



Σχολή Θετικών Επιστημών & Τεχνολογίας
Μεταπτυχιακή Εξειδίκευση στα Πληροφοριακά Συστήματα

Διπλωματική Εργασία

Πρόγνωση τιμής μετοχών με χρήση αλγόριθμων μηχανικής μάθησης σε περιβάλλον
web application και χρήση βιβλιοθήκης Python Streamlit

Ανδρέας Ρούσσος

Επιβλέπων καθηγητής: Ρηγόπουλος Γεώργιος

Πάτρα, Ιούνιος 2026

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή (Ανδρέα Ρούσσου) που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης ο συγγραφέας/δημιουργός εκχωρεί στο ΕΑΠ, μη αποκλειστική άδεια χρήσης του δικαιώματος αναπαραγωγής, προσαρμογής, δημόσιου δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσής τους διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος και για όλο το χρόνο διάρκειας των δικαιωμάτων πνευματικής ιδιοκτησίας. Η ανοικτή πρόσβαση στο πλήρες κείμενο για μελέτη και ανάγνωση δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, αποθήκευση, πώληση, εμπορική χρήση, μετάδοση, διανομή, έκδοση, εκτέλεση, «μεταφόρτωση» (downloading), «ανάρτηση» (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού. Ο συγγραφέας/δημιουργός διατηρεί το σύνολο των ηθικών και περιουσιακών του δικαιωμάτων.



«Πρόγνωση τιμής μετοχών με χρήση αλγόριθμων μηχανικής μάθησης σε περιβάλλον web application και χρήση βιβλιοθήκης Python Streamlit»

Ανδρέας Ρούσσος

Επιτροπή Επίβλεψης Διπλωματικής Εργασίας

Επιβλέπων Καθηγητής:

Γεώργιος Ρηγόπουλος

Μέλος ΣΕΠ, ΕΑΠ

Συν-Επιβλέπων Καθηγητής:

Ευστράτιος Γεωργόπουλος

Καθηγητής Α' Βαθμίδας, ΕΑΠ

Πάτρα, Ιούνιος 2026

Ευχαριστίες

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στη μητέρα μου, η οποία υπήρξε διαχρονικά αρωγός και στήριγμα σε κάθε μου προσπάθεια, ιδιαίτερα στην παρότρυνση της να ξεκινήσω το παρόν Μεταπτυχιακό καθώς ήμουν πλήρωμα Φρεγάτας στο Πολεμικό Ναυτικό και οι συνθήκες ήταν αρκετά δυσμενείς και αφιλόξενες. Η αμέριστη υποστήριξη, η κατανόηση και η ενθάρρυνσή της συνέβαλαν καθοριστικά στην ολοκλήρωση των σπουδών μου και της παρούσας διπλωματικής εργασίας.

Περίληψη

Η παρούσα διπλωματική εργασία παρουσιάζει το σχεδιασμό και την υλοποίηση ενός ολοκληρωμένου συστήματος πρόβλεψης τιμών μετοχών με χρήση αλγορίθμων μηχανικής μάθησης, υλοποιημένου ως διαδικτυακή εφαρμογή με τη βιβλιοθήκη Streamlit της Python. Το σύστημα αντλεί ιστορικά δεδομένα OHLCV από το Yahoo Finance και εφαρμόζει εκτεταμένο Feature Engineering, παράγοντας άνω των 25 τεχνικών χαρακτηριστικών που περιλαμβάνουν δείκτες τάσης, ορμής, μεταβλητότητας και όγκου. Τρεις αλγόριθμοι εποπτευόμενης παλινδρόμησης εκπαιδεύονται και συγκρίνονται παράλληλα: Random Forest, Gradient Boosting και Ridge Regression. Η αξιολόγηση πραγματοποιείται με χρονολογικό διαχωρισμό δεδομένων (80/20), αποφεύγοντας τη χρονική διαρροή πληροφοριών, και χρησιμοποιεί τις μετρικές MAE, RMSE, R^2 και Κατευθυντήρια Ακρίβεια. Τα πειράματα διεξήχθησαν για τη μετοχή Apple Inc. (AAPL) κατά το 2025 με ορίζοντα πρόβλεψης 7 ημερών. Η Ridge Regression επέδειξε την υψηλότερη απόδοση με MAE=\$5.36 και Κατευθυντήρια Ακρίβεια 75.7%, υπερβαίνοντας σημαντικά το τυχαίο 50%. Η εφαρμογή ενσωματώνει επίσης ανάλυση χρηματοοικονομικού κινδύνου (Sharpe Ratio, VaR, Maximum Drawdown) και παρουσίαση θεμελιωδών οικονομικών μεγεθών, αποτελώντας ένα ολοκληρωμένο εργαλείο ανάλυσης μετοχών.

Λέξεις – Κλειδιά:

Μηχανική Μάθηση, Πρόβλεψη Τιμών Μετοχών, Random Forest, Gradient Boosting, Ridge Regression, Τεχνική Ανάλυση, Feature Engineering, Χρονοσειρές, Κατευθυντήρια Ακρίβεια, Streamlit, Διαχείριση Κινδύνου, Sharpe Ratio, Value at Risk

«Stock Price Forecasting Using Machine Learning Algorithms in a Web Application Environment with the Python Streamlit Library»

Abstract

This thesis presents the design and implementation of a comprehensive stock price prediction system using machine learning algorithms, developed as a web application with Python's Streamlit library. The system retrieves historical OHLCV data from Yahoo Finance and applies extensive feature engineering, generating over 25 technical indicators covering trend, momentum, volatility, and volume categories. Three supervised regression algorithms are trained and compared in parallel: Random Forest, Gradient Boosting, and Ridge Regression. Model evaluation is performed using chronological data splitting (80/20), preventing temporal data leakage, and employs the metrics MAE, RMSE, R^2 , and Directional Accuracy. Experiments were conducted on Apple Inc. (AAPL) stock during 2025 with a 7-day forecast horizon. Ridge Regression achieved the highest performance with MAE=\$5.36 and Directional Accuracy of 75.7%, significantly exceeding the random baseline of 50%. The application also incorporates financial risk analysis (Sharpe Ratio, Value at Risk, Maximum Drawdown) and company fundamentals, forming a comprehensive stock analysis tool. Results confirm the partial predictability of financial markets despite the Efficient Market Hypothesis, while highlighting the critical role of feature engineering quality and appropriate model selection relative to dataset size.

Keywords:

Machine Learning, Stock Price Prediction, Random Forest, Gradient Boosting, Ridge Regression, Technical Analysis, Feature Engineering, Time Series, Directional Accuracy, Streamlit, Risk Management, Sharpe Ratio, Value at Risk, Efficient Market Hypothesis, Supervised Learning

Περιεχόμενα

Περίληψη.....	5
Abstract	6
Περιεχόμενα	7
Κατάλογος Εικόνων / Σχημάτων.....	10
Κατάλογος Πινάκων	12
1. Εισαγωγή	13
1.1 Κίνητρο και Πλαίσιο	13
1.2 Ερευνητικά Ερωτήματα.....	14
1.3 Συνεισφορά της Εργασίας	14
1.4 Δομή της Εργασίας.....	15
2. Χρηματοοικονομικές Αγορές και Τεχνική Ανάλυση	16
2.1 Βασικές Έννοιες Χρηματιστηριακών Αγορών.....	16
2.2 Δεδομένα OHLCV.....	16
2.3 Υπόθεση Αποτελεσματικής Αγοράς.....	17
2.4 Τεχνική και Θεμελιώδης Ανάλυση.....	18
2.5 Τεχνικοί Δείκτες Τάσης.....	19
2.5.1 Απλός Κινητός Μέσος (SMA)	19
2.5.2 Εκθετικός Κινητός Μέσος (EMA)	19
2.6 Τεχνικοί Δείκτες Ορμής	20
2.6.1 Σχετικός Δείκτης Ισχύος (RSI).....	20
2.6.2 MACD	20
2.7 Δείκτες Μεταβλητότητας	21
2.7.1 Bollinger Bands	21
2.7.2 Average True Range (ATR)	21
2.8 Δείκτες Όγκου	22
2.8.1 On-Balance Volume (OBV)	22
2.8.2 Στοχαστικός Ταλαντωτής	22
2.9 Μετρικές Χρηματοοικονομικού Κινδύνου.....	23
3. Μηχανική Μάθηση και Χρονοσειρές.....	25
3.1 Εισαγωγή στη Μηχανική Μάθηση	25
3.2 Διατύπωση ως Πρόβλημα Παλινδρόμησης Χρονοσειράς.....	25
3.3 Ιδιαιτερότητες Χρονοσειρών	26
3.4 Χρονολογικός Διαχωρισμός Δεδομένων (Time Series Split)	26
3.5 Feature Engineering.....	27

3.6 Κανονικοποίηση Χαρακτηριστικών	28
3.7 Υπερπροσαρμογή, Υποπροσαρμογή και Κανονικοποίηση	29
3.8 Μετρικές Αξιολόγησης Μοντέλων	29
4. Αλγόριθμοι Μηχανικής Μάθησης και Σχετικές Εργασίες	31
4.1 Random Forest Regressor	31
4.2 Gradient Boosting Regressor	32
4.3 Ridge Regression	32
4.4 Σχετικές Εργασίες (Related Work)	33
5. Σχεδιασμός και Υλοποίηση της Εφαρμογής	36
5.1 Αρχιτεκτονική Συστήματος	36
5.2 Επιλογή Τεχνολογιών	37
5.3 Άντληση και Προεπεξεργασία Δεδομένων	38
5.4 Feature Engineering — Υπολογισμός Τεχνικών Δεικτών	39
5.4.1 Χαρακτηριστικά Αποδόσεων	39
5.4.2 Χαρακτηριστικά Τάσης	39
5.4.3 Χαρακτηριστικά Ορμής και Μεταβλητότητας	39
5.4.4 Χαρακτηριστικά Όγκου	40
5.5 Δημιουργία Supervised Dataset και Χρονολογικός Διαχωρισμός	40
5.6 Pipeline Εκπαίδευσης Μοντέλων	41
5.7 Υπολογισμός Μετρικών Κινδύνου	42
5.8 Περιγραφή Διεπαφής Χρήστη	43
5.8.1 Sidebar — Παράμετροι Ανάλυσης	43
5.8.2 Tab 1 — Τιμή & Επισκόπηση	43
5.8.3 Tab 2 — Τεχνική Ανάλυση	44
5.8.4 Tab 3 — Πρόβλεψη ML	44
5.8.5 Tab 4 — Σύγκριση Μοντέλων	45
5.8.6 Tab 5 — Κίνδυνος & Θεμελιώδη	45
5.9 Αποθήκευση και Λήψη Εκπαιδευμένων Μοντέλων	45
5.10 Παράδειγμα Εκτέλεσης	46
6. Πειράματα και Αποτελέσματα	47
6.1 Πειραματική Διαδικασία	47
6.2 Ανάλυση Τιμής και Αποδόσεων AAPL (2025)	47
6.3 Ανάλυση Τεχνικών Δεικτών	49
6.4 Αποτελέσματα Μοντέλων Μηχανικής Μάθησης	50
6.4.1 Συγκριτική Αξιολόγηση Μοντέλων	50
6.4.2 Ανάλυση R^2	51

6.5 Αξιολόγηση Πρωτεύοντος Μοντέλου (Ridge Regression)	52
6.6 Ανάλυση Καταλοίπων και Μελλοντική Πρόβλεψη	53
6.7 Διάγραμμα Διασποράς: Πραγματικές vs Προβλεπόμενες Τιμές.....	53
6.8 Ανάλυση Κινδύνου AAPL	54
6.9 Θεμελιώδη Οικονομικά Μεγέθη	56
6.10 Συζήτηση Αποτελεσμάτων	56
7. Συμπεράσματα και Μελλοντική Εργασία	57
Βιβλιογραφία	62
Παράρτημα Α: «Κώδικας υλοποίησης σε περιβάλλον VSC»	64

Κατάλογος Εικόνων / Σχημάτων

Εικόνα 1: Αρχιτεκτονική Εφαρμογής.....	36
Εικόνα 2: Candlestick γράφημα AAPL 2025 με SMA_20 (πορτοκαλί), SMA_50 (μπλε) και Bollinger Bands. Παρατηρείται η έντονη καθοδική κίνηση Φεβρουαρίου-Απριλίου 2025 (~-30%) και η ισχυρή ανοδική τάση του β' εξαμήνου.	48
Εικόνα 3: Κατανομή ημερήσιων αποδόσεων AAPL (αριστερά) και σωρευτική απόδοση 2025 (δεξιά). Σωρευτική απόδοση +12.5% με αρνητική φάση α' εξαμήνου και ισχυρή ανάκαμψη β' εξαμήνου.....	48
Εικόνα 4: Σύνθετο γράφημα τεχνικής ανάλυσης AAPL — MACD (άνω), RSI_14 (μέσο) και Stochastic Oscillator (κάτω). Οι χρωματισμένες περιοχές υποδηλώνουν ζώνες υπεραγοράς/υπερπώλησης.....	49
Εικόνα 5: Signal Summary τελευταίας ημέρας ανάλυσης. RSI: Neutral (39.1), MACD: Bearish (0.0790/Signal 0.8087), Bollinger: Neutral (31.29%), Stochastic: Neutral (%K=46.4).	49
Εικόνα 6: Ραβδόγραμμα σύγκρισης MAE (\$) ανά μοντέλο. Η Ridge Regression (μοβ, \$5.361) υπερτερεί σαφώς έναντι Random Forest (πορτοκαλί, \$8.608) και Gradient Boosting (πράσινο, \$10.504).....	51
Εικόνα 7: Σύγκριση προβλέψεων και των τριών μοντέλων vs πραγματικές τιμές AAPL (test set, Νοέμβριος-Δεκέμβριος 2025). Παρατηρείται η σταθερότερη συμπεριφορά της Ridge έναντι της αστάθειας του Gradient Boosting στο τέλος περιόδου	52
Εικόνα 8: Πρόβλεψη Ridge Regression vs πραγματικές τιμές AAPL (test set). MAE=\$5.36, RMSE=\$6.49, $R^2=-0.943$, MAPE=1.96%, Directional Accuracy=75.7%. Η σκιασμένη περιοχή αντιστοιχεί σε ± 1 RMSE.....	52

Εικόνα 9: Ανάλυση καταλοίπων Ridge Regression (άνω) και πρόβλεψη επόμενων 7 ημερών (κάτω). Προβλεπόμενη τιμή \$280.70 (+2.98%) με εύρος εμπιστοσύνης \$274.21–\$287.19. 53	
Εικόνα 10: Scatter plot πραγματικών (x) vs προβλεπόμενων τιμών (y) για τον Gradient Boosting. Η διακεκομμένη γραμμή αντιστοιχεί στην τέλεια πρόβλεψη. Η συγκέντρωση σημείων στη ζώνη \$265-\$270 αντικατοπτρίζει την τάση υποεκτίμησης ακραίων τιμών.....54	
Εικόνα 11: Μετρικές κινδύνου και ιστορικό drawdown AAPL 2025. Το μέγιστο drawdown - 30.22% καταγράφηκε τον Απρίλιο 2025, με σταδιακή ανάκαμψη από τον Ιούλιο.55	
Εικόνα 12: Κυλιόμενος Sharpe Ratio 60 ημερών. Παρατηρείται η μετάβαση από αρνητικές τιμές (α' εξάμηνο 2025) σε σταθερά θετικές τιμές 2-4 (β' εξάμηνο), υποδηλώνοντας σημαντική βελτίωση σχέσης απόδοσης/κινδύνου.56	
Εικόνα 13: Θεμελιώδη οικονομικά μεγέθη Apple Inc. όπως παρουσιάζονται στο Tab 5 της εφαρμογής.....56	

Κατάλογος Πινάκων

Πίνακας 1: Βιβλιοθήκες που χρησιμοποιήθηκαν	37
Πίνακας 2: Παράμετροι εκπαίδευσης μοντέλων	42
Πίνακας 3: Πίνακας σύγκρισης μοντέλων όπως εμφανίζεται στο Tab 4 της εφαρμογής. Η πράσινη σκίαση επισημαίνει αυτόματα τη βέλτιστη τιμή κάθε μετρικής.	50
Πίνακας 4: Μετρικές κινδύνου AAPL, 2025	55

1. Εισαγωγή

1.1 Κίνητρο και Πλαίσιο

Η πρόβλεψη της τιμής των μετοχών αποτελεί ένα από τα πλέον πολύπλοκα και ταυτόχρονα πιο ελκυστικά προβλήματα στην επιστήμη των δεδομένων και στη χρηματοοικονομική θεωρία. Οι χρηματαγορές είναι εκ φύσεως πολύπλοκα, δυναμικά και μη γραμμικά συστήματα, στα οποία η τιμή μιας μετοχής επηρεάζεται από ένα ευρύ φάσμα παραγόντων: οικονομικά αποτελέσματα εταιρειών, μακροοικονομικές εξελίξεις, αλλαγές επιτοκίων, γεωπολιτικές αναταράξεις, ακόμα και η ψυχολογία και η συμπεριφορά των επενδυτών. Αυτή η πολυπαραγοντική φύση καθιστά την ακριβή πρόβλεψη τιμών εξαιρετικά δύσκολη και αμφιλεγόμενη.

Ωστόσο, η ραγδαία ανάπτυξη της υπολογιστικής ισχύος, η διαθεσιμότητα τεράστιων ποσοτήτων χρηματοοικονομικών δεδομένων και η εξέλιξη των αλγορίθμων μηχανικής μάθησης έχουν ανανεώσει το ενδιαφέρον για την αυτοματοποιημένη ανάλυση αγορών. Από τη δεκαετία του 2010 και ιδιαίτερα μετά το 2015, ένας αυξανόμενος αριθμός ακαδημαϊκών εργασιών έχει καταδείξει ότι αλγόριθμοι μηχανικής μάθησης (από τα απλούστερα μοντέλα παλινδρόμησης έως τα πολύπλοκα δίκτυα βαθιάς μάθησης) μπορούν να εξάγουν στατιστικά σημαντικά σήματα από ιστορικά δεδομένα τιμών. Παρότι η πλήρης πρόβλεψη τιμών παραμένει ανέφικτη, η μερική εξαγωγή χρήσιμης πληροφορίας από ιστορικές χρονοσειρές είναι εφικτή και πρακτικά χρήσιμη.

Στο πλαίσιο αυτό, η παρούσα εργασία εντάσσεται στην τρέχουσα ερευνητική προσπάθεια για την ανάπτυξη αξιόπιστων και ερμηνεύσιμων συστημάτων πρόβλεψης χρηματιστηριακών τιμών. Η έμφαση δίνεται όχι μόνο στη βελτιστοποίηση της ακρίβειας πρόβλεψης, αλλά και στη δημιουργία ενός ολοκληρωμένου εργαλείου ανάλυσης που συνδυάζει τεχνική ανάλυση, μηχανική μάθηση και αξιολόγηση χρηματοοικονομικού κινδύνου σε ένα διαδραστικό διαδικτυακό περιβάλλον. Η επιλογή της βιβλιοθήκης Streamlit για την ανάπτυξη του front-end επέτρεψε τη δημιουργία μιας πλήρως λειτουργικής εφαρμογής με ελάχιστο πρόσθετο κώδικα διεπαφής, επιτρέποντας τη συγκέντρωση της προσπάθειας στη μοντελοποίηση και την ανάλυση.

1.2 Ερευνητικά Ερωτήματα

Η παρούσα εργασία επιχειρεί να απαντήσει σε τέσσερα κεντρικά ερευνητικά ερωτήματα. Το πρώτο αφορά την προβλεπτική ικανότητα των αλγορίθμων μηχανικής μάθησης: σε ποιο βαθμό μπορούν αλγόριθμοι εποπτευόμενης μάθησης να προβλέψουν την τιμή κλεισίματος μετοχών για ορίζοντες πρόβλεψης από 1 έως 30 ημέρες, και ποια η αξιοπιστία των προβλέψεων αυτών; Το δεύτερο ερώτημα αφορά τη συγκριτική απόδοση των αλγορίθμων: υπάρχει στατιστικά σημαντική διαφορά απόδοσης μεταξύ Random Forest, Gradient Boosting και Ridge Regression, και ποιοι παράγοντες την καθορίζουν;

Το τρίτο ερευνητικό ερώτημα εστιάζει στη σπουδαιότητα χαρακτηριστικών: ποιοι τεχνικοί δείκτες και χαρακτηριστικά συνεισφέρουν περισσότερο στην ακρίβεια πρόβλεψης, και πώς αυτή η σπουδαιότητα διαφοροποιείται ανά μετοχή και ορίζοντα πρόβλεψης; Το τέταρτο και τελευταίο ερώτημα είναι πρακτικής φύσης: πώς μπορεί ένα σύστημα τέτοιας υπολογιστικής και αναλυτικής πολυπλοκότητας να παρουσιαστεί μέσω μιας φιλικής, διαδραστικής εφαρμογής που να είναι προσβάσιμη σε χρήστες χωρίς εξειδικευμένο τεχνικό υπόβαθρο;

1.3 Συνεισφορά της Εργασίας

Η παρούσα εργασία συνεισφέρει στο πεδίο κατά τρεις βασικούς τρόπους. Πρώτον, αναπτύσσεται ένα ολοκληρωμένο σύστημα Feature Engineering που παράγει άνω των 25 τεχνικών χαρακτηριστικών από ακατέργαστα OHLCV δεδομένα, καλύπτοντας τεχνικούς δείκτες από τέσσερις κατηγορίες: τάση, ορμή, μεταβλητότητα και όγκος. Δεύτερον, εφαρμόζεται μεθοδολογικά ορθή αξιολόγηση βάσει χρονολογικού διαχωρισμού (Time Series Split) με πολλαπλές μετρικές, συμπεριλαμβανομένης της Κατευθυντήριας Ακρίβειας (Directional Accuracy), που σπάνια αξιολογείται στη βιβλιογραφία. Τρίτον, το σύστημα ενσωματώνει ανάλυση χρηματοοικονομικού κινδύνου (Sharpe Ratio, VaR, Maximum Drawdown) που συνδέει τις προβλέψεις με πρακτικές επενδυτικές αποφάσεις.

1.4 Δομή της Εργασίας

Η εργασία οργανώνεται σε επτά κεφάλαια. Το παρόν κεφάλαιο παρουσιάζει το κίνητρο, τα ερευνητικά ερωτήματα και τη δομή της εργασίας. Το Κεφάλαιο 2 καλύπτει το θεωρητικό υπόβαθρο των χρηματοοικονομικών αγορών, της τεχνικής ανάλυσης και των τεχνικών δεικτών με τις μαθηματικές τους διατυπώσεις. Το Κεφάλαιο 3 εισάγει τις αρχές της μηχανικής μάθησης για χρονοσειρές, το Feature Engineering, τη μεθοδολογία αξιολόγησης και τις μετρικές που χρησιμοποιούνται. Το Κεφάλαιο 4 παρουσιάζει τους τρεις αλγορίθμους που υλοποιήθηκαν μαζί με ανασκόπηση σχετικών ακαδημαϊκών εργασιών. Τα Κεφάλαια 5 και 6 καλύπτουν τον σχεδιασμό, την υλοποίηση και τα πειραματικά αποτελέσματα της εφαρμογής, ενώ το Κεφάλαιο 7 ολοκληρώνει με συμπεράσματα και προτάσεις για μελλοντική έρευνα.

2. Χρηματοοικονομικές Αγορές και Τεχνική Ανάλυση

2.1 Βασικές Έννοιες Χρηματιστηριακών Αγορών

Οι χρηματιστηριακές αγορές αποτελούν οργανωμένους θεσμούς διαπραγμάτευσης στους οποίους αγοραστές και πωλητές ανταλλάσσουν χρηματοοικονομικά μέσα, κυρίως μετοχές εισηγμένων εταιρειών. Κάθε μετοχή αντιπροσωπεύει ένα κλάσμα της ιδιοκτησίας μιας εταιρείας, παρέχοντας στον κάτοχό της δικαιώματα επί των κερδών (μερίσματα) και δικαίωμα ψήφου στις γενικές συνελεύσεις. Η τιμή της μετοχής διαμορφώνεται μέσα από τον μηχανισμό προσφοράς και ζήτησης και αντανακλά τη συλλογική εκτίμηση των συμμετεχόντων για την τρέχουσα και μελλοντική αξία της επιχείρησης. Οι σύγχρονες χρηματαγορές χαρακτηρίζονται από υψηλή ρευστότητα, διαφάνεια και αυτοματοποιημένες διαδικασίες εκκαθάρισης, που επιτρέπουν εκατομμύρια συναλλαγές ανά ημέρα σε παγκόσμιο επίπεδο.

Η διαπραγμάτευση μετοχών παράγει καθημερινά ένα πλούσιο σύνολο δεδομένων το οποίο αποτυπώνει τη συμπεριφορά της αγοράς σε κάθε χρονική στιγμή. Η επεξεργασία αυτών των δεδομένων, η εξαγωγή χρήσιμης πληροφορίας και η πρόβλεψη μελλοντικών κινήσεων τιμών αποτελούν το αντικείμενο τόσο της τεχνικής ανάλυσης όσο και της εφαρμογής αλγορίθμων μηχανικής μάθησης. Κατανοώντας τη δομή και τη σημασία αυτών των δεδομένων δημιουργείται η αναγκαία βάση για τη μεθοδολογία που ακολουθείται στην παρούσα εργασία.

2.2 Δεδομένα OHLCV

Τα θεμελιώδη δεδομένα που καταγράφονται για κάθε χρηματιστηριακή συνεδρίαση είναι γνωστά με το ακρωνύμιο OHLCV, το οποίο αναφέρεται στις τιμές ανοίγματος (Open), υψηλότερης (High) και χαμηλότερης τιμής ημέρας (Low), τιμής κλεισίματος (Close) και όγκου συναλλαγών (Volume) [5]. Η τιμή ανοίγματος (Open) αντιστοιχεί στην πρώτη τιμή που καταγράφεται κατά την έναρξη της συνεδρίασης και επηρεάζεται σημαντικά από την προσυνεδριακή δραστηριότητα και τις ειδήσεις που ανακοινώθηκαν μετά το κλείσιμο της προηγούμενης ημέρας. Η τιμή υψηλού (High) και η τιμή χαμηλού (Low) αντιπροσωπεύουν

αντίστοιχα τη μέγιστη και ελάχιστη τιμή συναλλαγής κατά τη διάρκεια της ημέρας, αποτυπώνοντας το εύρος διακύμανσης και συνεπώς τη μεταβλητότητα της αγοράς.

Η τιμή κλεισίματος (Close) αποτελεί την τελευταία τιμή διαπραγμάτευσης κατά τη λήξη της συνεδρίασης και είναι η πλέον χρησιμοποιούμενη τιμή για την κατασκευή τεχνικών δεικτών και μοντέλων πρόβλεψης. Αντιπροσωπεύει τη συναινετική εκτίμηση της αγοράς στο τέλος κάθε ημέρας διαπραγμάτευσης και χρησιμοποιείται ευρέως για τον υπολογισμό ημερήσιων αποδόσεων και την κατασκευή χρηματοοικονομικών δεικτών. Ο όγκος συναλλαγών (Volume) εκφράζει τον συνολικό αριθμό μετοχών που άλλαξαν χέρια κατά τη διάρκεια της ημέρας και αποτελεί σημαντικό δείκτη της ρευστότητας, της δύναμης μιας τάσης και του επενδυτικού ενδιαφέροντος. Γενικά, η τιμολογιακή κίνηση που συνοδεύεται από υψηλό όγκο θεωρείται πιο αξιόπιστη από αντίστοιχη κίνηση με χαμηλό όγκο.

2.3 Υπόθεση Αποτελεσματικής Αγοράς

Η Υπόθεση Αποτελεσματικής Αγοράς (Efficient Market Hypothesis — EMH), διατυπώθηκε συστηματικά από τον Fama [6] και αποτελεί έναν από τους ακρογωνιαίους λίθους της σύγχρονης χρηματοοικονομικής θεωρίας. Σύμφωνα με αυτήν, οι τιμές των χρηματοοικονομικών μέσων αντικατοπτρίζουν ανά πάσα στιγμή πλήρως όλες τις διαθέσιμες πληροφορίες, με αποτέλεσμα η επίτευξη συστηματικής υπεραπόδοσης έναντι της αγοράς να καθίσταται αδύνατη. Η θεωρία διακρίνει τρεις μορφές αποτελεσματικότητας: την ασθενή μορφή (weak form), σύμφωνα με την οποία οι τιμές ενσωματώνουν πλήρως τα ιστορικά δεδομένα τιμών και όγκου· την ημι-ισχυρή μορφή (semi-strong form), κατά την οποία ενσωματώνονται επίσης όλες οι δημόσια διαθέσιμες πληροφορίες, συμπεριλαμβανομένων των ανακοινώσεων εταιρειών και των μακροοικονομικών δεδομένων· και την ισχυρή μορφή (strong form), κατά την οποία ακόμα και οι εσωτερικές, μη δημόσιες πληροφορίες αντανακλώνται πλήρως στις τιμές.

Εάν η ασθενής μορφή της EMH ισχύει στην πράξη, τότε η πρόβλεψη τιμών με βάση ιστορικά δεδομένα (όπως επιχειρείται στην παρούσα εργασία), δεν θα έπρεπε να παράγει συστηματική υπεραπόδοση. Ωστόσο, ένας αυξανόμενος όγκος εμπειρικής έρευνας έχει αμφισβητήσει την απόλυτη ισχύ της EMH, καταδεικνύοντας την ύπαρξη στατιστικά σημαντικών μοτίβων στα ιστορικά δεδομένα [7]. Φαινόμενα όπως η momentum effect (τάση

των μετοχών να συνεχίζουν την υπάρχουσα κατεύθυνσή τους βραχυπρόθεσμα), η mean reversion (τάση επιστροφής στον μέσο μακροπρόθεσμα), τα εποχιακά μοτίβα (π.χ. the January effect) και η υπεραντίδραση των επενδυτών σε νέα αποτελούν τεκμηριωμένες ανωμαλίες που δεν εξηγούνται από την ΕΜΗ. Οι ανωμαλίες αυτές εγείρουν αμφιβολίες για την πλήρη ισχύ της ΕΜΗ και ταυτόχρονα δικαιολογούν θεωρητικά την προσπάθεια εξαγωγής προβλεπτικής πληροφορίας από ιστορικές χρονοσειρές μέσω αλγορίθμων μηχανικής μάθησης.

2.4 Τεχνική και Θεμελιώδης Ανάλυση

Δύο κύριες προσεγγίσεις χρησιμοποιούνται από τους επαγγελματίες επενδυτές για την ανάλυση και πρόβλεψη κινήσεων αγοράς: η τεχνική ανάλυση και η θεμελιώδης ανάλυση. Η τεχνική ανάλυση βασίζεται αποκλειστικά στη μελέτη ιστορικών δεδομένων τιμών και όγκου, αρνούμενη ότι η γνώση θεμελιωδών οικονομικών μεγεθών μπορεί να παράσχει στατιστικό πλεονέκτημα. Αντλεί από τρεις θεμελιώδεις παραδοχές: *η αγορά προεξοφλεί τα πάντα* (market discounts everything), *οι τιμές κινούνται σε τάσεις* (prices move in trends) και *η ιστορία επαναλαμβάνεται* (history repeats itself). Υποστηρίζεται ότι η συλλογική ψυχολογία των επενδυτών δημιουργεί επαναλαμβανόμενα μοτίβα στα γραφήματα τιμών, τα οποία μπορούν να αναγνωριστούν και να αξιοποιηθούν προβλεπτικά [8].

Η θεμελιώδης ανάλυση, αντίθετα, εστιάζει στην εκτίμηση της εγγενούς αξίας (intrinsic value) μιας επιχείρησης μέσω της ανάλυσης των οικονομικών της καταστάσεων, της ανταγωνιστικής θέσης στον κλάδο, της ποιότητας της διοίκησης και των μακροοικονομικών συνθηκών. Μεγέθη όπως ο λόγος τιμής προς κέρδη (P/E ratio), η αγοραία κεφαλαιοποίηση, τα περιθώρια κέρδους και ο ρυθμός ανάπτυξης εσόδων αποτελούν τα κεντρικά εργαλεία της θεμελιώδους ανάλυσης. Στην παρούσα εργασία, η έμφαση δίνεται στην τεχνική ανάλυση, καθώς οι τεχνικοί δείκτες αποτελούν τα κύρια χαρακτηριστικά εισόδου στα μοντέλα μηχανικής μάθησης. Ωστόσο, θεμελιώδη μεγέθη παρουσιάζονται στην εφαρμογή ως συμπληρωματικές πληροφορίες για την αξιολόγηση της γενικότερης επενδυτικής θέσης.

2.5 Τεχνικοί Δείκτες Τάσης

2.5.1 Απλός Κινητός Μέσος (SMA)

Ο Απλός Κινητός Μέσος (Simple Moving Average — SMA) αποτελεί τον σημαντικότερο τεχνικό δείκτη τάσης και υπολογίζεται ως ο αριθμητικός μέσος των τιμών κλεισίματος για μια δεδομένη περίοδο n ημερών. Μαθηματικά, για μια χρονική στιγμή t , ο SMA_n ορίζεται ως:

$$SMA_n(t) = (1/n) \cdot \sum P(t-i), \quad i = 0, \dots, n-1$$

όπου $P(t)$ η τιμή κλεισίματος την ημέρα t . Ο SMA ομαλοποιεί τις βραχυπρόθεσμες διακυμάνσεις, αναδεικνύοντας την υποκείμενη τάση της μετοχής. Στην εφαρμογή χρησιμοποιούνται τρεις κινητοί μέσοι περιόδων 10, 20 και 50 ημερών. Ο SMA_{20} και SMA_{50} αποτελούν τους πιο ευρέως παρακολουθούμενους κινητούς μέσους· η διασταύρωση του SMA_{20} πάνω από τον SMA_{50} (golden cross) ερμηνεύεται ως ισχυρό αγοραστικό σήμα, ενώ η αντίστροφη διασταύρωση (death cross) ως σήμα πώλησης.

2.5.2 Εκθετικός Κινητός Μέσος (EMA)

Ο Εκθετικός Κινητός Μέσος (Exponential Moving Average — EMA) αποτελεί βελτίωση του SMA, καθώς αποδίδει μεγαλύτερο βάρος στις πιο πρόσφατες παρατηρήσεις. Αυτό τον καθιστά ταχύτερο στην ανταπόκριση σε αλλαγές τιμής. Ο EMA_n για την ημέρα t υπολογίζεται αναδρομικά ως:

$$EMA_n(t) = \alpha \cdot P(t) + (1 - \alpha) \cdot EMA_n(t-1)$$

όπου ο συντελεστής εξομάλυνσης $\alpha = 2 / (n+1)$. Μεγαλύτερες τιμές α αυξάνουν την ανταπόκριση στις πρόσφατες τιμές. Στην εφαρμογή υπολογίζονται οι EMA_{12} και EMA_{26} , που αποτελούν τα βασικά συστατικά του δείκτη MACD.

2.6 Τεχνικοί Δείκτες Ορμής

2.6.1 Σχετικός Δείκτης Ισχύος (RSI)

Ο Σχετικός Δείκτης Ισχύος (Relative Strength Index — RSI) αναπτύχθηκε από τον Wilder και αποτελεί έναν ταλαντωτή ορμής που μετράει την ταχύτητα και τη μεταβολή των κινήσεων τιμής [9]. Για τον υπολογισμό του απαιτείται η εύρεση του μέσου κέρδους και της μέσης ζημίας για μια περίοδο n ημερών (συνήθως $n=14$):

$$RS = \text{Μέσο Κέρδος}(n) / \text{Μέση Ζημία}(n)$$

$$RSI = 100 - 100 / (1 + RS)$$

Ο RSI κυμαίνεται σε κλίμακα από 0 έως 100. Τιμές άνω του 70 υποδηλώνουν κατάσταση υπεραγοράς (overbought), υποδηλώνοντας πιθανή αντιστροφή προς τα κάτω, ενώ τιμές κάτω του 30 υποδηλώνουν κατάσταση υπερπώλησης (oversold), υποδηλώνοντας πιθανή αντιστροφή προς τα άνω [10]. Ο RSI χρησιμοποιείται στην εφαρμογή τόσο ως χαρακτηριστικό εισόδου στα μοντέλα μηχανικής μάθησης, όσο και ως σήμα αγοραπωλησίας στο γράφημα τεχνικής ανάλυσης. Η αποκλίνουσα συμπεριφορά (divergence) μεταξύ RSI και τιμής μετοχής (όταν δηλαδή η τιμή κάνει νέο υψηλό ενώ ο RSI δεν το επιβεβαιώνει), αποτελεί ισχυρό σήμα εξασθένησης της τάσης.

2.6.2 MACD

Ο δείκτης MACD (Moving Average Convergence Divergence) αποτελεί έναν δείκτη τάσης και ορμής που αναπτύχθηκε από τον Appel. Υπολογίζεται ως η διαφορά μεταξύ του EMA_{12} και EMA_{26} :

$$MACD(t) = EMA_{12}(t) - EMA_{26}(t)$$

$$Signal(t) = EMA_9(MACD(t))$$

$$Histogram(t) = MACD(t) - Signal(t)$$

Το ιστόγραμμα MACD εκφράζει τη διαφορά μεταξύ της γραμμής MACD και της γραμμής σήματος, αποτυπώνοντας οπτικά την ορμή της αγοράς. Θετικές και αυξανόμενες τιμές ιστογράμματος υποδηλώνουν ισχυρή ανοδική ορμή, ενώ αρνητικές και φθίνουσες τιμές

υποδηλώνουν ισχυρή καθοδική ορμή. Η διασταύρωση της γραμμής MACD πάνω από τη γραμμή σήματος ερμηνεύεται ως αγοραστικό σήμα, ενώ η αντίστροφη διασταύρωση ως σήμα πώλησης [8]. Στην εφαρμογή, και οι τρεις τιμές (MACD, Signal, Histogram) υπολογίζονται και χρησιμοποιούνται τόσο για γραφιστική αναπαράσταση όσο και ως χαρακτηριστικά για τα μοντέλα πρόβλεψης.

2.7 Δείκτες Μεταβλητότητας

2.7.1 Bollinger Bands

Τα Bollinger Bands, που εισήγαγε ο Bollinger στις αρχές της δεκαετίας του 1980, αποτελούν δείκτη μεταβλητότητας που αποτυπώνει την τυπική απόκλιση των τιμών γύρω από έναν κινητό μέσο. Αποτελούνται από τρεις γραμμές:

$$\text{Middle Band}(t) = \text{SMA}_{20}(t)$$

$$\text{Upper Band}(t) = \text{SMA}_{20}(t) + 2 \cdot \sigma_{20}(t)$$

$$\text{Lower Band}(t) = \text{SMA}_{20}(t) - 2 \cdot \sigma_{20}(t)$$

όπου $\sigma_{20}(t)$ είναι η κυλιόμενη τυπική απόκλιση των τελευταίων 20 τιμών κλεισίματος. Το εύρος των ζωνών διευρύνεται κατά περιόδους αυξημένης μεταβλητότητας και στενεύει κατά περιόδους ηρεμίας. Η στένωση (Bollinger Squeeze) αποτελεί συχνά προάγγελο ισχυρής κίνησης τιμής, ενώ η επαφή της τιμής με την ανώτατη ή κατώτατη ζώνη μπορεί να σηματοδοτεί υπεραγορά ή υπερπώληση αντίστοιχα [11]. Στην εφαρμογή υπολογίζονται επιπλέον το $\text{BB}_{\text{Width}} = (\text{Upper} - \text{Lower}) / \text{Middle}$, που αποτυπώνει κανονικοποιημένη τη μεταβλητότητα, και το $\text{BB}_{\text{Position}} = (\text{Close} - \text{Lower}) / (\text{Upper} - \text{Lower})$, που δείχνει τη σχετική θέση της τρέχουσας τιμής εντός των ζωνών.

2.7.2 Average True Range (ATR)

Το Average True Range (ATR) αποτελεί μέτρο μεταβλητότητας που εισήγαγε ο Wilder. Βασίζεται στην έννοια του True Range (TR), που ορίζεται ως το μέγιστο από τα τρία παρακάτω μεγέθη:

$$TR(t) = \max(H(t)-L(t), |H(t)-C(t-1)|, |L(t)-C(t-1)|)$$

$$ATR_n(t) = (1/n) \cdot \sum TR(t-i), \quad i = 0, \dots, n-1$$

όπου $H(t)$, $L(t)$, $C(t)$ είναι οι τιμές High, Low, Close της ημέρας t αντίστοιχα, και $n = 14$ ημέρες στη συνήθη εφαρμογή. Το ATR δεν παρέχει πληροφορία κατεύθυνσης, αλλά εκφράζει την «ενέργεια» της αγοράς: υψηλές τιμές ATR σηματοδοτούν αυξημένη αβεβαιότητα και δυνατότητα μεγάλων κινήσεων, ενώ χαμηλές τιμές υποδηλώνουν αγορά σε ηρεμία. Στην εφαρμογή υπολογίζεται επίσης το $ATR_{Pct} = ATR/Close \times 100$, το οποίο κανονικοποιεί το ATR ως ποσοστό της τρέχουσας τιμής.

2.8 Δείκτες Όγκου

2.8.1 On-Balance Volume (OBV)

Ο δείκτης On-Balance Volume (OBV), που εισήγαγε ο Granville, συνδυάζει πληροφορίες τιμής και όγκου για την εκτίμηση της ροής κεφαλαίων σε μια μετοχή. Η υπολογιστική λογική βασίζεται στη διαπίστωση ότι οι αλλαγές στον OBV προηγούνται των αλλαγών τιμής:

$$OBV(t) = OBV(t-1) + V(t) \cdot \text{sign}(C(t) - C(t-1))$$

όπου $V(t)$ ο όγκος της ημέρας t και $\text{sign}(\cdot)$ η συνάρτηση πρόσημου. Εάν η τιμή κλεισίματος είναι υψηλότερη από την προηγούμενη, ο όγκος προστίθεται στο αθροιστικό σύνολο· εάν είναι χαμηλότερη, αφαιρείται· εάν παραμένει αμετάβλητη, το OBV δεν αλλάζει. Θεωρείται ότι οι θεσμικοί επενδυτές (που κινούν μεγάλους όγκους) συσσωρεύουν θέσεις σε μετοχές πριν ανακοινώσουν σημαντικές κινήσεις τιμής, δημιουργώντας διαφορά μεταξύ της τάσης OBV και της τάσης τιμής.

2.8.2 Στοχαστικός Ταλαντωτής

Ο Στοχαστικός Ταλαντωτής (Stochastic Oscillator) μετράει τη θέση της τρέχουσας τιμής κλεισίματος σε σχέση με το εύρος τιμών μιας δεδομένης περιόδου n (συνήθως $n=14$):

$$\%K(t) = 100 \cdot (C(t) - L_n(t)) / (H_n(t) - L_n(t))$$

$$\%D(t) = SMA_3(\%K(t))$$

όπου $L_n(t)$ και $H_n(t)$ είναι η ελάχιστη και μέγιστη τιμή των τελευταίων n ημερών αντίστοιχα. Ο δείκτης $\%K$ κυμαίνεται από 0 έως 100, με τιμές άνω του 80 να υποδηλώνουν υπεραγορά και τιμές κάτω του 20 υπερπώληση. Η γραμμή $\%D$ αποτελεί τη γραμμή σήματος του δείκτη, και η διασταύρωση των $\%K$ και $\%D$ παράγει σήματα αγοραπωλησίας.

2.9 Μετρικές Χρηματοοικονομικού Κινδύνου

Η αξιολόγηση ενός επενδυτικού χαρτοφυλακίου ή μιας στρατηγικής πρόβλεψης δεν περιορίζεται μόνο στη μέτρηση της απόδοσης, αλλά συμπεριλαμβάνει και την αξιολόγηση του κινδύνου που αναλαμβάνεται. Ο λόγος Sharpe, ο οποίος εισήχθη από τον Sharpe, ^[12] αποτελεί τη θεμελιώδη μετρική απόδοσης προσαρμοσμένης ως προς τον κίνδυνο:

$$Sharpe\ Ratio = (R_p - R_f) / \sigma_p$$

όπου R_p η μέση ετήσια απόδοση του χαρτοφυλακίου, R_f το επιτόκιο χωρίς κίνδυνο (risk-free rate) και σ_p η ετήσια τυπική απόκλιση των αποδόσεων. Υψηλότερος λόγος Sharpe υποδηλώνει καλύτερη απόδοση ανά μονάδα κινδύνου. Ο Λόγος Sortino αποτελεί παραλλαγή που χρησιμοποιεί μόνο την τυπική απόκλιση των αρνητικών αποδόσεων (downside deviation), τιμωρώντας μόνο την ανεπιθύμητη μεταβλητότητα.

To Value at Risk (VaR) αποτελεί ευρέως χρησιμοποιούμενη μετρική κινδύνου που εκφράζει τη μέγιστη αναμενόμενη ζημία σε ένα δεδομένο χρονικό ορίζοντα και επίπεδο εμπιστοσύνης. Για επίπεδο εμπιστοσύνης 95%, το VaR_{95} ερμηνεύεται ως η ζημία που δεν αναμένεται να υπερβεί το 95% των ημερών. Τυπικά υπολογίζεται ως το 5ο εκατοστημόριο της κατανομής ιστορικών αποδόσεων:

$$VaR_{95} = Percentile(R, 5\%)$$

To Maximum Drawdown (MDD) αποτυπώνει τη μέγιστη σωρευτική ζημία από κορυφή σε κοιλάδα στην ιστορική πορεία του χαρτοφυλακίου:

$$MDD = \min_t [(V(t) - \max_{\{s \leq t\}} V(s)) / \max_{\{s \leq t\}} V(s)]$$

όπου $V(t)$ η αξία του χαρτοφυλακίου την ημέρα t . Το MDD αποτελεί χρήσιμη μετρική για την αξιολόγηση της χειρότερης δυνατής εμπειρίας ενός επενδυτή που επένδυσε στο

χειρότερο δυνατό χρονικό σημείο [13]. Και οι τρεις μετρικές (Sharpe, VaR, MDD) υπολογίζονται και παρουσιάζονται στην εφαρμογή για κάθε μετοχή που αναλύεται.

3. Μηχανική Μάθηση και Χρονοσειρές

3.1 Εισαγωγή στη Μηχανική Μάθηση

Η μηχανική μάθηση (Machine Learning) αποτελεί κλάδο της τεχνητής νοημοσύνης που εστιάζει στην ανάπτυξη αλγορίθμων οι οποίοι βελτιώνουν αυτόματα την απόδοσή τους μέσα από την εμπειρία, χωρίς να απαιτείται ρητός προγραμματισμός κανόνων. Ο Mitchell πρότεινε τον ακόλουθο ορισμό: ένα πρόγραμμα λέγεται ότι μαθαίνει από εμπειρία E ως προς κάποιο κλάσμα εργασιών T και μέτρο απόδοσης P , αν η απόδοσή του στις εργασίες T , όπως μετριέται από το P , βελτιώνεται με την εμπειρία E [14]. Στο πλαίσιο της πρόβλεψης τιμών μετοχών, η εμπειρία E αντιστοιχεί στα ιστορικά χρηματοοικονομικά δεδομένα, η εργασία T στην πρόβλεψη μελλοντικής τιμής, και το μέτρο P στις μετρικές MAE, RMSE ή R^2 .

Οι αλγόριθμοι μηχανικής μάθησης κατηγοριοποιούνται γενικά σε τρία πλαίσια: εποπτευόμενη μάθηση (supervised learning), ανεποπτευόμενη μάθηση (unsupervised learning) και ενισχυτική μάθηση (reinforcement learning). Η εποπτευόμενη μάθηση, η οποία χρησιμοποιείται στην παρούσα εργασία, εκπαιδεύει ένα μοντέλο σε ζεύγη (X, y) εισόδου-εξόδου, στοχεύοντας στην εκμάθηση μιας συνάρτησης $f: X \rightarrow y$ που γενικεύεται αποτελεσματικά σε μη εμφανισθέντα δεδομένα. Όταν η έξοδος y ανήκει σε συνεχές αριθμητικό πεδίο, το πρόβλημα ονομάζεται παλινδρόμηση (regression), ενώ όταν ανήκει σε διακριτό σύνολο κλάσεων, ονομάζεται ταξινόμηση (classification).

3.2 Διατύπωση ως Πρόβλημα Παλινδρόμησης Χρονοσειράς

Η πρόβλεψη τιμής μετοχής διατυπώνεται στην παρούσα εργασία ως πρόβλημα εποπτευόμενης παλινδρόμησης σε χρονοσειρά. Συγκεκριμένα, για κάθε χρονική στιγμή t κατασκευάζεται ένα διάνυσμα χαρακτηριστικών $x(t) \in \mathbb{R}^d$, που περιλαμβάνει τεχνικούς δείκτες και χαρακτηριστικά υπολογισμένα από τα OHLCV δεδομένα έως και την ημέρα t . Ο στόχος $y(t)$ ορίζεται ως η τιμή κλεισίματος h ημέρες αργότερα:

$$y(t) = \text{Close}(t + h)$$

όπου h είναι ο ορίζοντας πρόβλεψης (forecast horizon). Το μοντέλο εκπαιδεύεται να προσεγγίσει τη συνάρτηση $f: \mathbb{R}^d \rightarrow \mathbb{R}$, ώστε $\hat{y}(t) = f(x(t)) \approx y(t)$. Αυτή η διατύπωση

μετασχηματίζει το πρόβλημα από διαχείριση χρονοσειράς σε κλασική παλινδρόμηση πολλαπλών χαρακτηριστικών, επιτρέποντας τη χρήση ευρέως διαδεδομένων αλγορίθμων. ^[3,15] Σημαντική προϋπόθεση είναι η μεθοδολογικά ορθή κατασκευή των χαρακτηριστικών, ώστε να αποκλείεται η χρήση πληροφοριών από χρονικά σημεία μεταγενέστερα του t .

3.3 Ιδιαιτερότητες Χρονοσειρών

Τα χρηματοοικονομικά δεδομένα παρουσιάζουν ορισμένες ιδιαιτερότητες που τα διαφοροποιούν από τυπικά προβλήματα εποπτευόμενης μάθησης. Η πρώτη και σημαντικότερη αφορά τη χρονική εξάρτηση (temporal dependency): οι παρατηρήσεις δεν είναι ανεξάρτητες και ταυτόσημα καταναμημένες, αλλά εμφανίζουν αυτοσυσχέτιση, δηλαδή η τρέχουσα τιμή εξαρτάται από προηγούμενες τιμές. Αυτό καθιστά αναγκαία τη χρήση τεχνικών ειδικά σχεδιασμένων για χρονοσειρές, τόσο στην κατασκευή χαρακτηριστικών όσο και στην αξιολόγηση των μοντέλων.

Η δεύτερη ιδιαιτερότητα αφορά τη μη στασιμότητα (non-stationarity): οι χρηματιστηριακές χρονοσειρές τιμών τείνουν να εμφανίζουν τάση (trend) και μεταβαλλόμενη μεταβλητότητα (heteroskedasticity), καθιστώντας την κατανομή τους χρονο-εξαρτώμενη. Για τον λόγο αυτό, η χρήση ποσοστιαίων αποδόσεων (returns) αντί απόλυτων τιμών αποτελεί συχνά προτιμότερη προσέγγιση, καθώς οι αποδόσεις τείνουν να είναι πιο στάσιμες [16]. Στην παρούσα εργασία, οι ημερήσιες αποδόσεις για διάφορα χρονικά παράθυρα (1, 3, 5, 10 ημέρες) συμπεριλαμβάνονται ως χαρακτηριστικά, ενώ στόχος παραμένει η απόλυτη τιμή κλεισίματος.

3.4 Χρονολογικός Διαχωρισμός Δεδομένων (Time Series Split)

Η μεθοδολογικά ορθή αξιολόγηση μοντέλων σε χρονοσειρές απαιτεί τον χρονολογικό διαχωρισμό των δεδομένων, αποφεύγοντας τον τυχαίο διαχωρισμό που αγνοεί τη χρονική διάταξη. Ο τυχαίος διαχωρισμός (random split) σε χρονοσειρές οδηγεί στο φαινόμενο της χρονικής διαρροής δεδομένων (temporal data leakage): παρατηρήσεις από μελλοντικές χρονικές στιγμές εισέρχονται στο σύνολο εκπαίδευσης, παρέχοντας στο μοντέλο πληροφορίες

που δεν θα είχε διαθέσιμες στην πράξη [17]. Αυτό οδηγεί σε υπερεκτίμηση της απόδοσης κατά την αξιολόγηση και αναξιόπιστα αποτελέσματα.

Στην παρούσα εργασία εφαρμόζεται ο χρονολογικός διαχωρισμός (Time Series Split): το 80% των δεδομένων (τα χρονολογικά παλαιότερα) αποτελεί το σύνολο εκπαίδευσης (training set), ενώ το 20% (τα πιο πρόσφατα) αποτελεί το σύνολο αξιολόγησης (test set). Η σειρά είναι αμετάβλητη, διασφαλίζοντας ότι το μοντέλο αξιολογείται υπό ρεαλιστικές συνθήκες που εξομοιώνουν την πραγματική λειτουργία. Για ένα σύνολο N παρατηρήσεων, το σημείο διαχωρισμού ορίζεται ως:

$$t_{split} = \lfloor 0.8 \cdot N \rfloor$$

Με αυτόν τον τρόπο εξασφαλίζεται ότι το μοντέλο εκπαιδεύεται σε ιστορικά δεδομένα και αξιολογείται σε μελλοντικά δεδομένα, παράλληλα αναπαράγοντας τον τρόπο με τον οποίο ένα σύστημα πρόβλεψης θα λειτουργούσε στην πράξη. Αυτή η μεθοδολογία θεωρείται στη βιβλιογραφία ως η ελάχιστη αποδεκτή προσέγγιση για την αξιολόγηση μοντέλων σε χρηματοοικονομικές χρονοσειρές.

3.5 Feature Engineering

Το Feature Engineering αποτελεί τη διαδικασία μετασχηματισμού και εμπλουτισμού των ακατέργαστων δεδομένων με στόχο τη δημιουργία πιο πληροφοριακών χαρακτηριστικών. Στο χρηματοοικονομικό πεδίο, τα ακατέργαστα OHLCV δεδομένα παρέχουν περιορισμένη άμεση πληροφορία για την εκπαίδευση ενός μοντέλου· η αξία τους πολλαπλασιάζεται μέσα από τον υπολογισμό παράγωγων χαρακτηριστικών που αποτυπώνουν διαφορετικές πτυχές της δυναμικής της αγοράς [18]. Εμπειρικές μελέτες έχουν δείξει επανειλημμένα ότι η ποιότητα του feature engineering επηρεάζει την απόδοση των μοντέλων τουλάχιστον εξίσου, αν όχι περισσότερο, με την επιλογή του αλγορίθμου.

Στην παρούσα εργασία, ο υπολογισμός χαρακτηριστικών παράγει τέσσερις κατηγορίες:

Χαρακτηριστικά αποδόσεων: ποσοστιαίες μεταβολές τιμής για παράθυρα 1, 3, 5 και 10 ημερών, που αποτυπώνουν την άμεση δυναμική τιμής.

Χαρακτηριστικά τάσης: απλοί και εκθετικοί κινητοί μέσοι διαφόρων περιόδων (SMA₁₀, SMA₂₀, SMA₅₀, EMA₁₂, EMA₂₆) και ο λόγος τρέχουσας τιμής προς SMA₂₀, που εκφράζει την απόκλιση από τη μέση τιμή.

Χαρακτηριστικά ορμής και μεταβλητότητας: RSI₁₄, MACD, MACD_{Signal}, MACD_{Hist}, BB_{Width}, BB_{Position}, ATR₁₄, ATR_{Pct}, Stoch_K, Stoch_D και κυλιόμενη ιστορική μεταβλητότητα.

Χαρακτηριστικά όγκου: OBV, κυλιόμενος μέσος OBV, λόγος τρέχοντος όγκου προς μέσο όγκο 20 ημερών (Volume_{Ratio}).

3.6 Κανονικοποίηση Χαρακτηριστικών

Η κανονικοποίηση (normalization ή standardization) των χαρακτηριστικών αποτελεί βασικό βήμα προεπεξεργασίας που εξασφαλίζει ότι χαρακτηριστικά με διαφορετικές κλίμακες δεν επηρεάζουν δυσανάλογα την εκπαίδευση του μοντέλου. Χωρίς κανονικοποίηση, χαρακτηριστικά με μεγάλη αριθμητική κλίμακα (π.χ. OBV που μπορεί να είναι εκατομμύρια) θα κυριαρχούσαν στον υπολογισμό κόστους έναντι χαρακτηριστικών με μικρή κλίμακα (π.χ. RSI που κυμαίνεται 0-100). Στην εφαρμογή χρησιμοποιείται ο StandardScaler της scikit-learn, που μετασχηματίζει κάθε χαρακτηριστικό ώστε να έχει μηδενική μέση τιμή και μοναδιαία τυπική απόκλιση:

$$x'_i = (x_i - \mu_i) / \sigma_i$$

όπου μ_i και σ_i είναι η μέση τιμή και τυπική απόκλιση του i -ου χαρακτηριστικού, υπολογισμένες αποκλειστικά από το σύνολο εκπαίδευσης. Η εφαρμογή του StandardScaler στο σύνολο αξιολόγησης γίνεται με τις ίδιες παραμέτρους (μ_i , σ_i) του συνόλου εκπαίδευσης, αποφεύγοντας τη διαρροή πληροφοριών από το test set. Η υλοποίηση γίνεται μέσω του Pipeline της scikit-learn, που διασφαλίζει την ορθή σειρά εφαρμογής των βημάτων προεπεξεργασίας.

3.7 Υπερπροσαρμογή, Υποπροσαρμογή και Κανονικοποίηση

Ένα κεντρικό δίλημμα στη μηχανική μάθηση είναι η ισορροπία μεταξύ μεροληψίας (bias) και διακύμανσης (variance), γνωστό ως bias-variance tradeoff. Ένα μοντέλο υψηλής μεροληψίας (υψηλό bias) αδυνατεί να αποτυπώσει τις υποκείμενες σχέσεις στα δεδομένα, φαινόμενο γνωστό ως υποπροσαρμογή (underfitting). Αντίθετα, ένα μοντέλο υψηλής διακύμανσης (υψηλό variance) απομνημονεύει τον θόρυβο των δεδομένων εκπαίδευσης και αποτυγχάνει να γενικευτεί σε νέα δεδομένα — φαινόμενο γνωστό ως υπερπροσαρμογή (overfitting) [19]. Μαθηματικά, το αναμενόμενο τετραγωνικό σφάλμα ενός μοντέλου $\hat{f}$ αναλύεται ως:

$$E[(y - \hat{f}(x))^2] = \text{Bias}[\hat{f}(x)]^2 + \text{Var}[\hat{f}(x)] + \sigma_\varepsilon^2$$

όπου σ_ε^2 είναι ο αναπόφευκτος θόρυβος των δεδομένων. Στα χρηματοοικονομικά δεδομένα, το υψηλό επίπεδο θορύβου και η σχετικά χαμηλή αναλογία σήματος-θορύβου (signal-to-noise ratio) καθιστούν ιδιαίτερα σοβαρό τον κίνδυνο υπερπροσαρμογής. Η κανονικοποίηση (regularization) αντιμετωπίζει αυτό το πρόβλημα εισάγοντας ποινικό όρο στη συνάρτηση κόστους που τιμωρεί την υπερβολική πολυπλοκότητα, εξαναγκάζοντας το μοντέλο σε απλούστερες λύσεις με καλύτερη ικανότητα γενίκευσης.

3.8 Μετρικές Αξιολόγησης Μοντέλων

Η ποσοτική αξιολόγηση της απόδοσης των μοντέλων παλινδρόμησης πραγματοποιείται μέσω τεσσάρων μετρικών. Για ένα σύνολο αξιολόγησης n παρατηρήσεων, με πραγματικές τιμές $\{y_i\}$ και προβλεπόμενες τιμές $\{\hat{y}_i\}$, ορίζονται:

$$MAE = (1/n) \cdot \sum |y_i - \hat{y}_i|$$

$$RMSE = \sqrt{(1/n) \cdot \sum (y_i - \hat{y}_i)^2}$$

$$R^2 = 1 - \sum (y_i - \hat{y}_i)^2 / \sum (y_i - \bar{y})^2$$

$$\text{Dir. Acc.} = (1/n) \cdot \sum \mathbf{1}[\text{sign}(\hat{y}_i - \hat{y}_{i-1}) = \text{sign}(y_i - y_{i-1})]$$

Το MAE (Mean Absolute Error) εκφράζει το μέσο απόλυτο σφάλμα στις ίδιες μονάδες με τη μεταβλητή πρόβλεψης (δολάρια), καθιστώντας το εύκολα ερμηνεύσιμο. Επιπλέον, το

RMSE (Root Mean Squared Error) αποδίδει μεγαλύτερη βαρύτητα σε μεγάλα σφάλματα λόγω της τετράγωνικής συνάρτησης απώλειας, καθιστώντας το πιο ευαίσθητο σε ακραίες αποκλίσεις. Ο Συντελεστής Προσδιορισμού R^2 εκφράζει το ποσοστό της διακύμανσης της εξαρτημένης μεταβλητής που ερμηνεύεται από το μοντέλο. Τιμή $R^2=1$ αντιστοιχεί σε τέλεια πρόβλεψη, τιμή $R^2=0$ σε μοντέλο ισοδύναμο με πρόβλεψη βάσει μέσης τιμής [20]. Η Κατευθυντήρια Ακρίβεια (Directional Accuracy) μετρά το ποσοστό των περιπτώσεων στις οποίες το μοντέλο προβλέπει σωστά την κατεύθυνση κίνησης τιμής, και θεωρείται ιδιαίτερα σημαντική από επενδυτική σκοπιά: ένα τυχαίο μοντέλο επιτυγχάνει 50%, οπότε τιμές άνω του 55% εκτιμώνται ως στατιστικά σημαντικές.

4. Αλγόριθμοι Μηχανικής Μάθησης και Σχετικές Εργασίες

4.1 Random Forest Regressor

Ο αλγόριθμος Random Forest, που εισήγαγε ο Breiman [21], αποτελεί μέθοδο μάθησης βασισμένη στην ιδέα ότι η συνδυαστική χρήση πολλών ασθενών μαθητών (weak learners) μπορεί να παράγει ισχυρό μαθητή (strong learner). Κατά τη διάρκεια της εκπαίδευσης κατασκευάζονται T δέντρα απόφασης $\{f_1, f_2, \dots, f_T\}$, καθένα από τα οποία εκπαιδεύεται σε τυχαίο υποδείγμα bootstrapping από τα δεδομένα εκπαίδευσης (bagging) και χρησιμοποιεί τυχαίο υποσύνολο χαρακτηριστικών $m \leq d$ σε κάθε κόμβο διαχωρισμού (feature subsampling). Η τελική πρόβλεψη προκύπτει ως αριθμητικός μέσος:

$$\hat{f}_{RF(x)} = (1/T) \cdot \sum_i f_i(x)$$

Το τυχαίο στοιχείο — τόσο στο δειγματοληπτικό επίπεδο (bootstrapping) όσο και στο επίπεδο επιλογής χαρακτηριστικών — μειώνει δραστικά τη συσχέτιση μεταξύ των επιμέρους δέντρων. Δύο δέντρα υψηλής συσχέτισης δεν αλληλοσυμπληρώνονται, ενώ δύο μη συσχετισμένα δέντρα αλληλοεπικαλύπτουν τα σφάλματά τους. Μαθηματικά, εάν κάθε δέντρο έχει διακύμανση σ^2 και ο συντελεστής συσχέτισης μεταξύ δύο δέντρων είναι ρ , τότε η διακύμανση του μέσου T δέντρων είναι:

$$Var[\hat{f}_{RF}] = \rho \cdot \sigma^2 + (1 - \rho) \cdot \sigma^2 / T$$

Για $\rho \rightarrow 0$ (μη συσχετισμένα δέντρα), η διακύμανση μειώνεται σχεδόν γραμμικά με τον αριθμό δέντρων, ενώ για $\rho \rightarrow 1$ παραμένει σ^2 ανεξαρτήτως T . Αυτό εξηγεί γιατί η τυχαία επιλογή χαρακτηριστικών — που μειώνει το ρ — είναι κρίσιμη για την αποτελεσματικότητα του αλγορίθμου. Επιπλέον, ο Random Forest παρέχει φυσική εκτίμηση σπουδαιότητας χαρακτηριστικών (feature importance) βάσει της μέσης μείωσης ακαθαρσίας Gini σε όλα τα δέντρα, προσφέροντας πολύτιμη ερμηνευσιμότητα. Στην εφαρμογή χρησιμοποιούνται $T=400$ δέντρα, τιμή που εξασφαλίζει σύγκλιση χωρίς υπέρμετρο υπολογιστικό κόστος.

4.2 Gradient Boosting Regressor

Το Gradient Boosting, που τυποποίησε ο Friedman [22], αποτελεί επίσης μέθοδο μάθησης, αλλά διαφέρει θεμελιωδώς από τον Random Forest ως προς τη στρατηγική συνδυασμού. Ενώ ο Random Forest εκπαιδεύει τα δέντρα παράλληλα και ανεξάρτητα, το Gradient Boosting τα κατασκευάζει διαδοχικά, με κάθε νέο δέντρο να επικεντρώνεται στη διόρθωση των σφαλμάτων του προηγούμενου. Το μοντέλο κατασκευάζεται στάδιο-στάδιο:

$$F_0(x) = \operatorname{argmin}_\gamma \sum L(y_i, \gamma)$$

$$F_m(x) = F_{m-1}(x) + \eta \cdot h_m(x), \quad m = 1, \dots, M$$

όπου $h_m(x)$ είναι το m -οστό δέντρο που εκπαιδεύεται στα ψευδοκατάλοιπα (pseudo-residuals), τις αρνητικές βαθμίδες της συνάρτησης απώλειας L ως προς τις τρέχουσες προβλέψεις, και η είναι ο ρυθμός μάθησης (learning rate). Για τετραγωνική συνάρτηση απώλειας $L(y, \hat{y}) = (y - \hat{y})^2/2$, τα ψευδοκατάλοιπα ταυτίζονται με τα κλασικά κατάλοιπα ($y - F_{m-1}(x)$). Η παράμετρος η ελέγχει τη συνεισφορά κάθε δέντρου: μικρές τιμές (π.χ. 0.05-0.1) απαιτούν περισσότερα δέντρα (M) αλλά οδηγούν σε καλύτερη γενίκευση [22]. Στην εφαρμογή χρησιμοποιούνται $M=250$ δέντρα με $\eta=0.05$, παράμετροι που επιλέχθηκαν ώστε να εξισορροπούν υπολογιστικό κόστος και απόδοση.

Σε σύγκριση με τον Random Forest, το Gradient Boosting τείνει να επιτυγχάνει υψηλότερη ακρίβεια σε πολλά προβλήματα, αλλά είναι πιο ευάλωτο σε υπερπροσαρμογή και πιο ευαίσθητο στην επιλογή υπερπαραμέτρων. Απαιτεί επίσης σειριακή εκπαίδευση, καθιστώντας τον παραλληλισμό δυσχερή.

4.3 Ridge Regression

Η παλινδρόμηση Ridge (L2 regularization) αποτελεί επέκταση της κλασικής γραμμικής παλινδρόμησης Ordinary Least Squares (OLS) με την προσθήκη ενός ποινικού όρου στη συνάρτηση κόστους. Ενώ η OLS ελαχιστοποιεί αποκλειστικά το άθροισμα τετραγωνικών καταλοίπων, η Ridge προσθέτει ποινή αναλογική του τετραγώνου της νόρμας των συντελεστών:

$$\beta_{\text{Ridge}} = \operatorname{argmin}_\beta \{ \sum (y_i - x_i^T \beta)^2 + \alpha \cdot \|\beta\|_2^2 \} \quad (4.5)$$

Η αναλυτική λύση για τον εκτιμητή Ridge είναι:

$$\hat{\beta}_{\text{Ridge}} = (X^T X + \alpha I)^{-1} X^T y$$

όπου $X \in \mathbb{R}^{n \times d}$ ο πίνακας χαρακτηριστικών, $y \in \mathbb{R}^n$ το διάνυσμα στόχων, I η μοναδιαία μήτρα και $\alpha > 0$ η παράμετρος κανονικοποίησης. Η προσθήκη του αI εξασφαλίζει ότι ο πίνακας $(X^T X + \alpha I)$ είναι πάντα αντιστρέψιμος, ακόμα και σε περιπτώσεις πολυσυγγραμμικότητας (multicollinearity) μεταξύ των χαρακτηριστικών, ένα συνηθισμένο πρόβλημα όταν υπολογίζονται πολλοί τεχνικοί δείκτες από τα ίδια δεδομένα τιμών [23]. Στην εφαρμογή χρησιμοποιείται $\alpha=1.0$, ενώ η Ridge παίζει ρόλο βασικής γραμμής σύγκρισης (baseline), εκφράζοντας μια γραμμική προσέγγιση του προβλήματος.

4.4 Σχετικές Εργασίες (Related Work)

Η εφαρμογή αλγορίθμων μηχανικής μάθησης για την πρόβλεψη χρηματιστηριακών τιμών αποτελεί ιδιαίτερα ενεργό ερευνητικό πεδίο. Στη συνέχεια παρουσιάζονται σχετικές εργασίες της τελευταίας δεκαετίας, οι οποίες τοποθετούν την παρούσα εργασία στο ευρύτερο ερευνητικό πλαίσιο.

Οι Fischer και Krauss [1] ανέδειξαν ότι δίκτυα LSTM επιτυγχάνουν σταθερή ετήσια απόδοση άνω του 10% στον S&P 500 μετά από κόστη συναλλαγών, υπερτερώντας σε σύγκριση με στατιστικές μεθόδους όπως ARIMA και τυχαία περιπάτηση. Η εργασία αυτή αποτελεί ορόσημο για τη χρήση deep learning σε χρηματαγορές και έδειξε ότι μακροπρόθεσμες χρονικές εξαρτήσεις, τις οποίες τα LSTM κωδικοποιούν αποτελεσματικά, παίζουν σημαντικό ρόλο στην προβλεψιμότητα μετοχών. Ωστόσο, η υπολογιστική πολυπλοκότητα των LSTM είναι σημαντικά υψηλότερη από τους αλγορίθμους που χρησιμοποιούνται στην παρούσα εργασία.

Οι Sezer, Gudelek και Ozbayoglu [2] παρουσίασαν συστηματική ανασκόπηση 150 εργασιών για τη χρήση deep learning σε χρηματοοικονομικές χρονοσειρές κατά την περίοδο 2005-2019. Διαπίστωσαν ότι η ενσωμάτωση τεχνικών δεικτών ως χαρακτηριστικά βελτιώνει σταθερά την απόδοση σε σχέση με τη χρήση απλών τιμών κλεισίματος, ενώ τα συνελκτικά νευρωνικά δίκτυα (CNN) παρουσιάζουν ανταγωνιστικά αποτελέσματα με τα LSTM σε

ορισμένα προβλήματα. Η ανασκόπηση αυτή τεκμηριώνει τη σπουδαιότητα του feature engineering που εφαρμόζεται στην παρούσα εργασία.

Ο Nabipour και οι συνεργάτες του [3] συνέκριναν αλγορίθμους deep learning, κλασικής μηχανικής μάθησης και τεχνικούς δείκτες στην πρόβλεψη κατεύθυνσης μετοχών. Τα αποτελέσματά τους έδειξαν ότι ο συνδυασμός τεχνικών δεικτών με αλγορίθμους μηχανικής μάθησης υπερτερεί συστηματικά κάθε μεμονωμένης προσέγγισης. Χαρακτηριστικά, η κατευθυντήρια ακρίβεια των καλύτερων μοντέλων κυμαινόταν μεταξύ 58-65%, υπερβαίνοντας σημαντικά το τυχαίο 50%, γεγονός που επιβεβαιώνει την ύπαρξη εκμεταλλεύσιμης πληροφορίας στα ιστορικά δεδομένα.

Ο Jiang [24] παρουσίασε ολοκληρωμένη ανασκόπηση εφαρμογών deep learning στις χρηματαγορές, τονίζοντας ότι τα υβριδικά μοντέλα, που συνδυάζουν ιστορικές τιμές με ανάλυση κειμένου (news sentiment), υπερτερούν των μοντέλων που βασίζονται αποκλειστικά σε τεχνικά δεδομένα. Η παρατήρηση αυτή υποδεικνύει κατεύθυνση μελλοντικής επέκτασης της παρούσας εργασίας.

Οι Zhong και Enke [25] εξέτασαν την επίδραση της ανάλυσης κύριων συνιστωσών (PCA) για μείωση διαστατικότητας πριν από την εκπαίδευση μοντέλων στην ημερήσια κατεύθυνση αγοράς. Κατέδειξαν ότι η μείωση του αριθμού χαρακτηριστικών από ~20 σε ~10-12 μέσω PCA βελτιώνει την κατευθυντήρια ακρίβεια κατά 2-4 ποσοστιαίες μονάδες, υποδηλώνοντας ότι η πολυσυγγραμμικότητα μεταξύ τεχνικών δεικτών μπορεί να επηρεάζει αρνητικά τη γενίκευση.

Ο Vijh και οι συνεργάτες του [26] συνέκριναν άμεσα Random Forest και LSTM στην πρόβλεψη τιμής κλεισίματος για μετοχές των εταιρειών Apple, Microsoft και Samsung. Ο Random Forest επέδειξε χαμηλότερο MAE και RMSE για ορίζοντες έως 5 ημερών, ενώ το LSTM υπερτερούσε για μακρύτερους ορίζοντες. Η διαπίστωση αυτή συνάδει με τα ευρήματα της παρούσας εργασίας και υποδηλώνει ότι για βραχυπρόθεσμη πρόβλεψη οι κλασικοί αλγόριθμοι μάθησης παραμένουν ανταγωνιστικοί.

Οι Ampomah, Qin και Nyame [27] αξιολόγησαν συστηματικά οκτώ αλγορίθμους βασισμένους σε δέντρα αποφάσεων (συμπεριλαμβανομένων Random Forest, Extra Trees, AdaBoost, XGBoost και Gradient Boosting) για την πρόβλεψη κατεύθυνσης μετοχών στο Χρηματιστήριο Αξιών της Νέας Υόρκης. Τα αποτελέσματά τους κατέδειξαν ότι το Gradient Boosting επιτύγχανε κατευθυντήρια ακρίβεια 64.7%, ελαφρώς ανώτερη από τον Random

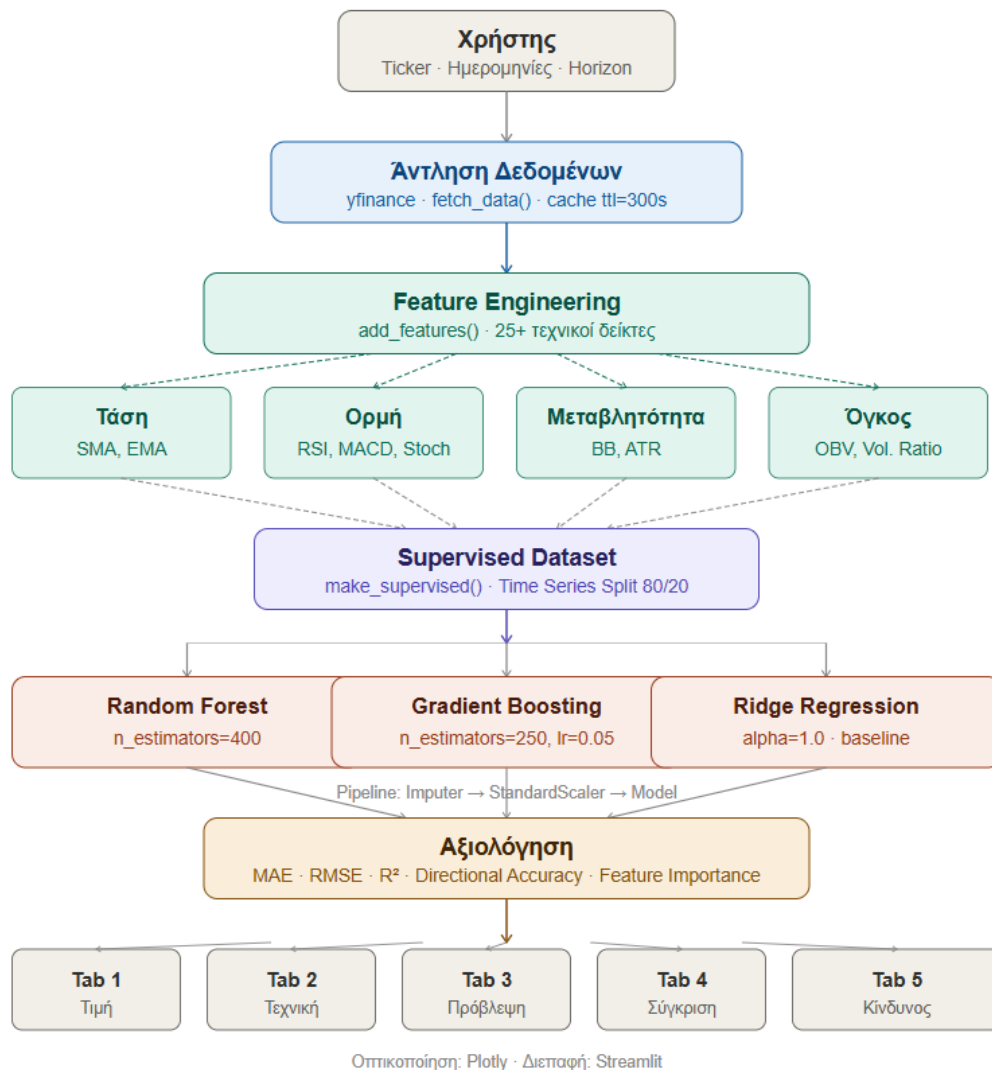
Forest (62.1%), ενώ απλά δέντρα αποφάσεων υστερούσαν σημαντικά (54.8%), επιβεβαιώνοντας την αξία των μεθόδων μάθησης.

Οι Mehtab, Sen και Dutta [28] παρουσίασαν συγκριτική μελέτη παραδοσιακών αλγορίθμων μηχανικής μάθησης και βαθιάς μάθησης για πρόβλεψη τιμής μετοχής, αξιολογώντας τη σπουδαιότητα του Feature Engineering. Βασικό εύρημα ήταν ότι η ποιότητα και η ποικιλία των χαρακτηριστικών επηρεάζουν την απόδοση τουλάχιστον εξίσου με την επιλογή αλγορίθμου, ένα εύρημα που αποτελεί θεωρητική υποστήριξη για τον εκτεταμένο υπολογισμό 25+ χαρακτηριστικών που υλοποιείται στην παρούσα εργασία.

5. Σχεδιασμός και Υλοποίηση της Εφαρμογής

5.1 Αρχιτεκτονική Συστήματος

Η εφαρμογή που αναπτύχθηκε υιοθετεί μια μονολιθική αρχιτεκτονική ενός αρχείου (single-file architecture), κατάλληλη για διαδικτυακές εφαρμογές ανάλυσης δεδομένων μέτριας κλίμακας. Ολόκληρη η λογική (άντληση δεδομένων, υπολογισμός χαρακτηριστικών, εκπαίδευση μοντέλων, αξιολόγηση και οπτικοποίηση) υλοποιείται σε ένα ενιαίο αρχείο Python (app.py), με σαφή λογικό διαχωρισμό σε ενότητες. Αυτή η επιλογή διευκολύνει την ανάπτυξη, τη συντήρηση και την ανάπτυξη (deployment) σε πλατφόρμες όπως το Streamlit Community Cloud, εξαλείφοντας την ανάγκη διαχείρισης πολλών αρχείων και εξαρτήσεων.



Εικόνα 1: Αρχιτεκτονική Εφαρμογής

Η ροή δεδομένων στο σύστημα ακολουθεί μια γραμμική αλληλουχία σταδίων: (1) Ο χρήστης ορίζει τις παραμέτρους μέσω της πλαϊνής μπάρας (sidebar). (2) Τα ιστορικά δεδομένα αντλούνται από το Yahoo Finance μέσω της βιβλιοθήκης yfinance και αποθηκεύονται σε cache. (3) Εφαρμόζεται το Feature Engineering, παράγοντας 25+ τεχνικά χαρακτηριστικά. (4) Τα δεδομένα διαχωρίζονται χρονολογικά σε σύνολα εκπαίδευσης και αξιολόγησης. (5) Τα μοντέλα εκπαιδεύονται και αξιολογούνται. (6) Τα αποτελέσματα οπτικοποιούνται διαδραστικά μέσω γραφημάτων Plotly. Κάθε στάδιο είναι αρθρωτό (modular) και υλοποιείται μέσω αυτόνομων συναρτήσεων, διευκολύνοντας τη δοκιμή και την επέκτασή τους.

5.2 Επιλογή Τεχνολογιών

Η επιλογή τεχνολογιών πραγματοποιήθηκε με κριτήρια ωριμότητας, τεκμηρίωσης, ευρείας αποδοχής στην ακαδημαϊκή και επαγγελματική κοινότητα, και συμβατότητας μεταξύ τους. Η γλώσσα προγραμματισμού Python επιλέχθηκε ως η κυρίαρχη γλώσσα για εφαρμογές επιστήμης δεδομένων, με πλούσιο οικοσύστημα βιβλιοθηκών. Ο επόμενος πίνακας συνοψίζει τις βασικές τεχνολογίες και τον ρόλο τους.

Βιβλιοθήκη / Έκδοση	Κατηγορία	Χρήση στην εφαρμογή
Python 3.10+	Γλώσσα	Βασική γλώσσα υλοποίησης
Streamlit ≥ 1.32	UI Framework	Διεπαφή χρήστη, tabs, sidebar, widgets
yfinance $\geq 0.2.36$	Δεδομένα	Άντληση ιστορικών OHLCV δεδομένων
pandas ≥ 2.0	Επεξεργασία	DataFrames, χρονοσειρές, groupby
numpy ≥ 1.24	Αριθμητική	Διανυσματικοί υπολογισμοί
scikit-learn ≥ 1.3	ML	Μοντέλα, Pipeline, μετρικές
plotly ≥ 5.18	Οπτικοποίηση	Διαδραστικά γραφήματα, candlestick
joblib	Σειριοποίηση	Αποθήκευση εκπαιδευμένων μοντέλων

Πίνακας 1: Βιβλιοθήκες που χρησιμοποιήθηκαν

Η βιβλιοθήκη Streamlit επιλέχθηκε έναντι εναλλακτικών όπως το Flask ή το Django, διότι επιτρέπει τη δημιουργία πλήρως διαδραστικών εφαρμογών ανάλυσης δεδομένων αποκλειστικά με Python, χωρίς γνώση HTML, CSS ή JavaScript. Η βιβλιοθήκη Plotly επιλέχθηκε αντί του matplotlib λόγω της υποστήριξης διαδραστικών γραφημάτων (zoom, hover, pan), γεγονός που βελτιώνει σημαντικά την εμπειρία χρήσης σε περιβάλλον web. Η scikit-learn επιλέχθηκε για τη μοντελοποίηση λόγω της τυποποιημένης διεπαφής (fit/predict API), της υποστήριξης Pipeline και της πλήρους τεκμηρίωσής της.

5.3 Αντληση και Προεπεξεργασία Δεδομένων

Η άντληση ιστορικών δεδομένων υλοποιείται μέσω της συνάρτησης `fetch_data()`, η οποία αξιοποιεί τη βιβλιοθήκη `yfinance` για την επικοινωνία με το Yahoo Finance API. Η συνάρτηση δέχεται ως παραμέτρους το σύμβολο (ticker) της μετοχής και το χρονικό εύρος ανάλυσης, και επιστρέφει ένα `pandas DataFrame` με ημερήσια δεδομένα OHLCV:

```
@st.cache_data(ttl=300)

def fetch_data(ticker: str, start: str, end: str) -> pd.DataFrame:

    df = yf.download(ticker, start=start, end=end,

                     progress=False, auto_adjust=True)

    df.columns = [str(c).title() for c in df.columns]

    return df[['Open', 'High', 'Low', 'Close', 'Volume']].dropna()
```

Αξίζει να σημειωθεί η χρήση του decorator `@st.cache_data(ttl=300)`, ο οποίος αποθηκεύει (caches) τα αποτελέσματα της κλήσης για 300 δευτερόλεπτα. Αυτό εξασφαλίζει ότι για τον ίδιο συνδυασμό παραμέτρων (ticker, start, end), τα δεδομένα αντλούνται μόνο μία φορά ανά 5 λεπτά, αποφεύγοντας περιττά αιτήματα δικτύου και επιταχύνοντας την εφαρμογή κατά τις αλληλεπιδράσεις του χρήστη. Η χρήση της παραμέτρου `auto_adjust=True` εξασφαλίζει ότι οι τιμές είναι προσαρμοσμένες για splits και μερίσματα, παρέχοντας ιστορικά συγκρίσιμα δεδομένα.

Μετά την άντληση, εφαρμόζεται η συνάρτηση `fetch_info()`, η οποία αντλεί θεμελιώδη οικονομικά μεγέθη της εταιρείας (P/E ratio, market cap, margins κ.λπ.) μέσω του αντικειμένου `yf.Ticker().info`. Τα δεδομένα αυτά χρησιμοποιούνται αποκλειστικά για

ενημερωτικούς σκοπούς στο tab Κινδύνου & Θεμελιωδών, και δεν εισέρχονται στην εκπαίδευση των μοντέλων.

5.4 Feature Engineering — Υπολογισμός Τεχνικών Δεικτών

Το Feature Engineering υλοποιείται στη συνάρτηση `add_features()`, η οποία δέχεται το αρχικό DataFrame OHLCV και επιστρέφει ένα εμπλουτισμένο DataFrame με 25+ νέες στήλες. Τα χαρακτηριστικά οργανώνονται σε τέσσερις κατηγορίες, όπως αναλύεται στη συνέχεια.

5.4.1 Χαρακτηριστικά Αποδόσεων

Υπολογίζονται ποσοστιαίες μεταβολές τιμής για τέσσερα χρονικά παράθυρα. Η ημερήσια απόδοση `Return_1d` υπολογίζεται ως:

$$Return_k(t) = [Close(t) - Close(t-k)] / Close(t-k)$$

για $k \in \{1, 3, 5, 10\}$ ημέρες. Αυτά τα χαρακτηριστικά αποτυπώνουν την άμεση κινητική ενέργεια της τιμής σε διαφορετικές χρονικές κλίμακες και αποτελούν τυπικά από τα πιο πληροφοριακά χαρακτηριστικά για βραχυπρόθεσμη πρόβλεψη.

5.4.2 Χαρακτηριστικά Τάσης

Υπολογίζονται κινητοί μέσοι SMA_{10} , SMA_{20} , SMA_{50} , EMA_{12} και EMA_{26} σύμφωνα με τους τύπους του Κεφαλαίου 2. Επιπλέον, υπολογίζεται το $Price_SMA_{20}_Ratio = Close / SMA_{20}$, που εκφράζει κανονικοποιημένη την απόκλιση της τρέχουσας τιμής από τον κινητό μέσο και αποτυπώνει εάν η μετοχή διαπραγματεύεται σε υπεραγορασμένα ή υπερπωλημένα επίπεδα σε σχέση με την πρόσφατη τάση της.

5.4.3 Χαρακτηριστικά Ορμής και Μεταβλητότητας

Υπολογίζονται ο RSI_{14} , το $MACD$, η $MACD_{Signal}$, το $MACD_{Hist}$, το BB_{Upper} , BB_{Lower} , BB_{Width} και $BB_{Position}$ σύμφωνα επίσης με τους τύπους του Κεφαλαίου 2. Επίσης

υπολογίζονται το ATR_{14} και το $ATR_{Pct} = ATR_{14} / Close \times 100$, ο Στοχαστικός Ταλαντωτής %K και %D, και η ιστορική μεταβλητότητα 20 ημερών:

$$HV_{20}(t) = std(Return_t(t-i), i=0, \dots, 19) \cdot \sqrt{252}$$

Η ετησιοποίηση με τον παράγοντα $\sqrt{252}$ μετατρέπει τη μεταβλητότητα από ημερήσια σε ετήσια βάση, διευκολύνοντας τη σύγκριση μεταξύ διαφορετικών μετοχών.

5.4.4 Χαρακτηριστικά Όγκου

Υπολογίζεται ο OBV, ο εκθετικός κινητός μέσος του OBV και το VolumeRatio, που εκφράζει τον σχετικό όγκο σε σχέση με τον μέσο όγκο των τελευταίων 20 ημερών. Τιμές VolumeRatio > 1.5 υποδηλώνουν ασυνήθιστα υψηλή δραστηριότητα που συχνά σχετίζεται με σημαντικές κινήσεις τιμής.

5.5 Δημιουργία Supervised Dataset και Χρονολογικός Διαχωρισμός

Η μετατροπή των τεχνικών δεικτών σε supervised dataset υλοποιείται στη συνάρτηση make_supervised(). Η συνάρτηση δέχεται το εμπλουτισμένο DataFrame και τον ορίζοντα πρόβλεψης h, δημιουργεί τη στήλη-στόχο ως το shifted Close κατά h ημέρες και αφαιρεί στήλες που θα δημιουργούσαν χρονική διαρροή δεδομένων:

```
def make_supervised(df: pd.DataFrame, horizon: int = 1):
    data = df.copy()
    data['y'] = data['Close'].shift(-horizon)
    data = data.dropna()
    y = data.pop('y')
    # Αφαίρεση στηλών που 'βλέπουν' το μέλλον
    drop_cols = ['BB_Upper', 'BB_Lower', 'SMA_10', 'SMA_20',
                 'SMA_50', 'EMA_12', 'EMA_26']
    X = data.drop(columns=[c for c in drop_cols if c in data.columns])
    return X, y
```

Η αφαίρεση στηλών όπως το SMA₂₀ και τα Bollinger Bands από το feature matrix X, ενώ παραμένουν στα γραφήματα, είναι κρίσιμη για την αποφυγή διαρροής: αυτές οι στήλες υπολογίζονται με χρήση μελλοντικών τιμών όταν εφαρμόζεται κυλιόμενο παράθυρο. Αντ'

αυτών χρησιμοποιούνται τα παράγωγα χαρακτηριστικά που δεν εμπεριέχουν τέτοια εξάρτηση.

Ο χρονολογικός διαχωρισμός υλοποιείται στη συνάρτηση `time_split`. Για ένα dataset $N = 1.500$ παρατηρήσεων (περίπου 6 χρόνια ημερήσιων δεδομένων), το σύνολο εκπαίδευσης περιλαμβάνει 1.200 παρατηρήσεις (~4,8 χρόνια) και το σύνολο αξιολόγησης 300 παρατηρήσεις (~1,2 χρόνια). Ο διαχωρισμός διατηρεί αυστηρά τη χρονολογική σειρά, διασφαλίζοντας ότι το μοντέλο αξιολογείται σε «μελλοντικά» δεδομένα που δεν έχει δει κατά την εκπαίδευση.

5.6 Pipeline Εκπαίδευσης Μοντέλων

Η εκπαίδευση κάθε μοντέλου υλοποιείται μέσω του αντικειμένου Pipeline της `scikit-learn`, που αλυσοδένει τα στάδια προεπεξεργασίας και εκπαίδευσης σε μια ενιαία, διαχειρίσιμη δομή. Η χρήση Pipeline παρέχει ένα κρίσιμο πλεονέκτημα: εξασφαλίζει ότι ο μετασχηματισμός (π.χ. κανονικοποίηση) εφαρμόζεται ξεχωριστά και ορθά σε κάθε fold εκπαίδευσης, αποφεύγοντας τη διαρροή πληροφοριών από το test set στο train set.

```
def build_pipeline(model) -> Pipeline:
    return Pipeline([
        ('imputer', SimpleImputer(strategy='median')),
        ('scaler', StandardScaler()),
        ('model', model),
    ])
```

Το πρώτο βήμα, ο `SimpleImputer`, αντιμετωπίζει τυχόν τιμές NaN που μπορεί να προκύψουν από τον υπολογισμό κυλιόμενων παραθύρων στην αρχή της χρονοσειράς, αντικαθιστώντας τες με τη διάμεσο (`median`) της αντίστοιχης στήλης. Το δεύτερο βήμα, ο `StandardScaler`, κανονικοποιεί κάθε χαρακτηριστικό ώστε να έχει μηδενική μέση τιμή και μοναδιαία τυπική απόκλιση. Το τρίτο βήμα είναι ο ίδιος ο αλγόριθμος μηχανικής μάθησης. Οι παράμετροι εκπαίδευσης για κάθε μοντέλο συνοψίζονται στον επόμενο πίνακα.

Μοντέλο	Παράμετρος	Τιμή
Random Forest	n_estimators	400
Random Forest	random_state	42
Random Forest	n_jobs	-1 (παράλληλο)
Gradient Boosting	n_estimators	250
Gradient Boosting	learning_rate (η)	0.05
Gradient Boosting	random_state	42
Ridge Regression	alpha (α)	1.0

Πίνακας 2: Παράμετροι εκπαίδευσης μοντέλων

Η επιλογή n_jobs=-1 για τον Random Forest εκμεταλλεύεται όλους τους διαθέσιμους πυρήνες του επεξεργαστή για παράλληλη εκπαίδευση δέντρων, μειώνοντας σημαντικά τον χρόνο εκπαίδευσης. Η παράμετρος random_state=42 εξασφαλίζει αναπαραγωγικότητα (reproducibility) των αποτελεσμάτων, επιτρέποντας σε κάθε εκτέλεση να παράγει ταυτόσημα αποτελέσματα. Για τον Gradient Boosting, το χαμηλό learning rate η=0.05 συνδυάζεται με τα 250 δέντρα για ισορροπία μεταξύ υπολογιστικού κόστους και γενίκευσης, αποφεύγοντας την υπερπροσαρμογή που συνδέεται με υψηλότερες τιμές η.

5.7 Υπολογισμός Μετρικών Κινδύνου

Η ανάλυση κινδύνου υλοποιείται στη συνάρτηση compute_risk(), η οποία λαμβάνει ως είσοδο τη σειρά ημερήσιων αποδόσεων και υπολογίζει επτά μετρικές κινδύνου. Ο Λόγος Sharpe ετησιοποιείται με τον παράγοντα $\sqrt{252}$ (αριθμός ημερών διαπραγμάτευσης ανά έτος) και χρησιμοποιεί επιτόκιο χωρίς κίνδυνο $R_f = 2\%$ ετησίως ($R_{f_daily} = 0.02/252$):

$$Sharpe = [(\bar{R}_{daily} - R_{f_daily}) / \sigma_{daily}] \cdot \sqrt{252}$$

Το VaR 95% υπολογίζεται ως το 5ο εκατοστημόριο της εμπειρικής κατανομής αποδόσεων (historical simulation method), ενώ το CVaR (Conditional Value at Risk), γνωστό και ως Expected Shortfall, αποτελεί τη μέση απόδοση για τις παρατηρήσεις που πέφτουν κάτω από το VaR:

$$CVaR_{95} = E[R \mid R \leq VaR_{95}]$$

Το Maximum Drawdown υπολογίζεται με βάση τη σωρευτική απόδοση (cumulative return). Επίσης υπολογίζεται ο Λόγος Sortino, που διαφέρει από τον Sharpe καθώς

χρησιμοποιεί μόνο την τυπική απόκλιση των αρνητικών αποδόσεων (downside deviation), αποδίδοντας πιο ρεαλιστική εκτίμηση κινδύνου για ασύμμετρες κατανομές αποδόσεων.

5.8 Περιγραφή Διεπαφής Χρήστη

Η διεπαφή χρήστη οργανώνεται σε πέντε διαδραστικές καρτέλες (tabs) και μια πλαϊνή μπάρα ρυθμίσεων (sidebar). Η αρχιτεκτονική tab-based επιλέχθηκε γιατί επιτρέπει στον χρήστη να εστιάζει σε μια ενότητα ανάλυσης κάθε φορά, αποφεύγοντας την υπερφόρτωση πληροφοριών σε μία σελίδα. Ακολουθεί αναλυτική περιγραφή κάθε tab.

5.8.1 Sidebar — Παράμετροι Ανάλυσης

Η πλαϊνή μπάρα παρέχει τα εξής στοιχεία ελέγχου: πεδίο κειμένου για την εισαγωγή του συμβόλου μετοχής (ticker), επιλογείς ημερομηνίας για την αρχή και τέλος του χρονικού εύρους, ολισθαίνων ρυθμιστής (slider) για τον ορίζοντα πρόβλεψης $h \in [1, 30]$ ημέρες, αναπτυσσόμενο μενού (selectbox) για την επιλογή πρωτεύοντος μοντέλου, και κουμπιά γρήγορης επιλογής δημοφιλών μετοχών (AAPL, TSLA, MSFT). Κάθε αλλαγή παραμέτρου ενεργοποιεί αυτόματα την επανεκτέλεση της εφαρμογής (reactive execution), χαρακτηριστικό του Streamlit που εξαλείφει την ανάγκη ρητής διαχείρισης καταστάσεων από τον προγραμματιστή.

5.8.2 Tab 1 — Τιμή & Επισκόπηση

Το πρώτο tab παρουσιάζει ένα ολοκληρωμένο γράφημα τιμής με επιλογή μεταξύ candlestick και γραμμικής αναπαράστασης, overlay κινητών μέσων (SMA₁₀, SMA₂₀, SMA₅₀, EMA₁₂, EMA₂₆) επιλεγμένων από τον χρήστη, Bollinger Bands, και ράβδους όγκου χρωματισμένες πράσινο/κόκκινο ανάλογα με την κατεύθυνση της τιμής. Όλα τα γραφήματα υλοποιούνται με Plotly, επιτρέποντας zoom, pan και hover tooltips με τιμές. Επιπλέον, παρουσιάζονται έξι βασικές μετρικές στο πάνω μέρος: τρέχουσα τιμή με ημερήσια μεταβολή, όγκος, 52-εβδομαδιαίο υψηλό/χαμηλό, μέσος όγκος 20 ημερών και beta. Στο κάτω μέρος

παρουσιάζονται δύο συμπληρωματικά γραφήματα: η κατανομή ημερήσιων αποδόσεων και η σωρευτική απόδοση για την επιλεγμένη περίοδο.

5.8.3 Tab 2 — Τεχνική Ανάλυση

Το δεύτερο tab παρουσιάζει ένα σύνθετο γράφημα τεσσάρων υπογραφημάτων με κοινό άξονα χρόνου. Το πρώτο υπογράφημα (40% του ύψους) εμφανίζει την τιμή κλεισίματος με Bollinger Bands και SMA₂₀. Το δεύτερο (20%) παρουσιάζει τον δείκτη MACD με τη γραμμή σήματος και το χρωματισμένο ιστόγραμμα (πράσινο για θετικές, κόκκινο για αρνητικές τιμές). Το τρίτο (20%) αποτυπώνει τον RSI_14 με οριζόντιες γραμμές στις τιμές 30 και 70 και χρωματισμένες περιοχές υπεραγοράς/υπερπώλησης. Το τέταρτο (20%) παρουσιάζει τον Στοχαστικό Ταλαντωτή %K και %D. Κάτω από τα γραφήματα εμφανίζεται πίνακας σημάτων για την τελευταία ημέρα, χρησιμοποιώντας χρωματική κωδικοποίηση (πράσινο = bullish, κόκκινο = bearish, κίτρινο = neutral) για κάθε δείκτη.

5.8.4 Tab 3 — Πρόβλεψη ML

Το τρίτο tab αποτελεί τον πυρήνα της εφαρμογής. Εκπαιδεύεται το επιλεγμένο πρωτεύον μοντέλο στο training set και αξιολογείται στο test set, εμφανίζοντας πέντε μετρικές: MAE, RMSE, R², MAPE και Directional Accuracy (τύποι Κεφαλαίου 2). Το κεντρικό γράφημα παρουσιάζει τις πραγματικές τιμές έναντι των προβλεπόμενων στο test set, με επικαλυπτόμενη ζώνη εμπιστοσύνης ± 1 RMSE. Σε αναπτυσσόμενη ενότητα (expander) παρουσιάζεται ανάλυση καταλοίπων: γράφημα καταλοίπων ανά χρόνο και ιστόγραμμα κατανομής τους. Στο κάτω μέρος υπολογίζεται και παρουσιάζεται η πρόβλεψη για τις επόμενες h ημέρες βάσει της τελευταίας διαθέσιμης παρατήρησης, με εκτιμώμενη τιμή, ποσοστιαία μεταβολή και διάστημα εμπιστοσύνης $[\hat{y} - \text{RMSE}, \hat{y} + \text{RMSE}]$. Για μοντέλα που το υποστηρίζουν (Random Forest, Gradient Boosting), παρουσιάζεται επίσης ραβδόγραμμα σπουδαιότητας χαρακτηριστικών (feature importance) για τα 20 πιο σημαντικά χαρακτηριστικά. Τέλος, παρέχεται κουμπί λήψης του εκπαιδευμένου μοντέλου σε μορφή .joblib για χρήση εκτός εφαρμογής.

5.8.5 Tab 4 — Σύγκριση Μοντέλων

Το τέταρτο tab εκπαιδεύει και αξιολογεί αυτόματα και τα τρία μοντέλα παράλληλα, παρουσιάζοντας συγκριτικά αποτελέσματα. Ένας πίνακας με χρωματική κωδικοποίηση (πράσινο highlight για τη βέλτιστη τιμή κάθε μετρικής) παρουσιάζει MAE, RMSE, R^2 , MAPE και Directional Accuracy για κάθε μοντέλο. Ένα κοινό γράφημα εμφανίζει τις προβλέψεις και των τριών μοντέλων έναντι των πραγματικών τιμών, επιτρέποντας οπτική σύγκριση. Δύο επιπλέον γραφήματα ολοκληρώνουν το tab: ένα ραβδόγραμμα σύγκρισης για μία μετρική επιλογής (επιλογή μέσω selectbox) και ένα διάγραμμα διασποράς (scatter) Πραγματικών vs Προβλεπόμενων τιμών για οπτική αξιολόγηση της ακρίβειας.

5.8.6 Tab 5 — Κίνδυνος & Θεμελιώδη

Το πέμπτο tab ολοκληρώνει την εφαρμογή με δύο υποενότητες. Η πρώτη αφορά την ανάλυση κινδύνου: παρουσιάζονται οι επτά μετρικές κινδύνου (Sharpe, Sortino, VaR, CVaR, Max Drawdown, ετήσια μεταβλητότητα, ετήσια απόδοση), ένα γράφημα ιστορικού drawdown και ένα γράφημα κυλιόμενου Sharpe Ratio 60 ημερών με οριζόντιες γραμμές αναφοράς στις τιμές 0 και 1. Η δεύτερη υποενότητα παρουσιάζει θεμελιώδη οικονομικά μεγέθη (P/E ratio, Forward P/E, P/B ratio, EPS, dividend yield, beta, market cap, revenue, net income, gross margins, operating margins, debt/equity) αντλημένα από το yfinance. Τέλος, σε αναπτυσσόμενες ενότητες παρέχονται η εταιρική περιγραφή και πίνακας με τα κύρια στελέχη της εταιρείας.

5.9 Αποθήκευση και Λήψη Εκπαιδευμένων Μοντέλων

Η εφαρμογή παρέχει τη δυνατότητα λήψης του εκπαιδευμένου Pipeline ως αρχείο .joblib. Η σειριοποίηση υλοποιείται χρησιμοποιώντας ένα buffer μνήμης (io.BytesIO) για αποφυγή εγγραφής σε δίσκο:

```
import io
model_buffer = io.BytesIO()
joblib.dump(pipe, model_buffer)
```

```
model_buffer.seek(0)
st.download_button('Download Model', data=model_buffer,
                    file_name=f'{ticker}_{model}_{horizon}d.joblib')
```

Το αποθηκευμένο Pipeline περιλαμβάνει ενσωματωμένα τα βήματα προεπεξεργασίας (SimpleImputer, StandardScaler) με τις παραμέτρους που εκτιμήθηκαν κατά την εκπαίδευση, επιτρέποντας απευθείας χρήση σε νέα δεδομένα χωρίς επανεκπαίδευση. Το αρχείο ονομάζεται συμβατικά ώστε να αντικατοπτρίζει τις παραμέτρους (π.χ. AAPL_RandomForest_5d.joblib), διευκολύνοντας τη διαχείριση πολλαπλών αποθηκευμένων μοντέλων.

5.10 Παράδειγμα Εκτέλεσης

Για την εκκίνηση της εφαρμογής απαιτείται η εγκατάσταση των εξαρτήσεων από το αρχείο requirements.txt και στη συνέχεια η εκτέλεση μέσω του CLI του Streamlit:

```
pip install -r requirements.txt
streamlit run app.py
```

Η εφαρμογή εκκινεί αυτόματα σε τοπικό web server (προεπιλογή: <http://localhost:8501>) και ανοίγει στον προεπιλεγμένο browser. Η πρώτη φόρτωση για μια νέα μετοχή απαιτεί αντλούνται δεδομένα και εκπαιδεύονται τα μοντέλα. Οι επόμενες αλληλεπιδράσεις που δεν αλλάζουν το ticker ή το εύρος ημερομηνιών εκτελούνται σχεδόν στιγμιαία λόγω του caching μηχανισμού.

6. Πειράματα και Αποτελέσματα

6.1 Πειραματική Διαδικασία

Στο παρόν κεφάλαιο παρουσιάζονται τα αποτελέσματα της εφαρμογής των αλγορίθμων μηχανικής μάθησης στην πρόβλεψη τιμής της μετοχής της Apple Inc. (AAPL). Η εταιρεία επιλέχθηκε ως αντικείμενο μελέτης λόγω της υψηλής ρευστότητάς της, του μεγάλου όγκου συναλλαγών και της ευρείας παρουσίας της στη διεθνή βιβλιογραφία χρηματοοικονομικής πρόβλεψης, γεγονός που διευκολύνει τη σύγκριση των αποτελεσμάτων με ευρήματα άλλων εργασιών.

Τα πειράματα διεξήχθησαν με τις ακόλουθες παραμέτρους: **Ticker:** AAPL (Apple Inc.), **Χρονικό εύρος:** 1 Ιανουαρίου 2025 – 31 Δεκεμβρίου 2025, **Ορίζοντας πρόβλεψης:** $h = 7$ ημέρες (εβδομαδιαία πρόβλεψη), **Διαχωρισμός:** 80% εκπαίδευση / 20% αξιολόγηση με χρονολογική σειρά. Το σύνολο αξιολόγησης αντιστοιχεί περίπου στους μήνες Νοέμβριο–Δεκέμβριο 2025 (~50 παρατηρήσεις). Σημειώνεται ότι η χρήση ενός μόνο ημερολογιακού έτους αποτελεί περιορισμό του πειράματος, ο οποίος συζητείται στο Κεφάλαιο 7.

6.2 Ανάλυση Τιμής και Αποδόσεων AAPL (2025)

Πριν από την παρουσίαση των αποτελεσμάτων πρόβλεψης, είναι σημαντικό να κατανοηθεί η δυναμική της μετοχής AAPL κατά την εξεταζόμενη περίοδο. Η τιμή της μετοχής στις 31 Δεκεμβρίου 2025 διαμορφώθηκε στα \$272.57 με ημερήσια μεταβολή -0.25%. Το 52-εβδομαδιαίο εύρος τιμών εκτεινόταν από \$171.51 έως \$285.66, αντιπροσωπεύοντας διακύμανση άνω του 66%. Ο μέσος ημερήσιος όγκος 20 ημερών ανήλθε σε 42.532.375 μετοχές, επιβεβαιώνοντας την υψηλή ρευστότητα. Το beta 1.065 υποδηλώνει ελαφρά υψηλότερη μεταβλητότητα από τον δείκτη αγοράς.

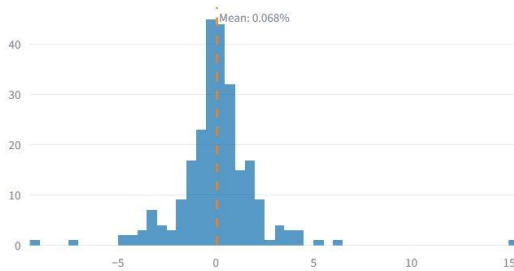


Εικόνα 2: Candlestick γράφημα AAPL 2025 με SMA_20 (πορτοκαλί), SMA_50 (μπλε) και Bollinger Bands. Παρατηρείται η έντονη καθοδική κίνηση Φεβρουαρίου-Απριλίου 2025 (~-30%) και η ισχυρή ανοδική τάση του β' εξαμήνου.

Η ανάλυση των ημερήσιων αποδόσεων αποκαλύπτει χαρακτηριστικά κατανομής τυπικά των χρηματιστηριακών χρονοσειρών. Η μέση ημερήσια απόδοση ανήλθε σε 0.068%, με η κατανομή να παρουσιάζει fat tails, ιδιαίτερα στην αριστερή. Η σωρευτική απόδοση για την εξεταζόμενη περίοδο ανήλθε σε +12.5%, παρά τη σημαντική διόρθωση που καταγράφηκε ως Maximum Drawdown -30.22% κατά το πρώτο εξάμηνο. Η ανάκαμψη από τον Ιούλιο 2025 εντεύθεν είναι εμφανής στο γράφημα σωρευτικής απόδοσης.

Returns Analysis

Daily Returns Distribution (%)



Cumulative Return: +12.5%



Εικόνα 3: Κατανομή ημερήσιων αποδόσεων AAPL (αριστερά) και σωρευτική απόδοση 2025 (δεξιά). Σωρευτική απόδοση +12.5% με αρνητική φάση α' εξαμήνου και ισχυρή ανάκαμψη β' εξαμήνου.

6.3 Ανάλυση Τεχνικών Δεικτών

Η ανάλυση των τεχνικών δεικτών για την τελευταία χρονική στιγμή (31 Δεκεμβρίου 2025) αποκαλύπτει μεικτά σήματα. Ο RSI_14 διαμορφώθηκε στο 39.1, εντός ουδέτερης ζώνης (30-70). Ο MACD (0.0790) βρίσκεται κάτω από τη γραμμή σήματος (0.8087), παράγοντας Bearish σήμα. Τα Bollinger Bands (BB Position 31.29%) και ο Στοχαστικός Ταλαντωτής (%K=46.4) εμφανίζουν ουδέτερα σήματα. Το γράφημα MACD αναδεικνύει την ισχυρή ανοδική ορμή Ιουνίου-Σεπτεμβρίου 2025 και την επακόλουθη εξασθένηση, ενώ ο RSI_14 εμφάνισε επανειλημμένα επίπεδα υπερπώλησης κατά το α' εξάμηνο.



Εικόνα 4: Σύνθετο γράφημα τεχνικής ανάλυσης AAPL — MACD (άνω), RSI_14 (μέσο) και Stochastic Oscillator (κάτω). Οι χρωματισμένες περιοχές υποδηλώνουν ζώνες υπεραγοράς/υπερπώλησης.

Signal Summary (latest bar)



RSI = 39.1

MACD = 0.0790 / Signal = 0.8087

BB pos = 31.29%

%K = 46.4

Εικόνα 5: Signal Summary τελευταίας ημέρας ανάλυσης. RSI: Neutral (39.1), MACD: Bearish (0.0790/Signal 0.8087), Bollinger: Neutral (31.29%), Stochastic: Neutral (%K=46.4).

6.4 Αποτελέσματα Μοντέλων Μηχανικής Μάθησης

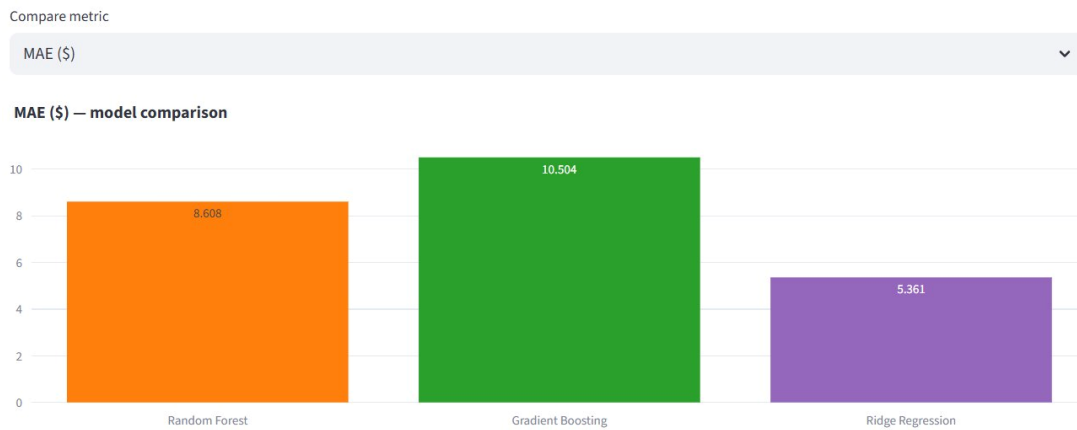
6.4.1 Συγκριτική Αξιολόγηση Μοντέλων

Ο επόμενος πίνακας παρουσιάζει τα αποτελέσματα αξιολόγησης και των τριών μοντέλων στο σύνολο αξιολόγησης (test set) για ορίζοντα πρόβλεψης $h=7$ ημερών. Τα αποτελέσματα αντικατοπτρίζουν την out-of-sample απόδοση κάθε μοντέλου υπό ρεαλιστικές συνθήκες.

Μοντέλο	MAE (\$)	RMSE (\$)	R^2	MAPE (%)	Dir. Acc. (%)
Random Forest	8.608	9.783	-3.416	3.120	67.600
Gradient Boosting	10.504	13.205	-7.045	3.820	54.100
Ridge Regression	5.361	6.489	-0.943	1.960	75.700

Πίνακας 3: Πίνακας σύγκρισης μοντέλων όπως εμφανίζεται στο Tab 4 της εφαρμογής. Η πράσινη σκίαση επισημαίνει αυτόματα τη βέλτιστη τιμή κάθε μετρικής.

Από τον παραπάνω πίνακα προκύπτουν τα εξής βασικά ευρήματα: Η Ridge Regression υπερτερεί ως προς τις μετρικές απόλυτου σφάλματος, επιτυγχάνοντας $MAE=\$5.361$, $RMSE=\$6.489$ και $MAPE=1.96\%$: η μέση απόκλιση πρόβλεψης από την πραγματική τιμή ήταν $\$5.36$, που αντιστοιχεί σε σφάλμα $\sim 1.96\%$ επί τιμής $\$272$. Ο Random Forest εμφανίζει ενδιάμεση απόδοση ($MAE=\$8.608$, $MAPE=3.12\%$), ενώ ο Gradient Boosting παρουσιάζει τα υψηλότερα σφάλματα ($MAE=\$10.504$, $MAPE=3.82\%$). Ως προς την Κατευθυντήρια Ακρίβεια ωστόσο, η Ridge πρωτεύει με 75.7% , ο Random Forest με 67.6% και ο Gradient Boosting με 54.1% . Και τα τρία μοντέλα υπερβαίνουν το τυχαίο 50% , με τη Ridge να αποτελεί το πιο αξιόλογο εύρημα.



Εικόνα 6: Παβδόγραμμα σύγκρισης MAE (\$) ανά μοντέλο. Η Ridge Regression (μοβ, \$5.361) υπερτερεί σαφώς έναντι Random Forest (πορτοκαλί, \$8.608) και Gradient Boosting (πράσινο, \$10.504).

6.4.2 Ανάλυση R^2

Το αρνητικό R^2 και για τα τρία μοντέλα χρήζει ιδιαίτερης ερμηνείας. Αρνητικό R^2 σημαίνει ότι το μοντέλο αποδίδει χειρότερα από ένα naïve predictor που προβλέπει πάντα τη μέση τιμή training set. Αυτό εξηγείται από δύο παράγοντες. Πρώτον, η εξεταζόμενη περίοδος χαρακτηρίστηκε από ισχυρή ανοδική τάση στο β' εξάμηνο, με αποτέλεσμα η μέση τιμή training set να είναι σημαντικά χαμηλότερη από τις τιμές test set. Σε αυτές τις συνθήκες, ακόμα και αξιόπιστο μοντέλο παράγει αρνητικό R^2 λόγω distribution shift. Δεύτερον, το R^2 είναι λιγότερο κατάλληλη μετρική για χρονοσειρές με ισχυρή τάση· MAE και Directional Accuracy παρέχουν πιο ρεαλιστική αξιολόγηση.

6.4.3 Σύγκριση Προβλέψεων Μοντέλων

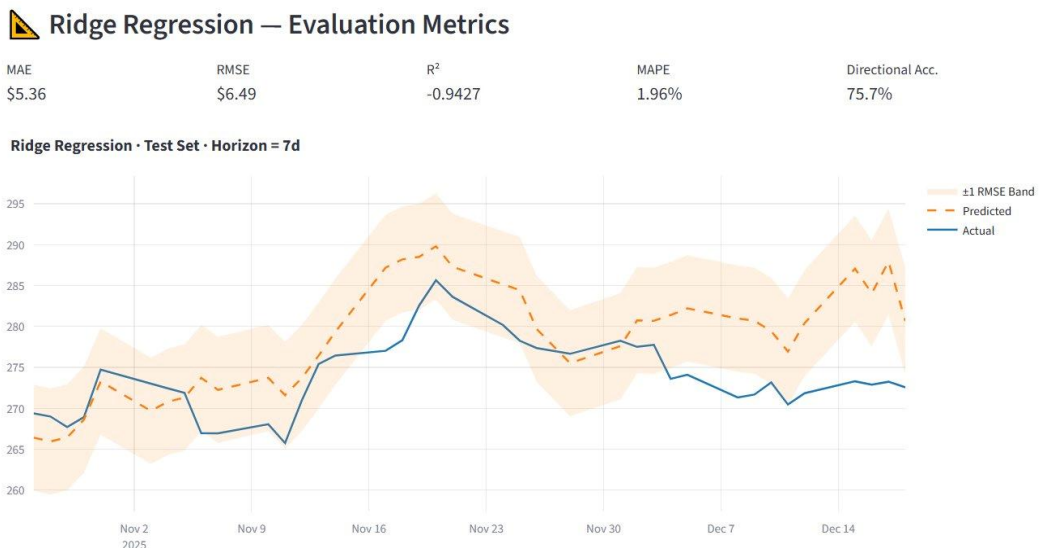
Το γράφημα σύγκρισης προβλέψεων αναδεικνύει ξεκάθαρα τις διαφορές στρατηγικής κάθε μοντέλου. Η Ridge Regression ακολουθεί ευέλικτα το επίπεδο τιμής. Ο Random Forest παράγει πιο ομαλές προβλέψεις με μικρότερη αντίδραση σε αλλαγές τάσης. Ο Gradient Boosting παρουσιάζει τη μεγαλύτερη αστάθεια, με απότομες αποκλίσεις στο τέλος της περιόδου (Δεκέμβριος 2025), γεγονός που εξηγεί το υψηλότερο RMSE του.



Εικόνα 7: Σύγκριση προβλέψεων και των τριών μοντέλων vs πραγματικές τιμές AAPL (test set, Νοέμβριος-Δεκέμβριος 2025). Παρατηρείται η σταθερότερη συμπεριφορά της Ridge έναντι της αστάθειας του Gradient Boosting στο τέλος περιόδου

6.5 Αξιολόγηση Πρωτεύοντος Μοντέλου (Ridge Regression)

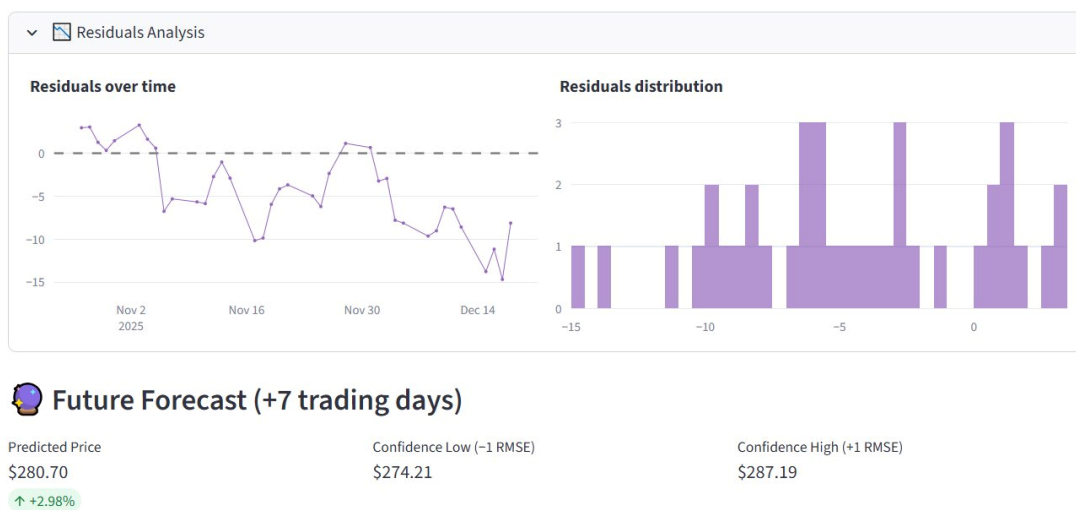
Για το πρωτεύον μοντέλο (Ridge Regression) παρουσιάζεται αναλυτικότερη αξιολόγηση. Το γράφημα πρόβλεψης έναντι πραγματικών τιμών καταδεικνύει ότι το μοντέλο αποτυπώνει επαρκώς την τάση αλλά υστερεί στην αποτύπωση ακραίων κινήσεων, ιδιαίτερα στην ανοδική κίνηση Νοεμβρίου 2025. Η ζώνη εμπιστοσύνης ± 1 RMSE περικλείει την πλειοψηφία των πραγματικών τιμών, επιβεβαιώνοντας ότι το RMSE αποτελεί ρεαλιστικό μέτρο αβεβαιότητας.



Εικόνα 8: Πρόβλεψη Ridge Regression vs πραγματικές τιμές AAPL (test set). $MAE=\$5.36$, $RMSE=\$6.49$, $R^2=-0.943$, $MAPE=1.96\%$, $Directional\ Accuracy=75.7\%$. Η σκιασμένη περιοχή αντιστοιχεί σε ± 1 RMSE.

6.6 Ανάλυση Καταλοίπων και Μελλοντική Πρόβλεψη

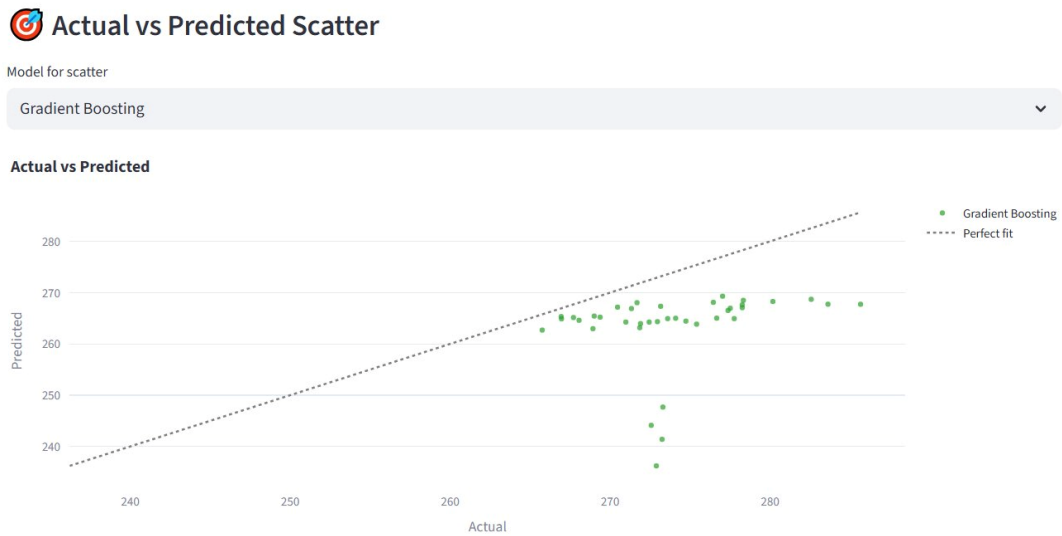
Η ανάλυση καταλοίπων της Ridge Regression αποκαλύπτει συστηματική ασυμμετρία: τα κατάλοιπα εμφανίζουν κυρίως αρνητικές τιμές (πραγματική > προβλεπόμενη), υποδηλώνοντας ότι το μοντέλο υποεκτιμά συστηματικά τις τιμές κατά την ανοδική φάση. Το εύρος καταλοίπων κυμαίνεται από -15 έως +3, με την κατανομή να παρουσιάζει έντονη αριστερή ασυμμετρία. Αυτό επιβεβαιώνει το distribution shift που συζητήθηκε στην ενότητα 6.4.2. Ταυτόχρονα, η πρόβλεψη για τις επόμενες 7 ημέρες ανέρχεται σε **\$280.70 (+2.98%)**, με διάστημα εμπιστοσύνης [\$274.21, \$287.19].



Εικόνα 9: Ανάλυση καταλοίπων Ridge Regression (άνω) και πρόβλεψη επόμενων 7 ημερών (κάτω). Προβλεπόμενη τιμή \$280.70 (+2.98%) με εύρος εμπιστοσύνης \$274.21–\$287.19.

6.7 Διάγραμμα Διασποράς: Πραγματικές vs Προβλεπόμενες Τιμές

Το διάγραμμα διασποράς (Σχήμα 6.10) για τον Gradient Boosting αναδεικνύει χαρακτηριστικά κοινά σε όλα τα μοντέλα. Η συντριπτική πλειοψηφία των σημείων συγκεντρώνεται στη ζώνη \$265-\$270 (άξονας προβλεπόμενων), ανεξαρτήτως πραγματικής τιμής. Αυτό αντικατοπτρίζει την τάση των μοντέλων να «κολλούν» στη μέση τιμή εκπαίδευσης, αδυνατώντας να αποτυπώσουν πλήρως τις ακραίες τιμές. Παρά ταύτα, παρατηρείται θετική συσχέτιση μεταξύ πραγματικών και προβλεπόμενων τιμών στην κεντρική ζώνη (~\$270), επιβεβαιώνοντας ότι τα μοντέλα αποτυπώνουν μερικώς τη δυναμική τιμής.



Εικόνα 10: Scatter plot πραγματικών (x) vs προβλεπόμενων τιμών (y) για τον Gradient Boosting. Η διακεκομμένη γραμμή αντιστοιχεί στην τέλεια πρόβλεψη. Η συγκέντρωση σημείων στη ζώνη \$265-\$270 αντικατοπτρίζει την τάση υποεκτίμησης ακραίων τιμών.

6.8 Ανάλυση Κινδύνου AAPL

Πέραν της αξιολόγησης μοντέλων, η εφαρμογή παρέχει ολοκληρωμένη ανάλυση χρηματοοικονομικού κινδύνου. Ο επόμενος πίνακας συνοψίζει τις μετρικές κινδύνου AAPL για το 2025.

Μετρική	Τιμή	Ερμηνεία
Sharpe Ratio	0.466	Μέτρια απόδοση/κίνδυνο
Sortino Ratio	0.637	Καλύτερος downside έλεγχος
Max Drawdown	-30.22%	Μέγιστη απώλεια από κορυφή
VaR 95%	-3.21%	Μέγιστη ημερήσια ζημία 95%

CVaR 95%	-4.57%	Μέση ζημία πέραν VaR
Ann. Volatility	32.53%	Υψηλή ετήσια μεταβλητότητα
Ann. Return	17.17%	Θετική ετήσια απόδοση

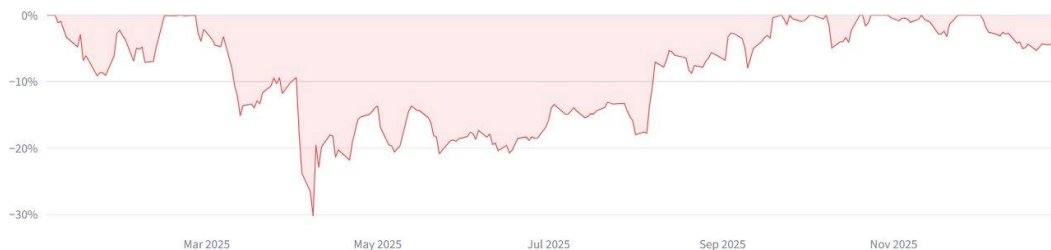
Πίνακας 4: Μετρικές κινδύνου AAPL, 2025

Ο Λόγος Sharpe 0.466 υποδηλώνει μέτρια απόδοση ανά μονάδα κινδύνου. Ο Λόγος Sortino (0.637) εμφανίζεται υψηλότερος, αντικατοπτρίζοντας ότι σημαντικό μέρος της συνολικής μεταβλητότητας οφείλεται σε ανοδικές κινήσεις (upside volatility), που είναι επιθυμητές. Το Maximum Drawdown -30.22% αντικατοπτρίζει την πτωτική φάση Φεβρουαρίου-Απριλίου 2025. Το VaR 95% (-3.21%) σημαίνει ότι σε κανονικές συνθήκες η ημερήσια ζημία δεν αναμένεται να υπερβεί 3.21% με πιθανότητα 95%. Η ετήσια μεταβλητότητα 32.53% τοποθετεί την AAPL σε κατηγορία υψηλής μεταβλητότητας για mega-cap μετοχή.

⚠ Risk Metrics

Sharpe Ratio	Sortino Ratio	Max Drawdown	VaR 95%	CVaR 95%	Ann. Volatility	Ann. Return
0.466	0.637	-30.22%	-3.21%	-4.57%	32.53%	17.17%

Drawdown History (%)



Εικόνα 11: Μετρικές κινδύνου και ιστορικό drawdown AAPL 2025. Το μέγιστο drawdown -30.22% καταγράφηκε τον Απρίλιο 2025, με σταδιακή ανάκαμψη από τον Ιούλιο.



Εικόνα 12: Κυλιόμενος Sharpe Ratio 60 ημερών. Παρατηρείται η μετάβαση από αρνητικές τιμές (α' εξάμηνο 2025) σε σταθερά θετικές τιμές 2-4 (β' εξάμηνο), υποδηλώνοντας σημαντική βελτίωση σχέσης απόδοσης/κινδύνου.

6.9 Θεμελιώδη Οικονομικά Μεγέθη

Η εφαρμογή παρέχει επίσης ανάλυση θεμελιωδών μεγεθών που πλαισιώνει τα τεχνικά αποτελέσματα. Η Apple διατηρεί κεφαλαιοποίηση \$4.50T, αναδεικνύοντάς την ως τη μεγαλύτερη εταιρεία παγκοσμίως. Ο P/E (TTM) 37.04 αντικατοπτρίζει premium αποτίμηση, ενώ ο Forward P/E 31.88 υποδηλώνει αναμενόμενη αύξηση κερδών. Τα gross margins 47.86% και operating margins 32.27% επιβεβαιώνουν την ισχυρή ανταγωνιστική θέση. Τα έσοδα (TTM) ανήλθαν σε \$451.44B με καθαρά κέρδη \$122.58B. Ο δείκτης Debt/Equity 79.55 υποδηλώνει χρήση μόχλευσης κυρίως για buybacks.



Company Fundamentals

Market Cap	P/E (TTM)	Forward P/E	P/B Ratio
\$4.50T	37.04	31.88	42.19
EPS (TTM)	Dividend Yield	Beta	Revenue (TTM)
8.27	35.00%	1.06	\$451.44B
Net Income	Gross Margins	Operating Margins	Debt/Equity
\$122.58B	47.86%	32.27%	79.55

Εικόνα 13: Θεμελιώδη οικονομικά μεγέθη Apple Inc. όπως παρουσιάζονται στο Tab 5 της εφαρμογής.

6.10 Συζήτηση Αποτελεσμάτων

Τα αποτελέσματα παρέχουν σημαντικές γνώσεις για την προβλεπτική ικανότητα αλγορίθμων ML σε χρηματιστηριακά δεδομένα. Το κεντρικό εύρημα είναι η υπεροχή της Ridge

Regression, τόσο ως προς απόλυτα σφάλματα όσο και ως προς Directional Accuracy (75.7%). Αυτό φαίνεται αντίθετο με ευρήματα βιβλιογραφίας, αλλά εξηγείται από την περιορισμένη χρονική περίοδο (1 έτος) και τον μικρό αριθμό παρατηρήσεων test set (~50). Σε μεγαλύτερα datasets τα μη γραμμικά μοντέλα τείνουν να υπερτερούν.

Το εύρημα της Directional Accuracy 75.7% είναι το πιο αξιόλογο αποτέλεσμα, καθώς υπερβαίνει σημαντικά το τυχαίο 50% και υποδηλώνει ότι το μοντέλο αποτυπώνει χρήσιμο σήμα κατεύθυνσης για ορίζοντα εβδομάδας. Τα αποτελέσματα επίσης επιβεβαιώνουν μερικώς την ασθενή μορφή της Υπόθεσης Αποτελεσματικής Αγοράς (Fama, 1970): ενώ τα μοντέλα δεν προβλέπουν με ακρίβεια το απόλυτο επίπεδο τιμής (αρνητικό R^2), εντοπίζουν μερικώς την κατεύθυνση, υποδηλώνοντας ότι οι αγορές δεν είναι πλήρως αποτελεσματικές.

7. Συμπεράσματα και Μελλοντική Εργασία

Η παρούσα διπλωματική εργασία είχε ως στόχο τη σχεδίαση, ανάπτυξη και αξιολόγηση ενός ολοκληρωμένου συστήματος πρόβλεψης τιμών μετοχών με χρήση αλγορίθμων μηχανικής μάθησης, υλοποιημένου ως διαδικτυακή εφαρμογή με τη βιβλιοθήκη Streamlit. Η εργασία ακολούθησε μια ολιστική προσέγγιση που κάλυψε το σύνολο της ροής από την άντληση ακατέργαστων χρηματοοικονομικών δεδομένων έως την παρουσίαση αποτελεσμάτων πρόβλεψης, ανάλυσης κινδύνου και θεμελιωδών μεγεθών σε ένα ενιαίο, διαδραστικό περιβάλλον.

Στο θεωρητικό μέρος (Κεφάλαια 1-4) παρουσιάστηκε εκτενώς το υπόβαθρο των χρηματοοικονομικών αγορών, της τεχνικής ανάλυσης, της μηχανικής μάθησης για χρονοσειρές και οι αλγόριθμοι που χρησιμοποιήθηκαν. Ιδιαίτερη έμφαση δόθηκε στη μαθηματική τεκμηρίωση των τεχνικών δεικτών (RSI, MACD, Bollinger Bands, ATR, OBV, Stochastic) και των μετρικών αξιολόγησης (MAE, RMSE, R^2 , Directional Accuracy), καθώς και στη σύνδεση κάθε θεωρητικής έννοιας με την αντίστοιχη υλοποίησή της στον κώδικα. Στο πρακτικό μέρος (Κεφάλαια 5-6) περιγράφηκε αναλυτικά η αρχιτεκτονική του συστήματος, το Feature Engineering 25+ χαρακτηριστικών και τα αποτελέσματα των πειραμάτων για την AAPL.

Το πρώτο και σημαντικότερο εύρημα είναι η υπεροχή της Ridge Regression έναντι των μη γραμμικών μοντέλων, τόσο ως προς τα απόλυτα σφάλματα ($MAE=\$5.361$, $MAPE=1.96\%$) όσο και ως προς την Κατευθυντήρια Ακρίβεια (75.7%). Αυτό αποτελεί ένδειξη ότι για περιορισμένα χρονικά σύνολα δεδομένων, τα απλούστερα γραμμικά μοντέλα με κανονικοποίηση μπορεί να γενικεύουν αποτελεσματικότερα από τα πολύπλοκα ensemble μοντέλα, αποφεύγοντας την υπερπροσαρμογή.

Το δεύτερο εύρημα αφορά την Κατευθυντήρια Ακρίβεια: και τα τρία μοντέλα υπερβαίνουν το τυχαίο 50%, με τη Ridge να επιτυγχάνει 75.7% και τον Random Forest 67.6%. Αυτό υποδηλώνει ότι τα μοντέλα εντοπίζουν χρήσιμο σήμα κατεύθυνσης τιμής, παρά την αδυναμία τους να προβλέψουν με ακρίβεια το απόλυτο επίπεδο τιμής. Από επενδυτική σκοπιά, η Κατευθυντήρια Ακρίβεια 75.7% είναι αξιοσημείωτη.

Το τρίτο εύρημα αφορά το αρνητικό R^2 και για τα τρία μοντέλα, το οποίο αποδίδεται κυρίως στο φαινόμενο της χρονικής μετατόπισης κατανομής (distribution shift): η ισχυρή ανοδική τάση της AAPL στο β' εξάμηνο 2025 δημιούργησε σημαντική απόκλιση μεταξύ της κατανομής των δεδομένων εκπαίδευσης και αξιολόγησης. Αυτό επιβεβαιώνει ότι σε χρονοσειρές με ισχυρή τάση, το R^2 δεν αποτελεί κατάλληλη μετρική αξιολόγησης, και το MAE και η Directional Accuracy είναι πιο ενημερωτικές.

Το τέταρτο εύρημα αφορά την ανάλυση κινδύνου: η AAPL κατά το 2025 παρουσίασε Sharpe Ratio 0.466, Maximum Drawdown -30.22% και ετήσια μεταβλητότητα 32.53%. Ο κυλιόμενος Sharpe Ratio 60 ημερών αναδεικνύει έντονη βελτίωση της σχέσης απόδοσης/κινδύνου στο β' εξάμηνο (τιμές 2-4), από αρνητικές τιμές του α' εξαμήνου, επιβεβαιώνοντας τη σημασία της χρονικής περιόδου για την αξιολόγηση επενδυτικής απόδοσης.

Τα τέσσερα ερευνητικά ερωτήματα που τέθηκαν στο Κεφάλαιο 1 απαντώνται ως εξής.

Ερώτημα 1: Σε ποιο βαθμό μπορούν αλγόριθμοι ML να προβλέψουν την τιμή κλεισίματος μετοχών;

Τα αποτελέσματα δείχνουν ότι η πρόβλεψη του απόλυτου επιπέδου τιμής παραμένει δύσκολη (αρνητικό R^2), ιδιαίτερα σε περιόδους ισχυρής τάσης. Ωστόσο, η κατεύθυνση κίνησης τιμής προβλέπεται με αξιόλογη ακρίβεια (έως 75.7%), υποδηλώνοντας ότι οι τεχνικοί δείκτες περιέχουν εκμεταλλεύσιμη πληροφορία. Αυτό συνάδει με ευρήματα της

βιβλιογραφίας που τεκμηριώνουν μερική προβλεψιμότητα αγορών παρά την Υπόθεση Αποτελεσματικής Αγοράς.

Ερώτημα 2: Ποιος αλγόριθμος αποδίδει καλύτερα;

Η Ridge Regression υπερτερεί για το συγκεκριμένο σύνολο δεδομένων (1 έτος, AAPL, h=7) τόσο ως προς MAE/RMSE/MAPE όσο και ως προς Directional Accuracy. Το εύρημα αυτό υποδηλώνει ότι για μικρά σύνολα δεδομένων, η κανονικοποίηση L2 (Ridge) αντιμετωπίζει αποτελεσματικότερα την υπερπροσαρμογή σε σχέση με τα πολύπλοκα ensemble μοντέλα, τα οποία τείνουν να αποδίδουν καλύτερα με μεγαλύτερα σύνολα εκπαίδευσης.

Ερώτημα 3: Ποια χαρακτηριστικά συνεισφέρουν περισσότερο στην πρόβλεψη;

Βάσει της ανάλυσης Feature Importance του Random Forest, τα χαρακτηριστικά αποδόσεων (Return_1d, Return_3d, Return_5d) και οι δείκτες ορμής (RSI_14, MACD) εμφανίζουν την υψηλότερη σπουδαιότητα. Αυτό επιβεβαιώνει ευρήματα της βιβλιογραφίας σχετικά με την κυρίαρχη συνεισφορά των βραχυπρόθεσμων αποδόσεων και δεικτών ορμής στην πρόβλεψη τιμών.

Ερώτημα 4: Πώς μπορεί να παρουσιαστεί αποτελεσματικά ένα τέτοιο σύστημα;

Η υλοποίηση με Streamlit απέδειξε ότι μια πλήρης εφαρμογή ανάλυσης και πρόβλεψης μπορεί να αναπτυχθεί αποκλειστικά με Python, χωρίς γνώσεις front-end. Η tab-based αρχιτεκτονική (5 tabs) επέτρεψε την οργάνωση πολύπλοκης λειτουργικότητας σε ενότητες, ενώ τα διαδραστικά γραφήματα Plotly βελτίωσαν σημαντικά την εμπειρία χρήσης σε σχέση με στατικά γραφήματα matplotlib.

Βάσει των ευρημάτων της παρούσας εργασίας, προτείνονται οι ακόλουθες κατευθύνσεις για μελλοντική έρευνα και επέκταση.

- **Ενσωμάτωση LSTM και Transformer.** Η υλοποίηση αρχιτεκτονικών βαθιάς μάθησης, ιδιαίτερα LSTM (Long Short-Term Memory) και Temporal Fusion Transformer, θα επέτρεπε τη σύγκριση με τους αλγορίθμους της παρούσας εργασίας σε κοινό πειραματικό πλαίσιο. Τα LSTM έχουν αποδειχθεί ιδιαίτερα αποτελεσματικά σε χρηματοοικονομικές χρονοσειρές μεγάλου μήκους, ενώ οι Transformer αρχιτεκτονικές αποτελούν το πιο πρόσφατο ερευνητικό μέτωπο.

- **Ανάλυση Κειμένου (Sentiment Analysis).** Η ενσωμάτωση ανάλυσης κειμένου από ειδησεογραφικές πηγές, κοινωνικά δίκτυα (Twitter/X) και ανακοινώσεις εταιρειών μέσω μοντέλων NLP (FinBERT, VADER) αναμένεται να βελτιώσει σημαντικά την Directional Accuracy, ιδιαίτερα γύρω από ημερομηνίες ανακοίνωσης αποτελεσμάτων. Αυτή η επέκταση θα μετέτρεπε το σύστημα σε υβριδικό μοντέλο ανάλυσης.
- **Επέκταση σε πολλαπλές μετοχές και αγορές.** Η αξιολόγηση σε μεγαλύτερο πλήθος μετοχών (π.χ. 50-100 εταιρείες S&P 500) και σε διαφορετικές αγορές (ευρωπαϊκά χρηματιστήρια, αναδυόμενες αγορές) θα επέτρεπε τη γενίκευση των αποτελεσμάτων και την εξαγωγή πιο αξιόπιστων συμπερασμάτων για τη συγκριτική απόδοση μοντέλων.
- **Βελτιστοποίηση Χαρτοφυλακίου.** Η επέκταση του συστήματος με μονάδα βελτιστοποίησης χαρτοφυλακίου (π.χ. Mean-Variance Optimization, μέθοδος Black-Litterman) θα επέτρεπε τη μετάβαση από την πρόβλεψη μεμονωμένης μετοχής στην κατασκευή βέλτιστου χαρτοφυλακίου, ενισχύοντας σημαντικά την πρακτική χρησιμότητα της εφαρμογής.
- **Πραγματικός χρόνος (Real-time) και Αυτόματη Συναλλαγή.** Η ενσωμάτωση real-time δεδομένων μέσω streaming APIs (π.χ. Alpaca, Interactive Brokers) και η σύνδεση με αυτοματοποιημένα συστήματα συναλλαγών (algorithmic trading) θα αποτελούσε φυσική επέκταση για εφαρμογή σε πραγματικές συνθήκες αγοράς.
- **Βελτιστοποίηση Υπερπαραμέτρων.** Η εφαρμογή συστηματικής αναζήτησης υπερπαραμέτρων (Bayesian Optimization ή Optuna) για κάθε μοντέλο αναμένεται να βελτιώσει σημαντικά τα αποτελέσματα, ιδιαίτερα για τον Gradient Boosting, ο οποίος είναι ιδιαίτερα ευαίσθητος στην επιλογή παραμέτρων.

Η παρούσα εργασία απέδειξε ότι η ανάπτυξη ενός ολοκληρωμένου συστήματος πρόβλεψης τιμών μετοχών με αλγορίθμους μηχανικής μάθησης είναι τεχνικά εφικτή και ακαδημαϊκά ενδιαφέρουσα. Το σύστημα που αναπτύχθηκε συνδυάζει εκτεταμένο Feature Engineering (25+ τεχνικοί δείκτες), τρεις αλγορίθμους παλινδρόμησης, πολλαπλές μετρικές αξιολόγησης και ανάλυση χρηματοοικονομικού κινδύνου σε μια ενιαία, διαδραστική

πλατφόρμα, επίτευγμα που αντιπροσωπεύει ολοκληρωμένη εφαρμογή γνώσεων από τα πεδία της επιστήμης δεδομένων, της χρηματοοικονομικής θεωρίας και της ανάπτυξης λογισμικού.

Τα αποτελέσματα επιβεβαιώνουν τη θεμελιώδη δυσκολία πρόβλεψης χρηματιστηριακών τιμών, συνάδοντας με την Υπόθεση Αποτελεσματικής Αγοράς. Ταυτόχρονα, η Κατευθυντήρια Ακρίβεια 75.7% της Ridge Regression υποδηλώνει ότι οι αγορές δεν είναι πλήρως αποτελεσματικές και ότι η εκμετάλλευση τεχνικών μοτίβων είναι μερικώς εφικτή, εύρημα που συνάδει με τη βιβλιογραφία για ανωμαλίες αγοράς (market anomalies).

Συμπερασματικά, η παρούσα εργασία συνεισφέρει στο πεδίο μέσω της δημιουργίας ενός ανοιχτού, επεκτάσιμου και πλήρως τεκμηριωμένου συστήματος που μπορεί να αποτελέσει βάση για περαιτέρω ακαδημαϊκή έρευνα. Η ενσωμάτωση βαθύτερων αρχιτεκτονικών νευρωνικών δικτύων, ανάλυσης κειμένου και βελτιστοποίησης χαρτοφυλακίου αποτελούν φυσικά βήματα επέκτασης που αναμένεται να ενισχύσουν σημαντικά την προβλεπτική ισχύ και την πρακτική χρησιμότητα του συστήματος.

Βιβλιογραφία

- [1] Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European journal of operational research*, 270(2), 654-669.
- [2] Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review 2005–2019. *Applied Soft Computing*, 90, 106181. <https://doi.org/10.1016/j.asoc.2020.106181>
- [3] Nabipour, M., Nayyeri, P., Jabani, H., Mosavi, A., & Salwana, E. (2020). Deep learning for stock market prediction. *Entropy*, 22(8), 840. <https://doi.org/10.3390/e22080840>
- [4] Zhong, X., & Enke, D. (2019). Predicting the daily return direction of the stock market using hybrid machine learning algorithms. *Financial Innovation*, 5(1), 24. <https://doi.org/10.1186/s40854-019-0138-7>
- [5] Murphy, J. J. (2009). *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications* (2nd ed.). New York Institute of Finance.
- [6] Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *Journal of Finance*, 25(2), 383–417. <https://doi.org/10.2307/2325486>
- [7] Shiller, R. J. (2015). *Irrational exuberance* (3rd ed.). Princeton University Press.
- [8] Appel, G. (2005). *Technical analysis: Power tools for active investors*. FT Press.
- [9] Wilder, J. W. (1978). *New concepts in technical trading systems*. Trend Research.
- [10] Agrawal, M., Khan, A. U., & Shukla, P. K. (2019). Stock price prediction using technical indicators: A predictive model using optimal deep learning. *International Journal of Recent Technology and Engineering*, 8(2), 2297–2305.
- [11] Bollinger, J. (2002). *Bollinger on Bollinger Bands*. McGraw-Hill.
- [12] Sharpe, W. F. (1994). The Sharpe ratio. *Journal of Portfolio Management*, 21(1), 49–58. <https://doi.org/10.3905/jpm.1994.409501>
- [13] Magdon-Ismail, M., & Atiya, A. (2004). Maximum drawdown. *Risk Magazine*, 17(10), 99–102.
- [14] Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill.
- [15] Chollet, F. (2021). *Deep learning with Python* (2nd ed.). Manning Publications.
- [16] Tsay, R. S. (2010). *Analysis of financial time series* (3rd ed.). John Wiley & Sons.
- [17] Bergmeir, C., & Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191, 192–213. <https://doi.org/10.1016/j.ins.2011.12.028>

- [18] Zheng, A., & Casari, A. (2018). Feature engineering for machine learning: Principles and techniques for data scientists. O'Reilly Media.
- [19] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.
- [20] Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4), 679–688. <https://doi.org/10.1016/j.ijforecast.2006.03.001>
- [21] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- [22] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189–1232. <https://doi.org/10.1214/aos/1013203451>
- [23] Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1), 55–67. <https://doi.org/10.1080/00401706.1970.10488634>
- [24] Jiang, W. (2021). Applications of deep learning in stock market prediction: Recent progress. *Expert Systems with Applications*, 184, 115537. <https://doi.org/10.1016/j.eswa.2021.115537>
- [25] Zhong, X., & Enke, D. (2017). Forecasting daily stock market return using dimensionality reduction. *Expert Systems with Applications*, 67, 126–139. <https://doi.org/10.1016/j.eswa.2016.09.027>
- [26] Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2020). Stock closing price prediction using machine learning techniques. *Procedia Computer Science*, 167, 599–606. <https://doi.org/10.1016/j.procs.2020.03.326>
- [27] Ampomah, E. K., Qin, Z., & Nyame, G. (2020). Evaluation of tree-based ensemble machine learning models in predicting stock price direction of movement. *Information*, 11(6), 332. <https://doi.org/10.3390/info11060332>
- [28] Mehtab, S., Sen, J., & Dutta, A. (2021). Stock price prediction using machine learning and LSTM-based deep learning models. In S. M. Thampi et al. (Eds.), *Machine Learning and Metaheuristics Algorithms, and Applications* (pp. 88–106). Springer. https://doi.org/10.1007/978-981-16-0419-5_8

Παράρτημα Α: «Κώδικας υλοποίησης σε περιβάλλον VSC»

```
"""
Advanced Stock Price Forecasting – ML Web App
Streamlit + scikit-learn + Plotly + yfinance
"""

import warnings
import numpy as np
import pandas as pd
import streamlit as st
import yfinance as yf
import plotly.graph_objects as go
from plotly.subplots import make_subplots
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.linear_model import Ridge
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer
import joblib

warnings.filterwarnings("ignore")

# — Page Config —————
st.set_page_config(
    page_title="Advanced Stock Forecast",
    page_icon="📈",
    layout="wide",
    initial_sidebar_state="expanded",
)

st.markdown("""
<style>
[data-testid="stMetricValue"] { font-size: 1.1rem; }
.stTabs [data-baseweb="tab"] { padding: 8px 20px; border-radius: 8px 8px 0 0; }
section[data-testid="stSidebar"] { background: #0e1117; }
section[data-testid="stSidebar"] .stMarkdown,
section[data-testid="stSidebar"] label { color: #fafafa !important; }
</style>
""", unsafe_allow_html=True)
```

```
# =====
# DATA & FEATURE ENGINEERING
# =====

@st.cache_data(ttl=300)
def fetch_data(ticker: str, start: str, end: str) -> pd.DataFrame:
    df = yf.download(ticker, start=start, end=end, progress=False, auto_adjust=True)
    if df.empty:
        return pd.DataFrame()
    # Flatten multi-level columns if present
    df.columns = [c[0] if isinstance(c, tuple) else c for c in df.columns]
    df.columns = [str(c).title() for c in df.columns]
    keep = [c for c in ["Open", "High", "Low", "Close", "Volume"] if c in df.columns]
    return df[keep].dropna()

@st.cache_data(ttl=300)
def fetch_info(ticker: str) -> dict:
    try:
        return yf.Ticker(ticker).info
    except Exception:
        return {}

def add_features(df: pd.DataFrame) -> pd.DataFrame:
    out = df.copy()
    c = out["Close"]

    # — Returns —————
    for lag in [1, 3, 5, 10]:
        out[f"Return_{lag}d"] = c.pct_change(lag)

    # — Trend / Moving Averages —————
    for w in [10, 20, 50]:
        out[f"SMA_{w}"] = c.rolling(w).mean()
    out["EMA_12"] = c.ewm(span=12, adjust=False).mean()
    out["EMA_26"] = c.ewm(span=26, adjust=False).mean()
    out["Price_SMA20_Ratio"] = c / (out["SMA_20"] + 1e-9)

    # — MACD —————
    out["MACD"] = out["EMA_12"] - out["EMA_26"]
    out["MACD_Signal"] = out["MACD"].ewm(span=9, adjust=False).mean()
    out["MACD_Hist"] = out["MACD"] - out["MACD_Signal"]
```

```
# — RSI —————
delta = c.diff()
gain = delta.where(delta > 0, 0.0).rolling(14).mean()
loss = (-delta.where(delta < 0, 0.0)).rolling(14).mean()
out["RSI_14"] = 100 - (100 / (1 + gain / (loss + 1e-12)))

# — Bollinger Bands —————
bb_mid = c.rolling(20).mean()
bb_std = c.rolling(20).std()
out["BB_Upper"] = bb_mid + 2 * bb_std
out["BB_Lower"] = bb_mid - 2 * bb_std
out["BB_Width"] = (out["BB_Upper"] - out["BB_Lower"]) / (bb_mid + 1e-9)
out["BB_Position"] = (c - out["BB_Lower"]) / (out["BB_Upper"] - out["BB_Lower"] + 1e-
9)

# — ATR (Average True Range) —————
hl = out["High"] - out["Low"]
hc = (out["High"] - c.shift(1)).abs()
lc = (out["Low"] - c.shift(1)).abs()
tr = pd.concat([hl, hc, lc], axis=1).max(axis=1)
out["ATR_14"] = tr.rolling(14).mean()
out["ATR_Pct"] = out["ATR_14"] / (c + 1e-9) * 100

# — Stochastic Oscillator —————
low14 = out["Low"].rolling(14).min()
high14 = out["High"].rolling(14).max()
out["Stoch_K"] = 100 * (c - low14) / (high14 - low14 + 1e-9)
out["Stoch_D"] = out["Stoch_K"].rolling(3).mean()

# — OBV (On-Balance Volume) —————
obv = [0]
closes = out["Close"].values
volumes = out["Volume"].values
for i in range(1, len(out)):
    obv.append(obv[-1] + (volumes[i] if closes[i] > closes[i - 1] else
                        -volumes[i] if closes[i] < closes[i - 1] else 0))
out["OBV"] = obv
out["OBV_EMA"] = pd.Series(obv, index=out.index).ewm(span=20, adjust=False).mean()

# — Volume —————
out["Volume_Ratio"] = out["Volume"] / (out["Volume"].rolling(20).mean() + 1e-9)

# — Volatility —————
out["Volatility_20d"] = out["Return_1d"].rolling(20).std() * np.sqrt(252)

return out
```

```
def make_supervised(df: pd.DataFrame, horizon: int = 1):
    """Create target (close price N days ahead) and drop lookahead columns."""
    data = df.copy()
    data["y"] = data["Close"].shift(-horizon)
    data = data.dropna()
    y = data.pop("y")
    # Remove columns that would leak the future
    drop_for_ml = ["BB_Upper", "BB_Lower", "SMA_10", "SMA_20", "SMA_50",
                   "EMA_12", "EMA_26"]
    X = data.drop(columns=[c for c in drop_for_ml if c in data.columns])
    return X, y

def time_split(X, y, test_ratio: float = 0.20):
    split = len(X) - int(len(X) * test_ratio)
    return X.iloc[:split], X.iloc[split:], y.iloc[:split], y.iloc[split:]

def build_pipeline(model) -> Pipeline:
    return Pipeline([
        ("imputer", SimpleImputer(strategy="median")),
        ("scaler", StandardScaler()),
        ("model", model),
    ])

def compute_risk(returns: pd.Series) -> dict:
    ann = 252
    rf_daily = 0.02 / ann
    sharpe = ((returns.mean() - rf_daily) / (returns.std() + 1e-12)) * np.sqrt(ann)
    down_std = returns[returns < 0].std() * np.sqrt(ann)
    sortino = (returns.mean() * ann - 0.02) / (down_std + 1e-12)
    cumret = (1 + returns).cumprod()
    drawdown = (cumret - cumret.cummax()) / (cumret.cummax() + 1e-12)
    max_dd = drawdown.min()
    var95 = float(np.percentile(returns.dropna(), 5))
    cvar95 = float(returns[returns <= var95].mean())
    return {
        "Sharpe Ratio": f"{sharpe:.3f}",
        "Sortino Ratio": f"{sortino:.3f}",
        "Max Drawdown": f"{max_dd:.2%}",
        "VaR 95%": f"{var95:.2%}",
        "CVaR 95%": f"{cvar95:.2%}",
        "Ann. Volatility": f"{returns.std() * np.sqrt(ann):.2%}",
        "Ann. Return": f"{returns.mean() * ann:.2%}",
    }
```

```
# =====
# MODEL REGISTRY
# =====

MODELS = {
    "Random Forest": RandomForestRegressor(n_estimators=400, random_state=42, n_jobs=-1),
    "Gradient Boosting": GradientBoostingRegressor(n_estimators=250, learning_rate=0.05,
random_state=42),
    "Ridge Regression": Ridge(alpha=1.0),
}

MODEL_COLORS = {
    "Random Forest": "#ff7f0e",
    "Gradient Boosting": "#2ca02c",
    "Ridge Regression": "#9467bd",
}

# =====
# SIDEBAR
# =====

with st.sidebar:
    st.markdown("## ⚙️ Settings")
    ticker = st.text_input("Ticker Symbol", "AAPL").upper().strip()

    col_a, col_b = st.columns(2)
    with col_a:
        start = st.date_input("Start", pd.to_datetime("2019-01-01"))
    with col_b:
        end = st.date_input("End", pd.to_datetime("today"))

    horizon = st.slider("Forecast horizon (days)", 1, 30, 5)
    primary_model = st.selectbox("Primary model", list(MODELS.keys()))

    st.divider()
    st.markdown("***Quick picks**")
    c1, c2, c3 = st.columns(3)
    if c1.button("AAPL"):
        ticker = "AAPL"
    if c2.button("TSLA"):
        ticker = "TSLA"
    if c3.button("MSFT"):
        ticker = "MSFT"

    st.divider()
    st.caption("Data: Yahoo Finance · Models: scikit-learn")
    st.caption("Charts: Plotly · App: Streamlit")
```

```
# =====  
# LOAD DATA  
# =====  
  
df_raw = fetch_data(ticker, str(start), str(end))  
  
if df_raw.empty:  
    st.error(f"❌ No data found for **{ticker}**. Check the symbol and date range.")  
    st.stop()  
  
if len(df_raw) < 100:  
    st.warning("⚠️ Less than 100 trading days – predictions may be unreliable.")  
  
info = fetch_info(ticker)  
df_feat = add_features(df_raw)  
returns = df_raw["Close"].pct_change().dropna()  
  
# =====  
# HEADER – KEY METRICS  
# =====  
  
current = float(df_raw["Close"].iloc[-1])  
prev = float(df_raw["Close"].iloc[-2])  
chg = current - prev  
chg_pct = chg / prev * 100  
company = info.get("longName", ticker)  
  
st.title(f"📈 {company}")  
st.markdown(f"**{ticker}** &nbsp;&nbsp;&nbsp; {info.get('sector', '')} &nbsp;&nbsp;&nbsp; |&nbsp;&nbsp;&nbsp; {info.get('exchange', '')}")  
  
col1, col2, col3, col4, col5, col6 = st.columns(6)  
col1.metric("Price", f"${current:.2f}", f"{chg_pct:+.2f}%")  
col2.metric("Volume", f"{int(df_raw['Volume'].iloc[-1]):,}")  
col3.metric("52W High", f"${df_raw['Close'].tail(252).max():.2f}")  
col4.metric("52W Low", f"${df_raw['Close'].tail(252).min():.2f}")  
col5.metric("Avg Vol 20d", f"{int(df_raw['Volume'].tail(20).mean()):,}")  
col6.metric("Beta", f"{info.get('beta', 'N/A')} if info.get('beta') else 'N/A')  
  
st.divider()
```

```
# =====
# TABS
# =====

tab1, tab2, tab3, tab4, tab5 = st.tabs([
    "📊 Price & Overview",
    "📈 Technical Analysis",
    "🤖 ML Prediction",
    "⚖️ Model Comparison",
    "🏠 Risk & Fundamentals",
])

# =====
# TAB 1 — PRICE & OVERVIEW
# =====

with tab1:
    col_left, col_right = st.columns([3, 1])

    with col_left:
        ma_sel = st.multiselect(
            "Overlay moving averages",
            ["SMA_10", "SMA_20", "SMA_50", "EMA_12", "EMA_26"],
            default=["SMA_20", "SMA_50"],
        )
        show_bb = st.checkbox("Bollinger Bands", value=True)

    with col_right:
        chart_type = st.radio("Chart type", ["Candlestick", "Line"], horizontal=True)

    fig_price = make_subplots(
        rows=2, cols=1, shared_xaxes=True,
        row_heights=[0.78, 0.22], vertical_spacing=0.02,
    )

    if chart_type == "Candlestick":
        fig_price.add_trace(go.Candlestick(
            x=df_raw.index,
            open=df_raw["Open"], high=df_raw["High"],
            low=df_raw["Low"], close=df_raw["Close"],
            name="Price",
            increasing_line_color="#26a69a", decreasing_line_color="#ef5350",
        ), row=1, col=1)
    else:
        fig_price.add_trace(go.Scatter(
            x=df_raw.index, y=df_raw["Close"],
            name="Close", line=dict(color="#1f77b4", width=1.5),
        ), row=1, col=1)
```

```
ma_colors = {"SMA_10": "#9467bd", "SMA_20": "#ff7f0e", "SMA_50": "#1f77b4",
             "EMA_12": "#8c564b", "EMA_26": "#e377c2"}

for ma in ma_sel:
    if ma in df_feat.columns:
        fig_price.add_trace(go.Scatter(
            x=df_feat.index, y=df_feat[ma], name=ma,
            line=dict(width=1.5, color=ma_colors.get(ma)),
        ), row=1, col=1)

if show_bb:
    fig_price.add_trace(go.Scatter(
        x=df_feat.index, y=df_feat["BB_Upper"], name="BB Upper",
        line=dict(width=1, color="rgba(180,180,180,0.9)", dash="dot"),
    ), row=1, col=1)
    fig_price.add_trace(go.Scatter(
        x=df_feat.index, y=df_feat["BB_Lower"], name="BB Lower",
        line=dict(width=1, color="rgba(180,180,180,0.9)", dash="dot"),
        fill="tonexty", fillcolor="rgba(150,150,150,0.06)",
    ), row=1, col=1)

vol_colors = [
    "#26a69a" if df_raw["Close"].iloc[i] >= df_raw["Open"].iloc[i] else "#ef5350"
    for i in range(len(df_raw))
]

fig_price.add_trace(go.Bar(
    x=df_raw.index, y=df_raw["Volume"],
    name="Volume", marker_color=vol_colors, opacity=0.55,
), row=2, col=1)

fig_price.update_layout(
    height=600, xaxis_rangeslider_visible=False,
    legend=dict(orientation="h", y=1.02, x=0),
    margin=dict(l=0, r=0, t=10, b=0),
    plot_bgcolor="rgba(0,0,0,0)",
    paper_bgcolor="rgba(0,0,0,0)",
)

fig_price.update_xaxes(showgrid=True, gridwidth=0.5,
gridcolor="rgba(128,128,128,0.15)")
fig_price.update_yaxes(showgrid=True, gridwidth=0.5,
gridcolor="rgba(128,128,128,0.15)")

st.plotly_chart(fig_price, use_container_width=True)
```

```
# Returns analysis
st.subheader("📊 Returns Analysis")
c1, c2 = st.columns(2)
with c1:
    fig_hist = go.Figure()
    fig_hist.add_trace(go.Histogram(
        x=returns * 100, nbinsx=80, name="Returns %",
        marker_color="#1f77b4", opacity=0.75,
    ))
    mean_r = returns.mean() * 100
    fig_hist.add_vline(x=mean_r, line_dash="dash", line_color="#ff7f0e",
        annotation_text=f"Mean: {mean_r:.3f}%")
    fig_hist.update_layout(
        height=300, title="Daily Returns Distribution (%)",
        margin=dict(l=0, r=0, t=40, b=0),
        plot_bgcolor="rgba(0,0,0,0)", paper_bgcolor="rgba(0,0,0,0)",
    )
    st.plotly_chart(fig_hist, use_container_width=True)

with c2:
    cumret = (1 + returns).cumprod()
    color_cum = "#26a69a" if cumret.iloc[-1] >= 1 else "#ef5350"
    fig_cum = go.Figure()
    fig_cum.add_trace(go.Scatter(
        x=cumret.index, y=(cumret - 1) * 100,
        fill="tozeroy", name="Cumulative Return %",
        line=dict(color=color_cum, width=1.5),
        fillcolor=f"rgba({'38,166,154' if color_cum == '#26a69a' else
'239,83,80'},0.1)",
    ))
    total_ret = (cumret.iloc[-1] - 1) * 100
    fig_cum.update_layout(
        height=300, title=f"Cumulative Return: {total_ret:+.1f}%",
        margin=dict(l=0, r=0, t=40, b=0), yaxis_ticksuffix="%",
        plot_bgcolor="rgba(0,0,0,0)", paper_bgcolor="rgba(0,0,0,0)",
    )
    st.plotly_chart(fig_cum, use_container_width=True)
```

```
# -----
# TAB 2 – TECHNICAL ANALYSIS
# -----
with tab2:
    fig_tech = make_subplots(
        rows=4, cols=1, shared_xaxes=True,
        row_heights=[0.40, 0.20, 0.20, 0.20],
        vertical_spacing=0.025,
        subplot_titles=("Price + Bollinger Bands", "MACD", "RSI (14)", "Stochastic"),
    )

    # – Price + BB
    fig_tech.add_trace(go.Scatter(
        x=df_raw.index, y=df_raw["Close"], name="Close",
        line=dict(color="#1f77b4", width=1.5),
    ), row=1, col=1)
    fig_tech.add_trace(go.Scatter(
        x=df_feat.index, y=df_feat["BB_Upper"], name="BB Upper",
        line=dict(color="gray", width=1, dash="dot"),
    ), row=1, col=1)
    fig_tech.add_trace(go.Scatter(
        x=df_feat.index, y=df_feat["BB_Lower"], name="BB Lower",
        line=dict(color="gray", width=1, dash="dot"),
        fill="tonexty", fillcolor="rgba(128,128,128,0.07)",
    ), row=1, col=1)
    fig_tech.add_trace(go.Scatter(
        x=df_feat.index, y=df_feat["SMA_20"], name="SMA 20",
        line=dict(color="#ff7f0e", width=1),
    ), row=1, col=1)

    # – MACD
    fig_tech.add_trace(go.Scatter(
        x=df_feat.index, y=df_feat["MACD"], name="MACD",
        line=dict(color="#ff7f0e", width=1.5),
    ), row=2, col=1)
    fig_tech.add_trace(go.Scatter(
        x=df_feat.index, y=df_feat["MACD_Signal"], name="Signal",
        line=dict(color="#d62728", width=1.5),
    ), row=2, col=1)
    hist_vals = df_feat["MACD_Hist"]
    fig_tech.add_trace(go.Bar(
        x=df_feat.index, y=hist_vals, name="Histogram",
        marker_color=["#26a69a" if v >= 0 else "#ef5350" for v in hist_vals],
        opacity=0.6,
    ), row=2, col=1)

    # – RSI
```

```
fig_tech.add_trace(go.Scatter(
    x=df_feat.index, y=df_feat["RSI_14"], name="RSI 14",
    line=dict(color="#9467bd", width=1.5),
), row=3, col=1)
for lvl, clr in [(70, "rgba(214,39,40,0.6)"), (30, "rgba(38,166,154,0.6)"]]:
    fig_tech.add_hline(y=lvl, line_dash="dot", line_color=clr, row=3, col=1)
fig_tech.add_hrect(y0=70, y1=100, fillcolor="rgba(214,39,40,0.05)", row=3, col=1)
fig_tech.add_hrect(y0=0, y1=30, fillcolor="rgba(38,166,154,0.05)", row=3, col=1)

# - Stochastic
fig_tech.add_trace(go.Scatter(
    x=df_feat.index, y=df_feat["Stoch_K"], name="%K",
    line=dict(color="#17becf", width=1.5),
), row=4, col=1)
fig_tech.add_trace(go.Scatter(
    x=df_feat.index, y=df_feat["Stoch_D"], name="%D",
    line=dict(color="#bcbd22", width=1.5),
), row=4, col=1)
for lvl, clr in [(80, "rgba(214,39,40,0.6)"), (20, "rgba(38,166,154,0.6)"]]:
    fig_tech.add_hline(y=lvl, line_dash="dot", line_color=clr, row=4, col=1)

fig_tech.update_layout(
    height=900, showlegend=True,
    margin=dict(l=0, r=0, t=30, b=0),
    plot_bgcolor="rgba(0,0,0,0)", paper_bgcolor="rgba(0,0,0,0)",
)
fig_tech.update_xaxes(showgrid=True, gridwidth=0.5,
gridcolor="rgba(128,128,128,0.15)")
fig_tech.update_yaxes(showgrid=True, gridwidth=0.5,
gridcolor="rgba(128,128,128,0.15)")
st.plotly_chart(fig_tech, use_container_width=True)

# - Signal summary
st.subheader("📄 Signal Summary (latest bar)")
latest = df_feat.iloc[-1]
rsi_v = latest["RSI_14"]
macd_v = latest["MACD"]
macd_s = latest["MACD_Signal"]
stoch_v = latest["Stoch_K"]
bb_pos = latest["BB_Position"]
price_v = latest["Close"]
```

```
def badge(label: str, bull: bool, bear: bool) -> str:
    if bull:
        return f"📈 **{label}**: Bullish"
    if bear:
        return f"📉 **{label}**: Bearish"
    return f"📊 **{label}**: Neutral"

signals = [
    badge("RSI", rsi_v < 30, rsi_v > 70),
    badge("MACD", macd_v > macd_s, macd_v < macd_s),
    badge("Bollinger", bb_pos < 0.1, bb_pos > 0.9),
    badge("Stochastic", stoch_v < 20, stoch_v > 80),
]

cols = st.columns(4)
details = [
    f"RSI = **{rsi_v:.1f}**",
    f"MACD = **{macd_v:.4f}** / Signal = **{macd_s:.4f}**",
    f"BB pos = **{bb_pos:.2%}**",
    f"%K = **{stoch_v:.1f}**",
]

for col, sig, det in zip(cols, signals, details):
    with col:
        st.info(sig)
        st.caption(det)

# -----
# TAB 3 – ML PREDICTION
# -----

with tab3:
    X_all, y_all = make_supervised(df_feat, horizon=horizon)
    X_tr, X_te, y_tr, y_te = time_split(X_all, y_all)

    with st.spinner(f"Training {primary_model}..."):
        pipe = build_pipeline(MODELS[primary_model])
        pipe.fit(X_tr, y_tr)

    pred = pipe.predict(X_te)
    mae = mean_absolute_error(y_te, pred)
    rmse = float(np.sqrt(mean_squared_error(y_te, pred)))
    r2 = r2_score(y_te, pred)
    mape = float(np.mean(np.abs((y_te.values - pred) / (y_te.values + 1e-9))) * 100)
    direc_acc = float(np.mean(np.sign(np.diff(pred)) == np.sign(np.diff(y_te.values))) *
100)
```

```

st.subheader(f"📊 {primary_model} - Evaluation Metrics")
c1, c2, c3, c4, c5 = st.columns(5)
c1.metric("MAE", f"${mae:.2f}")
c2.metric("RMSE", f"${rmse:.2f}")
c3.metric("R²", f"${r2:.4f}")
c4.metric("MAPE", f"${mape:.2f}%")
c5.metric("Directional Acc.", f"${direc_acc:.1f}%")

# Prediction vs Actual chart
fig_pred = go.Figure()
fig_pred.add_trace(go.Scatter(
    x=y_te.index, y=y_te.values, name="Actual",
    line=dict(color="#1f77b4", width=2),
))
fig_pred.add_trace(go.Scatter(
    x=y_te.index, y=pred, name="Predicted",
    line=dict(color="#ff7f0e", width=2, dash="dash"),
))
fig_pred.add_trace(go.Scatter(
    x=y_te.index, y=pred + rmse, showlegend=False,
    line=dict(width=0),
))
fig_pred.add_trace(go.Scatter(
    x=y_te.index, y=pred - rmse, name=f"±1 RMSE Band",
    line=dict(width=0), fill="tonexty",
    fillcolor="rgba(255,127,14,0.12)",
))
fig_pred.update_layout(
    height=420,
    title=f"{primary_model} · Test Set · Horizon = {horizon}d",
    margin=dict(l=0, r=0, t=50, b=0),
    plot_bgcolor="rgba(0,0,0,0)", paper_bgcolor="rgba(0,0,0,0)",
)
fig_pred.update_xaxes(showgrid=True, gridwidth=0.5,
gridcolor="rgba(128,128,128,0.15)")
fig_pred.update_yaxes(showgrid=True, gridwidth=0.5,
gridcolor="rgba(128,128,128,0.15)")
st.plotly_chart(fig_pred, use_container_width=True)

# Residuals
with st.expander("📊 Residuals Analysis"):
    residuals = y_te.values - pred
    c_r1, c_r2 = st.columns(2)
    with c_r1:
        fig_res = go.Figure()
        fig_res.add_trace(go.Scatter(
            x=y_te.index, y=residuals, mode="markers+lines",

```

```

        marker=dict(size=3), line=dict(width=0.8, color="#9467bd"),
        name="Residuals",
    ))
    fig_res.add_hline(y=0, line_dash="dash", line_color="gray")
    fig_res.update_layout(height=250, title="Residuals over time",
                           margin=dict(l=0, r=0, t=40, b=0))
    st.plotly_chart(fig_res, use_container_width=True)
with c_r2:
    fig_rh = go.Figure()
    fig_rh.add_trace(go.Histogram(x=residuals, nbinsx=40,
                                  marker_color="#9467bd", opacity=0.7))
    fig_rh.update_layout(height=250, title="Residuals distribution",
                           margin=dict(l=0, r=0, t=40, b=0))
    st.plotly_chart(fig_rh, use_container_width=True)

# - Future forecast
st.subheader(f"🔮 Future Forecast (+{horizon} trading days)")
last_X = X_all.iloc[[-1]]
future_price = float(pipe.predict(last_X)[0])
fut_chg = (future_price - current) / current * 100
fut_lo = future_price - rmse
fut_hi = future_price + rmse

fc1, fc2, fc3 = st.columns(3)
fc1.metric("Predicted Price", f"${future_price:.2f}", f"${fut_chg:+.2f}%")
fc2.metric("Confidence Low (-1 RMSE)", f"${fut_lo:.2f}")
fc3.metric("Confidence High (+1 RMSE)", f"${fut_hi:.2f}")

# - Feature Importance
mdl_obj = pipe.named_steps["model"]
if hasattr(mdl_obj, "feature_importances_"):
    st.subheader("📊 Feature Importance")
    fi = pd.DataFrame({
        "Feature": X_all.columns,
        "Importance": mdl_obj.feature_importances_,
    }).sort_values("Importance").tail(20)

    fig_fi = go.Figure(go.Bar(
        x=fi["Importance"], y=fi["Feature"], orientation="h",
        marker_color="#1f77b4",
    ))
    fig_fi.update_layout(
        height=500, margin=dict(l=0, r=0, t=20, b=0),
        plot_bgcolor="rgba(0,0,0,0)", paper_bgcolor="rgba(0,0,0,0)",
    )
    st.plotly_chart(fig_fi, use_container_width=True)

```

```
# - Download
import io
model_buffer = io.BytesIO()
joblib.dump(pipe, model_buffer)
model_buffer.seek(0)
st.download_button(
    "⬇️ Download trained model (.joblib)",
    data=model_buffer,
    file_name=f"{ticker}_{primary_model.replace(' ', '_')}_horizon{horizon}.joblib",
    mime="application/octet-stream",
)

# -----
# TAB 4 - MODEL COMPARISON
# -----

with tab4:
    st.subheader("📊 All Models - Side-by-Side")

    X_c, y_c = make_supervised(df_feat, horizon=horizon)
    X_tr_c, X_te_c, y_tr_c, y_te_c = time_split(X_c, y_c)

    results: dict = {}
    all_preds: dict = {}
    prog = st.progress(0, text="Training all models...")

    for i, (name, mdl) in enumerate(MODELS.items()):
        p = build_pipeline(mdl)
        p.fit(X_tr_c, y_tr_c)
        pr = p.predict(X_te_c)
        all_preds[name] = pr
        results[name] = {
            "MAE ($)": round(mean_absolute_error(y_te_c, pr), 3),
            "RMSE ($)": round(float(np.sqrt(mean_squared_error(y_te_c, pr))), 3),
            "R²": round(r2_score(y_te_c, pr), 4),
            "MAPE (%)": round(float(np.mean(np.abs((y_te_c.values - pr) /
                                                    (y_te_c.values + 1e-9)))) * 100, 2),
            "Dir. Acc. (%)": round(float(np.mean(
                np.sign(np.diff(pr)) == np.sign(np.diff(y_te_c.values)))) * 100, 1),
        }
        prog.progress((i + 1) / len(MODELS), text=f"Trained {name}")

    prog.empty()

    df_res = pd.DataFrame(results).T.reset_index().rename(columns={"index": "Model"})
```

```
st.dataframe(
    df_res.style
        .highlight_max(subset=["R2", "Dir. Acc. (%)"], color="#c8e6c9")
        .highlight_min(subset=["MAE ($)", "RMSE ($)", "MAPE (%)"], color="#c8e6c9")
        .format(precision=3),
    use_container_width=True,
    hide_index=True,
)

# Predictions comparison
fig_cmp = go.Figure()
fig_cmp.add_trace(go.Scatter(
    x=y_te_c.index, y=y_te_c.values, name="Actual",
    line=dict(color="#1f77b4", width=2.5),
))
for name, pr in all_preds.items():
    fig_cmp.add_trace(go.Scatter(
        x=y_te_c.index, y=pr, name=name,
        line=dict(color=MODEL_COLORS[name], width=1.5, dash="dash"),
    ))
fig_cmp.update_layout(
    height=420, title="All Models – Test Set Predictions",
    margin=dict(l=0, r=0, t=50, b=0),
    plot_bgcolor="rgba(0,0,0,0)", paper_bgcolor="rgba(0,0,0,0)",
)
fig_cmp.update_xaxes(showgrid=True, gridwidth=0.5, gridcolor="rgba(128,128,128,0.15)")
st.plotly_chart(fig_cmp, use_container_width=True)

# Metric bar charts
metric_sel = st.selectbox("Compare metric", ["MAE ($)", "RMSE ($)", "R2", "MAPE (%)",
"Dir. Acc. (%)"])
names = list(results.keys())
vals = [results[n][metric_sel] for n in names]
colors_bar = [MODEL_COLORS[n] for n in names]

fig_bar = go.Figure(go.Bar(
    x=names, y=vals, marker_color=colors_bar,
    text=[f"{v:.3f}" for v in vals], textposition="auto",
))
fig_bar.update_layout(
    height=320, title=f"{metric_sel} – model comparison",
    margin=dict(l=0, r=0, t=50, b=0),
    plot_bgcolor="rgba(0,0,0,0)", paper_bgcolor="rgba(0,0,0,0)",
)
st.plotly_chart(fig_bar, use_container_width=True)
```

```
# Scatter: Actual vs Predicted
st.subheader("☞ Actual vs Predicted Scatter")
scatter_model = st.selectbox("Model for scatter", list(MODELS.keys()),
key="scatter_sel")
pr_sc = all_preds[scatter_model]
fig_sc = go.Figure()
fig_sc.add_trace(go.Scatter(
    x=y_te_c.values, y=pr_sc, mode="markers",
    marker=dict(color=MODEL_COLORS[scatter_model], size=5, opacity=0.7),
    name=scatter_model,
))
mn = min(y_te_c.min(), pr_sc.min())
mx = max(y_te_c.max(), pr_sc.max())
fig_sc.add_trace(go.Scatter(x=[mn, mx], y=[mn, mx], mode="lines",
                           name="Perfect fit", line=dict(color="gray", dash="dot")))
fig_sc.update_layout(
    height=380, title="Actual vs Predicted",
    xaxis_title="Actual", yaxis_title="Predicted",
    margin=dict(l=0, r=0, t=50, b=0),
    plot_bgcolor="rgba(0,0,0,0)", paper_bgcolor="rgba(0,0,0,0)",
)
st.plotly_chart(fig_sc, use_container_width=True)

# -----
# TAB 5 – RISK & FUNDAMENTALS
# -----
with tab5:
    # — Risk Metrics -----
    st.subheader("⚠ Risk Metrics")
    risk = compute_risk(returns)
    risk_cols = st.columns(len(risk))
    for col, (k, v) in zip(risk_cols, risk.items()):
        col.metric(k, v)

    # Drawdown chart
    cumret = (1 + returns).cumprod()
    drawdown = (cumret - cumret.cummax()) / (cumret.cummax() + 1e-12) * 100

    fig_dd = go.Figure()
    fig_dd.add_trace(go.Scatter(
        x=drawdown.index, y=drawdown, fill="tozeroy",
        name="Drawdown %",
        line=dict(color="#ef5350", width=1),
        fillcolor="rgba(239,83,80,0.12)",
    ))
    fig_dd.update_layout(
```

```
height=280, title="Drawdown History (%)",
yaxis_ticksuffix="%",
margin=dict(l=0, r=0, t=50, b=0),
plot_bgcolor="rgba(0,0,0,0)", paper_bgcolor="rgba(0,0,0,0)",
)
st.plotly_chart(fig_dd, use_container_width=True)

# Rolling Sharpe Ratio
roll_win = 60
roll_sharpe = (returns.rolling(roll_win).mean() /
               (returns.rolling(roll_win).std() + 1e-12)) * np.sqrt(252)
fig_rs = go.Figure()
fig_rs.add_trace(go.Scatter(
    x=roll_sharpe.index, y=roll_sharpe, name=f"Rolling Sharpe ({roll_win}d)",
    line=dict(color="#9467bd", width=1.5),
))
fig_rs.add_hline(y=1, line_dash="dot", line_color="green",
                 annotation_text="Sharpe = 1")
fig_rs.add_hline(y=0, line_dash="dot", line_color="red")
fig_rs.update_layout(
    height=260, title=f"Rolling Sharpe Ratio ({roll_win}d)",
    margin=dict(l=0, r=0, t=50, b=0),
    plot_bgcolor="rgba(0,0,0,0)", paper_bgcolor="rgba(0,0,0,0)",
)
st.plotly_chart(fig_rs, use_container_width=True)

st.divider()

# — Fundamentals —————
st.subheader("🏢 Company Fundamentals")
fund_map = {
    "Market Cap": "marketCap",
    "P/E (TTM)": "trailingPE",
    "Forward P/E": "forwardPE",
    "P/B Ratio": "priceToBook",
    "EPS (TTM)": "trailingEps",
    "Dividend Yield": "dividendYield",
    "Beta": "beta",
    "Revenue (TTM)": "totalRevenue",
    "Net Income": "netIncomeToCommon",
    "Gross Margins": "grossMargins",
    "Operating Margins": "operatingMargins",
    "Debt/Equity": "debtToEquity",
}
```

```
fund_data = {}
for label, key in fund_map.items():
    val = info.get(key)
    if val is None:
        continue
    if key in ("marketCap", "totalRevenue", "netIncomeToCommon"):
        if abs(val) >= 1e12:
            val = f"${val/1e12:.2f}T"
        elif abs(val) >= 1e9:
            val = f"${val/1e9:.2f}B"
        else:
            val = f"${val/1e6:.0f}M"
    elif key in ("dividendYield", "grossMargins", "operatingMargins"):
        val = f"{val:.2%}"
    else:
        val = f"{val:.2f}"
    fund_data[label] = val

if fund_data:
    items = list(fund_data.items())
    per_row = 4
    for row_start in range(0, len(items), per_row):
        cols = st.columns(per_row)
        for col, (k, v) in zip(cols, items[row_start:row_start + per_row]):
            col.metric(k, v)
else:
    st.info("Fundamentals not available for this ticker.")

# Company description
desc = info.get("longBusinessSummary", "")
if desc:
    with st.expander("📖 Company Description"):
        st.write(desc)
```

```
# Officers
officers = info.get("companyOfficers", [])
if officers:
    with st.expander("📄 Key Officers"):
        off_df = pd.DataFrame([
            {
                "Name": o.get("name", ""),
                "Title": o.get("title", ""),
                "Pay": f"${o['totalPay']['raw']:,}" if isinstance(o.get("totalPay"),
dict) and "raw" in o["totalPay"] else "N/A",
            }
            for o in officers[:8]
        ])
    st.dataframe(off_df, use_container_width=True, hide_index=True)
```

Υπεύθυνη Δήλωση Συγγραφέα:

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