

Σχολή Θετικών επιστημών & Τεχνολογίας (ΣΘΕΤ)
Μεταπτυχιακή Εξειδίκευση στα Πληροφοριακά Συστήματα

Διπλωματική Εργασία

**Ανάπτυξη Συστήματος Ανάλυσης και Πρόβλεψης Πωλήσεων με
Χρήση Τεχνικών Μηχανικής Μάθησης**

Βασίλειος Λάρο

AM 166321

Πάτρα, Σεπτέμβριος 2026

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή («συγγραφέας/δημιουργός») 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.

**Ανάπτυξη Συστήματος Ανάλυσης και Πρόβλεψης Πωλήσεων με
Χρήση Τεχνικών Μηχανικής Μάθησης**

Βασίλειος Λάρο

Τριμελής επιτροπή εξέτασης Διπλωματικής Εργασίας:

Ιωάννης Ρεφανίδης - Επιβλέπων Καθηγητής

Νικόλαος Καρούσος

Γεωργία Γκαράνη

Πάτρα, Σεπτέμβριος 2026

*Στην οικογένειά μου για την στήριξή τους
και ειδικά στην αγαπημένη μου Χρυσάννα για την παρακίνηση της.*

Περίληψη

Η παρούσα διπλωματική εργασία παρουσιάζει τη σχεδίαση και υλοποίηση ενός ολοκληρωμένου συστήματος επιχειρησιακής ευφυΐας με δυνατότητες προβλεπτικής ανάλυσης, προσαρμοσμένου στις ανάγκες του λιανικού εμπορίου. Το σύστημα αναπτύχθηκε με χρήση Python και Streamlit, αξιοποιώντας MySQL ως υποδομή αποθήκευσης δεδομένων σε δύο επίπεδα: την ακατέργαστη βάση δεδομένων συναλλαγών (retail_db) και τη βάση επεξεργασμένων δεδομένων (retail_db_normalized).

Ο πυρήνας του συστήματος αποτελείται από ένα pipeline μηχανικής μάθησης πέντε σταδίων, το οποίο δημιουργεί χαρακτηριστικά χρονοσειράς (lag features, rolling statistics, μετεωρολογικά δεδομένα, εορτολόγιο), εκπαιδεύει οκτώ προγνωστικά μοντέλα ανά προϊόν (ARIMA, Holt-Winters, Moving Average, LightGBM, Random Forest, Extra Trees, Naive, Seasonal Naive), επιλέγει αυτόματα το βέλτιστο βάσει MAE σε χρονολογική διαχωρισμένη δοκιμαστική περίοδο (80/20) και παράγει προβλέψεις 30 ημερών με αναδρομική τεχνική για τα μοντέλα μηχανικής μάθησης. Επιπλέον, ενσωματώνεται αλγόριθμος ανίχνευσης ανωμαλιών βασισμένος στο Robust Z-Score με Μέση Απόλυτη Απόκλιση (MAD), ανθεκτικός σε ακραίες τιμές.

Το διαδραστικό dashboard εξυπηρετεί έξι θεματικές ενότητες: ανάλυση εσόδων, προβλέψεις ζήτησης, backtest αξιολόγηση, ανίχνευση ανωμαλιών, διαχείριση αποθέματος και ανάλυση πελατών. Η αξιολόγηση του συστήματος με συνθετικά δεδομένα που περιλαμβάνουν 3.223.031 γραμμές προϊόντων από 1.075.000 συναλλαγές και αφορούν 100 προϊόντα ανέδειξε ότι το ARIMA επιτυγχάνει τη χαμηλότερη μέση τιμή MAE (14,516), επιλεγόμενο ως βέλτιστο για 28 προϊόντα.

Λέξεις – Κλειδιά

επιχειρησιακή ευφυΐα, KPIs, προβλεπτική ανάλυση, μηχανική μάθηση, πρόβλεψη ζήτησης, ανίχνευση ανωμαλιών, λιανικό εμπόριο, Python, Streamlit

Development of a Sales Analysis and Forecasting System Using Machine Learning Techniques

Vasileios Laro

ID 166321

Abstract

This thesis presents the design and implementation of a comprehensive business intelligence system with predictive analytics capabilities, tailored to the needs of the retail sector. The system was developed using Python and Streamlit, with MySQL serving as the data storage infrastructure at two levels: a raw transactional database (retail_db) and a processed data database (retail_db_normalized).

The core of the system consists of a five-stage machine learning pipeline that generates time-series features (lag features, rolling statistics, meteorological data, and holiday indicators), trains eight forecasting models per product (ARIMA, Holt-Winters, Moving Average, LightGBM, Random Forest, Extra Trees, Naive, and Seasonal Naive), automatically selects the optimal model based on MAE over a temporally split test period (80/20), and produces 30-day forecasts using a recursive strategy for machine learning models. Additionally, an anomaly detection algorithm based on the Robust Z-Score with Median Absolute Deviation (MAD) is incorporated, providing resilience against outliers. The interactive dashboard serves six thematic modules: revenue analysis, demand forecasting, backtest evaluation, anomaly detection, inventory management, and customer analysis. System evaluation using synthetic data comprising 3,223,031 sale-item records from 1,075,000 transactions across 100 products demonstrated that ARIMA achieves the lowest mean MAE (14.516), being selected as the optimal model for 28 products.

Keywords

business intelligence, KPIs, predictive analytics, machine learning, demand forecasting, anomaly detection, retail, Python, Streamlit

Περιεχόμενα

Περίληψη.....	5
Abstract.....	6
Περιεχόμενα.....	7
Κατάλογος Συντομογραφιών και Ακρωνυμίων.....	9
1. Εισαγωγή.....	11
1.1 Σκοπός και κίνητρο της εργασίας.....	11
1.2 Δομή της εργασίας.....	12
2. Πεδίο του προβλήματος.....	13
2.1 Ο ρόλος των KPIs στο λιανεμπόριο.....	13
2.2 Ανασκόπηση σχετικών συστημάτων και εργαλείων.....	14
2.3 Λειτουργικές απαιτήσεις.....	15
2.4 Μη λειτουργικές απαιτήσεις.....	16
2.5 Προβλεπτική ανάλυση στο λιανεμπόριο.....	17
3. Τεχνολογίες και Εργαλεία.....	19
3.1 Επιλογή γλώσσας και πλαισίου ανάπτυξης.....	19
3.1.1 Python.....	19
3.1.2 Streamlit.....	20
3.2 Βάση δεδομένων.....	20
3.2.1 MySQL.....	20
3.2.2 SQLAlchemy.....	21
3.3 Μοντέλα Πρόβλεψης.....	22
3.3.1 Moving Average.....	22
3.3.2 Holt-Winters.....	22
3.3.3 ARIMA.....	23
3.3.4 LightGBM.....	23
3.3.5 Random Forest.....	23
3.3.6 Extra Trees.....	24
3.3.7 Naive.....	24
3.3.8 Seasonal Naive.....	24
4. Σχεδίαση συστήματος.....	25
4.1 Αρχιτεκτονική συστήματος.....	25
4.2 Σχεδίαση βάσης δεδομένων.....	26
4.2.1 Σχήμα retail_db.....	26
4.2.2 Σχήμα retail_db_normalized.....	28
4.3 Ανάλυση περιπτώσεων χρήσης.....	28
4.3.1 ΠΧ Σύνδεση χρήστη.....	28
4.3.2 ΠΧ Προβολή προβλέψεων ζήτησης.....	29

4.3.3 ΠΧ Διαχείριση αποθέματος.....	29
4.3.4 ΠΧ Ανίχνευση ανωμαλιών.....	30
4.4 UML Διαγράμματα.....	30
5. Υλοποίηση.....	40
5.1 Pipeline επεξεργασίας δεδομένων και μηχανολογία χαρακτηριστικών.....	40
5.2 Εκπαίδευση μοντέλων και επιλογή βέλτιστου.....	42
5.2.1 Διαχωρισμός δεδομένων εκπαίδευσης και ελέγχου.....	42
5.2.2 Στρατηγική αναδρομικής πρόβλεψης (Recursive Forecasting).....	42
5.3 Δημιουργία προβλέψεων.....	43
5.4 Ανίχνευση ανωμαλιών.....	44
5.5 Backend εφαρμογής.....	45
5.6 Dashboard - Καρτέλες εφαρμογής.....	47
5.6.1 Καρτέλα Revenue.....	47
5.6.2 Καρτέλα Forecast.....	50
5.6.3 Καρτέλα Backtest.....	52
5.6.4 Καρτέλα Anomalies.....	54
5.6.5 Καρτέλα Stock and Orders.....	56
5.6.6 Καρτέλα Customers.....	58
6. Αποτελέσματα και Αξιολόγηση.....	61
6.1 Μετρικές αξιολόγησης μοντέλων.....	61
6.2 Σύγκριση μοντέλων ανά προϊόν.....	62
6.3 Αποτελέσματα Backtest.....	63
7. Εγκατάσταση και Ανάπτυξη.....	64
7.1 Ρύθμιση περιβάλλοντος και βάσης δεδομένων.....	64
7.2 Διαμόρφωση Streamlit.....	65
7.3 Αυτόματη εκκίνηση στα Windows.....	65
8. Συμπεράσματα και Δυνατές επεκτάσεις.....	66
Παράρτημα Α: Οδηγίες Εγκατάστασης.....	68
A.1 Προαπαιτούμενα.....	68
A.2 Δημιουργία Εικονικού Περιβάλλοντος.....	68
A.3 Εγκατάσταση Εξαρτήσεων.....	68
A.4 Ρύθμιση Βάσης Δεδομένων.....	68
A.5 Ρύθμιση Αρχείων Παραμέτρων.....	69
A.6 Εκτέλεση ML Pipeline.....	70
A.7 Εκκίνηση Dashboard.....	70
Παράρτημα Β: Δομή Βάσης Δεδομένων.....	71
B.1 Βάση Δεδομένων retail_db.....	71
B.2 Βάση Δεδομένων retail_db_normalized.....	71
Βιβλιογραφία.....	73

Κατάλογος Συντομογραφιών και Ακρωνυμίων

Συντομογραφία	Πλήρης Όρος (Αγγλικά)	Απόδοση στα Ελληνικά
API	Application Programming Interface	Διεπαφή Προγραμματισμού Εφαρμογών
ARIMA	Autoregressive Integrated Moving Average	Αυτοπαλίνδρομο Ολοκληρωμένο Μοντέλο Κινητού Μέσου
BI	Business Intelligence	Επιχειρηματική Νοημοσύνη
CSV	Comma - Separated Values	Τιμές Διαχωρισμένες με Κόμμα
DB	Database	Βάση Δεδομένων
DSS	Decision Support System	Σύστημα Υποστήριξης Αποφάσεων
ERP	Enterprise Resource Planning	Σύστημα Διαχείρισης Επιχειρησιακών Πόρων
ETL	Extract, Transform, Load	Εξαγωγή, Μετασχηματισμός, Φόρτωση
KPI	Key Performance Indicator	Βασικός Δείκτης Απόδοσης
JSON	JavaScript Object Notation	Μορφή Ανταλλαγής Δεδομένων JavaScript
LightGBM	Light Gradient Boosting Machine	Μηχανή Ελαφριάς Ενίσχυσης Κλίσης (Αλγόριθμος Δέντρων Απόφασης)
MAD	Median Absolute Deviation	Διάμεσος Απόλυτης Απόκλισης
MAE	Mean Absolute Error	Μέσο Απόλυτο Σφάλμα
MAPE	Mean Absolute Percentage Error	Μέσο Απόλυτο Ποσοστιαίο Σφάλμα
ML	Machine Learning	Μηχανική Μάθηση

MOQ	Minimum Order Quantity	Ελάχιστη Ποσότητα Παραγγελίας
MIS	Management Information System	Πληροφοριακό Σύστημα Διοίκησης
ORM	Object-Relational Mapping	Αντικειμενοσχεσιακή Αντιστοίχιση
RMSE	Root Mean Squared Error	Τετραγωνική Ρίζα Μέσου Τετραγωνικού Σφάλματος
SKU	Stock Keeping Unit	Μονάδα Διατήρησης Αποθέματος
SQL	Structured Query Language	Δομημένη Γλώσσα Ερωτημάτων
UI	User Interface	Διεπαφή Χρήστη
YAML	YAML Ain't Markup Language	-

1. Εισαγωγή

1.1 Σκοπός και κίνητρο της εργασίας

Στο σύγχρονο περιβάλλον του εμπορίου, η αποτελεσματική διαχείριση ενός καταστήματος απαιτεί ταχεία πρόσβαση σε αξιόπιστες πληροφορίες σχετικά με την απόδοση των πωλήσεων, τα επίπεδα αποθέματος και τη συμπεριφορά των πελατών. Ωστόσο, τις περισσότερες φορές οι αποφάσεις λαμβάνονται βάσει εμπειρίας ή με καθυστέρηση, απουσία ενός ολοκληρωμένου συστήματος παρακολούθησης επιχειρησιακών δεικτών απόδοσης (KPIs). Η έλλειψη αυτή οδηγεί σε προβλήματα υπεραποθέματος ή έλλειψης προϊόντων, σε αδυναμία έγκαιρης ανίχνευσης ανωμαλιών στις πωλήσεις και σε χαμένες ευκαιρίες βελτιστοποίησης εσόδων. Σκοπός της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη ενός ολοκληρωμένου backend συστήματος επιχειρησιακών δεικτών απόδοσης με δυνατότητα προβλεπτικής ανάλυσης, το οποίο απευθύνεται σε διαχειριστές καταστημάτων λιανικής. Το σύστημα αξιοποιεί τεχνικές μηχανικής μάθησης για την πρόβλεψη της μελλοντικής ζήτησης ανά προϊόν, ενσωματώνει μηχανισμούς ανίχνευσης ανωμαλιών στα δεδομένα πωλήσεων και παρέχει εργαλεία υποστήριξης αποφάσεων για τη διαχείριση αποθέματος και αναπαραγγελιών. Όλες οι παραπάνω λειτουργίες είναι προσβάσιμες μέσω ενός διαδραστικού πίνακα ελέγχου (dashboard) που αναπτύχθηκε με τη χρήση του πλαισίου Streamlit.

Το κίνητρο για την ανάπτυξη του συστήματος προέκυψε από την ανάγκη αντικατάστασης χειροκίνητων και χρονοβόρων διαδικασιών με ένα αυτοματοποιημένο εργαλείο που παρέχει άμεση και τεκμηριωμένη πληροφόρηση. Για τον σκοπό αυτό αξιοποιούνται τόσο στατιστικά μοντέλα χρονοσειρών (Holt-Winters, ARIMA, Moving Average, Naive, Seasonal Naive) όσο και αλγόριθμοι μηχανικής μάθησης (LightGBM, Random Forest, Extra Trees), με αυτόματη επιλογή του βέλτιστου μοντέλου ανά προϊόν. Το σύστημα δίνει τη δυνατότητα λήψης αποφάσεων βάσει δεδομένων, μειώνοντας σημαντικά το λειτουργικό κόστος και την αβεβαιότητα στη διαχείριση του καταστήματος.

Ζητήματα όπως η προέλευση και ο όγκος των δεδομένων, η μεθοδολογία επιλογής μοντέλων και οι επιχειρησιακοί περιορισμοί αναλύονται διεξοδικά στα επόμενα κεφάλαια.

1.2 Δομή της εργασίας

Η παρούσα εργασία οργανώνεται σε οκτώ κεφάλαια ως εξής:

Στο Κεφ. 2 παρουσιάζεται το πεδίο του προβλήματος, αναλύεται ο ρόλος των KPIs στο σύγχρονο λιανεμπόριο, και πραγματοποιείται ανασκόπηση σχετικών συστημάτων και εργαλείων. Επίσης, καταγράφονται οι λειτουργικές και μη λειτουργικές απαιτήσεις του συστήματος.

Στο Κεφ. 3 περιγράφονται οι τεχνολογίες και τα εργαλεία που επιλέχθηκαν για την ανάπτυξη του συστήματος, με αιτιολόγηση κάθε επιλογής.

Στο Κεφ. 4 παρουσιάζεται η σχεδίαση του συστήματος, η αρχιτεκτονική του, το σχήμα της ΒΔ, η ανάλυση περιπτώσεων χρήσης και τα αντίστοιχα UML διαγράμματα.

Στο Κεφ. 5 αναλύεται η υλοποίηση του συστήματος, από το pipeline επεξεργασίας δεδομένων και την εκπαίδευση των μοντέλων μηχανικής μάθησης έως την ανάπτυξη του dashboard και των επιμέρους καρτελών του.

Στο Κεφ. 6 παρουσιάζονται τα αποτελέσματα της αξιολόγησης των μοντέλων με βάση τις μετρικές MAE, RMSE και MAPE, καθώς και τα αποτελέσματα του backtest.

Στο Κεφ. 7 περιγράφεται η διαδικασία εγκατάστασης και ανάπτυξης του συστήματος σε περιβάλλον Windows.

Τέλος, στο Κεφ. 8 παρουσιάζονται τα συμπεράσματα που προέκυψαν από την εκπόνηση της εργασίας και προτείνονται δυνατές μελλοντικές επεκτάσεις.

2. Πεδίο του προβλήματος

Το παρόν κεφάλαιο πραγματεύεται το θεωρητικό και επιχειρησιακό πλαίσιο στο οποίο εντάσσεται η παρούσα εργασία. Αρχικά αναλύεται η σημασία των επιχειρησιακών δεικτών απόδοσης στο λιανεμπόριο και τεκμηριώνεται η ανάγκη για ένα ολοκληρωμένο σύστημα υποστήριξης αποφάσεων. Στη συνέχεια παρουσιάζονται σχετικά συστήματα και εργαλεία που έχουν αναπτυχθεί για παρόμοιους σκοπούς, ενώ το κεφάλαιο ολοκληρώνεται με την καταγραφή των λειτουργικών και μη λειτουργικών απαιτήσεων του συστήματος που αναπτύχθηκε.

2.1 Ο ρόλος των KPIs στο λιανεμπόριο

Οι επιχειρησιακοί δείκτες απόδοσης (Key Performance Indicators - KPIs) αποτελούν μετρήσιμες τιμές που κατοπτρίζουν την αποτελεσματικότητα μιας επιχείρησης ως προς την επίτευξη των στόχων της. Στο λιανεμπόριο, οι KPIs καλύπτουν ένα ευρύ φάσμα λειτουργιών, από την παρακολούθηση των εσόδων και των κύκλων εργασιών έως τη διαχείριση αποθέματος και την ανάλυση της συμπεριφοράς των πελατών.

Η σημασία των KPIs στον κλάδο του λιανεμπορίου έχει αυξηθεί σημαντικά τα τελευταία χρόνια, καθώς ο ανταγωνισμός εντείνεται και τα περιθώρια κέρδους συρρικνώνονται. Η ικανότητα λήψης αποφάσεων βάσει δεδομένων, και όχι αποκλειστικά βάσει εμπειρίας, αποτελεί πλέον ανταγωνιστικό πλεονέκτημα. Ενδεικτικά, η έγκαιρη παρακολούθηση του ρυθμού πωλήσεων ανά προϊόν επιτρέπει την έγκαιρη αναπαραγγελία αποθέματος, ενώ η ανίχνευση ανωμαλιών στα δεδομένα συναλλαγών μπορεί να αποτρέψει απώλεια εσόδων λόγω ελλείψεων ή λανθασμένης τιμολόγησης.

Παρά τη σημασία τους, πολλές μικρομεσαίες επιχειρήσεις λιανικής εξακολουθούν να μην διαθέτουν εργαλεία συστηματικής παρακολούθησης KPIs. Οι αποφάσεις λαμβάνονται συχνά με

καθυστέρηση, βάσει περιοδικών αναφορών ή ακόμα και χειροκίνητης επεξεργασίας δεδομένων σε υπολογιστικά φύλλα. Το κενό αυτό επιχειρεί να καλύψει το σύστημα που αναπτύχθηκε στην παρούσα εργασία, παρέχοντας σε πραγματικό χρόνο ορατότητα στους βασικούς δείκτες απόδοσης του καταστήματος, με ταυτόχρονη δυνατότητα προβλεπτικής ανάλυσης.

2.2 Ανασκόπηση σχετικών συστημάτων και εργαλείων

Η ανάγκη για συστήματα παρακολούθησης επιχειρησιακών δεικτών και πρόβλεψης ζήτησης στο λιανεμπόριο έχει οδηγήσει στην ανάπτυξη πλήθους εμπορικών και ακαδημαϊκών λύσεων. Στην παρούσα ενότητα παρουσιάζονται ορισμένα από τα πιο αντιπροσωπευτικά, με σκοπό να αναδειχθεί η θέση της παρούσας εργασίας σε σχέση με την υπάρχουσα βιβλιογραφία και τεχνολογία.

Στον χώρο των εμπορικών λύσεων, συστήματα όπως το SAP Retail (SAP SE, 2024) και το Oracle Retail (Oracle Corporation, 2024) παρέχουν ολοκληρωμένες λειτουργίες διαχείρισης αποθέματος, πρόβλεψης ζήτησης και παρακολούθησης KPIs. Ωστόσο, απευθύνονται κυρίως σε μεγάλες αλυσίδες λιανικής και συνεπάγονται υψηλό κόστος αδειοδότησης και υλοποίησης, καθιστώντας τα απρόσιτα για μικρομεσαίες επιχειρήσεις. Παρόμοια χαρακτηριστικά έχουν και εργαλεία Business Intelligence όπως το Microsoft Power BI (Microsoft Corporation, 2024) το Tableau (Tableau Software, 2024), τα οποία προσφέρουν δυνατότητες οπτικοποίησης KPIs αλλά δεν ενσωματώνουν εγγενώς μηχανισμούς προβλεπτικής ανάλυσης ή διαχείρισης αποθέματος.

Στον ακαδημαϊκό χώρο, έχουν προταθεί διάφορες προσεγγίσεις για την πρόβλεψη ζήτησης στο λιανεμπόριο με χρήση μεθόδων μηχανικής μάθησης. Οι Fildes et al. (2019) συγκρίνουν στατιστικές και ML μεθόδους για retail forecasting, καταλήγοντας ότι τα gradient boosting μοντέλα υπερτερούν σε σύνθετα περιβάλλοντα με πολλαπλούς εξωτερικούς παράγοντες.

Αντίστοιχα, οι Makridakis et al. (2018) στο πλαίσιο του διαγωνισμού M4 καταδεικνύουν ότι

υβριδικές προσεγγίσεις που συνδυάζουν στατιστικά μοντέλα (SARIMA) με μοντέλα ML επιτυγχάνουν βελτιωμένη ακρίβεια πρόβλεψης.

Σε ό,τι αφορά εργαλεία ανοιχτού κώδικα, πλατφόρμες όπως το Metabase (Metabase Inc., 2024) και το Apache Superset (Apache Software Foundation, 2024) παρέχουν δυνατότητες δημιουργίας dashboards συνδεδεμένων με βάσεις δεδομένων, χωρίς όμως να ενσωματώνουν pipeline εκπαίδευσης μοντέλων. Το Streamlit (Streamlit Inc., 2024), που χρησιμοποιείται στην παρούσα εργασία, διαφέρει ουσιαστικά καθώς επιτρέπει την ενοποίηση του ML pipeline με το interface σε ενιαίο Python περιβάλλον.

Συγκριτικά με τις παραπάνω λύσεις, το σύστημα που αναπτύχθηκε στην παρούσα εργασία διαφοροποιείται ως προς τρία βασικά σημεία: πρώτον, ενσωματώνει πλήρες ML pipeline εκπαίδευσης και επιλογής βέλτιστου μοντέλου ανά προϊόν. Δεύτερον, λαμβάνει υπόψη επιχειρησιακούς περιορισμούς όπως η διάρκεια ζωής των προϊόντων και τρίτον, αναπτύχθηκε αποκλειστικά με εργαλεία ανοιχτού κώδικα, χωρίς κόστος αδειοδότησης, καθιστώντας το προσβάσιμο σε επιχειρήσεις με περιορισμένους πόρους.

2.3 Λειτουργικές απαιτήσεις

Οι λειτουργικές απαιτήσεις περιγράφουν τις συγκεκριμένες δυνατότητες που πρέπει να παρέχει το σύστημα στον τελικό χρήστη. Ο τελικός χρήστης του συστήματος είναι ο διαχειριστής καταστήματος λιανικής, ο οποίος χρειάζεται άμεση πρόσβαση σε πληροφορίες για την απόδοση του καταστήματος, την κατάσταση του αποθέματος και τις προβλέψεις ζήτησης, χωρίς να απαιτείται εξειδίκευση σε τεχνικά θέματα.

Οι λειτουργικές απαιτήσεις του συστήματος είναι οι εξής:

1. Δυνατότητα αυθεντικοποίησης χρήστη μέσω ονόματος χρήστη και κωδικού πρόσβασης.

2. Παρουσίαση συνολικών εσόδων και αριθμού συναλλαγών ανά προϊόν και χρονική περίοδο.
3. Δυνατότητα αναζήτησης εσόδων με φίλτρα ημερομηνίας, κατηγορίας και προϊόντος.
4. Προβολή 30ήμερης πρόβλεψης ζήτησης για κάθε προϊόν, βάσει του βέλτιστου εκπαιδευμένου μοντέλου.
5. Σύγκριση απόδοσης μοντέλων μηχανικής μάθησης ανά προϊόν με βάση τις μετρικές MAE, RMSE και MAPE.
6. Αξιολόγηση ακρίβειας προβλέψεων μέσω backtest σε ιστορικά δεδομένα.
7. Ανίχνευση και προβολή ανωμαλιών στα δεδομένα πωλήσεων ανά προϊόν.
8. Παρουσίαση watchlist αναπαραγγελίας με αυτόματο υπολογισμό ποσότητας παραγγελίας βάση πρόβλεψης, τρέχοντος αποθέματος και παραμέτρων ασφαλείας.
9. Δυνατότητα επεξεργασίας παραμέτρων αποθέματος ανά προϊόν (lead time, safety days, minimum order quantity).
10. Καταγραφή και εμφάνιση ιστορικού παραδόσεων και χειροκίνητης ενημέρωσης αποθέματος.
11. Προβολή leaderboard κορυφαίων πελατών και ιστορικού παραγγελίας ανά πελάτη.
12. Πλοήγηση μεταξύ καρτελών χωρίς επαναφόρτωση ολόκληρης της εφαρμογής.

2.4 Μη λειτουργικές απαιτήσεις

Οι μη λειτουργικές απαιτήσεις αφορούν τα ποιοτικά χαρακτηριστικά του συστήματος και τις συνθήκες λειτουργίας του, ανεξάρτητα από τις επιμέρους λειτουργίες που παρέχει.

Απόδοση (Performance)

Το σύστημα οφείλει να ανταποκρίνεται σε ερωτήματα χρήστη χωρίς αισθητή καθυστέρηση. Για τον σκοπό αυτό, όλα τα ερωτήματα προς τη βάση δεδομένων εκτελούνται μέσω cached συναρτήσεων με χρόνο ανανέωσης πέντε λεπτών, ενώ η απόδοση της εφαρμογής ενισχύεται περαιτέρω μέσω lazy rendering, δηλαδή μόνο η ενεργή καρτέλα εκτελείται κατά κάθε αλληλεπίδραση του χρήστη.

Ασφάλεια (Security)

Οι κωδικοί πρόσβασης δεν αποθηκεύονται σε μορφή απλού κειμένου, αλλά ως κρυπτογραφικές συναρτήσεις κατακερματισμού SHA-256, εξασφαλίζοντας ότι δεν είναι δυνατή η ανάκτησή τους ακόμα και σε περίπτωση μη εξουσιοδοτημένης πρόσβασης στο αρχείο διαμόρφωσης.

Χρηστικότητα (Usability)

Το σύστημα απευθύνεται σε χρήστες χωρίς τεχνική εξειδίκευση. Η πλοήγηση είναι απλή και βασίζεται σε οριζόντιο μενού καρτελών, ενώ οι πληροφορίες παρουσιάζονται σε μορφή πινάκων, γραφημάτων και συνοπτικών καρτών (metric cards).

Συντηρησιμότητα (Maintainability)

Ο κώδικας είναι οργανωμένος σε ανεξάρτητα modules ανά καρτέλα (tabs/), με κοινές συναρτήσεις σε αρχείο utils.py. Η αρχιτεκτονική αυτή επιτρέπει την ανεξάρτητη τροποποίηση κάθε τμήματος χωρίς επίδραση στα υπόλοιπα.

Φορητότητα (Portability)

Το σύστημα αναπτύχθηκε και δοκιμάστηκε σε περιβάλλον Windows. Η αρχιτεκτονική του, βασισμένη αποκλειστικά σε εργαλεία ανοιχτού κώδικα, επιτρέπει θεωρητικά τη μεταφορά του σε οποιοδήποτε λειτουργικό σύστημα ή περιβάλλον cloud χωρίς ουσιαστικές τροποποιήσεις.

Διαθεσιμότητα (Availability)

Το σύστημα εκκινεί αυτόματα κατά την εκκίνηση του λειτουργικού συστήματος μέσω μηχανισμού Startup shortcut, εξασφαλίζοντας ότι είναι διαθέσιμο στον χρήστη χωρίς χειροκίνητη ενέργεια.

2.5 Προβλεπτική ανάλυση στο λιανεμπόριο

Η προβλεπτική ανάλυση αποτελεί εφαρμογή των μεθόδων στατιστικής και μηχανικής μάθησης για την εξαγωγή συμπερασμάτων και πρόβλεψης μελλοντικών τιμών βάσει ιστορικών δεδομένων. Στο πλαίσιο του λιανεμπορίου, η πρόβλεψη ζήτησης αποτελεί έναν από τους κρίσιμότερους τομείς εφαρμογής, καθώς η ακριβής πρόβλεψη επηρεάζει άμεσα

τη διαχείριση αποθεμάτων, τον προγραμματισμό παραγγελιών και την ελαχιστοποίηση απωλειών λόγω αποθεματικής έλλειψης ή υπερβολικών αποθεμάτων.

Οι προκλήσεις της πρόβλεψης ζήτησης στο λιανεμπόριο είναι πολλαπλές. Η ζήτηση επηρεάζεται από εποχικότητα, εβδομαδιαίους κύκλους, διαφημιστικές ενέργειες, καιρικές συνθήκες και μη προβλέψιμα γεγονότα (Taylor & Letham, 2018). Επιπλέον, η ετερογένεια των προϊόντων - διαφορετικά μοτίβα ζήτησης, διάρκειες ζωής και κατηγορίες - καθιστά αδύνατη την εφαρμογή ενός ενιαίου μοντέλου για όλα τα SKU (Fildes, Ma & Kolassa, 2019· Hyndman & Koehler, 2006).

Η δημιουργία κατάλληλων χαρακτηριστικών (feature engineering) αποτελεί κρίσιμο στάδιο στην εφαρμογή μεθόδων μηχανικής μάθησης σε χρονοσειρές. Τεχνικές όπως τα lag features και τα rolling window στατιστικά επιτρέπουν στα μοντέλα να αντιλαμβάνονται χρονικές εξαρτήσεις και μοτίβα ζήτησης, μετατρέποντας το πρόβλημα πρόβλεψης σε πρόβλημα επιβλεπόμενης μάθησης (Bontempi, Ben Taieb & Le Borgne, 2013).

Η έρευνα M4 Competition (Makridakis, Spiliotis & Assimakopoulos, 2018) ανέδειξε ότι η συνδυαστική χρήση πολλαπλών μοντέλων και η αυτόματη επιλογή βέλτιστου ανά χρονοσειρά αποδίδει συστηματικά καλύτερα αποτελέσματα από την εφαρμογή ενός μοναδικού μοντέλου. Η αρχή αυτή αποτέλεσε τη βάση σχεδιασμού του pipeline εκπαίδευσης της παρούσας εργασίας, στο οποίο εκπαιδεύονται οκτώ διαφορετικά μοντέλα ανά προϊόν και επιλέγεται αυτόματα το βέλτιστο βάσει MAE στο σύνολο ελέγχου.

3. Τεχνολογίες και Εργαλεία

Στο παρόν κεφάλαιο παρουσιάζονται και αιτιολογούνται οι τεχνολογικές επιλογές που έγιναν για την ανάπτυξη του συστήματος. Για κάθε εργαλείο αναφέρονται τα βασικά χαρακτηριστικά του και οι λόγοι για τους οποίους επιλέχθηκε έναντι εναλλακτικών λύσεων.

3.1 Επιλογή γλώσσας και πλαισίου ανάπτυξης

3.1.1 Python

Η Python επιλέχθηκε ως η κύρια γλώσσα ανάπτυξης του συστήματος. Πρόκειται για μια γλώσσα γενικής χρήσης υψηλού επιπέδου, η οποία έχει επικρατήσει στον χώρο της επιστήμης δεδομένων και της μηχανικής μάθησης, χάρη στις εκτεταμένες βιβλιοθήκες της. Βασικοί λόγοι επιλογής της είναι:

- Η διαθεσιμότητα εξειδικευμένων βιβλιοθηκών για επεξεργασία δεδομένων (pandas, numpy), μηχανική μάθηση (scikit-learn, LightGBM) και στατιστική ανάλυση (statsmodels).
- Η ευρεία χρήση της στην ακαδημαϊκή και επαγγελματική κοινότητα, που διασφαλίζει συνεχή υποστήριξη και τεκμηρίωση.
- Η απλή σύνταξή της που επιτρέπει γρήγορη ανάπτυξη και εύκολη συντήρηση κώδικα.

Εναλλακτικές γλώσσες για ανάπτυξη συστήματος ανάλυσης δεδομένων αποτελούν η R και η Julia. Η R παρέχει ισχυρό στατιστικό οικοσύστημα αλλά υστερεί σε γενικού σκοπού ανάπτυξη εφαρμογών και ολοκλήρωση με βάσεις δεδομένων. Η Julia προσφέρει υψηλές επιδόσεις σε αριθμητικούς υπολογισμούς αλλά διαθέτει μικρότερο οικοσύστημα βιβλιοθηκών μηχανικής μάθησης. Python επιλέχθηκε λόγω της πληρότητας του οικοσυστήματός της (pandas, scikit-learn, statsmodels, LightGBM), της ευρείας υποστήριξης από τη βιβλιογραφία και της άμεσης ολοκλήρωσης με το Streamlit (Van Rossum & Drake, 2009).

3.1.2 Streamlit

Το Streamlit είναι ένα πλαίσιο ανάπτυξης (framework) ανοιχτού κώδικα για Python, το οποίο επιτρέπει τη δημιουργία διαδραστικών web εφαρμογών αποκλειστικά μέσω Python κώδικα, χωρίς γνώσεις HTML, CSS ή JavaScript. Επιλέχθηκε για τους εξής λόγους:

- Επιτρέπει την ενοποίηση του ML Pipeline με το interface σε ενιαίο περιβάλλον, χωρίς την ανάγκη ξεχωριστού backend και frontend.
- Παρέχει έτοιμα components για γραφήματα, πίνακες, φίλτρα και metric cards, μειώνοντας σημαντικά τον χρόνο ανάπτυξης.
- Υποστηρίζει μηχανισμό caching που βελτιστοποιεί την απόδοση σε εφαρμογές με συχνά ερωτήματα ΒΔ.
- Λειτουργεί headless σε τοπικό server, κατάλληλο για deployment σε περιβάλλον Windows, χωρίς επιπλέον υποδομή.

Εναλλακτικές για την ανάπτυξη του frontend αποτελούν τα Flask, Dash και Gradio. Το Flask απαιτεί ξεχωριστή ανάπτυξη frontend (HTML/CSS/JS) και σημαντικά μεγαλύτερο χρόνο υλοποίησης. Το Dash (Plotly) είναι κατάλληλο για dashboards αλλά απαιτεί callback-based αρχιτεκτονική που αυξάνει την πολυπλοκότητα. Το Streamlit επιλέχθηκε διότι επιτρέπει ανάπτυξη διαδραστικής εφαρμογής αποκλειστικά σε Python, με ενσωματωμένο session state, caching και native υποστήριξη Plotly, χωρίς να απαιτείται γνώση web τεχνολογιών (Streamlit Inc., 2024)

3.2 Βάση δεδομένων

3.2.1 MySQL

Ως σύστημα διαχείρισης ΒΔ επιλέχθηκε η MySQL, μια από τις πιο διαδεδομένες σχεσιακές βάσεις δεδομένων ανοιχτού κώδικα. Η MySQL οργανώνει τα δεδομένα σε πίνακες με σαφώς ορισμένες σχέσεις μεταξύ τους, επιτρέποντας την εκτέλεση σύνθετων ερωτημάτων μέσω της γλώσσας SQL.

Στην παρούσα εργασία χρησιμοποιούνται δύο ξεχωριστές ΒΔ με διακριτό ρόλο η καθεμία:

- **retail_db**: Περιέχει τα ακατέργαστα (raw) δεδομένα συναλλαγών. Ο πίνακας `full_retail_dataset` αποθηκεύει όλες τις συναλλαγές από 01/01/2022 και αποτελεί την κύρια πηγή για την ανάλυση εσόδων και την εξαγωγή ιστορικών στατιστικών.
- **retail_db_normalized**: Περιέχει τόσο την κανονικοποιημένη αναπαράσταση των πρωτογενών δεδομένων (customers, products, sales, sale_items, promotions, weather_daily, holidays) όσο και τα επεξεργασμένα δεδομένα που παράγουν τα scripts: χαρακτηριστικά ML, εκπαιδευμένα μοντέλα, προβλέψεις, ανωμαλίες και παραμέτρους αποθέματος.

Βασικοί λόγοι επιλογής της MySQL είναι η αξιοπιστία της, η ευρεία τεκμηρίωσή της, η συμβατότητα της με Python μέσω διαφόρων connectors και το μηδενικό κόστος αδειοδότησης της Community Edition.

Εναλλακτικές σχεσιακές βάσεις δεδομένων αποτελούν η PostgreSQL και η SQLite. Η SQLite αποκλείστηκε λόγω περιορισμών σε ταυτόχρονες εγγραφές και μέγεθος δεδομένων. Η PostgreSQL αποτελεί ισχυρή εναλλακτική, ωστόσο η MySQL επιλέχθηκε λόγω ευρύτερης υποστήριξης σε Windows περιβάλλοντα ανάπτυξης, απλούστερης ρύθμισης και πλήρους συμβατότητας με SQLAlchemy και pandas. Και οι δύο βάσεις (retail_db, retail_db_normalized) εκμεταλλεύονται το InnoDB storage engine για υποστήριξη foreign keys και transactions (MySQL Community Edition, 2024).

3.2.2 SQLAlchemy

Το SQLAlchemy είναι μία βιβλιοθήκη Python που παρέχει ένα ενιαίο επίπεδο επικοινωνίας με βάσεις δεδομένων, ανεξάρτητα από το υποκείμενο σύστημα. Στην παρούσα εργασία χρησιμοποιείται για τη δημιουργία του αντικειμένου engine μέσω του οποίου εκτελούνται όλα τα ερωτήματα SQL από την εφαρμογή Streamlit. Η χρήση του SQLAlchemy προσφέρει δύο βασικά πλεονεκτήματα: πρώτον, διαχειρίζεται αυτόματα τις συνδέσεις με τη ΒΔ και δεύτερον, επιτρέπει την άμεση φόρτωση αποτελεσμάτων σε αντικείμενα pandas DataFrame μέσω της συνάρτησης `pd.read_sql()`, απλοποιώντας σημαντικά τον κώδικα ανάκτησης δεδομένων.

Εναλλακτικά, η σύνδεση με τη βάση θα μπορούσε να γίνει απευθείας με `mysql-connector-python`. Η SQLAlchemy επιλέχθηκε διότι παρέχει ενιαίο επίπεδο

αφαίρεσης που επιτρέπει αλλαγή βάσης δεδομένων χωρίς αλλαγές στον κώδικα, ασφαλή διαχείριση παραμέτρων ερωτημάτων (αποτροπή SQL injection) και άριστη ολοκλήρωση με pandas μέσω `read_sql()` και `to_sql()` (SQLAlchemy, 2024).

3.3 Μοντέλα Πρόβλεψης

Για την πρόβλεψη ζήτησης εκπαιδεύτηκαν οκτώ διαφορετικά μοντέλα ανά προϊόν. Η επιλογή πολλαπλών μοντέλων έγινε σκόπιμα, ώστε το σύστημα να επιλέγει αυτόματα το βέλτιστο για κάθε προϊόν βάσει της μετρικής MAE στο σύνολο ελέγχου. Διαφορετικά προϊόντα εμφανίζουν διαφορετικά μοτίβα ζήτησης, και ένα μοντέλο που αποδίδει καλά για ένα προϊόν δεν εγγυάται αντίστοιχη απόδοση για κάποιο άλλο. Τα μοντέλα χωρίζονται σε τρεις κατηγορίες: στατιστικά μοντέλα χρονοσειρών (ARIMA, Holt-Winters, Moving Average), μοντέλα μηχανικής μάθησης (LightGBM, Random Forest, Extra Trees) και απλά μοντέλα αναφοράς / baseline (Naive, Seasonal Naive). Όλα τα μοντέλα αξιολογούνται ισότιμα σε κάθε προϊόν βάσει MAE στο σύνολο ελέγχου, χωρίς κανένα να λειτουργεί ως μηχανισμός fallback.

3.3.1 Moving Average

Το Moving average (Κινητός μέσος όρος) είναι η απλούστερη μέθοδος πρόβλεψης που συμπεριλήφθηκε στη σύγκριση. Η πρόβλεψη για κάθε βήμα υπολογίζεται ως ο μέσος όρος των τελευταίων 7 ημερών, με αναδρομική εφαρμογή για τον ορίζοντα των 30 ημερών. Χρησιμοποιείται ως baseline μοντέλο για την αξιολόγηση της προστιθέμενης αξίας των πιο σύνθετων μεθόδων (Makridakis, Wheelwright & Hyndman, 1998).

3.3.2 Holt-Winters

Το Holt-Winters (Εκθετική Εξομάλυνση) είναι ένα στατιστικό μοντέλο που μοντελοποιεί ταυτόχρονα τάση και εποχικότητα μέσω τριών εξομαλυντικών παραμέτρων. Στην υλοποίηση που χρησιμοποιήθηκε, εφαρμόζεται προσθετική τάση και προσθετική

εποχικότητα περιόδου 7 ημερών, εφόσον υπάρχουν επαρκή δεδομένα. Σε διαφορετική περίπτωση χρησιμοποιείται μόνο η συνιστώσα της τάσης (Holt, 2004).

3.3.3 ARIMA

Το ARIMA (AutoRegressive Integrated Moving Average) είναι ένα κλασικό στατιστικό μοντέλο χρονοσειρών που συνδυάζει αυτοπαλίνδρομα (AR), ολοκληρωμένα (I) και κινητού μέσου (MA) στοιχεία. Στην παρούσα εργασία χρησιμοποιείται με παραμέτρους (1,1,1), κατάλληλες για χρονοσειρές με γραμμική τάση. Συμπεριλήφθηκε ως αντιπροσωπευτική παραδοσιακή στατιστική μέθοδος (Box & Jenkins, 1976).

3.3.4 LightGBM

Το LightGBM (Light Gradient Boosting Machine) είναι ένας αλγόριθμος gradient boosting που αναπτύχθηκε από τη microsoft. Βασίζεται στη δημιουργία διαδοχικών δέντρων αποφάσεων, όπου κάθε νέο δέντρο διορθώνει τα σφάλματα των προηγούμενων. Χαρακτηρίζεται από υψηλή ταχύτητα εκπαίδευσης και αποτελεσματική διαχείριση μεγάλου αριθμού χαρακτηριστικών. Στην παρούσα εργασία εφαρμόζεται με αναδρομική στρατηγική πρόβλεψης (recursive forecasting), όπου κάθε πρόβλεψη χρησιμοποιείται ως είσοδος για την επόμενη (Ke et al., 2017).

3.3.5 Random Forest

Ο Random Forest είναι ένας αλγόριθμος ensemble learning που βασίζεται στην εκπαίδευση μεγάλου αριθμού ανεξάρτητων δέντρων αποφάσεων και στη συνδυαστική χρήση των προβλέψεών τους μέσω αριθμητικού μέσου. Χαρακτηρίζεται από υψηλή ανθεκτικότητα σε θόρυβο και ακραίες τιμές, και εφαρμόζεται επίσης με αναδρομική στρατηγική πρόβλεψης (Breiman, 2001).

3.3.6 Extra Trees

Ο Extra Trees (Extremely Randomized Trees) είναι παρόμοιος με τον Random Forest, με τη διαφορά ότι εισάγει επιπλέον τυχαιότητα στην επιλογή των ορίων διαχωρισμού κάθε κόμβου. Αυτό μειώνει περαιτέρω τη διακύμανση του μοντέλου και συχνά οδηγεί σε ταχύτερη εκπαίδευση. Συμπεριλήφθηκε ως εναλλακτική του Random Forest για να εξεταστεί αν η επιπλέον τυχαιότητα βελτιώνει την πρόβλεψη σε ορισμένα προϊόντα (Geurts et al., 2006).

3.3.7 Naive

Το Naive αποτελεί ένα απλό μοντέλο αναφοράς (baseline), το οποίο προβλέπει ότι η ζήτηση κάθε μελλοντικής ημέρας θα ισούται με την τελευταία παρατηρούμενη τιμή του συνόλου εκπαίδευσης. Παρά την απλότητά του, αξιολογείται ισότιμα με τα υπόλοιπα μοντέλα σε όλα τα προϊόντα βάσει MAE. Στα αποτελέσματα (Κεφ. 6.2), το Naive αναδεικνύεται ως το βέλτιστο μοντέλο για 14 από τα 100 προϊόντα, γεγονός που υποδεικνύει ότι για προϊόντα με σχετικά σταθερή ή χαμηλής μεταβλητότητας ζήτηση, ένα απλό μοντέλο αναφοράς μπορεί να αποδώσει εξίσου καλά ή και καλύτερα από πιο σύνθετα μοντέλα.

3.3.8 Seasonal Naive

Το Seasonal Naive αποτελεί επέκταση του Naive που λαμβάνει υπόψη την εβδομαδιαία εποχικότητα της ζήτησης: η πρόβλεψη κάθε μελλοντικής ημέρας ισούται με την παρατηρούμενη τιμή της αντίστοιχης ημέρας, μία εβδομάδα (7 ημέρες) πριν. Αξιολογείται εξίσου με τα υπόλοιπα μοντέλα σε όλα τα προϊόντα. Στο σύνολο δεδομένων της εργασίας, το Seasonal Naive εμφάνισε τη μεγαλύτερη μέση MAE (20.04) και δεν αναδείχθηκε ως βέλτιστο για κανένα προϊόν, ωστόσο η συμπερίληψή του διασφαλίζει ότι η αυτόματη επιλογή μοντέλου συγκρίνεται πάντα και με ένα απλό εποχικό σημείο αναφοράς

4. Σχεδίαση συστήματος

Στο παρόν κεφάλαιο περιγράφεται η σχεδίαση του συστήματος, από την αρχιτεκτονική και τη δομή της βάσης δεδομένων έως την ανάλυση των περιπτώσεων χρήσης και τα αντίστοιχα UML διαγράμματα. Στόχος είναι να αποτυπωθεί με σαφήνεια η δομή και η λογική του συστήματος πριν την παρουσίαση της υλοποίησης στο Κεφ. 5.

4.1 Αρχιτεκτονική συστήματος

Το σύστημα ακολουθεί αρχιτεκτονική τριών επιπέδων, η οποία αποτελεί κοινή πρακτική στην ανάπτυξη συστημάτων υποστήριξης αποφάσεων (Laudon & Laudon, 2020):

1. Επίπεδο δεδομένων: Δύο MySQL βάσεις δεδομένων - η `retail_db` για τα ακατέργαστα δεδομένα συναλλαγών και η `retail_db_normalized` για τα επεξεργασμένα δεδομένα, τις προβλέψεις και τα αποτελέσματα των μοντέλων.

2. Επίπεδο επεξεργασίας: Πέντε Python scripts που εκτελούνται ανεξάρτητα και διαχειρίζονται το ML pipeline:

- `1_create_ml_features.py` — δημιουργία χαρακτηριστικών
- `2_train_model.py` — εκπαίδευση μοντέλων και επιλογή βέλτιστου
- `3_generate_forecasts.py` — παραγωγή μελλοντικών προβλέψεων
- `4_detect_anomalies.py` — ανίχνευση ανωμαλιών
- `5_generate_backtest_forecasts.py` — παραγωγή backtest προβλέψεων για αξιολόγηση

3. Επίπεδο παρουσίασης: Η εφαρμογή Streamlit που διαβάζει δεδομένα από τη βάση και τα παρουσιάζει στον χρήστη μέσω διαδραστικού dashboard.

Η επικοινωνία μεταξύ των επιπέδων γίνεται αποκλειστικά μέσω της ΒΔ, τα scripts γράφουν αποτελέσματα στη βάση και η εφαρμογή τα διαβάζει. Δεν υπάρχει άμεση κλήση μεταξύ scripts και εφαρμογής, γεγονός που εξασφαλίζει χαλαρή σύζευξη (loose coupling) και ανεξάρτητη συντήρηση κάθε επιπέδου (Fowler, 2002).

4.2 Σχεδίαση βάσης δεδομένων

Η βάση δεδομένων αποτελεί τον κεντρικό κόμβο του συστήματος. Όλα τα scripts γράφουν σε αυτήν και η εφαρμογή Streamlit διαβάζει από αυτήν. Όπως αναφέρθηκε στην ενότητα 4.1, χρησιμοποιούνται δύο ξεχωριστές βάσεις με διακριτό ρόλο η καθεμία, σύμφωνα με την αρχή του διαχωρισμού ακατέργαστων και επεξεργασμένων δεδομένων (Kimball & Ross, 2013).

4.2.1 Σχήμα retail_db

Η βάση retail_db περιέχει τα πρωτογενή δεδομένα συναλλαγών και αποτελεί την κύρια πηγή για την ανάλυση εσόδων. Κεντρικός πίνακας είναι ο full_retail_dataset, ο οποίος περιλαμβάνει συνθετικά δεδομένα για την περίοδο 01/01/2022–31/12/2024. Στην τελική βάση καταγράφονται 1.075.000 κεφαλίδες συναλλαγών στον πίνακα sales και 3.223.031 γραμμές προϊόντων στον πίνακα sale_items. Ο full_retail_dataset περιλαμβάνει 3.223.031 εγγραφές που αφορούν 100 προϊόντα κατανεμημένα σε 5 κατηγορίες. Τα δεδομένα είναι συνθετικά και δεν αντιστοιχούν σε πραγματικές συναλλαγές ή πελάτες συγκεκριμένης επιχείρησης.

Η δημιουργία των δεδομένων πραγματοποιήθηκε με Python. Για την παραγωγή των πελατών χρησιμοποιήθηκε η βιβλιοθήκη Faker με ελληνικό locale (el_GR), η οποία παρήγαγε 10.000 εγγραφές με στοιχεία όπως όνομα, email, τηλέφωνο, ηλικία, φύλο και πόλη. Στη συνέχεια, οι τιμές φύλου διορθώθηκαν χειροκίνητα όπου δεν αντιστοιχούσαν στο όνομα. Για τις συναλλαγές, το πρόγραμμα επέλεγε τυχαία καταστήματα, πελάτες, προϊόντα και προωθητικές ενέργειες από τους αντίστοιχους πίνακες της βάσης, καθώς και ημερομηνίες εντός της περιόδου 2022-2024. Κάθε συναλλαγή περιλάμβανε από 1 έως 5 γραμμές προϊόντων, με ποσότητα 1 έως 5 τεμαχίων ανά γραμμή. Για κάθε τεμάχιο αφαιρούνταν από την τιμή μονάδας τυχαία χρηματική έκπτωση από 0 έως 5 €, και η συνολική αξία της συναλλαγής υπολογιζόταν από τις τιμές, τις ποσότητες και τις εκπτώσεις των προϊόντων.

Τα κριτήρια σχεδιασμού του dataset και η επιλογή των χαρακτηριστικών καθοδηγήθηκαν επίσης από την πρακτική εμπειρία μου στη λειτουργία οπωροπωλείου. Ελήφθησαν υπόψη η ποικιλία προϊόντων και κατηγοριών, καθώς και παράγοντες που μπορούν να σχετίζονται με τη ζήτηση, όπως η ημερομηνία, οι καιρικές συνθήκες, οι αργίες και οι προωθητικές ενέργειες. Η επιλογή αυτών των πεδίων αποσκοπούσε στην εξέταση διαφορετικών συνθηκών πώλησης και στην υποστήριξη της ανάλυσης ζήτησης και αποθέματος. Η εμπειρία από το οπωροπωλείο χρησιμοποιήθηκε ως πρακτική καθοδήγηση για τον σχεδιασμό και όχι ως πηγή πραγματικών συναλλαγών.

Τα βασικά πεδία του πίνακα είναι:

Πεδίο	Τύπος	Περιγραφή
sale_id	INT	Μοναδικό αναγνωριστικό συναλλαγής
sale_date	DATE	Ημερομηνία συναλλαγής
product_id	INT	Αναγνωριστικό προϊόντος
customer_id	INT	Αναγνωριστικό πελάτη
quantity	INT	Ποσότητα που αγοράστηκε
unit_price	DECIMAL	Τιμή μονάδας
total_amount	DECIMAL	Συνολική αξία συναλλαγής
product_category	VARCHAR	Κατηγορία προϊόντος
customer_city	VARCHAR	Πόλη πελάτη
temperature	FLOAT	Θερμοκρασία (°C) κατά την ημέρα της συναλλαγής
rain_mm	FLOAT	Ποσότητα βροχόπτωσης (mm) κατά την ημέρα της συναλλαγής
promotion_type	VARCHAR(50)	Τύπος προωθητικής ενέργειας που ίσχυε
is_holiday	TINYINT(1)	Δυαδική μεταβλητή που υποδηλώνει αν η ημέρα ήταν αργία (1=ναι, 0=όχι)

4.2.2 Σχήμα retail_db_normalized

Η βάση retail_db_normalized οργανώνεται σε τρεις λογικές ομάδες πινάκων:

- α) κανονικοποιημένους συναλλακτικούς πίνακες (customers, products, sales, sale_items, promotions, weather_daily, holidays)
- β) πίνακες ML pipeline (ml_features_ready, product_model_scores, product_best_model, product_forecasts, product_anomalies)
- γ) πίνακες αποθέματος (product_stock, product_inventory_settings)

4.3 Ανάλυση περιπτώσεων χρήσης

Η ανάλυση περιπτώσεων χρήσης αποτυπώνει τις αλληλεπιδράσεις του χρήστη με το σύστημα και αποτελεί βασικό εργαλείο σχεδιασμού λογισμικού (Jacobson, Booch & Rumbaugh, 1999). Ο τελικός χρήστης του συστήματος είναι ο διαχειριστής καταστήματος, ο οποίος αλληλεπιδρά αποκλειστικά μέσω dashboard.

4.3.1 ΠΧ Σύνδεση χρήστη

Αναγνωριστικό: ΣΧ1

Περιγραφή: Ο χρήστης εισάγει τα διαπιστευτήριά του για πρόσβαση στο σύστημα

Βασική ροή:

1. Ο χρήστης ανοίγει την εφαρμογή στο <http://localhost:8502>
2. Το σύστημα εμφανίζει φόρμα εισαγωγής ονόματος χρήστη και κωδικού
3. Ο χρήστης συμπληρώνει τα διαπιστευτήριά του
4. Το σύστημα υπολογίζει το SHA-256 hash του κωδικού και το συγκρίνει με το αποθηκευμένο
5. Εφόσον η επαλήθευση είναι επιτυχής, ο χρήστης αποκτά πρόσβαση στο dashboard.

Εναλλακτική ροή: Εφόσον τα διαπιστευτήρια είναι λανθασμένα, το σύστημα εμφανίζει μήνυμα σφάλματος και δεν επιτρέπει την πρόσβαση.

4.3.2 ΠΧ Προβολή προβλέψεων ζήτησης

Αναγνωριστικό: ΠΠΖ1

Περιγραφή: Ο χρήστης επιλέγει προϊόν και προβάλλει την 30ήμερη πρόβλεψη ζήτησης

Βασική ροή:

1. Ο χρήστης επιλέγει την καρτέλα Forecast
2. Επιλέγει το προϊόν από τη λίστα
3. Το σύστημα ανακτά από τη βάση τη πρόβλεψη του βέλτιστου μοντέλου για το συγκεκριμένο προϊόν
4. Εμφανίζεται γραφική αναπαράσταση της πρόβλεψης για τις επόμενες 30 ημέρες
5. Εμφανίζεται πίνακας σύγκρισης όλων των μοντέλων με τις μετρικές MAE, RMSE, MAPE

4.3.3 ΠΧ Διαχείριση αποθέματος

Αναγνωριστικό: ΔΑ1

Περιγραφή: Ο χρήστης παρακολουθεί το απόθεμα και λαμβάνει προτάσεις αναπαραγγελίας.

Βασική ροή:

1. Ο χρήστης επιλέγει την καρτέλα Stock and Orders
2. Το σύστημα εμφανίζει watchlist με όλα τα προϊόντα που χρειάζονται αναπαραγγελία
3. Για κάθε προϊόν εμφανίζεται: τρέχον απόθεμα, ημέρες κάλυψης, προτεινόμενη ποσότητα αναπαραγγελίας και επίπεδο κινδύνου
4. Ο χρήστης μπορεί να τροποποιήσει τις παραμέτρους αναπαραγγελίας (lead time, safety days, MOQ)
5. Ο χρήστης καταγράφει νέα παράδοση ή ενημερώνει χειροκίνητα το απόθεμα

4.3.4 ΠΧ Ανίχνευση ανωμαλιών

Αναγνωριστικό: ΑΑ1

Περιγραφή: Ο χρήστης προβάλλει τις ανωμαλίες που εντοπίστηκαν στα δεδομένα πωλήσεων

Βασική ροή:

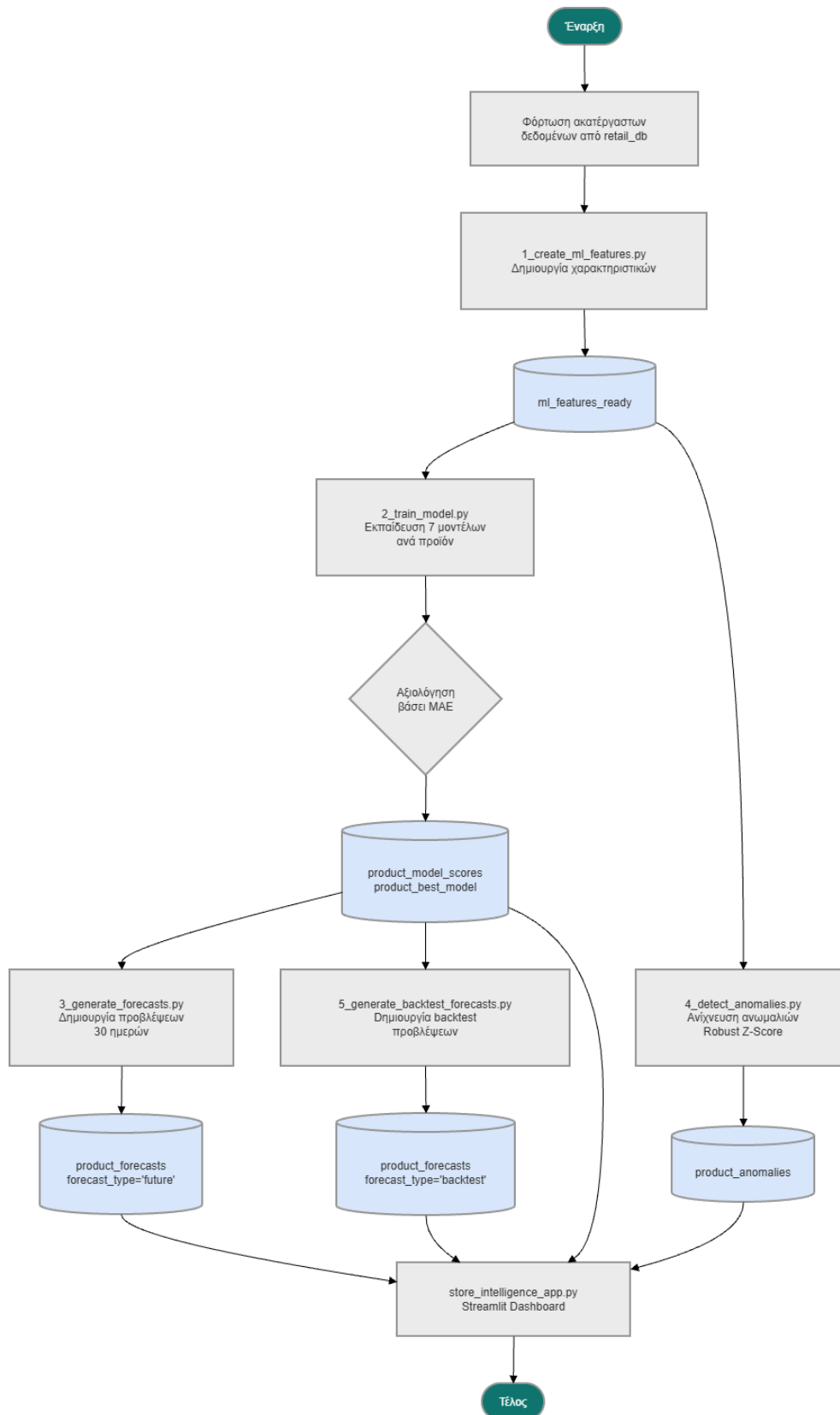
1. Ο χρήστης επιλέγει την καρτέλα Anomalies
2. Επιλέγει προϊόν
3. Εμφανίζεται πίνακας με τις ανιχνευμένες ανωμαλίες σε σελιδοποιημένη μορφή (20 εγγραφές ανά σελίδα): ημερομηνία, τύπος, βαθμός σοβαρότητας και score.

4.4 UML Διαγράμματα

Για την τεκμηρίωση της αρχιτεκτονικής και της λειτουργίας του συστήματος χρησιμοποιούνται τέσσερα διαγράμματα UML (Unified Modeling Language): Διάγραμμα Δραστηριοτήτων, Διάγραμμα Οντοτήτων-Συσχετίσεων, Διάγραμμα Ακολουθίας και Διάγραμμα Περιπτώσεων Χρήσης.

Διάγραμμα Δραστηριοτήτων

Το Διάγραμμα Δραστηριοτήτων (Activity Diagram) απεικονίζει τη ροή εκτέλεσης του συνολικού pipeline του συστήματος, από τη φόρτωση ακατέργαστων δεδομένων έως την εμφάνιση των αποτελεσμάτων στο dashboard.

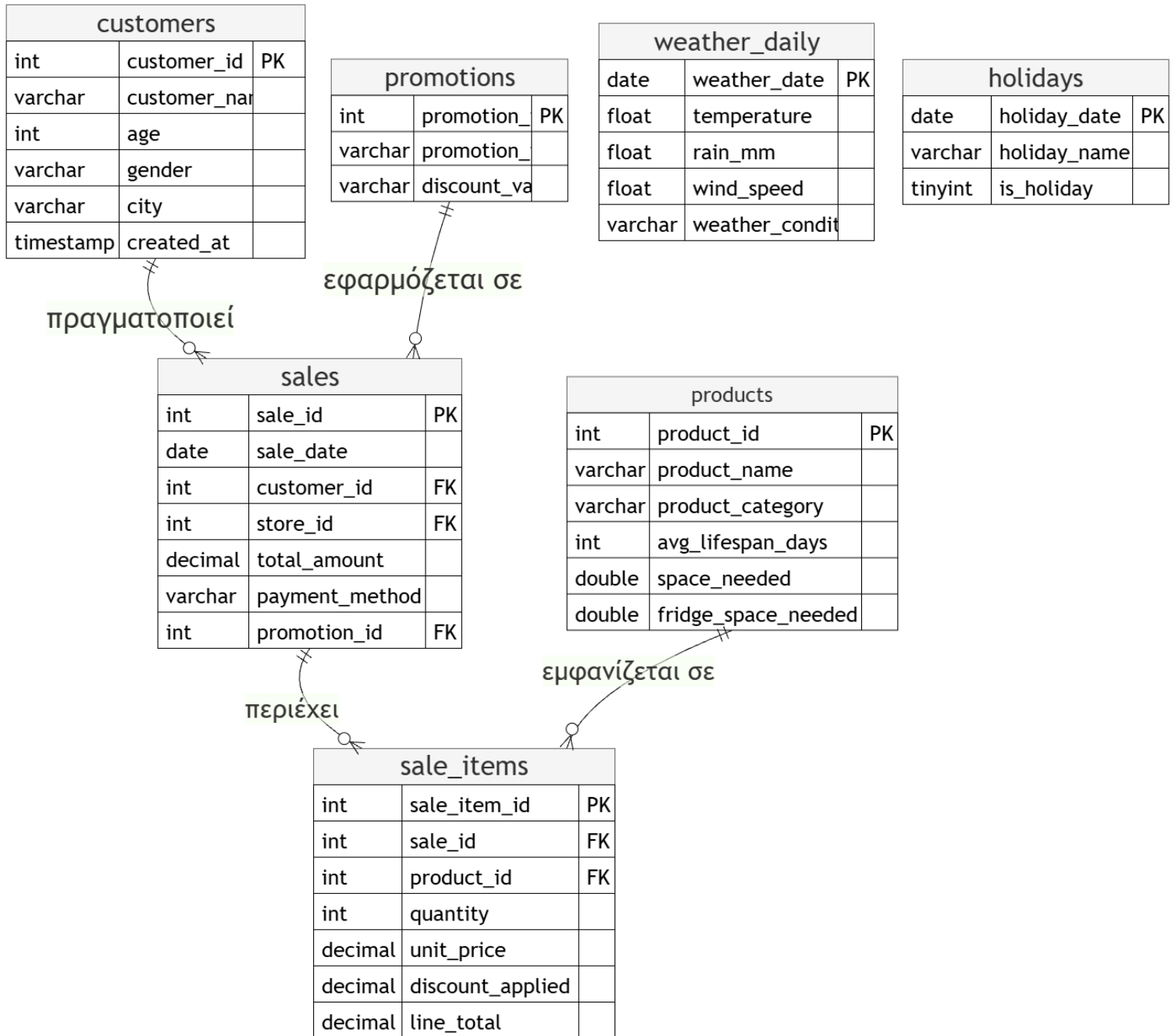


Σχήμα 4.1: Διάγραμμα Δραστηριοτήτων - Ροή εκτέλεσης ML Pipeline

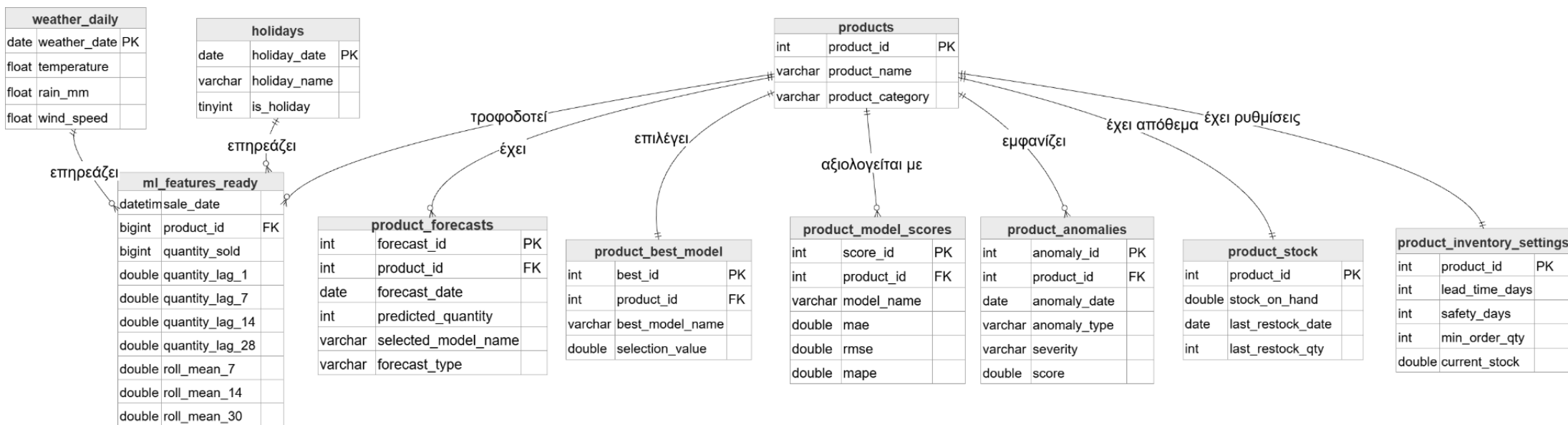
Η ροή ξεκινάει με τη φόρτωση των ακατέργαστων δεδομένων από τη βάση `retail_db`. Το script `1_create_ml_features.py` δημιουργεί τον πίνακα χαρακτηριστικών `ml_features_ready`, ο οποίος τροφοδοτεί παράλληλα δύο κλάδους: την εκπαίδευση μοντέλων (`2_train_model.py`) και την ανίχνευση ανωμαλιών (`4_detect_anomalies.py`). Μετά την εκπαίδευση και αξιολόγηση των οκτώ μοντέλων βάσει MAE, παράγονται προβλέψεις μέλλοντος (`3_generate_forecasts.py`) και backtest προβλέψεις (`5_generate_backtest_forecasts.py`). Όλα τα αποτελέσματα συγκεντρώνονται στο Streamlit dashboard.

Διάγραμμα Οντοτήτων Συσχετίσεων

Το Διάγραμμα Οντοτήτων-Συσχετίσεων (Entity-Relationship Diagram) απεικονίζει τη λογική δομή της βάσης δεδομένων `retail_db_normalized`, η οποία αποτελεί τον πυρήνα αποθήκευσης για τα επεξεργασμένα δεδομένα, τα χαρακτηριστικά ML, τις προβλέψεις και τις ανωμαλίες του συστήματος.



Σχήμα 4.2α: ER Diagram — Συναλλακτικό Επίπεδο (retail_db_normalized)



Σχήμα 4.2β: ER Diagram — ML Pipeline (*retail_db_normalized*)

Κεντρικές οντότητες συναλλαγών: Ο πίνακας customers αποθηκεύει τα δημογραφικά στοιχεία των πελατών. Ο πίνακας products περιέχει κάθε προϊόν με κατηγορία, μέση διάρκεια ζωής και απαιτήσεις χώρου αποθήκευσης. Ο πίνακας sales καταγράφει τις συναλλαγές, ενώ ο sale_items αποθηκεύει τις επιμέρους γραμμές ανά πώληση. Ο πίνακας promotions ορίζει τον τύπο και την τιμή κάθε προσφοράς.

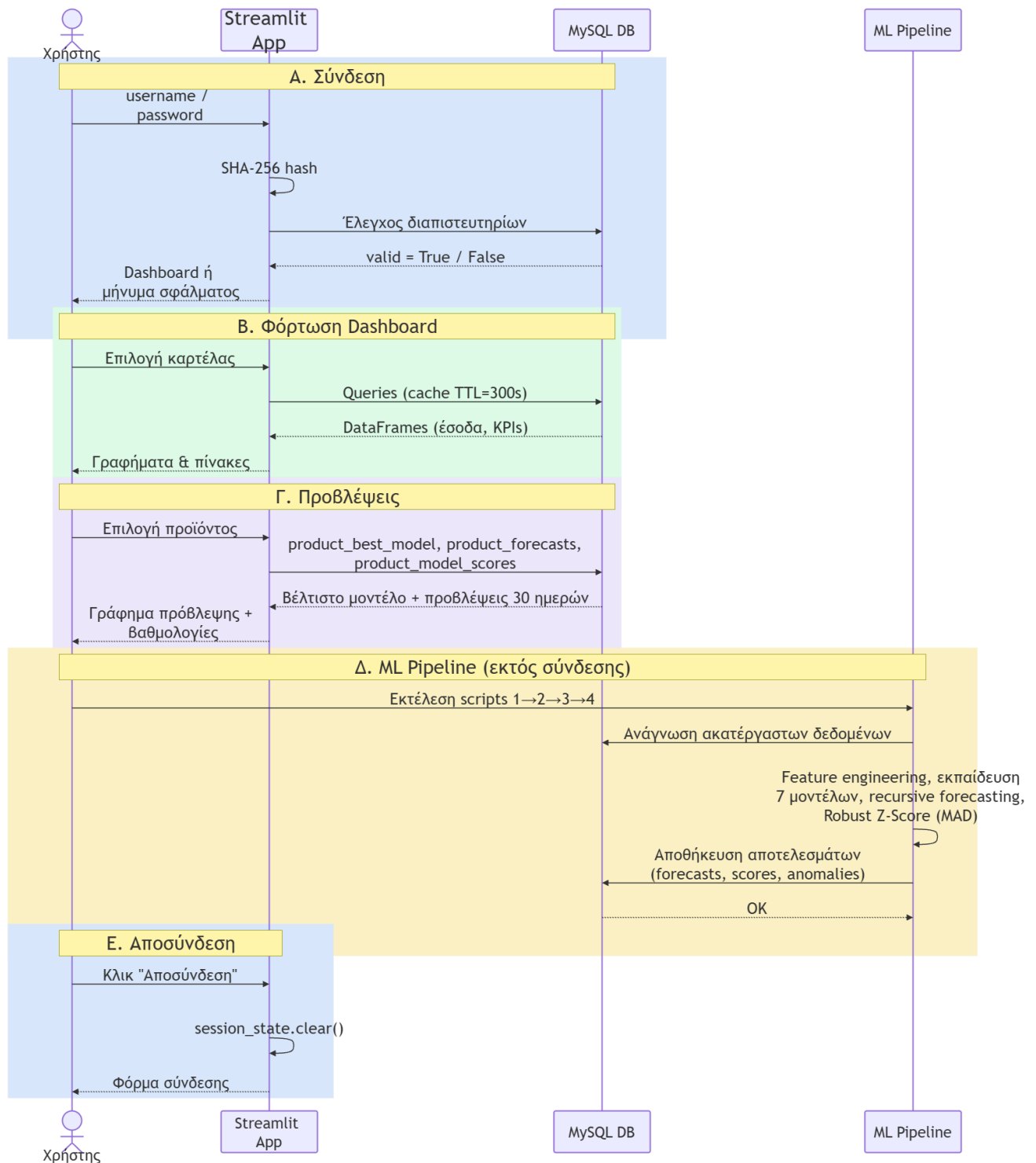
Περιβαλλοντικές οντότητες: Ο πίνακας weather_daily αποθηκεύει ημερήσιες μετεωρολογικές συνθήκες και ο holidays τηρεί ημερολόγιο αργιών. Και οι δύο τροφοδοτούν το pipeline ως εξωγενείς μεταβλητές.

Οντότητες ML Pipeline: Ο πίνακας ml_features_ready αποθηκεύει τον επεξεργασμένο πίνακα χαρακτηριστικών (lag features, rolling windows, μετεωρολογικά, αργίες). Ο product_forecasts περιέχει τις προβλέψεις ανά τύπο (future/backtest). Οι πίνακες product_best_model και product_model_scores καταγράφουν το βέλτιστο μοντέλο και τις βαθμολογίες MAE/RMSE/MAPE. Ο product_anomalies αποθηκεύει τις ανιχνευθείσες ανωμαλίες, ενώ οι product_stock και product_inventory_settings διαχειρίζονται το απόθεμα.

Κάθε πώληση συνδέεται με έναν πελάτη και μία προαιρετική προσφορά (πολλά-προς-ένα). Κάθε γραμμή πώλησης αναφέρεται σε ένα προϊόν. Ο πίνακας ml_features_ready παράγεται από τα δεδομένα πωλήσεων εμπλουτισμένα με μετεωρολογικά και εορτολόγιο, και τροφοδοτεί την εκπαίδευση που αποθηκεύει αποτελέσματα στον product_forecasts.

Διάγραμμα Ακολουθίας

Το Διάγραμμα Ακολουθίας (Sequence Diagram) απεικονίζει τη χρονολογική αλληλεπίδραση μεταξύ των συμμετεχόντων του συστήματος: του Χρήστη, της Streamlit App, της MySQL DB και του ML Pipeline.



Σχήμα 4.3: Διάγραμμα Ακολουθίας - Store Intelligence System

A. Σύνδεση χρήστη: Ο χρήστης εισάγει τα διαπιστευτήριά του. Η εφαρμογή υπολογίζει SHA-256 hash του κωδικού, τον ελέγχει έναντι του secrets.toml και ενημερώνει το `st.session_state["authenticated"]`.

B. Φόρτωση δεδομένων dashboard: Κάθε επιλογή καρτέλας ενεργοποιεί queries στη βάση δεδομένων. Τα αποτελέσματα αποθηκεύονται προσωρινά (`st.cache_data(ttl=300)`) για 5 λεπτά, αποφεύγοντας επαναλαμβανόμενα queries.

Γ. Καρτέλα προβλέψεων: Εκτελούνται τρία διαδοχικά queries — ανάκτηση βέλτιστου μοντέλου, προβλέψεων και βαθμολογιών — και εμφανίζονται γράφημα και πίνακας σύγκρισης μοντέλων.

Δ. Pipeline εκπαίδευσης: Τα scripts εκτελούνται διαδοχικά. Το `1_create_ml_features.py` δημιουργεί χαρακτηριστικά, το `2_train_model.py` εκπαιδεύει 8 μοντέλα και αποθηκεύει το βέλτιστο ανά προϊόν στη βάση δεδομένων, το `3_generate_forecasts.py` εφαρμόζει recursive forecasting 30 ημερών.

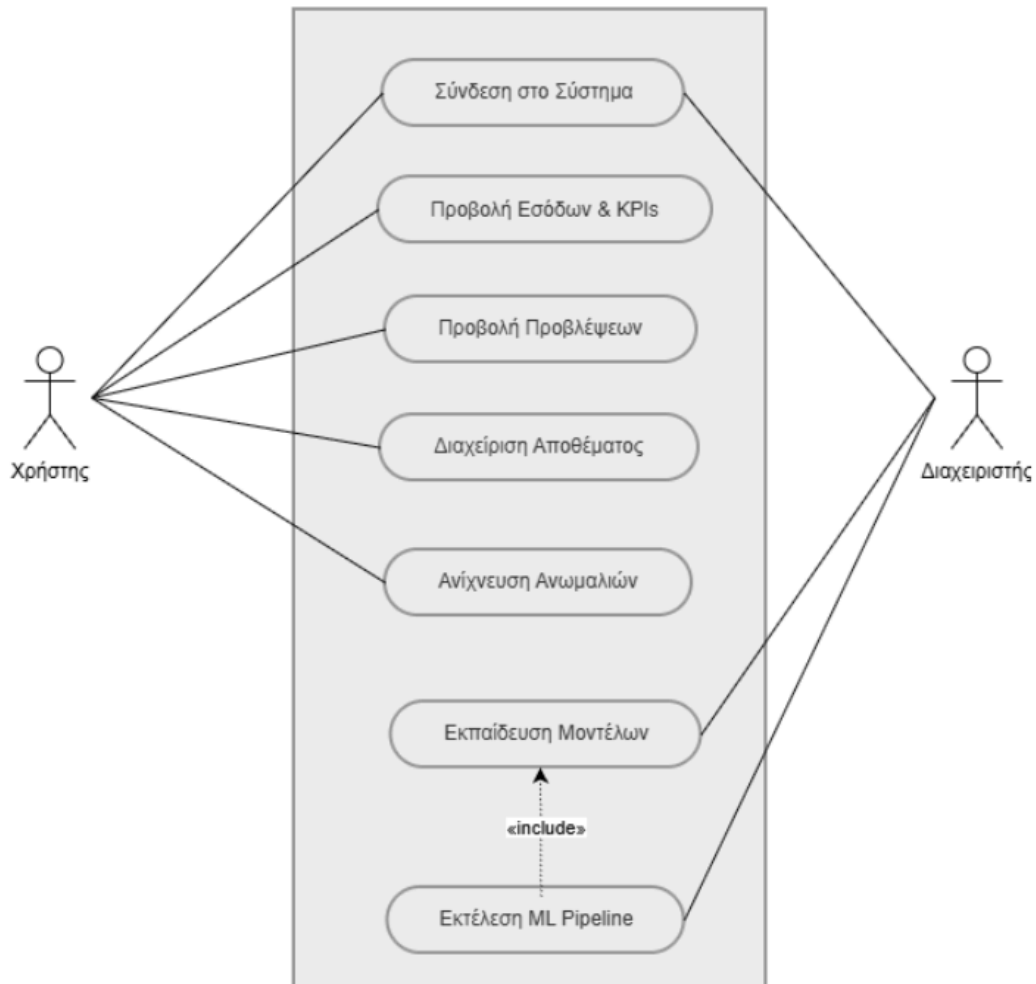
E. Ανίχνευση ανωμαλιών: Το `4_detect_anomalies.py` εφαρμόζει Robust Z-Score με MAD (`window=28`, `threshold=3.0`) και αποθηκεύει τις ανωμαλίες στον πίνακα `product_anomalies`.

ΣΤ. Αποσύνδεση: Κλικ στο κουμπί αποσύνδεσης εκκαθαρίζει το `st.session_state` και επιστρέφει τη φόρμα σύνδεσης.

Διάγραμμα Περιπτώσεων Χρήσης

Το Διάγραμμα Περιπτώσεων Χρήσης (Use Case Diagram) ορίζει τις λειτουργικές αλληλεπιδράσεις μεταξύ των δύο ρόλων χρηστών και του συστήματος.

Store Intelligence System



Σχήμα 4.4: Διάγραμμα Περιπτώσεων Χρήσης - Store Intelligence System

Ο Χρήστης (αριστερά) έχει πρόσβαση σε πέντε περιπτώσεις χρήσης: σύνδεση στο σύστημα, προβολή εσόδων & KPIs, προβολή προβλέψεων, διαχείριση αποθέματος και ανίχνευση ανωμαλιών. Ο Διαχειριστής (δεξιά) συμμετέχει στη σύνδεση και έχει αποκλειστική πρόσβαση στην εκτέλεση του ML Pipeline, η οποία με σχέση «include» ενεργοποιεί αυτόματα την περίπτωση χρήσης εκπαίδευσης και αξιολόγησης μοντέλων.

5. Υλοποίηση

Στο παρόν κεφάλαιο περιγράφεται αναλυτικά η υλοποίηση του συστήματος. Το κεφάλαιο οργανώνεται σε δύο μέρη: στο πρώτο παρουσιάζεται το ML pipeline, από την επεξεργασία των δεδομένων και την εκπαίδευση των μοντέλων έως την παραγωγή προβλέψεων και την ανίχνευση ανωμαλιών. Στο δεύτερο παρουσιάζεται η υλοποίηση του backend και του dashboard, με ανάλυση κάθε καρτέλας εφαρμογής.

5.1 Pipeline επεξεργασίας δεδομένων και μηχανολογία χαρακτηριστικών

Περιγραφή και προέλευση δεδομένων

Τα δεδομένα είναι συνθετικά και δημιουργήθηκαν με Python, όπως περιγράφεται στην ενότητα 4.2.1. Η τελική βάση περιλαμβάνει 1.075.000 συναλλαγές και 3.223.031 γραμμές προϊόντων, για 100 προϊόντα και 5 κατηγορίες κατά την περίοδο 2022-2024. Για τις ανάγκες του ML pipeline, οι συναλλαγές συγκεντρώνονται σε ημερήσια δεδομένα ανά προϊόν.

Βήματα επεξεργασίας

Η επεξεργασία των δεδομένων υλοποιείται στο `1_create_ml_features.py` και ακολουθεί επτά διαδοχικά βήματα:

Βήμα 1: Συγκέντρωση δεδομένων πωλήσεων

Τα ακατέργαστα δεδομένα συναλλαγών αθροίζονται σε ημερήσιο επίπεδο ανά προϊόν. Για κάθε συνδυασμό ημέρας - προϊόντος υπολογίζονται: συνολική ποσότητα (`quantity_sold`), συνολικά έσοδα (`revenue`) και αριθμός συναλλαγών (`transaction_count`).

Βήμα 2: Φόρτωση διαστατικών δεδομένων

Φορτώνονται τέσσερις πρόσθετες πηγές δεδομένων που θα χρησιμοποιηθούν ως χαρακτηριστικά:

- Πληροφορίες προϊόντων: Κατηγορία, μέση διάρκεια ζωής(`avg_lifespan_days`), χώρος αποθήκευσης.
- Καιρικά δεδομένα: θερμοκρασία, βροχόπτωση(`mm`), ταχύτητα ανέμου.
- Αργίες: δυαδική μεταβλητή `is_holiday`

- Προωθητικές ενέργειες: τύπος, ποσοστό έκπτωσης, δυαδική μεταβλητή `is_promo`

Βήμα 3: Δημιουργία πλέγματος ημέρας - προϊόντος

Δημιουργείται πλήρες πλέγμα (cross join) όλων των ημερομηνιών με όλα τα προϊόντα, ώστε να μην υπάρχουν κενές ημέρες στο dataset. Ημέρες χωρίς πωλήσεις για ένα προϊόν λαμβάνουν τιμή 0.

Βήμα 4: Ένωση δεδομένων

Όλες οι παραπάνω πηγές ενώνονται στο πλέγμα. Ο τύπος προώθησης μετατρέπεται σε δυαδικές μεταβλητές (0/1) ανά κατηγορία, ώστε να είναι κατανοητός από τα μοντέλα μηχανικής μάθησης.

Βήμα 5: Χρονικά χαρακτηριστικά (Temporal features)

Από την ημερομηνία κάθε εγγραφής εξάγονται: `year`, `month`, `day`, `day_of_week`, `week_of_year`, `is_weekend`, `day_of_month`.

Βήμα 6: Lag features

Για κάθε προϊόν δημιουργούνται χαρακτηριστικά που αντικατοπτρίζουν τις πωλήσεις προηγούμενων ημερών. Οι τιμές lag ορίζονται στο αρχείο `config.yaml` (πχ. `lag 1, 7, 14, 28` ημέρες), δημιουργώντας στήλες `quantity_lag_1`, `quantity_lag_7` κλπ.

Βήμα 7: Rolling window features

Για κάθε παράθυρο που ορίζεται στο `config.yaml` (πχ 7, 14, 28 ημέρες) υπολογίζονται τέσσερα στατιστικά κυλιόμενου παραθύρου: μέσος όρος (`roll_mean`), τυπική απόκλιση (`roll_std`), ελάχιστο (`roll_min`) και μέγιστο (`roll_max`). Όλα τα rolling features υπολογίζονται με χρονική μετατόπιση μιας ημέρας (`shift=1`), αποτρέποντας τη διαρροή πληροφορίας από το μέλλον (data leakage) κατά την εκπαίδευση των μοντέλων.

Το τελικό dataset αποθηκεύεται στον πίνακα `ml_features_ready` της βάσης `retail_db_normalized`.

5.2 Εκπαίδευση μοντέλων και επιλογή βέλτιστου

Η εκπαίδευση των μοντέλων υλοποιείται στο script `2_train_model.py`. Για κάθε προϊόν εκπαιδεύονται και αξιολογούνται ισότιμα και τα οκτώ μοντέλα που περιγράφηκαν στο κεφάλαιο 3.3, με αυτόματη επιλογή βέλτιστου βάσει της μετρικής MAE.

5.2.1 Διαχωρισμός δεδομένων εκπαίδευσης και ελέγχου

Για κάθε προϊόν εφαρμόζεται χρονικός διαχωρισμός 80/20: το 80% των μοναδικών ημερομηνιών αποτελεί το σύνολο εκπαίδευσης (train set) και το υπόλοιπο 20% το σύνολο ελέγχου (test set). Ο διαχωρισμός γίνεται αυστηρά χρονολογικά, τα δεδομένα ελέγχου είναι πάντα μεταγενέστερα των δεδομένων εκπαίδευσης, αποτρέποντας οποιαδήποτε διαρροή πληροφορίας από το μέλλον. Προϊόντα με λιγότερες από 60 ημέρες ιστορικού παραλείπονται.

5.2.2 Στρατηγική αναδρομικής πρόβλεψης (Recursive Forecasting)

Για τα τρία ML μοντέλα (LightGBM, Random Forest, Extra Trees) χρησιμοποιείται στρατηγική αναδρομικής πρόβλεψης. Το μοντέλο εκπαιδεύεται να προβλέπει την επόμενη ημέρα και στη συνέχεια εφαρμόζεται αναδρομικά για 30 ημέρες: κάθε πρόβλεψη χρησιμοποιείται ως είσοδος για την επόμενη. Τα lag και rolling features ενημερώνονται δυναμικά σε κάθε βήμα βάσει του ιστορικού (Bontempi, Ben Taieb & Le Borgne, 2013).

Μετρικές αξιολόγησης

Κάθε μοντέλο αξιολογείται στο σύνολο ελέγχου με τρεις μετρικές:

1. MAE (Mean Absolute Error): μέσο απόλυτο σφάλμα - ερμηνεύσιμο στις ίδιες μονάδες με τις πωλήσεις
2. RMSE (Root Mean Squared Error): τετραγωνική ρίζα του μέσου τετραγωνικού σφάλματος - πιο ευαίσθητο σε μεγάλα σφάλματα
3. MAPE (Mean Absolute Percentage Error): μέσο απόλυτο ποσοστιαίο σφάλμα - επιτρέπει σύγκριση μεταξύ προϊόντων διαφορετικής κλίμακας

Το MAE επιλέχθηκε ως κριτήριο επιλογής βέλτιστου μοντέλου διότι είναι πιο ανθεκτικό σε ακραίες τιμές σε σχέση με το RMSE, και πιο άμεσα ερμηνεύσιμο στο πλαίσιο της πρόβλεψης ζήτησης (Hyndman & Koehler, 2006)

Αυτόματη επιλογή βέλτιστου μοντέλου

Μετά την αξιολόγηση όλων των μοντέλων, το μοντέλο με τη χαμηλότερη τιμή MAE επιλέγεται αυτόματα ως βέλτιστο για το συγκεκριμένο προϊόν. Τα αποτελέσματα αποθηκεύονται σε δύο πίνακες:

- `product_model_scores`: όλες οι μετρικές για όλα τα μοντέλα ανα προϊόν
- `product_best_model`: το βέλτιστο μοντέλο ανά προϊόν με την αντίστοιχη τιμή MAE

5.3 Δημιουργία προβλέψεων

Η παραγωγή των μελλοντικών προβλέψεων υλοποιείται στο script `3_generate_forecasts.py`. Για κάθε προϊόν χρησιμοποιείται αποκλειστικά το βέλτιστο μοντέλο που επιλέχθηκε στο προηγούμενο βήμα και παράγεται πρόβλεψη για τις επόμενες 30 ημέρες από την τελευταία διαθέσιμη ημερομηνία του ιστορικού. Το μοντέλο εκπαιδεύεται εκ νέου σε ολόκληρο το ιστορικό, χωρίς διαχωρισμό `train/test`, ώστε να αξιοποιεί τη μέγιστη διαθέσιμη πληροφορία. Τα αποτελέσματα αποθηκεύονται στον πίνακα `product_forecasts` με `forecast_type = 'future'`.

Για τα ML μοντέλα (LightGBM, Random Forest, Extra Trees) εφαρμόζεται η αναδρομική στρατηγική που περιγράφηκε στο 5.2: σε κάθε βήμα ενημερώνονται δυναμικά τα χρονικά χαρακτηριστικά, τα `lag features` και τα `rolling features` βάσει του ιστορικού που εμπλουτίζεται με τις προηγούμενες προβλέψεις. Τα υπόλοιπα πέντε μοντέλα (Moving Average, Holt-Winters, ARIMA, Naive, Seasonal Naive) παράγουν τον πλήρη ορίζοντα των 30 ημερών απευθείας, χωρίς αναδρομική επανατροφοδότηση.

Αναφορικά με τα εξωγενή χαρακτηριστικά (θερμοκρασία, βροχόπτωση, ένδειξη αργίας, ένδειξη προωθητικής ενέργειας), αυτά δεν ενημερώνονται κατά την αναδρομική πρόβλεψη. Διατηρούν την τιμή της τελευταίας γνωστής ημέρας του ιστορικού σε όλο τον ορίζοντα των 30 ημερών. Αυτό αποτελεί συνειδητή απλοποιητική παραδοχή: η απόκτηση

αξιόπιστων μετεωρολογικών προβλέψεων ή προγραμματισμένων προσφορών για το μέλλον δεν ήταν εφικτή στο πλαίσιο των συνθετικών δεδομένων. Χρονικά χαρακτηριστικά (ημέρα εβδομάδας, μήνας κλπ.) ενημερώνονται κανονικά. Η βελτίωση αυτού του σημείου με χρήση ιστορικών κλιματολογικών μέσων ανά ημερομηνία αποτελεί κατεύθυνση μελλοντικής επέκτασης.

Αξιοσημείωτο χαρακτηριστικό της υλοποίησης είναι η ενσωμάτωση του πεδίου `avg_lifespan_days` από τον πίνακα προϊόντων. Ο ορίζοντας πρόβλεψης περιορίζεται αυτόματα στη διάρκεια ζωής του προϊόντος όταν αυτή είναι μικρότερη από τις 30 ημέρες, αποφεύγοντας την υπερπρόβλεψη για ευπαθή ή εποχικά προϊόντα.

5.4 Ανίχνευση ανωμαλιών

Το script `4_detect_anomalies.py` υλοποιεί την αυτόματη ανίχνευση ανωμαλιών στη ζήτηση σε ημερήσιο επίπεδο για κάθε προϊόν. Ο αλγόριθμος βασίζεται σε Robust z-score με χρήση κινητού παραθύρου (rolling window), αξιοποιώντας τη διάμεσο και την απόλυτη διάμεσο απόκλισης (MAD - Median Absolute Deviation) αντί της τυπικής απόκλισης, καθιστώντας τον ανθεκτικό σε ακραίες τιμές (outliers).

Για κάθε ημέρα και κάθε προϊόν, ο αλγόριθμος εξετάζει τα προηγούμενα ημερήσια δεδομένα και υπολογίζει το robust z-score ως:

$$Z = \frac{0.6745(x - \bar{x})}{MAD}$$

όπου x είναι η τρέχουσα τιμή, $\bar{x}$ η διάμεσος του παραθύρου και $MAD = \text{median}(|x_i - \bar{x}|)$. Ο συντελεστής 0.6745 κλιμακώνει το MAD ώστε να είναι συγκρίσιμο με την τυπική απόκλιση υπό κανονική κατανομή. Σε περίπτωση που το MAD είναι σχεδόν μηδέν (ομοιόμορφες τιμές), ο αλγόριθμος χρησιμοποιεί εναλλακτικά την τυπική απόκλιση ως fallback.

Κάθε σημείο ταξινομείται ως εξής:

- Spike (ανοδική ανωμαλία): $z \geq z_{\text{threshold}}$
- Dip (καθοδική ανωμαλία): $z \leq -z_{\text{threshold}}$

Επιπλέον, κάθε ανωμαλία χαρακτηρίζεται από σημείο σοβαρότητας:

- High: $|z| \geq z_{\text{threshold}} + 1.5$
- Medium: $z_{\text{threshold}} \leq |z| < z_{\text{threshold}} + 1.5$

Στην παρούσα υλοποίηση επιλέχθηκε $z_{\text{threshold}} = 3.0$ και μέγεθος κινητού παραθύρου $\text{window} = 28$ ημέρες. Η τιμή 3.0 αποτελεί ευρέως αποδεκτό κριτήριο για στατιστική ανίχνευση ακραίων τιμών και εξασφαλίζει ισορροπία μεταξύ ευαισθησίας ανίχνευσης και αποφυγής ψευδώς θετικών αποτελεσμάτων. Συνεπώς, τα όρια σοβαρότητας ισούνται με: High για $|z| \geq 4.5$ και Medium για $3.0 \leq |z| < 4.5$.

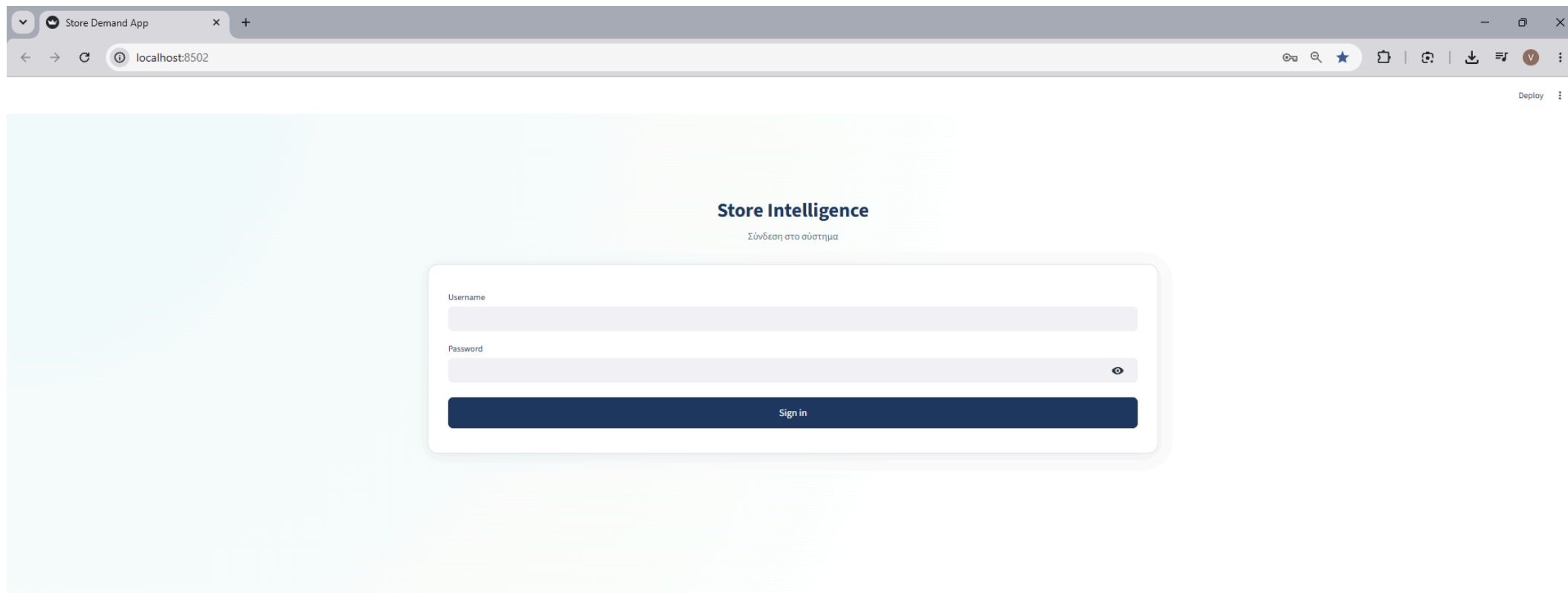
Τα αποτελέσματα αποθηκεύονται στον πίνακα `product_anomalies` της βάσης δεδομένων και περιλαμβάνουν για κάθε ανωμαλία: το προϊόν, την ημερομηνία, τον τύπο (spike/dip), το επίπεδο σοβαρότητας, το z-score, την πραγματική ποσότητα, τη baseline ποσότητα (διάμεσος παραθύρου) και το MAD.

5.5 Backend εφαρμογής

Η εφαρμογή ακολουθεί αρχιτεκτονική single-page application με αποκλειστικό σημείο εισόδου το αρχείο `store_intelligence_app.py`. Το backend περιλαμβάνει τρεις βασικές υπηρεσίες: έλεγχο πρόσβασης, caching δεδομένων και πλοήγηση καρτελών.

Έλεγχος πρόσβασης

Η εφαρμογή υλοποιεί σύστημα εισόδου βάσει ονόματος χρήστη και κωδικού πρόσβασης. Τα διαπιστευτήρια αποθηκεύονται στο αρχείο `secrets.toml` σε μορφή SHA-256 hash, ο κώδικας δεν αποθηκεύεται ποτέ σε plaintext. Κατά τη σύνδεση, ο κωδικός κατακερματίζεται και συγκρίνεται με τον αποθηκευμένο hash. Η κατάσταση σύνδεσης διατηρείται στο `st.session_state["authenticated"]` και εφόσον ο χρήστης δεν έχει εκτελέσει είσοδο, το σύστημα επιστρέφει αμέσως χωρίς να φορτώσει δεδομένα ή να αποδώσει περιεχόμενο.



Σχήμα 5.1: Οθόνη σύνδεσης χρήστη - Φόρμα εισαγωγής διαπιστευτηρίων

Caching δεδομένων

Για την αποφυγή επαναλαμβανόμενων ερωτημάτων SQL σε κάθε αλληλεπίδραση του χρήστη, χρησιμοποιείται ο decorator `st.cache_data` με χρόνο ζωής 300 δευτερολέπτων. Η συνάρτηση `load_data()` φορτώνει όλους τους απαιτούμενους πίνακες σε ένα πέρασμα: ιστορικά πωλήσεων, προβλέψεις, βέλτιστα μοντέλα, μετρικές, ρυθμίσεις αποθέματος και ανωμαλίες. Η ακύρωση της cache πραγματοποιείται αυτόματα μετά από ενέργειες εγγραφής, όπως η αποθήκευση παραμέτρων παραγγελίας ή η καταχώρηση παράδοσης αποθέματος.

Πλοήγηση καρτελών

Η πλοήγηση μεταξύ των έξι καρτελών υλοποιείται με custom κουμπιά αντί των native Streamlit tabs, επιτρέποντας πλήρη έλεγχο της κατάστασης. Η ενεργή καρτέλα αποθηκεύεται στο `st.session_state["active_tab"]` και αλλάζει με `st.rerun()`. Κάθε καρτέλα αποδίδεται σε ξεχωριστό module (π.χ. `revenue.py`, `forecast.py`) μέσω αντίστοιχης συνάρτησης `render()`, υλοποιώντας την αρχή της διαχωρισμένης ευθύνης.

5.6 Dashboard - Καρτέλες εφαρμογής

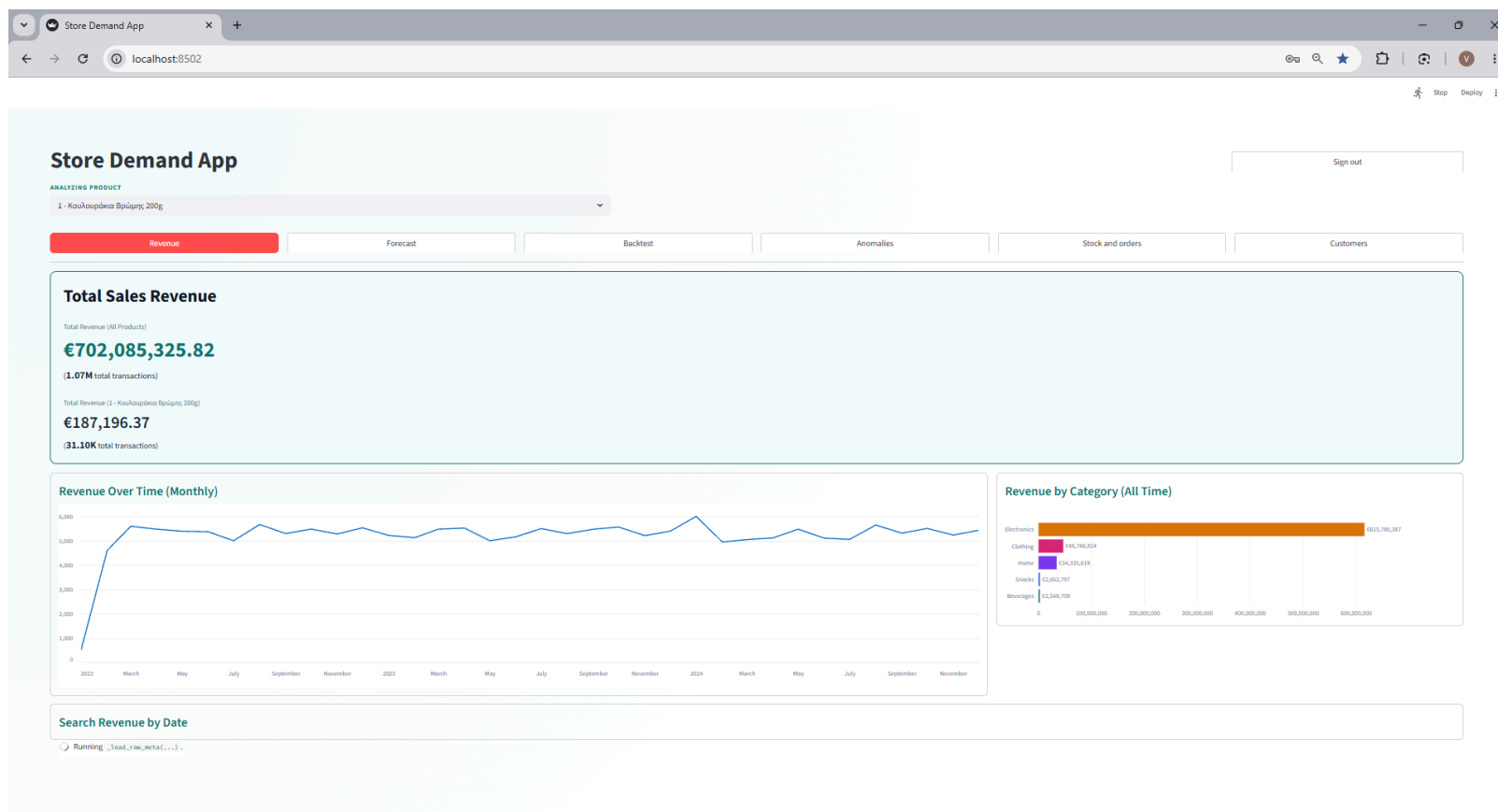
Το dashboard οργανώνεται σε έξι καρτέλες, καθεμία υλοποιημένη σε ξεχωριστό module. Ο χρήστης επιλέγει πρώτα το προϊόν που τον ενδιαφέρει μέσω dropdown και στη συνέχεια πλοηγείται μεταξύ καρτελών για να εξετάσει τις διαφορετικές πτυχές του.

5.6.1 Καρτέλα Revenue

Η καρτέλα Revenue παρέχει συνολική εικόνα των εσόδων και των πωλήσεων. Στο επάνω μέρος εμφανίζονται δύο KPIs cards: το συνολικό έσοδο όλων των προϊόντων και το έσοδο του επιλεγμένου προϊόντος, συνοδευόμενα από τον αντίστοιχο αριθμό συναλλαγών.

Ακολουθούν δυο γραφήματα: αριστερά, γραμμικό γράφημα εσόδων ανά μήνα για το επιλεγμένο προϊόν, και δεξιά, οριζόντιο ραβδόγραμμα με τα έσοδα ανα κατηγορία προϊόντων για το σύνολο των δεδομένων.

Η καρτέλα περιλαμβάνει επίσης εργαλείο αναζήτησης εσόδων με φίλτρα ημερομηνίας, κατηγορίας και προϊόντος. Τα αποτελέσματα υπολογίζονται απευθείας με SQL aggregation χωρίς φόρτωση ολόκληρου του πίνακα στη μνήμη, εξασφαλίζοντας απόδοση ακόμα και για μεγάλα σύνολα δεδομένων. Τέλος, εμφανίζεται ραβδόγραμμα με τα κορυφαία 10 προϊόντα βάσει εσόδων για την κατηγορία που επιλέγει ο χρήστης.

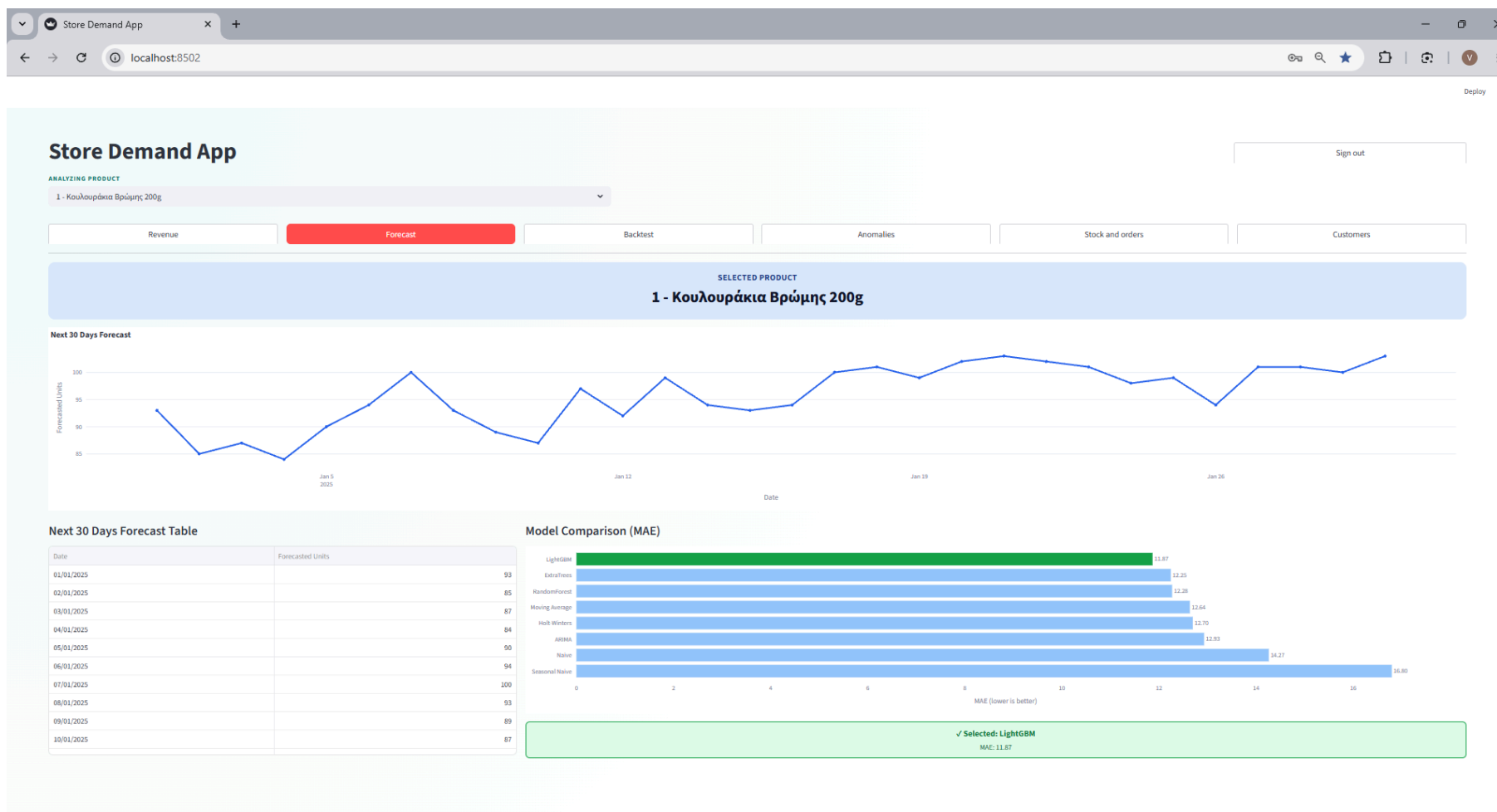


Σχήμα 5.2: Καρτέλα Εσόδων - Ανάλυση ημερησίων εσόδων & KPIs

5.6.2 Καρτέλα Forecast

Η καρτέλα Forecast εμφανίζει τις μελλοντικές προβλέψεις ζήτησης για το επιλεγμένο προϊόν. Στο επάνω μέρος εμφανίζεται γραμμικό γράφημα με τις προβλεπόμενες ποσότητες για τις επόμενες 30 ημέρες.

Στο κάτω μέρος η οθόνη, χωρίζεται σε δύο στήλες. Αριστερά εμφανίζεται πίνακας με τις ημερήσιες τιμές πρόβλεψης. Δεξιά εμφανίζεται οριζόντιο ραβδόγραμμα σύγκρισης όλων των μοντέλων βάσει MAE. Το βέλτιστο προϊόν επισημαίνεται με πράσινο χρώμα, ενώ τα υπόλοιπα εμφανίζονται με μπλε. Κάτω από το γράφημα εμφανίζεται πλαίσιο που αναδεικνύει το επιλεγμένο μοντέλο και την τιμή MAE του.



Σχήμα 5.3: Καρτέλα Προβλέψεων - Πρόβλεψη ζήτησης 30 ημερών

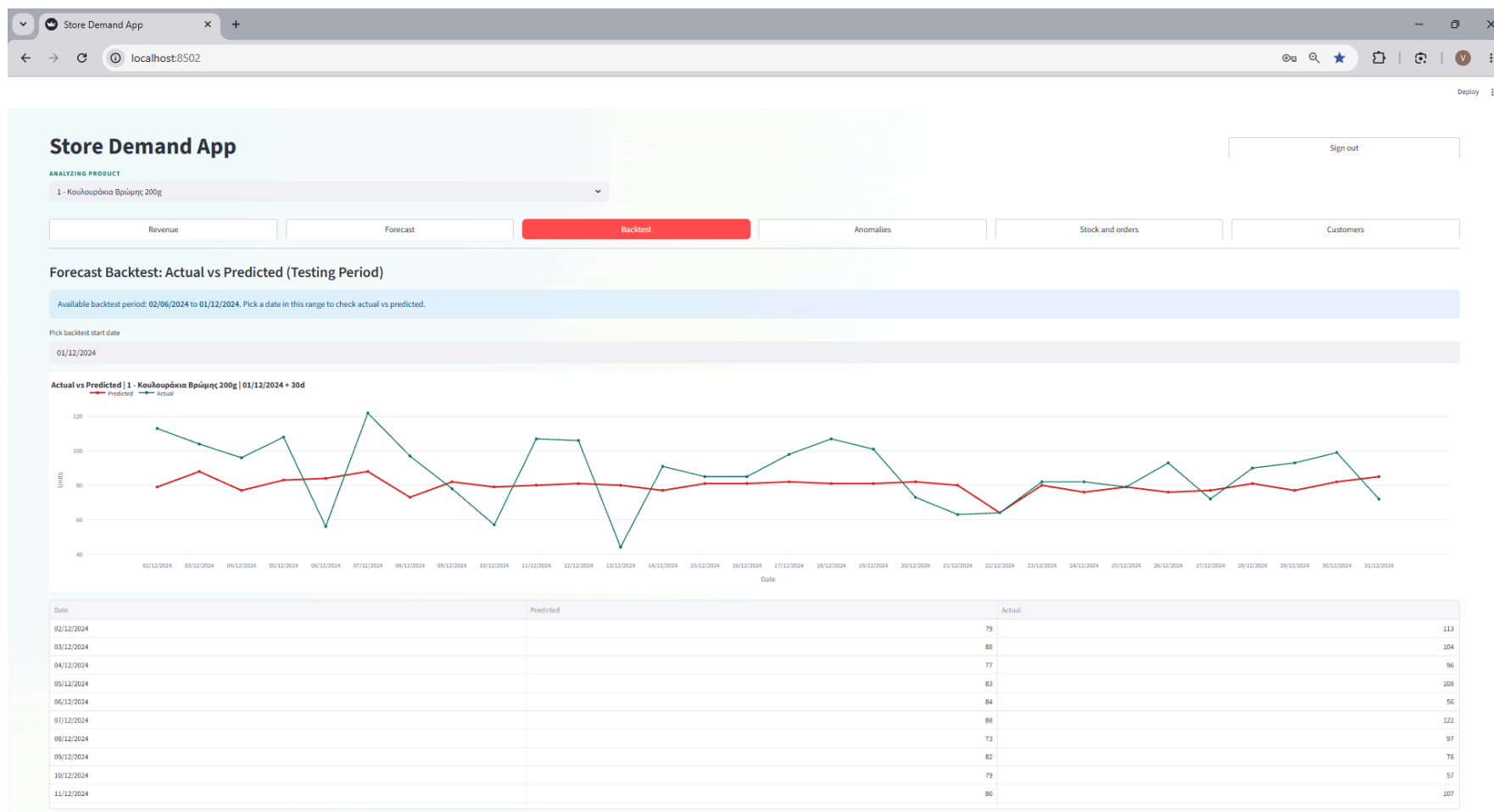
5.6.3 Καρτέλα Backtest

Η καρτέλα Backtest επιτρέπει την οπτική αξιολόγηση της ακρίβειας των προβλέψεων συγκρίνοντας τις προβλεπόμενες με τις πραγματικές τιμές σε ιστορική περίοδο.

Ο χρήστης επιλέγει μια ημερομηνία εκκίνησης μέσα στο διαθέσιμο εύρος backtest. Το σύστημα εντοπίζει αυτόματα την πλησιέστερη διαθέσιμη ημερομηνία για την οποία υπάρχουν αποθηκευμένες προβλέψεις (`forecast_type = 'backtest'`), και εμφανίζει γράφημα με δύο γραμμές: τις πραγματικές ποσότητες (πράσινο) και τις προβλεπόμενες (κόκκινο) για τις επόμενες 30 ημέρες από την επιλεγμένη ημερομηνία. Κάτω από το γράφημα εμφανίζεται ο αντίστοιχος πίνακας τιμών.

Οι προβλέψεις αυτές παράγονται με walk-forward (rolling) μεθοδολογία: για κάθε εβδομαδιαία ημερομηνία εκτέλεσης στο διάστημα Ιουνίου–Δεκεμβρίου 2024, το βέλτιστο μοντέλο του εκάστοτε προϊόντος επανεκπαιδεύεται αποκλειστικά με δεδομένα που προηγούνται εκείνης της ημερομηνίας, και στη συνέχεια προβλέπει τις επόμενες 30 ημέρες. Με τον τρόπο αυτό εξασφαλίζεται ότι οι αποθηκευμένες προβλέψεις backtest είναι γνήσια out-of-sample και δεν έχουν «δει» τα δεδομένα που αξιολογούνται.

Η λειτουργία αυτή επιτρέπει στον χρήστη να αξιολογεί εμπειρικά την ποιότητα του μοντέλου σε πραγματικές συνθήκες, χωρίς να απαιτείται τεχνική γνώση μετρικών.



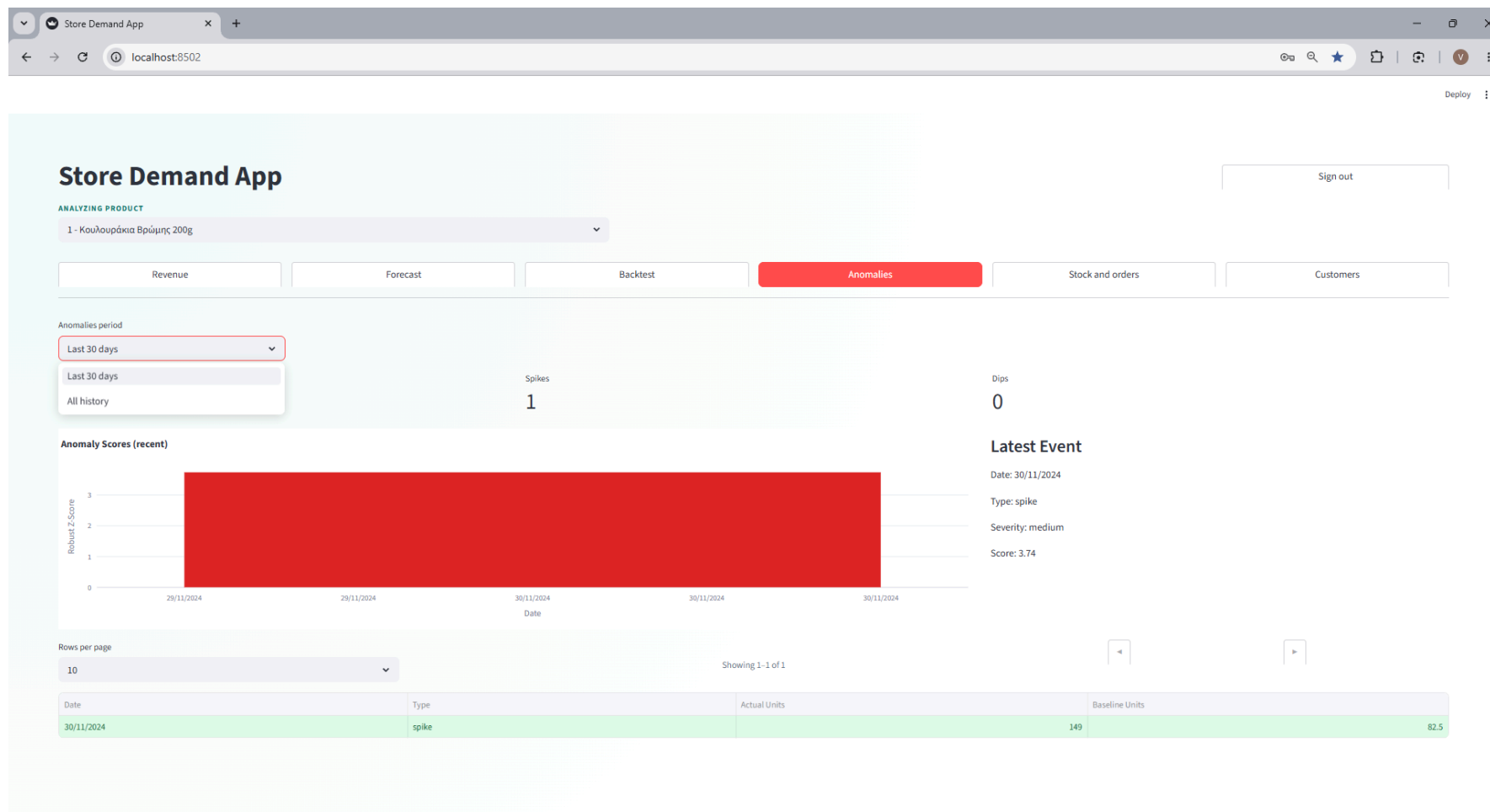
Σχήμα 5.4: Καρτέλα Backtest - Αξιολόγηση μοντέλων σε ιστορικά δεδομένα

5.6.4 Καρτέλα Anomalies

Η καρτέλα Anomalies εμφανίζει τις ανωμαλίες της ζήτησης που εντοπίστηκαν για το επιλεγμένο προϊόν. Ο χρήστης μπορεί να επιλέξει μεταξύ δύο χρονικών περιόδων: τελευταίες 30 ημέρες ή ολόκληρο το ιστορικό.

Στο επάνω μέρος εμφανίζονται τρία KPI metrics: συνολικός αριθμός ανωμαλιών, αριθμός spikes και αριθμός dips. Ακολουθεί γράφημα z-score για τις πιο πρόσφατες ανωμαλίες, και δίπλα του πλαίσιο με το τελευταίο γεγονός (ημερομηνία, τύπος, σοβαρότητα, score).

Στο κάτω μέρος εμφανίζεται πίνακας με όλες τις ανωμαλίες της επιλεγμένης περιόδου, ταξινομημένες από την πιο πρόσφατη. Κάθε γραμμή χρωματίζεται ανάλογα τον τύπο ανωμαλίας: πράσινο για spike, κόκκινο για dip. Ο πίνακας υποστηρίζει σελιδοποίηση για εύκολη πλοήγηση σε μεγάλο αριθμό εγγραφών.



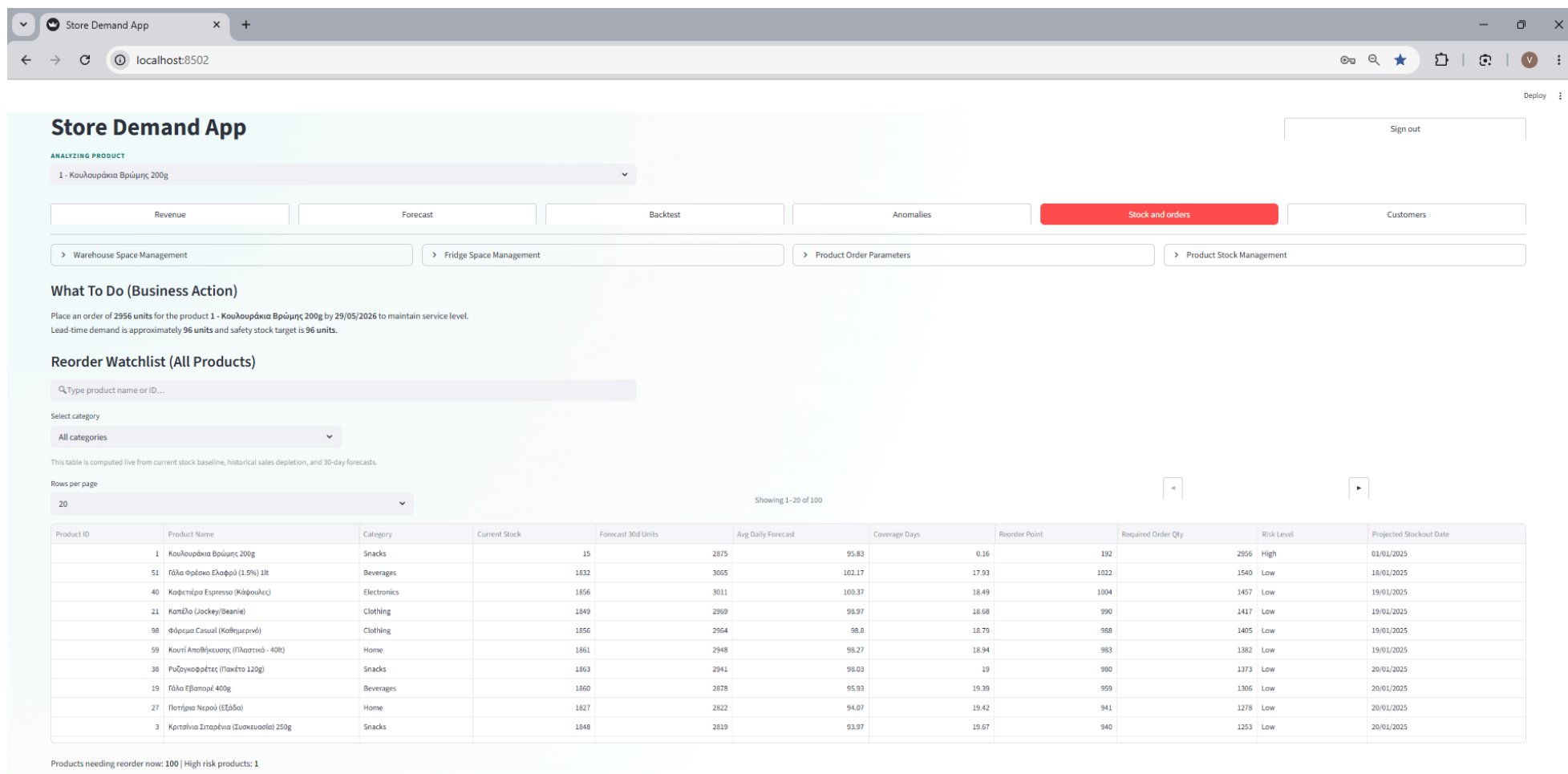
Σχήμα 5.5: Καρτέλα Ανωμαλιών - Χάρτης ανιχνευθεισών ανωμαλιών

5.6.5 Καρτέλα Stock and Orders

Η καρτέλα Stock and Orders είναι η πιο σύνθετη καρτέλα και αποτελεί το κεντρικό εργαλείο διαχείρισης αποθέματος.

Στο επάνω μέρος εμφανίζονται τέσσερα αναπτυσσόμενα πλαίσια: Warehouse Space Management και Fridge Space Management, όπου ο χρήστης ορίζει τη συνολική χωρητικότητα αποθήκης και ψυγείου και βλέπει πόσο χώρο καταλαμβάνει το τρέχον απόθεμα, με προειδοποίηση αν ξεπεραστεί η χωρητικότητα. Product Order Parameters, όπου ρυθμίζονται ο χρόνος παράδοσης (lead time), οι ημέρες ασφάλειας (safety days) και η ελάχιστη ποσότητα παραγγελίας (MOQ). Product Stock Management, όπου εμφανίζεται το τρέχον απόθεμα και ο χρήστης μπορεί να το ορίσει χειροκίνητα ή να καταχωρήσει νέα παράδοση.

Ακολουθεί το κύριο τμήμα: σύσταση επιχειρησιακής ενέργειας για το επιλεγμένο προϊόν. Εάν απαιτείται παραγγελία, εμφανίζεται η προτεινόμενη ποσότητα και η καταληκτική ημερομηνία υποβολής. Τέλος, εμφανίζεται Reorder Watchlist με όλα τα προϊόντα, φιλτραρισμένα ανά κατηγορία ή με αναζήτηση κειμένου, που δείχνει ποιά προϊόντα χρειάζονται άμεσα παραγγελία και ποια βρίσκονται σε υψηλό κίνδυνο αποθεματικής έλλειψης.



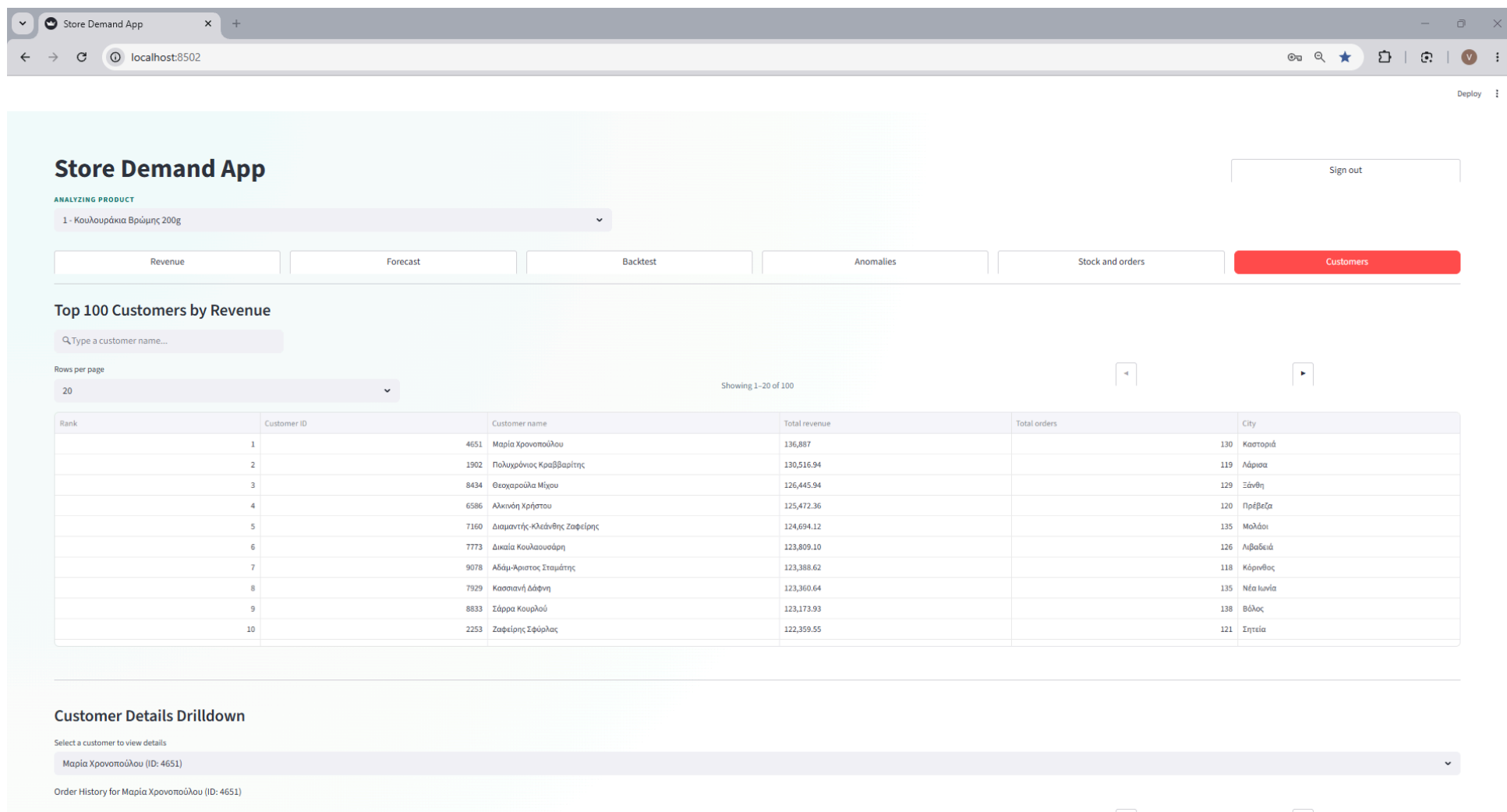
Σχήμα 5.6: Καρτέλα Αποθέματος - Διαχείριση αποθέματος & παραγγελιών

5.6.6 Καρτέλα Customers

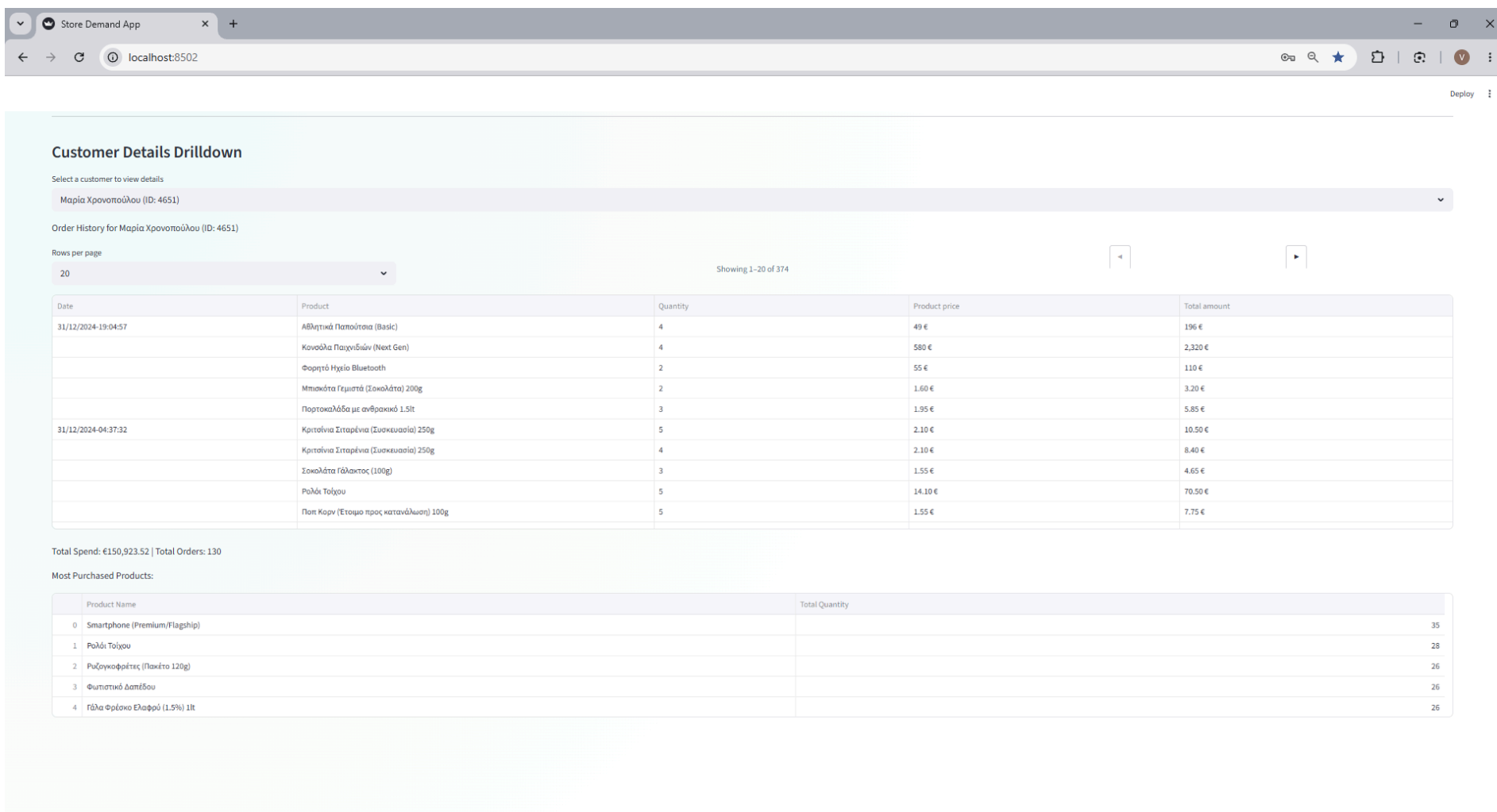
Η καρτέλα Customers παρέχει εικόνα της συμπεριφοράς των πελατών σε δύο επίπεδα.

Στο πρώτο επίπεδο εμφανίζεται leaderboard με τους 100 κορυφαίους πελάτες βάσει συνολικών εσόδων, με δυνατότητα αναζήτησης ονόματος. Ο πίνακας περιλαμβάνει: κατάταξη, αναγνωριστικό, όνομα, συνολικά έσοδα, αριθμό παραγγελιών και πόλη.

Στο δεύτερο επίπεδο ο χρήστης επιλέγει έναν συγκεκριμένο πελάτη από dropdown και βλέπει το πλήρες ιστορικό παραγγελιών του (εως 500 εγγραφές), ταξινομημένο από την πιο πρόσφατη συναλλαγή. Εμφανίζονται επίσης συνοπτικά στοιχεία: συνολικές δαπάνες, αριθμός παραγγελιών και τα 5 προϊόντα που αγοράστηκαν περισσότερο.



Σχήμα 5.7.1: Καρτέλα Πελατών - Ανάλυση συμπεριφοράς πελατών



Σχήμα 5.7.2: Καρτέλα Πελατών - Ανάλυση συμπεριφοράς πελατών

6. Αποτελέσματα και Αξιολόγηση

Το κεφάλαιο αυτό παρουσιάζει τα αποτελέσματα της αξιολόγησης των μοντέλων πρόβλεψης βάσει των συνθετικών δεδομένων που χρησιμοποιήθηκαν στο σύστημα. Η αξιολόγηση πραγματοποιήθηκε σε 100 προϊόντα, με χρονικό διαχωρισμό 80/20.

6.1 Μετρικές αξιολόγησης μοντέλων

Ο παρακάτω πίνακας παρουσιάζει τη μέση απόδοση των μοντέλων πρόβλεψης στο σύνολο ελέγχου των 100 προϊόντων. Για κάθε μοντέλο υπολογίζονται οι μετρικές MAE, RMSE και MAPE. Οι αριθμοί του πίνακα δεν είναι προβλέψεις πωλήσεων, αλλά μετρήσεις του σφάλματος μεταξύ των πραγματικών και των προβλεπόμενων πωλήσεων. Το MAE (Mean Absolute Error) εκφράζει τη μέση απόλυτη απόκλιση και βρίσκεται στις ίδιες μονάδες με τη ζήτηση. Για παράδειγμα, MAE ίσο με 14,516 σημαίνει ότι η πρόβλεψη διαφέρει κατά μέσο όρο περίπου 14,5 μονάδες προϊόντος από την πραγματική ζήτηση. Το RMSE (Root Mean Squared Error) μετρά επίσης το σφάλμα, αλλά επηρεάζεται περισσότερο από τις μεγάλες αποκλίσεις. Το MAPE (Mean Absolute Percentage Error) εκφράζει το μέσο σφάλμα ως ποσοστό και διευκολύνει τη σύγκριση προϊόντων με διαφορετική κλίμακα πωλήσεων. Και στις τρεις μετρικές ισχύει ότι όσο μικρότερη είναι η τιμή, τόσο καλύτερη είναι η πρόβλεψη. Ο πίνακας ταξινομείται κατά αύξον MAE, επειδή το MAE χρησιμοποιείται ως βασικό κριτήριο επιλογής του καλύτερου μοντέλου. Οι τιμές αποτελούν μέσους όρους και επομένως δεν σημαίνει ότι κάθε προϊόν παρουσιάζει ακριβώς το ίδιο σφάλμα.

Μοντέλο	MAE	RMSE	MAPE(%)
ARIMA	14,516	17,922	17,456
Holt-Winters	14,601	18,022	17,502
Random Forest	15,391	18,952	18,690
Moving Average	15,395	19,093	18,289
Extra Trees	15,471	19,049	18,661
LightGBM	15,918	19,653	19,040

Naive	19,3	23,200	22,747
Seasonal Naive	20,036	24,962	23,751

6.2 Σύγκριση μοντέλων ανά προϊόν

Η αυτόματη επιλογή του βέλτιστου μοντέλου ανά προϊόν βασίζεται στη χαμηλότερη τιμή MAE στο σύνολο ελέγχου. Επομένως, ο παρακάτω πίνακας δεν δείχνει ποιο μοντέλο είναι γενικά καλύτερο για όλα τα προϊόντα, αλλά πόσα προϊόντα ανέθεσε το σύστημα σε κάθε μοντέλο ως καταλληλότερη επιλογή. Για κάθε προϊόν συγκρίνονται όλα τα διαθέσιμα μοντέλα και επιλέγεται εκείνο που έχει τη μικρότερη μέση απόκλιση από τις πραγματικές πωλήσεις.

Μοντέλο	Αριθμός προϊόντων
ARIMA	28
Moving Average	17
Holt-Winters	15
Naive	14
LightGBM	11
Extra Trees	9
Random Forest	6

Τα στατιστικά μοντέλα (ARIMA, Holt-Winters, Moving Average) κυριάρχησαν στην επιλογή βέλτιστου, καλύπτοντας συνολικά το 60% των προϊόντων. Τα ML μοντέλα (LightGBM, Extra Trees, Random Forest) κατέλαβαν το 26%, ενώ το απλό μοντέλο αναφοράς Naive αναδείχθηκε βέλτιστο για το υπόλοιπο 14% (14 προϊόντα). Το εύρημα αυτό δεν υποδηλώνει αποτυχία των υπόλοιπων μοντέλων, αλλά ότι για προϊόντα με ιδιαίτερα σταθερή ή χαμηλής μεταβλητότητας ζήτηση, η απλή επανάληψη της τελευταίας τιμής μπορεί να αποδώσει χαμηλότερο MAE από πιο σύνθετα μοντέλα. Το Seasonal Naive, αν και αξιολογήθηκε εξίσου σε όλα τα προϊόντα, δεν αναδείχθηκε βέλτιστο σε

κανένα, εμφανίζοντας τη μεγαλύτερη μέση MAE (20.04) του συνόλου. Το αποτέλεσμα για τα υπόλοιπα 86 προϊόντα είναι συνεπές με τη βιβλιογραφία, που υποδεικνύει ότι απλούστερα στατιστικά μοντέλα συχνά υπερτερούν σε σύντομους ορίζοντες πρόβλεψης με εβδομαδιαία εποχικότητα (Makridakis et al., 2018).

6.3 Αποτελέσματα Backtest

Η λειτουργία backtest χρησιμοποιείται για την αξιολόγηση των μοντέλων σε ιστορικές συνθήκες, χρησιμοποιώντας δεδομένα τα οποία δεν συμμετείχαν στη διαδικασία εκπαίδευσης. Για μια επιλεγμένη ημερομηνία, κάθε μοντέλο παράγει πρόβλεψη για τις επόμενες 30 ημέρες. Στη συνέχεια, οι προβλεπόμενες τιμές συγκρίνονται με τις πραγματικές ποσότητες που καταγράφηκαν κατά την ίδια περίοδο.

Η σύγκριση πραγματοποιείται τόσο οπτικά, μέσω γραφημάτων, όσο και αριθμητικά, μέσω των μετρικών MAE, RMSE και MAPE. Το μοντέλο με τη χαμηλότερη τιμή MAE θεωρείται ότι προσεγγίζει καλύτερα την πραγματική ζήτηση για το συγκεκριμένο προϊόν και τη συγκεκριμένη περίοδο. Η χρήση περισσότερων από μίας μετρικών είναι χρήσιμη, επειδή το MAE δείχνει τη μέση απόκλιση σε μονάδες προϊόντος, το RMSE αναδεικνύει τις μεγάλες αστοχίες και το MAPE εκφράζει το σφάλμα ως ποσοστό.

Στις περιόδους ομαλής ζήτησης, οι προβλεπόμενες τιμές ακολουθούν γενικά την πορεία των πραγματικών πωλήσεων. Αντίθετα, μεγαλύτερες αποκλίσεις εμφανίζονται σε περιόδους απότομης αύξησης ή μείωσης της ζήτησης. Οι αποκλίσεις αυτές συνδέονται με ανωμαλίες, επειδή ένα μοντέλο εκπαιδευμένο σε προηγούμενα μοτίβα δυσκολεύεται να προβλέψει ένα ξαφνικό και μη επαναλαμβανόμενο γεγονός.

Τα αποτελέσματα του backtest πρέπει να ερμηνεύονται σε συνδυασμό με τις συνολικές μετρικές των ενοτήτων 6.1 και 6.2. Οι μέσες τιμές του κεφαλαίου 6 δείχνουν τη γενική συμπεριφορά των μοντέλων, ενώ το backtest δείχνει πώς συμπεριφέρονται σε συγκεκριμένα προϊόντα και χρονικές περιόδους. Επομένως, ένα μοντέλο μπορεί να έχει την καλύτερη συνολική επίδοση, αλλά διαφορετικό μοντέλο να αποδίδει καλύτερα σε ένα συγκεκριμένο προϊόν ή σε μία συγκεκριμένη περίοδο.

7. Εγκατάσταση και Ανάπτυξη

Το κεφάλαιο αυτό περιγράφει τα βήματα που απαιτούνται για την εγκατάσταση και εκκίνηση του συστήματος σε περιβάλλον Windows.

7.1 Ρύθμιση περιβάλλοντος και βάσης δεδομένων

Προαπαιτούμενα:

- Python 3.14 (<https://www.python.org>)
- MySQL 8.0 (<https://dev.mysql.com/downloads/mysql/>)

Μετά την εγκατάσταση των παραπάνω, δημιουργούνται οι δύο βάσεις δεδομένων (retail_db και retail_db_normalized) και εκτελούνται διαδοχικά τα πέντε scripts επεξεργασίας:

```
python scripts/1_create_ml_features.py  
python scripts/2_train_model.py  
python scripts/3_generate_forecasts.py  
python scripts/4_detect_anomalies.py  
python scripts/5_generate_backtest_forecasts.py
```

Οι εξαρτήσεις εγκαθίστανται με:

```
pip install -r requirements.txt
```

Οι παράμετροι σύνδεσης με τη βάση δεδομένων (host, user, password) ορίζονται στο αρχείο config.yaml, ενώ οι παράμετροι των μοντέλων (lag, days, rolling windows, LightGBM hyperparameters, ανώφλια ανωμαλιών) ρυθμίζονται επίσης από το ίδιο αρχείο.

7.2 Διαμόρφωση Streamlit

Η εφαρμογή Streamlit ρυθμίζεται μέσω του αρχείου config.toml:

```
[server]
headless = true
port = 8502

[browser]
gatherUsageStats = false
```

Τα διαπιστευτήρια χρηστών αποθηκεύονται στο secrets.toml σε μορφή SHA-256 hash. Η εκκίνηση της εφαρμογής γίνεται με:

```
streamlit run store_intelligence_app.py
```

και είναι προσβάσιμη στο <http://localhost:8502>.

7.3 Αυτόματη εκκίνηση στα Windows

Για αυτόματη εκκίνηση της εφαρμογής κατά την έναρξη του συστήματος, δημιουργείται αρχείο .bat με την εντολή εκκίνησης του Streamlit και καταχωρείται στο Task Scheduler των Windows ώστε να εκτελείται αυτόματα κατά τη σύνδεση του χρήστη.

8. Συμπεράσματα και Δυνατές επεκτάσεις

Η παρούσα εργασία παρουσίασε τη σχεδίαση και υλοποίηση ενός ολοκληρωμένου backend συστήματος επιχειρησιακών δεικτών απόδοσης (KPIs) με δυνατότητες προβλεπτικής ανάλυσης για το λιανεμπόριο. Το σύστημα συνδυάζει επεξεργασία δεδομένων, εκπαίδευση μοντέλων μηχανικής μάθησης και διαδραστικό dashboard σε μια ενιαία εφαρμογή.

Τα κύρια συμπεράσματα ήταν τα εξής:

- Η αυτόματη επιλογή βέλτιστου μοντέλου ανά προϊόν αποδείχθηκε αποτελεσματική στρατηγική, καθώς δεν υπάρχει ένα μοντέλο που υπερτερεί για όλα τα προϊόντα. Τα στατιστικά μοντέλα (ARIMA, Holt-Winters, Moving Average) κυριάρχησαν σε 60% των προϊόντων, επιβεβαιώνοντας την αξία τους σε σύντομους ορίζοντες πρόβλεψης.
- Η ανίχνευση ανωμαλιών με Robust z-score αποδείχθηκε κατάλληλη για δεδομένα λιανεμπορίου, καθώς είναι ανθεκτική σε ακραίες τιμές που χαρακτηρίζουν τη ζήτηση προϊόντων.
- Η χρήση Streamlit για το frontend επέτρεψε ταχεία ανάπτυξη διαδραστικού dashboard χωρίς την ανάγκη ξεχωριστής web εφαρμογής, μειώνοντας σημαντικά την πολυπλοκότητα της αρχιτεκτονικής.
- Η τριεπίπεδη αρχιτεκτονική (presentation - application - data) εξασφάλισε διαχωρισμό ευθυνών και ευκολία συντήρησης.

Το σύστημα μπορεί να επεκταθεί προς τις ακόλουθες κατευθύνσεις:

- Αυτόματη επανεκπαίδευση μοντέλων: Προγραμματισμένη εκτέλεση των scripts εκπαίδευσης σε τακτά διαστήματα (π.χ. εβδομαδιαία) ώστε τα μοντέλα να ενημερώνονται με νέα δεδομένα χωρίς χειροκίνητη παρέμβαση.
- Ενσωμάτωση πραγματικών δεδομένων: Σύνδεση με POS σύστημα ή ERP για φόρτωση πραγματικών συναλλαγών σε πραγματικό χρόνο.
- Πρόβλεψη τιμής: Επέκταση του μοντέλου ώστε να προβλέπει όχι μόνο την ποσότητα αλλά και τα αναμενόμενα έσοδα, λαμβάνοντας υπόψη δυναμική τιμολόγηση.

- Ειδοποιήσεις: Αυτόματη αποστολή email ή push notification όταν ανιχνευθεί ανωμαλία ή όταν το απόθεμα πλησιάζει κρίσιμο επίπεδο.
- Ανάπτυξη σε cloud: Μεταφορά της εφαρμογής σε cloud περιβάλλον (π.χ. AWS, Azure) για απομακρυσμένη πρόσβαση πολλαπλών χρηστών με διαφορετικά επίπεδα δικαιωμάτων.
- Επεξηγησιμότητα: Ενσωμάτωση βιβλιοθηκών ερμηνείας μοντέλων (π.χ. SHAP) ώστε ο χρήστης να βλέπει ποιοί παράγοντες επηρεάζουν περισσότερο κάθε πρόβλεψη.

Παράρτημα Α: Οδηγίες Εγκατάστασης

Το παράρτημα αυτό περιγράφει αναλυτικά τα βήματα εγκατάστασης και εκκίνησης του συστήματος Store Intelligence σε περιβάλλον Windows.

A.1 Προαπαιτούμενα

Πριν την εγκατάσταση, απαιτείται η ύπαρξη των παρακάτω:

Python 3.11 ή νεότερο (δοκιμασμένο με Python 3.14)

MySQL Server 8.0 ή νεότερο με δικαιώματα δημιουργίας βάσεων δεδομένων

Git (προαιρετικό, για κλωνοποίηση αποθετηρίου)

Ελεύθερος χώρος δίσκου: τουλάχιστον 2 GB για δεδομένα και μοντέλα

A.2 Δημιουργία Εικονικού Περιβάλλοντος

Ανοίξτε το PowerShell στον φάκελο του έργου και εκτελέστε:

```
python -m venv .venv  
.venv\Scripts\Activate.ps1
```

A.3 Εγκατάσταση Εξαρτήσεων

Εγκαταστήστε όλα τα απαιτούμενα πακέτα μέσω του αρχείου requirements.txt:

```
pip install -r requirements.txt  
pip install prophet
```

Τα κύρια πακέτα που εγκαθίστανται περιλαμβάνουν: pandas, numpy, scikit-learn, lightgbm, statsmodels, streamlit, plotly, SQLAlchemy, mysql-connector-python και loguru.

A.4 Ρύθμιση Βάσης Δεδομένων

Συνδεθείτε στο MySQL και εκτελέστε τα παρακάτω:

```
CREATE DATABASE retail_db CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;  
CREATE DATABASE retail_db_normalized CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci;
```

Στη συνέχεια, εισαγάγετε τα δεδομένα στη retail_db από το αρχείο dump ή εκτελέστε το script δημιουργίας δεδομένων.

A.5 Ρύθμιση Αρχείων Παραμέτρων

config.yaml -Ορισμός παραμέτρων pipeline:

```
database:  
  host: localhost  
  port: 3306  
  user: root  
  password: YOUR_PASSWORD  
  raw_db: retail_db  
  normalized_db: retail_db_normalized
```

secrets.toml -Διαπιστευτήρια χρηστών (SHA-256 hashed):

```
[passwords]  
admin = "sha256_hash_του_κωδικού"
```

config.toml - Ρυθμίσεις Streamlit:

```
[server]  
headless = true  
port = 8502
```

A.6 Εκτέλεση ML Pipeline

Τα scripts εκτελούνται διαδοχικά με την παρακάτω σειρά:

```
python scripts/1_create_ml_features.py  
python scripts/2_train_model.py  
python scripts/3_generate_forecasts.py  
python scripts/4_detect_anomalies.py
```

Κάθε script καταγράφει την πρόοδό του στον φάκελο logs. Η συνολική διάρκεια εκτέλεσης εξαρτάται από τον αριθμό προϊόντων (για 100 προϊόντα: περίπου 15–30 λεπτά).

A.7 Εκκίνηση Dashboard

Εκτελέστε την εφαρμογή Streamlit:

```
streamlit run store_intelligence_app.py
```

Ανοίξτε τον browser στη διεύθυνση: <http://localhost:8502>

Εισάγετε τα διαπιστευτήριά σας και το dashboard είναι έτοιμο προς χρήση.

Παράρτημα Β: Δομή Βάσης Δεδομένων

Το σύστημα Store Intelligence αξιοποιεί δύο βάσεις δεδομένων MySQL: την retail_db, η οποία αποθηκεύει τα ακατέργαστα συναλλακτικά δεδομένα, και την retail_db_normalized, η οποία περιέχει τα επεξεργασμένα δεδομένα, τα χαρακτηριστικά ML και τα αποτελέσματα του pipeline πρόβλεψης.

B.1 Βάση Δεδομένων retail_db

Ο πίνακας full_retail_dataset αποθηκεύει σε ενοποιημένη μορφή τα δεδομένα συναλλαγών, ενσωματώνοντας πληροφορίες από πολλαπλές πηγές. Τα βασικά πεδία του είναι: sale_date (ημερομηνία συναλλαγής), product_id και product_name (αναγνωριστικό και όνομα προϊόντος), product_category (κατηγορία προϊόντος), total_amount (συνολική αξία συναλλαγής), customer_id (αναγνωριστικό πελάτη), quantity (ποσότητα), unit_price και discount_applied (τιμή μονάδας και έκπτωση), καθώς και εξωτερικές μεταβλητές όπως temperature, rain_mm (καιρικές συνθήκες), is_holiday (αργία) και is_promo, promotion_type, promotion_discount_value (προωθητικές ενέργειες). Ο πίνακας χρησιμοποιείται αποκλειστικά από την καρτέλα Revenue του dashboard για ανάλυση εσόδων και δεν τροποποιείται από τα scripts επεξεργασίας.

B.2 Βάση Δεδομένων retail_db_normalized

Η βάση retail_db_normalized αποτελεί το επεξεργασμένο επίπεδο του συστήματος και οργανώνεται σε τρεις λογικές ομάδες πινάκων.

Συναλλακτικοί πίνακες: Οι πίνακες customers, products, sales, sale_items, promotions, weather_daily και holidays αποτελούν την κανονικοποιημένη αναπαράσταση των πρωτογενών δεδομένων. Κάθε πίνακας διαθέτει πρωτεύον κλειδί και ξένα κλειδιά που εξασφαλίζουν αναφορική ακεραιότητα. Ο πίνακας products περιλαμβάνει επιπλέον πεδία που αφορούν στις λειτουργικές ανάγκες του συστήματος: avg_lifespan_days για τη διάρκεια ζωής προϊόντος, space_needed(m³) και fridge_space_needed(m³) για τις ανάγκες αποθήκευσης.

Πίνακες ML Pipeline: Ο πίνακας ml_features_ready αποθηκεύει τον πλήρη πίνακα χαρακτηριστικών που παράγεται από το script 1_create_ml_features.py. Περιλαμβάνει lag

features για ορίζοντες 1, 7, 14, 21 και 28 ημερών, rolling window στατιστικά (μέσος όρος, τυπική απόκλιση, ελάχιστο, μέγιστο) για παράθυρα 7, 14 και 30 ημερών, καθώς και μετεωρολογικές μεταβλητές και δείκτες αργιών και Σαββατοκύριακων.

Ο πίνακας `product_model_scores` αποθηκεύει τις βαθμολογίες MAE, RMSE και MAPE για κάθε συνδυασμό προϊόντος και μοντέλου, ενώ ο πίνακας `product_best_model` καταγράφει το βέλτιστο μοντέλο ανά προϊόν μαζί με την τιμή MAE επιλογής. Ο πίνακας `product_forecasts` αποθηκεύει τις παραγόμενες προβλέψεις με διάκριση τύπου (future για προβλέψεις μέλλοντος, backtest για ιστορική αξιολόγηση). Ο πίνακας `product_anomalies` περιέχει τις ανιχνευθείσες ανωμαλίες με πεδία τύπου (spike/dip), σοβαρότητας (high/medium), τιμής Robust Z-Score, πραγματικής και αναμενόμενης ποσότητας και τιμής MAD.

Πίνακες αποθέματος: Ο πίνακας `product_stock` αποθηκεύει το τρέχον απόθεμα κάθε προϊόντος, την ημερομηνία και ποσότητα τελευταίου ανεφοδιασμού. Ο πίνακας `product_inventory_settings` περιέχει τις παραμέτρους επαναπαραγγελίας: `lead_time_days` (χρόνος παράδοσης), `safety_days` (ημέρες ασφαλείας), `min_order_qty` (ελάχιστη ποσότητα παραγγελίας) και `current_stock` (τρέχον επίπεδο αποθέματος). Τα δύο αυτά δεδομένα τροφοδοτούν την καρτέλα διαχείρισης αποθέματος του dashboard, όπου υπολογίζεται αυτόματα η ημερομηνία αναπλήρωσης βάσει της πρόβλεψης ζήτησης.

Βιβλιογραφία

- Apache Software Foundation. (2024). Apache Superset Documentation.
<https://superset.apache.org/docs/intro>
- Bontempi, G., Ben Taieb, S., & Le Borgne, Y. A. (2013). Machine learning strategies for time series forecasting. *Business Intelligence*, 62–77.
https://doi.org/10.1007/978-3-642-36318-4_3
- Box, G. E. P., & Jenkins, G. M. (1976). *Time Series Analysis: Forecasting and Control*. Holden-Day. <https://www.wiley.com/en-us/Time+Series+Analysis>
- Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32.
<https://doi.org/10.1023/A:1010933404324>
- Fildes, R., Ma, S., & Kolassa, S. (2019). Retail forecasting: Research and practice. *International Journal of Forecasting*, 38(4), 1283–1318.
<https://doi.org/10.1016/j.ijforecast.2019.06.004>
- Fowler, M. (2002). *Patterns of Enterprise Application Architecture*. Addison-Wesley. <https://www.martinfowler.com/books/ea.html>
- Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1), 3–42. <https://doi.org/10.1007/s10994-006-6226-1>
- Holt, C. C. (2004). Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, 20(1), 5–10.
<https://doi.org/10.1016/j.ijforecast.2003.09.015>
- 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>
- Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The Unified Software Development Process*. Addison-Wesley.
<https://www.informit.com/store/unified-software-development-process-9780201571691>

- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 30.
https://papers.nips.cc/paper_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html
- Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit* (3rd ed.). Wiley.
<https://www.wiley.com/en-us/The+Data+Warehouse+Toolkit>
- Laudon, K. C., & Laudon, J. P. (2020). *Management Information Systems* (16th ed.). Pearson.
<https://www.pearson.com/en-us/subject-catalog/p/management-information-systems>
- Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). The M4 Competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34(4), 802–808. <https://doi.org/10.1016/j.ijforecast.2018.06.001>
- Makridakis, S., Wheelwright, S. C., & Hyndman, R. J. (1998). *Forecasting: Methods and Applications* (3rd ed.). Wiley.
<https://www.wiley.com/en-us/Forecasting>
- Metabase Inc. (2024). *Metabase Documentation*.
<https://www.metabase.com/docs/latest/>
- Microsoft Corporation. (2024). *Microsoft Power BI*. <https://powerbi.microsoft.com>
- MySQL Community Edition. (2024). *MySQL 8.0 Reference Manual*. Oracle Corporation. <https://dev.mysql.com/doc/refman/8.0/en/>
- Oracle Corporation. (2024). *Oracle Retail Solutions*. <https://www.oracle.com/retail/>
- SAP SE. (2024). *SAP S/4HANA for Retail*.
<https://www.sap.com/products/scm/retail.html>
- SQLAlchemy. (2024). *SQLAlchemy Documentation*. <https://docs.sqlalchemy.org>
- Streamlit Inc. (2024). *Streamlit Documentation*. <https://docs.streamlit.io>

Tableau Software. (2024). Tableau Business Intelligence Platform.

<https://www.tableau.com>

Taylor, S. J., & Letham, B. (2018). Forecasting at scale. The American Statistician, 72(1), 37–45. <https://doi.org/10.1080/00031305.2017.1380080>

Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. CreateSpace.

<https://www.python.org/doc/>

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.