμοποιώντας την βιβλιοθήκη Sklearn. Το SVM είναι μια μέθοδος Μηχανικής Μάθησης. Φορτώνουμε τα δεδομένα του αρχείου CSV σε ένα dataframe ονόματι df μαζί με όλα τα features του. Στην συνέχεια φτιάχνουμε 2 μεταβλητές X και y, όπου το X είναι τα features μας και το y ο στόχος μας (Close). Χρησιμοποιούμε το Scale για να μετατρέψουμε τα δεδομένα μας πριν την εκπαίδευση. Μετά χωρίζουμε τα δεδομένα μας, 80% για την εκπαίδευση και 20% για το test, δημιουργούμε το μοντέλο και ξεκινάμε τη διαδικασία της εκπαίδευσης. Τέλος χρησιμοποιούμε το εκπαιδευμένο πλέον μοντέλο για να προβλέψουμε τις τιμές και υπολογίζουμε το μέσο τετραγωνικό σφάλμα μεταξύ των τιμών των προβλέψεων και των πραγματικών τιμών.

5.4 Ανάλυση του LSTM

```
import pandas as pd
import numpy as np
df = pd.read_csv('nvda.csv', parse_dates=['Date'])

from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
scaled_data = scaler.fit_transform(df.drop('Date', axis=1))
```

```

train_size = int(len(scaled_data) * 0.8)
train_data = scaled_data[:train_size, :]
test_data = scaled_data[train_size:, :]

n_features = df.shape[1] - 1
n_steps = 60
n_samples_train = train_data.shape[0] - n_steps
n_samples_test = test_data.shape[0] - n_steps
X_train = np.zeros((n_samples_train, n_steps, n_features))
y_train = np.zeros((n_samples_train, 1))

for i in range(n_samples_train):
    X_train[i] = train_data[i:i+n_steps, :]
    y_train[i] = train_data[i+n_steps, 4]

X_test = np.zeros((n_samples_test, n_steps, n_features))
y_test = np.zeros((n_samples_test, 1))

for i in range(n_samples_test):
    X_test[i] = test_data[i:i+n_steps, :]
    y_test[i] = test_data[i+n_steps, 4]

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout

model = Sequential([
    LSTM(128, input_shape=(n_steps, n_features)),
    Dropout(0.2),
    Dense(1)
])
model.compile(optimizer='adam', loss='mse')
history = model.fit(X_train, y_train, epochs=10, batch_size=2, validation_split=0.1)

from sklearn.metrics import mean_squared_error
y_pred = model.predict(X_test)
me = mean_squared_error(y_test, y_pred)
print(f'{me:.4f}')

```

Αυτός ο κώδικας εφαρμόζει ένα μοντέλο LSTM χρησιμοποιώντας την βιβλιοθήκη του TensorFlow Keras για την πρόβλεψη των τιμών των μετοχών. Ο κώδικας ξεκινάει εισάγοντας τις απαραίτητες βιβλιοθήκες, συμπεριλαμβανομένων των Pandas, NumPy, Matplotlib και TensorFlow Keras. Μετά, φορτώνει δεδομένα τιμών μετοχών από ένα αρχείο CSV χρησιμοποιώντας Pandas. Στη συνέχεια, τα δεδομένα κανονικοποιούνται χρησιμοποιώντας το MinMaxScaler από το Scikit-Learn, εξαιρουμένης της στήλης ημερομηνίας. Τα κανονικοποιημένα δεδομένα χωρίζονται σε σύνολα εκπαίδευσης και test,

όπου το 80% των δεδομένων χρησιμοποιούνται για εκπαίδευση και το 20% για test. Στη συνέχεια, ορίζεται το σχήμα εισόδου του μοντέλου LSTM, το οποίο περιλαμβάνει τον αριθμό των χαρακτηριστικών (εξαιρουμένης της στήλης ημερομηνίας) και τον αριθμό των χρονικών βημάτων που πρέπει να λαμβάνονται υπόψη κατά την πραγματοποίηση προβλέψεων.

Τα δεδομένα εκπαίδευσης και δοκιμών αναδιαμορφώνονται ώστε να ταιριάζουν στο σχήμα εισόδου του μοντέλου LSTM. Μετά από αυτό, το μοντέλο LSTM ορίζεται χρησιμοποιώντας TensorFlow Keras. Αποτελείται από ένα στρώμα LSTM με 128 μονάδες, ένα στρώμα Dropout 0,2 και ένα Dense στρώμα με μία μονάδα εξόδου. Στη συνέχεια, το μοντέλο χρησιμοποιεί τον βελτιστοποιητή Adam. Το μοντέλο εκπαιδεύεται στα δεδομένα εκπαίδευσης για δέκα εποχές με Batch Size δύο. Τέλος χρησιμοποιούμε το εκπαιδευμένο πλέον μοντέλο για να προβλέψουμε τις τιμές και υπολογίζουμε το μέσο τετραγωνικό σφάλμα μεταξύ των τιμών των προβλέψεων και των πραγματικών τιμών.

5.5 Ανάλυση Αποτελεσμάτων

Σε αυτό το κεφάλαιο θα παρουσιαστούν τα αποτελέσματα των μοντέλων μας. Στην συνέχεια θα αναλύσουμε αυτά τα αποτελέσματα με σκοπό να εξάγουμε συμπεράσματα. Οι αλγόριθμοι που σχολιάστηκαν παραπάνω (LR, SVM, LSTM), θα χρησιμοποιηθούν αντίστοιχα για δεδομένα 8 διαφορετικών εταιρειών. Για όλες τις περιπτώσεις θα υπάρχει γραφική παράσταση των πραγματικών τιμών των μετοχών και των προβλέψεων που έγιναν από τον εκάστοτε αλγόριθμο. Επίσης για όλες τις περιπτώσεις θα υπάρχει πίνακας που θα δείχνει το Mean Squared Error ως τρόπο αξιολόγησης των αλγορίθμων. Με το μοντέλο LSTM θα γίνουν και διαφορετικές δοκιμές με άλλους optimizers και activation functions με σκοπό να βρούμε ποιος είναι ο πιο κατάλληλος για την πρόβλεψη χρηματοοικονομικών δεδομένων.

Για έναν αλγόριθμο Deep Learning συνήθως περισσότερα δεδομένα υπόσχονται και καλύτερα αποτελέσματα, αλλά από την στιγμή που τα αρχεία δεν είναι απεριόριστα και δεν είναι ο ίδιος αριθμός δεδομένων για κάθε εταιρεία δεν μπορεί να χρησιμοποιηθεί και ο ίδιος αριθμός δεδομένων για την εκπαίδευση. Με τον LR αλγόριθμο οι αλλαγές που μπορούμε να κάνουμε είναι λιγότερες εφόσον και το μοντέλο είναι πιο απλό και πιο περιορισμένο. Εδώ όπως έχει προαναφερθεί στην ανάλυση του κώδικα, τα δεδομένα εκπαίδευσης τα δεχόμαστε ποσοστιαία.

Optimizers	Activation Function	MSE
Adam	tanh	0.0286
	relu	0.0325
SGD	tanh	0.0196
	relu	0.0234

Πίνακας 5.2 : Μέσο τετραγωνικό σφάλμα χρησιμοποιώντας διαφορετικούς αλγορίθμους βελτιστοποίησης και συναρτήσεων ενεργοποίησης στο μοντέλο LSTM.

Στον πίνακα αυτόν θα γίνει η σύγκριση πάνω στον κώδικα του LSTM ως προς το MSE χρησιμοποιώντας 2 διαφορετικά functions και 2 optimizers, όλα αυτά θα γίνουν στην πρόβλεψη της Coca Cola.. Από τα νούμερα είναι αισθητά καλύτερο το μοντέλο με την χρήση του Adam. Αυτό γίνεται αρχικά, γιατί ως βελτιστοποιητής ο Adam είναι φτιαγμένος για να λειτουργεί και να προβλέπει μη γραμμικά δεδομένα, άρα στην περίπτωση των χρηματοοικονομικών ταιριάζει. Ο SGD αντίθετα, ενώ μπορεί να δουλέψει πάνω σε μη γραμμικά δεδομένα, δυσκολεύει τον χρήστη ως προς την εκπαίδευση του και βρίσκεται σε υποδεέστερη θέση απέναντι στον Adam. Επίσης ο Adam μπορεί και προσαρμόζεται πάνω στις πληροφορίες που του δίνουμε. Αυτός είναι ο κύριος λόγος που ο Adam χρησιμοποιείται σε αυτά τα δεδομένα. Τα νούμερα των χρηματοοικονομικών είναι συνεχώς μεταβαλλόμενα και συνήθως δεν προβλέπονται από τα περισσότερα μοντέλα.

Όσον αφορά την χρήση των activation functions, τα αποτελέσματα είναι πιο κοντά. Η συνάρτηση tanh προσφέρει λίγο καλύτερα αποτελέσματα, αλλά και η relu κάνει εξίσου καλή δουλειά. Η συνάρτηση tanh, ονομάζεται υπερβολική εφαπτομένη (hyperbolic tangent), βασίζεται στην σιγμοειδή συνάρτηση και δέχεται εισόδους από -1 έως και 1. Αυτό μπορεί να μην ακούγεται πολύ, αλλά δίνει μεγάλο πλεονέκτημα στην relu, όπου χρησιμοποιεί δυαδική ενεργοποίηση. Αυτή η λειτουργία της δίνει την δυνατότητα για μία πιο λεπτομερή αναπαράσταση των εισόδων.

Για το μοντέλο του SVM θα χρησιμοποιήσουμε διαφορετικούς kernels για να συγκρίνουμε το μέσο τετραγωνικό σφάλμα και να αποφασίσουμε ποιο μας δίνει το καλύτερο δυνατό αποτέλεσμα. Να σημειωθεί ότι με όλους τους kernels ο χρόνος που τρέχει το πρόγραμμα είναι πρακτικά ίδιος. Ο πίνακας που ακολουθεί κατέχει τα νούμερα από την υλοποίηση του SVM πάνω στα δεδομένα της Coca Cola. Οι kernels που θα δοκιμαστούν είναι οι : Linear, Sigmoid, Poly, και RBF.

Kernels	MSE
Linear	0.041
Sigmoid	0.037
Polynomial	0.032
RBF	0.029

Πίνακας 5.3: Δοκιμή διαφορετικών Kernels πάνω στο μοντέλο του SVM.

Το χαμηλότερο MSE αποδίδεται από τον RBF. Ο γραμμικός kernel είναι αναμενόμενο να μην είναι τόσο αποδοτικός, επειδή τα χρηματοοικονομικά δεδομένα δεν απαρτίζονται μόνο από γραμμικούς παράγοντες. Στην συνέχεια, ο Sigmoid και ο Polynomial κάνουν φέρνουν πολύ κοντινά αποτελέσματα μεταξύ τους και με τον RBF. Γενικά αυτοί οι kernels είναι δυνατοί σε μη γραμμικά δεδομένα και μπορούν να λειτουργήσουν σε περισσότερες διαστάσεις.

Ο Polynomial kernel μπορεί να “πιάσει” πολυωνυμικές σχέσεις αλλά και πάλι έως ένα βαθμό. Ο Σιγμοειδής είναι ευρέως χρησιμοποιημένος kernel, αλλά λειτουργεί αποδοτικότερα σε προβλήματα δυαδικής ταξινόμησης. Στο πρόβλημα μας ο linear και ο sigmoid βρίσκονται

μειονεκτική θέση λόγω την δυνατοτήτων τους. Ο Polynomial από την άλλη είναι φτιαγμένος για να βρίσκει πυλωονυμικές σχέσεις που το χρειαζόμαστε για την επίλυση του προβλήματός μας, αλλά δυστυχώς δεν κατέχει την “ευλιγισία” του RBF. Αυτό συμβαίνει επειδή ο RBF αντιστοιχίζει τα δεδομένα εισόδου σε έναν απεριόριστο χώρο χαρακτηριστικών, αντίθετα με τον πολυωνυμικό.

Δεδομένα	Epoch 1		Epoch 2		Epoch 3	
	Χρόνος (s)	Loss (MSE)	Χρόνος (s)	Loss (MSE)	Χρόνος (s)	Loss (MSE)
36000	70	0.011	70	0.0012	72	0.0010
25000	58	0.0013	60	0.0013	60	0.0012

Πίνακας 5.4: Δοκιμή του LSTM με διαφορετικά ποσά δεδομένων και αποτελέσματα των εποχών.

Σε όλες τις περιπτώσεις λειτουργούμε με batch size : 2. Το πρόγραμμα μας χρησιμοποιεί περισσότερες εποχές αλλά για το παράδειγμα μας θα δείξουμε το αποτέλεσμα από 3 εποχές. Είναι εμφανές ότι όσο περισσότερα δεδομένα τροφοδοτούντε στο πρόγραμμα, τόσο περισσότερο αργεί η διαδικασία της εκπαίδευσης. Εφόσον υλοποιούντε 10 εποχές, αν τροφοδοτηθούν 36000 δεδομένα, η εκπαίδευση θα χρειαστεί 25 λεπτά. Φαίνεται ότι όσο περισσότερα είναι τα δεδομένα εκπαίδευσης, τόσο χαμηλότερη είναι και η απώλεια μας, στην συγκεκριμένη περίπτωση έχουμε θέσει ως απώλεια το σφάλμα μας (MSE).

Η διαφορά στον χρόνο είναι σημαντική, από την στιγμή που τα μισά δεδομένα σημαίνουν και μισό χρόνο εκτέλεσης, αλλά το MSE αυξάνεται εξίσου. Φαίνεται ότι από μία εποχή στην επόμενη, η απώλεια μας μειώνεται, αυτό σημαίνει ότι η εκπαίδευση είναι αποτελεσματική. Το σφάλμα δεν δείχνει να μειώνεται δραστικά, αλλά πρέπει να έχουμε υπό όψη μας ότι η διαδικασία αυτή υπάρχει για 50 εποχές. Στον επόμενο πίνακα που θα δείτε παρουσιάζονται τα καλύτερα αποτελέσματα όλων των εταιρειών, ως προς το MSE. Με λίγα λόγια η μεταβλητή του χρόνου δεν συμμετείχε ως κριτήριο στην διαδικασία.

Εταιρείες	Τεχνική	Mean Squared Error
Coca Cola	Linear Regression	0.0173
	SVM	0.0159
	LSTM	0.0010
AMD	Linear Regression	0.1209
	SVM	0.0004
	LSTM	0.0006
Microsoft	Linear Regression	0.2600
	SVM	0.2602
	LSTM	0.0035
Toyota Motors	Linear Regression	0.1309
	SVM	0.1291
	LSTM	0.0040
BMW	Linear Regression	0.1766
	SVM	0.1246
	LSTM	0.0056

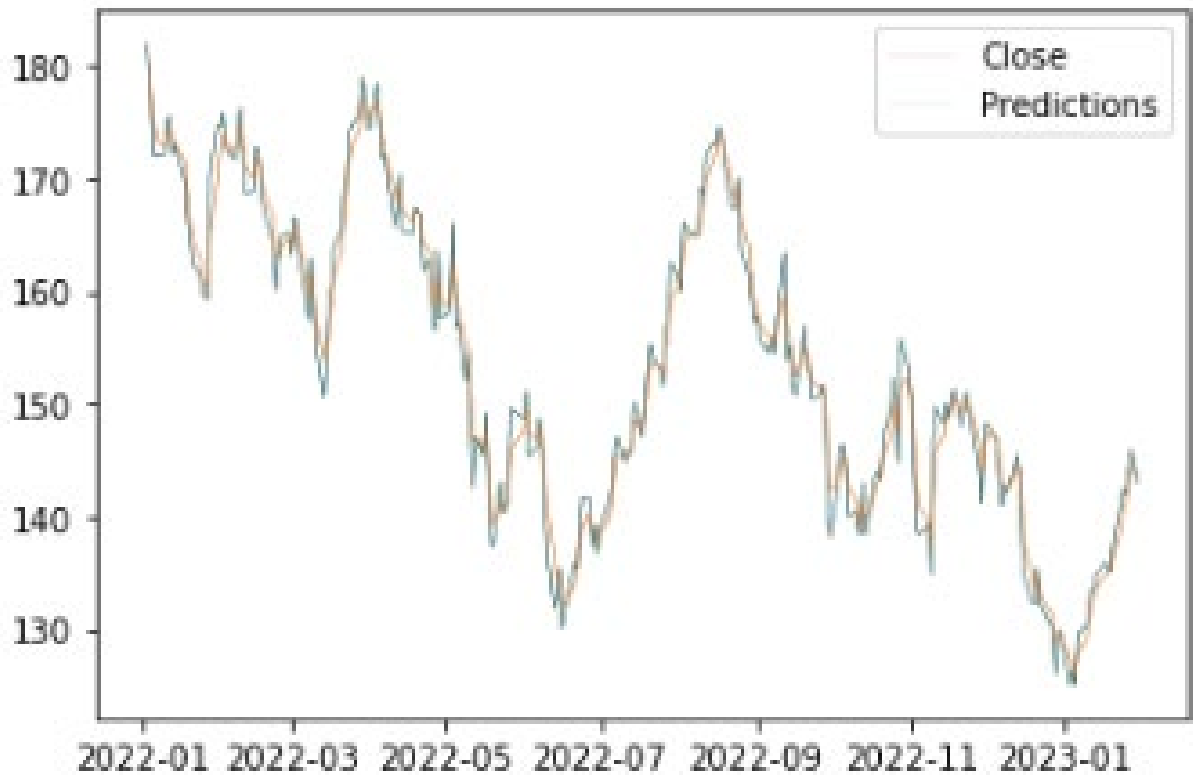
Amazon	Linear Regression	0.1100
	SVM	0.0040
	LSTM	0.0011
Apple	Linear Regression	0.0504
	SVM	0.0417
	LSTM	0.0152
Nvidia	Linear Regression	0.0162
	SVM	0.0128
	LSTM	0.0028

Πίνακας 5.5: Τελικά αποτελέσματα όλων των μοντέλων

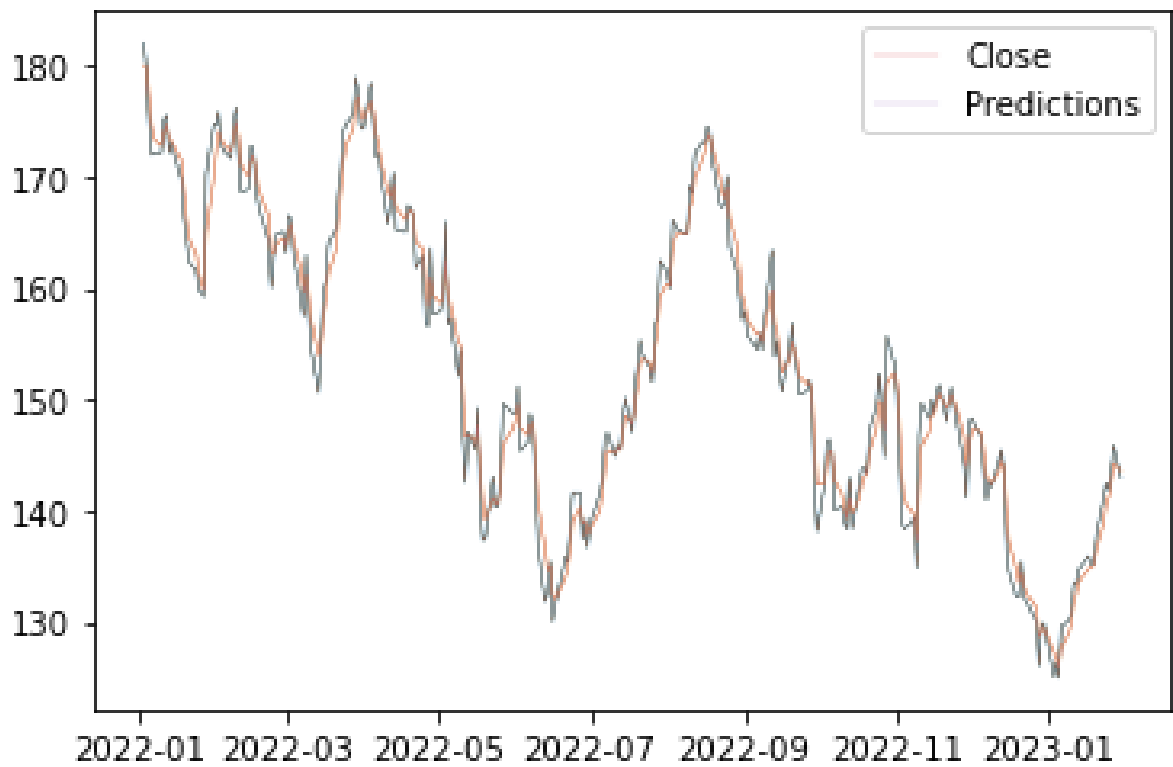
Ο πίνακας αυτός παρουσιάζει και τους τρεις αλγόριθμους στην βέλτιστη απόδοσή τους. Για τον LSTM χρησιμοποιήθηκε ως optimizer ο Adam και ως activation function η tanh. Για τις εταιρείες που είναι περισσότερα χρόνια εισηγμένες στο χρηματιστήριο και υπάρχουν περισσότερα δεδομένα, χρησιμοποιήθηκαν 40000 δεδομένα για εκπαίδευση, για τις πιο νέες χρησιμοποιήθηκαν 25000 αναλόγως την κάθε εταιρεία. Στο μοντέλο του SVM χρησιμοποιήσαμε τον RBF ως καλύτερο kernel για να επιτύχουμε την καλύτερη δυνατή απόδοση. Επίσης για την εκπαίδευση χρησιμοποιήσαμε ακριβώς τα δεδομένα που χρησιμοποιήσαμε και στον αλγόριθμο γραμμικής παλινδρόμησης. Παρατηρείται ότι από εταιρεία σε εταιρεία το MSE έχει κάποιες αποκλίσεις. Αυτό οφείλεται στην αστάθεια των δεδομένων. Η χρηματιστηριακή αγορά επηρεάζεται δραματικά από γεγονότα που δεν μπορούν να προβλεφθούν εύκολα από έναν αλγόριθμο. Αυτά περιλαμβάνουν οικονομικά, πολιτικά γεγονότα και κινήσεις διάφορων εταιρειών που βρίσκονται στον ίδιο κλάδο.

Στη συνέχεια παρουσιάζονται οι γραφικές παραστάσεις, στις οποίες συγκρίνουμε τις τιμές που έχουν οι μετοχές στο χρηματιστήριο στο τέλος της ημέρας (Close), με τις τιμές που προσπαθούμε να προβλέψουμε μέσω του προγράμματος μας (Prediction). Οι γραφικές παραστάσεις θα παρουσιάζονται για όλες τις εταιρείες από το 2022 και έπειτα. Οι συνολικές προβλέψεις σε κάποιες εταιρείες μπορεί να υπολογίζονται και πριν το 2022, αλλά για να έχουμε μία καλύτερα εικόνα της «γραφικής» ως πούμε παράστασης, θα επιλέξουμε να την οριοθετήσουμε στους 15 περίπου μήνες. Έχουμε υπ' όψη μας ότι συνάρτηση σφάλματος μας, το MSE, έχει μικρή διαφορά από τον αλγόριθμο Linear Regression, στον SVM και μετά στον LSTM. Αυτό καθιστά τις γραφικές παραστάσεις μας πολύ όμοιες.

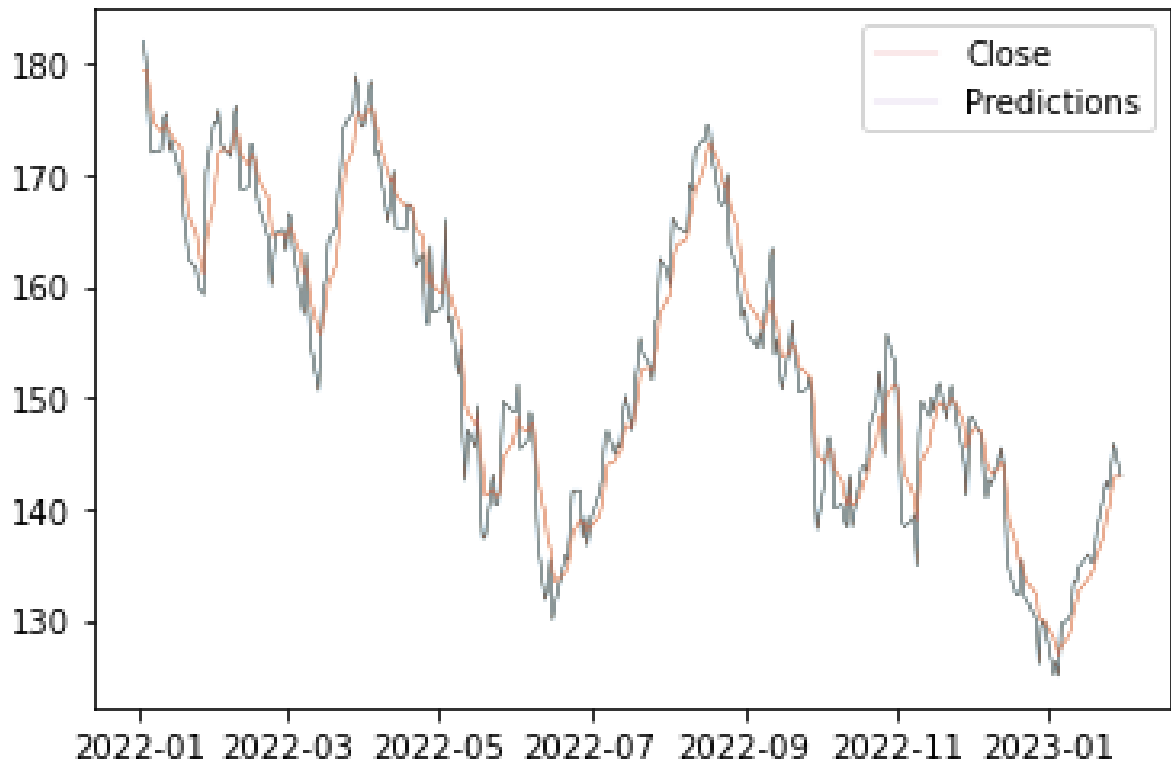
Apple



Σχήμα 5.1 : Ανάλυση πραγματικές τιμών και προβλέψεων της Apple με LSTM

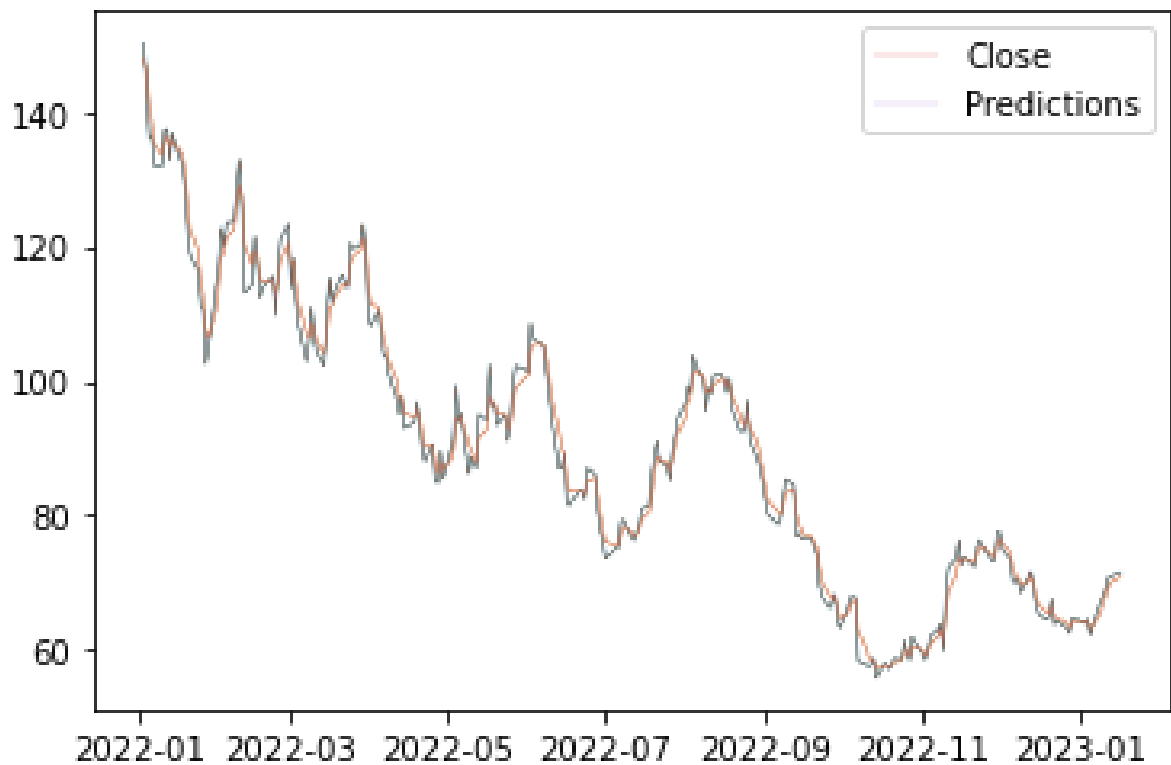


Σχήμα 5.2 : Ανάλυση πραγματικές τιμών και προβλέψεων της Apple με SVM

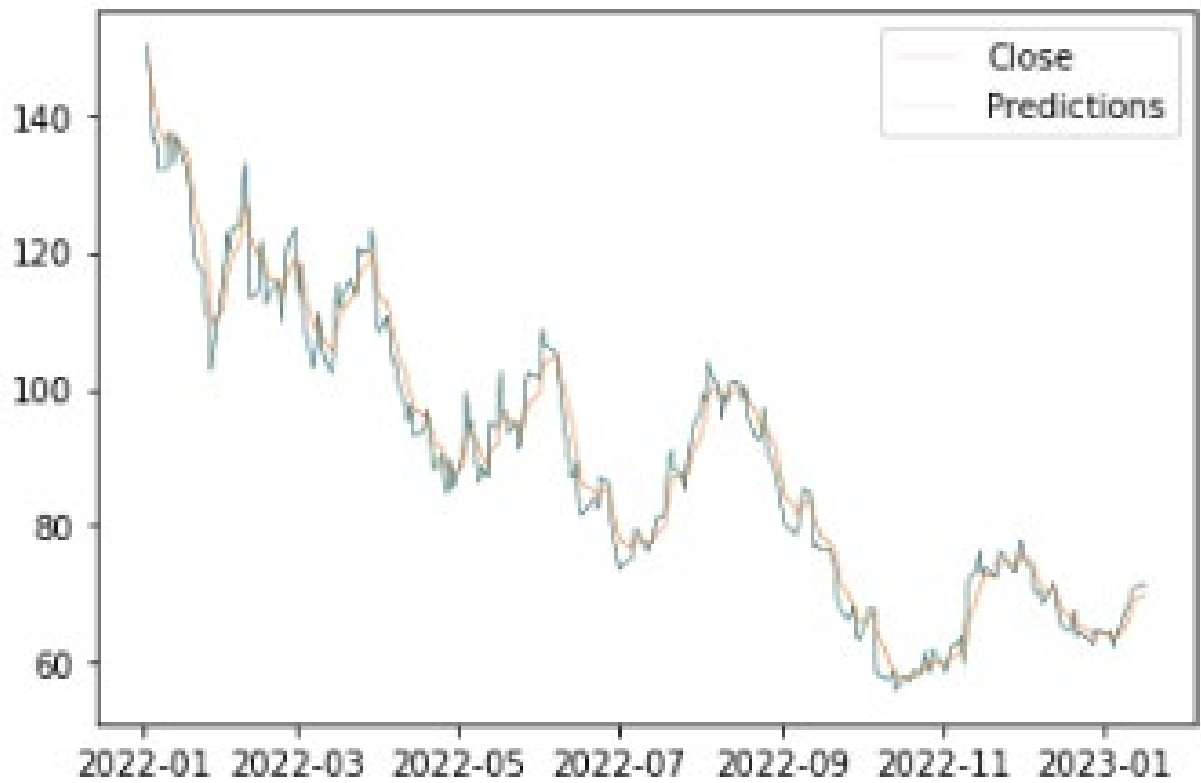


Σχήμα 5.3 : Ανάλυση πραγματικές τιμών και προβλέψεων της Apple με Γραμμική Παλινδρόμηση

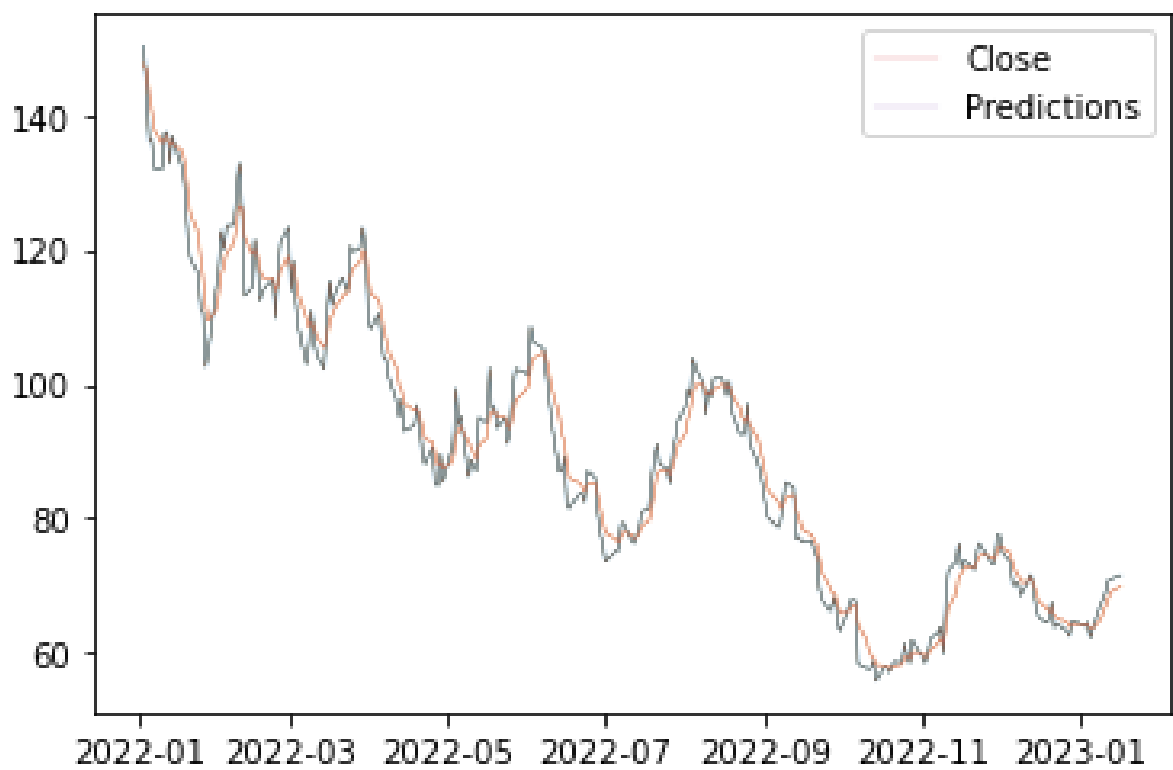
AMD



Σχήμα 5.4 : Ανάλυση πραγματικές τιμών και προβλέψεων της AMD με LSTM

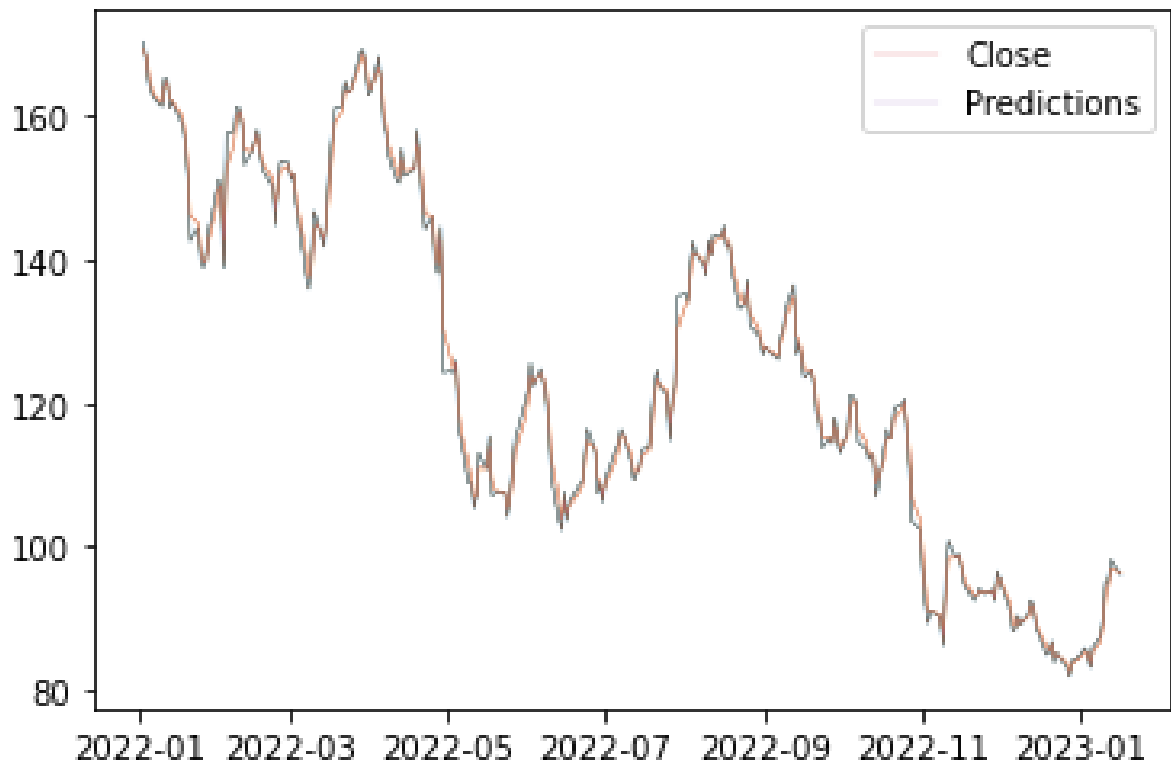


Σχήμα 5.5 : Ανάλυση πραγματικές τιμών και προβλέψεων της AMD με SVM

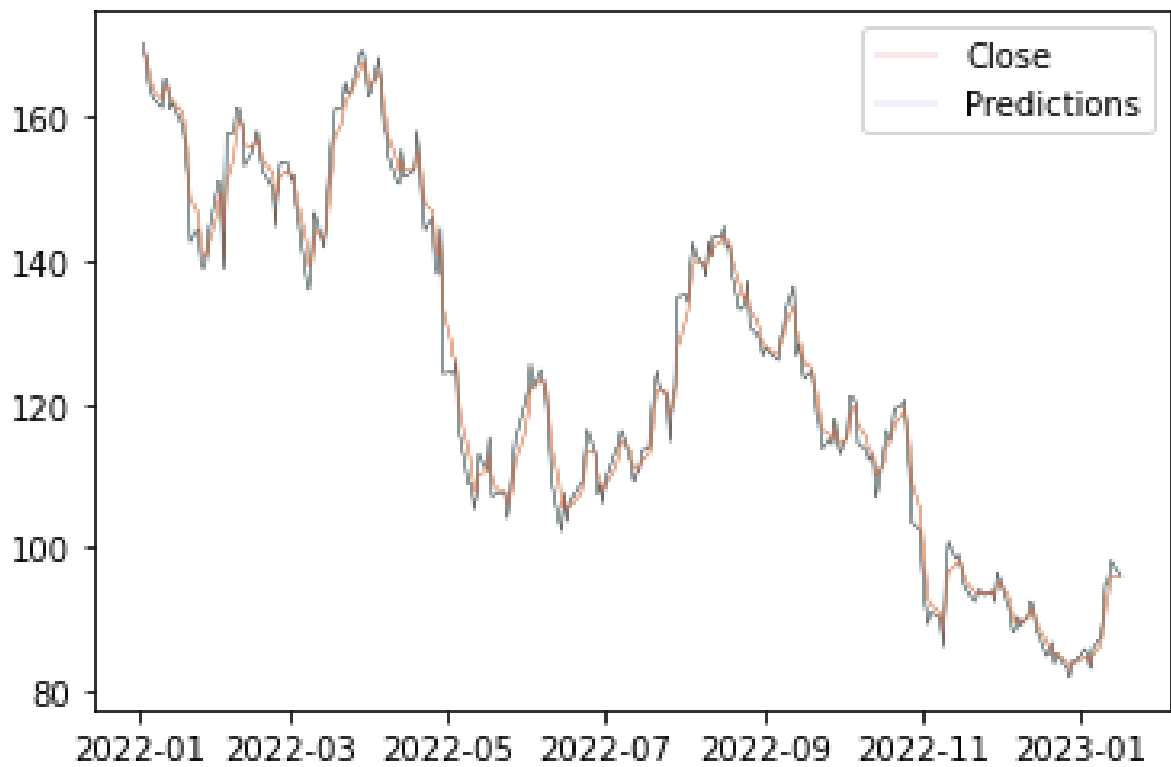


Σχήμα 5.6 : Ανάλυση πραγματικές τιμών και προβλέψεων της AMD με Γραμμική Παλινδρόμηση

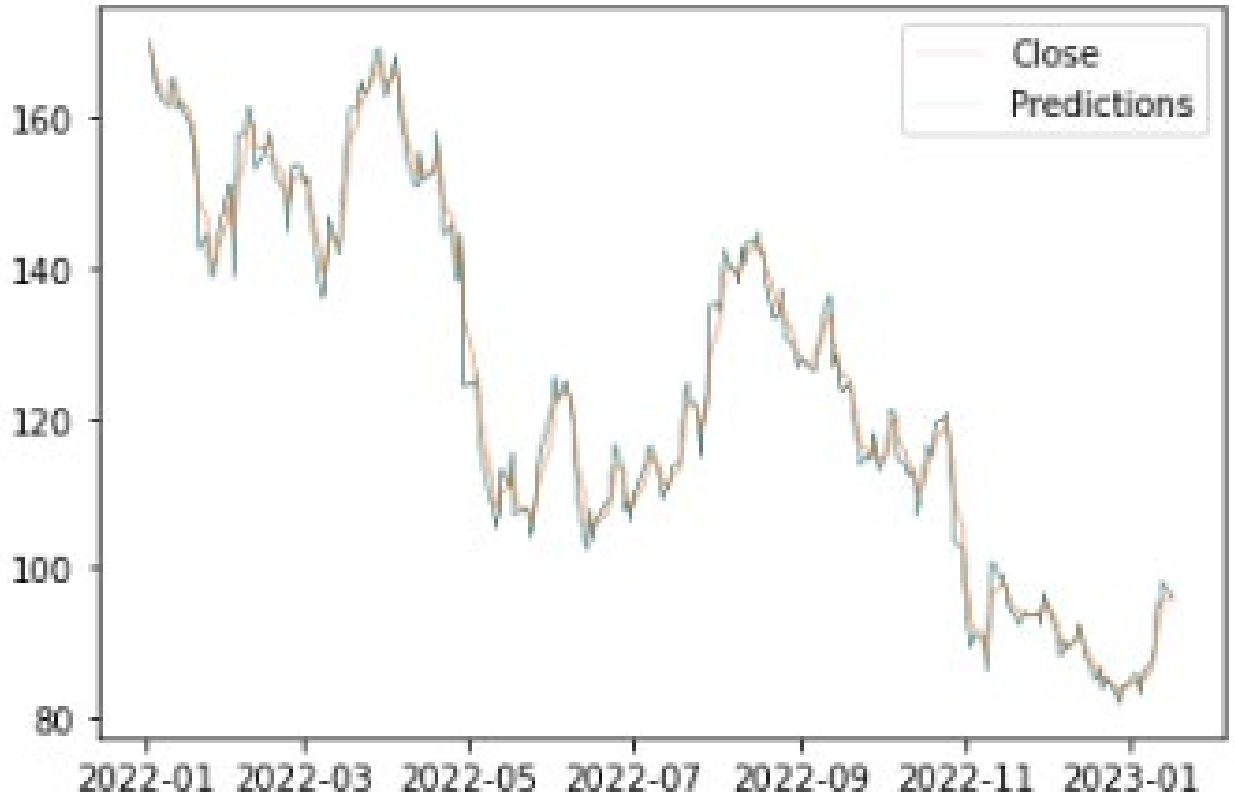
AMAZON



Σχήμα 5.7 : Ανάλυση πραγματικές τιμών και προβλέψεων της Amazon με LSTM

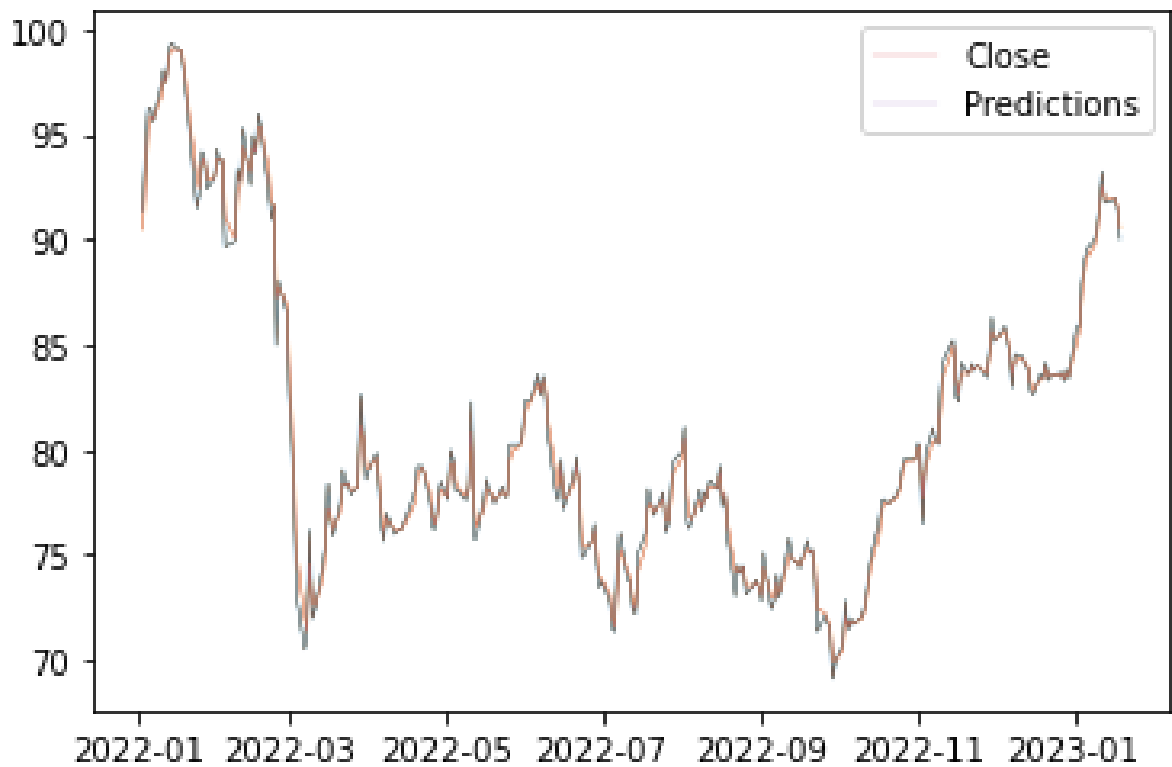


Σχήμα 5.8 : Ανάλυση πραγματικές τιμών και προβλέψεων της Amazon με SVM

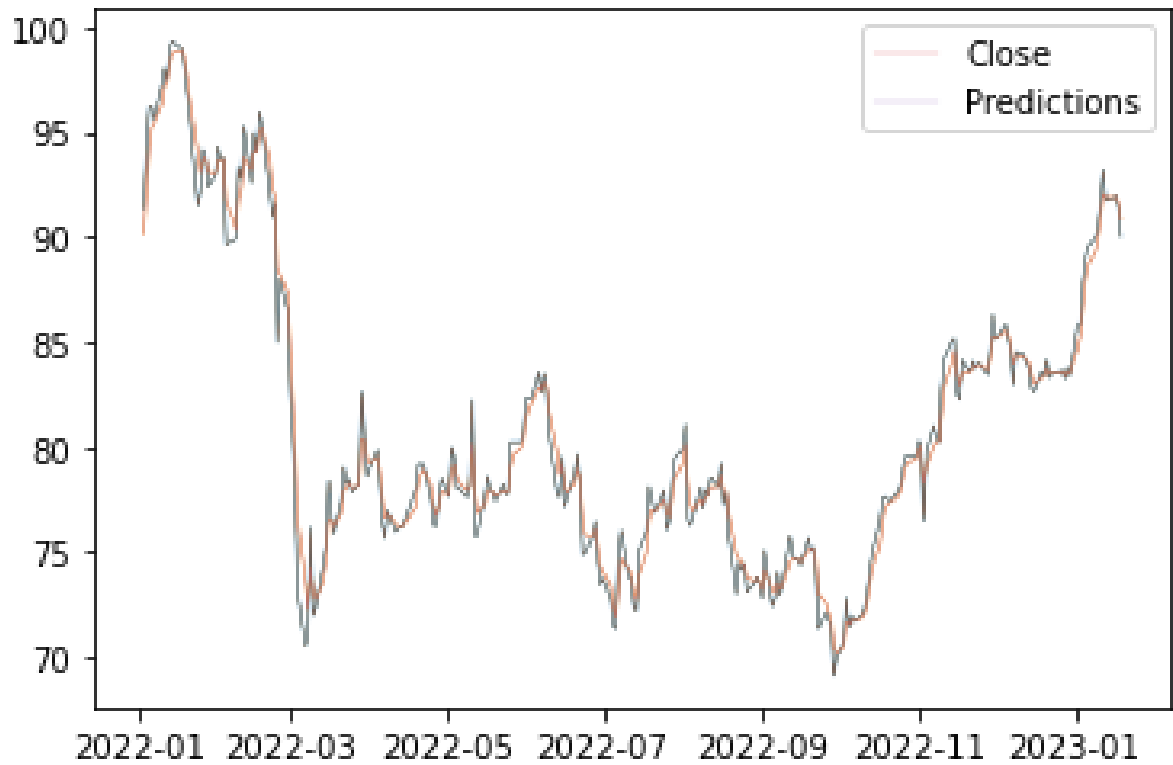


Σχήμα 5.9 : Ανάλυση πραγματικές τιμών και προβλέψεων της Amazon με Γραμμική Παλινδρόμηση

BMW



Σχήμα 5.10 : Ανάλυση πραγματικές τιμών και προβλέψεων της BMW με LSTM

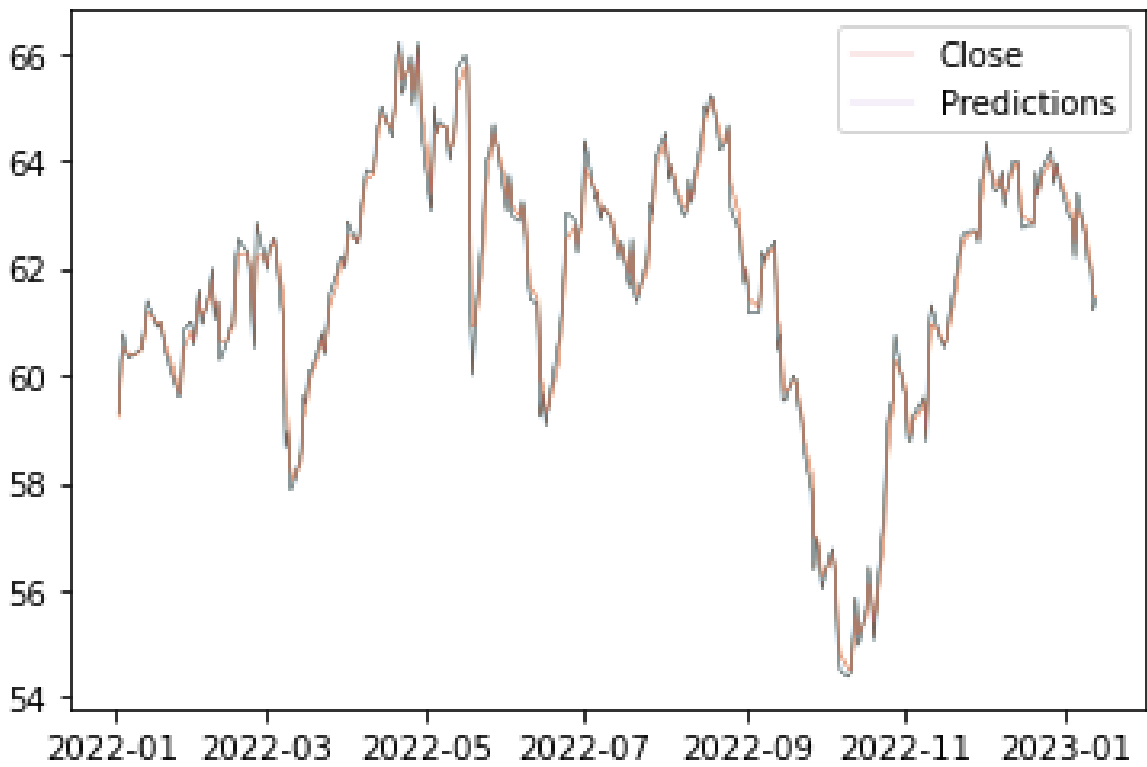


Σχήμα 5.11 : Ανάλυση πραγματικές τιμών και προβλέψεων της BMW με SVM

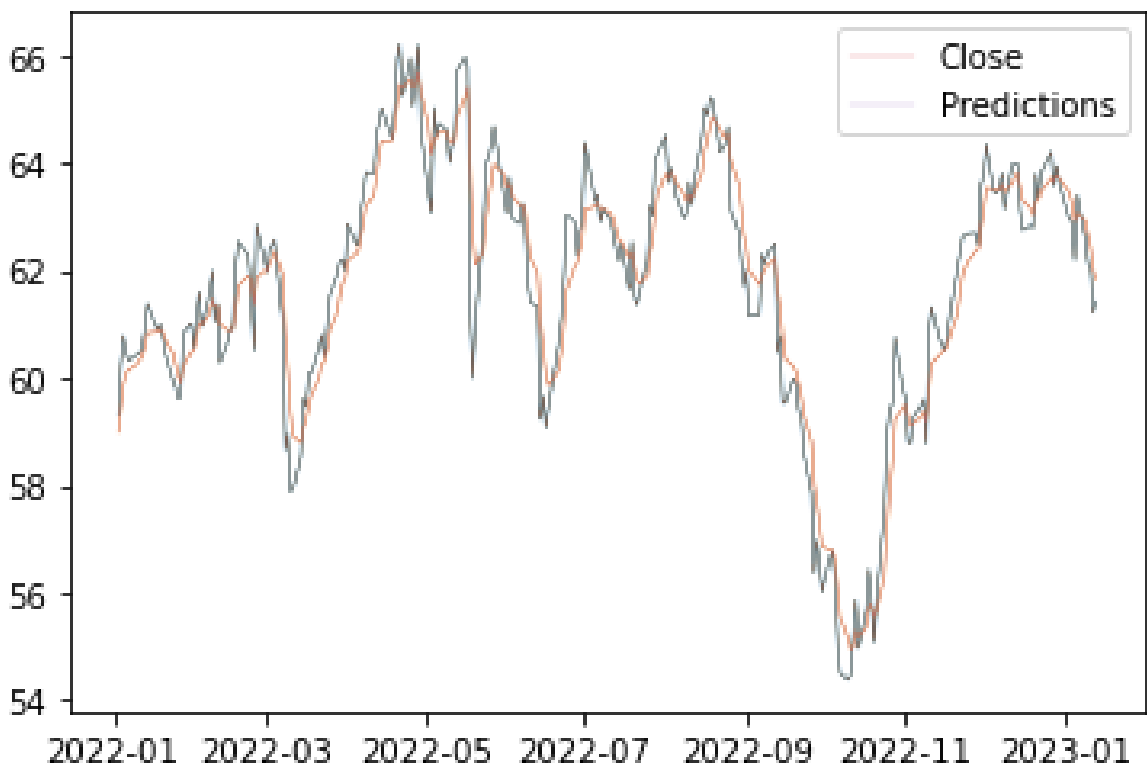


Σχήμα 5.12 : Ανάλυση πραγματικές τιμών και προβλέψεων της BMW με Γραμμική Παλινδρόμηση

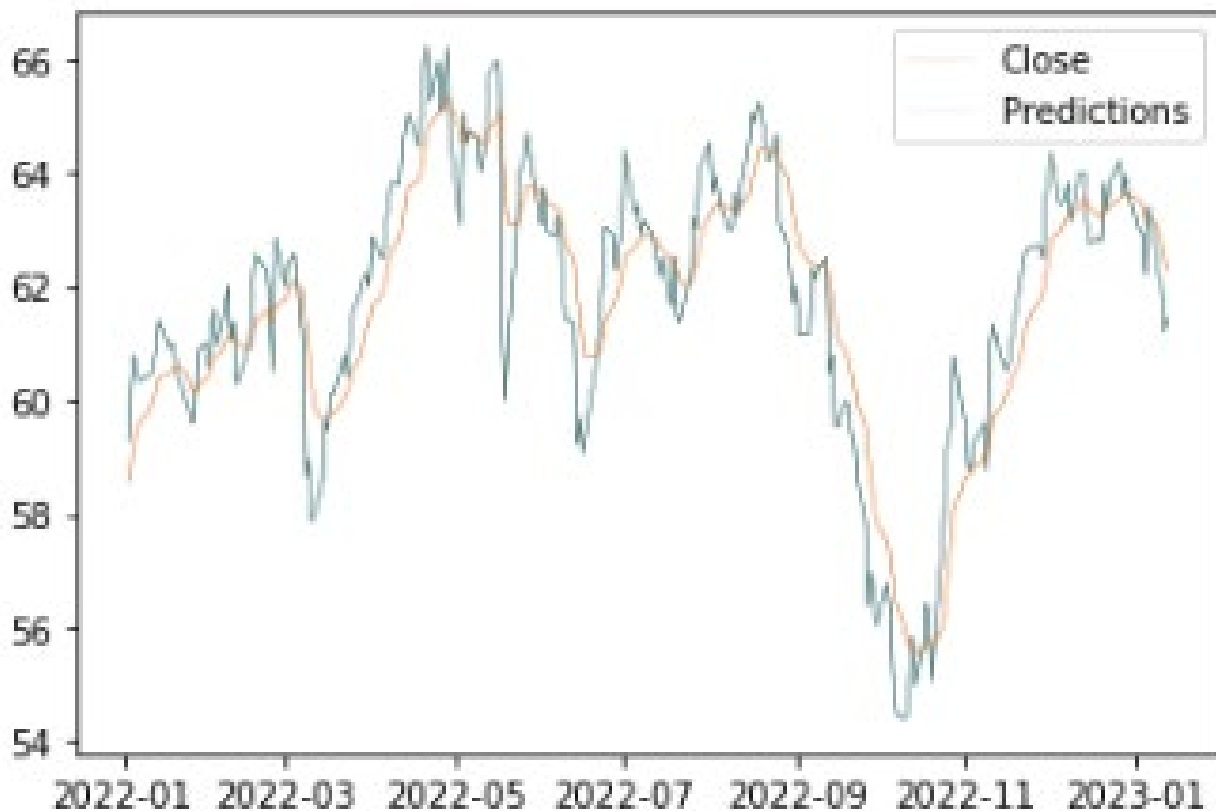
COCA COLA



Σχήμα 5.13 : Ανάλυση πραγματικές τιμών και προβλέψεων της COCA COLA με LSTM

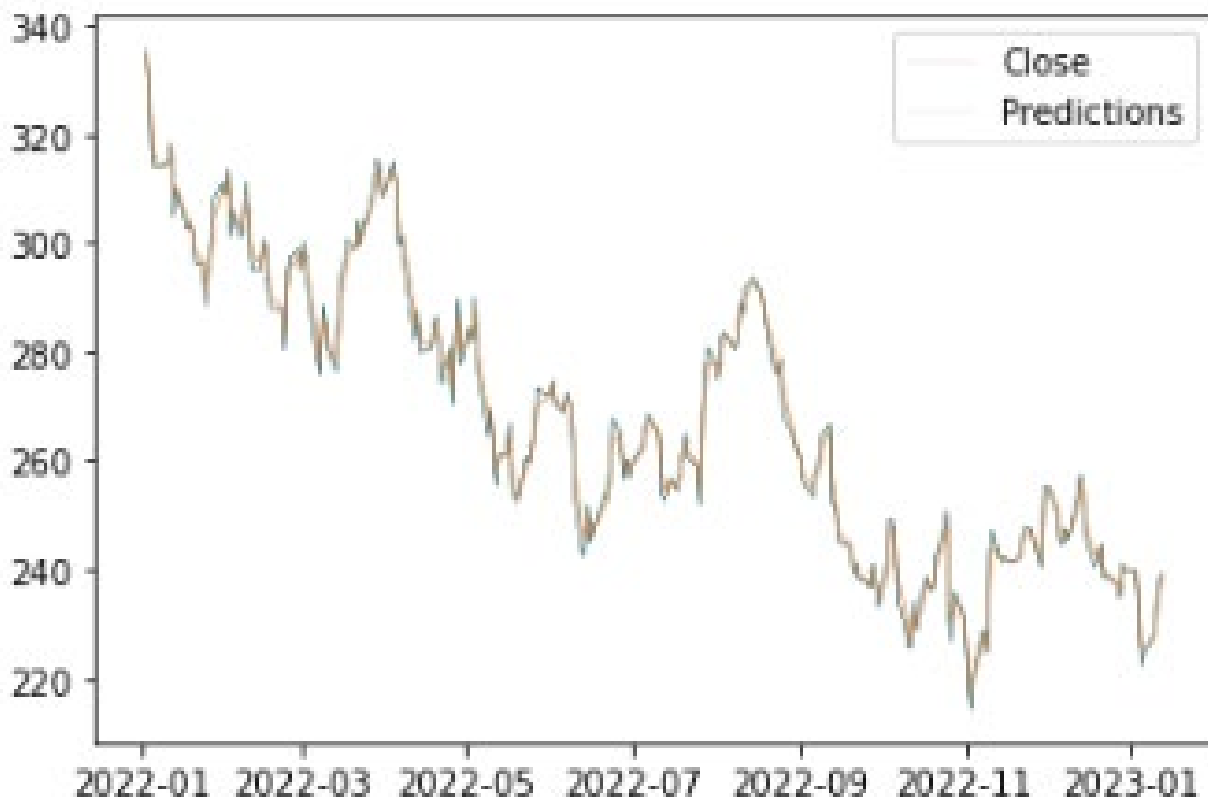


Σχήμα 5.14 : Ανάλυση πραγματικές τιμών και προβλέψεων της COCA COLA με SVM

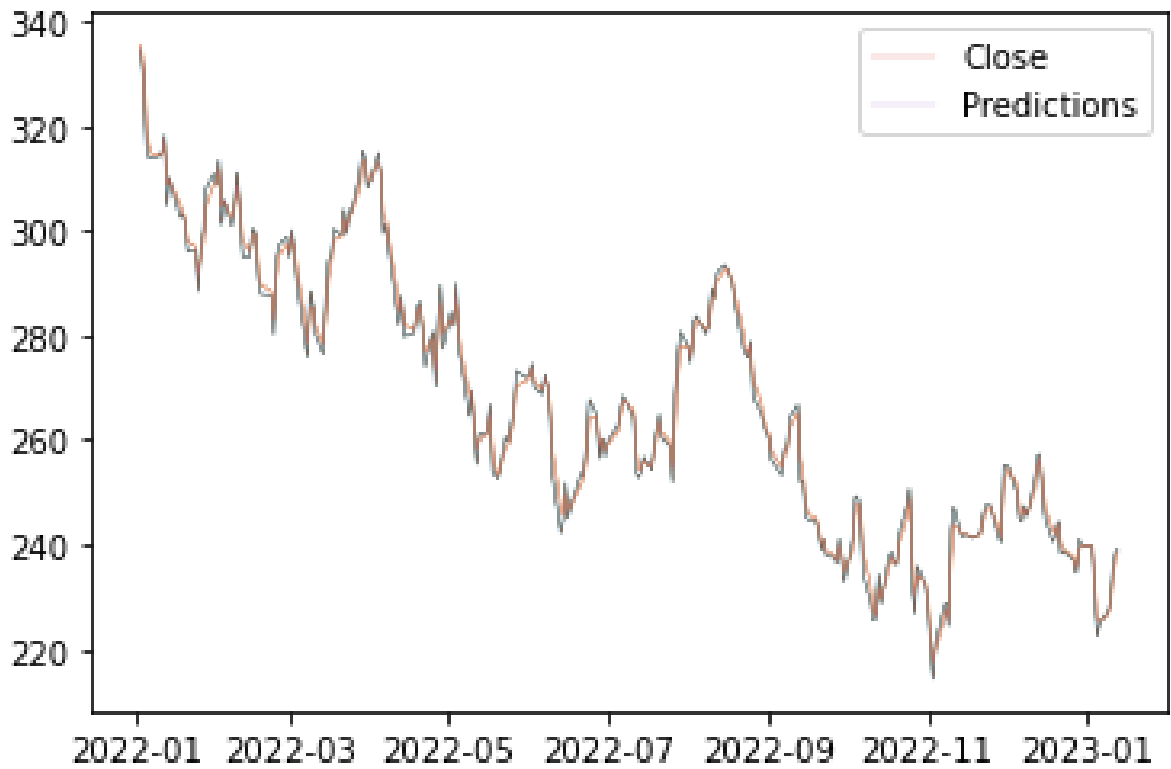


Σχήμα 5.15 : Ανάλυση πραγματικές τιμών και προβλέψεων της COCA COLA με Γραμμική Παλινδρόμηση

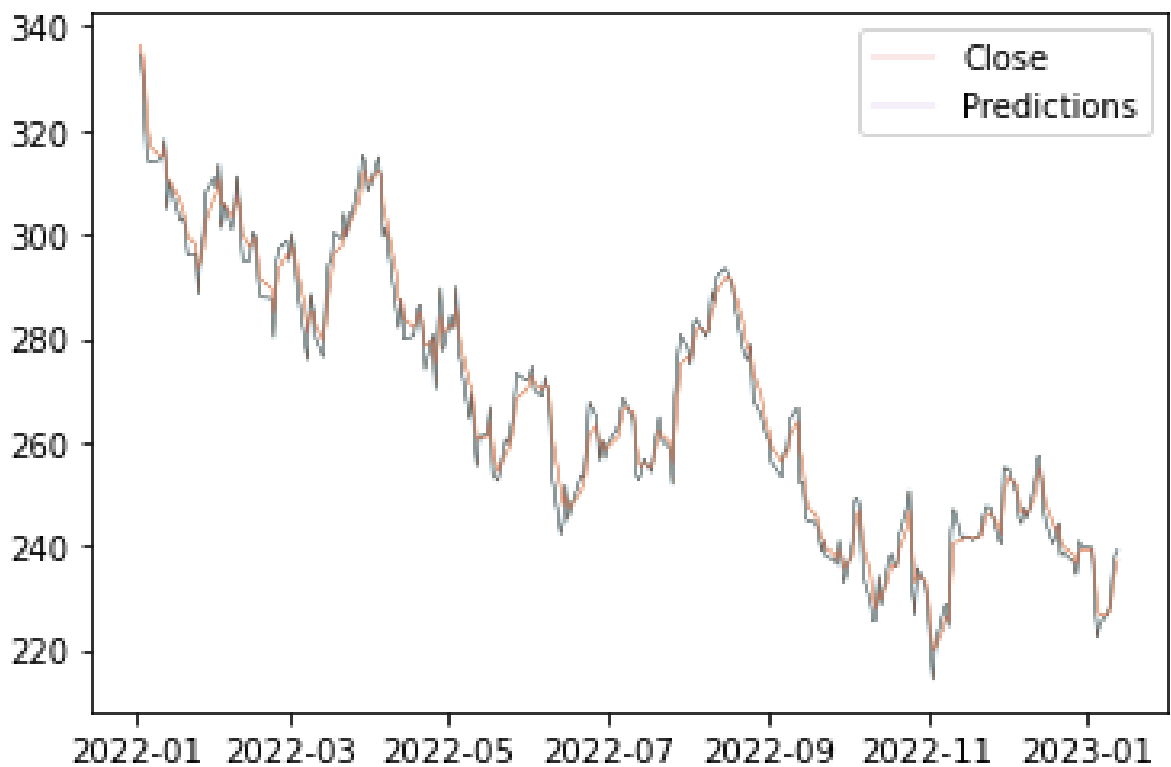
MICROSOFT



Σχήμα 5.16 : Ανάλυση πραγματικές τιμών και προβλέψεων της Microsoft με LSTM

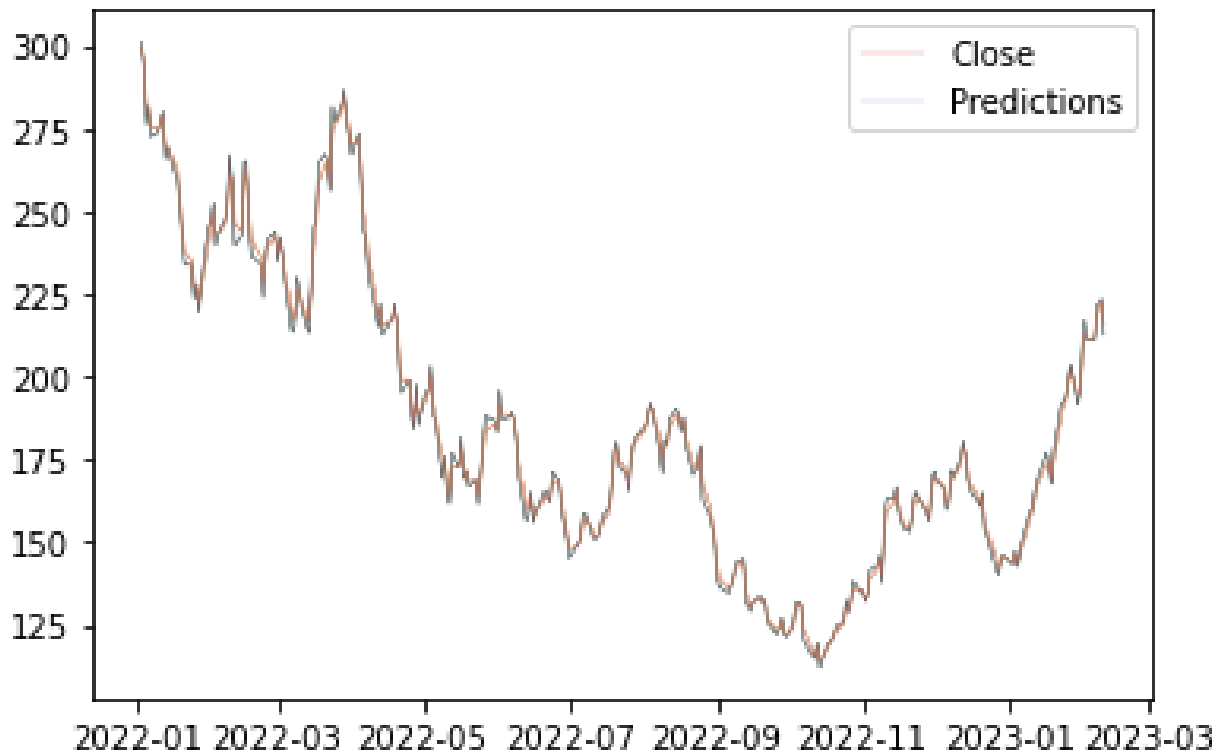


Σχήμα 5.17 : Ανάλυση πραγματικές τιμών και προβλέψεων της Microsoft με SVM

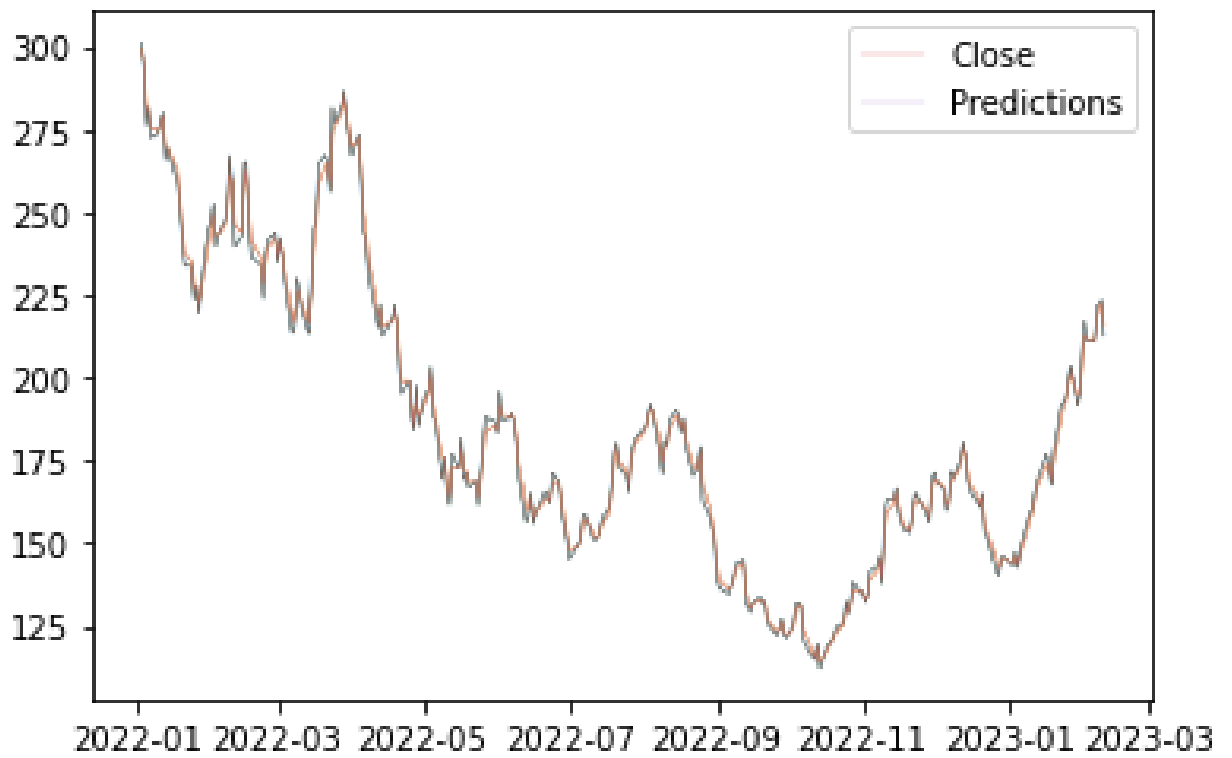


Σχήμα 5.18 : Ανάλυση πραγματικές τιμών και προβλέψεων της Microsoft με Γραμμική Παλινδρόμηση

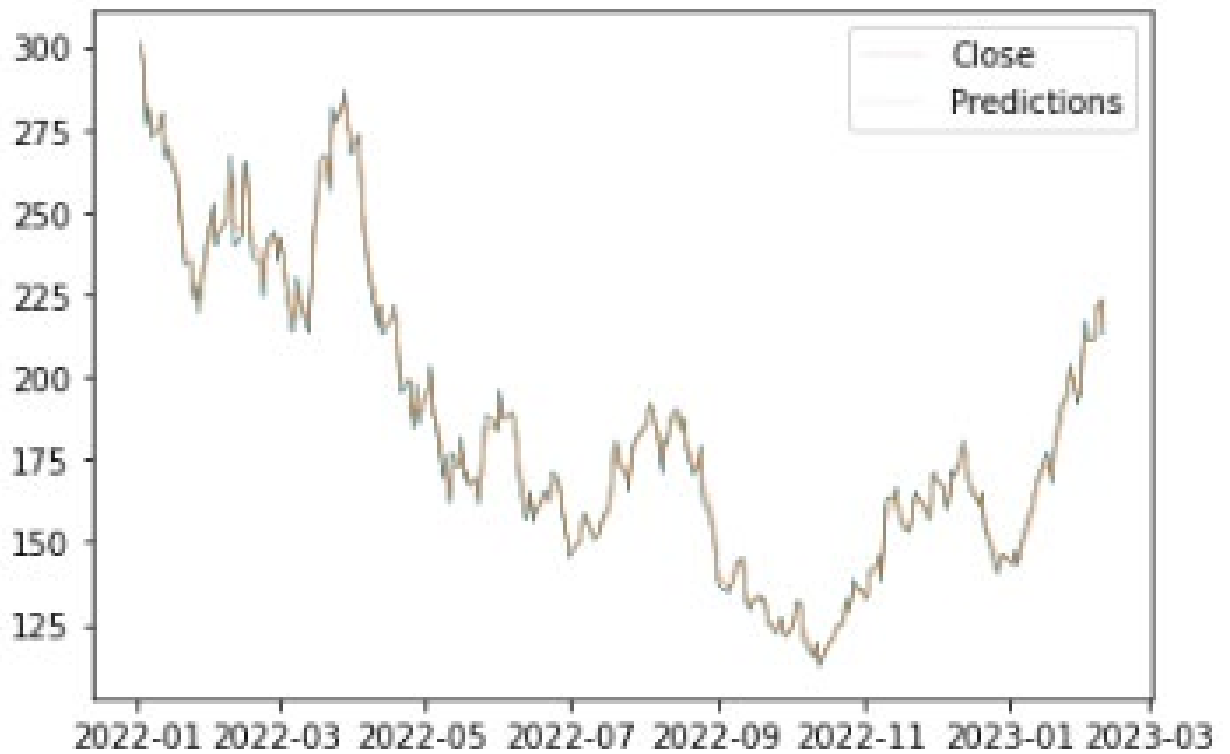
NVIDIA



Σχήμα 5.19 : Ανάλυση πραγματικές τιμών και προβλέψεων της Nvidia με LSTM

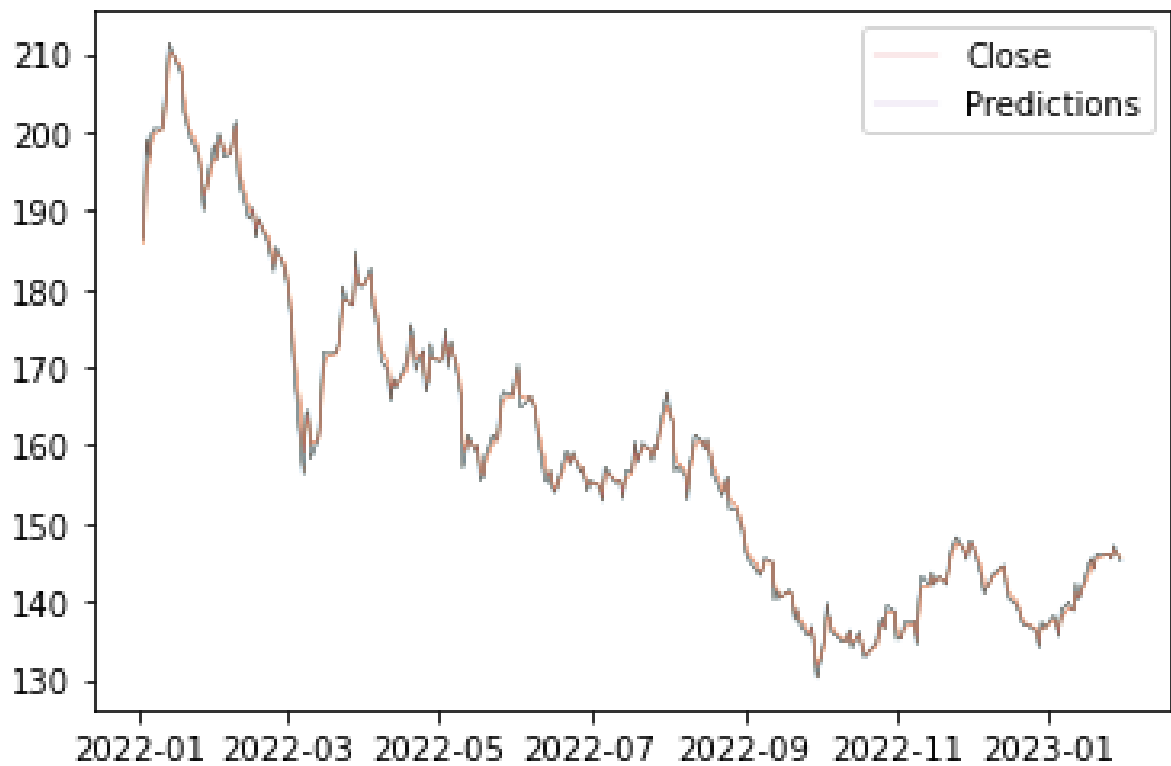


Σχήμα 5.20 : Ανάλυση πραγματικές τιμών και προβλέψεων της Nvidia με SVM

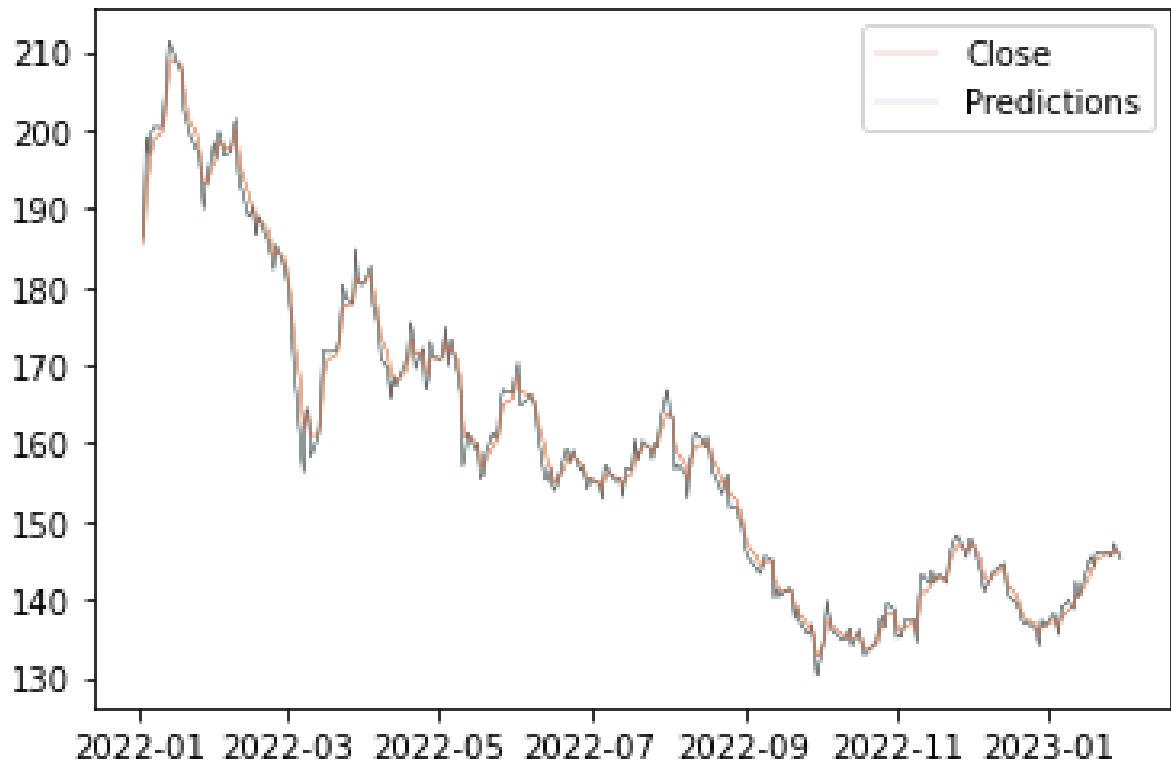


Σχήμα 5.21 : Ανάλυση πραγματικές τιμών και προβλέσεων της Nvidia με Γραμμική Παλινδρόμηση

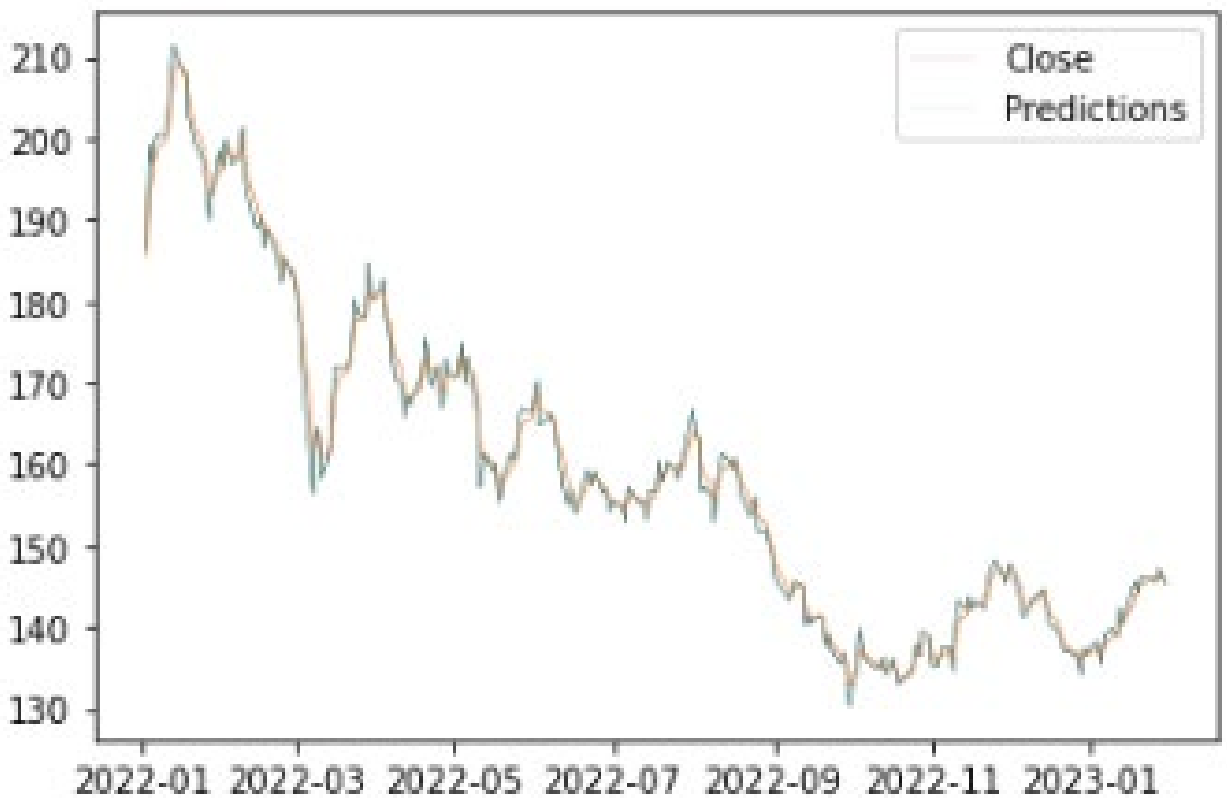
TOYOTA MOTORS



Σχήμα 5.22 : Ανάλυση πραγματικές τιμών και προβλέσεων της Toyota Motors με LSTM



Σχήμα 5.23 : Ανάλυση πραγματικές τιμών και προβλέψεων της Toyota Motors με SVM



Σχήμα 5.24 : Ανάλυση πραγματικές τιμών και προβλέψεων της Toyota Motors με Γραμμική Παλινδρόμηση

5.6 Συμπεράσματα

Είναι εμφανές πως ένας αλγόριθμος Μηχανικής Μάθησης όπως ο Linear Regression έχει τα όρια του. Τα αποτελέσματα που επιφέρει δεν φτάνουν τις δυνατότητες ενός LSTM μοντέλου ή ακόμα και του SVM μοντέλου. Το θέμα είναι πόσο ωφέλιμο είναι αυτό το ποσοστό διαφοράς που έχουν τα μοντέλα. Το μοντέλο της γραμμικής παλινδρόμησης είναι το πιο απλοϊκό μοντέλο που χρησιμοποιήσαμε και ως θεωρία και ως κώδικας. Τα στατιστικά το αποδεικνύουν ως το πιο αδύναμο απ τα τρία προγράμματα.

Το SVM μπορεί να θεωρηθεί ότι βρίσκεται ανάμεσα στα 3 μοντέλα σε θέμα πολυπλοκότητας και αυτό φαίνεται και στον πίνακα 5.5. Προφανώς και τα τρία μοντέλα έχουν χώρο για αναβάθμιση, ειδικά το LSTM, όπου υπάρχουν πολλές διαφορετικές μεθόδους για να βελτιώσουν την απόδοσή του. Αυτό θα μπορούσε να αποτελέσει ένα μεγαλύτερο χάσμα στο σφάλμα τους. Αυτοί οι τρόποι θα εξηγηθούν παρακάτω. Έχοντας υλοποιήσει αυτούς τους αλγορίθμους, ένα συμπέρασμα που αντλώ είναι ότι οι αλγόριθμοι Βαθιάς Μάθησης μπορούν να παράγουν καλύτερο αποτέλεσμα από τους αλγορίθμους απλής Μηχανικής Μάθησης, όσο δίνεται περισσότερος χρόνος και δυνατότητα για δεδομένα. Αυτό όμως εκφράζει μία μεγάλη απορία όσο αναφορά την αποτελεσματικότητά τους : Αξίζει η μεγαλύτερη αναμονή για να τρέξουμε ένα κώδικα Βαθιάς Μάθησης ενάντια ενός απλής Μηχανικής Μάθησης ;

Ο κώδικας του Linear Regression υλοποιείται σε μόλις 5 δευτερόλεπτα, ο SVM χρειάζεται λίγο περισσότερο χρόνο, που και πάλι δεν υπερβαίνει το ένα λεπτό, για τον LSTM όμως τα 5 δευτερόλεπτα ή ακόμα και το ένα λεπτό, δεν αρκεί ούτε για την υλοποίηση μίας εποχής. Επίσης στην δικιά μας περίπτωση τα δεδομένα μας δεν θεωρούνται πολλά συγκριτικά με τα δεδομένα που μπορεί να χρησιμοποιεί για επιχείρηση για το δικό της μοντέλο. Τέτοιον πολύπλοκο κώδικας μπορεί να χρειάζονται από ώρες έως και μέρες για να υλοποιηθούν.

Για να απαντήσουμε την ερώτησή μας όμως, και οι τρεις αλγόριθμοι είναι χρήσιμοι και μπορούν να εφαρμοστούν. Αναλόγως την αιτία που προσπαθούμε να δραστηριοποιήσουμε, ο ένας υπερτερεί του άλλου. Αν χρειαζόμαστε να ελαχιστοποιήσουμε το σφάλμα μας όσο το δυνατόν περισσότερο και δεν υπάρχει όριο για τον χρόνο, τότε ένας πολύπλοκος αλγόριθμος Βαθιάς Μάθησης είναι η καλύτερη επιλογή μας. Τέτοιες περιπτώσεις παρουσιάζονται συχνά σε θέματα ιατρικού περιεχομένου όπου το παραμικρό λάθος μπορεί να καταλήξει να αφαιρέσει μία ανθρώπινη ζωή. Στον χώρο των χρηματοοικονομικών όμως, συναλλαγές εκτελούνται καθημερινά. Το χρηματιστήριο είναι ασταθές και οι μετοχές αλλάζουν δραστικά από μέρα σε μέρα. Η επιλογή να λειτουργούμε με σφάλμα 0.2 αντί για 0.4, αλλά να μπορούμε να λειτουργούμε άμεσα, υπερτερεί από την περίπτωση να αργήσουμε να κάνουμε την αγορά την επόμενη μέρα κι ας είναι πιο σίγουρη. Αυτό όμως σημαίνει ότι το μοντέλο της γραμμικής παλινδρόμησης, είναι το πιο αχρείαστο. Έχει το υψηλότερο σφάλμα και άμα το συγκρίνουμε με το SVM, δεν μας γλιτώνει και χρόνο. Άρα μπορούμε να αποκλείσουμε τον Linear Regression από τις επιλογές μας και να τον αντικαταστήσουμε με ένα μοντέλο SVM που μας προσφέρει χαμηλότερο σφάλμα λειτουργώντας στο ίδιο χρονικό διάστημα.

Οι αλγόριθμοι Linear Regression και LSTM αντίστοιχα, αντιμετωπίζουν κάποια προβλήματα ως προς την φύση τους και την εκτέλεσή τους, παρακάτω παρουσιάζονται τα πιο σοβαρά :

Linear Regression :

- Γραμμικότητα : Ο αλγόριθμος αυτός προϋποθέτει ότι η σχέση μεταξύ των προβλέψεων και της απόκρισης είναι γραμμική, κάτι που μπορεί να μην συμβαίνει πάντα στα δεδομένα της χρηματιστηριακής αγοράς.
- Outliers : Οι ακραίες τιμές μπορεί να έχουν μεγάλο αντίκτυπο στα αποτελέσματα του μοντέλου, καθώς το μοντέλο είναι ευαίσθητο σε ακραίες τιμές.

SVM :

- Ευαισθησία στην επιλογή παραμέτρων: Ο SVM απαιτεί τον συντονισμό πολλών παραμέτρων, όπως την επιλογή του πυρήνα (kernel), την παράμετρο κανονικοποίησης C και τον συντελεστή gamma του kernel. Η απόδοση του μοντέλου μπορεί να είναι ευαίσθητη στην επιλογή αυτών των παραμέτρων, γεγονός που μπορεί να δυσκολέψει την εύρεση των βέλτιστων τιμών.
- Εύρος Δεδομένων: Τα SVM απαιτούν σημαντικούς υπολογιστικούς πόρους και αρκετή μνήμη για την διαδικασία της εκπαίδευσης, ιδιαίτερα όταν ασχολούνται με μεγάλα σύνολα δεδομένων. Αυτό μπορεί να κάνει τα SVM ακατάλληλα για ορισμένες εφαρμογές όπου υπάρχουν αυστηροί περιορισμοί χρόνου ή μνήμης.

LSTM :

- Χρόνος εκπαίδευσης: Ο LSTM μπορεί να είναι δύσκολος ως προς την εκπαίδευση και συνήθως απαιτεί μεγάλες ποσότητες δεδομένων και υπολογιστικών πόρων για την επίτευξη καλής απόδοσης, και αυτό καταλήγει στο να καταναλώνει πολύ χρόνο.
- Απαίτηση δεδομένων: Ο LSTM έχει ανάγκη πολλά δεδομένα για την εκπαίδευση του για να λειτουργήσει ποιοτικά. Αυτό συνδέεται και με τον πολύ χρόνο εκπαίδευσης. Άρα το μοντέλο αυτό χρειάζεται και μεγάλο αρχείο ως τροφοδοσία, δεν μπορεί να χρησιμοποιηθεί για το οποιοδήποτε πρόβλημα.
- Cuda Architecture: Για να λειτουργήσει το πακέτο tensorflow χρειάζεται η κάρτα γραφικών να έχει Cuda Architecture. Αυτή η αρχιτεκτονική υπάρχει αυστηρά στις κάρτες γραφικών της Nvidia, μπορεί να λειτουργήσει και χωρίς τέτοια GPU, αλλά ο χρόνος για να τρέξει το πρόγραμμα πολλαπλασιάζεται δραματικά.

Όσον αφορά τα προβλήματα που αναφέρθηκαν προηγουμένως, υπάρχουν πιθανές λύσεις. Για την γραμμική παλινδρόμηση και το θέμα της γραμμικότητας υπάρχει η παραλαγή του ως Polynomial. Αυτό δίνει στο μοντέλο την δυνατότητα να χειρίζεται και τις μη γραμμικές σχέσεις. Για το επόμενο πρόβλημα μπορεί να χρησιμοποιηθεί η ποσοστιαία απώλεια (quantile loss), για να διαχειριστούν τα ακραία σημεία πιο δυναμικά. Για το SVM πρέπει να υπάρχει απόλυτη γνώση των δεδομένων ώστε να μπορεί αυτός που το χειρίζεται να επηρεάσει κατάλληλα τους παραμέτρους.

Για το LSTM, ο χρόνος εκπαίδευσης μπορεί να ελαχιστοποιηθεί δραματικά με την χρήση καλύτερου συστήματος. Όσο μεγαλύτερη η ισχύς της κάρτας γραφικών του υπολογιστή, τόσο πιο γρήγορα θα επιτευχθεί και το training. Υπάρχουν και τρόποι που θα επηρεάσουν

αρνητικά την απόδοση του μοντέλου, αλλά θα γλιτώσουν χρόνο, αυτοί οι τρόποι είναι οι εξής:

- Batch Size: Να μειωθεί ο αριθμός του ώστε να υπάρχει λιγότερη ανάγκη για μνήμη.
- Dense Layer: Να υπάρχει ένα μόνο κρυφό στρώμα (Hidden Layer).

Οι τεχνικές Βαθιάς Μάθησης υπερτερούν ως προς τις τεχνικές Μηχανικής Μάθησης για την πρόβλεψη δεδομένων χρηματιστηρίου. Το σφάλμα στο μοντέλο LSTM συγκεκριμένα είναι χαμηλότερο σε όλες τις περιπτώσεις με κάποιες να είναι λίγο πιο κοντά στο SVM . Παρουσιάζονται διάφοροι προβληματισμοί στο προηγούμενο κεφάλαιο, κυρίως ως προς την φύση του χρηματιστηρίου όπως και μέθοδοι που μπορούν να κάνουν πιο ισχυρό το σύστημα.

Βιβλιογραφία

[1] "Τεχνητή Νοημοσύνη", Ι. Βλαχάβας, Π. Κεφαλάς, Ν.Βασιλειάδης, Φ. Κόκκορας, Η. Σακελλάριου, Εκδόσεις Πανεπιστημίου Μακεδονίας, 4η Έκδοση Ιούνιος 2020

[2] "The Elements of Statistical Learning Data Mining, Inference, and Prediction", Trevor Hastie, Robert Tibshirani, Jerome Friedman, 2η Έκδοση 2017

[3] Τεχνητή Νοημοσύνη, Πηγή:

<https://www.g2.com/articles/types-of-artificial-intelligence>

[4] Τι είναι η Μηχανική Μάθηση, Πηγή:

<https://www.csc.com.gr/machine-learning-%CE%BC%CE%B7%CF%87%CE%B1%CE%BD%CE%B9%CE%BA%CE%AE-%CE%BC%CE%AC%CE%B8%CE%B7%CF%83%CE%B7-%CF%84%CE%B9-%CE%B5%CE%AF%CE%BD%CE%B1%CE%B9/>

[5] Ταξινομητές Μηχανικής Μάθησης, Πηγή :

<https://medium.com/fuzz/machine-learning-classification-models-3040f71e2529>

[6] Τι είναι η γραμμική παλινδρόμηση, Πηγή :

<https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/what-is-linear-regression/>

[7] Τι είναι ο Εκθετικός κινούμενος μέσος όρος, Πηγή :

<https://www.fidelity.com/learning-center/trading-investing/technical-analysis/technical-indicator-guide/ema>

[8] Μηχανές Διαωσμάτων Υποστήριξης, Πηγή :

<https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>

[9] Μηχανές Διαωσμάτων Υποστήριξης για Παλινδρόμηση, Πηγή :

https://www.researchgate.net/publication/253804636_Sea_Surface_Temperature_prediction_via_Support_Vector_Machines_combined_with_Particle_Swarm_Optimization

[10] Τι είναι η Βαθιά Μάθηση, Πηγή:

<https://iguru.gr/einai-kai-pos-leitourgei-vathia-mathisi/>

[11] AI vs ML vs DL, Πηγή:

<https://ai.plainenglish.io/artificial-intelligence-vs-machine-learning-vs-deep-learning-whats-the-difference-dccce18efe7f>

[12] Νευρωνικά Δίκτυα, Πηγή:

<https://www.ibm.com/uk-en/cloud/learn/neural-networks>

[13] RNN, Πηγή:

<https://www.ibm.com/topics/recurrent-neural-networks>

[14] LSTM για παλινδρόμηση, Πηγή:

<https://bobrupakroy.medium.com/lstms-for-regression-cc9b6677697f>

[15] CNN, Πηγή:

<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

[16] Βελτιστοποιητές :

<https://medium.com/mllearning-ai/optimizers-in-deep-learning-7bf81fed78a0>

[17] Οπισθοδιάδοση :

<https://www.guru99.com/backpropagation-neural-network.html>

[18] Μέσο Τετραγωνικό Σφάλμα, Πηγή:

<https://deepchecks.com/glossary/mean-square-error-mse/>

[19] Μέσο Απόλυτο Σφάλμα, Πηγή:

<https://www.statisticshowto.com/absolute-error/>

[20] Μέσο Λογαριθμικό Τετραγωνικό Σφάλμα, Πηγή:

<https://techwithshadab.medium.com/root-mean-squared-error-root-mean-squared-logarithmic-error-70e645efcccb>

[21] Τι είναι το Χρηματιστήριο, Πηγή:

<https://vontikakis.com/investing/xrimatistirio>

[22] Η ιστορία των χρηματιστηρίων, Πηγή:

<https://www.bankingnews.gr/bankingnews-questions-and-answers1/%CF%83%CF%84%CE%BF%CE%BD-%CF%80%CE%B1%CE%BB%CE%BC%CF%8C-%CF%84%CE%B7%CF%82->

[23] Τι είναι οι μετοχές, Πηγή:

<https://euretirio.com/metoxes-stocks/>

[24] Μηχανική Μάθηση για χρηματιστηριακές προβλέψεις, Πηγή:

<https://www.analyticsvidhya.com/blog/2021/10/machine-learning-for-stock-market-prediction-with-step-by-step-implementation/>

Πηγές Εικονών

Εικόνα 1.1, Πηγή : <https://www.g2.com/articles/types-of-artificial-intelligence>

Εικόνα 1.2, Πηγή : <https://www.mathworks.com/solutions/machine-learning.html>

Εικόνα 1.3, Πηγή : <https://www.ml-science.com/mean-squared-error>,

Εικόνα 1.4, Πηγή : <https://levity.ai/blog/difference-machine-learning-deep-learning>

Εικόνα 2.1, Πηγή : <https://www.javatpoint.com/linear-regression-in-machine-learning>

Εικόνα 2.2, Πηγή : <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>

Εικόνα 3.1, Πηγή : <https://www.ibm.com/uk-en/cloud/learn/neural-networks>

Εικόνα 3.2, Πηγή : <https://machinelearningmastery.com/adam-optimization-from-scratch/>

Εικόνα 3.3, Πηγή : <https://www.kdnuggets.com/2020/12/optimization-algorithms-neural-networks.html>

Εικόνα 3.4, Πηγή : <https://www.kdnuggets.com/2020/12/optimization-algorithms-neural-networks.html>

Εικόνα 3.5, Πηγή : <https://www.guru99.com/backpropagation-neural-network.html>

Εικόνα 3.6, Πηγή : <https://www.pluralsight.com/guides/introduction-to-lstm-units-in-rnn>

Κατάλογος Πινάκων

Πίνακας 5.1 : Παράδειγμα αρχείου CSV

Πίνακας 5.2 : Μέσο τετραγωνικό σφάλμα χρησιμοποιώντας διαφορετικούς αλγορίθμους βελτιστοποίησης και συναρτήσεων ενεργοποίησης στο μοντέλο LSTM.

Πίνακας 5.3 : Δοκιμή διαφορετικών Kernels πάνω στο μοντέλο του SVM.

Πίνακας 5.4 : Δοκιμή του LSTM με διαφορετικά ποσά δεδομένων και αποτελέσματα των εποχών.

Πίνακας 5.5 : Τελικά αποτελέσματα όλων των μοντέλων