ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΜΣ «Προηγμένα Συστήματα Πληροφορικής – Ανάπτυξη Λογισμικού και Τεχνητής Νοημοσύνης»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Πρόβλεψη Αστάθειας σε Ανταλλακτήρια με Μηχανική Μάθηση** **Financial Markets Volatility Prediction using Machine Learning**
Ονοματεπώνυμο Φοιτητή	**Γεώργιος Πάνου**
Πατρώνυμο	**Ιωάννης**
Αριθμός Μητρώου	**ΜΠΣΠ18019**
Επιβλέπων	**Θεμιστοκλής Παναγιωτόπουλος, Καθηγητής**

Ημερομηνία Παράδοσης: **Νοέμβριος 2022**

Τριμελής Εξεταστική Επιτροπή

Θ. Παναγιωτόπουλος	**Δ. Αποστόλου**	**Ά. Πικράκης**
Καθηγητής	**Καθηγητής**	**Επίκουρος Καθηγητής**

Περίληψη

Η παρούσα εργασία έχει σαν αντικείμενο την μελέτη της συμπεριφοράς των εφαρμογών μηχανικής μάθησης στο περιβάλλον των ανταλλακτηρίων χρηματιστηριακών προϊόντων.

Για την εκπόνηση της εργασίας έγινε ανάπτυξη αλλά και εκτέλεση αλγορίθμων για την πρόγνωση αστάθειας χρηματιστηριακών προϊόντων. Η ανάπτυξη πραγματοποιήθηκε στα πλαίσια ενός διαγωνισμού του Kaggle, μιας διαδικτυακή κοινότητας επιστημόνων δεδομένων και επαγγελματιών μηχανικής μάθησης.

Πραγματοποιήθηκε μια σειρά πειραμάτων και έγινε σύγκριση των αποτελεσμάτων μεταξύ των διαφορετικών τεχνικών. Επίσης διερευνήθηκε η αύξηση της απόδοσης με χρήση παραλληλισμού στον επεξεργαστή αλλά και με χρήση κάρτας γραφικών.

Ο διοργανωτής του διαγωνισμού παρείχε τα δεδομένα για την εκπαίδευση των μοντέλων που αναπτύχθηκαν καθώς και ένα μηχανισμό αξιολόγησης των αλγορίθμων.

Abstract

The purpose of this project is to study the behavior of machine learning applications in the environment of stock exchanges.

In this regard, algorithms were developed and executed to forecast the volatility of stock exchange products. The implementation was developed n the context of a competition which was organized by Kaggle, an online community of data science and machine learning professionals.

A series of experiments were performed using different techniques of implementation and their results were compared. Furthermore the performance gains of parallelization of CPU tasks as well as using a GPU were investigated.

The organizer of the competition provided the data to be used both for the developed models' training and as a mechanism for algorithms' evaluation.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

ΠΙΝΑΚΑΣ ΠΙΝΑΚΩΝ

ΠΙΝΑΚΑΣ ΕΞΙΣΩΣΕΩΝ

Κεφάλαιο 1ο

1 Εισαγωγή

Η εργασία επιχειρεί να διερευνήσει την εφαρμογή διάφορων τεχνικών τεχνητής μάθησης στον τομέα των χρηματιστηριακών προϊόντων. Πιο συγκεκριμένα επιχειρείται μια πρόγνωση της αστάθειας μιας χρηματιστηριακής αγοράς ανά μετοχή και σε βάθος χρόνου 10 λεπτών. Τα δεδομένα και το πρόβλημα προς επίλυση έχουν οριστεί από την εταιρεία Optiver στα πλαίσια ενός διαγωνισμού του Kaggle – μιας διαδικτυακής κοινότητας της Goolge. Οι πληροφορίες των μετοχών και των χρονικών στιγμών έχουν αποκρυφτεί από τους συμμετέχοντες στον διαγωνισμό (data anonymization).

1.1 Περιγραφή του υπό μελέτη προβλήματος

Το πρόβλημα που επιχειρείται να επιλύσει η παρούσα εργασία είναι η πρόγνωση της Αστάθειας (Volatility) των τιμών των προϊόντων ανταλλακτηρίων. Δεδομένου ενός χρονικού διαστήματος στο οποίο είναι γνωστά μια σειρά από στοιχεία της αγοράς οι αλγόριθμοι επιχειρούν να προβλέψουν κατά πόσο θα είναι ασταθής ή όχι η αγορά στο επόμενο χρονικό διάστημα.

1.2 Σκοπός και στόχοι της εργασίας

Οι στόχοι της εργασίας περιλαμβάνουν τα ακόλουθα:

- Την μελέτη των αλγορίθμων μηχανικής μάθησης ως εργαλεία πρόγνωσης της συμπεριφοράς των αγορών
- Η υλοποίηση διαφορετικών αλγορίθμων και η συγκριτική μελέτη των αποτελεσμάτων τους
- Τη συμμετοχή στον διαγωνισμό της κοινότητας με σκοπό την διεύρυνση των οριζόντων μέσω της ανταλλαγής πληροφορίας αλλά και της εμπειρίας τους ανταγωνισμού και του συναγωνισμού
- Την αξιολόγηση των διαφορετικών μεθόδων προσέγγισης του προβλήματος με όσο το δυνατό πιο αντικειμενικά κριτήρια όπως αυτή προκύπτει από τους μηχανισμούς αξιολόγησης τους διαγωνισμού

1.3 Οικονομικές έννοιες και δεδομένα

1.3.1 Βιβλίο Εντολών (Order book)

Ο όρος Order book (βιβλίο εντολών) αναφέρεται σε μια ηλεκτρονική λίστα εντολών αγοράς και πώλησης για ένα συγκεκριμένο τίτλο ή χρηματοπιστωτικό προϊόν και διαμορφώνεται σε επίπεδα τιμών.

Ουσιαστικά περιέχει όλες τις εντολές αγοράς/πώλησης που οι συμμετέχοντες στην αγορά έχουν δημιουργήσει. Ένα βιβλίο εντολών παραθέτει τον αριθμό των μετοχών που προσφέρονται ή ζητούνται σε κάθε επίπεδο τιμών. Τα επίπεδα τιμών είναι ταξινομημένα και το περιεχόμενό του είναι ιδιαίτερα ρευστό στο πέρασμα του χρόνου.

bid #	price	ask #
	151	196
	150	189
	149	148
	148	221
251	147	
321	146	
300	145	
20	144	

Πίνακας 1. Στιγμιότυπο ενός Order book (βιβλίο εντολών)

Στην παραπάνω εικόνα παρουσιάζεται ένα στιγμιότυπο ενός Order book που αφορά σε μία μετοχή. Όπως φαίνεται, όλες οι προβλεπόμενες εντολές αγοράς βρίσκονται στην αριστερή πλευρά του βιβλίου και εμφανίζονται ως "bid", ενώ όλες οι προβλεπόμενες εντολές πώλησης βρίσκονται στα δεξιά η πλευρά του βιβλίου εμφανίζεται ως "ask".

Ένα χρηματοοικονομικός οργανισμός που λειτουργεί ενεργά έχει πάντα ένα πυκνό ή ρευστό βιβλίο εντολών (liquid book). Καθώς τα δεδομένα του βιβλίου παραγγελιών είναι μια συνεχής αναπαράσταση της ζήτησης/προσφοράς της αγοράς, θεωρείται πάντα ως η νούμερο ένα πηγή δεδομένων για έρευνα αγοράς.

Μια άλλη μέρα, το βιβλίο παραγγελιών του προιόντος Α φαίνεται στην παρακάτω εικόνα. Όπως μπορείτε να δείτε το βιβλίο δεν είναι τόσο πυκνό όσο το προηγούμενο, και μπορεί κανείς να πει, σε σύγκριση με το προηγούμενο, ότι αυτό το βιβλίο είναι λιγότερο ρευστό. Οι συναλλαγές συνήθως θα είναι πιο ακριβές και εάν θέλει κανείς αξιόπιστη εκτέλεση των συναλλαγών του, θα αντιμετωπίσει υψηλότερο κίνδυνο, για αυτό το λόγο οι επενδυτές προτιμούν τη ρευστότητα. Η τελική επίπτωση που έχει στην αγορά είναι ότι επειδή μειώνει την πιθανότητα οι επενδυτές να μπορούν να ικανοποιήσουν τις ανάγκες τους, ο συνολικός όγκος των συναλλαγών αναμένεται σε τέτοιες περιπτώσεις να είναι μειωμένος.

bid #	price	ask #
	151	20
	150	12
	149	1
	148	
5	147	
2	146	
	145	
16	144	

Πίνακας 2. Στιγμιότυπο ενός αραιού Order book

1.3.2 Βιβλίο Παραγγελιών (Trade book)

Ενώ ένα βιβλίο εντολών είναι μια αναπαράσταση της πρόθεσης συναλλαγών στην αγορά, για να πραγματοποιηθεί μια συναλλαγή χρειάζεται ένας αγοραστής και έναν πωλητής να βρεθούν στην θέση να πραγματοποιήσουν μια συναλλαγή στην ίδια τιμή.

Ως εκ τούτου, μερικές φορές, όταν κάποιος θέλει να κάνει μια συναλλαγή σε μια μετοχή, ελέγχει το βιβλίο εντολών και βρίσκει κάποιον με αντίθετο ενδιαφέρον για συναλλαγή.

Για παράδειγμα, αν φανταστούμε ότι θέλουμε να αγοράσουμε 20 μετοχές μιας μετοχής που έχει το βιβλίο παραγγελιών της προηγούμενης παραγράφου, θα πρέπει να βρεθούν κάποια άνθρωποι που να είναι πρόθυμοι να κάνουν συναλλαγές με εμάς πουλώντας συνολικά 20 μετοχές ή περισσότερες. Ελέγχοντας την πλευρά της προσφοράς του βιβλίου ξεκινώντας από τη χαμηλότερη τιμή: υπάρχουν 221 μετοχές με ενδιαφέρον πώλησης στην τιμή των 148. Μπορούμε να αγοράσουμε 20 μετοχές στην τιμή των 148 και η εκτέλεση της εντολής θα είναι εγγυημένη. Αυτό θα ήταν το βιβλίο εντολών του αποθέματος Α μετά τη συναλλαγή:

bid #	price	ask #
	151	196
	150	189
	149	148
	148	201
251	147	
321	146	
300	145	
20	144	

Πίνακας 3. Πραγματοποίηση μιας εντολής αγοράς

Σε αυτήν την περίπτωση, ο πωλητής πούλησε 20 μετοχές και ο αγοραστής αγόρασε 20 μετοχές. Η συναλλαγή μεταξύ πωλητή και αγοραστή ολοκληρώνεται και ένα εμπορικό μήνυμα θα μεταδοθεί στο κοινό:

"20 μετοχές της μετοχής Α ανταλλάσσονται στην αγορά στην τιμή των 148."

Παρόμοια με τα δεδομένα του βιβλίου εντολών, τα δεδομένα συναλλαγών είναι επίσης εξαιρετικά σημαντικά για τους επιστήμονες δεδομένων, καθώς αντικατοπτρίζουν πόσο ενεργή είναι η αγορά. Στην πραγματικότητα, ορισμένα κοινά τεχνικά σήματα της χρηματοπιστωτικής αγοράς προέρχονται απευθείας από δεδομένα συναλλαγών, όπως ο υψηλός χαμηλός ή ο συνολικός όγκος συναλλαγών.

1.4 Στατιστικοί δείκτες

Υπάρχουν πολλοί στατιστικοί/οικονομικοί δείκτες που μπορεί να αντλήσει ένας επιστήμονας δεδομένων από τα ακατέργαστα δεδομένα του βιβλίου εντολών ώστε να επιδιώξει να αναπαραστήσει τη ρευστότητα της αγοράς και την αποτίμηση των μετοχών. Αυτά τα στατιστικά στοιχεία είναι αποδεδειγμένα θεμελιώδεις εισροές οποιωνδήποτε αλγορίθμων χρησιμοποιούνται για προβλέψεις της αγοράς. Χρησιμοποιούνται εδώ και χρόνια από στατιστικολόγους, οικονομολόγους και αναλυτές των αγορών για εκτίμηση των αξιών αλλά και προβλέψεις.

Παρακάτω παρατίθενται ορισμένοι κοινά αποδεκτοί στατιστικοί δείκτες που χρησιμοποιήθηκαν στα πλαίσια της εργασίας ώστε να εξορυχτούν πολύτιμα σήματα από τα δεδομένα του βιβλίου εντολών. Κάποιοι χρησιμοποιούνται και σαν μέτρο σύγκρισης των αποτελεσμάτων.

Συνοπτικά οι στατιστικοί και οικονομικοί δείκτες που χρησιμοποιήθηκαν για την ανάλυση των δεδομένων της εργασίας αλλά ταυτόχρονα και από τους αλγόριθμους μηχανικής μάθησης που αναπτύχθηκαν είναι:

- Bid/Ask Spread
- Weighted Average Price
- Log Returns
- Realized volatility

1.4.1 Bid/Ask Spread

Καθώς οι μετοχές διαπραγματεύονται συνεχώς σε διαφορετικά επίπεδα στην αγορά, λαμβάνουμε την αναλογία της καλύτερης τιμής προσφοράς και της καλύτερης τιμής ζήτησης για να υπολογίσουμε το spread προσφοράς-ζήτησης.

Ο μαθηματικός τύπος της διαφοράς προσφοράς/ζήτησης μπορεί να γραφτεί στην παρακάτω μορφή:

$$BidAskSpread = BestOffer/BestBid - 1$$

Εξίσωση 1. Bid/Ask Spread

1.4.2 Weighted Averaged Price (WAP)

Το βιβλίο παραγγελιών είναι πέρα από πηγή δεδομένων για ανάλυση της αγοράς μια από τις πιο αξιόπιστες πηγές για την αποτίμηση των μετοχών. Μια δίκαιη λογιστική αποτίμηση πρέπει να λαμβάνει υπόψη δύο παράγοντες: το επίπεδο και το μέγεθος των παραγγελιών. Σε αυτόν τον διαγωνισμό χρησιμοποιήσαμε τη Weighted Averaged Price (σταθμισμένη μέση τιμή), ή WAP, για να υπολογίσουμε τη στιγμιαία αποτίμηση των μετοχών.

Ο τύπος του WAP μπορεί να γραφτεί ως εξής, (λαμβάνει υπόψη τις πληροφορίες τιμής και όγκου ανώτατου επιπέδου):

$$WAP = \frac{BidPrice_1 * AskSize_1 + AskPrice_1 * BidSize_1}{BidSize_1 + AskSize_1}$$

Εξίσωση 2. WAP (Weighted Average Price)

Όπως φαίνεται, εάν δύο βιβλία έχουν προσφορά και ζήτηση στο ίδιο επίπεδο τιμής αντίστοιχα, αυτό με την μεγαλύτερη προσφορά θα δημιουργήσει χαμηλότερη αποτίμηση των μετοχών, καθώς υπάρχουν περισσότεροι πωλητές στο βιβλίο και περισσότεροι πωλητές συνεπάγονται μεγαλύτερη προσφορά στην αγορά με αποτέλεσμα χαμηλότερη αποτίμηση των προιόντων.

Αξίζει να σημειωθεί ότι στις περισσότερες περιπτώσεις, κατά τη διάρκεια συνεχών ωρών συνεδριών, ένα βιβλίο εντολών δεν θα πρέπει να έχει σε καμία περίπτωση εντολές αγοράς σε υψηλότερες τιμές από των εντολών πώλησης.

1.4.3 Log Returns (Λογαριθμικά Κέρδη)

Τα κέρδη (returns) των μετοχών μπορούν να υπολογιστούν με διάφορους τρόπους, εδώ θα εξετάσουμε έναν καθολικό τρόπο αποτίμησης των κερδών που διευκολύνει και υπολογιστικές διαδικασίες.

Αν κάνουμε μια απόπειρα να συγκρίνουμε την τιμή μιας μετοχής μεταξύ χθες και σήμερα ένας απλοϊκός τρόπος θα ήταν να παίρναμε την καθαρή διαφορά μεταξύ των τιμών στο κλείσιμο. Όμως αυτός δεν θα ήταν ένας αποδοτικός τρόπος να υπολογίζουμε το κέρδος που μας απέφεραν διαφορετικές μετοχές. Αυτό συμβαίνει διότι ενώ η τιμή μιας μετοχής σε σχέση με μια άλλη μπορεί να είναι πολύ διαφορετική αυτό που μας ενδιαφέρει περισσότερο είναι η ποσοστιαία μεταβολή.

Ο λόγος λοιπόν της διαφοράς των τιμών ανοίγματος και κλεισίματος προς την τιμή ανοίγματος είναι η απόδοση της κάθε μετοχής μεμονωμένα. Η απόδοση της μετοχής συμπίπτει με την ποσοστιαία μεταβολή του επενδυμένου κεφαλαίου ενός επενδυτή.

Ενώ η απόδοση αυτή χρησιμοποιούνται ευρέως στα χρηματοοικονομικά, ωστόσο προτιμάται, όποτε απαιτείται κάποια μαθηματική μοντελοποίηση να μετατρέπουμε αυτό το λόγο σε λογάριθμο. Οι λογαριθμικές αποδόσεις (log returns) παρουσιάζουν πολλά πλεονεκτήματα, όπως για παράδειγμα:

- είναι αθροιστικές στο χρόνο

- Οι μή-λογαριθμικές αποδόσεις δεν μπορούν να πέφτουν κάτω από -100%, ενώ οι λογαριθμικές δεν περιορίζονται

Έτσι λοιπόν ορίζουμε τη λογαριθμική απόδοση ως την τιμή r, μεταξύ των χρονικών στιγμών t_1

$$r_{t_1,t_2} = \log\left(\frac{S_{t_2}}{S_{t_1}}\right)$$

και t_2:

Εξίσωση 3. Log Return (Λογαριθμική απόδοση)

Όπου το S_t αντιπροσωπεύει την τιμή μιας μετοχής S την χρονική στιγμή t

1.4.4 Volatility (Αστάθεια)

Στις χρηματοπιστωτικές αγορές, η αστάθεια αποτυπώνει το μέγεθος της διακύμανσης των τιμών. Η υψηλή αστάθεια σχετίζεται με περιόδους αναταραχής της αγοράς και μεγάλες διακυμάνσεις των τιμών, ενώ η χαμηλή αστάθεια περιγράφει πιο ήρεμες και ήσυχες αγορές. Για τις χρηματιστηριακές εταιρείες η ακριβής πρόβλεψη της αστάθειας είναι απαραίτητη για τη διαπραγμάτευση συναλλαγών, των οποίων η τιμή σχετίζεται άμεσα με την αστάθεια του υποκείμενου προϊόντος.

Ένα πολύτιμο στοιχείο για τα μοντέλα μας είναι η τυπική απόκλιση των log returns των μετοχών. Η τυπική απόκλιση θα είναι διαφορετική για τις αποδόσεις καταγραφής που υπολογίζονται σε μεγαλύτερα ή μικρότερα διαστήματα, για το λόγο αυτό συνήθως κανονικοποιείται σε περίοδο 1 έτους και η ετήσια τυπική απόκλιση ονομάζεται αστάθεια.

Το μέγεθος αυτό αποτελεί και τον στόχο και την μονάδα αξιολόγησης των αλγορίθμων μηχανικής μάθησης της εργασίας. Σε αυτή τον διαγωνισμό, μας δόθηκαν δεκάλεπτα δεδομένων των βιβλίων εντολών και συναλλαγών και επιχειρείται να προβλέψουμε ποια θα είναι η τιμή της στα επόμενα 10 λεπτά. Για τα δεδομένα του προβλήματος η αστάθεια την ορίζεται ως εξής:

Υπολογίζουμε τα log returns για όλες τις διαδοχικές χρονοθυρίδες των δύο βιβλίων όπως παραπάνω και ορίζουμε την τετραγωνική ρίζα του αθροίσματος των τετραγώνων των log returns:

$$\sigma = \sqrt{\sum_t r_{t-1,t}^2}$$

Εξίσωση 4. Realized log returns

r: Log returns
t: χρονοθυρίδα

1.5 Παραδοτέα της εργασίας

Η εργασία περιλαμβάνει τόσο το παρόν κείμενο όσο και τον κώδικα αλλά και ένα υπολογιστικό φύλλο που συγκεντρώνει τα δεδομένα των δοκιμών:

1. Το έντυπο κείμενο της πτυχιακής εργασίας, το οποίο περιλαμβάνει την επισκόπηση της σχετικής με το χώρο βιβλιογραφίας, την μελέτη του προβλήματος, την περιγραφή της μεθοδολογίας που ακολουθήθηκε για την ανάπτυξη και την παρουσίαση των αποτελεσμάτων των δοκιμών.
2. Λογισμικό σε μορφή Jupyter notebook σε γλώσσα Python.
3. Ένα υπολογιστικό φύλλο που συγκεντρώνει τα δεδομένα των δοκιμών

1.6 Δομή της εργασίας

Στο κεφάλαιο 2 γίνεται μια επισκόπηση του χώρου της μηχανικής μάθησης με βάση την βιβλιογραφία. Καταρχήν δίνεται η κατηγοριοποίηση των διαφορετικών πεδίων του χώρου όπως και το ευρύτερο περιβάλλον του Data Science που ανήκει. Το κεφάλαιο ξεκινά από τα πιο γενικά και εξειδικεύει εστιάζοντας σε συναφή πεδία με αυτά της εφαρμογής που αναπτύχθηκε. Στο τέλος του κεφαλαίου περιγράφονται τεχνικές ανάπτυξης πολλές από τις οποίες εφαρμόστηκαν.

Στο κεφάλαιο 3 αρχικά δίνεται ο ορισμός του προβλήματος έπειτα δίνεται μια περιγραφή της υλοποίησης και τέλος παρουσιάζονται οι δοκιμές και τα αποτελέσματά τους.

Στο κεφάλαιο 4 παρουσιάζονται τα συμπεράσματα της εργασίας όπου σχολιάζονται τα αποτελέσματα και παρουσιάζεται η εμπειρία εφαρμογής των τεχνικών που ακολουθήθηκαν για την ανάπτυξη της εφαρμογής.

Κεφάλαιο 2ο

2 Επισκόπηση του χώρου

Ο χώρος στον οποίο εμπίπτει η παρούσα εργασία είναι ο ευρύτερος τομέας των λογισμικών χρηματοοικονομικής ανάλυσης (Fintech: Financial Technology), δηλαδή μπορεί να μελετηθεί τόσο από τη σκοπιά της Οικονομικής επιστήμης τόσο και από την σκοπιά της Πληροφορικής. Όμως στην παρούσα εργασία εστιάζουμε στο πως μπορούν να εφαρμοστούν μέθοδοι τεχνητής νοημοσύνης σε τέτοιου είδους λογισμικά με τη χρήση μόνο των απαραίτητων όρων που χρειάζονται για την κατανόηση του δοσμένου προβλήματος.

Τα επιστημονικά πεδία στα οποία εστιάζουμε στην παρούσα εργασία είναι η Ανάλυση Δεδομένων (Data Science) και πιο ειδικά η Μηχανική Μάθηση (Machine Learning).

Η Μηχανική Μάθηση μας επιτρέπει να επεξεργαστούμε δεδομένα με έναν τρόπο διαφορετικό από αυτό των “παραδοσιακών” αλγορίθμων. Το αποτέλεσμα που παρέχει ένας τέτοιος αλγόριθμος σε αντίθεση με τους “παραδοσιακούς” έχει μια πιθανότητα να αληθεύει ή και να σφάλει. Παρόλα αυτά οι σύγχρονες εφαρμογές του πεδίου έχουν πολύ ικανοποιητικά αποτελέσματα – σε τέτοιο βαθμό που η πλειοψηφία των κατασκευαστών αυτοκινήτων εξοπλίζουν τα οχήματα τους με συστήματα ασφαλείας αυτόματης ακινητοποίησης σε έκτακτη ανάγκη, συστήματα αποφυγής εκούσιας αλλαγής λωρίδας κ.α. .

Με την ανάπτυξη του διαδικτύου και την παραγωγή και συλλογή τεράστιου όγκου δεδομένων προέκυψε η ανάγκη και η επιθυμία αξιοποίησής τους. Οι διαρκείς βελτιώσεις στο υλισμικό (hardware) – αν και ο ρυθμός ανάπτυξης δείχνει σημάδια κάμψης τα τελευταία χρόνια υπάρχει σημαντική πρόοδος – και συγκεκριμένα σε μαζικά παράλληλους επεξεργαστές (π.χ. Cuda) κατέστησαν δυνατή την ανάπτυξη αλγορίθμων μηχανικής μάθησης. Έτσι αν και στο παρελθόν βρίσκονταν στο περιθώριο ο συνδυασμός των παραπάνω παραγόντων επέτρεψαν στους αλγόριθμους μηχανικής μάθησης να δώσουν λύσεις σε πολύ σύνθετα προβλήματα και με πολύ μικρό κόστος. Αν και θεωρούνται τεχνολογία αιχμής το εξαιρετικά χαμηλό κόστος - αυτό που αφορά την επεξεργασία των δεδομένων - που χρειάζεται για την ανάπτυξή τους σε σύγκριση με το απαγορευτικό κόστος των άλλων μεθόδων είναι που επιτρέπει να παραβλέπεται η ανακρίβεια στα αποτελέσματα που παράγουν. Ένα συμπέρασμα που μπορεί να εξαχθεί είναι ότι με αυτό τον τρόπο σπάει το φράγμα της διαθέσιμης επεξεργαστικής ισχύς για αυτούς που έχουν πρόσβαση σε μεγάλο όγκο δεδομένων. [10]

Η εξέλιξη μέσω των καινοτομιών στις ΤΠΕ, καθώς και στις τεχνολογίες Διαδικτύου και λογισμικού έχουν διεισδύσει σε κάθε τεχνολογικό δημιούργημα και προϊόν στην καθημερινή μας ζωή. Τα παραδοσιακά προϊόντα γίνονται όλο και πιο πολυλειτουργικά, έξυπνα, δικτυωμένα, ευέλικτα και περιλαμβάνουν υπηρεσίες που σχετίζονται με αυτά. Ωστόσο, δεν είναι μόνο τα καταναλωτικά αγαθά που γίνονται “έξυπνα” αλλά ταυτόχρονα και τα βιομηχανικά μηχανήματα. Η διείσδυση της τεχνητής νοημοσύνης στους περισσότερους βιομηχανικούς τομείς στο άμεσο μέλλον θα μας οδηγήσει στην 4η Βιομηχανική Επανάσταση.

Έτσι, η απόκτηση τέτοιων εργαλείων θα είναι ζωτικής σημασίας για την ανταγωνιστικότητα των βιομηχανικών επιχειρήσεων.

Η Τέταρτη Βιομηχανική Επανάσταση (4IR ή Industry 4.0) είναι η συνεχής αυτοματοποίηση των παραδοσιακών σχεδιαστικών, κατασκευαστικών και βιομηχανικών πρακτικών, με χρήση σύγχρονης έξυπνης τεχνολογίας. Επιπλέον και στον τομέα του μάρκετινγκ η έξυπνη αξιοποίηση μεγάλου όγκου δεδομένων (ακόμα και προσωποποιημένων) καταναλωτικών προτιμήσεων μπορεί να δώσει το πάνω χέρι σε όσους αξιοποιήσουν την τεχνολογία. Η επικοινωνία μεγάλης κλίμακας από μηχανή με μηχανή (M2M) και το Διαδίκτυο των πραγμάτων (IoT) θα είναι διασυνδεδεμένα για πιο προηγμένη αυτοματοποίηση, βελτιωμένη επικοινωνία και αυτο-παρακολούθηση. Η παραγωγή έξυπνων μηχανών που μπορούν να αναλύουν και να διαγνώσουν προβλήματα χωρίς την ανάγκη ανθρώπινης παρέμβασης θα διαδραματίσει σημαντικό ρόλο. [8]

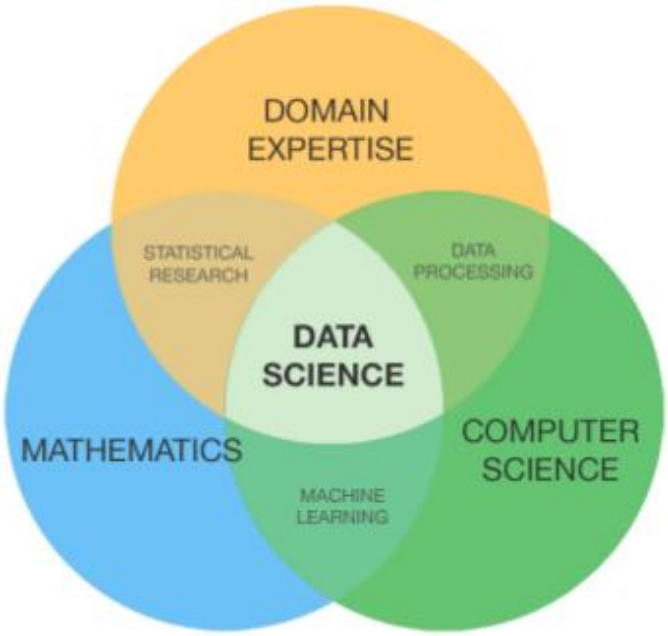
Στην εποχή της Τέταρτης Βιομηχανικής Επανάστασης, που βιώνουμε το ξεκίνημά της, ο ψηφιακός κόσμος διαθέτει πληθώρα δεδομένων, όπως δεδομένα διαδικτύου, μικροσυσκευών (IoT), δεδομένα κυβερνο-ασφάλειας, δεδομένα κινητής τηλεφωνίας, δεδομένα επιχειρήσεων, δεδομένα κοινωνικών μέσων, δεδομένα υγείας , κ.λ.π. Για την έξυπνη ανάλυση αυτών των δεδομένων και την ανάπτυξη των αντίστοιχων έξυπνων και αυτοματοποιημένων εφαρμογών, η χρήση της τεχνητής νοημοσύνης (AI), ιδιαίτερα της μηχανικής μάθησης (ML) έχει αποκτήσει πολύ σημαντική θέση.

Παρακάτω παρατίθεται μια πιο εκτενής περιγραφή των τομέων της επιστήμης που εμπίπτει η εργασία.

2.1 Αλγόριθμοι και Ανάλυση Δεδομένων – Data Science

Η επιστήμη των δεδομένων (Data Science) είναι ένας διεπιστημονικός τομέας που χρησιμοποιεί επιστημονικές μεθόδους, διαδικασίες, αλγόριθμους και συστήματα για την εξαγωγή γνώσεων και γνώσεων από θορυβώδη, δομημένα και αδόμητα δεδομένα, και εφαρμογή γνώσεων και πρακτικών γνώσεων από δεδομένα σε ένα ευρύ φάσμα τομέων εφαρμογών. Αυτό σημαίνει ότι με την επιστήμη δεδομένων, οι οργανισμοί μπορούν να χρησιμοποιήσουν δεδομένα για να καταλάβουν τι συνέβη, γιατί συνέβη, τι θα συμβεί και τι πρέπει να κάνουν για να πάρουν ένα επιθυμητό αποτέλεσμα.

Καθώς τα δεδομένα που έχουν στην διάθεσή τους οι εταιρείες αυξάνονται με εκθετικό ρυθμό, η αξιοποίηση τους γίνεται ολοένα και δυσκολότερη. Οι περισσότεροι οργανισμοί αντιμετωπίζουν μια αδυναμία να αναλύσουν τα δεδομένα τους με τις κλασσικές μεθόδους που θα χρησιμοποιούσαν οι εμπειρογνώμονες ακριβώς λόγω του τεράστιου όγκου. Πολλές επιχειρήσεις προμηθεύουν τώρα τους εργαζόμενους τους με πλατφόρμες τεχνητής νοημοσύνης που μπορούν να τους βοηθήσουν να διεξάγουν δικές τους διαδικασίες μηχανικής μάθησης.

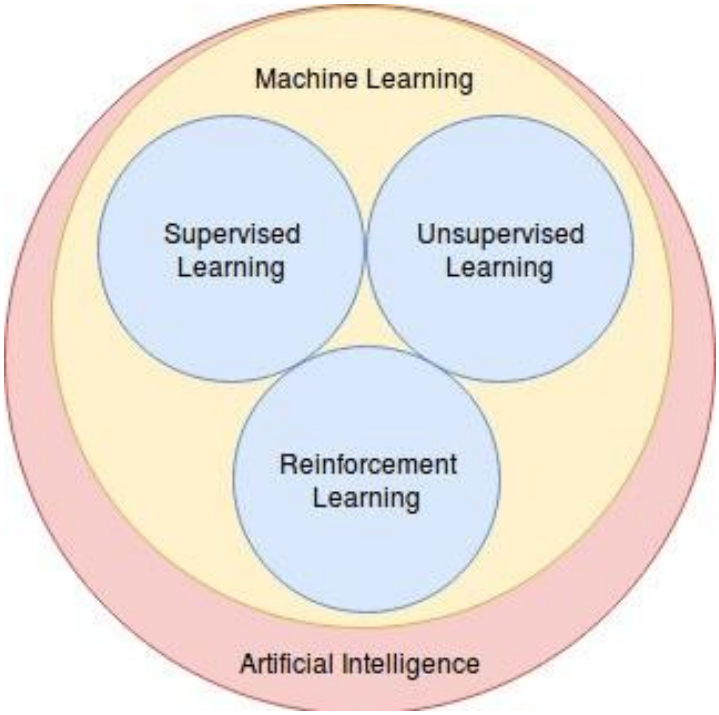


Εικόνα 1. Διάγραμμα Venn – Data Science

2.2 Μηχανική Μάθηση - Machine Learning

Η Μηχανική μάθηση είναι η επιστήμη (και η τέχνη) του να προγραμματίζεις υπολογιστές ώστε να μπορούν να εκπαιδεύονται από τα δεδομένα. Οι αλγόριθμοι αυτοί βελτιώνονται αυτόματα και θεωρούνται μέρος του πεδίου της τεχνητής νοημοσύνης.[1]

Οι αλγόριθμοι μηχανικής μάθησης έχουν υιοθετηθεί ευρέως σε μεγάλη ποικιλία εφαρμογών που επηρεάζουν σημαντικά τις ζωές των ανθρώπων, όπως στην ιατρική, το φιλτράρισμα ηλεκτρονικού ταχυδρομείου, την αναγνώριση ομιλίας και την τεχνητή αίσθηση, όπου είναι δύσκολο ή ανέφικτο να αναπτυχθούν συμβατικοί αλγόριθμοι για την εκτέλεση των απαιτούμενων εργασιών. [1] Η μηχανική μάθηση έχει παρουσιάσει τα τελευταία χρόνια μεγάλη άνθηση, καθώς ολοένα και περισσότερες εταιρείες αναγνωρίζουν την αξία της και επενδύουν σε αυτή την τεχνολογία. Ο μεγάλος όγκος διαθέσιμων δεδομένων (ο οποίος μάλιστα αυξάνεται καθημερινά) καθιστά τους αλγορίθμους αυτούς ιδιαίτερα ελκυστικούς ως προς το κόστος υλοποίησης τους και το αποτέλεσμά τους.



Εικόνα 2. Διάγραμμα Venn – Machine Learning

Στο πεδίο της μηχανικής μάθησης υπάρχουν διάφοροι τύποι αλγορίθμων μηχανικής μάθησης, όπως supervised (εποπτευόμενη), unsupervised (χωρίς επίβλεψη), semi-supervised (ημι-εποπτευόμενη), και reinforcement learning (ενισχυτική μάθηση) Επιπλέον, το deep learning (βαθιά εκμάθηση), η οποία αποτελεί μέρος μιας ευρύτερης οικογένειας μεθόδων μηχανικής μάθησης, μπορεί να αναλύσει δεδομένα μεγάλης κλίμακας.

Συνοπτικά το πεδίο της μηχανικής μάθησης περιλαμβάνεται στον τομέα της τεχνητής νοημοσύνης. Ομοίως, διάφοροι αλγόριθμοι μάθησης εμπίπτουν στην Μηχανική Μάθηση. Το παραπάνω διάγραμμα μας βοηθάει να κατανοήσουμε τους διάφορους αλγόριθμους που εμπίπτουν στη κατηγορίας της μηχανική μάθηση, συνοπτικά:

- Supervised learning: Όπως υποδηλώνει το όνομα, είναι μια τεχνική μάθησης όπου η διαδικασία εποπτεύεται. Ο κύριος στόχος αυτών των αλγορίθμων εκμάθησης είναι να προβλέψουν το αποτέλεσμα, δεδομένου ενός συνόλου δειγμάτων εκπαίδευσης μαζί με τις ετικέτες εκπαίδευσης (labels). Δηλαδή είναι γνωστές τόσο οι ανεξάρτητες μεταβλητές όσο και η λύση. Με αυτή τη μέθοδο οι αλγόριθμοι μαθαίνουν “εμπειρικά” δηλαδή από προσπαθούν να ρυθμίσουν τις εσωτερικές τους παραμέτρους ώστε τα αίτια (ανεξάρτητες μεταβλητές) να οδηγούν στο αποτέλεσμα.

- Unsupervised Learning: Σε αντίθεση με την εποπτευόμενη μάθηση, δεν υπάρχουν ετικέτες εκπαίδευσης για τα δείγματα εκπαίδευσης. Οι αλγόριθμοι είναι κατασκευασμένοι με τέτοιο τρόπο ώστε να μπορούν να βρουν υπάρχουσες δομές και μοτίβα στα δεδομένα. Μόλις γίνουν εμφανή αυτά τα συνεπή μοτίβα, τα παρόμοια σημεία

δεδομένων μπορούν να ομαδοποιηθούν μαζί και διαφορετικά σημεία δεδομένων θα βρίσκονται σε διαφορετικά συμπλέγματα. Χρησιμοποιείται κυρίως σε προβλήματα κατάταξης (classification) όπως και για την προβολή δεδομένων υψηλών διαστάσεων σε χαμηλής διάστασης για σκοπούς απεικόνισης ή ανάλυσης.

- Reinforcement Learning: Είναι ένας τύπος μηχανικής μάθησης που έχει έναν πράκτορα (όπως ένα ρομπότ) που μαθαίνει πώς να συμπεριφέρεται σε ένα περιβάλλον κάνοντας ενέργειες και ποσοτικοποιώντας τα αποτελέσματα. Εάν ο πράκτορας απαντήσει σωστά, λαμβάνει έναν πόντο ανταμοιβής, ο οποίος ενισχύει την αυτοπεποίθηση του πράκτορα να προβεί σε περισσότερες τέτοιες ενέργειες.

Οι αλγόριθμοι μηχανικής μάθησης έχουν πλέον εισέλθει στην καθημερινότητά μας σε μορφές προϊόντων και υπηρεσιών είτε το γνωρίζουμε είτε όχι. Ανάμεσα σε αυτές βρίσκονται:

- Λογισμικό προτάσεων σε μηχανές αναζήτησης, κοινωνικά δίκτυα, ηλεκτρονικά καταστήματα, πλατφόρμες βίντεο περιεχομένου κ.α. (Recommender Systems)
- Αυτόνομα οχήματα
- Ψηφιακοί βοηθοί σε κινητά, υπολογιστές και τηλεοράσεις
- Φίλτρα ανεπιθύμητης αλληλογραφίας
- Εφαρμογές πλοήγησης
- Upscaling σε εφαρμογές γραφικών
- Λογισμικό αυτόματης συμπλήρωσης κώδικα
- Λογισμικό αναγνώρισης κακόβουλων ενεργειών

Οι αλγόριθμοι μηχανικής μάθησης χτίζουν ένα μοντέλο βασισμένο σε δείγματα δεδομένων, γνωστά ως «δεδομένα εκπαίδευσης», προκειμένου να κάνουν προβλέψεις ή να πάρουν αποφάσεις χωρίς να έχουν προγραμματιστεί ρητά για αυτό. Για την ανάπτυξή τους συνήθως χρειάζεται μεγάλος όγκος δεδομένων. Ένα σημαντικό στοιχείο ως προς την επιτυχία τους είναι και η ποιότητα των δεδομένων.

2.3 Κοινωνικά Ζητήματα της Μηχανικής Μάθησης

Εφόσον οι αλγόριθμοι μηχανικής μάθησης δεν είναι τέλειοι με την ντετερμινιστική έννοια όπως οι κλασικοί αλγόριθμοι θα πρέπει να ορίσουμε το ελάχιστο σφάλμα ως κριτήριο για την αποτελεσματικότητά τους και την αξιοπιστία των εκάστοτε εξόδων τους.

Γενικότερα τα προβλήματα με τους αλγόριθμους μηχανικής μάθησης συνήθως εντοπίζονται σε δυο σημεία, σε αυτό των **δεδομένων** και στην **αδιαφάνεια της λειτουργίας** τους.

Σε σχέση με τα δεδομένα εντοπίζονται προβλήματα **εκδημοκρατισμού των δεδομένων** καθώς δεν έχουν όλοι πρόσβαση σε ποιοτικά δεδομένα ή και αρκετά μεγάλο όγκο δεδομένων. Από μια άλλη οπτική πολλές φορές αυτά τα δεδομένα (παρά τις προσπάθειες όπως το GDPR) έχουν συλλεχθεί με πλάγιους ή παράνομους τρόπους χωρίς συναίνεση των χρηστών.

Η **αδιαφάνεια στην λειτουργία τους** είναι κάτι το εγγενές (ειδικά σε εφαρμογές deep learning) που απαιτεί πολύ μεγάλη επένδυση σε χρόνο για να επιτύχουμε ερμηνευσιμότητα και αυτό με αμφίβολα αποτελέσματα. Χωρίς όμως να γνωρίζουμε για ποιο λόγο πάρθηκε μια απόφαση σε ένα τέτοιο σύστημα είναι πιθανό να μην μπορούν να αποδοθούν σωστά ευθύνες σε περίπτωση σφάλματος ή ακόμα και να γίνει root cause analysis.

Επιπλέον πρόβλημα που συνήθως προκύπτει από την ίδια την φύση των δεδομένων είναι η **μεροληψία** (biasing) έχει το χαρακτηριστικό να συντηρεί ή ακόμα και να ενισχύει την διαιώνιση υπαρχόντων αρνητικών διακρίσεων και στερεοτύπων.

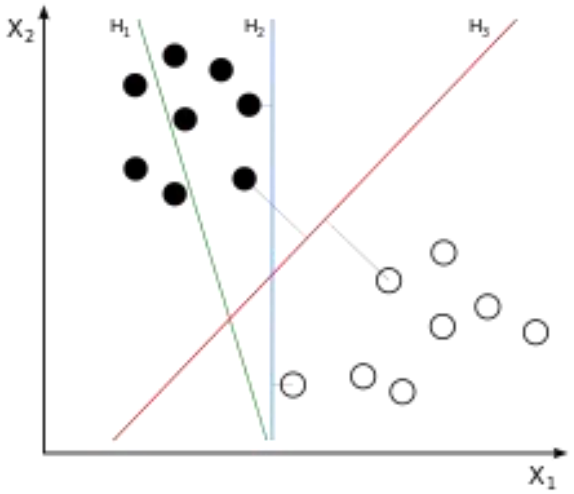
2.4 Τύποι Αλγορίθμων Μηχανικής Μάθησης

2.4.1 Μηχανές διανυσμάτων υποστήριξης (Support Vector Machines)

Τα SVM είναι μια μέθοδος μηχανικής μάθησης με επιτήρηση (supervised learning) που χρησιμοποιούνται για την ταξινόμηση (classification) και την ανάλυση παλινδρόμησης (regression analysis). Οι θεμελιώδεις αρχές των SVM βασίζονται στη θεωρία της στατιστικής μάθησης (statistical learning).

Θα εξετάσουμε ένα παράδειγμα για να κατανοήσουμε την λειτουργία του. Ας υποθέσουμε ότι ορισμένα σημεία ανήκουν το καθένα σε μία από συνολικά δύο κλάσεις και ο στόχος μας είναι να αποφασιστεί σε ποια κλάση ανήκει ένα νέο δεδομένο. Στην περίπτωση των SVM, ένα σημείο λογίζεται ως διάνυσμα p-διαστάσεων και θέλουμε να μάθουμε αν μπορούμε να κατηγοριοποιήσουμε τέτοια σημεία με ένα υπερεπίπεδο (p-1) διαστάσεων.

Αυτό ονομάζεται γραμμικός ταξινομητής. Υπάρχουν πολλά υπερεπίπεδα που θα μπορούσαν να ταξινομήσουν τα δεδομένα. Η βέλτιστη επιλογή υπερεπίπεδου είναι αυτή που αντιπροσωπεύει τον καλύτερο διαχωρισμό μεταξύ των δύο κατηγοριών. Όπου ο καλύτερος διαχωρισμός ορίζεται αν ως το υπερεπίπεδο που διατηρεί την μέγιστη απόσταση μέχρι το πλησιέστερο σημείο δεδομένων της κάθε κλάσης.



Εικόνα 3. Διαχωρισμός του επιπέδου σε SVM

Στο παράδειγμα 2-διαστάσεων της εικόνας, το H1 δεν διαχωρίζει σωστά τις κλάσεις αφού περιλαμβάνονται στοιχεία και από τις δύο σε κάθε πλευρά. Το H2 το κάνει, αλλά έχει μικρότερη απόσταση από την βέλτιστη από τα σημεία τους. Το H3 τα διαχωρίζει με τη μέγιστη απόσταση.
[1]

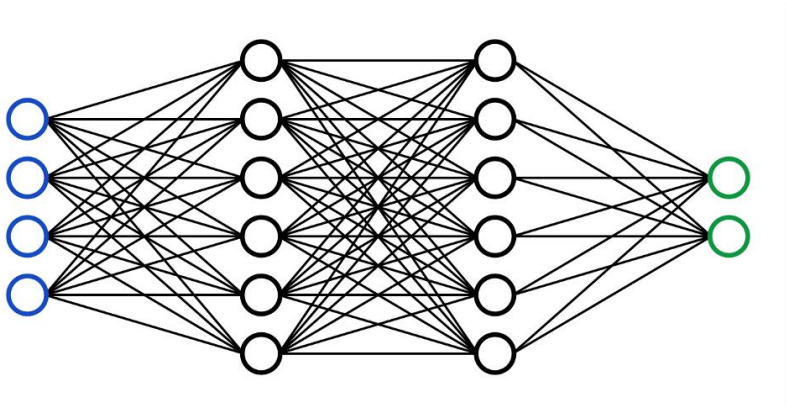
2.4.2 Νευρωνικά Δίκτυα

Τα ΤΝΔ (Τεχνητά Νευρωνικά Δίκτυα) είναι συστήματα με δομή εμπνευσμένη από τη λειτουργία του νευρικού συστήματος και του εγκεφάλου. Το κύριο επεξεργαστικό στοιχείο είναι μια απομίμηση του βιολογικού νευρώνα, ο τεχνητός νευρώνας. Είναι ένα χρονικά αναλλοίωτο σύστημα χωρίς μνήμη, με πολλές εισόδους και μία έξοδο.

Ο πυρήνας του νευρώνα δέχεται σήματα από άλλους νευρώνες, μέσω καναλιών εισόδου που ονομάζονται δενδρίτες (dendrites), και τα επεξεργάζεται για τη δημιουργία ενός καινούριου σήματος. Αν το σήμα αυτό είναι αρκετά ισχυρό, ενεργοποιείται η έξοδος του νευρώνα και παράγεται ένα σήμα εξόδου που μεταδίδεται μέσω ενός καναλιού εξόδου. Το κανάλι εξόδου ονομάζεται άξονας (axon) και η σύνδεσή του με τους δενδρίτες των άλλων νευρώνων γίνεται μέσω συνάψεων (synapses). Το μεταδιδόμενο σήμα μεταβάλλεται ανάλογα με την ισχύ της αντίστοιχης σύναψης. Ο ανθρώπινος εγκέφαλος αποτελείται από δεκάδες δισεκατομμυρίων νευρώνων.

Η ιδέα των τεχνητών νευρωνικών δικτύων (Artificial Neural Networks) άρχισε να αναπτύσσεται την δεκαετία του ‘50 από τον Frank Rosenblatt [9], ο οποίος εφηύρε τον νευρώνα Perceptron. Αν και οι νευρώνες Perceptron φάνηκαν πολλά υποσχόμενοι στην αρχή, εν τέλει αποδείχτηκε ότι δεν μπορούν να εκπαιδευτούν για να αναγνωρίζουν πολλές κατηγορίες προτύπων και η έρευνα στα νευρωνικά δίκτυα για πολλά χρόνια είχε εγκαταλειφθεί. Έπειτα εφευρέθηκαν δίκτυα με δύο ή περισσότερα επίπεδα τα οποία είχαν πολύ καλύτερα αποτελέσματα και ο χώρος ανέκτησε το χαμένο ενδιαφέρον της επιστημονικής κοινότητας.

Ένα ΤΝΔ αποτελείται από τη σειριακή, παράλληλη και με ανατροφοδότηση σύνδεση νευρώνων. Γενικά, τα ΤΝΔ διακρίνονται σε νευρωνικά δίκτυα πολλαπλών επιπέδων (multilayer neural networks) και επαναληπτικά (αναδρομικά) νευρωνικά δίκτυα (recurrent neural networks).[5]



Εικόνα 4. Παράδειγμα Τοπολογίας Νευρωνικού δικτύου δύο κρυφών επιπέδων

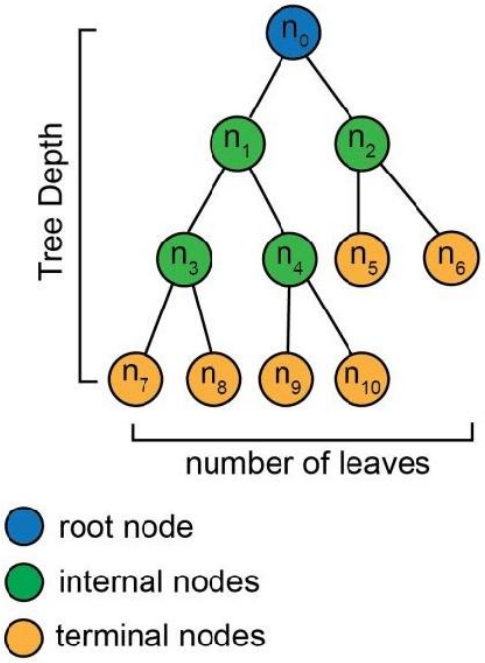
Ένα νευρωνικό δίκτυο μοιάζει περισσότερο με ένα “black box”, με την έννοια ότι μας είναι γνωστές οι αρχές λειτουργίας του όσον αφορά τις εισόδους και τις εξόδους του (ή τα χαρακτηριστικά του), χωρίς όμως να γνωρίζουμε την εσωτερική λειτουργία του. Για αυτό το λόγο μπορεί να παρέχει αποτελέσματα χωρίς εξήγηση για το πώς αυτά προέκυψαν. Έτσι, είναι δύσκολο ή αδύνατο να εξηγηθεί ο τρόπος λήψης αποφάσεων σε αντιπαραβολή με τα αποτελέσματα.

2.4.3 Δέντρα αποφάσεων

Η Μηχανική Μάθηση με δέντρα αποφάσεων (Decision Trees / Induction Decision Trees Learning) είναι μια από τις προσεγγίσεις προγνωστικής μοντελοποίησης που χρησιμοποιούνται στη στατιστική, την εξόρυξη δεδομένων και άλλα επιστημονικά πεδία.

Ένα δέντρο αποφάσεων είναι μια δομή που θα μπορούσε να αντιστοιχεί σε διάγραμμα ροής στην οποία κάθε εσωτερικός κόμβος (internal node) αντιπροσωπεύει έλεγχο σε ένα χαρακτηριστικό (π.χ. εάν μια αναστροφή νομίσματος έρχεται σε κορώνα ή γράμματα), κάθε κλάδος αντιπροσωπεύει το αποτέλεσμα της δοκιμής και κάθε φύλλο (terminal node) αντιπροσωπεύει μια ετικέτα κατάταξης. Στην ουσία είναι ένα δέντρο που μοντελοποιεί ένα σύνολο διαδοχικών, ιεραρχικών αποφάσεων που τελικά οδηγούν σε κάποιο τελικό αποτέλεσμα.

View Article Online
DOI: 10.1039/D0AY01389G



Εικόνα 5. Παράδειγμα Δέντρου απόφασης

Τα μοντέλα δέντρων στα οποία η μεταβλητή στόχος λαμβάνει ένα διακριτό σύνολο τιμών ονομάζονται δέντρα ταξινόμησης (Classification trees). Σε αυτές τις δομές δέντρων, τα φύλλα αντιπροσωπεύουν ετικέτες κλάσης και τα κλαδιά αντιπροσωπεύουν συνδέσμους χαρακτηριστικών που οδηγούν σε αυτές τις ετικέτες κλάσης. Τα δέντρα απόφασης όπου η μεταβλητή στόχος λαμβάνει συνεχείς τιμές (συνήθως πραγματικούς αριθμούς) ονομάζονται δέντρα παλινδρόμησης (Regression Trees).

Εάν κληθεί κάποιος να διερευνήσει τους λόγους που πάρθηκε μια απόφαση ενός συστήματος που βασίζεται σε νευρωνικά δίκτυα, είναι πολύ δύσκολο το να εξηγηθεί και να δικαιολογηθεί σε μη τεχνικά καταρτισμένα άτομα το πώς πάρθηκαν οι αποφάσεις και γιατί.

Αντίθετα, ένα δέντρο αποφάσεων έχει μια δομή που προσομοιάζει στον τρόπο λήψης αποφάσεων που χρησιμοποιούν οι άνθρωποι και έτσι η διαδικασία με την οποία μια συγκεκριμένη απόφαση λαμβάνεται είναι εύκολο να καταγραφεί και μπορεί να εξαχθεί από το μοντέλο άμεσα και να οπτικοποιηθεί.[6]

Gradient Boosting

Οι αλγόριθμοι Gradient Boosting έχουν εξελιχθεί τα τελευταία χρόνια ως επαναληπτικοί λειτουργικοί αλγόριθμοι κατάταξης κλίσης. Δηλαδή οι αλγόριθμοι όπου βελτιστοποιούν μια

συνάρτηση κόστους σε σχέση με το χώρο λειτουργίας επιλέγοντας με επαναληπτικό τρόπο μια συνάρτηση όπου δείχνει την κατεύθυνση αρνητικής κλίσης. Αυτή η λειτουργική άποψη κλίσης του Gradient Boosting οδηγεί στην ανάπτυξη αλγορίθμων Gradient Boosting σε πολλούς τομείς της μηχανικής μάθησης.

Όπως και με άλλες μεθόδους, το Gradient Boosting συνδυάζει τον αδύνατο παράγοντα με τον πιο ισχυρό με επαναληπτικό τρόπο. Για κάθε m, όπου m μεγαλύτερο ή ίσο με 1 και μικρότερο ή ίσο με M, της υποβοηθητικής κλίσης, μπορεί να υποθέτει ότι υπάρχει κάποιο ατελές μοντέλο Fm . Καλό θα ήταν στην αρχή να χρησιμοποιηθεί ένα πολύ αδύναμο μοντέλο που θα προβλέπει απλός τη μέση τιμή y για το εκπαιδευτικό κομμάτι του αλγορίθμου. Ο αλγόριθμος Gradient Boosting βελτιώνεται στο Fm με την κατασκευή ενός νέου μοντέλου που προσθέτει έναν εκτιμητή h για να παρέχει ένα καλύτερο μοντέλο. Fm+1(x) = Fm(x) +h(x). Για να βρούμε το h, η λύση αύξησης της κλίσης ξεκινάει με την παρατήρηση ότι μια τέλεια h υποδηλώνει Fm+1(x) = Fm(x) + h(x) = y

2.4.4 Δέντρα LightGBM

Το LightGBM είναι ένα framework gradient boosting που βασίζεται σε δέντρα αποφάσεων για την αύξηση της απόδοσης του μοντέλου και τη μείωση της χρήσης μνήμης.

Το **Gradient Boosting Decision Tree (GBDT)** είναι ένας αλγόριθμος μηχανικής μάθησης. Οι υλοποιήσεις του όπως το XGBoost και το pGBRT είναι κοινά αποδεκτές ως αποτελεσματικές. Όμως, αν και έχουν υιοθετηθεί πολλές βελτιστοποιήσεις σε αυτές τις υλοποιήσεις, η αποτελεσματικότητα και η αποδοτικότητα εξακολουθούν να μην είναι ικανοποιητικές όταν η διάσταση των features και το μέγεθος των δεδομένων αυξάνονται. Ένας κυρίαρχος λόγος είναι ότι για κάθε feature, πρέπει να σαρώσουν όλα τα υφιστάμενα δεδομένα για να επανεκτιμηθεί το **κέδρος** για όλα τα πιθανά **σημεία διακλάδωσης**, κάτι που είναι πολύ χρονοβόρο.

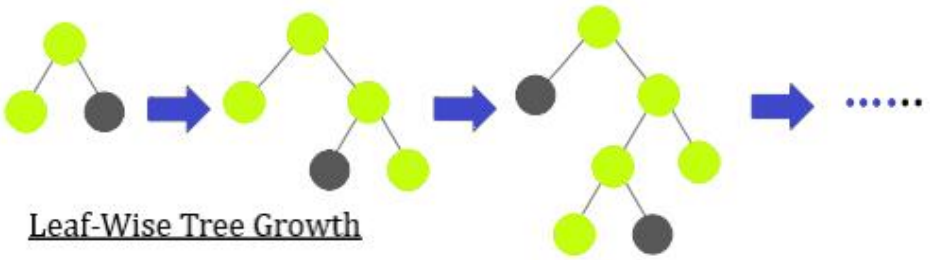
Για την αντιμετώπιση αυτού του προβλήματος, οι εφευρέτες του αλγορίθμου χρησιμοποίησαν δύο καινοτόμες τεχνικές: Gradient-based One-Side Sampling (GOSS) και Exclusive Feature Bundling (EFB). Με το GOSS, αποκλείεται ένα σημαντικό ποσοστό δεδομένων με μικρές διαβαθμίσεις και χρησιμοποιούνται μόνο τα υπόλοιπα για να εκτιμηθεί το κέρδος.

Έχει αποδειχθεί ότι, όταν τα δεδομένα με μεγαλύτερο βάρος παίζουν σημαντικότερο ρόλο στον υπολογισμό του κέρδους, το GOSS μπορεί να κάνει αρκετά ακριβείς εκτιμήσεις έχοντας στη διάθεσή του πολύ μικρότερο μέγεθος δεδομένων εισόδου.

Με το EFB, ομαδοποιούνται αμοιβαία αποκλειόμενα features, για να μειωθεί η διάσταση του προβλήματος. Η εύρεση της βέλτιστης ομαδοποίησης είναι NP-Hard, αλλά ένας άπληστος αλγόριθμος μπορεί να επιτύχει αρκετά καλή προσέγγιση χωρίς να βλάψει την ακρίβεια του προσδιορισμού σημείου διάσπασης.

Η συνδυαστική υλοποίηση GBDT με GOSS και EFB ονομάστηκε **LightGBM**. Πειράματά σε πολλαπλά ανοιχτά και δημοσίως διαθέσιμα σύνολα δεδομένων δείχνουν ότι το LightGBM

επιταχύνει τη διαδικασία εκπαίδευσης των συμβατικών GBDT έως και πάνω από 20 φορές, ενώ επιτυγχάνει σχεδόν την ίδια ακρίβεια.[3]



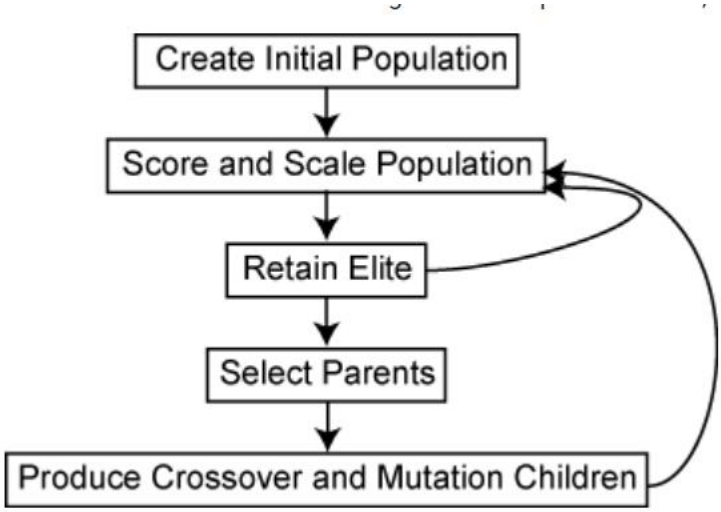
Εικόνα 6. Ανάπτυξη δέντρου leaf-wise στον LightGBM

Το LightGBM χωρίζει το δέντρο πρώτα κατά βάθος (leaf wise) σε αντίθεση με άλλους αλγόριθμους ενίσχυσης που αναπτύσσονται σε επίπεδο δέντρου. Επιλέγει το φύλλο με μέγιστη απώλεια δέλτα για να αναπτυχθεί. Δεδομένου ότι το φύλλο είναι σταθερό, ο αλγόριθμος σε επίπεδο φύλλων έχει χαμηλότερη απώλεια σε σύγκριση με τον αλγόριθμο επιπέδου. Η ανάπτυξη δέντρων από άποψη φύλλων μπορεί να αυξήσει την πολυπλοκότητα του μοντέλου και μπορεί να οδηγήσει σε overfitting σε μικρά σύνολα δεδομένων.

2.4.5 Εξελικτικοί αλγόριθμοι

Οι εξελικτικοί αλγόριθμοι (genetic algorithms) είναι μια μέθοδος για την επίλυση τόσο περιορισμένων όσο και μη περιορισμένων προβλημάτων βελτιστοποίησης που βασίζεται στη φυσική επιλογή, τη διαδικασία που οδηγεί τη βιολογική εξέλιξη. Ένας γενετικός αλγόριθμος τροποποιεί επανειλημμένα έναν πληθυσμό μεμονωμένων λύσεων. Σε κάθε βήμα, ο γενετικός αλγόριθμος επιλέγει άτομα από τον τρέχοντα πληθυσμό ως γονείς και τα χρησιμοποιεί για να παράγει παιδιά για την επόμενη γενιά. Κατά τη διάρκεια των διαδοχικών γενεών, ο πληθυσμός «εξελίσσεται» προς μια βέλτιστη λύση. Μπορούν να αντιμετωπίσουν προβλήματα προγραμματισμού μικτών ακέραιων και κινητής υποδιαστολής αριθμών, όπου ορισμένα στοιχεία είναι περιορισμένα να έχουν ακέραιες τιμές.

Το παρακάτω διάγραμμα ροής περιγράφει τα κύρια αλγοριθμικά βήματα:



Εικόνα 7. Διάγραμμα ροής γενετικού αλγόριθμου

Ένας γενετικός αλγόριθμος χρησιμοποιεί τρεις κύριους τύπους κανόνων σε κάθε βήμα για να δημιουργήσει την επόμενη γενιά από τον τρέχοντα πληθυσμό:

- Οι κανόνες επιλογής (Selection rules) επιλέγουν τα άτομα, που ονομάζονται γονείς, που συμβάλλουν στον πληθυσμό της επόμενης γενιάς. Η επιλογή είναι γενικά στοχαστική και μπορεί να εξαρτάται από τις βαθμολογίες των ατόμων.
- Οι κανόνες crossover (Crossover rules) συνδυάζουν δύο γονείς για να σχηματίσουν παιδιά για την επόμενη γενιά.
- Οι κανόνες μετάλλαξης (Mutation rules) εφαρμόζουν τυχαίες αλλαγές σε μεμονωμένους γονείς για να σχηματίσουν παιδιά.

Περίγραμμα του Αλγορίθμου

1. Το ακόλουθο περίγραμμα συνοψίζει πώς λειτουργεί ο γενετικός αλγόριθμος:
2. Ο αλγόριθμος ξεκινά δημιουργώντας έναν τυχαίο αρχικό πληθυσμό. Στη συνέχεια, ο αλγόριθμος δημιουργεί μια ακολουθία νέων πληθυσμών. Σε κάθε βήμα, ο αλγόριθμος χρησιμοποιεί τα άτομα της τρέχουσας γενιάς για να δημιουργήσει τον επόμενο πληθυσμό. Για τη δημιουργία του νέου πληθυσμού, ο αλγόριθμος εκτελεί τα ακόλουθα βήματα:
    a. Βαθμολογεί κάθε μέλος του τρέχοντος πληθυσμού υπολογίζοντας την τιμή καταλληλότητάς του. Αυτές οι τιμές ονομάζονται ακατέργαστες βαθμολογίες φυσικής κατάστασης.

    b. Κλιμακώνει τις ακατέργαστες βαθμολογίες φυσικής κατάστασης για να τις μετατρέψει σε ένα πιο χρησιμοποιήσιμο εύρος τιμών. Αυτές οι κλιμακούμενες τιμές ονομάζονται τιμές προσδοκίας.

c. Επιλέγει μέλη, που καλούνται γονείς, με βάση τις προσδοκίες τους.

d. Μερικά από τα άτομα του τρέχοντος πληθυσμού που έχουν χαμηλότερη φυσική κατάσταση επιλέγονται ως ελίτ (elite). Αυτά τα άτομα της ελίτ περνούν στον επόμενο πληθυσμό.

e. Παράγει παιδιά από τους γονείς. Τα παιδιά παράγονται είτε κάνοντας τυχαίες αλλαγές σε έναν μόνο γονέα—μετάλλαξη—ή συνδυάζοντας τις διανυσματικές εγγραφές ενός ζεύγους γονέων—διασταύρωση.

f. Αντικαθιστά τον σημερινό πληθυσμό με τα παιδιά για να σχηματίσει την επόμενη γενιά.

3. Ο αλγόριθμος σταματά όταν πληρούται ένα από τα κριτήρια διακοπής.
4. Ο αλγόριθμος κάνει τροποποιημένα βήματα για γραμμικούς και ακέραιους περιορισμούς.
5. Ο αλγόριθμος τροποποιείται περαιτέρω για μη γραμμικούς περιορισμούς.

[5]

2.5 Προκλήσεις της ανάπτυξης αλγορίθμων Μηχανικής Μάθησης

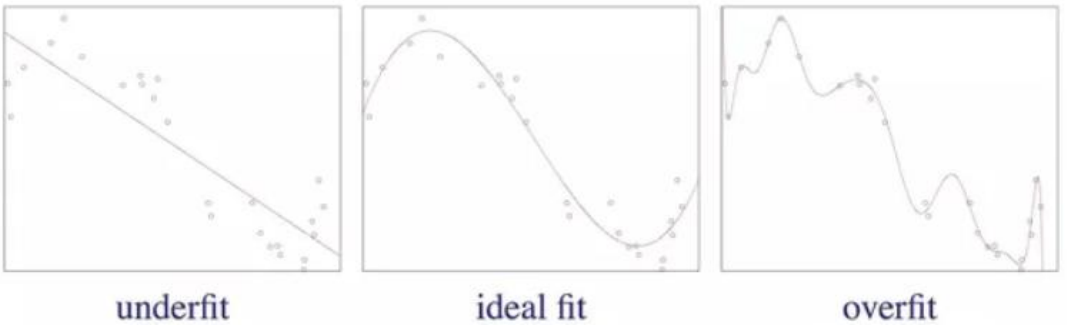
Ολοκληρώνοντας αυτή την αναφορά στις θεμελιώδεις έννοιες της μηχανικής μάθησης, κρίνεται σκόπιμη η παρουσίαση ενός συνόλου παραγόντων καθοριστικής σημασίας για τη σχεδίαση ενός αποδοτικού αλγόριθμου μηχανικής μάθησης.

Ο πλέον σημαντικός παράγοντας θεωρείται η ικανότητα γενίκευσης (**generalization ability**), η ικανότητά, δηλαδή, του αλγορίθμου να εφαρμόζει την «γνώση» που απέκτησε μετά την τροφοδότηση του από τα δεδομένα εκπαίδευσης σε δεδομένα που δεν έχει ξανασυναντήσει. Με την ικανότητα γενίκευσης να είναι ο απώτερος σκοπός ανακύπτουν μια σειρά από προκλήσεις ώστε να αποφευχθεί το αντίθετο αποτέλεσμα.

Στην αρχή θα εστιάσουμε σε εγγενή προβλήματα που συναντάμε κατά την ανάπτυξή τους και έπειτα σε μεθόδους αντιμετώπισής τους μέσω καλών πρακτικών (best practices).

2.5.1 Overfitting

Οι αλγόριθμοι μηχανικής μάθησης είναι πολύ αποτελεσματικοί στην εκμάθηση μιας αντιστοίχισης μεταξύ των features και των γνωστών τιμών στόχου (target) στα υπάρχοντα δεδομένα. Αν αφεθούν χωρίς επίβλεψη, μπορούν συχνά να δημιουργήσουν μια 100% ακριβή χαρτογράφηση.



Εικόνα 8. Παράδειγμα Underfitting, Ideal fit, Overfitting

Οπότε ένα μοντέλο που είναι αρκετά περίπλοκο ώστε να ταιριάζει απόλυτα στα υπάρχοντα δεδομένα δεν θα μπορεί να γενικεύσει σωστά νέες παρατηρήσεις. Το φαινόμενο αυτό ονομάζεται **overfitting**. Μπορεί να δώσει ακριβείς απαντήσεις για κάποιες παρατηρήσεις κατά τύχη, αλλά γενικά δεν θα αντιπροσωπεύει την τάση των δεδομένων. Ένα δέντρο αποφάσεων είναι ένα εξαιρετικό παράδειγμα ενός αλγορίθμου για να εξηγηθεί το φαινόμενο, εάν το δέντρο επιτρέπεται να συνεχίσει να χωρίζει τα δεδομένα μέχρι κάθε παρατήρηση να είναι στο δικό της φύλλο, θα είναι 100% ακριβές για κάθε παρατήρηση στα δεδομένα εκπαίδευσης. Αλλά μετά από ένα ορισμένο βάθος, το δέντρο δεν παρέχει καμία πληροφορία που μπορεί να γενικευθεί.

2.5.2 Ακρίβεια των δεδομένων / Data Acuracy

Οι αλγόριθμοι μηχανικής μάθησης δημιουργούν μοντέλα που είναι αντιπροσωπευτικά των διαθέσιμων δεδομένων εκπαίδευσης, οι αλγόριθμοι μπορεί να είναι πολύ ανακριβές έξω από αυτόν τον υπο-χώρο των δεδομένων

2.5.3 Απούσες τιμές

Ένα μεγάλο πρόβλημα κατά την ανάπτυξη αλγορίθμων τεχνητής μάθησης είναι όταν στα δεδομένα απουσιάζουν τιμές, ειδικά όταν αυτές αφορούν τη μεταβλητή στόχο (target). Όταν αντιμετωπίζουμε τιμές που λείπουν στις μεταβλητές εισόδου, πρέπει να εξεταστεί εάν οι τιμές που λείπουν κατανέμονται τυχαία ή εάν η έλλειψη μπορεί κατά κάποιο τρόπο να οδηγήσει στον στόχο. Εάν οι τιμές που λείπουν εμφανίζονται τυχαία στα δεδομένα εισόδου, μπορούν να απορριφθούν από την ανάλυση χωρίς να εισάγεται μεροληψία στο μοντέλο.

Ωστόσο, μια τέτοια επιλογή αν εφαρμοστεί σε μεγάλη κλίμακα μπορεί να αφαιρέσει έναν τεράστιο όγκο πληροφοριών από το δεδομένα εκπαίδευσης και μείωση της ακρίβειας του μοντέλου.

2.5.4 Non-Standardization

Όταν τα δεδομένα έχουν features των οποίων οι τιμές διαφέρουν σημαντικά ως προς το μέγεθος και το εύρος, αυτή η διαφορά μπορεί να υποβαθμίσει την απόδοση του αλγορίθμου μηχανικής μάθησης. Τα features που έχουν μεγάλη διακύμανση στις τιμές τους κυριαρχούν σε σχέση με άλλα χαρακτηριστικά εισόδου και εμποδίζουν το μοντέλο να μπορέσει να μάθει τη σχέση με τα άλλα features.

2.5.5 Ακραίες τιμές / Outliers

Όταν ένα dataset έχει παρατηρήσεις που είναι πολύ διαφορετικές από τις άλλες σε μία ή περισσότερες από τις τιμές των features τότε - αν και οι ακραίες τιμές μπορούν σίγουρα να είναι πολύ κατατοπιστικές και μπορούν να εντοπίσουν ανωμαλίες που αξίζουν ιδιαίτερης προσοχής - μπορεί να είναι και αρκετά επιζήμιες για την εκπαίδευση ενός αποτελεσματικού γενικεύσιμου μοντέλου.

2.5.6 Dimensionality

Διαισθητικά μπορεί να υποθέσουμε ότι το να έχουμε περισσότερες πληροφορίες θα μας επιτρέψει να πάρουμε καλύτερες αποφάσεις. Ωστόσο, αυτή η πεποίθηση βασίζεται στην υπόθεση ότι θα είμαστε σε θέση να διακρίνουμε εύκολα τις σημαντικές πληροφορίες από τις ασήμαντες και να τις επεξεργαστούμε αποτελεσματικά. Στην πραγματικότητα, τόσο περισσότερες οι πληροφορίες, τόσο πιο περίπλοκη και δαπανηρή γίνεται η διαδικασία λήψης αποφάσεων. Το φαινόμενο αυτό είναι γνωστό ως κατάρα του Dimensionality (Bellman 1957). Η αύξηση του αριθμού των features αυξάνει των χώρο των features εκθετικά. με τη σειρά του, αυτός ο χώρος υψηλότερων διαστάσεων απαιτεί εκθετικά περισσότερα σημεία δεδομένων για να γεμίσει επαρκώς ώστε τα features να λαμβάνονται υπόψη.[11]

2.5.7 Interpretability (Ερμηνευσιμότητα)

Στα συστήματα που χρησιμοποιούν αλγορίθμους Μηχανικής Μάθησης υπάρχει μια εγγενής δυσκολία στο γνωρίζουμε το λόγο που πάρθηκε μια απόφαση από το σύστημα. Το πρόβλημα αυτό είναι πιο συχνά εμφανές σε αλγορίθμους βαθιάς μάθησης (Deep learning).

Ο αριθμός των features ενός μοντέλου μπορεί να είναι της τάξης των εκατοντάδων εκατομμυρίων. Αυτή η μη γραμμικότητα μπορεί να μην οδηγεί απαραίτητα σε αδιαφάνεια (για παράδειγμα, ένα μοντέλο δέντρου αποφάσεων δεν είναι γραμμικό αλλά ερμηνεύσιμο), η σειρά μη γραμμικών πράξεων της βαθιάς μάθησης πράγματι μας εμποδίζει να κατανοήσουμε την εσωτερική του λειτουργία. Επιπλέον, η αναδρομικότητα είναι μια άλλη πηγή δυσκολίας. Είναι γνωστό ότι ακόμη και ένα απλό αναδρομικό μαθηματικό μοντέλο μπορεί να οδηγήσει σε μια δυσεπίλυτη δυναμική. Έχει αποδειχθεί ότι υπάρχουν χαοτικές συμπεριφορές όπως διακλαδώσεις ακόμη και σε απλά νευρωνικά δίκτυα. Σε χαοτικά συστήματα, μικροσκοπικές

αλλαγές των αρχικών εισροών μπορεί να οδηγήσουν σε τεράστιες διαφορές στα αποτελέσματα, προσθέτοντας στην πολυπλοκότητα στις μεθόδους ερμηνείας.

Είναι ένας τομέας που λαμβάνει όλο και περισσότερη δημοσιότητα σε συνάρτηση με την αύξηση του αριθμού των εφαρμογών Deep learning που εισέρχονται στη ζωή μας. Οι λόγοι που είναι σημαντικό εμπόδιο που πρέπει να το ξεπεράσει μια εφαρμογή είναι:

- Το interpretability παίζει σημαντικό ρόλο στην ηθική χρήση των τεχνικών μηχανικής μάθησης όπως για παράδειγμα όταν χρειαστεί να λογοδοτήσει ένα σύστημα μηχανικής μάθησης.
- Επίσης μπορεί να βοηθήσει στον εντοπισμό πιθανών τρωτών σημείων ενός πολύπλοκου μοντέλου, βελτιώνοντας έτσι την ακρίβεια και την αξιοπιστία του.

Εάν ένας κατασκευαστής μοντέλων μπορεί να εξηγήσει γιατί ένα μοντέλο λαμβάνει μια συγκεκριμένη απόφαση υπό ορισμένες συνθήκες και οι χρήστες θα γνωρίζουν εάν ένα τέτοιο μοντέλο συμβάλλει σε ένα ανεπιθύμητο συμβάν ή όχι. [12]

2.5.8 Biasing (Μεροληψία)

Τέλος πολύ σημαντικός παράγοντας που αφορά τις κοινωνικές προεκτάσεις τους και παρόλα αυτά μπορεί να βρεθεί στον πυρήνα των αλγόριθμων (τα δεδομένα εκπαίδευσης) είναι η αμερόληπτη λειτουργία τους. Η αποφυγή δηλαδή του να εισαχθούν στους αλγόριθμους αυτούς μεροληψίες εναντίων κοινωνικών ομάδων και μειονοτήτων. Η σκοπιά αυτή έχει λάβει μεγάλη δημοσιότητα τα τελευταία έτη.

Έχει αποδειχθεί ότι χωρίς την κατάλληλη παρέμβαση κατά την εκμάθηση ή αξιολόγηση, τα μοντέλα μπορεί να είναι προκατειλημμένα έναντι ορισμένων κοινωνικών ομάδων, δηλαδή μπορεί να είναι επιρρεπή σε διακρίσεις:

- φυλετικές,
- σεξιστικές,
- ηλικιακές,
- σεξουαλικού προσανατολισμού,
- θρησκευτικές,
- εθνικιστικές. [7]

Αυτό οφείλεται στο γεγονός ότι τα δεδομένα που χρησιμοποιούνται για την εκπαίδευση τους περιέχουν συχνά προκαταλήψεις που ενισχύονται στο μοντέλο.

(Δείτε και τον πειραματικό κώδικα της Google στη διεύθυνση https://github.com/google-research/google-research/tree/master/label_bias.) [7]

2.6 Συνήθεις καλές πρακτικές ανάπτυξης αλγορίθμων Μηχανικής Μάθησης

2.6.1 Διαχωρισμός δεδομένων εκπαίδευσης/επαλήθευσης

Για την ανάπτυξη τέτοιου είδους αλγορίθμων είναι απαραίτητο να πραγματοποιούνται πολλές δοκιμές μέχρι να επιτευχθεί το επιθυμητό αποτέλεσμα. Αυτό συμβαίνει διότι οι αλγόριθμοι είναι μεν γενικά εφαρμόσιμοι σε πολλά και διαφορετικά μεταξύ τους προβλήματα και ακόμα και πεδία, αλλά το κάθε πρόβλημα έχει μια μοναδικότητα στα δεδομένα του αλλά και στο στόχο που επιχειρεί να πετύχει.

Όταν λοιπόν έχουμε διαθέσιμο ένα σύνολο δεδομένων και αναπτύσσουμε ένα μοντέλο με σκοπό την πρόβλεψη τιμών πολλές φορές δεν μπορούμε να κάνουμε δοκιμές σε νέα πραγματικά δεδομένα. Αντ' αυτού διαχωρίζουμε εκ των προτέρων τα δεδομένα που έχουμε σε δύο σύνολα : εκπαίδευσης και επαλήθευσης (training, validation/test)

Τα δεδομένα εκπαίδευσης χρησιμοποιούνται για την ανάπτυξη του μοντέλου και τα επαλήθευσης για την επαλήθευση της ακρίβειάς του. Τα τελευταία δεν είναι διαθέσιμα παρά μόνο μετά την διαδικασία εκπαίδευσης και είναι ενδεικτικά ως προς την ικανότητα γενίκευσης του αλγορίθμου.

2.6.2 k-fold

Ο αλγόριθμος k-fold είναι μια τεχνική επικύρωσης του βαθμού με τον οποίο τα αποτελέσματα μιας στατιστικής ανάλυσης μπορούν να γενικευθούν σε ένα ανεξάρτητο σύνολο δεδομένων.

Σύμφωνα με την k-fold διασταυρούμενη επικύρωση (cross validation), το αρχικό δείγμα διαιρείται τυχαία σε k ίσου μεγέθους υπο-δείγματα. Από τα υπο-δείγματα k, ένα μόνο υπο-δείγμα διατηρείται ως σύνολο επικύρωσης για τη δοκιμή του μοντέλου και τα υπόλοιπα υπο-δείγματα k − 1 χρησιμοποιούνται ως δεδομένα εκπαίδευσης.

Στη συνέχεια, η διαδικασία διασταυρούμενης επικύρωσης επαναλαμβάνεται k φορές, με καθένα από τα υπο-δείγματα k να χρησιμοποιείται ακριβώς μία φορά ως σύνολο επικύρωσης. Τέλος, ο μέσος όρος των αποτελεσμάτων k μπορεί να υπολογιστεί για να παραχθεί μία μοναδική εκτίμηση.

Το πλεονέκτημα αυτής της μεθόδου έναντι της επαναλαμβανόμενης τυχαίας υπο-δειγματοληψίας είναι ότι όλες οι παρατηρήσεις χρησιμοποιούνται τόσο για εκπαίδευση όσο και για επικύρωση και κάθε παρατήρηση χρησιμοποιείται για επικύρωση ακριβώς μία φορά.

2.6.3 Standardization

Σε περιπτώσεις που τα δεδομένα έχουν πολλή μεγάλη διακύμανση στις πιθανές τιμές τους - ίσως και τάξεις μεγέθους - όπως αναφέρθηκε παραπάνω μπορεί κάποια features να επισκιάσουν άλλα με αρνητική επίδραση στην εκπαίδευση του μοντέλου. Μετατρέποντας τα ώστε να βρίσκονται σε παρόμοια κλίμακα μπορεί να μετριαστεί η αρνητική επίδραση των τιμών αυτών. Μια τεχνική που μπορεί να ακολουθηθεί είναι να μετατραπούν οι τιμές των

features ώστε να είναι σε παρόμοια κλίμακα. Μια συνήθης τακτική είναι να κανονικοποιηθούν ώστε να έχουν μέση τιμή 0 και διακύμανση 1 (z-score). [11]

2.6.4 Διαχείριση Outliers

Outliers είναι παρατηρήσεις που είναι πολύ διαφορετικές από τις άλλες σε μία ή περισσότερες από τις τιμές των features. Οι ακραίες αυτές τιμές μπορούν συνήθως να ανιχνευθούν με κάποια απλή αρχική εξέταση των δεδομένων. Αρχικά, μπορεί να προσδιοριστεί εάν μια ακραία τιμή είναι απλώς μια μη έγκυρη ή εσφαλμένη καταχώριση που μπορεί να αγνοηθεί. Εάν διαπιστωθεί ότι ένα ακραίο στοιχείο δεν παρέχει πολύτιμες πληροφορίες, είναι αποδεκτό απλώς να απορριφθεί. Ωστόσο, εάν διαπιστωθεί ότι οι ακραίες τιμές μπορεί να αντιπροσωπεύουν κάποια πραγματική αλλά σπάνια σχέση ή αν είναι πιθανό οι πληροφορίες από τα άλλα χαρακτηριστικά σε αυτές τις παρατηρήσεις να είναι πολύ πολύτιμες για να απορριφθούν, μπορεί να χρησιμοποιηθεί μία από τις ακόλουθες τεχνικές:

- Για κατηγορικές μεταβλητές, οι τιμές μπορούν να τοποθετηθούν σε μια γενική κατηγορία «Άλλο».
- Οι ακραίες ποσοτικές μεταβλητές, μπορούν να οριστούν στη χαμηλότερη ή υψηλότερη μη ακραία τιμή, ή αναγκάζοντας τους να μην είναι μεγαλύτερες από τρεις τυπικές αποκλίσεις από τον μέσο όρο

Οποιαδήποτε από αυτές τις προσεγγίσεις διατηρεί την παρατήρηση, το οποίο μπορεί να περιέχει άλλες πολύτιμες πληροφορίες από άλλα χαρακτηριστικά.

Για αλγόριθμους που ενσωματώνουν ένα loss function για να κατευθύνει τη διαδικασία εκπαίδευσης, μπορεί να χρησιμοποιηθεί το Huber loss function (Huber 1964), η οποία μειώνει σημαντικά την επίδραση των ακραίων τιμών στον υπολογισμό του loss[11]

2.6.5 Binning

Binning είναι η διαδικασία διακριτοποίησης αριθμητικών μεταβλητών σε λιγότερες κατηγορικές αντίστοιχες. Για παράδειγμα, οι μεταβλητές «ηλικίας» συχνά δεσμεύονται σε κατηγορίες όπως 20–39, 40–59 και 60–79. Χτίζοντας ένα μοντέλο για κάθε μεμονωμένη ηλικία πιθανότατα δεν παρέχει περισσότερες πληροφορίες συγκριτικά με ένα μοντέλο που χρησιμοποιεί ηλικιακές ομάδες. Το Binning τείνει να δημιουργεί ένα πιο αποτελεσματικό μοντέλο πρόβλεψης και μπορεί να κάνει το μοντέλο πιο ερμηνεύσιμο :

- Μείωση του αντίκτυπου των Outliers
- Ενσωμάτωση τιμών που λείπουν
- Διαχείριση μεταβλητών υψηλής πληθικότητας
- Μείωση του θορύβου ή της μη γραμμικότητας

[11]

2.6.6 Αντιμετώπιση τιμών που απουσιάζουν

Σε περιπτώσεις που τιμές απουσιάζουν από τα δεδομένα κάποιες πρακτικές για την διαχείριση της καταστάσεις είναι:

- Χρήση Naïve Bayes μοντέλων. Οι τιμές που λείπουν για την εκπαίδευση και τη βαθμολογία προκύπτουν υπολογίζοντας την πιθανότητα με βάση τα παρατηρούμενα χαρακτηριστικά. Λόγω της υπό όρους ανεξαρτησίας μεταξύ των χαρακτηριστικών, ο Naive Bayes αλγόριθμος αγνοεί ένα χαρακτηριστικό μόνο όταν λείπει η τιμή του.
- Imputation: αντικατάσταση μιας τιμής που λείπει με πληροφορίες που προέρχονται από άλλες υπαρκτές στα δεδομένα εκπαίδευσης. Για κανονικά κατανεμημένες μεταβλητές η αντικατάσταση μιας τιμής που λείπει μπορεί να γίνει με τον μέσο όρο. Όμως για μη κανονικά κατανεμημένες μεταβλητές ή μεταβλητές που έχουν υψηλό ποσοστό τιμών που λείπουν, ο καταλογισμός μέσου μπορεί να αλλάξει δραστικά την κατανομή μιας μεταβλητής και να επηρεάσει αρνητικά προγνωστική ακρίβεια.
- Δέντρα απόφασης: Τα δέντρα απόφασης επιτρέπουν άμεση χρήση τιμών που λείπουν με δύο συνήθεις τρόπους:
  - Όταν καθορίζεται ένας κανόνας διαχωρισμού, η έλλειψη μπορεί να θεωρηθεί ως έγκυρη τιμή εισόδου και οι τιμές που λείπουν μπορούν είτε να τοποθετηθούν στην πλευρά του κανόνα διαχωρισμού που κάνει την καλύτερη πρόβλεψη ή να αντιστοιχιστούν σε ξεχωριστό κλάδο διαχωρισμού.
  - Μπορούν να οριστούν κανόνες υποκατάστασης. Όταν ένα λείπει η τιμή ενός feature τότε ο διαχωρισμός μπορεί να αποφασίζεται από ένα ορισμένο εκ των προτέρων feature αντικατάστασης. Π.χ. αν λείπει η τιμή του “ταχυδρομικού κώδικα” θα μπορούσε να γίνεται η επιλογή με βάση το feature “περιοχή”

[11]

Κεφάλαιο 3ο

3 Πρόβλεψη αστάθειας με Μηχανική Μάθηση

3.1 Ανάλυση του Dataset

Στα πλαίσια του διαγωνισμού παρέχονται δεδομένα τα οποία είναι κατανεμημένα σε χρονοθυρίδες των δέκα λεπτών. Τα δεδομένα είναι πραγματικά και έχουν γίνει anonymized από κάποιο άγνωστο σε εμάς ανταλλακτήριο. Η κάθε χρονοθυρίδα έχει ένα αναγνωριστικό time_id το οποίο ταυτοποιεί ένα χρονικό διάστημα κοινό για όλες τις μετοχές του ανταλλακτηρίου.

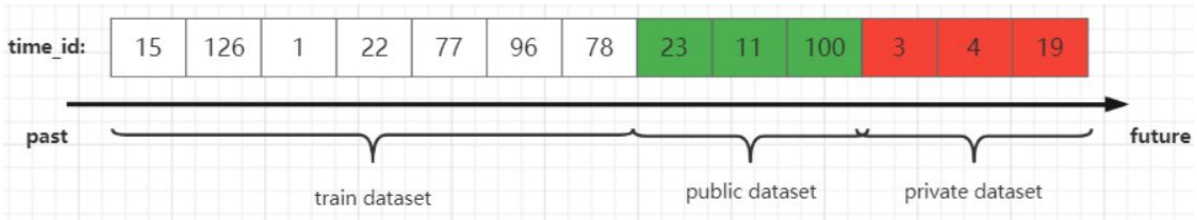
Η χρονική του τοποθέτηση με την έννοια της αλληλουχίας όμως δεν είναι γνωστή. Οι διοργανωτές έχουν **ανακατανείμει τις χρονοθυρίδες με τυχαίο τρόπο** ώστε να μην έχουμε στην διάθεσή μας μια χρονολογική σειρά. Οι μετοχές επίσης χαρακτηρίζονται από ένα μοναδικό χαρακτηριστικό “stock_id” και επίσης μας είναι γνωστό το ότι τα time_ids είναι τα ίδια για το σύνολο των μετοχών.

Ναι μεν τα δεδομένα μέσα στην κάθε χρονοθυρίδα μπορούν να χρησιμοποιηθούν σαν είσοδος για την πρόγνωση της αστάθειας αλλά η μεγαλύτερη εικόνα της διαδοχικότητας των τιμών είναι κρυμμένη. Κατά συνέπεια η τοποθέτηση τους σε σειρά αν και θα μπορούσε να βοηθήσει με την εκπαίδευση του μοντέλου δεν είναι κάτι που αφορά τον αλγόριθμο που αναπτύχθηκε.

Το **train dataset** αποτελείται από δεδομένα διαθέσιμα τοπικά σε όλους ώστε να γίνει η ανάπτυξη λογισμικού.

Το **public dataset** είναι ένα “**κρυφό**” dataset το οποίο έχει επιπλέον test δεδομένα στα οποία δόθηκε πρόσβαση με έμμεσο τρόπο, αφού αυτά ήταν διαθέσιμα μόνο στο εκτελέσιμο της εφαρμογής (run time) και μόνο μέσω της cloud πλατφόρμας. Υπήρχε η δυνατότητα να τρέξουμε τον κώδικα με αυτά όταν ανεβάζαμε τον κώδικά μας στην cloud πλατφόρμα και να πάρουμε το αποτέλεσμα. Με αυτό τον τρόπο μπορούσε να ελεγχθεί με μεγαλύτερη αξιοπιστία το πόσο καλά μπορεί να γενικεύσει ο αλγόριθμος.

Το **private dataset** είναι αυτό πάνω στο οποίο έγινε η αξιολόγηση και τα δεδομένα του συλλέχθηκαν μετά την καταληκτική ημερομηνία υποβολής.



Εικόνα 9. Απεικόνιση της τυχαίας διάταξης των χρονοθυρίδων του dataset

Στο παραπάνω διάγραμμα απεικονίζονται οι χρονοθυρίδες όπως θεωρητικά θα έπρεπε να ήταν ταξινομημένες με βάση τον χρόνο. Επιπλέον είναι πολύ σημαντικό το ότι πέρα από τη διάταξή τους **υπάρχουν και κενά μεταξύ τους**. Άρα ακόμη και αν γνωρίζαμε την διάταξη δεν θα μπορούσαμε να ξέρουμε αν για παράδειγμα η χρονοθυρίδα 78 είναι το αμέσως επόμενο δεκάλεπτο της 96 αλλά ούτε και το χρονικό διάστημα μεταξύ τους.

Το μόνο γνωστό είναι ότι το public dataset αφορά σε πιο μελλοντικές τιμές από το train dataset και ότι το private dataset σε ακόμη πιο μελλοντικές οι οποίες μάλιστα ήταν πραγματικές και συλλέχθηκαν μετά την προθεσμία του διαγωνισμού και την παράδοση του κώδικα από τους συμμετέχοντες.

Άρα το πρόβλημα ανάγεται σε ανάπτυξη λογισμικού πρόβλεψης σημάτων (short-term signals), με ορίζοντα δεκαλέπτου.

	A	B	C	D	E	F	G
1		time_id	seconds_in_bucket	price	size	order_coun	stock_id
2	0	5	21	1.0023013	326	12	0
3	1	5	46	1.002778	128	4	0
4	2	5	50	1.0028185	55	1	0
5	3	5	57	1.0031554	121	5	0
6	4	5	68	1.0036459	4	1	0
7	5	5	78	1.0037625	134	5	0
8	6	5	122	1.0042067	102	3	0
9	7	5	127	1.0045768	1	1	0
10	8	5	144	1.00437	6	1	0
11	9	5	147	1.0039636	233	4	0
12	10	5	177	1.0038528	1	1	0
13	11	5	183	1.0039562	2	1	0
14	12	5	187	1.0042665	165	2	0
15	13	5	207	1.0035425	72	4	0
16	14	5	218	1.0041553	33	5	0

Average of observations in time bucket	32.23

Πίνακας 4. Παράδειγμα Trade Book για το stock_id 0

	A	B	C	D	E	F	G	H	I	J	K	L
1		time_id	seconds_in_bucket	bid_price1	ask_price1	bid_price2	ask_price2	bid_size1	ask_size1	bid_size2	ask_size2	stock_id
2	0	5	0	1.0014222	1.0023013	1.0013704	1.0023531	3	226	2	100	0
3	1	5	1	1.0014222	1.0023013	1.0013704	1.0023531	3	100	2	100	0
4	2	5	5	1.0014222	1.0023013	1.0013704	1.0024048	3	100	2	100	0
5	3	5	6	1.0014222	1.0023013	1.0013704	1.0024048	3	126	2	100	0
6	4	5	7	1.0014222	1.0023013	1.0013704	1.0024048	3	126	2	100	0
7	5	5	11	1.0014222	1.0023013	1.0013704	1.0024048	3	100	2	100	0
8	6	5	12	1.0014222	1.0023013	1.0013704	1.0024048	3	126	2	100	0
9	7	5	14	1.0014222	1.0023013	1.0013704	1.0024048	3	126	2	100	0
10	8	5	15	1.0014222	1.0023013	1.0013704	1.0024048	3	126	2	100	0
11	9	5	16	1.0014222	1.0023013	1.0013704	1.0024048	3	126	2	100	0
12	10	5	17	1.0014222	1.0023013	1.0013704	1.0024048	3	100	2	100	0
13	11	5	18	1.0014222	1.0023013	1.0013704	1.0024048	3	126	2	100	0
14	12	5	19	1.0014222	1.0023013	1.0013704	1.0024048	3	126	2	100	0
15	13	5	21	1.0014222	1.0028185	1.0013704	1.0029219	3	30	2	100	0
16	14	5	24	1.0014739	1.0028185	1.0014222	1.0029219	155	30	3	100	0
17	15	5	25	1.0017325	1.0028185	1.0014222	1.0029219	83	30	3	100	0
18	16	5	44	1.0017325	1.0028185	1.0014222	1.0029219	83	28	3	100	0
19	17	5	46	1.0028185	1.0032322	1.0023013	1.0038011	155	1	200	34	0
20	18	5	47	1.0028185	1.0032322	1.0023013	1.0038011	55	1	100	34	0

Average Count of Observations in time bucket	239.57

Πίνακας 5. Παράδειγμα Order Book για το stock_id 0

Στις παραπάνω εικόνες έχουμε ένα κομμάτι των δεδομένων που αφορούν σην μετοχή με stock_id = 0. Τα δεδομένα χωρίζονται σε δύο αρχεία. Το πρώτο είναι μια απεικόνιση του trade book του ανταλλακτηρίου. Το trade book αποτυπώνει τις συναλλαγές που πραγματοποιήθηκαν. Τα στοιχεία που περιλαμβάνει είναι:

- time_id: unique identifier της χρονοθυρίδας, κοινό για όλες τις μετοχές αλλά και σε σχέση με το order book
- seconds_in_bucket: τα δευτερόλεπτα εντός του δεκαλέπτου που πάρθηκε το στιγμιότυπο
- price: η τιμή της μετοχής για την συναλλαγή
- size: ο αριθμός μετοχών που πουλήθηκαν
- order_count: ο αριθμός των διακριτών εντολών που υλοποιήθηκαν (συνήθως υποδεικνύει αριθμό αγοραστών)
- stock_id: unique identifier της μετοχής, κοινός με το order book

Το order book αφορά τα στοιχεία εντολών και υποδηλώνει την πρόθεση αγοραπωλησιών. Περιλαμβάνει τα εξής δεδομένα:

- time_id: unique identifier της χρονοθυρίδας, κοινό για όλες τις μετοχές αλλά και σε σχέση με το trade_book
- seconds_in_bucket: τα δευτερόλεπτα εντός του δεκαλέπτου που πάρθηκε το στιγμιότυπο
- bid_price_1: η τιμή της πιο χαμηλής εντολής αγοράς
- bid_price_2: η τιμή της δεύτερης πιο χαμηλής εντολής αγοράς
- ask_price_1: η τιμή της πιο χαμηλής εντολής πώλησης
- ask_price_2: η τιμή της δεύτερης πιο χαμηλής εντολής πώλησης
- bid_size_1: ο αριθμός μετοχών που ζητούνται στην τιμή bid_price_1
- bid_size_2: ο αριθμός μετοχών που ζητούνται στην τιμή bid_price_2
- ask_size_1: ο αριθμός μετοχών που πωλούνται στην τιμή ask_price_1
- ask_size_2: ο αριθμός μετοχών που πωλούνται στην τιμή ask_price_2
- stock_id: unique identifier της μετοχής, κοινός με το trade book

Σχετικά με την συχνότητα λήψης των στιγμιότυπων προκύπτει ότι το order book είναι πολύ πιο πυκνό από το trade book. Στο μεν trade book συναντάμε κατά μέσο όρο περίπου 30 παρατηρήσεις ενώ στο order book περίπου 240 μέσα σε ένα δεκάλεπτο. Αυτό είναι πολύ σύνηθες στα ανταλλακτήρια και παίζει τον ρόλο του στην διαδικασία binning των τιμών όπως θα δούμε παρακάτω.

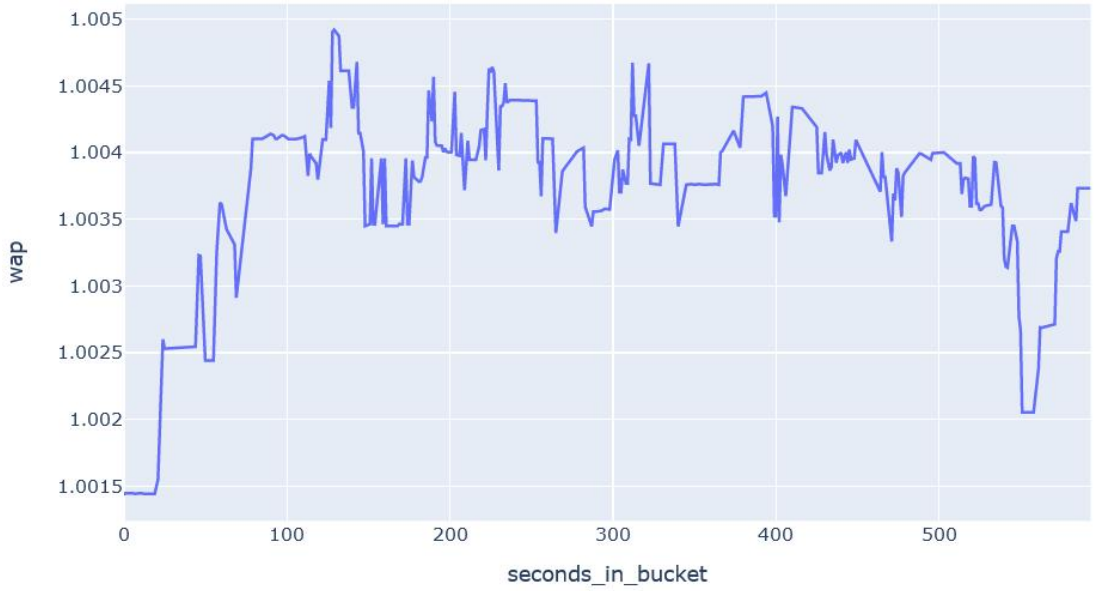
Ως target του προβλήματος είναι η πρόβλεψη της αστάθειας του επόμενου δεκαλέπτου του order book. Οπως αναλύθηκε στο 1ο κεφάλαιο, ο υπολογισμός της αστάθειας βασίζεται σε αυτόν του Log Return. Με τη σειρά του το Log return υπολογίζεται με βάση το Weighted Average Price και τέλος αυτό υπολογίζεται από τα παραπάνω δεδομένα. Για κάθε δεκάλεπτο του training set είναι γνωστές οι τιμές του target και αυτό δίνεται ως είσοδος ελέγχου για την ανάπτυξη του μοντέλου μηχανικής μάθησης.

3.2 Εφαρμογή στατιστικών δεικτών στα δεδομένα

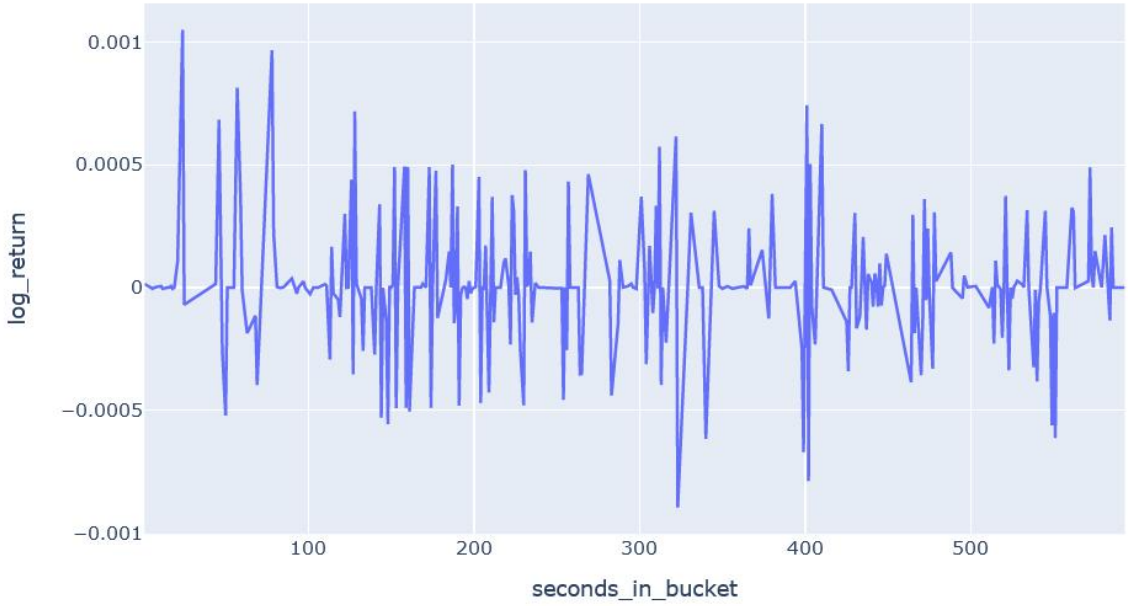
Στο 1ο κεφάλαιο ορίσαμε κάποιους στατιστικούς δείκτες οι οποίοι έχουν ενδιαφέρον όταν εφαρμοστούν στα δεδομένα. Στα παρακάτω γραφήματα φαίνεται η εξέλιξη στον χρόνο του Weighted Average Price και του Log return

Όπως παρατηρούμε το Log return τείνει να παίρνει τιμές γύρο από τον άξονα Χ και παρουσιάζει πυκνές εναλλαγές που αποτυπώνουν την αστάθεια. Η αποτύπωση της αστάθειας στο γράφημα του WAP είναι μεν παρούσα αλλά επιπλέον παρατηρούμε ότι υπάρχει και η πληροφορία του ύψους της τιμής.

Η τιμή του realized volatility αφορά όλο το διάστημα της χρονοθυρίδας και είναι και η τιμή στόχος του συστήματος.



Εικόνα 10. WAP του stock_id 0, για την χρονοθυρίδα 5



Εικόνα 11. Log return του stock_id 0, για την χρονοθυρίδα 5

3.3 Η γενικότερη προσέγγιση της υλοποίησης

3.3.1 Εφαρμογή του αλγορίθμου LGBM

Ένας βασικός λόγος επιλογής των δέντρων απόφασης για την εκπόνηση της εργασίας ήταν η ερμηνευσιμότητα της λειτουργίας τους. Ίσως θα μπορούσαν να βοηθήσουν δίνοντας μας πιο εύκολα πληροφορίες σχετικά με το πόσο πιο σημαντικά είναι κάποια δεδομένα ή στατιστικοί δείκτες από κάποια άλλα για την πρόβλεψη των τιμών του χρηματιστηρίου.

Έτσι λοιπόν επιλέχθηκε να χρησιμοποιηθούν δέντρα αποφάσεων και πιο ειδικά ο αλγόριθμος **LGBM**. Οι λόγοι που επιλέχθηκε ο συγκεκριμένος είναι ότι:

- μπορεί να εφαρμοστεί σε μεγάλη ποικιλία προβλημάτων,
- υποστηρίζει τόσο κατηγορικά (categorical) όσο και αριθμητικά δεδομένα ταυτόχρονα στο ίδιο μοντέλο
- έχει πολύ γρήγορη εκτέλεση
- είναι συνήθως εύκολο να ρυθμιστούν οι παράμετροι του ώστε να φέρει ένα αξιοπρεπές αποτέλεσμα

- ήταν πολύ δημοφιλής στην κοινότητα του Kaggle έχοντας αναδειχτεί νικητής πολλές φορές έναντι άλλων μορφών μηχανικής μάθησης

Ο αλγόριθμος LGBM μπορεί με χρήση της μεθόθου lgb.train να δεχτεί ως είσοδο ένα dataframe τύπου lgb με στοιχεία εκπαίδευσης και ένα με επαλήθευσης ώστε να εκπαιδεύσει ένα μοντέλο και είναι βασισμένος σε δέντρα απόφασης. Για την δημιουργία αυτού του τύπου dataframe αρκεί η χρήση της μεθόδου lgb.Dataset ώστε να μετατραπεί ένα υπάρχον pandas dataframe στην κατάλληλη μορφή. Σε αυτό το στάδιο μπορούν να οριστούν τα categorical features του μοντέλου αν υπάρχουν.

Έπειτα η συνάρτηση fit εφαρμόζει το μοντέλο που αναπτύχθηκε στο επιθυμητό dataset, είτε αυτό αφορά δεδομένα ενός παραγωγικού λογισμικού είτε σε κάποιο testing dataset.

Ιδιαίτερο ενδιαφέρον παρουσιάζει η συνάρτηση lgb.plot_importance η οποία μπορεί να οπτικοποιήσει την συνεισφορά του κάθε feature στο τελικό αποτέλεσμα με μορφή bar chart (vertical) και ένα σκορ στο κάθε ένα.

3.3.2 Επεξεργασία δεδομένων εκπαίδευσης και επαλήθευσης

Η επεξεργασία των δεδομένων σε ένα αλγόριθμο μηχανικής μάθησης θα πρέπει να είναι η ίδια τόσο για τα δεδομένα επαλήθευσης όσο και για αυτά της εκπαίδευσης αλλά τέλος και για τα δεδομένα του παραγωγικού μοντέλου. Συνοπτικά θα πρέπει να ακολουθηθεί η ίδια προεπεξεργασία για όλα τα δεδομένα.

Αναλύοντας τα δεδομένα διαπιστώθηκε ότι παράχθηκαν δειγματοληπτικά και ότι το μόνο χρονικό στοιχείο που διαθέτουμε είναι το seconds_in_bucket. Το time_id όπως αναλύθηκε και σε προηγούμενο κεφάλαιο δεν μπορεί να τοποθετηθεί σε αλληλουχία παρά μόνο να ομαδοποιήσει δεδομένα από άλλες μετοχές.

Για να λειτουργήσει ένας αλγόριθμος μηχανικής μάθησης θα πρέπει συνήθως να έχει ένα αυστηρά ορισμένο feature set. Άρα οι τιμές του seconds_in_bucket θα έπρεπε είτε να αξιοποιηθούν εξολοκλήρου, δηλαδή να δημιουργήσουμε 600 features ένα για κάθε δευτερόλεπτο ή να υλοποιηθεί thinning. Δηλαδή να υπάρξει μια επεξεργασία ώστε να ομαδοποιηθούν τα δεδομένα σε λιγότερα features.

Μετά από μια πρώτη ανάλυση των συχνοτήτων της δειγματοληψίας βρέθηκε ότι έχουμε μέσο όρο 30 παρατηρήσεις στο δεκάλεπτο για το trade book και 200 για το order book. Άρα αν δημιουργήσουμε παραπάνω από 30 buckets πολλές από τις τιμές που υπάρχουν στο trade_book θα ήταν μηδενικές. Όπως αναφέρθηκε και στο κεφάλαιο 2.6.3 οι απούσες τιμές μπορεί να δημιουργήσουν προβλήματα στην απόδοση του αλγορίθμου. Με βάση αυτά τα στοιχεία όλες οι δοκιμές έγιναν με ελάχιστο αριθμό χρονοθυρίδων 5 και μέγιστο 30. Σε πειραματισμούς για περισσότερες χρονοθυρίδες από 30 επαληθεύτηκε ότι δεν αυξάνεται σημαντικά η ακρίβεια πρόβλεψης.

Συγκεκριμένα στην καλύτερη εκδοχή του αλγορίθμου δεν παρατηρήθηκε αύξηση της απόδοσης που να δικαιολογεί την αύξηση του χρόνου εκτέλεσης:

No of buckets	25	50
Μέσο σφάλμα	0.207538	0.20683
Χρόνος εκτέλεση	9min 51s	23min 43s

Πίνακας 6. Πλήθος χρονοθυρίδων, μέσο σφάλμα και χρόνος εκτέλεσης

Ως μέθοδος άθροισης των δεδομένων δοκιμάστηκαν διάφορα εργαλεία, πιο συγκεκριμένα:

- η μέση τιμή,
- η τυπική απόκλιση,
- το άθροισμα,
- το ελάχιστο και
- το μέγιστο.

Και έγιναν και δοκιμές συνδυασμού ανάλογα με το feature. Τέλος παράχθηκαν νέα δεδομένα από τα πρωτογενή με βάση τους στατιστικούς δείκτες όπως περιγράφηκαν στο κεφάλαιο 1.4 :

- Bid/Ask Spread
- Weighted Averaged Price
- Log Returns

Έγιναν δοκιμές με όλους τους ανωτέρω δείκτες αλλά και με άλλους όπως η διαφορά μεταξύ χαμηλότερης τιμής προσφοράς και υψηλότερης τιμής ζήτησης.

3.3.3 Διαχωρισμός δεδομένων εκπαίδευσης και επαλήθευσης

Δύο τεχνικές που συνήθως χρησιμοποιούνται για αυτό τον σκοπό είναι το train/test split και ο αλγόριθμος kFold. Στην πρώτη περίπτωση αποφασίζουμε ποιο ποσοστό των δεδομένων θα χρησιμοποιηθεί για επαλήθευση και διαχωρίσουμε με τυχαία δειγματοληψία το σύνολο δεδομένων.

Για τον υπολογισμό του τελικού αποτελέσματος λαμβάνεται υπόψη ο μέσος όρος των τιμών της πρόβλεψης του κάθε ενός μοντέλου.

	Split train test	kFolds
Local result	0.2207	0.2219

Public dataset	0.2491	0.2453

Πίνακας 7, Train/test split έναντι kFolds για την δοκιμή της 2ης απόπειρας

Η διαφορά που παρατηρήθηκε δεν αφήνει χώρο για εξαγωγή συμπερασμάτων στην προκειμένη περίπτωση καθώς βρίσκεται πολύ κοντά στα όρια του σφάλματος. Παρόλα αυτά λόγω της ελαφρώς καλύτερης απόδοσης του kFolds στο κρυφό public dataset διατηρήθηκε μέχρι τα τελευταία στάδια ανάπτυξης.

3.3.4 Βελτιστοποίηση Hyperparameters

Στο LGBM, η πιο σημαντική παράμετρος για τον έλεγχο της δομής του δέντρου είναι το **num_leaves**. Όπως υποδηλώνει το όνομα, ελέγχει τον αριθμό των φύλλων απόφασης. Το φύλλο απόφασης ενός δέντρου είναι ο κόμβος όπου λαμβάνεται η «πραγματική απόφαση».

Το επόμενο είναι το **max_depth**. Όσο υψηλότερο είναι το max_depth, τόσο περισσότερα επίπεδα έχει το δέντρο, γεγονός που το καθιστά πιο περίπλοκο και επιρρεπές σε overfitting.

Το num_leaves βρίσκεται σε συνάρτηση με το max_depth όσο αφορά τα αποτελέσματα και συνήθως τίθεται μεταξύ 3-12. Υπάρχει ένας απλός τύπος που δίνεται στην τεκμηρίωση LGBM - το μέγιστο όριο στα num_leaves θα πρέπει να είναι 2^(max_depth). Αυτό σημαίνει ότι η βέλτιστη τιμή για το num_leaves βρίσκεται εντός του εύρους (2^3, 2^12).

Ωστόσο, το num_leaves επηρεάζει τη μάθηση στο LGBM περισσότερο από το max_depth. Αυτό σημαίνει ότι πρέπει αρχικά να καθορίσουμε ένα πιο συντηρητικό εύρος αναζήτησης.

Μια άλλη σημαντική δομική παράμετρος για ένα δέντρο είναι το **min_data_in_leaf**. Με απλά λόγια, το min_data_in_leaf καθορίζει τον ελάχιστο αριθμό παρατηρήσεων που πληρούν τα κριτήρια απόφασης σε ένα φύλλο. Το μέγεθός του σχετίζεται επίσης με το αν παρουσιάζεται overfitting ή όχι.

Για παράδειγμα, εάν το φύλλο απόφασης ελέγχει εάν ένα χαρακτηριστικό είναι μεγαλύτερο από, ας πούμε, 13 — η ρύθμιση του min_data_in_leaf σε 100 σημαίνει ότι θέλουμε να αξιολογήσουμε αυτό το φύλλο μόνο εάν τουλάχιστον 100 παρατηρήσεις είναι μεγαλύτερες από 13. Η βέλτιστη τιμή για το min_data_in_leaf εξαρτάται από τον αριθμό των δειγμάτων εκπαίδευσης και του num_leaves.

Μια κοινή πρακτική για την επίτευξη υψηλότερης ακρίβειας είναι η χρήση πολλών δέντρων αποφάσεων και η μείωση του ρυθμού εκμάθησης. Με άλλα λόγια, βρίσκουμε την χρυσή τομή μεταξύ **n_estimators** και **learning_rate**.

Το n_estimators ελέγχει τον αριθμό των δέντρων απόφασης, ενώ το Learning_rate είναι η παράμετρος μεγέθους βήματος της gradient descent. Σύνολα όπως το LGBM δημιουργούν δέντρα σε επαναλήψεις και κάθε νέο δέντρο χρησιμοποιείται για να διορθώσει τα «λάθη» των προηγούμενων δέντρων. Αυτή η προσέγγιση είναι γρήγορη και ισχυρή και επιρρεπής σε overfitting. Αυτός είναι ο λόγος για τον οποίο τα gradient boost ensembles έχουν μια

παράμετρο Learn_rate που ελέγχει την ταχύτητα εκμάθησης. Οι τυπικές τιμές βρίσκονται εντός 0,01 και 0,3, αλλά είναι δυνατόν να προχωρήσουμε πέρα από αυτές, ειδικά προς το 0.

Το LGBM έχει επίσης σημαντικές παραμέτρους κανονικοποίησης. Τα **lambda_l1** και **lambda_l2** (L1, L2), όπως και τα **reg_lambda** και **reg_alpha** καθορίζουν την κανονικοποίηση του XGBoost. Η βέλτιστη τιμή για αυτές τις παραμέτρους είναι πιο δύσκολο να βρεθεί επειδή το μέγεθός τους δεν συσχετίζεται άμεσα με το overfitting. Ωστόσο, ένα καλό εύρος αναζήτησης είναι (0, 100) και για τα δύο.

Στη συνέχεια, έχουμε **min_gain_to_split**, παρόμοιο με το gamma του XGBoost. Ένα συντηρητικό εύρος είναι (0, 15). Μπορεί να χρησιμοποιηθεί ως επιπλέον κανονικοποίηση σε περιπτώσεις μεγάλου πλήθους παραμέτρων.

Τέλος, έχουμε **bagging_fraction** και **feature_fraction**. Το bagging_fraction παίρνει μια τιμή εντός (0, 1) και καθορίζει το ποσοστό των δειγμάτων που θα χρησιμοποιηθούν για την εκπαίδευση κάθε δέντρου (ακριβώς όπως το subsample στο XGBoost). Το **feature_fraction** καθορίζει το ποσοστό των χαρακτηριστικών προς δειγματοληψία κατά την εκπαίδευση κάθε δέντρου. Έτσι, παίρνει επίσης μια τιμή μεταξύ (0, 1). Με αυτό τον τρόπο μπορούμε να εισάγουμε τυχαιότητα ώστε να μειώσουμε το φαινόμενο του overfitting.

3.4 Βελτιστοποίηση Χρόνου Εκτέλεσης

Για την βελτιστοποίηση του χρόνου εκτέλεσης τους αλγορίθμου επιστρατεύθηκαν δύο τεχνικές: η παράλληλη επεξεργασία με χρήση **threads** και η χρήση της τεχνολογίας **Cuda** της κάρτας γραφικών.

Για τις ανάγκες του feature engineering και του binning χρησιμοποιήθηκε ο επεξεργαστής και πιο συγκεκριμένα η βιβλιοθήκη της Python joblib που επιτρέπει να τρέξουμε μια συνάρτηση του κώδικα σε πολλαπλά ανεξάρτητα threads και έπειτα να συλλέξουμε τα αποτελέσματά τους σε ένα iterable. Από τις μετρήσεις που έγιναν παρατηρούμε ότι σε 6 πυρήνες / 12 threads ο αλγόριθμος τρέχει 6.5 φορές πιο γρήγορα

	Feature Engineering	secs
Parallel (12 Threads / 6 cores)	2min 37s	157
Sequential	17min 7s	1027
	Model Training	secs
GPU 3070 ti	2min 4s	124
CPU (5600x)	4min 25s	265
Rows	428932	
Colums	384	

Πίνακας 8, Σύγκριση σειριακής εκτέλεσης/παραλληλίας και μαζικής παραλληλίας cuda για την δοκιμή με τα εκτενή features

Στην περίπτωση της ανάπτυξης του μοντέλου έγινε αξιοποίηση της παραμέτρου παράλληλης επεξεργασίας της βιβλιοθήκης lightgbm για παράλληλη επεξεργασία με τον επεξεργαστή αλλά και με μαζική παράλληλη επεξεργασία με χρήση της κάρτας γραφικών. Από τις μετρήσεις που έγιναν παρατηρούμε ότι ο αλγόριθμος τρέχει 2.13 φορές πιο γρήγορα χρησιμοποιώντας την κάρτα γραφικών.

Τα οφέλη δεν είναι καθόλου αμελητέα από δύο απόψεις, από τη μία μέσα σε ένα χρονικό διάστημα μπορεί κανείς να τρέξει 2 πειράματα στην διάρκεια του ενός, αυτό έχει σαν συνέπεια να μπορούν να εξερευνηθούν περισσότερα μονοπάτια για την επίλυση του προβλήματος. Από την άλλη είναι τόσο μεγάλοι οι χρόνοι εκτέλεσης που ένα σφάλμα κατά την ανάπτυξη μπορεί να εντοπιστεί πολύ συντομότερα μειώνοντας τη σπατάλη χρόνου. Οτιδήποτε μπορεί να βοηθήσει στην καλή διαχείριση του χρόνου είναι καλό να αξιοποιηθεί γιατί μπορεί να επηρεάσει την ποιότητα του τελικού αποτελέσματος.

3.5 Μια “αφελής” προσέγγιση [2]

Μέσο σφάλμα: 0.341

Αριθμός χρονοθυρίδων: ~230 (όσες και οι παρατηρήσεις)

Είναι κοινώς αποδεκτό σχετικά με την αστάθεια ότι τείνει σε ένα βαθμό να αυτό-συσχετίζεται. Η πιο απλή προσέγγιση που μπορεί να ακολουθήσει κανείς για την επίλυση του προβλήματος είναι απλώς να χρησιμοποιήσει ως πρόβλεψη την τιμή του προηγούμενου δεκαλέπτου στο επόμενο. [2]

Ενδεικτικά αξίζει να αναφέρουμε ότι το σφάλμα της αφελούς προσέγγισης είναι στο 34% και του νικητή του διαγωνισμού ανέρχεται σε 18% ενώ το καλύτερο αποτέλεσμα του κώδικα της παρούσας εργασίας είναι στο 21%. Φυσικά δεν είναι σωστό να τα συγκρίνουμε άμεσα καθώς ο υπολογισμός του μέσου σφάλματος κρύβει περιόδους σταθερών τιμών όπου η αφελής προσέγγιση μοιάζει να είναι αποδοτική και αστάθειας των τιμών όπου η αφελής προσέγγιση σφάλει με μεγάλο ποσοστό.

Στο παρόν πρόβλημα όμως μπορεί να μας δώσει μια σημαντική πληροφορία, μπορεί να λειτουργήσει ως ένα σημείο αναφοράς στην αξιολόγηση του υπό ανάπτυξη αλγορίθμου. Οποιοδήποτε αποτέλεσμα κοντά σε αυτό της συγκεκριμένης μεθόδου είναι μη αποδεκτό αφού δεν βελτιώνει την εικόνα μας για το χρηματιστήριο.

3.6 Μια πρώτη προσέγγιση

Μέσο σφάλμα: 0.28789

Αριθμός χρονοθυρίδων: 9

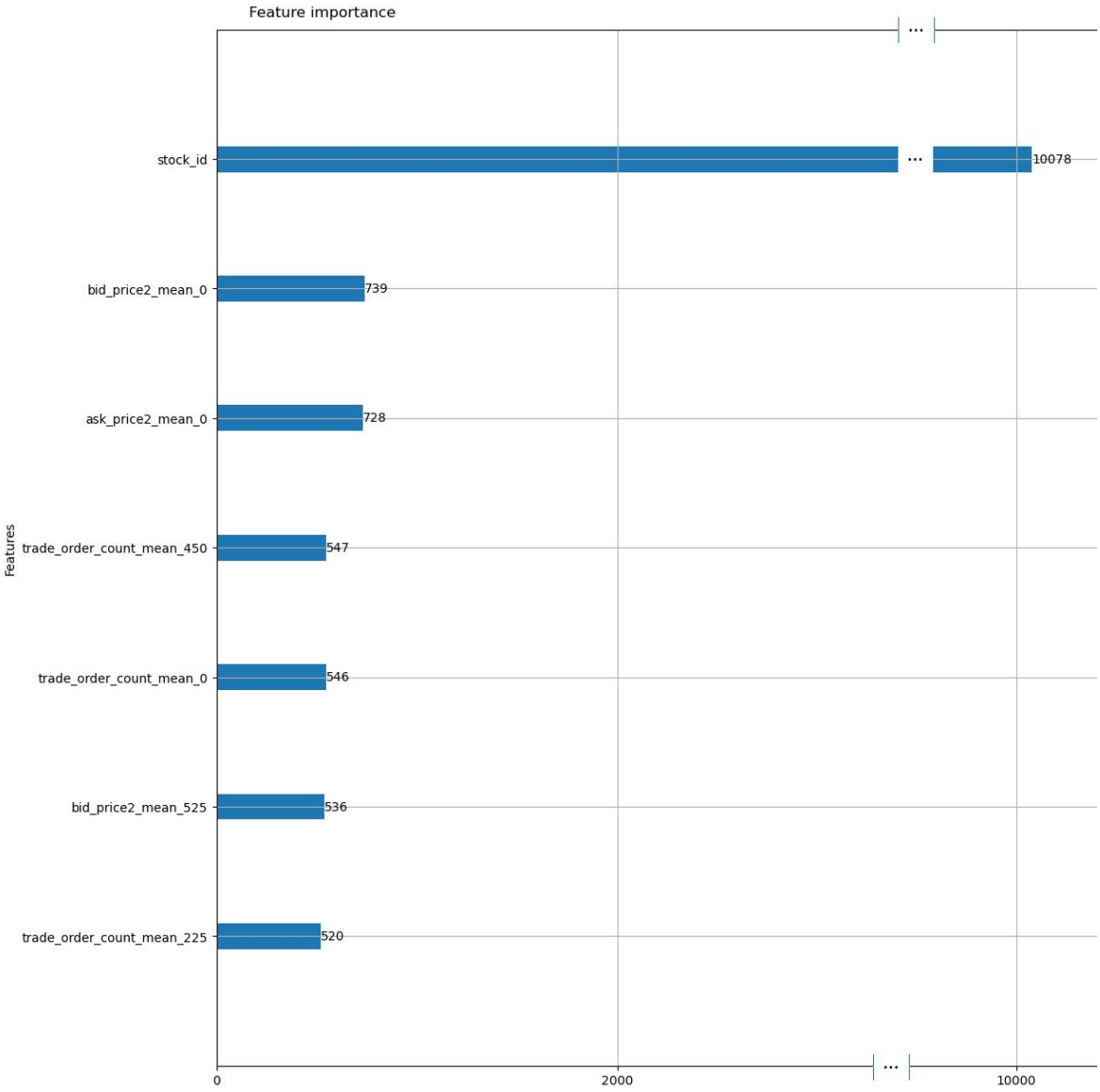
Features: εφεξής [raw_features]=(order_book: bid_price1, ask_price1, bid_price2, ask_price2, bid_size1, ask_size1, bid_size2, ask_size2, trade_book: price, size, order_count

Στην πρώτη απόπειρα για την επίλυση του προβλήματος χρησιμοποιήθηκαν τα ανεπεξέργαστα δεδομένα, που δόθηκαν από τους διοργανωτές όπως περιγράφηκαν προηγουμένως.

Η μέθοδος ομαδοποίησης ήταν ο υπολογισμός του μέσου όρου των τιμών που ανήκαν σε κάθε μια χρονοθυρίδα.

Σε αυτό το σημείο αξίζει να γίνει μια σύγκριση αυτών των δύο πρώτων και τελείως διαφορετικών μεταξύ τους μεθόδων. Η μαθηματική σχέση της αστάθειας προκύπτει από τα πρωτογενή δεδομένα έχοντας περάσει από δύο στάδια μαθηματικών υπολογισμών. Από τη μία στην αφελή προσέγγιση η τιμή στόχος υπολογίζεται με γνώσεις στατιστικής μεταφέροντας τις τιμές του προηγούμενου δεκαλέπτου από την άλλη στην δεύτερη περίπτωση η μέθοδος υπολογισμού της, δηλαδή ο ορισμός της ως μαθηματική έννοια είναι παντελώς άγνωστη στο μοντέλο της μηχανικής μάθησης. Παρότι σε καμία περίπτωση δεν έχει αποκρυπτογραφηθεί από το μοντέλο είναι εντυπωσιακό το ότι το αποτέλεσμα που “εμπειρικά” παρέχει είναι πιο κοντά στον στόχο από την “αφελή” προσέγγιση.

Ιδιαίτερο ενδιαφέρον προκαλεί η παρατήρηση ότι τα πλέον σημαντικά features είναι οι τιμές ζήτησης και προσφοράς του 2ου επιπέδου του order book. Το 2ο επίπεδο είναι πιο κοντά στην εικόνα των μελλοντικών συναλλαγών από το πρώτο επίπεδο τιμών σε σχέση με το 1ο επίπεδο. Αυτό διότι σε ένα ρευστό ανταλλακτήριο οι τιμές του 1ου επιπέδου δεν μένουν εκεί για πολύ είναι σχεδόν τρέχουσες. Επίσης αξιοσημείωτο είναι ότι και οι δύο έχουν σχεδόν τον ίδιο βαθμό συμμετοχής που πιθανά υποδηλώνει ότι η ζήτηση και η προσφορά έχουν τον ίδιο ρυθμό μεταβολής.



Εικόνα 12. Βαθμός συμμετοχής features στην πρώτη προσπάθεια

3.7 Πρόγνωση με το time_id

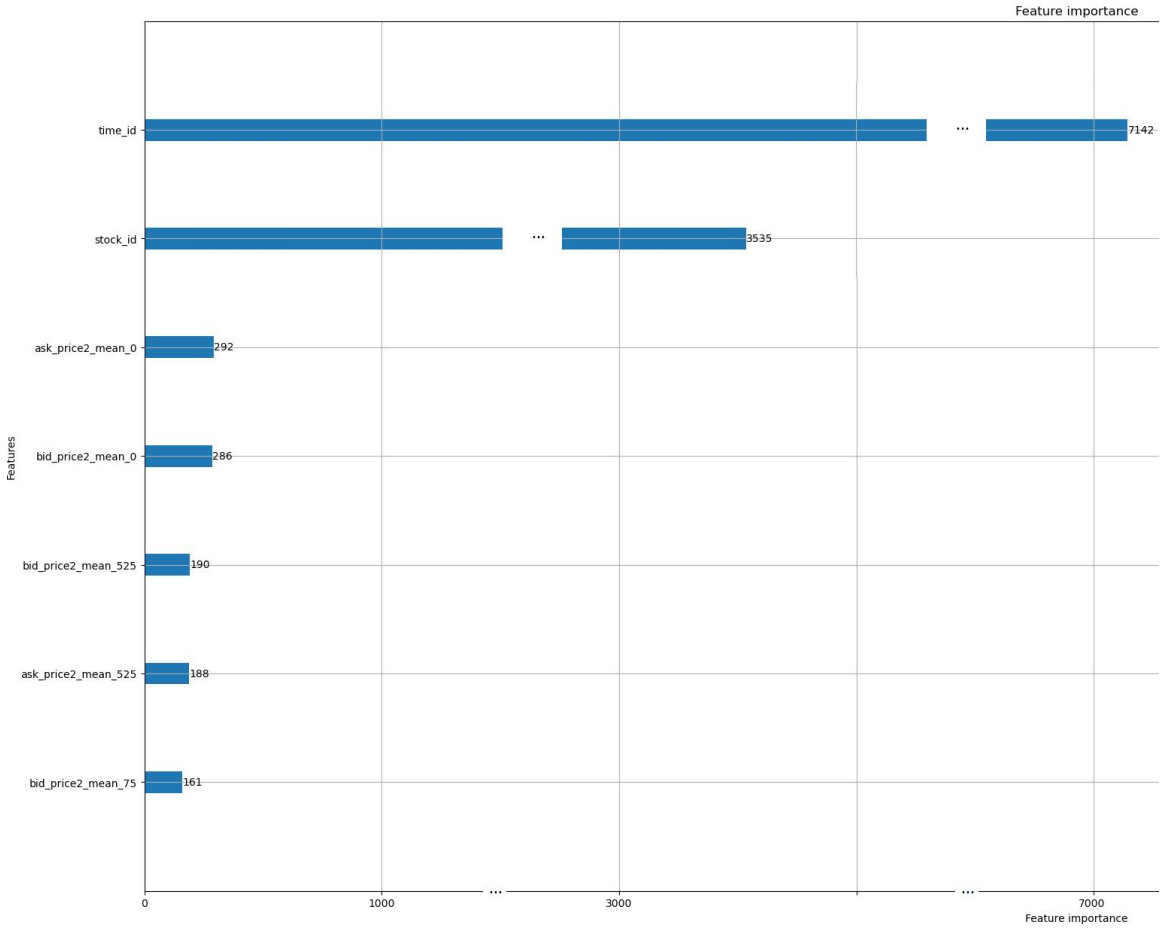
Μέσο σφάλμα: 0.2354

Αριθμός χρονοθυρίδων: 9

Features: [raw_features]

Από τον ορισμό του προβλήματος ήταν σαφές ότι η συνεισφορά του time_id θα ήταν feature αμφιλεγόμενη. Στις μόνες περιπτώσεις που φάνηκε να έχει κάποιο αντίκτυπο ήταν στα αρχικά στάδια όταν τα προβλέψεις ήταν στο 27%, ποσοστό κοντά σε αυτό της αφελούς προσέγγισης.

Η υπόθεση το να εκμαιευτεί η πραγματική αλληλουχία των χρονοθυρίδων από τον αλγόριθμο θεωρείται απίθανη καθώς δεν παρατηρήθηκε βελτίωση κατά τις δοκιμές με την χρήση της. Επίσης θα προϋπέθετε να έχει γίνει κάποια κακή επιλογή τυχαιοποίησης από τους διοργανωτές και δεν επισημάνθηκε κάτι σχετικό από κάποιον άλλο συμμετέχοντα. Ακόμη όμως και αν ήταν δυνατή τα κενά ανάμεσα στις χρονοθυρίδες δημιουργούν ένα επιπλέον πρόβλημα στην αξιοποίηση του εν λόγω στοιχείου.



Εικόνα 13. Βαθμός συμμετοχής features στην πρώτη προσπάθεια με προσθήκη του time_id

Από την μια πλευρά μπορεί να υποτεθεί ότι εισάγοντας το ο αλγόριθμος μηχανικής μάθησης να εξάγει συμπεράσματα για την συνολική εικόνα κατά μέσο όρο του χρηματιστηρίου. Μάλιστα ενδιαφέρον έχει ότι έχει ξεπεράσει σε συμμετοχή στο αποτέλεσμα το stock_id. Από την άλλη αυτά τα δεδομένα μπορούν να προστεθούν και προστέθηκαν στην παρούσα υλοποίηση με άλλον τρόπο προσφέροντας καλύτερα αποτελέσματα.

3.8 Η δεύτερη απόπειρα

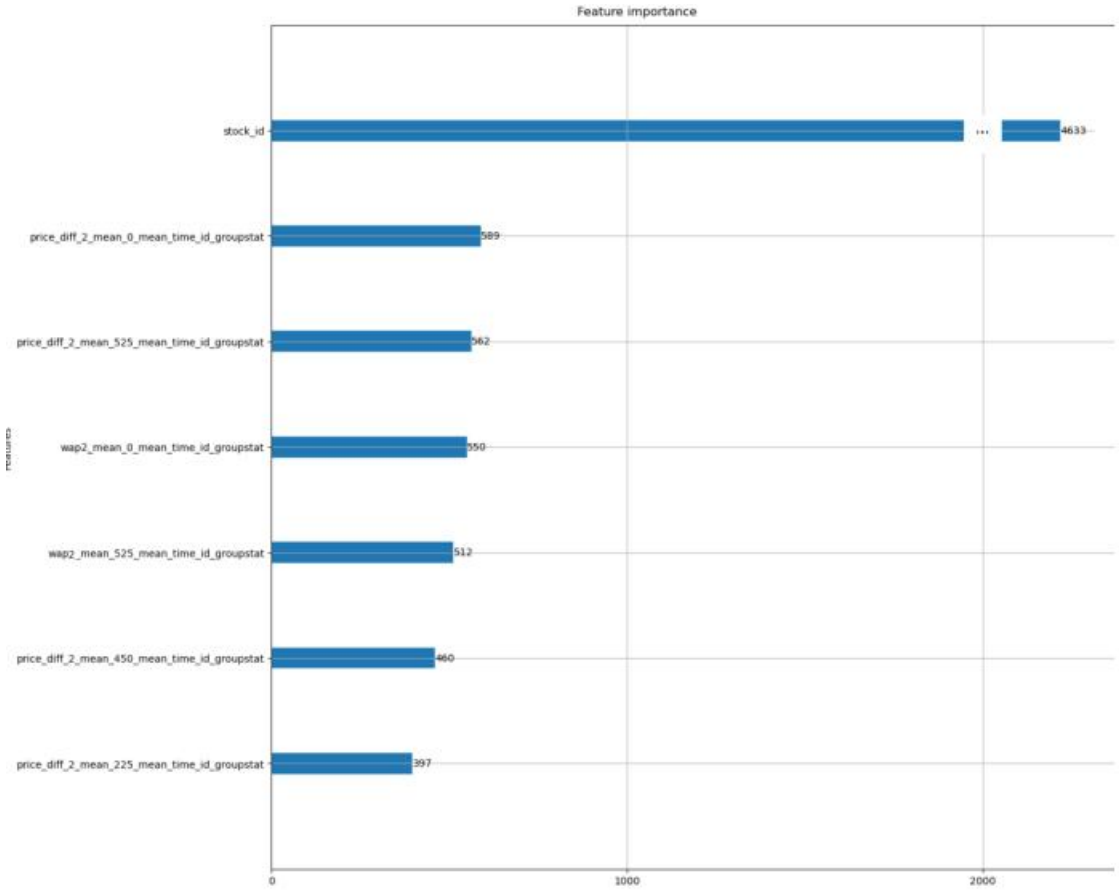
Μέσο σφάλμα: 0.2207

Αριθμός χρονοθυρίδων: 9

Features: [raw_features], wap2, price_diff_2

Σε αυτό το σημείο και με δεδομένο το παραπάνω σκεπτικό αποφασίστηκε να προστεθούν συνολικά στατιστικά στοιχεία για όλες τις μετοχές ανά χρονοθυρίδα (time_id). Διαισθητικά επιλέχθηκε η τιμή όπως αποτυπώνεται στο trade book (price) και ο όγκος συναλλαγών (size).

Πράγματι υπήρξε μια αξιοσημείωτη βελτίωση στις τιμές σφάλματος και η μέθοδος των συνολικών στατιστικών για όλες τις μετοχές διατηρήθηκε στα επόμενα στάδια της ανάπτυξης. Μάλιστα τα ομαδοποιημένα στατιστικά φαίνεται να έχουν επικρατήσει των μεμονωμένων και μεταξύ τους επικρατούν τα price_difference των wap2.



Εικόνα 14. Βαθμός συμμετοχής features στην δεύτερη προσπάθεια

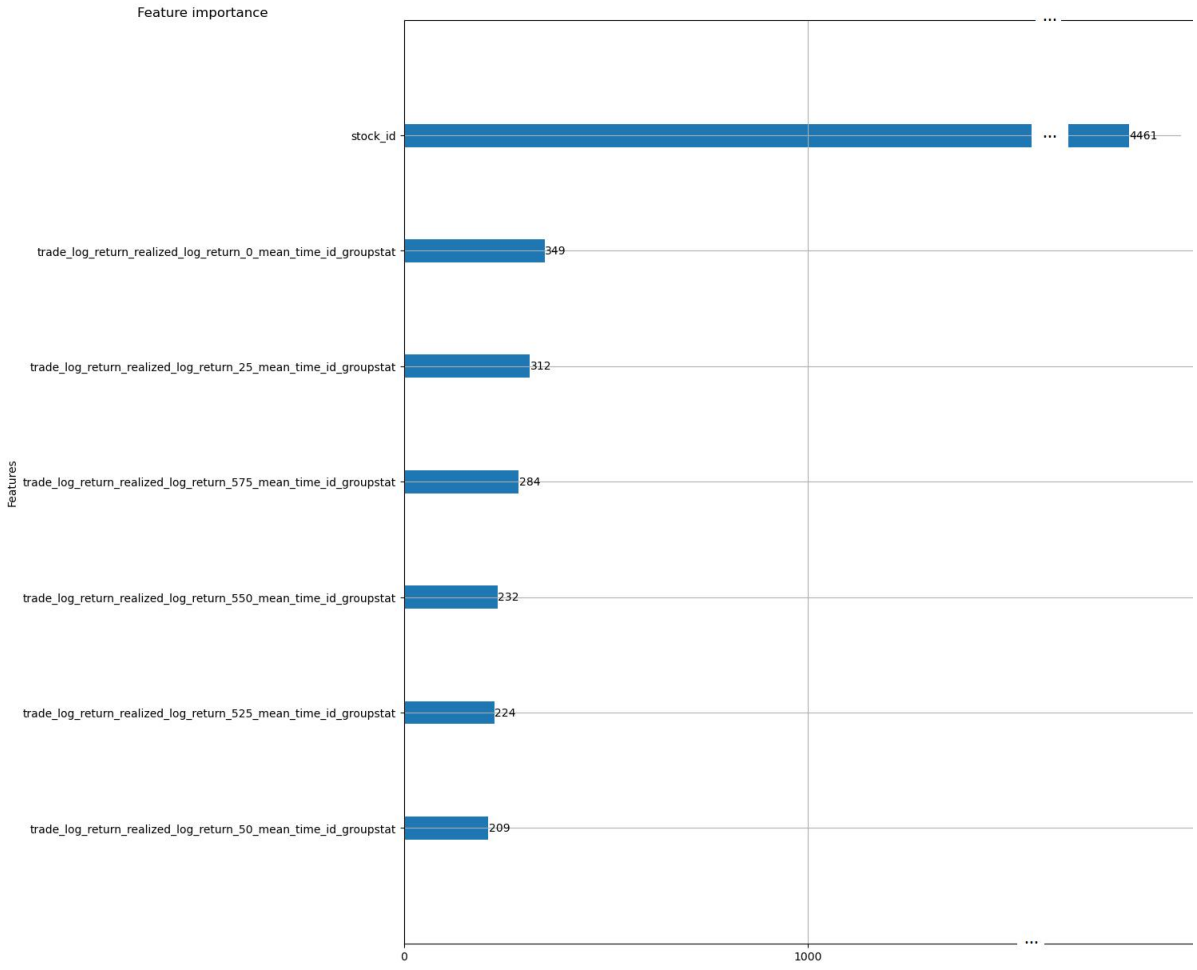
3.9 Περισσότερες χρονοθυρίδες και μείωση feature set

Μέσο σφάλμα: 0.2132

Αριθμός χρονοθυρίδων: 25

Features: wap2, price_diff, price_spread, order_size, | log_return, order_count

Αφαιρώντας τα πρωτογενή features και αυξάνοντας τις χρονοθυρίδες παρατηρείται ακόμα μεγαλύτερη βελτίωση ενώ παράλληλα ο χρόνος εκτέλεσης παραμένει ανεκτός. Και σε αυτή την δοκιμή τα ομαδοποιημένα στατιστικά φαίνεται να έχουν επικρατήσει των μεμονωμένων και μεταξύ τους επικρατούν τα log_returns του price του trade book της αρχής και του τέλους του δεκαλέπτου.



Εικόνα 15. Βαθμός συμμετοχής features στην δοκιμή με περισσότερες χρονοθυρίδες και μείωση feature set

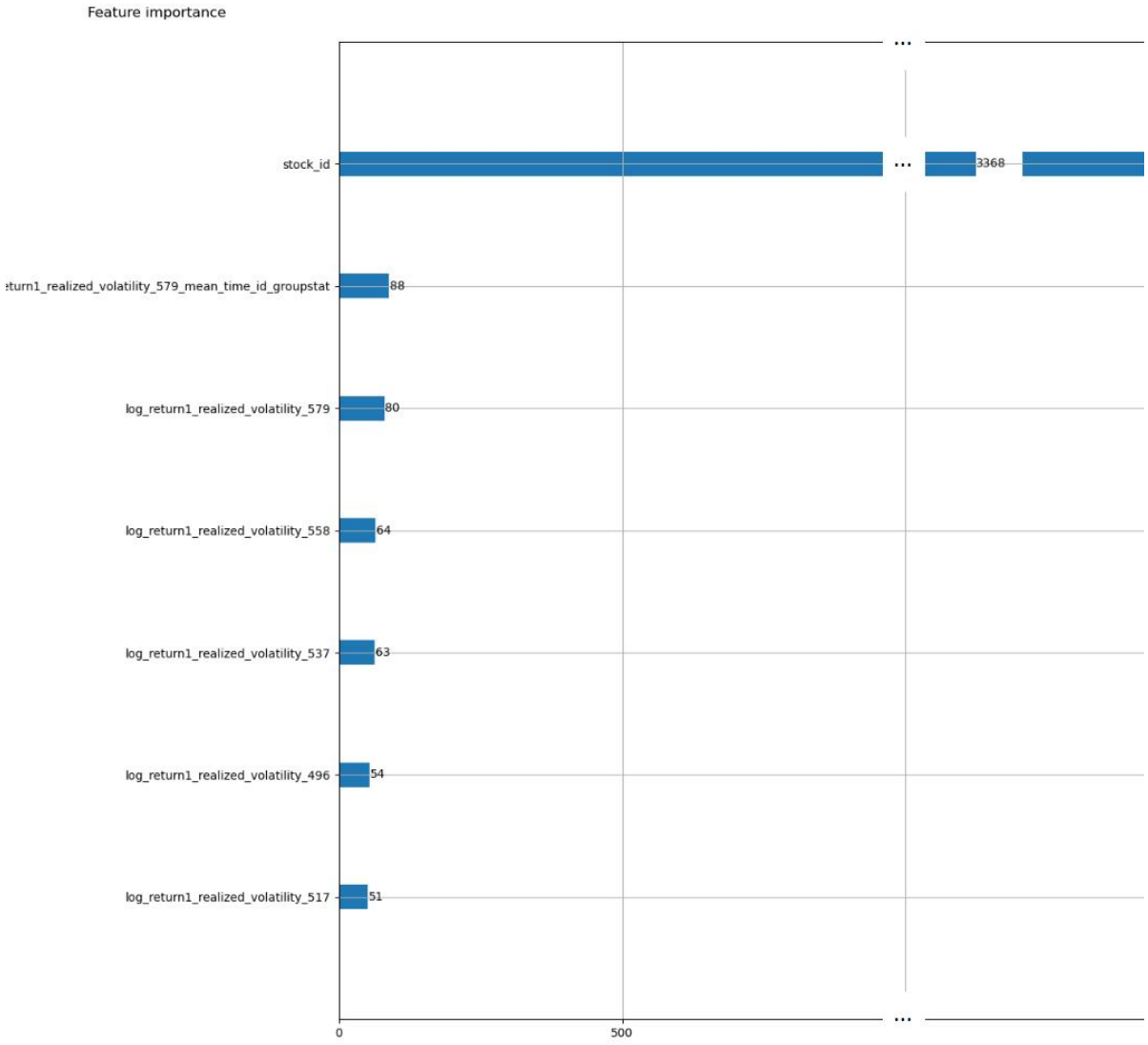
3.10 Εκτενή features

Μέσο σφάλμα: 0.2004

Αριθμός χρονοθυρίδων: 30

Στα επόμενα βήματα έγιναν δοκιμές με περισσότερα time buckets αλλά και ολοένα και περισσότερα features. Το σκεπτικό ήταν να παραχθούν όσο περισσότερα features γίνεται με χρήση στατιστικών δεικτών και η μετέπειτα αφαίρεσή τους ανάλογα με την συνεισφορά τους στο αποτέλεσμα.

Ένα στοιχείο που διαφαίνεται από τα αποτελέσματα αυτής της προσέγγισης είναι ότι ακολουθώντας μια λογική brute force μπορεί να φέρει βελτιώσεις στον αλγόριθμο με την αντίστοιχη βέβαια ποινή στον χρόνο εκτέλεσης. Εδώ αξίζει να αναφέρουμε ότι αυτοί οι αλγόριθμοι έχουν αναπτυχθεί για να αντιμετωπίζουν πολύ μεγαλύτερο αριθμό δεδομένων οπότε ακόμη και αυτός ο αυξημένος αριθμός features που προέκυψε είναι μηδαμινός για τις προδιαγραφές τους.



Εικόνα 16. Βαθμός συμμετοχής features στην δοκιμή με περισσότερες χρονοθυρίδες και μείωση feature set

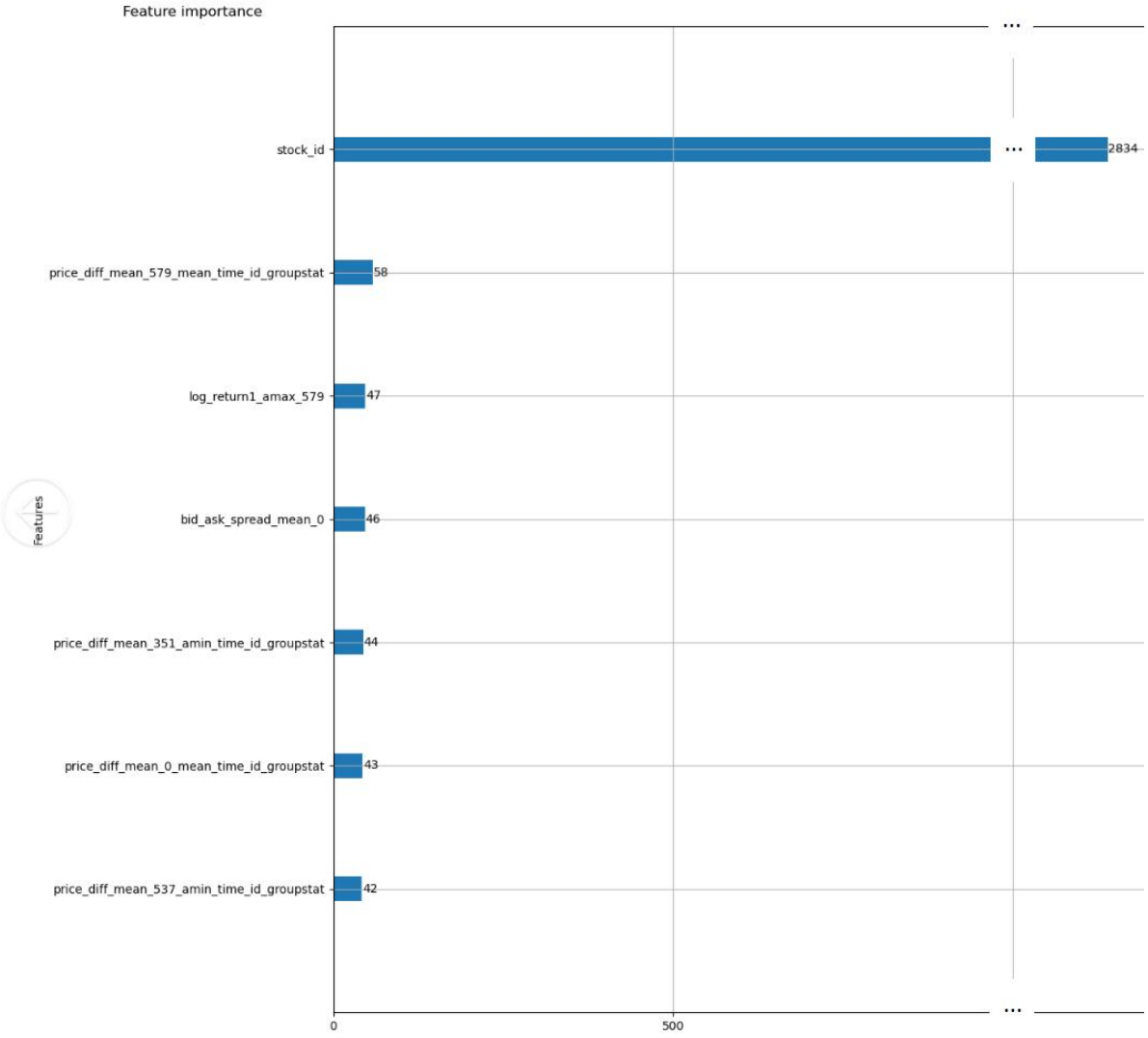
3.11 Η τελική μορφή

Μέσο σφάλμα: 0.2029

Αριθμός χρονοθυρίδων: 30

Στο τελικό στάδιο επιλέχθηκε να γίνει thinning στα δεδομένα εισόδου του LGBM αφαιρώντας features που μοιάζουν να μην προσθέτουν επιπλέον πληροφορία στον αλγόριθμο.

Παραδείγματος χάρη το weighted average price περιέχει την ίδια πληροφορία στη βάση του με το log return υπολογισμό του, αυτό που αλλάζει είναι τα αριθμητικά του χαρακτηριστικά. Κατά τα άλλα υπολογίζεται ορίζοντας κάποιες σχέσεις με όλα τα πρωτογενή στοιχεία: ask_price1,bid_price1, ask_price_2,bid_size_2. Το ποιο από τα δύο λειτουργεί καλύτερα είναι θέμα που αξίζει διερεύνησης αλλά η ταυτόχρονη χρήση τους μοιάζει με πλεονασμό.



Εικόνα 17. Βαθμός συμμετοχής features στην δοκιμή με περισσότερες χρονοθυρίδες και μείωση feature set

Από όλες τις δοκιμές και την αξιοποίηση του private dataset που ήταν διαθέσιμο για online δοκιμές παρατηρήθηκε ότι όσο μεγαλύτερη είναι η διαφορά μεταξύ σφάλματος training και σφάλματος validation τόσο χειρότερα γενικεύει ο αλγόριθμος.

Με αυτό το σκεπτικό επιλέχθηκε ένα υποσύνολο των features της προηγούμενης προσέγγισης που να επιτρέπει παράλληλα στον αλγόριθμο να γενικεύσει. Το αποτέλεσμα ήταν θετικό αφού έδωσε στον αλγόριθμο μεγαλύτερη ικανότητα γενίκευσης, όπως παρατηρούμε από την διαφορά μεταξύ training msrpe και validation rmspe στην εκτενή καταγραφή που ακολουθεί. Επιπρόσθετα ο χρόνος εκτέλεσης μειώθηκε σημαντικά.

3.12 Αναλυτική καταγραφή δοκιμών

Παρακάτω μπορείτε να βρείτε μια καταγραφή στιγμιοτύπων της εφαρμογής καθ' όλη την εξέλιξη της. Τα πιο σημαντικά έχουν περιγραφεί στα ανωτέρω κεφάλαια. Στο συνοδευτικό υπολογιστικό φύλλο περιλαμβάνονται και τα γραφήματα του βαθμού συμμετοχής των features για κάθε δοκιμή.

Name	Time buckets	features	Columns	train (rmspe)	score (validation rmspe)	feature engineering time	model training time	Total time (s)
Naive	~230 (all)	null	n/a	n/a	0.341	0	n/a	fast
1st_attempt_raw_data	200	[raw_features]	2203	0.244869	0.272660	20min 6s	2min 38s	1244
>>	30	>>	322	0.211449	0.272464	45.2 s	2min 25s	190
1st_attempt_raw_time_id	30	[raw_features] + time_id	323	0.167892	0.235497	46.7 s	4min 24s	310
only_realized_volatility	200	realized_voltility (on: log_return of wap1)	203	0.228246	0.247682	17min 15s	41.6 s	1076
>>	30	>>	33	0.231564	0.243114	2min 59s	14.5 s	193
2nd_attempt	30	[raw_features] + wap2, price_diff2 + groupstats	612	0.163802	0.231193	1min 59s	3min 49s	348
>>	9	>>	172	0.179622	0.232634	26.3 s	2min 51s	197
2nd_attempt_feature_reduction	30	wap2, price_diff, price_spread, order_size | log_return, price, size, order_count + groupstats	468	0.195476	0.211989	3min 44s	2min 26s	370
>>	9	>>	132	0.183557	0.219367	1min 12s	1min 50s	182

Multiple_engineered_features	30	wap1', 'wap2', 'log_return1', 'log_return2', 'wap_balance', 'price_diff', 'total_size', 'ask_size', 'bid_size', 'bid_ask_spread', 'total_volume', 'volume_imbalance', 'ask_volume', 'bid_volume', 'order_size' | 'trade_log_return', 'price', 'size', 'order_count' | 'log_return1_mean', 'log_return2_mean', 'price_diff_mean'	1251	0.167354	0.200487	12min 15s	3min 54s	969
>>	9	>>	348	0.160964	0.200889	3min 28s	2min 10s	338
Thinned_features	30	log_return1', 'log_return2', 'wap_balance', 'price_diff', 'total_size', 'bid_ask_spread' | 'trade_log_return', 'price', 'size', 'order_count' | 'trade_log_return_mean', 'log_return2_mean', 'price_diff_mean'	990	0.177348	0.202905	4min 11s	3min 3s	434
>>	9	>>	276	0.171445	0.205422	1min 5s	1min 38s	163

[raw_features]=	order_book: bid_price1, ask_price1, bid_price2, ask_price2, bid_size1, ask_size1, bid_size2, ask_size2 trade_book: price, size, order_count
Featues fomat	<order_book> | <trade_book> | <groupstats>

Πίνακας 9, Καταγραφή όλων των δοκιμών εκτέλεσης

Κεφάλαιο 4ο

4 Συμπεράσματα

Το δοθέν πρόβλημα ήταν πολύ δύσκολο από την φύση του, με σημαντικότερο παράγοντα την τυχαία διάταξη των χρονοθυρίδων αλλά και το μέγεθος του ενός δεκαλέπτου και συνεπώς το μικρό πλήθος δεδομένων στην κάθε χρονοθυρίδα. Μετά από πλήθος πειραματισμών διαπιστώθηκε ότι εφαρμόζοντας τις καλές πρακτικές σχετικά με την ανάπτυξη αλγορίθμων μηχανικής μάθησης μπορούν να βελτιωθούν σημαντικά οι προβλέψεις.

Αν και τα κέρδη της ακρίβειας σε ορισμένες περιπτώσεις μπορεί να μοιάζουν μικρά, η αντικειμενική δυσκολία του προβλήματος θέτει μια άλλη κλίμακα αξιολόγησης όπου ακόμη και μικρές βελτιώσεις μπορούν να θεωρηθούν ουσιαστικές. Αυτό επιβεβαιώνεται από την μικρή διαφορά στην ακρίβεια μεταξύ του σκορ των νικητών και της παρούσας εργασίας (3%) η οποία βρέθηκε στα ανώτερα στρώματα της μέσης της κατάταξης.

Ένα σημαντικό στοιχείο είναι η άμεση εξάρτηση των hyperparameters με το feature set. Οι πειραματισμοί είχαν σαν σημείο εκκίνησης την αλλαγή του feature set και σε κάθε τέτοια αλλαγή για να εξαχθεί ένα ασφαλές συμπέρασμα έπρεπε να επαναπροσδιοριστούν τα hyperparameters προτού προκύψει κάποια βελτίωση. Η επιλογή χρήσης δέντρου αποφάσεων προσέφερε εγγενώς την πληροφορία του πόσο σημαντικό είναι ένα feature για την διαμόρφωση του αποτελέσματος. Σαν συνέπεια χρησιμοποιήθηκε σαν οδηγός γαι το feature engineering και επίσης ήταν δυνατό να γίνει thinning που πέρα από τον χρόνο εκτέλεσης βελτίωσε ελαφρά και την ακρίβεια.

Μια ενδιαφέρουσα σκοπιά ήταν το αν θα μπορούσε ο αλγόριθμος να μας υποδείξει ποιοι στατιστικοί δείκτες των δεδομένων των μετοχών είναι οι πιο κρίσιμοι για την λήψη μιας απόφασης αγοράς η πώλησης. Από τα αποτελέσματα των δοκιμών δεν διαπιστώθηκε κάτι τέτοιο. Παρατηρήθηκε ότι το καλύτερο μοντέλο έδειχνε προτίμηση στο feature του στόχου για να προβλέπει το αποτέλεσμα. Τέλος κατά την εκτέλεση όμως με μόνο ανεπεξέργαστα δεδομένα (raw) φάνηκε κάποια από αυτά να διαδραματίζουν προεξάρχοντα ρόλο.

Τέλος μια άλλη πλευρά που διερευνήθηκε ήταν η βελτιστοποίηση του χρόνου εκτέλεσης με παραλληλισμό και μαζικά παράλληλη επεξεργασία με GPU που φάνηκε να έχει μόνο θετικά οφέλη. Στην περίπτωση παράλληλης επεξεργασίας με επεξεργαστή κατά την προ επεξεργασία των δεδομένων παρατηρήθηκε μια σχεδόν γραμμική μείωση του χρόνου αντιστρόφως ανάλογη του αριθμού των πυρήνων. Σε αυτή της μαζικής παράλληλης διαπιστώθηκε μείωση πάνω από το μισό στον χρόνο εκτέλεσης. Η δε υλοποίηση και στις δύο περιπτώσεις ήταν σημαντικά εύκολη (σύνολο 5 γραμμές κώδικα) οπότε ο προγραμματιστής κερδίζει χρόνο που μπορεί να αφιερώσει στην βελτίωση του μοντέλου.

5 Βιβλιογραφικές Πηγές

[1]. **Aurélien Géron.** *Hands-On Machine Learning with Scikit-Learn*, *Keras, and TensorFlow: 2nd Edition,* O'Reilly Media, Inc, 2019.
[2]. **Jiashen Liu.** *Introduction to financial concepts and data. Amsterdam: Optiver Kaggle Competition*, 2021.
[3]. **Guolin Ke Qi Meng Thomas Finely Taifeng Wang Wei Chen Weidong Ma Qiwei Ye Tie-Yan Liu.** *LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Advances in Neural Information Processing Systems*, December 2017
[4] **Microsoft.** *LightGBM's documentation.* 2021
[5] **Γεώργιος Α. Τσιχριντζής,** *Σημειώσεις για το μάθημα Αναγνώριση Προτύπων και Μηχανική Μάθηση,* Πειραιεύς 2005
[6]**Leo Breiman, Jerome H. Friedman, Richard A. Olshen, Charles J. Stone,** *Classification And Regression Trees.* 1984
[7] **Heinrich Jiang, Ofir Nachum,** *Identifying and Correcting Label Bias in Machine Learning*, PMLR 108:702-712 2020, 2019
[8] **Michael Abramovici, Jens Christian Göbel, and Matthias Neges,** *Smart Engineering as Enabler for the 4th Industrial Revolution*, Ruhr-University Bochum, Germany, 2015
[9]**Rosenblatt, Frank** *"The Perceptron—a perceiving and recognizing automaton"* Report 85-460-1. Cornell Aeronautical Laboratory, 1957
[10] **Neha Sharma, Reecha Sharma, Neeru Jindal** *"Machine Learning and Deep Learning Applications-A Vision"*, Patiala, Punjab, India, 2021
[11]**Brett Wujek, Patrick Hall, and Funda Güneş,** *"Best Practices for Machine Learning Applications"*, SAS Institute Inc, 2016
[12] **Feng-Lei Fan, Mengzhou Li,** *"On Interpretability of Artificial Neural Networks: A Survey"*, IEEE, 2021