Εργασία 3 – Συνδυασμός Μοντέλων

Χρηματιστηριακή εταιρία θέλει να κατασκευάσει εφαρμογή πρόβλεψης του Αμερικανικού δολαρίου (USD). Χρησιμοποιείστε τη βιβλιοθήκη yfinance στο colab, ώστε να κατεβάσετε δεδομένα tickers για τη τιμή του δολαρίου τα τελευταία 5 έτη, όπως φαίνεται στην παρακάτω εικόνα.

```
1 import yfinance as yf
2
3 ticker_symbol = "DX-Y.NYB"
4 ticker = yf.Ticker(ticker_symbol)
5 df = ticker.history(period="5y").reset_index(drop=False)
6 df
```

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2019-10-23 00:00:00-04:00	97.489998	97.650002	97.419998	97.470001	0	0.0	0.0
1	2019-10-24 00:00:00-04:00	97.459999	97.779999	97.279999	97.629997	0	0.0	0.0
2	2019-10-25 00:00:00-04:00	97.690002	97.900002	97.570000	97.830002	0	0.0	0.0
3	2019-10-28 00:00:00-04:00	97.849998	97.889999	97.669998	97.760002	0	0.0	0.0
4	2019-10-29 00:00:00-04:00	97.760002	97.930000	97.610001	97.690002	0	0.0	0.0
...	...	...	...	...	...	...	...	...
1254	2024-10-17 00:00:00-04:00	103.559998	103.870003	103.440002	103.830002	0	0.0	0.0
1255	2024-10-18 00:00:00-04:00	103.809998	103.809998	103.459999	103.489998	0	0.0	0.0
1256	2024-10-21 00:00:00-04:00	103.459999	104.019997	103.419998	104.010002	0	0.0	0.0
1257	2024-10-22 00:00:00-04:00	103.970001	104.129997	103.820000	104.080002	0	0.0	0.0
1258	2024-10-23 00:00:00-04:00	104.091003	104.545998	104.089996	104.516998	0	0.0	0.0

1259 rows × 8 columns

Περιγραφή Χαρακτηριστικών:

- **Date**: Η ημερομηνία κάθε εγγραφής σε ημερήσια συχνότητα.
- **Open**: Η τιμή εκκίνησης στην αρχή της ημέρας.
- **High**: Η υψηλότερη τιμή μέσα στη μέρα.
- **Low**: Η χαμηλότερη τιμή μέσα στη μέρα.
- **Close**: Η τιμή κλεισίματος στο τέλος της μέρας.

Τις στήλες Volume, Dividends, Stock Splits μπορείτε να τις αφαιρέσετε με την εντολή drop https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.drop.html

Ερωτήματα:

1. Να δημιουργηθεί **ένα μόνο line plot** με άξονα y την τιμή κάθε στήλης (open, high, low, close) και άξονα x την ημερομηνία. Χρησιμοποιείστε τη βιβλιοθήκη *plotly* *https://plotly.com/python/line-charts/* ώστε να υποστηρίζεται η δυνατότητα zoom στο γράφημα. Επίσης μεγαλώστε το γράφημα ώστε να είναι πιο ευκρινές.
2. Από το line plot του ερωτήματος (1), παρατηρείτε πως είναι στατικές ή μη στατικές οι χρονοσειρές? Αιτιολογήστε.
3. Τι παρατηρείτε για την τάση του δολαρίου τον Οκτώβριο του 2024? Να αναζητήσετε τις πιθανές αιτίες που να εξηγούν αυτήν τη τάση.

4. Είναι δυνατόν να προβλέψουμε την τιμή (close) του δολαρίου για την επόμενη μέρα, αν έχουμε ως πληροφορία τα σημερινά open, high, low, close? Εναλλακτικά, είναι δυνατή η εύρεση συνάρτησης $f(o_t, h_t, l_t, c_t) = c_{t+1}$? Αιτιολογείστε.
5. Έχει νόημα να γίνει τυχαίος διαχωρισμός των δεδομένων σε train-test? Αιτιολογήστε.
6. Χωρίστε το σύνολο δεδομένων σε train-test dataframes όπου train όλα τα δεδομένα πριν το 2024 και test όλα τα δεδομένα του 2024. Στη συνέχεια, αφαιρέστε τη στήλη Date από κάθε DataFrame.
7. Δημιουργήστε συνάρτηση που θα παίρνει ως όρισμα ένα dataframe (είτε του train είτε του test), καθώς και μια παράμετρο N και θα επιστρέφει 2 numpy arrays: Inputs (x) και Targets (y), όπου x τα timeframes και y η τιμή close της επόμενης ημέρες. Τα timeframe είναι ένα σύνολο N διαδοχικών γραμμών. Για παράδειγμα αν $N = 3$, τότε τα timeframes θα ήταν πίνακες $Nx4$, όπου N οι γραμμές και 4 τα χαρακτηριστικά open, high, low, close. Το 1ο Timeframe θα περιλαμβάνει τις γραμμές x_0: $[0, N-1]$, το 2ο τις x_1: $[1, N]$, το 3ο x_2: $[2, N+1]$, κλπ. Για τα targets, το y_0 θα ήταν το αντίστοιχο c_N το y_1 το C_{N+1}, κλπ. Για τα Targets μπορείτε να χρησιμοποιήσετε την εντολή shift της pandas.

	Date	Open	High	Low	Close	
0	2019-10-23 00:00:00-04:00	97.489998	97.650002	97.419998	97.470001	x0
1	2019-10-24 00:00:00-04:00	97.459999	97.779999	97.279999	97.629997	x1
2	2019-10-25 00:00:00-04:00	97.690002	97.900002	97.570000	97.830002	x2
3	2019-10-28 00:00:00-04:00	97.849998	97.889999	97.669998	97.760002	
4	2019-10-29 00:00:00-04:00	97.760002	97.930000	97.610001	97.690002	y1
...	...	...	...	...	...	y2
1254	2024-10-17 00:00:00-04:00	103.559998	103.870003	103.440002	103.830002	y3
1255	2024-10-18 00:00:00-04:00	103.809998	103.809998	103.459999	103.489998	
1256	2024-10-21 00:00:00-04:00	103.459999	104.019997	103.419998	104.010002	
1257	2024-10-22 00:00:00-04:00	103.970001	104.129997	103.820000	104.080002	
1258	2024-10-23 00:00:00-04:00	104.091003	104.545998	104.089996	104.516998	

1259 rows × 8 columns

8. Αν θέλουμε να προβλέψουμε την επόμενη ημέρα, το μέγεθος του timeframe N πρέπει να είναι μικρό ή μεγάλο? Τι προτείνετε για το μέγεθος του *N* αν θέλουμε να προβλέψουμε πιο μακρινό ορίζοντα (πχ ένα μήνα)? Αιτιολογείστε.
9. Δημιουργήστε τα *x_train, y_train, x_test, y_test* ορίζοντας ως N το 5. Καθώς τα μοντέλα μηχανικής μάθησης που θα χρησιμοποιήσουμε δέχονται διανύσματα στις εισόδους τους, μετατρέψτε τους πίνακες των inputs σε διανύσματα μεγέθους $Nx4$, δηλαδή 20. Μπορείτε να χρησιμοποιήσετε την εντολή reshape της numpy https://numpy.org/doc/stable/reference/generated/numpy.reshape.html Ελέγξτε ότι το πλήθος των input είναι ίδιο με το πλήθος των target. Ελέγξτε ότι το preprocessing έγινε σωστά, τυπώνοντας το 1ο input του *x_train*.
10. Εκπαιδεύστε τα παρακάτω μοντέλα στο train και υπολογίστε το MAE στα train και test:

a) Linear Regressor Linear Regression

b) Voting με Random Forest Regressor Random Forest Regressor
c) Bagging χρησιμοποιώντας ως estimator: Linear Regression Bagging Regressor
d) Boosting εφαρμόζοντας τον XG-Boost XG-Boost Regressor
e) Stacking χρησιμοποιώντας ως estimator: Linear Regression Stacking Regressor

Στη συνέχεια, να κατασκευαστεί **ένα μόνο bar-plot** για το MAE των a,b,c,d,e στα train και test (μπορείτε να ορίσετε ως μπλε το train και ως πράσινο το test). Τέλος, να εξηγήσετε συνοπτικά σε 2 γραμμές για το κάθε μοντέλο πως δουλεύει.

11. Θα προσπαθήσετε να βελτιώσετε την ακρίβεια των μοντέλων, με τους εξής τρόπους:

a) Για κάθε εγγραφή των δεδομένων, να εξάγετε ως επιπλέον χαρακτηριστικό το μήνα (Χρησιμοποιήστε τη στήλη Date από το αρχικό dataset). Χρησιμοποιείστε τιμή από 0 ως 11 για κάθε μήνα. Στη συνέχεια, διαιρέστε όλες τις τιμές με το 11. Να αναφέρετε για ποιους λόγους ο μήνας ενδεχομένως θα συμβάλει στη βελτίωση των προβλέψεων.
b) Για κάθε στήλη y={Open,High,Low,Close}, εφαρμόστε τη φόρμουλα των Λογαριθμικών Επιστροφών (Log Returns):

$$y_t = ln\left(\frac{y_{t+1}}{y_t}\right).$$

c) Δημιουργήστε ιστόγραμμα για κάθε στήλη και εξηγήστε πως η παραπάνω φόρμουλα ενδεχομένως θα βελτιώσει τις προβλέψεις των μοντέλων.
d) Αν κάποιο μοντέλο προβλέψει $C_{t+1} = 0.01$, τι σημαίνει αυτό για την τιμή Close? Επιπλέον, αν η τιμή $C_t = 0.95\$$, ποια θα είναι η απόλυτη τιμή C_{t+1} σύμφωνα με την παραπάνω πρόβλεψη?
e) Επαναλάβετε τη διαδικασία 7⃣9⃣10.

12. Στη συνέχεια, Εκπαιδεύστε Lasso Regressor (Lasso Regressor) και εμφανίστε τις απόλυτες τιμές των coefficients. Ποια features είναι τα πιο 10 σημαντικά και σε ποιες ημέρες του timeframe αντιστοιχούνται? Αιτιολογήστε. Υπάρχει κάποια μέρα η οποία συμβάλει περισσότερο στην εκτίμηση των προβλέψεων για το timestep $t + 1$?
13. Δημιουργήστε line-plot με τις προβλέπεις (*y_pred*) του lasso και τις πραγματικές τιμές του C_{t+1}. **Προσοχή**: Θα πρέπει να μετατρέψετε τις λογαριθμικές επιστροφές σε απόλυτες τιμές.

Οδηγίες

- Χρησιμοποιείστε την πλατφόρμα Google Colab για την υλοποίηση της άσκησης.
- Τα plots/πινακάκια να εμφανίζονται επάνω στο Colab. Επίσης, μπορείτε να εισάγετε κελιά για κείμενο και πινακάκια όπως φαίνεται στον παρακάτω σύνδεσμο:
- **Δώστε έμφαση στην παρουσίαση της εργασίας. Αν βρεθεί Copy-Paste από το ChatGPT θα μηδενίζεται η άσκηση!**
- Στο e-learning θα υποβάλλετε το link της εργασίας σας στο Google Colab.