

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Σχολή Χρηματοοικονομικής και Στατιστικής



ΤΜΗΜΑ ΣΤΑΤΙΣΤΙΚΗΣ ΚΑΙ ΑΣΦΑΛΙΣΤΙΚΗΣ ΕΠΙΣΤΗΜΗΣ
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ ΣΤΗΝ
ΕΦΑΡΜΟΣΜΕΝΗ ΣΤΑΤΙΣΤΙΚΗ

ΑΝΑΛΥΣΗ ΧΡΗΜΑΤΟΟΙΚΟΝΟΜΙΚΩΝ ΔΕΔΟΜΕΝΩΝ ΜΕ ΕΝΙΣΧΥΣΗ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ

ΣΟΦΟΣ ΓΕΩΡΓΙΟΣ

Διπλωματική Εργασία

που υποβλήθηκε στο Τμήμα Στατιστικής και
Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς
ως μέρος των απαιτήσεων για την απόκτηση
του Μεταπτυχιακού Διπλώματος Ειδίκευσης στην
Εφαρμοσμένη Στατιστική

Πειραιάς

Σεπτέμβριος 2026

Η παρούσα Διπλωματική Εργασία εγκρίθηκε ομόφωνα από την Τριμελή Εξεταστική Επιτροπή που ορίστηκε από τη Συνέλευση του Τμήματος Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς στην υπ' αριθμ..... συνεδρίασή της, σύμφωνα με τον Εσωτερικό Κανονισμό Λειτουργίας του Προγράμματος Μεταπτυχιακών Σπουδών στην Εφαρμοσμένη Στατιστική.

Τα μέλη της Επιτροπής ήταν:

- Καθηγητής Πελέκης Νικόλαος (Επιβλέπων)
- Αναπληρωτής Καθηγητής Ευαγγελάρας Χαράλαμπος
- Αναπληρωτής Καθηγητής Ρακιτζής Αθανάσιος

Η έγκριση της Διπλωματικής Εργασίας από το Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης του Πανεπιστημίου Πειραιώς δεν υποδηλώνει αποδοχή των γνωμών του συγγραφέα.

UNIVERSITY OF PIRAEUS

School of Finance and Statistics



DEPARTMENT OF STATISTICS AND INSURANCE SCIENCE
MASTER'S PROGRAM IN
APPLIED STATISTICS

**AN AI-ENHANCED FINANCIAL
DATA ANALYTICS DASHBOARD**

GEORGIOS SOFOS

MSc Dissertation

submitted to the Department of Statistics and
Insurance Science of the University of Piraeus
in partial fulfillment of the requirements for the degree of
Master of Science in
Applied Statistics

Piraeus

September 2026

Στους γονείς μου

Παντελής και Δέσποινα

Πρόλογος και Ευχαριστίες

Η συνεχής ανάπτυξη των πληροφοριακών συστημάτων και η διαθεσιμότητα των επιχειρησιακών δεδομένων έχουν δημιουργήσει την ανάγκη για αποτελεσματικότερες διαδικασίες επεξεργασίας και αξιοποίησής τους. Η ποιότητα και η σωστή οργάνωση των δεδομένων αποτελούν βασική προϋπόθεση τόσο για την ανάλυση όσο και για την παραγωγή αξιόπιστων προβλέψεων.

Στην παρούσα διπλωματική εργασία αναπτύσσεται μία διαδικτυακή εφαρμογή για την αυτοματοποιημένη εισαγωγή, καθαρισμό, κατηγοριοποίηση, αποθήκευση και ανάλυση δεδομένων, με έμφαση στα οικονομικά στοιχεία και δεδομένα κρατήσεων. Παράλληλα, εξετάζεται η αξιοποίηση ιστορικών δεδομένων για την πρόβλεψη μελλοντικών τιμών και την υποστήριξη της διαδικασίας λήψης αποφάσεων.

Με την ολοκλήρωση της εργασίας αυτής, θα ήθελα να ευχαριστήσω θερμά τον Επιβλέποντα Καθηγητή μου για την καθοδήγηση, τις συμβουλές και την υποστήριξή του καθ' όλη την διάρκεια της εκπόνησης της διπλωματικής εργασίας.

Τέλος, ιδιαίτερες ευχαριστίες οφείλω στην οικογένειά μου και στους κοντινούς μου ανθρώπους για την υπομονή, την κατανόηση και την διαρκή τους στήριξη σε όλη τη διάρκεια των σπουδών μου.

Πειραιάς, Σεπτέμβριος 2026

Σοφός Γεώργιος

Περίληψη

Η συνεχής ψηφιοποίηση των επιχειρησιακών διαδικασιών έχει αυξήσει τον όγκο των διαθέσιμων δεδομένων, χωρίς όμως να εξασφαλίζεται η άμεση αξιοποίησή τους. Στην πράξη, αρχεία διαφορετικής προέλευσης παρουσιάζουν συχνά ανομοιογένεια ως προς τη δομή, τις ονομασίες πεδίων, τις ημερομηνίες και τις αριθμητικές τιμές, γεγονός που καθιστά απαραίτητη την εφαρμογή συστηματικών διαδικασιών καθαρισμού, τυποποίησης και ελέγχου. Σκοπός της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός και η ανάπτυξη μιας διαδικτυακής εφαρμογής για την αυτοματοποιημένη εισαγωγή, επεξεργασία, αποθήκευση, κατηγοριοποίηση, ανάλυση και αξιοποίηση επιχειρησιακών δεδομένων. Η εφαρμογή υλοποιείται με διαχωρισμένη αρχιτεκτονική, χρησιμοποιώντας Next.js, React και TypeScript στο επίπεδο παρουσίασης, Python και FastAPI στο backend και Supabase για την σχεσιακή αποθήκευση μέσω PostgreSQL και διαχείριση φυσικών αρχείων μέσω Storage. Ιδιαίτερη έμφαση δίνεται στην ιχνηλασιμότητα, καθώς το αρχικό αρχείο, τα μετα-δεδομένα του, οι αρχικές και καθαρισμένες εγγραφές και τα βασικά στάδια επεξεργασίας διατηρούνται συνδεδεμένα μέσω μοναδικών αναγνωριστικών και σχέσεων μεταξύ των πινάκων. Παράλληλα, αναπτύσσεται ανεξάρτητο σύστημα πρόβλεψης χρονοσειρών, στο οποίο διαφορετικά μοντέλα αξιολογούνται με κοινές μετρικές και διαδικασίες backtesting πριν από την επιλογή και αποθήκευση των αποτελεσμάτων. Τα επεξεργασμένα δεδομένα και οι προβλέψεις διατίθενται στη συνέχεια σε περιβάλλον επιχειρηματικής ευφυΐας για τη δημιουργία διαδραστικών αναφορών. Η τελική προσέγγιση δημιουργεί μια επαναλήψιμη ροή από το αρχικό αρχείο έως την οργανωμένη πληροφορία, περιορίζοντας την χειροκίνητη επεξεργασία και ενισχύοντας την συνέπεια και τη δυνατότητα μελλοντικής επέκτασης του συστήματος.

Λέξεις-κλειδιά: ποιότητα δεδομένων, καθαρισμός δεδομένων, τυποποίηση, διαδικτυακή εφαρμογή, FastAPI, Supabase, χρονοσειρές, πρόβλεψη, επιχειρηματική ευφυΐα.

Abstract

The continuous digitalization of business processes has significantly increased the volume of available data, without necessarily ensuring that such data can be directly used for analysis. In practice, files originating from different systems frequently present heterogeneous structures, field names, date formats, numerical representations and levels of completeness, making systematic cleaning, standardization and validation necessary. The aim of this thesis is the design and development of a web-based application for the automated ingestion, processing, storage, classification, analysis and utilization of business data. The application follows a separated architecture, using Next.js, React and TypeScript for the presentation layer, Python and FastAPI for the backend, and Supabase for relational persistence through PostgreSQL and physical file management through object storage. Particular emphasis is placed on traceability, since the original file, its metadata, the raw and cleaned records and the main processing stages remain connected through unique identifiers and relationships between database tables. In parallel, an independent time-series forecasting subsystem is developed, in which multiple forecasting models can be evaluated using common error metrics and backtesting procedures before model selection. Processed data and forecasting outputs can subsequently be consumed by a business intelligence environment for the development of interactive reports and analytical visualizations. The resulting approach establishes a repeatable and extensible workflow from heterogeneous source files to structured and analytically useful information, reducing repeated manual processing while improving consistency, auditability and the ability to extend the system with additional processing rules, data sources and analytical methods in the future.

Keywords: data quality, data cleaning, standardization, web application, FastAPI, Supabase, time series, forecasting, business intelligence.

Περιεχόμενα

Κατάλογος Πινάκων	4
Κατάλογος Διαγραμμάτων	5
1 Εισαγωγή	7
1.1 Αντικείμενο της εργασίας	7
1.2 Αναγκαιότητα της εργασίας	8
1.3 Σκοπός και επιμέρους στόχοι	9
1.4 Μεθοδολογική προσέγγιση	9
1.5 Δομή της εργασίας	10
2 Θεωρητικό Υπόβαθρο	11
2.1 Από τα δεδομένα στην επιχειρησιακή γνώση	11
2.2 Ποιότητα των δεδομένων	12
2.3 Καθαρισμός, τυποποίηση και ολοκλήρωση των δεδομένων	13
2.4 Επιχειρηματική ευφυΐα και αναλυτικές τεχνικές	14
2.5 Χρονοσειρές και πρόβλεψη	15
2.5.1 Seasonal Naïve	15
2.5.2 Σταθμισμένος εποχικός μέσος	15
2.5.3 ETS (Error, Trend, Seasonality)	16
2.5.4 ARIMA και SARIMA	16
2.5.5 Παλινδρόμηση με εποχικά χαρακτηριστικά	17
2.5.6 Νευρωνικά δίκτυα και LSTM	17
2.6 Αξιολόγηση των μοντέλων πρόβλεψης	18
2.6.1 Μέσο απόλυτο σφάλμα (MAE - Mean Absolute Error)	18
2.6.2 Ρίζα Μέσου Τετραγωνικού Σφάλματος (RMSE)	19
2.6.3 Συμμετρικό Μέσο Απόλυτο Ποσοστιαίο Σφάλμα (sMAPE)	19

2.6.4	Μεροληψία (Bias)	19
2.6.5	Rolling και expanding-window backtesting	20
2.7	Εφαρμογές πρόβλεψης στον ξενοδοχειακό και τουριστικό τομέα	21
3	Υλοποίηση της Εφαρμογής	22
3.1	Συνολική αρχιτεκτονική και ροή λειτουργίας	22
3.2	Frontend, αρχική σελίδα και έλεγχος πρόσβασης	24
3.3	Επιλογή, μεταφόρτωση και καταχώριση αρχείων	25
3.4	Επικοινωνία μεταξύ διεπαφής και υποσυστήματος επεξεργασίας	27
3.5	Κεντρική διαδικασία επεξεργασίας και επιλογή μηχανισμού καθαρισμού	31
3.6	Καθαρισμός, τυποποίηση και έλεγχος των δεδομένων	32
3.7	Οργάνωση της βάσης δεδομένων, κλειδιά και σχέσεις	34
3.8	Υποσύστημα πρόβλεψης και τεχνική δημιουργία της χρονοσειράς	38
4	Εφαρμογή της Μεθοδολογίας στα Δεδομένα της Μελέτης	42
4.1	Οργάνωση των δεδομένων και καθορισμός της μονάδας ανάλυσης	42
4.2	Εναλλακτικές ονομασίες και δημιουργία του κανονικοποιημένου σχήματος	43
4.3	Μετασχηματισμός των δεδομένων κρατήσεων	45
4.4	Επεξεργασία των εσόδων και δημιουργία εγγραφής ανά παραστατικό	47
4.4.1	Κατηγοριοποίηση του πελάτη	51
4.4.2	Συμπεριλαμβανόμενα και εξαιρούμενα έσοδα (included/excluded)	53
4.5	Κατηγοριοποίηση των εξόδων και δημιουργία των οικονομικών πεδίων	53
4.5.1	Κατανομή του ΦΠΑ	58
4.6	Από το αποτέλεσμα του μηχανισμού καθαρισμού στην εγγραφή της βάσης δεδομένων	59
4.6.1	Διαφορά <i>raw_data</i> και <i>cleaned_data</i>	62
4.6.2	Σύνδεση με το <i>uploaded_files</i>	63
4.7	Δημιουργία των SQL views και του τελικού αναλυτικού σχήματος	64
4.7.1	Δημιουργία της προβολής εσόδων (<i>v_income_clean</i>)	65
4.7.2	Δημιουργία της προβολής εξόδων (<i>v_expenses_clean</i>)	66
4.7.3	Δημιουργία της προβολής κρατήσεων (<i>v_bookings_clean</i>)	67

4.7.4	Από τις SQL views στους τελικούς δείκτες	67
4.8	Δημιουργία των μηνιαίων χρονοσειρών και υπολογισμός της πληρότητας	68
4.8.1	Ρύθμιση υπερπαραμέτρων των SARIMA και LSTM και διαχωρισμός από τη ροή παραγωγής	72
5	Ανάλυση και Οπτικοποίηση στο Power BI	76
5.1	Διασύνδεση του Power BI με τη βάση PostgreSQL του Supabase . .	76
5.2	Πηγές δεδομένων και οργάνωση του μοντέλου	78
5.3	Φίλτρα, navigation buttons και εναλλαγή οπτικοποιήσεων	80
5.4	Υπολογιστικό επίπεδο DAX και βασικοί δείκτες	83
5.5	Overview και οικονομική ανάλυση εσόδων--εξόδων	86
5.5.1	Revenue Analysis	87
5.5.2	Expenses Analysis	89
5.6	Κρατήσεις, δωμάτια, διάρκεια διαμονής και πληρότητα	91
5.6.1	Bookings Analysis	91
5.6.2	Room Performance	92
5.6.3	Stay Length Analysis	94
5.6.4	Occupancy Analysis	94
5.7	Forecasting Analysis και επιλογή μοντέλου	96
5.8	Συμπεράσματα από την ανάλυση των dashboards	101
6	Συμπεράσματα και Μελλοντικές Επεκτάσεις	102
6.1	Συνολική αποτίμηση της εργασίας	102
6.2	Σημασία της ολοκληρωμένης ανάλυσης για τη λήψη αποφάσεων . . .	103
6.3	Περιορισμοί της παρούσας υλοποίησης	104
6.4	Επέκταση της επεξεργασίας σε περισσότερες ξενοδοχειακές μονάδες	105
6.5	Εξέλιξη του υποσυστήματος πρόβλεψης και των στρατηγικών εκπαίδευσης	107
6.6	Περαιτέρω αναλυτικές και επιχειρησιακές επεκτάσεις	108
6.7	Τελικό συμπέρασμα	109
	Βιβλιογραφία	111

Κατάλογος πινάκων

3.1	Κύρια τελικά σημεία του API της εφαρμογής.	30
3.2	Κύρια κλειδιά και σχέσεις του λειτουργικού σχήματος.	35
3.3	Ρόλος των βασικών πινάκων του υποσυστήματος πρόβλεψης.	37
4.1	Διαχωρισμός των δεδομένων και μονάδα ανάλυσης ανά κατηγορία. . . .	43
4.2	Ενδεικτικές εναλλακτικές ονομασίες βασικών πεδίων των δεδομένων κρα- τήσεων.	45
4.3	Βασικά κανονικοποιημένα πεδία και εναλλακτικές ονομασίες των αρ- χείων εσόδων.	48
4.4	Βασική ταξινόμηση λογιστικών γραμμών των εσόδων.	49
4.5	Βασικές εναλλακτικές ονομασίες των λογιστικών στηλών στα αρχεία εξό- δων.	54
4.6	Βασική λογική κατηγοριοποίησης των εξόδων ανά λογιστικό πρόθεμα. . .	57
4.7	Κύριες δομές της βάσης που συμμετέχουν στην αποθήκευση των επεξερ- γασμένων δεδομένων.	60
4.8	Βασικές φυσικές στήλες του πίνακα <i>file_rows</i>	61
4.9	Κύρια SQL views και ρόλος τους στο αναλυτικό επίπεδο.	65
4.10	Δημιουργία των μηνιαίων μεταβλητών που χρησιμοποιούνται στην πρό- βλεψη.	69
4.11	Ενδεικτικός μηνιαίος υπολογισμός πληρότητας.	71
4.12	Κύριες παράμετροι που εξετάζονται κατά τη ρύθμιση υπερπαραμέτρων των SARIMA και LSTM.	74
5.1	Στοιχεία της σύνδεσης Power BI--Supabase.	77
5.2	Κύριες πηγές του Supabase που χρησιμοποιούνται στο Power BI.	79
5.3	Διαδραστικά στοιχεία ανά βασική σελίδα του report.	82
6.1	Ενδεικτικές κατευθύνσεις εξέλιξης του υποσυστήματος πρόβλεψης. . . .	108

Κατάλογος σχημάτων

3.1	Συνολική αρχιτεκτονική και διαδρομές επικοινωνίας της εφαρμογής. . . .	23
3.2	Βασικό σχεσιακό σχήμα της λειτουργικής ροής και του Supabase Storage.	36
3.3	Σχεσιακή οργάνωση των αποτελεσμάτων της πρόβλεψης.	37
5.1	Κουμπιά πλοήγησης μεταξύ των σελίδων του Power BI dashboard. . . .	80
5.2	Επιλογή έτους.	81
5.3	Επιλογή μήνα.	81
5.4	Επιλογή ξενοδοχείου.	81
5.5	Κεντρική σελίδα επισκόπησης (Overview) του Power BI report.	87
5.6	Σελίδα Revenue Analysis του Power BI με τα βασικά φίλτρα, τα navigation buttons και τη συνολική ανάλυση των καναλιών εσόδων. . . .	88
5.7	Ανάλυση του καναλιού Online Travel Agency στη σελίδα Revenue Analysis.	88
5.8	Ανάλυση του καναλιού Direct στη σελίδα Revenue Analysis.	89
5.9	Σελίδα Expenses Analysis του Power BI με τη χρονική εξέλιξη και την ανάλυση των κατηγοριών εξόδων.	90
5.10	Εναλλαγή της σελίδας Expenses Analysis σε αναλυτικότερη προβολή κατηγοριών εξόδων.	90
5.11	Σελίδα Bookings Analysis του Power BI με τους βασικούς δείκτες κρατήσεων και τις αναλύσεις ανά χρόνο και προέλευση πελατών.	91
5.12	Αναλυτικότερη προβολή της γεωγραφικής κατανομής στη σελίδα Bookings Analysis.	92
5.13	Σελίδα Room Performance του Power BI με σύγκριση κρατήσεων, εσόδων και ADR ανά τύπο δωματίου.	93
5.14	Αναλυτικός πίνακας δεικτών στη σελίδα Room Performance.	93
5.15	Σελίδα Stay Length Analysis του Power BI με ανάλυση διάρκειας διαμονής και booking lead time.	94
5.16	Σελίδα Occupancy Analysis του Power BI με πληρότητα, ADR, RevPAR, GOPPAR, CPOR και αξιοποίηση της διαθέσιμης δυναμικότητας.	95
5.17	Σελίδα Forecasting Analysis του Power BI με επιλογή ξενοδοχείου, μοντέλου, training strategy και χρονικού ορίζοντα, καθώς και σύγκριση actual και forecast τιμών.	96

5.18	Forecasting Analysis για horizon 3 μηνών με strategy <i>All available history</i> .	98
5.19	Forecasting Analysis για horizon 3 μηνών με strategy <i>Last 30 months</i> . . .	99
5.20	Forecasting Analysis για horizon 12 μηνών με strategy <i>Last 30 months</i> . . .	100

Κεφάλαιο 1

Εισαγωγή

1.1 Αντικείμενο της εργασίας

Η συνεχής ανάπτυξη των πληροφοριακών συστημάτων και η ψηφιοποίηση των επιχειρησιακών διαδικασιών έχουν μεταβάλει σημαντικά τον τρόπο με τον οποίο οι οργανισμοί δημιουργούν, αποθηκεύουν και αξιοποιούν την πληροφορία. Καθημερινά παράγονται δεδομένα που σχετίζονται με οικονομικές συναλλαγές, πωλήσεις, αγορές, πελάτες, προμηθευτές, κρατήσεις, υπηρεσίες και γενικότερα με τη λειτουργία ενός οργανισμού. Η ύπαρξη αυτών των δεδομένων δημιουργεί τη δυνατότητα για συστηματικότερη παρακολούθηση της δραστηριότητας, αναγνώριση τάσεων, εντοπισμό προβλημάτων και υποστήριξη της διαδικασίας λήψης αποφάσεων. Ωστόσο, η απλή συγκέντρωση μεγάλου όγκου δεδομένων δεν συνεπάγεται αυτόματα την παραγωγή χρήσιμης πληροφορίας. Τα διαθέσιμα στοιχεία προέρχονται συχνά από διαφορετικές εφαρμογές και πληροφοριακά συστήματα και καταλήγουν σε ανεξάρτητα αρχεία, τα οποία μπορεί να έχουν διαφορετικές ονομασίες στηλών, διαφορετική θέση των επικεφαλίδων, εναλλακτικούς τρόπους αναπαράστασης ημερομηνιών και αριθμών, κενά πεδία ή πληροφορίες που για να χρησιμοποιηθούν αναλυτικά απαιτούν πρόσθετη ερμηνεία.

Το αντικείμενο της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός και η ανάπτυξη μιας ολοκληρωμένης διαδικτυακής εφαρμογής για την αυτοματοποιημένη εισαγωγή, επεξεργασία, αποθήκευση, κατηγοριοποίηση και ανάλυση επιχειρησιακών δεδομένων. Στόχος δεν είναι η δημιουργία ενός απλού μηχανισμού μεταφόρτωσης αρχείων, αλλά μιας ολοκληρωμένης και επαναλήψιμης ροής, η οποία ξεκινά από το αρχικό αρχείο και καταλήγει σε δεδομένα κατάλληλα για αναλυτική αξιοποίηση. Κάθε αρχείο αποκτά μοναδικό αναγνωριστικό, αποθηκεύεται αυτούσιο, επεξεργάζεται από το κατάλληλο υποσύστημα και το τελικό αποτέλεσμα διατηρείται τόσο σε επίπεδο αρχείου όσο και σε επίπεδο μεμονωμένης εγγραφής. Παράλληλα, δίνεται ιδιαίτερη σημασία στην ιχνηλασιμότητα, ώστε η αρχική και η καθαρισμένη μορφή να παραμένουν συνδεδεμένες και να

είναι δυνατό να αναζητηθεί η πορεία που οδήγησε στο τελικό αποτέλεσμα.

Ένα επιπλέον αντικείμενο της εργασίας είναι η αξιοποίηση του οργανωμένου ιστορικού για την εφαρμογή μεθόδων πρόβλεψης χρονοσειρών. Η πρόβλεψη δεν εφαρμόζεται απευθείας πάνω στα ανεπεξέργαστα αρχεία, αλλά ακολουθεί τον καθαρισμό, την τυποποίηση και τη δημιουργία συνεπών μηνιαίων χρονικών σειρών. Στην τελική εφαρμογή οι σειρές πρόβλεψης αφορούν τις ενεργές κρατήσεις, τα included καθαρά έσοδα και τα included έξοδα. Διαφορετικές μέθοδοι εκπαιδεύονται και αξιολογούνται μέσα σε κοινό χρονικά συνεπές πλαίσιο, ώστε η επιλογή να βασίζεται στην πραγματική προβλεπτική τους συμπεριφορά και όχι μόνο στην πολυπλοκότητά τους. Με τον τρόπο αυτό η εργασία συνδέει την ανάπτυξη διαδικτυακού λογισμικού, τη διαχείριση και την ποιότητα των δεδομένων και την αναλυτική αξιοποίησή τους σε ένα ενιαίο πληροφοριακό σύστημα.

1.2 Αναγκαιότητα της εργασίας

Η επεξεργασία επιχειρησιακών δεδομένων πραγματοποιείται συχνά με χειροκίνητες ενέργειες πάνω στα αρχικά αρχεία. Ο χρήστης χρειάζεται να εντοπίσει τις κατάλληλες στήλες, να διορθώσει μορφές ημερομηνιών και αριθμητικών τιμών, να εφαρμόσει φίλτρα, να δημιουργήσει κατηγορίες και στη συνέχεια να μεταφέρει τα αποτελέσματα σε νέο αρχείο ή σε κάποιο αναλυτικό περιβάλλον. Η διαδικασία αυτή μπορεί να είναι λειτουργική για μία μεμονωμένη εργασία, παρουσιάζει όμως σημαντικές αδυναμίες όταν επαναλαμβάνεται συστηματικά. Κάθε νέο αρχείο απαιτεί σημαντικό μέρος των ίδιων ενεργειών, η διαδικασία εξαρτάται από τον χρήστη που την εκτελεί και αυξάνεται η πιθανότητα διαφορετικής εφαρμογής των ίδιων κανόνων σε διαφορετικές χρονικές περιόδους.

Ένα δεύτερο πρόβλημα αφορά τη δυνατότητα ελέγχου. Μετασχηματισμοί που πραγματοποιούνται απευθείας σε ένα υπολογιστικό φύλλο μπορεί να είναι δύσκολο να αναπαραχθούν. Ένα φίλτρο που παρέμεινε ενεργό, μία διαγραφή, μία λανθασμένη μετατροπή ή μία χειροκίνητη αλλαγή μπορεί να επηρεάσει το τελικό αποτέλεσμα χωρίς να υπάρχει σαφής καταγραφή της διαδρομής. Το πρόβλημα γίνεται σημαντικότερο όταν τα ίδια δεδομένα χρησιμοποιούνται για πρόβλεψη, επειδή ένα μαθηματικά ορθό μοντέλο δεν μπορεί να διορθώσει συστηματικά σφάλματα που έχουν ήδη ενσωματωθεί στη χρονοσειρά. Για τον λόγο αυτό απαιτείται μια κεντρική, αυτοματοποιημένη και επαναλήψιμη διαδικασία, στην οποία οι μετασχηματισμοί ενσωματώνονται στον κώδικα και μπορούν να εφαρμοστούν με την ίδια λογική σε νέα δεδομένα.

1.3 Σκοπός και επιμέρους στόχοι

Κύριος σκοπός της διπλωματικής εργασίας είναι η ανάπτυξη μιας λειτουργικής, επεκτάσιμης και ελέγξιμης διαδικτυακής εφαρμογής, η οποία μετατρέπει ανομοιογενή αρχεία σε οργανωμένα δεδομένα και δημιουργεί τις προϋποθέσεις για την περαιτέρω ανάλυση και πρόβλεψή τους. Πρώτος επιμέρους στόχος είναι η δημιουργία οργανωμένου μηχανισμού μεταφόρτωσης, μέσω του οποίου κάθε αρχείο αποθηκεύεται και αποκτά μοναδική ταυτότητα. Δεύτερος στόχος είναι η αυτοματοποίηση του καθαρισμού και της τυποποίησης, ώστε η εφαρμογή να μπορεί να αντιμετωπίζει διαφοροποιήσεις στη θέση των επικεφαλίδων, στις ονομασίες των πεδίων, στις κωδικοποιήσεις χαρακτήρων και στις μορφές ημερομηνιών και αριθμητικών τιμών. Τρίτος στόχος είναι η ανάπτυξη γενικού μηχανισμού εφαρμογής κανόνων και mappings, ώστε οι επαναλαμβανόμενες επιχειρησιακές αποφάσεις να εφαρμόζονται με σταθερό τρόπο και, όπου προβλέπεται, να διατηρείται πληροφορία για τον κανόνα που οδήγησε στην τελική ταξινόμηση.

Επιπλέον, επιδιώκεται η διατήρηση της ιχνηλασιμότητας με σύνδεση της αρχικής γραμμής, της καθαρισμένης γραμμής και του αρχικού αρχείου, καθώς και η καταγραφή των βασικών σταδίων επεξεργασίας. Ένας ακόμη στόχος είναι η ανάπτυξη ανεξάρτητου μηχανισμού πρόβλεψης, στον οποίο διαχωρίζονται η προετοιμασία της χρονοσειράς, η εφαρμογή των μοντέλων, η ιστορική αξιολόγηση, η επιλογή στρατηγικής και η αποθήκευση των αποτελεσμάτων. Τέλος, επιδιώκεται η σύνδεση των τελικών δεδομένων με περιβάλλον επιχειρηματικής ευφυΐας, ώστε τα επεξεργασμένα στοιχεία και τα αποτελέσματα πρόβλεψης να μπορούν να χρησιμοποιηθούν για τη δημιουργία δεικτών και διαδραστικών αναφορών.

1.4 Μεθοδολογική προσέγγιση

Η εργασία ακολουθεί συνδυασμό βιβλιογραφικής, σχεδιαστικής, αναπτυξιακής και εμπειρικής προσέγγισης. Αρχικά εξετάζεται το θεωρητικό υπόβαθρο σχετικά με την ποιότητα και την προετοιμασία των δεδομένων, την επιχειρηματική ευφυΐα, τις χρονοσειρές, τις μεθόδους πρόβλεψης και τις τεχνικές αξιολόγησης. Στη συνέχεια προσδιορίζονται οι τεχνικές απαιτήσεις και σχεδιάζεται η αρχιτεκτονική της εφαρμογής με διαχωρισμό μεταξύ διεπαφής χρήστη, επιχειρησιακής λογικής, αποθήκευσης και αναλυτικών διαδικασιών. Η υλοποίηση πραγματοποιείται σταδιακά, ώστε η μεταφόρτωση, η πρόσβαση στη βάση, η ανάγνωση των αρχείων, η τυποποίηση, η εφαρμογή κανόνων και το forecasting να μπορούν να ελεγχθούν αρχικά ως διακριτές μονάδες και στη συνέχεια να συνδεθούν σε ενιαία ροή.

Στο αναλυτικό μέρος χρησιμοποιείται χρονικά συνεπής αξιολόγηση. Τα μοντέλα δεν αξιολογούνται πάνω στις ίδιες παρατηρήσεις στις οποίες εκπαιδεύτηκαν, αλλά σε μετα-

γενέστερες χρονικές περιόδους, με στόχο να αποφεύγεται η χρήση μελλοντικής πληροφορίας. Ως προς το είδος της, η παρούσα διπλωματική αποτελεί εφαρμοσμένη αναπτυξιακή εργασία με εμπειρικά στοιχεία: το αναπτυξιακό μέρος αφορά την υλοποίηση του πληροφοριακού συστήματος, ενώ το εμπειρικό μέρος αφορά τη λειτουργία του πάνω σε πραγματικά επιχειρησιακά δεδομένα και τη συγκριτική αξιολόγηση των αναλυτικών μεθόδων.

1.5 Δομή της εργασίας

Η διπλωματική εργασία οργανώνεται σε έξι κύρια κεφάλαια. Στο πρώτο κεφάλαιο παρουσιάζονται το αντικείμενο, η αναγκαιότητα, ο σκοπός και η μεθοδολογική προσέγγιση. Στο δεύτερο κεφάλαιο αναπτύσσεται το θεωρητικό υπόβαθρο σχετικά με την ποιότητα και τον καθαρισμό των δεδομένων, την επιχειρηματική ευφυΐα, τις χρονοσειρές, τις μεθόδους πρόβλεψης και τις τεχνικές αξιολόγησης. Το τρίτο κεφάλαιο επικεντρώνεται στην τεχνική υλοποίηση της εφαρμογής και περιγράφει την αρχιτεκτονική, το frontend, το backend, τις REST διεπαφές, τον μηχανισμό επεξεργασίας, το σχήμα αποθήκευσης και το υποσύστημα πρόβλεψης. Στο τέταρτο κεφάλαιο παρουσιάζεται η συγκεκριμένη εφαρμογή των παραπάνω μηχανισμών στην περίπτωση που εξετάζεται, με τους ειδικούς κανόνες, τις αντιστοιχίσεις και τη συγκεκριμένη διαμόρφωση των αναλυτικών διαδικασιών. Το πέμπτο κεφάλαιο αφορά τη σύνδεση με το Power BI και την παρουσίαση των αποτελεσμάτων, ενώ στο έκτο κεφάλαιο συνοψίζονται τα συμπεράσματα, οι περιορισμοί και οι δυνατότητες μελλοντικής επέκτασης.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

2.1 Από τα δεδομένα στην επιχειρησιακή γνώση

Η μετατροπή των δεδομένων από την αρχική, συχνά ανομοιογενή μορφή τους σε χρήσιμη γνώση δεν αποτελεί μία μεμονωμένη ενέργεια, αλλά μια ακολουθία αλληλεξαρτώνων σταδίων. Οι *Han, Kamber & Pei (2011)* περιγράφουν την διαδικασία ανακάλυψης της γνώσης από τα δεδομένα ως μια επαναληπτική ακολουθία επιλογής, μετασχηματισμού, εφαρμογής αναλυτικών μεθόδων και αξιολόγησης των αποτελεσμάτων. Η εξόρυξη των δεδομένων αποτελεί ένα από τα βασικά μέρη της διαδικασίας και τα αποτελέσματά της εξαρτώνται από την ποιότητα και την κατάλληλη προετοιμασία των δεδομένων.

Η προσέγγιση αυτή είναι ιδιαίτερα σημαντική για τους διάφορους οργανισμούς, γιατί τα δεδομένα που παράγονται σε καθημερινή βάση δεν έχουν δημιουργηθεί με αρχικό στόχο την ανάλυση. Οι εγγραφές αυτές προέρχονται από αρχεία συναλλαγών, λογιστικά συστήματα και πληροφοριακές εφαρμογές τα οποία χρησιμοποιούν συνήθως διαφορετική δομή, ονομασίες πεδίων ή τρόπο καταχώρισης. Για τον λόγο αυτό, πριν την αξιοποίηση τους χρειάζονται διαδικασίες καθαρισμού, ενοποίησης και μετασχηματισμοί για τον περιορισμό της πολυπλοκότητας και του όγκου. Οι διαδικασίες αυτές βοηθούν στην αποτελεσματικότητα και την ακρίβεια της ανάλυσης γιατί μία μετέπειτα απόφαση δεν μπορεί να βασιστεί σε ελλιπή ή ασυνεπή δεδομένα (*Han, Kamber & Pei, 2011*).

Στο πλαίσιο αυτής της εργασίας, η θεωρητική αυτή προσέγγιση οδηγεί στην ανάγκη δημιουργίας μίας ενιαίας διαδικασίας που συνδέει το αρχικό αρχείο με το τελικό αποτέλεσμα. Η διαδικασία δεν περιορίζεται μόνο στην παραγωγή και παρουσίαση ενός καθαρισμένου συνόλου αλλά διατηρεί την προέλευση των δεδομένων και τους μετασχηματισμούς που εφαρμόστηκαν. Με αυτόν τον τρόπο, η γνώση που παράγεται μπορεί να ελεγχθεί και να βελτιωθεί.

2.2 Ποιότητα των δεδομένων

Η ποιότητα των δεδομένων δεν εξαρτάται μόνο από την μαθηματική τους ακρίβεια αλλά συνδέεται και με τον τελικό σκοπό για τον οποίον θα χρησιμοποιηθούν. Οι *Batini & Scannapieco (2006)* σημειώνουν ότι εκτός από την ακρίβεια πρέπει να εξετάζονται καταστάσεις όπως η πληρότητα, η συνέπεια και η επικαιρότητα. Αντίστοιχα οι *Han, Kamber & Pei (2011)* αναφέρουν ότι η ποιότητα ισοδυναμεί με την καταλληλότητα των δεδομένων ως προς τον τελικό σκοπό της ανάλυσης και προσθέτουν έννοιες όπως η ευκολία ερμηνείας τους και η αξιοπιστία. Επομένως, ένα σύνολο δεδομένων μπορεί να περιέχει αριθμητικά ορθές τιμές αλλά να παραμένει ακατάλληλο για ανάλυση, εάν κρίσιμες πληροφορίες είναι ελλιπείς ή οι τιμές δεν μπορούν να ερμηνευθούν με συνέπεια.

Η ακρίβεια αφορά το πόσο καλά μπορεί να περιγράψει μία τιμή ένα πραγματικό γεγονός. Η λανθασμένη γραφή ενός κωδικού, μία λάθος ημερομηνία ή ένα ποσό σε λάθος πεδίο αποτελούν προβλήματα ακρίβειας. Η πληρότητα εκφράζει σε τι βαθμό είναι διαθέσιμη μια πληροφορία, μια εγγραφή μπορεί να έχει σωστά ποσά αλλά να μην μπορεί να χρησιμοποιηθεί όταν λείπει η ημερομηνία, ο κωδικός παραστατικού ή άλλο σημαντικό πεδίο. Βέβαια, ο έλεγχος της πληρότητας εξαρτάται και από τον σκοπό της ανάλυσης, γιατί ένα χρήσιμο πεδίο σε μία διαδικασία μπορεί να είναι προαιρετικό σε μία άλλη (*Batini & Scannapieco, 2006*).

Η συνέπεια σχετίζεται με την τήρηση κοινών λογικών και μορφολογικών κανόνων. Διαφορετικές μορφές ημερομηνιών, εναλλακτικά σύμβολα διαχωρισμού δεκαδικών ή χιλιάδων και διαφορετικές ονομασίες για τα ίδια πεδία μπορούν να οδηγήσουν σε λανθασμένες κατηγοριοποιήσεις και υπολογισμούς. Η επικαιρότητα, από την άλλη πλευρά, αφορά το κατά πόσο τα δεδομένα είναι ενημερωμένα και διαθέσιμα τη στιγμή που απαιτείται η ανάλυση. Ένα σύνολο δεδομένων μπορεί συνεπώς να είναι ορθό, αλλά να έχει περιορισμένη επιχειρησιακή αξία αν είναι διαθέσιμο μετά το χρονικό σημείο λήψης της απόφασης (*Batini & Scannapieco, 2006*).

Σημαντικό ρόλο έχουν επίσης η εγκυρότητα και η μοναδικότητα των δεδομένων. Η μοναδικότητα αφορά τον εντοπισμό περιπτώσεων που μία εγγραφή ή μια συναλλαγή μπορεί να έχει αποθηκευτεί παραπάνω από μία φορές. Όμως η απλή διαγραφή τέτοιων περιπτώσεων δεν είναι πάντα σωστή λύση, γιατί μπορεί να δείχνουν διαφορετικές λογιστικές κινήσεις. Τέλος, η εγκυρότητα αφορά την συμφωνία των τιμών με προκαθορισμένους κανόνες, όπως η λογική σχέση μεταξύ πεδίων και αποδεκτά εύρη. Όλοι οι παραπάνω έλεγχοι πρέπει να προηγούνται της τελικής παραγωγής αναφορών και προβλέψεων.

2.3 Καθαρισμός, τυποποίηση και ολοκλήρωση των δεδομένων

Ο καθαρισμός των δεδομένων αφορά τον εντοπισμό, την αξιολόγηση και εφόσον είναι δυνατό την διόρθωση προβληματικών εγγραφών. Σύμφωνα με τους *Han, Kamber και Pei (2011)*, οι διαδικασίες καθαρισμού χρησιμοποιούνται για την διαχείριση κενών τιμών, την αποκατάσταση ασυνεπειών και την κατηγοριοποίηση. Η διαδικασία αυτή δεν είναι μία απλή διαγραφή των τιμών που δεν συμφωνούν με τον σκοπό της ανάλυσης. Πολλές φορές χρειάζεται να γίνει διάκριση μεταξύ ενός πραγματικού σφάλματος, μίας επιτρεπτής εξαίρεσης και μίας ελλιπούς πληροφορίας.

Βασικό μέρος της διαδικασίας αυτής αποτελεί η τυποποίηση των τιμών και της δομής. Διαφορετικά αρχεία μπορεί να έχουν συνώνυμα, συντομογραφίες ή διαφορετική γραφή για το ίδιο πεδίο. Η αντιστοίχιση των ονομάτων αυτών σε ένα κοινό σχήμα επιτρέπει στον μηχανισμό να λειτουργεί ανεξάρτητα από την μορφή του αρχείου. Αντίστοιχη διαδικασία τυποποίησης πρέπει να γίνει στις ημερομηνίες και στις αριθμητικές τιμές. Τα οικονομικά στοιχεία μπορεί να αποθηκεύονται με διαχωριστικό δεκαδικών το κόμμα και διαχωριστικό χιλιάδων την τελεία ή αντίστροφο και οι ημερομηνίες μπορεί να είναι αποθηκευμένες ως κείμενο ή χρονικοί δείκτες. Η μετατροπή τους σε κοινό τύπο είναι βασική προϋπόθεση για την σωστή ταξινόμησή τους.

Μετά την τυποποίηση και τον καθαρισμό ακολουθεί η ολοκλήρωση των δεδομένων σε κοινό αναλυτικό πλαίσιο. Η ολοκλήρωση δεν σημαίνει υποχρεωτικά ότι όλες οι πηγές συγχωνεύονται σε έναν ενιαίο πίνακα. Μπορεί να επιτευχθεί μέσω κοινών αναγνωριστικών, mappings, σχέσεων και views, αρκεί οι πληροφορίες να αποκτούν σταθερή σημασιολογία και να μπορούν να αναλυθούν με συνεπή τρόπο. Ιδιαίτερη προσοχή απαιτείται όταν υπάρχουν επαναλαμβανόμενες εγγραφές, διαφορετικές εκδόσεις της ίδιας οντότητας ή αντικρουόμενες πληροφορίες. Για την αντιμετώπιση αυτών των περιπτώσεων χρησιμοποιούνται κανόνες αντιστοίχισης και ελέγχονται τα πιθανά διπλότυπα (*Han, Kamber & Pei, 2011*).

Ο καθαρισμός και η ολοκλήρωση των δεδομένων δεν αποτελούν αυστηρά γραμμική διαδικασία, καθώς ένας μετασχηματισμός ή ένας κανόνας μπορεί να αποκαλύψει νέα προβλήματα ή αποκλίσεις που απαιτούν επανέλεγχο. Πάνω σε αυτό το θέμα οι *Han, Kamber & Pei (2011)* περιγράφουν την διαδικασία καθαρισμού ως επαναληπτική, ενώ οι *Batini & Scannapieco (2006)* τονίζουν την σημασία εντοπισμού και διόρθωσης των σφαλμάτων ανάλογα με την διάσταση της ποιότητας που εξετάζεται. Στην παρούσα εργασία, η αποθήκευση τόσο της αρχικής μορφής όσο και της καθαρισμένης μορφής των δεδομένων μαζί με τους κανόνες επιτρέπει την άμεση σύγκριση πριν και μετά την επεξεργασία. Έτσι οι μετασχηματισμοί μπορούν να ελεγχθούν χωρίς να χάνεται η σύνδεση με την πρωτότυπη πηγή.

2.4 Επιχειρηματική ευφυΐα και αναλυτικές τεχνικές

Η επιχειρηματική ευφυΐα και η επιχειρηματική ανάλυση αξιοποιούν τα δεδομένα με στόχο την δημιουργία αξιοποιήσιμης γνώσης για την λήψη αποφάσεων. Η διαδικασία αυτή αποτελείται από την οργάνωση, την ανάλυση, την κατηγοριοποίηση και την παρουσίαση των αποτελεσμάτων μέσω αναφορών. Έτσι τα δεδομένα μετατρέπονται σε πληροφορία που μπορεί να χρησιμοποιηθεί λειτουργικά και στρατηγικά εντός ενός οργανισμού.

Σύμφωνα με την παρουσίαση του *Raisinghani (2025)*, οι αναλυτικές τεχνικές διακρίνονται σε τέσσερις κατηγορίες: την περιγραφική, την διαγνωστική, την προβλεπτική και την καθοδηγητική ανάλυση. Κάθε κατηγορία εξετάζει κάτι διαφορετικό από τα διαθέσιμα δεδομένα και απαντά σε διαφορετικό ερώτημα. Οι κατηγορίες αυτές δεν είναι ανεξάρτητες καθώς η αξιοπιστία της κάθε κατηγορίας στηρίζεται στην σωστή διαχείριση της προηγούμενης.

Η περιγραφική ανάλυση ασχολείται με τα ιστορικά δεδομένα με σκοπό την κατανόηση των γεγονότων και τάσεων που έχουν δημιουργηθεί. Απαντά κυρίως στο ερώτημα «τι συνέβη;» και περιλαμβάνει την σύνοψη και οπτικοποίηση των δεδομένων. Σε αυτή την κατηγορία παρουσιάζονται υπολογισμοί αθροισμάτων, μέσων όρων και μεταβολών ανάμεσα σε διαφορετικές χρονικές περιόδους. Μέσα από την περιγραφική ανάλυση παρουσιάζεται μία γενική εικόνα για την λειτουργία ενός οργανισμού.

Η διαγνωστική ανάλυση εξετάζει τους λόγους για τους οποίους παρατηρήθηκε ένα αποτέλεσμα ή μία μεταβολή. Στηρίζεται στην διερεύνηση συσχετίσεων και αιτιών ανάμεσα στα δεδομένα. Για παράδειγμα, μπορεί να χρησιμοποιηθεί για να εξεταστεί αν μια μεταβολή σχετίζεται με μία συγκεκριμένη χρονική περίοδο ή κατηγορία πελατών. Άρα η διαγνωστική ανάλυση απαντά στο ερώτημα «γιατί συνέβη;».

Η προβλεπτική ανάλυση χρησιμοποιεί ιστορικά δεδομένα για την εκτίμηση μελλοντικών τιμών ή καταστάσεων. Συνήθως βασίζεται σε στατιστικά μοντέλα, αλγόριθμους μηχανικής μάθησης ή συνδυασμό αυτών. Βασικός στόχος είναι η αναγνώριση σχέσεων που μπορούν να αξιοποιηθούν για την πρόβλεψη μελλοντικών τιμών, όπως πωλήσεων, ζήτησης ή συμπεριφοράς. Όπως έχει αναφερθεί και προηγουμένως η αξιοπιστία της πρόβλεψης στηρίζεται στην ποιότητα των δεδομένων από τα οποία προήλθε.

Τέλος, η καθοδηγητική ανάλυση αξιοποιεί όλα τα δεδομένα και τα αποτελέσματα των προηγούμενων σταδίων για να προτείνει ενέργειες ή στρατηγικές. Δεν μένει μόνο στην πρόβλεψη του τι μπορεί να γίνει αλλά εξετάζει και ποια απόφαση μπορεί να οδηγήσει στο επιθυμητό αποτέλεσμα. Άρα η καθοδηγητική ανάλυση απαντά στο ερώτημα «τι πρέπει να γίνει;» και συνδέει την ανάλυση των δεδομένων με την διαδικασία λήψης αποφάσεων.

2.5 Χρονοσειρές και πρόβλεψη

Μια χρονοσειρά αποτελεί ακολουθία παρατηρήσεων που καταγράφονται σε διαδοχικές χρονικές στιγμές. Σε αντίθεση με ένα απλό σύνολο δεδομένων, η χρονική σειρά των παρατηρήσεων είναι ουσιώδης, γιατί οι προηγούμενες τιμές μπορεί να περιέχουν πληροφορία για τις επόμενες. Οι μέθοδοι πρόβλεψης πολλές φορές εκτός από το ιστορικό της μεταβλητής χρησιμοποιούν και εξωτερικούς παράγοντες που μπορεί να επηρεάσουν την τιμή και την συμπεριφορά της μεταβλητής.

Στις χρονοσειρές υπάρχουν στοιχεία τάσης, εποχικότητας και τυχαίας διακύμανσης. Η τάση περιγράφει την κατεύθυνση των τιμών, ενώ η εποχικότητα αφορά μοτίβα που επαναλαμβάνονται σε σταθερά χρονικά διαστήματα. Η μελέτη των παραπάνω χαρακτηριστικών κρίνεται ιδιαίτερα σημαντική για την κατανόηση της σειράς και την επιλογή της μεθόδου πρόβλεψης και χρησιμοποιείται ως στάδιο της διαδικασίας πρόβλεψης.

Η επιλογή μίας μεθόδου πρόβλεψης εξαρτάται από το μήκος της χρονοσειράς, την εποχικότητα, την τάση και την μεταβολή από εξωτερικούς παράγοντες. Δεν υπάρχει μία μέθοδος που είναι σταθερά ανώτερη από τις άλλες, για τον λόγο αυτό προτείνεται η χρήση benchmarks και η σύγκρισή τους με πιο σύνθετα μοντέλα πάνω σε δεδομένα που δεν χρησιμοποιήθηκαν κατά την διαδικασία της εκπαίδευσης. Η τελική επιλογή του μοντέλου πρέπει να βασίζεται στην προβλεπτική απόδοση και όχι μόνο στην πολυπλοκότητα του μοντέλου.

2.5.1 Seasonal Naive

Η μέθοδος Seasonal Naive χρησιμοποιεί ως πρόβλεψη την τιμή της αντίστοιχης χρονικής περιόδου του προηγούμενου κύκλου. Για μηνιαία δεδομένα, η πρόβλεψη ενός μήνα ισούται με την παρατηρηθείσα τιμή του ίδιου μήνα του προηγούμενου έτους. Παρότι η μέθοδος είναι πολύ απλή, αποτελεί σημείο αναφοράς. Ένα πιο σύνθετο μοντέλο πρόβλεψης δεν θεωρείται χρήσιμο επειδή διαθέτει παραπάνω παραμέτρους αλλά θα πρέπει να παρουσιάζει και μετρήσιμη βελτίωση σε σχέση με ένα απλό benchmark.

Η βασική παράμετρος της μεθόδου είναι η περίοδος εποχικότητας m , η οποία καθορίζει πόσες χρονικές περιόδους πίσω θα αναζητηθεί η αντίστοιχη παρατήρηση. Για μηνιαία δεδομένα με ετήσια εποχικότητα χρησιμοποιείται συνήθως $m=12$. Η μέθοδος δεν διαθέτει παραμέτρους εκπαίδευσης, γεγονός που την καθιστά απλή και κατάλληλη ως benchmark.

2.5.2 Σταθμισμένος εποχικός μέσος

Ο σταθμισμένος εποχικός μέσος προχωράει την προηγούμενη λογική ένα βήμα παραπάνω, χρησιμοποιώντας περισσότερο από έναν ιστορικούς κύκλους. Οι πιο πρόσφατες περίοδοι λαμβάνουν μεγαλύτερο βάρος έτσι ώστε η πρόβλεψη να ανταποκρίνεται στα

νεότερα δεδομένα. Το μοντέλο χαρακτηρίζεται από υψηλή ερμηνευσιμότητα, γιατί είναι εύκολο να προσδιοριστεί ποια έτη χρησιμοποιήθηκαν στην τελική τιμή. Όμως η απόδοση του μπορεί να περιοριστεί όταν υπάρχει μεγάλη μεταβολή (θετική ή αρνητική) που δεν έχει εμφανιστεί ξανά στο δείγμα.

Οι βασικές παράμετροι της μεθόδου είναι η περίοδος εποχικότητας m , ο αριθμός των ιστορικών εποχικών κύκλων που χρησιμοποιούνται και τα βάρη που αποδίδονται στις αντίστοιχες παλαιότερες παρατηρήσεις. Μεγαλύτερο βάρος στις πιο πρόσφατες τιμές κάνει την πρόβλεψη πιο ευαίσθητη στις νεότερες μεταβολές, ενώ πιο ομοιόμορφα βάρη αυξάνουν τη συμμετοχή του παλαιότερου ιστορικού.

2.5.3 ETS (Error, Trend, Seasonality)

Τα μοντέλα ETS περιγράφουν την χρονοσειρά χρησιμοποιώντας συνιστώσες για την τάση, την εποχικότητα και το σφάλμα. Δηλαδή εξετάζουν το επίπεδο της σειράς, την κατεύθυνση που ακολουθεί και μοτίβα που πιθανόν να επαναλαμβάνονται.

Η εκθετική εξομάλυνση θα δώσει συνήθως μεγαλύτερη σημασία στις πιο πρόσφατες παρατηρήσεις ενώ οι παλιότερες επηρεάζουν όλο και λιγότερο την πρόβλεψη. Ανάλογα με την συμπεριφορά των δεδομένων, η τάση και η εποχικότητα μπορούν να έχουν προσθετικό ή πολλαπλασιαστικό χαρακτήρα. Η προσθετική μορφή είναι κατάλληλη όταν οι μεταβολές είναι περίπου ίδιες σε όλη την διάρκεια της σειράς, ενώ όταν οι μεταβολές γίνονται μεγαλύτερες όσο αυξάνεται το επίπεδο της σειράς είναι καλύτερη επιλογή η πολλαπλασιαστική μορφή. Με απλά λόγια, η προσθετική εποχικότητα εκφράζεται ως μια σταθερή διαφορά, ενώ η πολλαπλασιαστική ως ένα ποσοστό.

Οι βασικές παράμετροι εξομάλυνσης είναι η α για το επίπεδο της σειράς, η β για την τάση και η γ για την εποχικότητα. Όταν χρησιμοποιείται εξασθενημένη τάση μπορεί να προστεθεί και η παράμετρος ϕ , η οποία ελέγχει πόσο γρήγορα μειώνεται η επίδραση της τάσης στις μελλοντικές προβλέψεις. Επιπλέον ορίζεται η εποχική περίοδος m και η προσθετική ή πολλαπλασιαστική μορφή των συνιστωσών. Μεγαλύτερες τιμές των παραμέτρων εξομάλυνσης δίνουν γενικά μεγαλύτερη βαρύτητα στις πιο πρόσφατες παρατηρήσεις της αντίστοιχης συνιστώσας.

2.5.4 ARIMA και SARIMA

Το μοντέλο ARIMA χρησιμοποιείται για πρόβλεψη χρονοσειρών, αξιοποιώντας τις ιστορικές τιμές και τα προηγούμενα σφάλματα πρόβλεψης. Αποτελείται από τις παραμέτρους p, d και q . Η παράμετρος p δείχνει πόσες προηγούμενες τιμές λαμβάνονται υπόψη, η d πόσες φορές διαφοροποιείται η σειρά ώστε να γίνει σταθερή, ενώ η q δείχνει πόσα προηγούμενα σφάλματα χρησιμοποιήθηκαν στην πρόβλεψη.

Το SARIMA αποτελεί επέκταση του προηγούμενου μοντέλου για σειρές που παρουσιάζουν εποχικότητα. Για τον λόγο αυτό εκτός από τις p, d, q υπάρχουν οι εποχικές παράμετροι P, D, Q και η παράμετρος m που εκφράζει την διάρκεια του εποχικού κύκλου (π.χ. $m=12$ για μηνιαία δεδομένα με ετήσια εποχικότητα).

Η διαφοροποίηση d χρησιμοποιείται για την σταθεροποίηση του μέσου επιπέδου της σειράς ενώ το p εκφράζει την εξάρτηση από τις προηγούμενες τιμές. Τέλος το q δείχνει την εξάρτηση από τα προηγούμενα σφάλματα πρόβλεψης. Οι επιπλέον παράμετροι του SARIMA δείχνουν τα ίδια αλλά για το εποχικό μέρος της σειράς.

Η επιλογή των κατάλληλων παραμέτρων μπορεί να γίνει είτε με πληροφοριακά κριτήρια είτε με αυτοματοποιημένες διαδικασίες και η τελική αξιολόγηση όπως αναφέραμε και προηγουμένως γίνεται σε δείγμα που δεν λαμβάνει μέρος στην εκπαίδευση.

2.5.5 Παλινδρόμηση με εποχικά χαρακτηριστικά

Ένα μοντέλο παλινδρόμησης μπορεί να χρησιμοποιήσει τον χρόνο ως μία ανεξάρτητη μεταβλητή, ώστε να δείξει την πορεία (ανοδική ή καθοδική) της χρονοσειράς. Παράλληλα, μπορούν να προστεθούν ψευδομεταβλητές για τους μήνες ή τις εποχές του έτους. Έτσι το μοντέλο εκτιμά ξεχωριστά την τάση και την εποχική επίδραση κάθε περιόδου.

Η παλινδρόμηση Ridge μπορεί να προσθέσει περιορισμό στο μέγεθος των συντελεστών του μοντέλου. Έτσι αποφεύγεται το ενδεχόμενο συγκεκριμένοι συντελεστές να πάρουν μεγάλες τιμές, ιδιαίτερα όταν υπάρχουν μεταβλητές με έντονες συσχετίσεις μεταξύ τους. Η μέθοδος αυτή δίνει σταθερότητα και μπορεί να συνεισφέρει στην βελτίωση της απόδοσης πάνω σε νέα δεδομένα. Όμως, επειδή στηρίζεται σε σχέσεις που είναι γραμμικές συνηθίζεται να μην μπορεί να περιγράψει σωστά απότομες αλλαγές ή μη γραμμική μεταβολή.

Βασική παράμετρος της Ridge είναι ο συντελεστής κανονικοποίησης α , ο οποίος ελέγχει την ένταση του περιορισμού στους συντελεστές του μοντέλου. Μικρή τιμή της α αφήνει το μοντέλο πιο κοντά στην απλή γραμμική παλινδρόμηση, ενώ μεγαλύτερη τιμή προκαλεί ισχυρότερη συρρίκνωση των συντελεστών. Στην πρόβλεψη χρονοσειρών μπορούν παράλληλα να χρησιμοποιηθούν ως χαρακτηριστικά η χρονική τάση και εποχικές ψευδομεταβλητές, ώστε να αποτυπώνονται οι διαφορές μεταξύ των περιόδων του έτους.

2.5.6 Νευρωνικά δίκτυα και LSTM

Τα νευρωνικά δίκτυα μπορούν να προσεγγίσουν σύνθετες και μη γραμμικές σχέσεις στα δεδομένα. Για ακολουθιακά προβλήματα χρησιμοποιούνται ειδικές αρχιτεκτονικές επαναλαμβανόμενων νευρωνικών δικτύων, στις οποίες η πληροφορία από προηγούμενα χρονικά βήματα μπορεί να επηρεάζει την επεξεργασία των επόμενων. Η ιδιότητα αυτή

είναι ιδιαίτερα χρήσιμη σε προβλήματα χρονοσειρών, όπου η χρονική διάταξη των παρατηρήσεων αποτελεί μέρος της πληροφορίας.

Τα συμβατικά επαναλαμβανόμενα νευρωνικά δίκτυα μπορεί να παρουσιάσουν δυσκολίες κατά την εκμάθηση εξαρτήσεων που βρίσκονται σε μεγάλα χρονικά διαστήματα. Η αρχιτεκτονική LSTM χρησιμοποιεί ειδικές μονάδες μνήμης και μηχανισμούς ροής της πληροφορίας, ώστε να καθορίζει ποια στοιχεία θα διατηρηθούν και ποια θα απορριφθούν. Η λειτουργία αυτή καθιστά τα LSTM κατάλληλα για την ανάλυση χρονοσειρών και την αναγνώριση μοτίβων και εξαρτήσεων.

Η χρήση ενός μοντέλου LSTM σε περιορισμένο σύνολο δεδομένων απαιτεί προσεκτική αξιολόγηση. Λόγω του μεγάλου αριθμού παραμέτρων και της πολυπλοκότητας του μοντέλου, υπάρχει κίνδυνος υπερβολικής προσαρμογής στα δεδομένα εκπαίδευσης (overfitting) και χαμηλότερης απόδοσης σε νέες χρονικές περιόδους. Για τον λόγο αυτό είναι απαραίτητη η χρήση κατάλληλης κανονικοποίησης, η αξιολόγηση σε δεδομένα που δεν συμμετείχαν στην εκπαίδευση και η σύγκριση με πιο απλές μεθόδους πρόβλεψης.

Οι βασικές υπερπαραμέτροι ενός LSTM είναι το μήκος του ιστορικού παραθύρου (lookback), ο αριθμός των μονάδων μνήμης (units), ο ρυθμός μάθησης, το batch size, το dropout και ο αριθμός των epochs. Μεγαλύτερος αριθμός μονάδων αυξάνει την ικανότητα του δικτύου να αναπαριστά σύνθετες σχέσεις, αλλά αυξάνει και τον κίνδυνο υπερπροσαρμογής. Ο ρυθμός μάθησης καθορίζει το μέγεθος των βημάτων κατά τη βελτιστοποίηση, το batch size πόσες ακολουθίες χρησιμοποιούνται πριν από κάθε ενημέρωση των βαρών και το dropout λειτουργεί ως μέσο κανονικοποίησης. Τέλος, η παράμετρος patience επιτρέπει την πρόωρη διακοπή όταν η απόδοση στο σύνολο επικύρωσης παύει να βελτιώνεται.

2.6 Αξιολόγηση των μοντέλων πρόβλεψης

Η ακρίβεια και η σταθερότητα ενός μοντέλου πρόβλεψης δεν πρέπει να αξιολογούνται αποκλειστικά στα δεδομένα που χρησιμοποιήθηκαν για την εκπαίδευσή του. Η προβλεπτική του ικανότητα εκτιμάται σε μεταγενέστερες παρατηρήσεις που δεν συμμετείχαν στη συγκεκριμένη προσαρμογή του μοντέλου. Για τον λόγο αυτό η αξιολόγηση μιας χρονοσειράς πρέπει να διατηρεί τη χρονική σειρά των παρατηρήσεων: το σύνολο εκπαίδευσης προηγείται χρονικά και το σύνολο ελέγχου περιλαμβάνει μεταγενέστερες περιόδους.

2.6.1 Μέσο απόλυτο σφάλμα (MAE - Mean Absolute Error)

Το μέσο απόλυτο σφάλμα ορίζεται ως:

$$MAE = \frac{1}{n} \sum_{t=1}^n |y_t - \hat{y}_t|$$

και υπολογίζει τη μέση απόλυτη απόσταση μεταξύ των προβλεπόμενων και των πραγματικών τιμών. Επειδή χρησιμοποιείται η απόλυτη τιμή, δεν έχει σημασία αν το μοντέλο θα προβλέψει περισσότερο ή λιγότερο. Για παράδειγμα μία απόκλιση +3 και μια απόκλιση -3 συνεισφέρουν εξίσου στο αποτέλεσμα. Εκφράζεται στις ίδιες μονάδες με την μεταβλητή, οπότε έχει και άμεση ερμηνεία, δηλαδή $MAE = 3$ σημαίνει ότι οι προβλέψεις αποκλίνουν κατά μέσο όρο 3 μονάδες.

2.6.2 Ρίζα Μέσου Τετραγωνικού Σφάλματος (RMSE)

Η ρίζα του μέσου τετραγωνικού σφάλματος ορίζεται ως:

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2}$$

Το RMSE μετράει πάλι την συνολική απόκλιση των προβλέψεων, αλλά τα σφάλματα υψώνονται στο τετράγωνο και έτσι οι μεγάλες αποκλίσεις αποκτούν μεγαλύτερη βαρύτητα. Για παράδειγμα η απόκλιση 10 μονάδων επηρεάζει πολύ περισσότερο το αποτέλεσμα από μια απόκλιση 5 μονάδων. Για τον λόγο αυτό το RMSE είναι ιδιαίτερα ευαίσθητο σε μεγάλες αστοχίες και λειτουργεί συμπληρωματικά προς το MAE. Όπως και το MAE, εκφράζεται στις ίδιες μονάδες με τη μεταβλητή.

2.6.3 Συμμετρικό Μέσο Απόλυτο Ποσοστιαίο Σφάλμα (sMAPE)

Το συμμετρικό μέσο απόλυτο ποσοστιαίο σφάλμα ορίζεται ως:

$$sMAPE = \frac{100}{n} \sum_{t=1}^n \frac{2 |y_t - \hat{y}_t|}{|y_t| + |\hat{y}_t|}$$

Το sMAPE παρουσιάζει το μέγεθος του σφάλματος σε ποσοστιαία μορφή, λαμβάνοντας υπόψη τόσο την πραγματική τιμή όσο και την προβλεπόμενη. Αυτό επιτρέπει τη σύγκριση των μοντέλων όπου οι μεταβλητές τους είναι διαφορετικής κλίμακας. Μικρότερη τιμή σημαίνει μεγαλύτερη ακρίβεια, ενώ μία τιμή κοντά στο μηδέν υποδηλώνει σχεδόν τέλειες προβλέψεις.

2.6.4 Μεροληψία (Bias)

Η μεροληψία ορίζεται ως :

$$Bias = \frac{1}{n} \sum_{t=1}^n (\hat{y}_t - y_t)$$

Το bias δείχνει την μέση κατεύθυνση των σφαλμάτων και όχι το συνολικό τους μέγεθος. Θετική τιμή σημαίνει ότι το μοντέλο δίνει μεγαλύτερες προβλέψεις από τις πραγματικές τιμές, δηλαδή υπερεκτιμά. Αντίθετα, αρνητική τιμή δείχνει υποεκτίμηση από το μοντέλο. Τιμή κοντά στο μηδέν είναι επιθυμητή αλλά δεν εγγυάται ότι οι προβλέψεις είναι ακριβείς επειδή τα θετικά και τα αρνητικά σφάλματα μπορεί να εξουδετερώνονται. Για αυτό τον λόγο θα πρέπει να εξετάζεται σε συνδυασμό με το MAE ή RMSE.

2.6.5 Rolling και expanding-window backtesting

Ο απλός διαχωρισμός των δεδομένων σε ένα σύνολο εκπαίδευσης και ένα σύνολο ελέγχου μπορεί να οδηγήσει σε συμπεράσματα που εξαρτώνται υπερβολικά από μία συγκεκριμένη χρονική περίοδο. Για πιο αξιόπιστη αξιολόγηση χρησιμοποιείται rolling-origin backtesting, δηλαδή επαναλαμβανόμενη αξιολόγηση σε διαδοχικά χρονικά σημεία.

Σε κάθε επανάληψη ο χρονικός άξονας μετακινείται προς τα εμπρός, το μοντέλο εκπαιδεύεται μόνο στις παρατηρήσεις που θα ήταν διαθέσιμες μέχρι το συγκεκριμένο χρονικό σημείο και στη συνέχεια αξιολογείται σε μεταγενέστερη περίοδο. Με αυτόν τον τρόπο προσεγγίζεται καλύτερα η πραγματική συνθήκη πρόβλεψης και αποφεύγεται η διαρροή μελλοντικής πληροφορίας.

Όταν το training set διατηρεί την ίδια αρχική παρατήρηση και αυξάνεται σταδιακά με τις νεότερες διαθέσιμες τιμές, η στρατηγική ονομάζεται expanding-window backtesting. Αντίθετα, σε ένα rolling window σταθερού μήκους απομακρύνονται σταδιακά οι παλαιότερες παρατηρήσεις. Στην παρούσα εργασία η βασική λογική αξιολόγησης είναι rolling-origin με expanding training window. Για παράδειγμα το μοντέλο αρχικά θα εκπαιδευτεί στις παρατηρήσεις $1, \dots, t$ για να προβλέψει την παρατήρηση $t + 1$. Στην επόμενη επανάληψη θα εκπαιδευτεί στις παρατηρήσεις $1, \dots, t, t + 1$ για να προβλέψει την παρατήρηση $t + 2$ και η ίδια διαδικασία θα συνεχίσει έως ότου χρησιμοποιηθούν όλες οι διαθέσιμες περίοδοι.

Η λογική αυτή μπορεί να αποδοθεί συνοπτικά ως:

$$1\text{η επανάληψη: } \{y_1, \dots, y_t\} \longrightarrow y_{t+1},$$

$$2\text{η επανάληψη: } \{y_1, \dots, y_t, y_{t+1}\} \longrightarrow y_{t+2},$$

$$k\text{-οστή επανάληψη: } \{y_1, \dots, y_{t+k-1}\} \longrightarrow y_{t+k}.$$

Η επαναλαμβανόμενη αξιολόγηση χρησιμοποιώντας διαφορετικά χρονικά σημεία δίνει καλύτερη εικόνα για την σταθερότητα και την ακρίβεια της πρόβλεψης και παράλληλα δίνει την δυνατότητα να αξιολογηθεί αν η απόδοση του μοντέλου πρόβλεψης παραμένει σταθερή στον χρόνο ή αν επηρεάζεται από αλλαγές στην χρονοσειρά.

2.7 Εφαρμογές πρόβλεψης στον ξενοδοχειακό και τουριστικό τομέα

Η πρόβλεψη των αφίξεων και της ζήτησης αποτελεί βασικό στάδιο για την διαχείριση εσόδων στον ξενοδοχειακό τομέα. Οι *Weatherford & Kimes (2003)* συνέκριναν διαφορετικές μεθόδους πρόβλεψης χρησιμοποιώντας πραγματικά δεδομένα ξενοδοχείων. Τα αποτελέσματα έδειξαν ότι οι μέθοδοι pickup, παλινδρόμησης, εκθετικής εξομάλυνσης και κινητού μέσου μπορούσαν να δώσουν ικανοποιητικά αποτελέσματα με σταθερή συμπεριφορά. Το αποτέλεσμα αυτό δείχνει ότι ακόμα και απλούστερες μέθοδοι δεν θα πρέπει να παραλείπονται κατά την έρευνα.

Η σύγκριση των διαθέσιμων τεχνικών δείχνει ότι δεν υπάρχει μια τεχνική πρόβλεψης που να υπερέχει σε κάθε περίπτωση για όλα τα δεδομένα. Η αποτελεσματικότητα της μεθόδου επηρεάζεται πολλές φορές από τα χαρακτηριστικά των δεδομένων, το επίπεδο της ανάλυσης και τον χρονικό ορίζοντα της πρόβλεψης. Για τον λόγο αυτό, η επιλογή του κατάλληλου μοντέλου θα πρέπει να στηρίζεται στη συγκριτική αξιολόγηση διαφορετικών μεθόδων πάνω στα ίδια δεδομένα (*Weatherford & Kimes, 2003*). Παράλληλα, η σύγχρονη έρευνα έχει επεκταθεί από τις στατιστικές μεθόδους προς την μηχανική μάθηση και τα νευρωνικά δίκτυα, τα οποία μπορούν να αξιοποιήσουν παραπάνω μεταβλητές και να αναγνωρίζουν μη γραμμικές χρονικές μεταβολές (*Dowlut & Gobin-Rahimbux, 2023*).

Σημαντική πηγή πρόβλεψης αποτελεί και η πληροφορία των προγενέστερων κρατήσεων (booking on the books), οι κρατήσεις δηλαδή που είναι ήδη καταχωρημένες πριν από την ημερομηνία άφιξης. Οι συγκεκριμένες κρατήσεις μπορούν να δείξουν μέρος της μελλοντικής ζήτησης, ωστόσο δεν αποτελούν την τελική ζήτηση γιατί μέχρι την ημερομηνία άφιξης μπορεί να έχουν προκύψει ακυρώσεις ή νέες κρατήσεις.

Οι παραπάνω πληροφορίες μπορούν να χρησιμοποιηθούν τόσο από παραδοσιακές μεθόδους πρόβλεψης όσο και από πιο σύνθετα μοντέλα που αξιοποιούν περισσότερες παραμέτρους και μπορούν να αναγνωρίσουν χρονικά μοτίβα. Στο πλαίσιο αυτό, τεχνικές βαθιάς μάθησης έχουν χρησιμοποιηθεί και σε προβλήματα πρόβλεψης του ξενοδοχειακού και τουριστικού τομέα. Τα LSTM αποτελούν μία από τις αρχιτεκτονικές που μπορούν να εφαρμοστούν σε χρονοσειρές, επειδή έχουν σχεδιαστεί ώστε να διατηρούν πληροφορία από προηγούμενα χρονικά βήματα και να μοντελοποιούν χρονικές εξαρτήσεις. Ωστόσο, η χρήση ενός πιο σύνθετου μοντέλου δεν εγγυάται μεγαλύτερη ακρίβεια. Η απόδοση εξαρτάται από την ποιότητα και το μέγεθος των δεδομένων, την κατάλληλη επιλογή των παραμέτρων και την αξιολόγηση σε χρονικές περιόδους που δεν συμμετείχαν στην εκπαίδευση (*Dowlut & Gobin-Rahimbux, 2023*).

Κεφάλαιο 3

Υλοποίηση της Εφαρμογής

3.1 Συνολική αρχιτεκτονική και ροή λειτουργίας

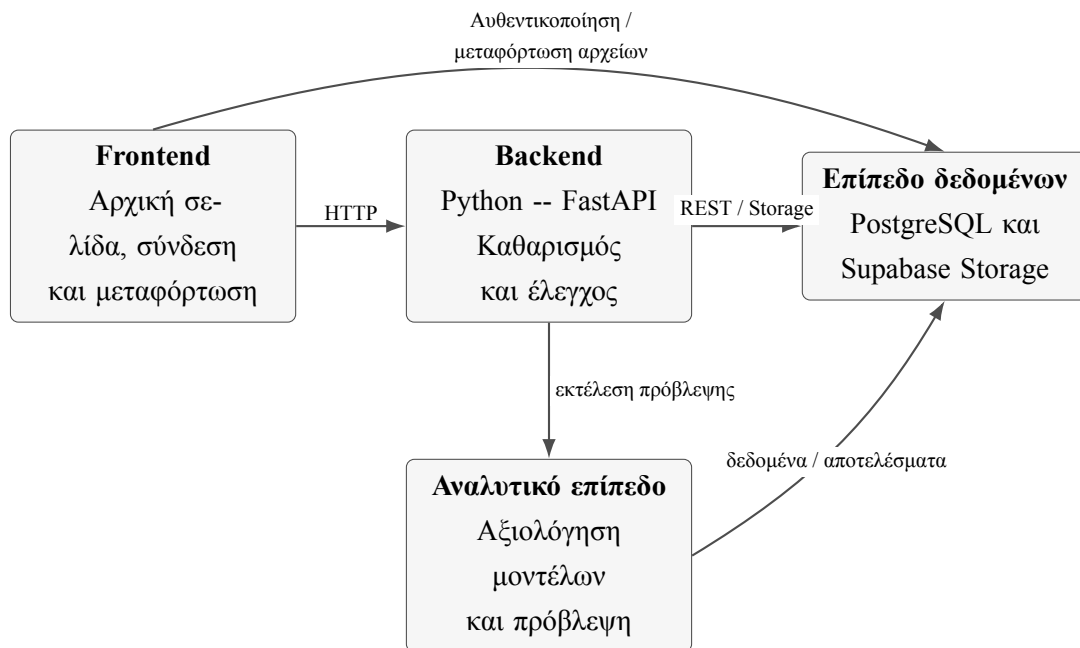
Η εφαρμογή σχεδιάστηκε και υλοποιήθηκε ως ολοκληρωμένο διαδικτυακό πληροφοριακό σύστημα και όχι ως ένα μεμονωμένο πρόγραμμα που εκτελεί έναν απλό καθαρισμό σε ένα συγκεκριμένο αρχείο. Η διάκριση αυτή είναι σημαντική, επειδή ένα πραγματικό σύστημα πρέπει να διαχειρίζεται όχι μόνο τον μετασχηματισμό των δεδομένων, αλλά και την είσοδο του χρήστη, την αυθεντικοποίηση, την επιλογή και αποθήκευση των αρχείων, την κατάσταση της εκτέλεσης, την επικοινωνία μεταξύ διαφορετικών υπηρεσιών, την καταγραφή των σφαλμάτων και τη μόνιμη αποθήκευση των αποτελεσμάτων. Για τον λόγο αυτό η εφαρμογή οργανώθηκε σε επίπεδα, τα οποία συνεργάζονται μέσω καθορισμένων σημείων επικοινωνίας.

Το πρώτο επίπεδο είναι το frontend. Έχει υλοποιηθεί με Next.js, React και TypeScript και είναι υπεύθυνο για την άμεση αλληλεπίδραση με τον χρήστη: την αρχική σελίδα, την διαδικασία σύνδεσης, την επιλογή της επιθυμητής κατηγορίας, την επιλογή των αρχείων και την παρουσίαση της κατάστασης της μεταφόρτωσης και της επεξεργασίας. Το δεύτερο επίπεδο είναι το backend, το οποίο έχει υλοποιηθεί σε Python με FastAPI. Εκεί βρίσκεται η κύρια λογική ανάκτησης των αρχείων, ο έλεγχος της δομής τους, η επιλογή του κατάλληλου μηχανισμού καθαρισμού και η αποθήκευση του αποτελέσματος. Το τρίτο επίπεδο είναι το επίπεδο δεδομένων και αποτελείται από τη PostgreSQL βάση και το Supabase Storage. Η βάση αποθηκεύει οργανωμένες εγγραφές, μεταδεδομένα, καταστάσεις και σχέσεις, ενώ το Storage διατηρεί τα πραγματικά αρχεία. Τέλος, το τέταρτο επίπεδο περιλαμβάνει τις αναλυτικές διαδικασίες και το σύστημα πρόβλεψης, το οποίο λειτουργεί πάνω στα δεδομένα που έχουν ήδη επεξεργαστεί από τα προηγούμενα στάδια.

Το frontend επικοινωνεί με το Supabase για την αυθεντικοποίηση, τις αναγνώσεις και εγγραφές που επιτρέπονται από το περιβάλλον του χρήστη και την μεταφόρτωση του αρχείου, και με το Python backend μέσω αιτημάτων HTTP. Το Python backend επικοινωνεί

νεί με την σειρά του με το Supabase μέσω των διεπαφών REST χρησιμοποιώντας διαπιστευτήρια. Δημιουργείται έτσι μία σαφής αλυσίδα: ο browser ξεκινά τη διαδικασία, το Supabase διατηρεί τη μόνιμη κατάσταση και τα αρχεία, ενώ το backend εκτελεί τις διαδικασίες.

Σε επίπεδο κώδικα, το frontend οργανώνεται στον κατάλογο *app/*. Η αρχική σελίδα βρίσκεται στο *app/page.tsx*, η σελίδα σύνδεσης στο *app/login/page.tsx* και η κύρια σελίδα μεταφόρτωσης στο *app/upload/page.tsx*. Η σύνδεση με το Supabase δημιουργείται κεντρικά στο *lib/supabaseClient.ts*. Στο backend, το *main.py* αποτελεί το βασικό σημείο εκκίνησης της εφαρμογής FastAPI και τον κεντρικό συντονιστή της επεξεργασίας, ενώ οι περισσότερο εξειδικευμένοι μηχανισμοί έχουν μεταφερθεί σε ανεξάρτητες μονάδες κώδικα, όπως τα *income_cleaner.py*, *expenses_cleaner.py* και ο κατάλογος *forecasting/*. Ο συγκεκριμένος διαχωρισμός περιορίζει τη σύγχυση μεταξύ των τμημάτων και επιτρέπει την αλλαγή ενός μηχανισμού χωρίς τον επανασχεδιασμό ολόκληρης της εφαρμογής.



Σχήμα 3.1: Συνολική αρχιτεκτονική και διαδρομές επικοινωνίας της εφαρμογής.

Η παραπάνω αρχιτεκτονική διατηρεί τα όρια μεταξύ παρουσίασης, επεξεργασίας και αποθήκευσης. Το frontend δεν περιέχει τους κανόνες καθαρισμού, ενώ το backend δεν διαχειρίζεται την εμφάνιση της εφαρμογής. Αντίστοιχα, το Storage δεν χρησιμοποιείται ως υποκατάστατο της βάσης και η βάση δεν χρησιμοποιείται για την αποθήκευση. Ο διαχωρισμός αυτός κάνει τη ροή πιο διαχειρίσιμη και δημιουργεί τις προϋποθέσεις ώστε το σύστημα να επεκταθεί σε νέους τύπους επεξεργασίας χωρίς να αλλάξει η βασική του αρχιτεκτονική.

3.2 Frontend, αρχική σελίδα και έλεγχος πρόσβασης

Η σύνδεση του frontend με το Supabase δημιουργείται στο αρχείο *lib/supabaseClient.ts*. Στο αρχείο αυτό χρησιμοποιείται η *createClient()* της βιβλιοθήκης *@supabase/supabase-js*. Η διεύθυνση του Supabase και το δημόσιο κλειδί πρόσβασης δεν έχουν ενσωματωθεί ως σταθερές τιμές στον κώδικα, αλλά ανακτώνται από μεταβλητές περιβάλλοντος. Η πρακτική αυτή επιτρέπει στην ίδια βάση κώδικα να χρησιμοποιείται σε διαφορετικά περιβάλλοντα, όπως στην τοπική ανάπτυξη και στην παραγωγική λειτουργία, χωρίς να απαιτείται αλλαγή των αρχείων του κώδικα.

```
const supabaseUrl =
  process.env.NEXT_PUBLIC_SUPABASE_URL!;

const supabaseAnonKey =
  process.env.NEXT_PUBLIC_SUPABASE_ANON_KEY!;

export const supabase =
  createClient(supabaseUrl, supabaseAnonKey);
```

Η αρχική σελίδα της εφαρμογής δημιουργείται στο *app/page.tsx*. Ο ρόλος της είναι περιορισμένος: παρουσιάζει το σημείο εισόδου στην εφαρμογή και οδηγεί τον χρήστη στη διαδικασία σύνδεσης. Δεν πραγματοποιεί αναγνώσεις δεδομένων, δεν αποθηκεύει αρχεία και δεν εκτελεί διαδικασίες καθαρισμού. Ο διαχωρισμός αυτός επιτρέπει η πρώτη σελίδα να λειτουργεί ως μια απλή πύλη προς τις λειτουργίες.

Η διαδικασία σύνδεσης υλοποιείται στο *app/login/page.tsx*. Η σελίδα είναι στοιχείο React που εκτελείται στην πλευρά του χρήστη και χρησιμοποιεί το *useState* για να διατηρεί προσωρινά το email, τον κωδικό, την κατάσταση της προσπάθειας σύνδεσης και το μήνυμα που εμφανίζεται στον χρήστη. Με την ενεργοποίηση της συνάρτησης *handleLogin()*, το email αρχικά καθαρίζεται από περιττά κενά και μετατρέπεται σε πεζούς χαρακτήρες.

Πριν πραγματοποιηθεί η πραγματική αυθεντικοποίηση, η εφαρμογή εκτελεί πρόσθετο έλεγχο στον πίνακα *allowed_uploaders*. Αναζητείται εγγραφή με το συγκεκριμένο email και με ενεργή κατάσταση. Το σχετικό τμήμα του κώδικα είναι:

```
const { data: allowedUser } = await supabase
  .from("allowed_uploaders")
  .select("email")
  .eq("email", normalizedEmail)
  .eq("is_active", true)
  .maybeSingle();
```

Ο έλεγχος αυτός διαχωρίζει δύο διαφορετικά επίπεδα ασφάλειας. Το πρώτο είναι η ταυτοποίηση του χρήστη, δηλαδή αν υπάρχουν έγκυρα στοιχεία σύνδεσης. Το δεύτερο είναι

η εξουσιοδότηση, δηλαδή αν ο συγκεκριμένος λογαριασμός επιτρέπεται να χρησιμοποιεί τη συγκεκριμένη εφαρμογή. Η ύπαρξη λογαριασμού στο σύστημα αυθεντικοποίησης δεν συνεπάγεται συνεπώς αυτόματα με δικαίωμα πρόσβασης στη λειτουργία μεταφόρτωσης.

Εφόσον ο έλεγχος εξουσιοδότησης ολοκληρωθεί επιτυχώς, καλείται η *supabase.auth.signInWithPassword()*. Το Supabase Auth αναλαμβάνει την επαλήθευση του email και του κωδικού και, σε περίπτωση επιτυχίας, δημιουργεί ενεργή σύνδεση. Η εφαρμογή στη συνέχεια μεταφέρει τον χρήστη στο */upload*. Η συνεδρία αυτή χρησιμοποιείται από το frontend στις επόμενες κλήσεις, ώστε να είναι γνωστή η ταυτότητα του συνδεδεμένου χρήστη.

Ο έλεγχος πρόσβασης επαναλαμβάνεται στην κύρια σελίδα μεταφόρτωσης και δεν θεωρείται ότι η επιτυχής σύνδεση στο παρελθόν είναι επαρκής. Μέσα σε *useEffect* καλείται η *supabase.auth.getUser()* για να διαπιστωθεί αν υπάρχει ενεργός χρήστης. Αν δεν υπάρχει, ο browser ανακατευθύνεται στη σελίδα σύνδεσης. Αν υπάρχει, πραγματοποιείται εκ νέου έλεγχος στον *allowed_uploaders*. Εφόσον η εγγραφή δεν είναι πλέον ενεργή, η εφαρμογή πραγματοποιεί *signOut()* και διακόπτει την πρόσβαση. Η επανάληψη αυτή είναι σημαντική διότι επιτρέπει την ανάκληση του δικαιώματος χρήσης ακόμη και όταν μία συνεδρία είχε δημιουργηθεί σε προηγούμενο χρόνο.

Μετά την επιβεβαίωση της πρόσβασης, η ίδια διαδικασία αρχικοποίησης ανακτά από τον πίνακα *hotels* τις διαθέσιμες εγγραφές. Οι επιλογές δεν είναι ενσωματωμένες ως σταθερή λίστα στον κώδικα του frontend. Ανακτώνται από τη βάση και αποθηκεύονται στην τοπική κατάσταση της σελίδας. Εάν δημιουργηθεί νέα εγγραφή μέσω της εφαρμογής, αυτή εισάγεται στη βάση, προστίθεται άμεσα στην κατάσταση του στοιχείου React και μπορεί να χρησιμοποιηθεί χωρίς νέα δημοσίευση της εφαρμογής. Με αυτόν τον τρόπο η διεπαφή βασίζεται σε δυναμικά δεδομένα της βάσης και όχι σε χειροκίνητη συντήρηση επιλογών.

3.3 Επιλογή, μεταφόρτωση και καταχώριση αρχείων

Η κύρια διαδικασία μεταφόρτωσης υλοποιείται στο *app/upload/page.tsx*. Η σελίδα διατηρεί την κατάσταση της διαδικασίας μέσω της κατάστασης React: το επιλεγμένο ξενοδοχείο, την κατηγορία του αρχείου, την χρονική περίοδο, τα αρχεία που έχουν επιλεγεί αλλά δεν έχουν ακόμη μεταφορτωθεί, καθώς και την κατάσταση κάθε μεταφόρτωσης. Με αυτόν τον τρόπο η διεπαφή μπορεί να ενημερώνεται δυναμικά χωρίς πλήρη επαναφόρτωση της σελίδας.

Για την επιλογή των αρχείων χρησιμοποιείται η βιβλιοθήκη *react-dropzone*. Η εφαρμογή υποστηρίζει τόσο την επιλογή από τον επιλογέα αρχείων όσο και μεταφορά και απόθεση αρχείων. Οι επιτρεπόμενες επεκτάσεις είναι *.csv*, *.xls* και *.xlsx*, ενώ εφαρμόζεται

και ανώτατο όριο μεγέθους 25 MB. Ο έλεγχος πραγματοποιείται τόσο μέσω των αποδεκτών τύπων MIME όσο και μέσω της συνάρτησης *isAllowedFile()*, η οποία εξετάζει την επέκταση του ονόματος.

Τα αρχεία που περνούν τον αρχικό έλεγχο τοποθετούνται προσωρινά στη λίστα *pendingFiles*. Η μεταφόρτωση δεν ξεκινά αυτόματα μόλις επιλεγεί ένα αρχείο. Ο χρήστης έχει έτσι την δυνατότητα να ελέγξει το σύνολο που πρόκειται να αποσταλεί, ενώ η εφαρμογή μπορεί να συνδέσει ολόκληρη την ομάδα αρχείων με την ίδια επιλογή ξενοδοχείου και τα ίδια βασικά μεταδεδομένα. Όταν ξεκινήσει η διαδικασία, κάθε αρχείο επεξεργάζεται μέσω της *uploadFile(file, hotel)*.

Το πρώτο βήμα της *uploadFile()* είναι η δημιουργία τεχνικού ονόματος. Χαρακτήρες που δεν ανήκουν στο σύνολο γραμμάτων, αριθμών, τελείας, παύλας και κάτω παύλας αντικαθίστανται, ώστε το όνομα να μπορεί να χρησιμοποιηθεί με μεγαλύτερη ασφάλεια μέσα σε URL και διαδρομές αποθήκευσης. Το αρχικό όνομα δεν χάνεται, καθώς αποθηκεύεται αργότερα ξεχωριστά ως μεταδεδομένο.

```
const safeFilename = file.name.replace(
  /[^a-zA-Z0-9._-]/g,
  "_ "
);
```

Στην συνέχεια δημιουργείται διαδρομή αποθήκευσης. Ο κώδικας χρησιμοποιεί τον κωδικό του ξενοδοχείου, την κατηγορία του αρχείου, το έτος, τον μήνα και χρονική σήμανση. Η λογική αυτή παράγει μια ιεραρχική δομή αντί να τοποθετεί όλα τα αντικείμενα σε έναν ενιαίο χώρο.

```
const storagePath =
  `${hotel.code}/` +
  `${category}/` +
  `${fileYear}/` +
  `${monthFolder}/` +
  `${timestamp}-${safeFilename}`;
```

Η συγκεκριμένη οργάνωση διευκολύνει την διαχείριση των αντικειμένων και μειώνει την πιθανότητα σύγκρουσης μεταξύ δύο αρχείων με ίδιο όνομα. Η χρονική σήμανση αποτελεί μέρος της διαδρομής, επομένως δύο διαφορετικές μεταφορτώσεις του ίδιου ονόματος αρχείου δημιουργούν διαφορετικές θέσεις αποθήκευσης. Το πραγματικό αρχείο μεταφέρεται απευθείας από το frontend στον κάδο *business-files* του Supabase Storage.

Μετά την επιτυχημένη μεταφόρτωση δημιουργείται η εγγραφή του αρχείου στον πίνακα *uploaded_files*. Στην εγγραφή αυτή αποθηκεύονται, μεταξύ άλλων, το *hotel_id*, η κατηγορία, το αρχικό όνομα αρχείου, το *storage_path*, η χρονική περίοδος, τον τύπο MIME,

το μέγεθος, το email του χρήστη και η αρχική κατάσταση *uploaded*. Η εγγραφή δεν αποτελεί αντίγραφο του ίδιου του αρχείου, αλλά την ταυτότητα της μεταφόρτωσης. Η βασική της σημασία είναι ότι επιστρέφει το *id*, το οποίο λειτουργεί ως κοινό αναγνωριστικό για όλα τα επόμενα στάδια.

```
const { data: insertedFile } = await supabase
  .from("uploaded_files")
  .insert({
    hotel_id: hotel.id,
    category,
    original_filename: file.name,
    storage_path: storagePath,
    file_month: fileMonth,
    file_year: fileYear,
    mime_type: file.type || null,
    file_size: file.size,
    uploaded_by_email: email,
    status: "uploaded",
  })
  .select("id")
  .single();
```

Αμέσως μετά δημιουργούνται οι πρώτες εγγραφές στον *processing_logs*. Το γεγονός *upload_completed* καταγράφει ότι το αρχείο αποθηκεύτηκε επιτυχώς και περιλαμβάνει στο πεδίο λεπτομερειών το όνομα αρχείου, τη διαδρομή αποθήκευσης, την κατηγορία και τα αναγνωριστικά. Ακολουθεί το *processor_check*, το οποίο καταγράφει ότι η εφαρμογή πρόκειται να καλέσει το Python σύστημα. Η καταγραφή συμβάντων ξεκινά επομένως πριν από την ίδια την επεξεργασία και δημιουργεί χρονική ακολουθία της διαδικασίας.

Πρώτα αποθηκεύεται το αρχείο, έπειτα δημιουργείται η εγγραφή στη βάση και μόνο τότε καλείται το backend. Αν η μεταφόρτωση στο Storage αποτύχει, δεν δημιουργείται ψευδής εικόνα ότι το αρχείο είναι διαθέσιμο προς επεξεργασία. Αντίστοιχα, το backend API δεν καλείται πριν υπάρξει το *file_id* και το *storage_path* στην βάση.

3.4 Επικοινωνία μεταξύ διεπαφής και υποσυστήματος επεξεργασίας

Η σύνδεση μεταξύ του frontend και του backend υλοποιείται ως επικοινωνία δύο ανεξάρτητων εφαρμογών μέσω HTTP. Το frontend, το οποίο εκτελείται στο περιβάλλον του browser, είναι υπεύθυνο για την αλληλεπίδραση με τον χρήστη και για τις λειτουργίες προς το Supabase. Το backend αποτελεί ξεχωριστή υπηρεσία FastAPI και αναλαμβάνει τις λειτουργίες που δεν πρέπει ή δεν είναι πρακτικό να εκτελούνται στον browser, όπως η ανάκτηση αρχείων από το Storage, ο καθαρισμός με pandas, η εφαρμογή των επιχειρησιακών

κανόνων και η εκτέλεση των μοντέλων πρόβλεψης. Η διεύθυνση της Python υπηρεσίας δεν είναι ενσωματωμένη στα στοιχεία React αλλά ανακτάται από τη μεταβλητή περιβάλλοντος `NEXT_PUBLIC_PROCESSOR_URL`. Με αυτόν τον τρόπο το ίδιο frontend μπορεί να συνδεθεί με διαφορετικό backend στο περιβάλλον ανάπτυξης και στο παραγωγικό περιβάλλον.

Η βασική σύνδεση των δύο άκρων εμφανίζεται στην συνάρτηση μεταφόρτωσης του `app/upload/page.tsx`. Αφού το αρχείο έχει αποθηκευτεί στο Supabase Storage και έχει δημιουργηθεί η αντίστοιχη εγγραφή στον `uploaded_files`, το frontend καλεί το τελικό σημείο API `/process-file` και αποστέλλει μόνο το αναγνωριστικό της εγγραφής:

```
const processorResponse = await fetch(
  `${process.env.NEXT_PUBLIC_PROCESSOR_URL}/process-file`,
  {
    method: "POST",
    headers: {
      "Content-Type": "application/json",
    },
    body: JSON.stringify({
      file_id: insertedFile.id,
    }),
  },
);
```

Η επιλογή αυτή είναι βασική για τη συνοχή της εφαρμογής. Το περιεχόμενο του αρχείου δεν μεταδίδεται δεύτερη φορά από το frontend προς το backend API. Το backend λαμβάνει το `file_id`, αναζητά την εγγραφή του αρχείου στην βάση και από εκεί διαβάζει το `storage_path`, την κατηγορία, το `hotel_id` και τα υπόλοιπα μεταδεδομένα. Επομένως, η εγγραφή της βάσης λειτουργεί ως το κοινό σημείο αναφοράς των δύο εφαρμογών και αποφεύγεται η δημιουργία δύο διαφορετικών εκδοχών της ίδιας πληροφορίας.

Το FastAPI πραγματοποιεί τον αρχικό έλεγχο της δομής του JSON πριν εκτελεστεί η κεντρική συνάρτηση:

```
class ProcessFileRequest(BaseModel):
    file_id: str

@app.post("/process-file")
def process_file(request: ProcessFileRequest):
    ...
```

Η εφαρμογή FastAPI φορτώνει από το περιβάλλον τις `SUPABASE_URL`, `SUPABASE_SERVICE_ROLE_KEY` και `SUPABASE_BUCKET`. Το `service_role` δεν αποστέλλεται ποτέ στον browser. Χρησιμοποιείται μόνο από το backend για τις

λειτουργίες προς το PostgreSQL REST API και το Supabase Storage. Αντίθετα, το frontend χρησιμοποιεί το δημόσιο κλειδί *anon* και περιορίζεται από τις πολιτικές πρόσβασης που έχουν οριστεί.

Για την επικοινωνία του backend με το Supabase έχουν δημιουργηθεί κοινές συναρτήσεις, όπως *supabase_get_one()*, *supabase_insert()*, *supabase_update()* και *supabase_delete()*, καθώς και οι *storage_download()* και *storage_upload()*. Η ύπαρξη αυτού του ενδιάμεσου επιπέδου είναι σημαντική, επειδή οι μηχανισμοί καθαρισμού και οι μονάδες πρόβλεψης δεν χρειάζεται να επαναλαμβάνουν τον τρόπο κατασκευής των κεφαλίδων HTTP, την αυθεντικοποίηση ή τον βασικό χειρισμό των αποκρίσεων.

Εκτός από το τελικό σημείο καθαρισμού, το backend εκθέτει ανεξάρτητο σύνολο τελικών σημείων API για την πρόβλεψη μέσω του *forecasting/router.py*. Ο δρομολογητής δημιουργείται με πρόθεμα */forecast* και διαχωρίζει την ανάγνωση ρυθμίσεων, την εκτέλεση νέας πρόβλεψης και την ανάκτηση αποθηκευμένων αποτελεσμάτων. Η βασική δομή του δρομολογητή είναι αντίστοιχη με την ακόλουθη:

```
router = APIRouter(prefix="/forecast")

@router.get("/monthly/config")
def get_monthly_forecast_config():
    ...

@router.post("/monthly/run")
def run_monthly_forecast(request):
    ...

@router.post("/monthly/run-all")
def run_all_monthly_forecasts(request):
    ...

@router.get("/monthly/latest")
def get_latest_monthly_forecast(...):
    ...
```

Τα βασικά τελικά σημεία API της εφαρμογής παρουσιάζονται στον Πίνακα 3.1. Στον πίνακα διαχωρίζονται οι λειτουργίες επεξεργασίας αρχείων από τις λειτουργίες πρόβλεψης, ώστε να είναι σαφές ποιο τελικό σημείο μεταβάλλει δεδομένα και ποιο χρησιμοποιείται μόνο για ανάγνωση.

Για μία μεμονωμένη πρόβλεψη το αίτημα περιλαμβάνει το ξενοδοχείο και την μεταβλητή που θα προβλεφθεί. Η ίδια λογική χρησιμοποιείται και από το frontend όταν ο χρήστης επιλέξει πρόβλεψη για συγκεκριμένη μεταβλητή-στόχο:

```
await fetch(`${processorUrl}/forecast/monthly/run`, {
```

Πίνακας 3.1: Κύρια τελικά σημεία του API της εφαρμογής.

Μέθοδος	Διαδρομή	Λειτουργία
GET	/	Έλεγχος διαθεσιμότητας της υπηρεσίας Python.
POST	<i>/process-file</i>	Εκκίνηση καθαρισμού για το αρχείο που αντιστοιχεί στο <i>file_id</i> .
GET	<i>/forecast/monthly/config</i>	Ανάκτηση της ενεργής ρύθμισης της μηνιαίας πρόβλεψης, όπως μεταβλητές-στόχοι, ορίζοντες και διαθέσιμα μοντέλα.
POST	<i>/forecast/monthly/run</i>	Εκτέλεση πρόβλεψης για ένα <i>hotel_id</i> και μία <i>target_name</i> .
POST	<i>/forecast/monthly/run-all</i>	Εκτέλεση των μεταβλητών-στόχων για μία μονάδα ή για συγκεκριμένη λίστα στόχων.
GET	<i>/forecast/monthly/latest</i>	Ανάκτηση της τελευταίας ολοκληρωμένης εκτέλεσης για συγκεκριμένη μονάδα και μεταβλητή-στόχο.
GET	<i>/forecast/monthly/latest-all</i>	Ανάκτηση των τελευταίων ολοκληρωμένων προβλέψεων όλων των μεταβλητών-στόχων μιας μονάδας.

```

method: "POST",
headers: { "Content-Type": "application/json" },
body: JSON.stringify({
  hotel_id: hotelId,
  target_name: targetName,
  trigger_type: "manual",
}),
});

```

Το *trigger_type="manual"* είναι σημαντικό για την λειτουργία. Ο καθαρισμός ενός αρχείου και η πρόβλεψη είναι δύο ξεχωριστές διαδικασίες. Η ενεργή ρύθμιση της πρόβλεψης έχει *automatic_after_upload=false*, επομένως μία επιτυχής επεξεργασία αρχείου δεν συνεπάγεται με αυτόματη εκπαίδευση όλων των μοντέλων. Το frontend μπορεί, μετά τον καθαρισμό, να ξεκινήσει την πρόβλεψη εφόσον αυτό έχει επιλεγεί από τον χρήστη. Με τον διαχωρισμό αυτό μία αποτυχία ή μία χρονοβόρα εκτέλεση πρόβλεψης δεν αλλάζει την επιτυχή κατάσταση του καθαρισμένου αρχείου.

Η απόκριση του */process-file* χρησιμοποιείται από το frontend για την ενημέρωση της κατάστασης της μεταφόρτωσης. Αν το backend επιστρέψει σφάλμα HTTP, δημιουργείται εγγραφή καταγραφής *processor_call_failed* και εμφανίζεται το αντίστοιχο μήνυμα. Με αυτόν τον τρόπο διαχωρίζεται ένα σφάλμα της επεξεργασίας από ένα πρόβλημα δικτυακής επικοινωνίας.

3.5 Κεντρική διαδικασία επεξεργασίας και επιλογή μηχανισμού καθαρισμού

Η συνάρτηση *process_file()* αποτελεί το βασικό σημείο συντονισμού του backend. Ο ρόλος της δεν είναι να περιέχει όλους τους κανόνες καθαρισμού, αλλά να οργανώνει με σταθερή σειρά τα βήματα που πρέπει να ακολουθήσει οποιοδήποτε αρχείο. Η προσέγγιση αυτή διαχωρίζει τη ροή της εφαρμογής από την εξειδικευμένη λογική κάθε κατηγορίας.

Το πρώτο βήμα είναι η ανάκτηση της κεντρικής εγγραφής από τον *uploaded_files* με βάση το *file_id*. Αν δεν βρεθεί εγγραφή, η διαδικασία σταματά με HTTP 404, επειδή δεν υπάρχει έγκυρη πηγή πάνω στην οποία μπορεί να εκτελεστεί η επεξεργασία. Στη συνέχεια ανακτάται το *hotel_id* και το backend πραγματοποιεί νέα ανάγνωση από τον πίνακα *hotels*.

Αφού επιβεβαιωθεί η συσχέτιση με έγκυρη εγγραφή, ελέγχεται η κατηγορία του αρχείου και ανακτώνται το *storage_path* και το *original_filename*. Η κατάσταση του αρχείου ενημερώνεται σε *cleaning* και δημιουργείται εγγραφή καταγραφής με *cleaning_started*. Η αλλαγή της κατάστασης πραγματοποιείται πριν από την επεξεργασία, ώστε η βάση να δείχνει ότι η εκτέλεση έχει ξεκινήσει ακόμη και αν η επόμενη φάση αποτύχει.

Στο επόμενο βήμα το backend καλεί τη *storage_download()* και ανακτά το περιεχόμενο από το Supabase Storage σε μορφή bytes. Επειδή οι βιβλιοθήκες επεξεργασίας αρχείων λειτουργούν αποτελεσματικά με τοπικό path, το περιεχόμενο γράφεται προσωρινά μέσω *tempfile.NamedTemporaryFile*. Η κατάληξη του προσωρινού αρχείου επιλέγεται με βάση το αρχικό όνομα αρχείου, ώστε να διατηρείται η σωστή πληροφορία. Το προσωρινό αρχείο διαγράφεται μέσα σε *finally*, δηλαδή ανεξάρτητα από την τελική επιτυχία ή αποτυχία της επεξεργασίας.

Η επιλογή του κατάλληλου μηχανισμού καθαρισμού πραγματοποιείται μόνο αφού ολοκληρωθούν οι προηγούμενοι έλεγχοι. Η *process_file()* εξετάζει την αποθηκευμένη κατηγορία και δρομολογεί την εκτέλεση στον αντίστοιχο μηχανισμό:

```
if category == "bookings":
    result = clean_bookings_file(
        local_path,
        original_filename,
    )

elif category == "expenses":
    result = clean_expenses_file(
        local_path=local_path,
        original_filename=original_filename,
    )

elif category == "income":
```

```
customer_mappings = fetch_customer_mappings()

result = clean_income_file(
    local_path=local_path,
    original_filename=original_filename,
    customer_mappings=customer_mappings,
)
```

Ο μηχανισμός για τις κρατήσεις βρίσκεται στο κύριο αρχείο του backend, ενώ οι διαδικασίες για τις άλλες κατηγορίες βρίσκονται στα *expenses_cleaner.py* και *income_cleaner.py*. Η συγκεκριμένη διάσπαση έχει πρακτικό όφελος: κάθε μηχανισμός καθαρισμού μπορεί να εξελίσσεται ανεξάρτητα και να περιλαμβάνει τους δικούς του κανόνες χωρίς να επιβαρύνει τον κεντρικό συντονιστή. Η *process_file()* παραμένει υπεύθυνη για όσα είναι κοινά σε όλες τις κατηγορίες, όπως η ανάκτηση του αρχείου, η κατάσταση, η καταγραφή συμβάντων, η αποθήκευση και η δημιουργία της απόκρισης.

Οι μηχανισμοί καθαρισμού επιστρέφουν κοινή δομή αποτελέσματος. Η δομή αυτή περιλαμβάνει το επεξεργασμένο DataFrame, την αρχική αναπαράσταση των γραμμών, την καθαρισμένη αναπαράσταση και τις πληροφορίες ελέγχου. Μετά την επιστροφή από τον μηχανισμό καθαρισμού, η κεντρική διαδικασία προσθέτει τα κοινά μεταδεδομένα στο DataFrame και στις καθαρισμένες εγγραφές.

Η έννοια της «επεξεργασίας» είναι επομένως ευρύτερη από την έννοια του «καθαρισμού». Η επεξεργασία περιλαμβάνει ολόκληρη τη διαδρομή από την ανάκτηση της πηγής μέχρι την ενημέρωση της βάσης και την παραγωγή της τελικής κατάστασης. Ο καθαρισμός είναι το εσωτερικό στάδιο στο οποίο το περιεχόμενο του αρχείου διαβάζεται, τυποποιείται, ελέγχεται και μετασχηματίζεται. Ο διαχωρισμός των δύο εννοιών είναι χρήσιμος για την κατανόηση της αρχιτεκτονικής, διότι επιτρέπει το ίδιο πλαίσιο επεξεργασίας να χρησιμοποιεί διαφορετικούς μηχανισμούς καθαρισμού ανάλογα με την κατηγορία.

3.6 Καθαρισμός, τυποποίηση και έλεγχος των δεδομένων

Ο καθαρισμός αποτελεί το στάδιο κατά το οποίο το αρχικό αρχείο μετατρέπεται από μία ανομοιογενή μορφή σε μια συνεπή δομή. Η διαδικασία δεν περιορίζεται στην αφαίρεση κενών γραμμών ή στην απλή αλλαγή ονομάτων στηλών. Περιλαμβάνει την αναγνώριση της πραγματικής δομής του αρχείου, την αντιστοίχιση των πεδίων, την μετατροπή τύπων, τον έλεγχο της εγκυρότητας και την παραγωγή της καθαρισμένης εγγραφής.

Το πρώτο βήμα είναι η ανάγνωση του αρχείου. Τα CSV δεν θεωρείται ότι χρησιμοποιούν πάντοτε το ίδιο διαχωριστικό ούτε ότι η πρώτη γραμμή αποτελεί την πραγματική επικεφαλίδα. Για τον λόγο αυτό μπορούν να διαβαστούν αρχικά με *header=None* και με δυνατότητα αυτόματης αναγνώρισης του διαχωριστικού. Ως προς την κωδικοποίηση,

επιχειρείται αρχικά ανάγνωση με *utf-8-sig*. Εάν το αρχείο προέρχεται από παλαιότερο περιβάλλον και εμφανιστεί σφάλμα αποκωδικοποίησης, χρησιμοποιείται εναλλακτικά η *cp1253*. Η αντιμετώπιση αυτή είναι σημαντική για ελληνικά αρχεία διαφορετικής προέλευσης.

Μετά την ανάγνωση ακολουθεί ο εντοπισμός της πραγματικής γραμμής επικεφαλίδων. Ο κώδικας δεν χρησιμοποιεί σταθερό αριθμό γραμμής. Εξετάζει τις πρώτες γραμμές και συγκρίνει τις τιμές τους με σύνολο αναμενόμενων όρων. Κάθε πιθανή γραμμή λαμβάνει βαθμολογία και επιλέγεται εκείνη με την ισχυρότερη αντιστοίχιση. Η λογική αυτή επιτρέπει στο σύστημα να αγνοεί τίτλους αναφοράς, κενές γραμμές ή άλλα στοιχεία που μπορεί να προηγούνται του πραγματικού πίνακα.

Η σύγκριση δεν πραγματοποιείται πάνω στο ακατέργαστο κείμενο. Η *normalize_text()* μετατρέπει τις συμβολοσειρές σε κοινή πεζή μορφή, εφαρμόζει Unicode normalization, αφαιρεί διακριτικά σημεία όπως οι τόνοι και αντικαθιστά σημεία στίξης. Η τυποποιημένη τιμή χρησιμοποιείται μόνο για την αντιστοίχιση και δεν αντικαθιστά την αρχική τιμή που πρέπει να διατηρηθεί. Έτσι, μικρές διαφορές στη γραφή δεν εμποδίζουν το σύστημα να αναγνωρίσει ότι δύο επικεφαλίδες εκφράζουν την ίδια έννοια.

Το επόμενο στάδιο είναι η αντιστοίχιση των εξωτερικών στηλών σε ένα κοινό εσωτερικό σχήμα. Για κάθε πεδίο υπάρχει ομάδα από εναλλακτικές ονομασίες. Επομένως, ο υπόλοιπος κώδικας μπορεί να χρησιμοποιεί μία σταθερή εσωτερική ονομασία ακόμη και όταν τα εξωτερικά αρχεία χρησιμοποιούν διαφορετικές επικεφαλίδες. Η προσέγγιση αυτή περιορίζει τη κωδικοποίηση πάνω σε μία συγκεκριμένη έκδοση αρχείου και επιτρέπει την υποστήριξη νέων παραλλαγών με προσθήκη αντιστοιχίσεων αντί με ανασχεδιασμό της επεξεργασίας.

Μετά την τυποποίηση ακολουθεί η μετατροπή των τιμών. Οι ημερομηνίες μετατρέπονται σε κοινή αναπαράσταση ημερομηνίας και ώρας και οι μη αναγνωρίσιμες τιμές μπορούν να μετατραπούν σε ελλιπείς τιμές ώστε να αξιολογηθούν στο στάδιο ελέγχου εγκυρότητας. Αντίστοιχα, στις αριθμητικές τιμές λαμβάνονται υπόψη διαφορετικές χρήσεις για το δεκαδικό και το διαχωριστικό χιλιάδων. Μία τιμή της μορφής *1.234,56* δεν μπορεί να θεωρηθεί αξιόπιστος αριθμός πριν μετατραπεί σε κοινή αριθμητική αναπαράσταση. Η τυποποίηση προηγείται συνεπώς των υπολογισμών.

Ο έλεγχος της ποιότητας πραγματοποιείται σε περισσότερα από ένα επίπεδα. Υπάρχουν δομικοί έλεγχοι για την παρουσία απαραίτητων πεδίων, έλεγχοι τύπου για ημερομηνίες και αριθμούς και λογικοί έλεγχοι μεταξύ πεδίων. Δεν αντιμετωπίζονται όλες οι αποκλίσεις ως σφάλματα. Αν ένα πρόβλημα καθιστά αδύνατη την επεξεργασία, η διαδικασία διακόπτεται. Αντίθετα, μικρότερες αποκλίσεις μπορούν να καταγραφούν ως προειδοποιήσεις και το αρχείο να συνεχίσει την επεξεργασία. Η διάκριση αυτή αποτρέπει τόσο την υπερβολικά αυστηρή απόρριψη όσο και τη αγνόηση πραγματικών προβλημάτων.

Το αποτέλεσμα ενός μηχανισμού καθαρισμού δεν είναι μόνο ένα DataFrame. Επιστρέφονται επίσης *raw_records*, *cleaned_records*, συνοπτικά στοιχεία, προειδοποιήσεις και σφάλματα. Αυτή η δομή είναι σημαντική για τον έλεγχο ιχνηλασιμότητας της εφαρμογής. Για κάθε γραμμή μπορεί να διατηρηθεί η μορφή πριν από τον καθαρισμό και η μορφή μετά τον καθαρισμό, ενώ τα μεταδεδομένα περιγράφουν την ποιότητα και το τελικό αποτέλεσμα της διαδικασίας. Οι ειδικοί επιχειρησιακοί κανόνες που εφαρμόζονται πάνω σε αυτή την καθαρισμένη βάση εξετάζονται στο επόμενο κεφάλαιο, ώστε στο παρόν να παραμένει σαφής ο γενικός μηχανισμός.

Πριν από την αποστολή των αποτελεσμάτων προς το Supabase εφαρμόζεται η *sanitize_for_json()*. Η ανάγκη προκύπτει επειδή το pandas και το NumPy χρησιμοποιούν τύπους και ειδικές τιμές που δεν αποτελούν έγκυρη τυπική JSON αναπαράσταση, όπως *NaN* και *NaT*. Η συνάρτηση μετατρέπει τις ημερομηνίες, *missing* ή μη πεπερασμένες αριθμητικές τιμές σε *null* και ειδικούς αριθμητικούς τύπους στους αντίστοιχους βασικούς Python τύπους. Έτσι δημιουργείται ένα ενιαίο πλαίσιο πριν από την αποθήκευση.

3.7 Οργάνωση της βάσης δεδομένων, κλειδιά και σχέσεις

Η PostgreSQL βάση του Supabase σχεδιάστηκε ώστε να διαχωρίζει την κατάσταση μιας μεταφόρτωσης από το περιεχόμενο που παράγεται μετά τον καθαρισμό. Το Supabase Storage διατηρεί τα αρχικά αρχεία, ενώ η σχεσιακή βάση διατηρεί τα αναγνωριστικά, τις σχέσεις, τα μεταδεδομένα και τα αποτελέσματα της επεξεργασίας. Το *storage_path* αποτελεί επομένως αναφορά στη θέση ενός αντικειμένου στο Storage και όχι ξένο κλειδί προς κάποιον σχεσιακό πίνακα.

Το σχήμα υλοποιήθηκε απευθείας στην PostgreSQL του Supabase μέσω SQL. Οι βασικοί πίνακες δημιουργήθηκαν με *CREATETABLE*, ενώ μεταγενέστερες επεκτάσεις, όπως η σύνδεση των μεταφορτώσεων με τις ξενοδοχειακές μονάδες, προστέθηκαν με μεταβολές του σχήματος. Οι αναλυτικές προβολές δημιουργούνται με *CREATEVIEW* και δεν αποτελούν αντίγραφα των δεδομένων. Το υποσύστημα πρόβλεψης διαθέτει ξεχωριστό SQL query για εκτελέσεις μοντέλων, μετρικές, επιλογές και προβλέψεις. Με αυτή την προσέγγιση η δομή της βάσης είναι ορισμένη στο SQL επίπεδο και οι κανόνες δεν εξαρτώνται αποκλειστικά από τον Python κώδικα.

Στο σχεσιακό μοντέλο χρησιμοποιούνται πρωτεύοντα κλειδιά (Primary Keys) για τη μοναδική ταυτοποίηση κάθε εγγραφής και ξένα κλειδιά (Foreign Keys) για τη σύνδεση δύο πινάκων. Ένα πρωτεύον κλειδί δεν επιτρέπεται να ταυτοποιεί δύο διαφορετικές εγγραφές, ενώ ένα ξένο κλειδί αναφέρεται σε υπάρχον πρωτεύον κλειδί άλλου πίνακα και εκφράζει τη σχέση μεταξύ των δύο οντοτήτων. Η συγκεκριμένη διάκριση είναι ουσιαστική στην εφαρμογή, επειδή επιτρέπει μία μεταφόρτωση να αποτελεί την βασική οντότητα για

πολλές καθαρισμένες γραμμές και πολλά γεγονότα επεξεργασίας.

Ο πίνακας *hotels* διατηρεί τις ξενοδοχειακές μονάδες. Η τρέχουσα εφαρμογή διαβάζει από αυτόν τα *id*, *name*, *code* και *is_active*. Το *id* αποτελεί το μοναδικό αναγνωριστικό της μονάδας και το *uploaded_files.hotel_id* συνδέει κάθε μεταφόρτωση με την μονάδα στην οποία ανήκει. Έτσι μία μονάδα μπορεί να έχει πολλές μεταφορτώσεις, ενώ κάθε μεταφόρτωση ανήκει σε μία συγκεκριμένη μονάδα.

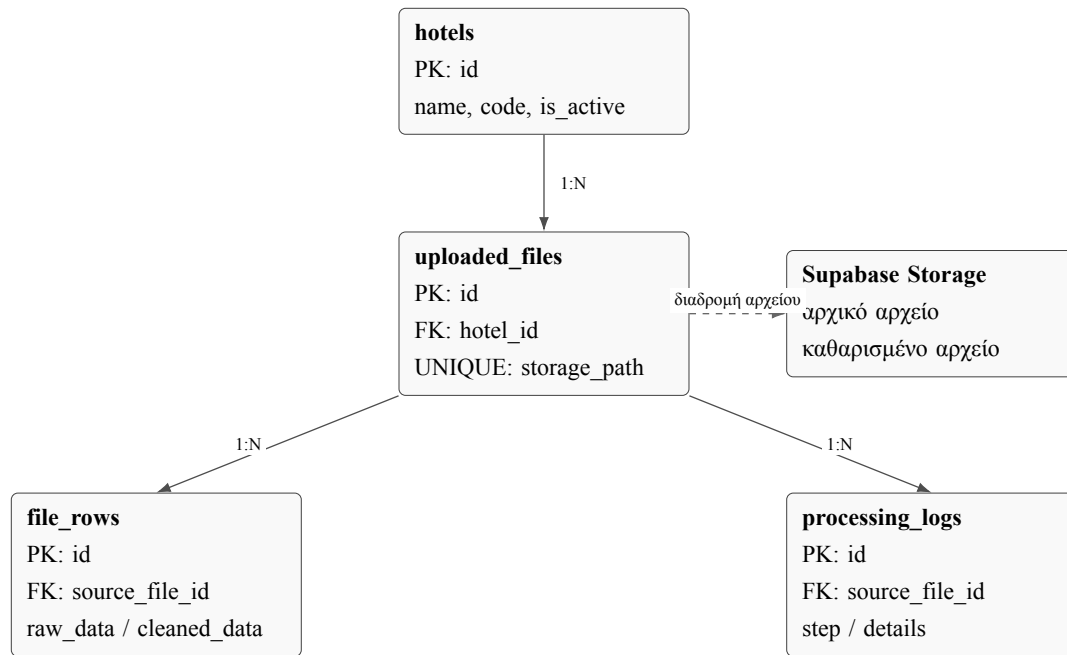
Ο *uploaded_files* είναι ο κεντρικός πίνακας της ροής. Το *id* του είναι πρωτεύον κλειδί και το *storage_path* προστατεύεται από περιορισμό *UNIQUE*, ώστε η ίδια διαδρομή αντικειμένου να μην είναι σε δύο διαφορετικές μεταφορτώσεις. Το *id* είναι επίσης το *file_id* που επιστρέφεται στο frontend μετά την εισαγωγή και στην συνέχεια αποστέλλεται στο */process-file*. Με αυτόν τον τρόπο το ίδιο αναγνωριστικό χρησιμοποιείται σε ολόκληρη τη διαδρομή από το frontend έως τις εγγραφές ανά γραμμή.

Ο *file_rows* διαθέτει δικό του πρωτεύον κλειδί *id*, ενώ το *source_file_id* είναι ξένο κλειδί προς το *uploaded_files.id*. Η σχέση είναι ένα-προς-πολλά (1:N): ένα αρχείο μπορεί να δημιουργήσει πολλές εγγραφές *file_rows*, αλλά κάθε γραμμή παραπέμπει σε ένα συγκεκριμένο αρχείο. Αντίστοιχα, ο *processing_logs* διαθέτει πρωτεύον κλειδί *id* και ξένο κλειδί *source_file_id* προς το ίδιο *uploaded_files.id*. Επομένως, η μεταφόρτωση λειτουργεί ως κοινός παρανομαστής τόσο για το περιεχόμενο όσο και για το ιστορικό της εκτέλεσης.

Πίνακας 3.2: Κύρια κλειδιά και σχέσεις του λειτουργικού σχήματος.

Πίνακας	Κλειδί / περιορισμός	Ρόλος στη σχέση
<i>hotels</i>	PK: <i>id</i>	Βασική οντότητα της ξενοδοχειακής μονάδας.
<i>uploaded_files</i>	PK: <i>id</i> ; FK: <i>hotel_id</i> ; UNIQUE: <i>storage_path</i>	Κεντρική εγγραφή κάθε μεταφόρτωσης και σύνδεση με το ξενοδοχείο.
<i>file_rows</i>	PK: <i>id</i> ; FK: <i>source_file_id</i>	Πολλές αρχικές και καθαρισμένες εγγραφές μπορούν να ανήκουν σε μία μεταφόρτωση.
<i>processing_logs</i>	PK: <i>id</i> ; FK: <i>source_file_id</i>	Πολλά γεγονότα επεξεργασίας μπορούν να ανήκουν στην ίδια μεταφόρτωση.
<i>allowed_uploaders</i>	PK: <i>id</i> ; UNIQUE: <i>email</i>	Ανεξάρτητος έλεγχος εξουσιοδότησης χρηστών.

Η δομή των βασικών σχέσεων παρουσιάζεται στο Σχήμα 3.2. Οι συνεχείς γραμμές δηλώνουν σχεσιακή σύνδεση μέσω ξένου κλειδιού, ενώ το *Storage* παρουσιάζεται ξεχωριστά επειδή δεν αποτελεί σχεσιακό πίνακα.



Σχήμα 3.2: Βασικό σχεσιακό σχήμα της λειτουργικής ροής και του Supabase Storage.

Στον `file_rows` οι στήλες `raw_data` και `cleaned_data` αποθηκεύονται ως JSONB. Η επιλογή αυτή επιτρέπει στις τρεις κατηγορίες αρχείων να χρησιμοποιούν τον ίδιο πίνακα ανά γραμμή. Τα σταθερά στοιχεία που χρησιμοποιούνται σε συνδέσεις και φίλτρα, όπως `id`, `source_file_id`, `category` και `row_number`, παραμένουν κανονικές σχεσιακές στήλες. Η μετατροπή των JSONB σε σταθερές αναλυτικές στήλες γίνεται αργότερα μέσω προβολών SQL και αναλύεται για τη συγκεκριμένη περίπτωση στο Κεφάλαιο 4.

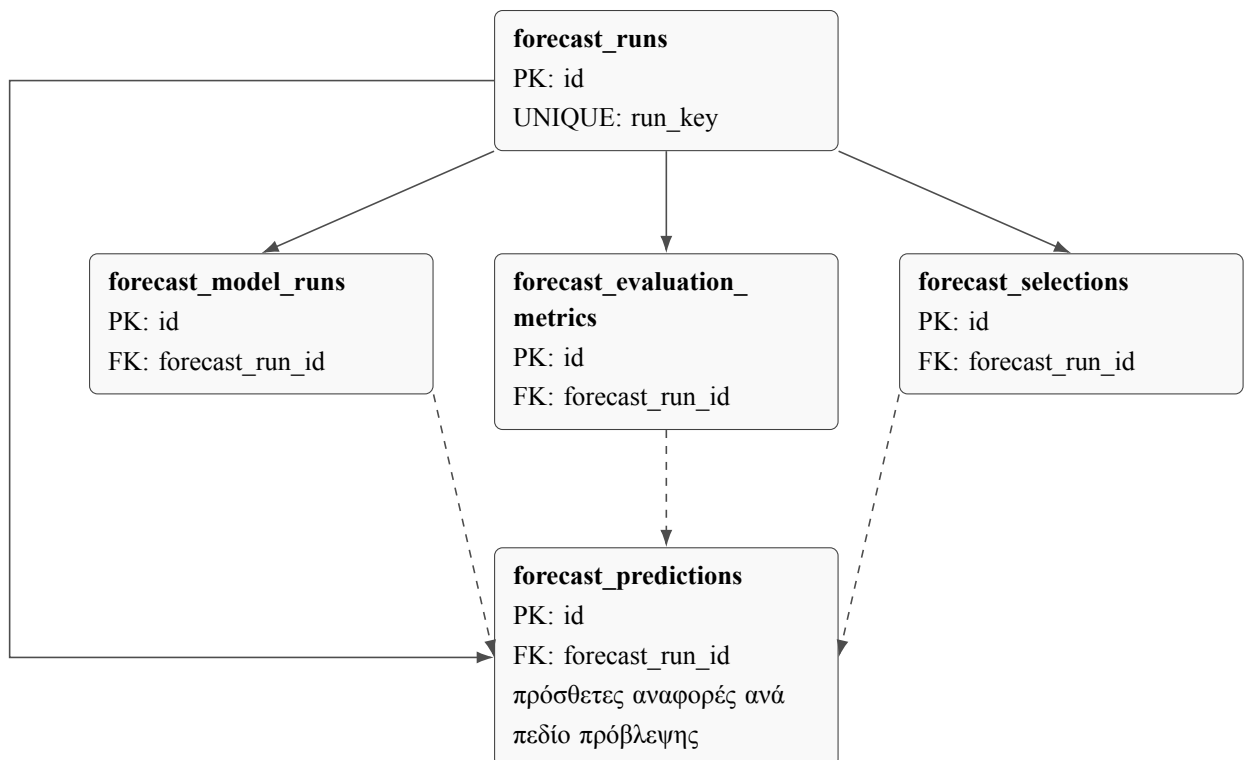
Μετά τον καθαρισμό, το τελικό DataFrame μετατρέπεται σε CSV με κωδικοποίηση `utf-8-sig` και αποθηκεύεται σε ξεχωριστή θέση στο Storage. Παράλληλα, πριν από την εισαγωγή των νέων `file_rows`, διαγράφονται τυχόν προηγούμενες εγγραφές με το ίδιο `source_file_id`. Οι νέες εγγραφές συγκεντρώνονται σε ομάδες των 500, ώστε να περιορίζεται ο αριθμός των REST κλήσεων. Στο τέλος ενημερώνεται το `uploaded_files` με την κατάσταση `cleaned`, το `processed_at`, το `cleaned_storage_path` και το `rows_count`. Αν η επεξεργασία αποτύχει, η κατάσταση μεταβάλλεται σε `failed` και αποθηκεύεται το σχετικό μήνυμα.

Για την πρόβλεψη χρησιμοποιείται ξεχωριστό σχεσιακό υποσύστημα. Ο `forecast_runs` αποτελεί τη βασική εγγραφή μιας ολοκληρωμένης εκτέλεσης πρόβλεψης και διαθέτει πρωτεύον κλειδί UUID. Το `run_key` είναι μοναδικό και χρησιμοποιείται για την αποφυγή επανάληψης εκτελέσεων. Οι `forecast_model_runs`, `forecast_evaluation_metrics` και `forecast_selections` διαθέτουν το καθένα δικό του πρωτεύον κλειδί και ξένο κλειδί `forecast_run_id` προς τον `forecast_runs`. Ο `forecast_predictions` συνδέεται επίσης με τη βασική εκτέλεση και, ανάλογα με το `prediction_scope`, μπορεί να συνδέεται με συγκεκριμένη εκτέλεση

μοντέλου, επιλογή ή αξιολόγηση μέσω πρόσθετων σύνθετων ξένων κλειδιών.

Πίνακας 3.3: Ρόλος των βασικών πινάκων του υποσυστήματος πρόβλεψης.

Πίνακας	Περιεχόμενο	Σχέση
<i>forecast_runs</i>	Μεταδεδομένα συνολικής εκτέλεσης, μεταβλητή-στόχος, περίοδοι, κατάσταση και επίσημη επιλογή.	Βασικός πίνακας, PK <i>id</i> , UNIQUE <i>run_key</i> .
<i>forecast_model_runs</i>	Εκτέλεση και παράμετροι κάθε υποψήφιου μοντέλου και ρύθμισης.	FK <i>forecast_run_id</i> προς <i>forecast_runs</i> .
<i>forecast_evaluation_metrics</i>	Διαδοχικά σύνολα ιστορικής αξιολόγησης και MAE, RMSE, sMAPE, Bias.	FK <i>forecast_run_id</i> προς <i>forecast_runs</i> .
<i>forecast_selections</i>	Υποψήφιες και επίσημη επιλογή μοντέλου ή στρατηγικής.	FK <i>forecast_run_id</i> προς <i>forecast_runs</i> .
<i>forecast_predictions</i>	Προβλέψεις βασικών μοντέλων, επιλογών και περιόδων αξιολόγησης.	FK προς τη κύρια εκτέλεση και πρόσθετες αναφορές ανά πεδίο εφαρμογής της πρόβλεψης.



Σχήμα 3.3: Σχεσιακή οργάνωση των αποτελεσμάτων της πρόβλεψης.

Στις βασικές σχέσεις της πρόβλεψης χρησιμοποιείται *ON DELETE CASCADE*. Έτσι, αν διαγραφεί μία εγγραφή *forecast_runs*, διαγράφονται και οι εξαρτώμενες εγγραφές. Επιπλέον, *UNIQUE* και *CHECK* περιορισμοί προστατεύουν τη λογική ακεραιότητα και οι πίνακες πρόβλεψης προστατεύονται επίσης με ασφάλεια σε επίπεδο γραμμής (Row Level Security) και οι εγγραφές πραγματοποιούνται από το backend μέσω του *service_role*.

3.8 Υποσύστημα πρόβλεψης και τεχνική δημιουργία της χρονοσειράς

Το υποσύστημα πρόβλεψης έχει υλοποιηθεί ως ανεξάρτητη μονάδα στον κατάλογο *forecasting/*. Ο διαχωρισμός αυτός είναι ουσιαστικός, επειδή η πρόβλεψη δεν αποτελεί απλό πρόσθετο βήμα του μηχανισμού καθαρισμού. Προϋποθέτει δική της πηγή δεδομένων, δημιουργία συνεχούς χρονοσειράς, κατασκευή χρονικών παραθύρων αξιολόγησης, εκτέλεση πολλών μοντέλων, υπολογισμό μετρικών, επιλογή αποτελέσματος και μόνιμη αποθήκευση. Η τοποθέτηση όλων αυτών των λειτουργιών μέσα στη *process_file()* θα συνέδεε υπερβολικά δύο διαφορετικές λειτουργίες.

Η δομή του υποσυστήματος ακολουθεί διαχωρισμό ρόλων. Το *config.py* ορίζει τις κεντρικές ρυθμίσεις, το *data.py* ανακτά και οργανώνει τη μηνιαία σειρά, το *models.py* περιέχει τις υλοποιήσεις των μοντέλων, το *metrics.py* τις μετρικές αξιολόγησης και το *evaluation.py* την ιστορική αξιολόγηση τύπου rolling-origin. Οι μονάδες επιλογής συγκρίνουν τα αποτελέσματα, το *service.py* συντονίζει τη δημιουργία της τελικής πρόβλεψης, το *repository.py* αναλαμβάνει τη μόνιμη αποθήκευση στη βάση και το *router.py* διοργανώνει τη λειτουργία μέσω HTTP. Η δομή αυτή επιτρέπει ένα μοντέλο ή μία διαδικασία επιλογής να αλλάξει χωρίς να αλλάξει ο τρόπος με τον οποίο το frontend καλεί το API.

Το σύστημα χρησιμοποιεί ως βασική μηνιαία πηγή την προβολή *v_forecast_monthly_series* και ως συμπληρωματική πηγή λειτουργίας το *v_hotel_operating_calendar*. Οι υποστηριζόμενες μεταβλητές-στόχοι είναι *active_bookings*, *included_net_revenue* και *included_operating_expenses*. Η συχνότητα είναι μηνιαία και οι ορίζοντες πρόβλεψης είναι 1, 3, 6 και 12 μήνες. Για κρατήσεις και έσοδα η περίοδος λειτουργίας ορίζεται στους μήνες Απρίλιο έως Οκτώβριο, ενώ τα έξοδα αντιμετωπίζονται σε ετήσια βάση.

Το πρώτο στάδιο της πρόβλεψης είναι η κατασκευή του συνόλου δεδομένων για ένα συγκεκριμένο *hotel_id* και *target_name*. Το *data.py* ελέγχει ότι το ερώτημα επέστρεψε μία μόνο μονάδα, μία μόνο μεταβλητή-στόχο και μία μόνο παρατήρηση ανά *period_month*. Στη συνέχεια δημιουργείται μηνιαίος δείκτης από τον πρώτο έως τον τελευταίο διαθέσιμο μήνα. Εάν λείπει ενδιάμεσος μήνας, συμπληρώνεται με μηδενική τιμή ώστε όλα τα μοντέλα να λαμβάνουν συνεχή *DatetimeIndex*. Παράλληλα διατηρείται *observed_mask*, ώστε το σύστημα να γνωρίζει ποιοι μήνες υπήρχαν πραγματικά στην πηγή και ποιοι προ-

στέθηκαν μόνο για να εξασφαλιστεί η συνέχεια της σειράς. Οι περίοδοι ελέγχου της ιστορικής αξιολόγησης επιτρέπεται να βασίζονται μόνο σε παρατηρημένους μήνες.

Η λογική αυτή αποτυπώνεται στον κώδικα με τη δημιουργία πλήρους χρονικού δείκτη:

```
full_index = pd.date_range(
    start=observed_series.index.min(),
    end=observed_series.index.max(),
    freq="MS",
)

complete_series = observed_series.reindex(full_index)
observed_mask = complete_series.notna()
complete_series = complete_series.fillna(0.0)
```

Ο τελευταίος διαθέσιμος ολοκληρωμένος μήνας καθορίζει δυναμικά και το σημείο έναρξης της πρόβλεψης. Το *forecast_origin* ορίζεται ως ο επόμενος μήνας μετά το *data_end*, επομένως δεν υπάρχει προκαθορισμένο έτος πρόβλεψης μέσα στον κώδικα. Όταν προστεθούν νέα δεδομένα, το χρονικό σημείο από το οποίο ξεκινά η πρόβλεψη μετακινείται αυτόματα.

Η αξιολόγηση γίνεται με χρονικά συνεπή ιστορική αξιολόγηση τύπου rolling-origin. Η σειρά δεν ανακατεύεται τυχαία σε σύνολα εκπαίδευσης και ελέγχου, επειδή κάτι τέτοιο θα επέτρεπε σε μελλοντικές παρατηρήσεις να επηρεάσουν την εκπαίδευση του παρελθόντος. Στο *evaluation.py* δημιουργούνται διαδοχικά παράθυρα αξιολόγησης, στα οποία κάθε περίοδος εκπαίδευσης προηγείται χρονικά της περιόδου ελέγχου. Η ρύθμιση επιτρέπει έως τρία παράθυρα ιστορικής αξιολόγησης. Οι μετρικές που χρησιμοποιούνται είναι MAE, RMSE, sMAPE και Bias, όπως αναλύθηκαν θεωρητικά στο Κεφάλαιο 2. Στο τεχνικό επίπεδο, η κύρια μετρική επιλογής είναι το MAE και οι RMSE, sMAPE και απόλυτο Bias χρησιμοποιούνται ως κριτήρια ισοβαθμίας.

Οι δύο στρατηγικές εκπαίδευσης πραγματοποιούνται ως διαφορετικές επιλογές του χρονικού ιστορικού που χρησιμοποιείται σε κάθε παράθυρο αξιολόγησης. Η αντιστοίχιση της στρατηγικής στο αντίστοιχο training window πραγματοποιείται στον κώδικα ως εξής:

```
ALL_HISTORY_STRATEGY = "expanding_all_available_history"
ROLLING_30_MONTH_STRATEGY = "30_month_history"
ROLLING_30_MONTH_WINDOW = 30

TRAINING_STRATEGIES = (
    ALL_HISTORY_STRATEGY,
    ROLLING_30_MONTH_STRATEGY,
)

def _training_window_for_strategy(
```

```

    training_strategy: str,
) -> int | None:
    if training_strategy == ALL_HISTORY_STRATEGY:
        return None

    if training_strategy == ROLLING_30_MONTH_STRATEGY:
        return ROLLING_30_MONTH_WINDOW

```

Στην περίπτωση *AllHistory*, η απουσία σταθερού ορίου επιτρέπει τη χρήση όλου του διαθέσιμου ιστορικού πριν από το αντίστοιχο *forecast origin*. Αντίθετα, στη στρατηγική *Last30Months* το σύνολο εκπαίδευσης περιορίζεται στους 30 αμέσως προηγούμενους μήνες. Με αυτόν τον τρόπο οι δύο στρατηγικές αξιολογούνται μέσα στον ίδιο μηχανισμό *backtesting* αλλά με διαφορετικό βάθος ιστορικής πληροφορίας.

Τα υπονήφια μοντέλα της τρέχουσας ρύθμισης περιλαμβάνουν Seasonal Naive, Weighted Seasonal Mean, ETS Additive, SARIMA, Seasonal Ridge και LSTM. Η εποχική περίοδος είναι 12 μήνες. Η ύπαρξη απλών μοντέλων αναφοράς μαζί με πιο σύνθετα μοντέλα είναι σκόπιμη: το σύστημα δεν θεωρεί ότι η μεγαλύτερη υπολογιστική πολυπλοκότητα συνεπάγεται αυτόματα καλύτερη πρόβλεψη, αλλά απαιτεί από κάθε υπονήφιο να αποδείξει την επίδοσή του στα ίδια ιστορικά παράθυρα αξιολόγησης.

Μετά την αξιολόγηση επιλέγεται η επίσημη ρύθμιση και δημιουργείται η τελική πρόβλεψη για τους καθορισμένους ορίζοντες. Η ρύθμιση περιλαμβάνει επίπεδο διαστήματος πρόβλεψης 95%, ενώ για μεταβλητές-στόχους που δεν επιτρέπεται να λαμβάνουν αρνητικές τιμές εφαρμόζονται οι αντίστοιχοι περιορισμοί. Για τις κρατήσεις και τα έσοδα, οι μήνες εκτός της ορισμένης περιόδου λειτουργίας μπορούν να μηδενίζονται σύμφωνα με την τελική πολιτική πρόβλεψης, αντί να θεωρούνται κανονικοί μήνες λειτουργίας.

Το αποτέλεσμα δεν παραμένει μόνο στη μνήμη της Python. Το *repository.py* μετατρέπει την απόκριση σε εγγραφές για τους πίνακες πρόβλεψης. Αρχικά δημιουργείται η εγγραφή του *forecast_runs*. Στη συνέχεια αποθηκεύονται οι εκτελέσεις των μοντέλων, οι μετρικές ιστορικής αξιολόγησης, οι επιλογές και οι προβλέψεις. Η εγγραφή ολοκληρώνεται σε κατάσταση *completed* μόνο όταν η διαδικασία μόνιμης αποθήκευσης έχει ολοκληρωθεί επιτυχώς, ενώ σε περίπτωση σφάλματος μπορεί να μεταβεί σε *failed*. Αν υπάρχει ήδη ολοκληρωμένη εκτέλεση με το ίδιο σταθερά παραγόμενο *run_key*, η μονάδα αποθήκευσης μπορεί να επαναχρησιμοποιήσει το υπάρχον αποτέλεσμα αντί να δημιουργήσει διπλότυπο.

Στην τρέχουσα λειτουργία η πρόβλεψη είναι ανεξάρτητη από τον καθαρισμό και ενεργοποιείται χειροκίνητα μέσω του API. Αυτό είναι σημαντικό για τη συνολική ροή: πρώτα δημιουργείται και αποθηκεύεται το αναλυτικό επίπεδο και στη συνέχεια ο χρήστης επιλέγει αν θα εκτελέσει πρόβλεψη για μία μεταβλητή-στόχο ή περισσότερες μεταβλητές-στόχους της επιλεγμένης μονάδας. Έτσι, η επιτυχία της εισαγωγής και του καθαρισμού

δεν εξαρτάται από τον χρόνο ή την έκβαση μιας διαδικασίας πρόβλεψης.

Το Κεφάλαιο αυτό περιγράφει τη γενική τεχνική υποδομή: πώς συνδέονται τα δύο άκρα της εφαρμογής, πώς αποθηκεύεται και επεξεργάζεται ένα αρχείο, πώς οργανώνεται το σχεσιακό μοντέλο και πώς δημιουργείται μία πρόβλεψη. Οι ειδικοί κανόνες με τους οποίους οι πραγματικές κρατήσεις, τα έσοδα και τα έξοδα μετατρέπονται στα πεδία των προβολών, καθώς και ο συγκεκριμένος υπολογισμός των μηνιαίων σειρών και της πληρότητας, εξετάζονται στο επόμενο κεφάλαιο.

Κεφάλαιο 4

Εφαρμογή της Μεθοδολογίας στα Δεδομένα της Μελέτης

4.1 Οργάνωση των δεδομένων και καθορισμός της μονάδας ανάλυσης

Μετά την ανάπτυξη της γενικής αρχιτεκτονικής και των μηχανισμών επεξεργασίας που παρουσιάστηκαν στο προηγούμενο κεφάλαιο, το επόμενο στάδιο αφορά την εφαρμογή τους στα πραγματικά δεδομένα της ξενοδοχειακής μονάδας. Το βασικό πρόβλημα σε αυτό το στάδιο δεν είναι πλέον ο τρόπος επικοινωνίας του frontend με το backend ή η μεταφόρτωση ενός αρχείου, αλλά ο καθορισμός του τρόπου με τον οποίο διαφορετικές εγγραφές θα μετατρέπονταν σε κοινές δομές. Τα αρχεία που χρησιμοποιήθηκαν δεν είχαν δημιουργηθεί με σκοπό την τροφοδότηση ενός ενιαίου συστήματος, αποτελούσαν εξαγωγές διαφορετικών υποσυστημάτων και περιέγραφαν διαφορετικές πλευρές της λειτουργίας της επιχείρησης. Για τον λόγο αυτό η πληροφορία δεν μπορούσε να αντιμετωπιστεί με έναν ενιαίο κανόνα καθαρισμού.

Τα δεδομένα διαχωρίστηκαν σε τρεις βασικές κατηγορίες: κρατήσεις, έσοδα και έξοδα. Ο διαχωρισμός αυτός δεν αποτελεί μόνο μια διάκριση αρχείων, αλλά αντιστοιχεί σε διαφορετική μονάδα παρατήρησης. Στα δεδομένα των κρατήσεων η βασική επιχειρησιακή οντότητα είναι η κράτηση και τα χαρακτηριστικά που τη συνοδεύουν, όπως οι ημερομηνίες άφιξης και αναχώρησης, η κατάσταση ακύρωσης, το δωμάτιο και τα οικονομικά στοιχεία της διαμονής. Στα έξοδα η αρχική μονάδα είναι η λογιστική γραμμή, καθώς η σημασία μιας εγγραφής εξαρτάται από τον λογιστικό κωδικό, την περιγραφή και την αιτιολογία της. Στα έσοδα, αντίθετα, η αρχική λογιστική γραμμή δεν είναι από μόνη της το τελικό αντικείμενο ανάλυσης. Ένα παραστατικό εμφανίζεται συνήθως σε περισσότερες γραμμές, οι οποίες περιλαμβάνουν τον λογαριασμό του πελάτη, την καθαρή αξία πώλησης, τον ΦΠΑ και ενδεχομένως άλλες κινήσεις. Συνεπώς, για τα έσοδα ήταν απαραίτητη

η συνθεση των λογιστικών γραμμών σε μία ενιαία εγγραφή ανά παραστατικό.

Πίνακας 4.1: Διαχωρισμός των δεδομένων και μονάδα ανάλυσης ανά κατηγορία.

Κατηγορία	Αρχική μονάδα δεδομένων	Τελική μονάδα ανάλυσης	Κύριος μετασχηματισμός
Κρατήσεις	Γραμμή εξαγωγής συστήματος κρατήσεων	Κράτηση	Τυποποίηση ημερομηνιών, κατάστασης, δωματίου και παραγόμενων πεδίων.
Έσοδα	Λογιστική γραμμή	Παραστατικό	Ομαδοποίηση λογιστικών γραμμών, υπολογισμός καθαρής αξίας, ΦΠΑ, τζίρου και κατηγορίας πελάτη.
Έξοδα	Λογιστική γραμμή	Ταξινομημένη γραμμή εξόδου	Ταξινόμηση μέσω λογιστικού κωδικού και αιτιολογίας, ένδειξη συμμετοχής/εξαίρεσης και κατανομή ΦΠΑ.

Παράλληλα, κάθε σύνολο δεδομένων συνδέθηκε με συγκεκριμένη ξενοδοχειακή μονάδα. Η σύνδεση αυτή πραγματοποιείται μέσω του *hotel_id*, το οποίο αποθηκεύεται μαζί με τη μεταφόρτωση και χρησιμοποιείται από το backend για την ανάκτηση της επίσημης εγγραφής του ξενοδοχείου. Τα στοιχεία του ξενοδοχείου δεν λαμβάνονται από ελεύθερο κείμενο του αρχείου, αλλά από τον κεντρικό πίνακα *hotels*, ώστε οι ίδιες ονομασίες και οι ίδιοι κωδικοί να χρησιμοποιούνται σε όλες τις κατηγορίες δεδομένων. Το backend προσθέτει τα στοιχεία της μονάδας κεντρικά στα καθαρισμένα δεδομένα μετά την εκτέλεση του αντίστοιχου μηχανισμού καθαρισμού, γεγονός που εξασφαλίζει ότι κρατήσεις, έσοδα και έξοδα ακολουθούν την ίδια λογική. Η υλοποίηση ανακτά το *hotel_id* από τη μεταφόρτωση, αναζητά το ξενοδοχείο στην βάση και χρησιμοποιεί από εκεί τα επίσημα στοιχεία του.

Η συγκεκριμένη οργάνωση αποτέλεσε την βάση για όλα τα επόμενα στάδια. Πριν υπολογιστούν συνολικά έσοδα, συνολικά έξοδα, αριθμός κρατήσεων ή προβλέψεις, έπρεπε πρώτα να οριστεί ποια είναι η μονάδα που μετριέται, ποια αρχική πληροφορία την δημιουργεί και σε ποια μορφή αποθηκεύεται.

4.2 Εναλλακτικές ονομασίες και δημιουργία του κανονικοποιημένου σχήματος

Ο γενικός μηχανισμός αναγνώρισης επικεφαλίδων και κανονικοποίησης περιγράφηκε στο Κεφάλαιο 3. Στην παρούσα ενότητα μας ενδιαφέρει κυρίως ο τρόπος με τον οποίο

εφαρμόστηκε ο μηχανισμός αυτός στα συγκεκριμένα δεδομένα. Η εφαρμογή δεν επιχείρησε να επιβάλει στα αρχεία ένα πρότυπο το οποίο θα έπρεπε να τηρεί ο χρήστης. Αντίθετα, δημιουργήθηκε για κάθε κατηγορία ένα σύνολο εσωτερικών, σταθερών ονομάτων και αντιστοιχίστηκαν σε αυτά οι διαφορετικές ονομασίες που είχαν παρατηρηθεί στις εξαγωγές.

Η έννοια της εναλλακτικής ονομασίας (alias) πρέπει να διαχωριστεί από την έννοια της αντιστοίχισης (mapping). Ένα alias απαντά στο ερώτημα «ποια στήλη του αρχείου εκφράζει αυτή την έννοια;». Μία επιχειρησιακή αντιστοίχιση απαντά στο ερώτημα «σε ποια επιχειρησιακή κατηγορία πρέπει να ενταχθεί αυτή η τιμή;». Για παράδειγμα, οι επικεφαλίδες *Ημερομηνία*, *Ημερ/νία*, *Date* και *TransactionDate* μπορούν να θεωρηθούν εναλλακτικές ονομασίες του ίδιου πεδίου ημερομηνίας. Αντίθετα, η αντιστοίχιση μιας αιτιολογίας σε ΟΤΑ ή ΤΑ αποτελεί αντιστοίχιση και εφαρμόζεται αφού έχει προηγηθεί η αναγνώριση της στήλης.

Στα έσοδα, το σχήμα οργανώθηκε γύρω από τα πεδία *date*, *document_no*, *account_code*, *account_description*, *cost_center*, *movement_reason*, *debit* και *credit*. Για παράδειγμα, το *date* μπορεί να προέλθει από επικεφαλίδες όπως *HMEPNIA*, *HMEPOMHNNIA*, *HMNIA* ή *DATE*, ενώ η αιτιολογία μπορεί να εμφανίζεται ως *AITIOΛΟΓΙΑΚΙΝΗΣΗΣ*, *AITIOΛΟΓΙΑ*, *REASON* ή *DESCRIPTION*. Η σημασία του σχεδιασμού αυτού είναι ότι μετά τη δημιουργία του *column_map* ο υπόλοιπος κώδικας δεν χρειάζεται να γνωρίζει ποια από τις παραπάνω επικεφαλίδες υπήρχε στο αρχείο.

Αντίστοιχη προσέγγιση χρησιμοποιήθηκε στα έξοδα. Το *transaction_date* μπορεί να αντιστοιχιστεί σε *Ημερ/νία*, *Ημερομηνία*, *Date* ή *TransactionDate*, το *document_number* σε *Παραστατικό*, *Αρ.Παραστατικού*, *Document*, *Invoice* ή *Voucher*, ενώ ο λογιστικός κωδικός αναγνωρίζεται μέσα από παραλλαγές όπως *Κωδικός*, *ΚωδικόςΛογαριασμού*, *AccountCode* και *Code*. Αυτό επέτρεψε να χρησιμοποιηθεί ο ίδιος μηχανισμός καθαρισμού σε περισσότερα έτη χωρίς να συνδεθεί η επιχειρησιακή λογική με μία ακριβή ονομασία στήλης.

Στις κρατήσεις το σύνολο των εναλλακτικών ονομασιών ήταν μεγαλύτερο, επειδή οι εξαγωγές περιείχαν περισσότερες έννοιες και υπήρχαν ελληνικές και αγγλικές παραλλαγές. Ο κωδικός κράτησης μπορεί να εμφανίζεται ως *Κωδικός*, *ΚωδικόςΚράτησης*, *Κωδ.Κράτησης*, *Αρ.Κράτησης*, *ReservationCode* ή *BookingID*. Παρόμοια, η αναμενόμενη άφιξη αναγνωρίζεται από τίτλους όπως *ΠιθανήΆφιξη*, *ΗμερομηνίαΆφιξης*, *ExpectedArrival*, *Arrival* ή *CheckIn*, ενώ η αναμενόμενη αναχώρηση υποστηρίζει αντίστοιχες παραλλαγές.

Η σημασία των εναλλακτικών ονομασιών είναι επομένως μεγαλύτερη από μία απλή μετονομασία στηλών. Αποτελούν το πρώτο επίπεδο διαχωρισμού μεταξύ της μορφής της πηγής και του μοντέλου της εφαρμογής. Εάν σε μία νέα εξαγωγή αλλάξει η ονομασία

μιας στήλης χωρίς να αλλάξει η επιχειρησιακή σημασία της, μπορεί να προστεθεί μία νέα εναλλακτική ονομασία χωρίς να τροποποιηθεί η λογική που ακολουθεί. Με αυτόν τον τρόπο οι κανόνες ταξινόμησης εσόδων και εξόδων παραμένουν ανεξάρτητοι από μικρές διαφορές των αρχείων.

4.3 Μετασχηματισμός των δεδομένων κρατήσεων

Στα δεδομένα κρατήσεων ο στόχος δεν ήταν να αλλάξει η έννοια της εγγραφής, αλλά να δημιουργηθεί μία σταθερή αναπαράσταση της κράτησης. Η αρχική εξαγωγή περιείχε αρκετά πεδία τα οποία δεν ήταν αναγκαία για τη συγκεκριμένη ανάλυση, ενώ σημαντικά πεδία μπορούσαν να εμφανίζονται με διαφορετικές ονομασίες. Αφού ολοκληρωθεί η αντιστοίχιση των εναλλακτικών ονομασιών, τα βασικά χρονικά πεδία μετατρέπονται σε κοινή μορφή ημερομηνίας και οι αριθμητικές τιμές σε κατάλληλους αριθμητικούς τύπους.

Πίνακας 4.2: Ενδεικτικές εναλλακτικές ονομασίες βασικών πεδίων των δεδομένων κρατήσεων.

Κανονικοποιημένο πεδίο	Ενδεικτικές αποδεκτές επικεφαλίδες
<i>booking_code</i>	Κωδικός, Κωδικός Κράτησης, Αρ. Κράτησης, Reservation Code, Booking ID, Code.
<i>booking_date</i>	Κράτηση, Ημερομηνία Κράτησης, Ημ/νία Κράτησης, Booking Date, Reservation Date.
<i>expected_arrival</i>	Πιθανή Άφιξη, Πιθ. Άφιξη, Expected Arrival, Arrival, Check In.
<i>expected_departure</i>	Πιθανή Αναχώρηση, Πιθ. Αναχώρηση, Expected Departure, Check Out.
<i>cancellation_date</i>	Ακύρωση, Ημερομηνία Ακύρωσης, Cancellation Date, Cancelled At.
<i>actual_arrival</i>	Άφιξη, Πραγματική Άφιξη, Actual Arrival.
<i>actual_departure</i>	Αναχώρηση, Πραγματική Αναχώρηση, Actual Departure.
<i>total_guests</i>	Σύνολο Ατόμων, Άτομα, Total Guests, Guests, Pax.
<i>adults</i>	Ενήλικες, Adults.
<i>nights</i>	Ημ. Διαμονής, Ημέρες Διαμονής, Διανυκτερεύσεις, Nights.
<i>room_number</i>	Δωμάτιο, Δωμάτιο.1, Αριθμός Δωματίου, Room Number, Room No, Room #.

Ιδιαίτερη σημασία δόθηκε στη διάκριση μεταξύ αναμενόμενων και πραγματικών ημερομηνιών. Στο τελικό μοντέλο διατηρούνται η ημερομηνία δημιουργίας της κράτησης, οι αναμενόμενες ημερομηνίες άφιξης και αναχώρησης, οι πραγματικές ημερομηνίες και η ημερομηνία ακύρωσης. Η ημερομηνία κράτησης σχετίζεται με το πότε δημιουργήθηκε η ζήτηση, ενώ η ημερομηνία άφιξης με το ποτε η κράτηση προσήλθε στην ξενοδοχειακή μονάδα.

Από τα αρχικά δεδομένα δημιουργούνται και παράγωγα πεδία. Η κατάσταση της κράτησης προσδιορίζεται με βάση την ύπαρξη ημερομηνίας ακύρωσης, ώστε μία εγγραφή να χαρακτηρίζεται ως ενεργή ή ακυρωμένη. Παράλληλα, όταν υπάρχουν συνολικά άτομα και ενήλικες, ο αριθμός των παιδιών μπορεί να εξαχθεί ως η μη αρνητική διαφορά των δύο τιμών. Ο αριθμός δωματίου καθορίζεται σε ξεχωριστό πεδίο *room_number*, ώστε τιμές που έχουν εξαχθεί από το Excel ως αριθμητικά αντικείμενα να μην δημιουργούν διαφορετικές κατηγορίες για το ίδιο δωμάτιο.

$$booking_status = \begin{cases} \text{Ακυρώθηκε,} & \text{αν } cancellation_date \neq NULL, \\ \text{Ενεργή,} & \text{αν } cancellation_date = NULL, \end{cases}$$

$$children = \max(total_guests - adults, 0).$$

Στην επόμενη φάση εφαρμόζεται αντιστοίχιση στους τύπους δωματίων. Το αρχικό *room_type* διατηρείται, επειδή αποτελεί την αρχική πληροφορία του συστήματος κρατήσεων, αλλά δεν χρησιμοποιείται μόνο του για παρουσίαση. Μέσω της αντιστοίχισης προστίθενται τα *room_type_name*, *room_type_short_name*. Έτσι το σύστημα διατηρεί ταυτόχρονα την πηγή και μία σταθερή ονομασία. Τα συγκεκριμένα πεδία εμφανίζονται στο τελικό *v_bookings_clean* μαζί με το *hotel_id*, το *hotel_name* και το *hotel_code*.

Για την πιο σαφή αποτύπωση του τελικού σχήματος, τα βασικά πεδία του αναλυτικού επιπέδου των κρατήσεων μπορούν να ομαδοποιηθούν ως εξής:

- **Ταυτοποίηση:** *row_id*, *source_file_id*, *row_number*, *booking_code*.
- **Χρονικά πεδία:** *booking_date*, *expected_arrival*, *expected_departure*, *arrival_date*, *departure_date*, *cancellation_date*.
- **Κατάσταση και πελάτες διαμονής:** *booking_status*, *adults*, *children*, *total_guests*, *meal_guests*.
- **Διαμονή και δωμάτιο:** *stay_days*, *expected_days*, *room_number*, *room_type*, *room_type_name*, *room_type_short_name*.

- **Οικονομικά πεδία:** *payable_amount*, *daily_amount*, *room_amount*, *stay_total*, *average_daily_price* και τα επιμέρους ποσά υπηρεσιών και ακυρώσεων.
- **Ιχνηλασιμότητα:** *full_cleaned_data*, *full_raw_data*, *row_created_at*, *hotel_id*, *hotel_name*, *hotel_code*.

Ένα επιπλέον πρόβλημα των δεδομένων κρατήσεων ήταν η πιθανή επανάληψη της ίδιας κράτησης όταν υπήρχαν περισσότερες μεταφορτώσεις. Στην τελική προβολή εφαρμόζεται *row_number()* σε συνδυασμό στοιχείων της μονάδας, του κωδικού κράτησης, του έτους, των ημερομηνιών άφιξης και αναχώρησης και του δωματίου. Οι εγγραφές ταξινομούνται με προτεραιότητα στις νεότερες επεξεργασίες και διατηρείται η εγγραφή με *duplicate_rank=1*. Η τεχνική αυτή είναι σημαντική για την ανάλυση, διότι εμποδίζει μία νεότερη εξαγωγή της ίδιας κράτησης να αυξήσει τον αριθμό των κρατήσεων.

Το τελικό αποτέλεσμα είναι μία σταθερή επιχειρησιακή κράτηση, στην οποία έχουν διαχωριστεί οι χρονικές έννοιες, έχει αποδοθεί κατάσταση, έχουν τυποποιηθεί τα στοιχεία δωματίου και έχουν προστεθεί τα μεταδεδομένα της ξενοδοχειακής μονάδας. Το συγκεκριμένο επίπεδο αποτελεί στη συνέχεια τη βάση για τον υπολογισμό κρατήσεων ανά μήνα και για την δημιουργία των χρονοσειρών.

4.4 Επεξεργασία των εσόδων και δημιουργία εγγραφής ανά παραστατικό

Η επεξεργασία των εσόδων ήταν το σημείο στο οποίο η απόσταση μεταξύ αρχικής και τελικής εγγραφής ήταν μεγαλύτερη. Τα αρχεία δεν παρείχαν μία έτοιμη γραμμή με πεδία «καθαρό έσοδο», «ΦΠΑ», «κατηγορία πελάτη» και «συνολικός τζίρος». Οι πληροφορίες αυτές έπρεπε να εξαχθούν από τη δομή της λογιστικής κίνησης.

Πίνακας 4.3: Βασικά κανονικοποιημένα πεδία και εναλλακτικές ονομασίες των αρχείων εσόδων.

Κανονικοποιημένο πεδίο	Ενδεικτικές εναλλακτικές ονομασίες
<i>date</i>	ΗΜΕΡ/ΝΙΑ, ΗΜΕΡΟΜΗΝΙΑ, ΗΜ ΝΙΑ, DATE.
<i>document_no</i>	ΠΑΡΑΣΤΑΤΙΚΟ, DOCUMENT, DOC.
<i>account_code</i>	ΚΩΔΙΚΟΣ, ΚΩΔΙΚΟΣ ΛΟΓΑΡΙΑΣΜΟΥ, ACCOUNT CODE.
<i>account_description</i>	ΠΕΡΙΓΡΑΦΗ ΛΟΓΑΡΙΑΣΜΟΥ, ΠΕΡΙΓΡΑΦΗ, ACCOUNT DESCRIPTION.
<i>cost_center</i>	ΚΚ, ΚΕΝΤΡΟ ΚΟΣΤΟΥΣ, COST CENTER.
<i>movement_reason</i>	ΑΙΤΙΟΛΟΓΙΑ ΚΙΝΗΣΗΣ, ΑΙΤΙΟΛΟΓΙΑ, REASON, DESCRIPTION.
<i>debit</i>	ΧΡΕΩΣΗ, DEBIT.
<i>credit</i>	ΠΙΣΤΩΣΗ, CREDIT.

Αρχικά κάθε λογιστική γραμμή διατηρεί την ημερομηνία, τον αριθμό παραστατικού, τον κωδικό και την περιγραφή λογαριασμού, την αιτιολογία, τη χρέωση και την πίστωση. Παράλληλα καταγράφεται το *source_excel_row*, δηλαδή ο αριθμός της γραμμής στο αρχικό αρχείο. Με αυτόν τον τρόπο η τελική εγγραφή μπορεί να συνδεθεί όχι μόνο με το αρχείο αλλά και με τις συγκεκριμένες γραμμές από τις οποίες προέκυψε.

Στη συνέχεια κάθε λογιστική γραμμή αποκτά *line_type*. Οι κωδικοί 70-73 αντιμετωπίζονται ως γραμμές καθαρής πώλησης, ο 54 ως ΦΠΑ, ο 30 ως συνολική κίνηση πελάτη, ο 53 ή αντίστοιχες αιτιολογίες ως φιλοδωρήματα, ενώ υπάρχουν ξεχωριστές κατηγορίες για τόκους και εσωτερικές λογιστικές προσαρμογές. Η ταξινόμηση αυτή δεν αποτελεί ακόμη την τελική κατηγορία εσόδου. Χρησιμοποιείται για να προσδιοριστεί ποιον οικονομικό ρόλο παίζει κάθε γραμμή μέσα στο ίδιο παραστατικό.

Πίνακας 4.4: Βασική ταξινόμηση λογιστικών γραμμών των εσόδων.

Κωδικός θήκη	ή συν-	Τύπος γραμμής	Ερμηνεία
54...		<i>vat</i>	Γραμμή ΦΠΑ του παραστατικού.
30...		<i>customer_total</i>	Συνολική κίνηση λογαριασμού πελάτη.
53... ή σχετική αιτιολογία		<i>tips</i>	Φιλοδώρημα ή αντίστοιχη κίνηση.
70...-73...		<i>sales_net</i>	Καθαρή αξία πώλησης.
74...		<i>subsidy_income</i>	Έσοδο επιδότησης.
76...		<i>interest_income</i>	Έσοδο τόκων.
62..., 78...		<i>internal_adjustment</i>	Εσωτερική λογιστική προσαρμογή.
Ιδιόχρηση		<i>self_use_adjustment</i>	Λογιστική προσαρμογή ιδιόχρησης.
Άλλη περίπτωση		<i>other</i>	Γραμμή που δεν εντάσσεται στις προηγούμενες κατηγορίες.

Το κρίσιμο επόμενο στάδιο είναι η ομαδοποίηση με βάση την ημερομηνία συναλλαγής και τον αριθμό παραστατικού. Αν ένα παραστατικό αποτελείται, για παράδειγμα, από μία γραμμή πελάτη, μία γραμμή καθαρής πώλησης και μία γραμμή ΦΠΑ, οι τρεις γραμμές μετατρέπονται σε μία τελική εγγραφή. Στην εγγραφή αυτή υπολογίζονται *net_amount*, *vat_amount*, *total_turnover*, *tips_amount*, *accounting_debit_total*, *accounting_credit_total* και *accounting_movement_total*. Παράλληλα ενώνονται οι λογιστικοί κωδικοί που συμμετείχαν, οι τύποι των γραμμών, οι αρχικές αιτιολογίες και οι περιγραφές. Η τελική δομή του μηχανισμού καθαρισμού εσόδων περιλαμβάνει ακριβώς αυτά τα πεδία.

Η ομαδοποίηση των λογιστικών γραμμών και ο υπολογισμός των βασικών οικονομικών ποσών του παραστατικού πραγματοποιούνται ενδεικτικά ως εξής:

```
group_cols = ["transaction_date", "document_no"]

for (transaction_date, document_no), group in df.groupby(
    group_cols,
    dropna=False,
):
    net_amount = round(
        float(group["net_line_amount"].sum()), 2
    )
```

```

vat_amount = round(
    float(group["vat_line_amount"].sum()), 2
)
customer_total = round(
    float(group["customer_total_line_amount"].sum()), 2
)
tips_amount = round(
    float(group["tips_line_amount"].sum()), 2
)

if customer_total != 0:
    total_turnover = customer_total
else:
    total_turnover = round(
        net_amount + vat_amount + tips_amount,
        2,
    )

```

Το απόσπασμα δείχνει ότι οι επιμέρους λογιστικές γραμμές δεν αντιμετωπίζονται ως ανεξάρτητες εγγραφές εσόδου. Αρχικά ομαδοποιούνται ανά ημερομηνία και αριθμό παραστατικού και στη συνέχεια τα επιμέρους ποσά συγκεντρώνονται σε μία τελική εγγραφή.

Για συνοπτική παρουσίαση, στην τελική εγγραφή του παραστατικού διατηρούνται:

- *transaction_date* και *document_no*, που προσδιορίζουν την οικονομική εγγραφή.
- *customer_category*, *include_in_turnover* και *exclude_reason*, που περιγράφουν την επιχειρησιακή ταξινόμηση και τη συμμετοχή στον τζίρο.
- *net_amount*, *vat_amount*, *total_turnover* και *tips_amount*, που περιγράφουν τα βασικά ποσά του παραστατικού.
- *accounting_debit_total*, *accounting_credit_total* και *accounting_movement_total*, που διατηρούν την εικόνα της λογιστικής κίνησης.
- *account_codes*, *line_types* και *source_excel_rows*, που επιτρέπουν να εντοπιστούν οι γραμμές που συμμετείχαν στην ανασύνθεση.
- *raw_reason_combined* και *raw_description_combined*, που διατηρούν ενιαία την αρχική λεκτική πληροφορία του παραστατικού.

Η επιλογή να διατηρηθούν τόσο τα ποσά όσο και η προέλευσή τους ήταν σκόπιμη. Μία εγγραφή με *net_amount*=2.008,81 δεν αποθηκεύεται ως ένας αριθμός χωρίς εξήγηση. Μαζί της μπορούν να διατηρούνται οι κωδικοί 30..., 54..., 73..., η αρχική αιτιολογία και οι γραμμές του Excel από τις οποίες προέκυψε. Στα πραγματικά δεδομένα, για παράδειγμα,

παραστατικό με αιτιολογία *Booking.com B.V.* εμφανίζεται με *customer_category=ota*, *line_types=customer_total,sales_net,vat*, καθαρή αξία, ΦΠΑ και συνολικό turnover.

4.4.1 Κατηγοριοποίηση του πελάτη

Μετά τη δημιουργία του παραστατικού εφαρμόζεται ξεχωριστή διαδικασία κατηγοριοποίησης πελάτη. Η συγκεκριμένη διαδικασία σχεδιάστηκε έτσι ώστε η βάση αντιστοιχίσεων να προηγείται των γενικών ευρετικών κανόνων. Το backend ανακτά από τον πίνακα *customer_mappings* εγγραφές και διαβάζει τα πεδία *match_text*, *match_text_normalized*, *customer_category*, *match_type*, *priority* και *is_active*, ταξινομημένα κατά αύξουσα προτεραιότητα.

Η ύπαρξη του *priority* είναι σημαντική όταν ένα κείμενο θα μπορούσε να ταιριάζει σε περισσότερους από έναν κανόνες. Οι αντιστοιχίσεις εξετάζονται με τη σειρά προτεραιότητας. Το *match_type* επιτρέπει τόσο ακριβή αντιστοίχιση (*exact*) όσο και αντιστοίχιση όταν το ζητούμενο κείμενο περιέχεται στην αιτιολογία (*contains*).

Η προτεραιότητα των αποθηκευμένων αντιστοιχίσεων έναντι των γενικότερων κανόνων αποτυπώνεται και στη σειρά εκτέλεσης της διαδικασίας:

```
mappings = normalize_mapping_rows(customer_mappings)

mapped_category = classify_customer_from_mapping(
    combined,
    mappings,
)

if mapped_category:
    return mapped_category

if any(keyword in combined for keyword in OTA_KEYWORDS):
    return "ota"

if any(keyword in combined for keyword in TA_KEYWORDS):
    return "ta"

if any(
    keyword in combined
    for keyword in INDIVIDUAL_KEYWORDS
):
    return "individual_customer"

if any(
    contains_token(reason_doc, token)
    for token in BUSINESS_TOKENS
```

```
) :
    return "business"

return "other"
```

Με τη σειρά αυτή, όταν υπάρχει ενεργή αντιστοίχιση στη βάση, η αντίστοιχη κατηγορία χρησιμοποιείται άμεσα. Οι γενικοί κανόνες εφαρμόζονται μόνο όταν δεν έχει βρεθεί συγκεκριμένο mapping και λειτουργούν ως διαδοχικά επίπεδα εναλλακτικής κατηγοριοποίησης.

Εάν δεν βρεθεί ενεργή αντιστοίχιση, εφαρμόζονται γενικότεροι κανόνες. Υπάρχουν λέξεις-κλειδιά για OTA, TA, πελάτες λιανικής και εταιρικές μορφές. OTA ενδείξεις αποτελούν, μεταξύ άλλων, οι Booking, Expedia, Agoda, Airbnb και Hotelbeds, ενώ για ταξιδιωτικά πρακτορεία εξετάζονται όροι όπως *TRAVEL*, *TOUR*, *AGENCY*, *DMC* και ελληνικά αντίστοιχα. Για επιχειρηματικούς πελάτες χρησιμοποιούνται ενδείξεις εταιρικού τύπου όπως IKE, AE, ΕΠΕ, ΟΕ, ΕΕ και LTD. Ο τύπος του παραστατικού λειτουργεί ως πρόσθετος κανόνας, ώστε τιμολόγια να μπορούν να καταταχθούν σε Business και αποδείξεις λιανικής σε individual customer όταν δεν υπάρχει άλλη ένδειξη.

Η σειρά εφαρμογής των κανόνων μπορεί να συνοψιστεί ως εξής:

1. αναγνώριση ειδικών περιπτώσεων, όπως η ιδιόχρηση,
2. αναζήτηση στις ενεργές αντιστοιχίσεις με βάση την προτεραιότητα,
3. έλεγχος γενικών λέξεων-κλειδιών OTA,
4. έλεγχος λέξεων-κλειδιών TA,
5. αναγνώριση πελατών λιανικής,
6. έλεγχος εταιρικών ενδείξεων,
7. αξιολόγηση του τύπου παραστατικού,
8. τελική κατάταξη στην κατηγορία *other* όταν δεν υπάρχει ισχυρότερη ένδειξη.

Η τελική κατηγοριοποίηση οδηγεί στις αναλυτικές ομάδες Direct, OTA, TA, Business και Other. Η αρχική ονομασία δεν απορρίπτεται. Διατηρείται ξεχωριστά ώστε να είναι δυνατή η αναζήτηση του πραγματικού πελάτη, ενώ η κατηγορία χρησιμοποιείται για τις ομαδοποιήσεις.

4.4.2 Συμπεριλαμβανόμενα και εξαιρούμενα έσοδα (included/excluded)

Η δημιουργία ενός παραστατικού δεν σημαίνει αυτόματα ότι η καθαρή του αξία πρέπει να συμπεριληφθεί στον πραγματικό τζίρο. Στα λογιστικά δεδομένα εντοπίστηκαν εγγραφές που αντιπροσώπευαν ιδιόχρηση, εκκαθαρίσεις ΦΠΑ, δημοτικά τέλη, τέλη παρεπιδημούντων, διορθώσεις ή άλλες λογιστικές κινήσεις. Για τον λόγο αυτό δημιουργήθηκε το *include_in_turnover* και το συνοδευτικό *exclude_reason*.

Ενδεικτικές ειδικές περιπτώσεις που διατηρούνται αλλά μπορούν να εξαιρεθούν από τον υπολογισμό του τζίρου είναι:

- *self_use_adjustment*,
- *municipal_tax_adjustment*,
- *tips_reconciliation_adjustment*,
- *internal_vat_clearing_adjustment*,
- *internal_goods_granted_adjustment*.

Η διάκριση αυτή αποτελεί σημαντικό μέρος του τελικού μοντέλου, επειδή επιτρέπει να διατηρείται μία λογιστική κίνηση στη βάση χωρίς να συμβάλλει στο KPI του included revenue. Αν η εγγραφή διαγραφόταν ολοκληρωτικά, θα χανόταν η δυνατότητα ελέγχου. Αν, αντίθετα, αθροιζόταν μαζί με τις πραγματικές πωλήσεις, ο οικονομικός δείκτης θα ήταν λανθασμένος. Η λύση ήταν επομένως η διατήρηση της εγγραφής και η επισήμανση της συμμετοχής της.

4.5 Κατηγοριοποίηση των εξόδων και δημιουργία των οικονομικών πεδίων

Στα έξοδα η βασική δυσκολία ήταν η διάκριση μεταξύ πραγματικής λειτουργικής δαπάνης και λογιστικής κίνησης που εμφανίζεται στο αρχείο. Η λογική ταξινόμησης βασίστηκε κατά κύριο λόγο στον λογιστικό κωδικό και στις αιτιολογίες και όχι στην δημιουργία μεγάλης λίστας συγκεκριμένων προμηθευτών. Η επιλογή αυτή ήταν απαραίτητη ώστε ο μηχανισμός καθαρισμού να είναι επαναχρησιμοποιήσιμος σε μελλοντικά δεδομένα.

Πίνακας 4.5: Βασικές εναλλακτικές ονομασίες των λογιστικών στηλών στα αρχεία εξόδων.

Κανονικοποιημένο πεδίο	Ενδεικτικές εναλλακτικές ονομασίες
<i>transaction_date</i>	Ημερ/νία, Ημερομηνία, Date, Transaction Date.
<i>document_number</i>	Παραστατικό, Αρ. Παραστατικού, Document, Invoice, Voucher.
<i>account_code</i>	Κωδικός, Κωδικός Λογαριασμού, Account Code, Code.
<i>account_description</i>	Περιγραφή Λογαριασμού, Περιγραφή, Account Description.
<i>cost_center</i>	ΚΚ, Κ.Κ., Κέντρο Κόστους, Cost Center.
<i>reason</i>	Αιτιολογία Κίνησης, Αιτιολογία, Reason, Movement Reason.
<i>debit</i>	Χρέωση, Debit.
<i>credit</i>	Πίστωση, Credit.

Μετά την ανάγνωση της λογιστικής γραμμής δημιουργούνται τα προθέματα του λογαριασμού και το *signed_amount*. Η βασική έννοια είναι ότι το διψήφιο πρόθεμα δίνει την οικογένεια του λογαριασμού, ενώ σε περιπτώσεις στις οποίες απαιτείται μεγαλύτερη ακρίβεια εξετάζεται και μεγαλύτερο τμήμα του κωδικού. Παράλληλα χρησιμοποιούνται η περιγραφή και η αιτιολογία, έτσι ώστε δύο λογαριασμοί της ίδιας γενικής οικογένειας να μπορούν να καταλήξουν σε διαφορετικές κατηγορίες.

Σε ορισμένες περιπτώσεις χρησιμοποιούνται επιπλέον συγκεκριμένες λεκτικές ενδείξεις που έχουν επιβεβαιωθεί από τα διαθέσιμα δεδομένα. Χαρακτηριστικό παράδειγμα αποτελεί ο λογαριασμός 24, στον οποίο μπορεί να εμφανίζονται τόσο αγορές τροφίμων και ποτών όσο και αγορές αναλωσίμων. Για τον λόγο αυτό, πέρα από τις γενικές λέξεις-κλειδιά, ο μηχανισμός περιλαμβάνει και επιβεβαιωμένες ονομασίες προμηθευτών ως συμπληρωματικό κανόνα ταξινόμησης. Ενδεικτικό απόσπασμα της σχετικής λογικής είναι το ακόλουθο:

```
if prefix_2 == "24":
    if has_any(
        text,
        [
            .....
            .....
            .....
            .....
            "CARREFOUR",
            "PETITS GRECS",
```

```

        "ΒΑΣΙΛΟΠΟΥΛΟΣ",
        "ΛΙΝΤΛ",
        "ΑΒΑΝΤΙΣ",
        .....
        .....
        .....
        .....
    ],
):
    return classification_result(
        "f&b",
        "τρόφιμα / ποτά",
        True,
        "expense",
        "confirmed_fnb_24_supplier",
        "high",
    )

```

Η χρήση των συγκεκριμένων ονομασιών δεν αποτελεί τον βασικό μηχανισμό κατηγοριοποίησης των εξόδων, αλλά συμπληρωματικό κανόνα για περιπτώσεις στις οποίες ο λογιστικός κωδικός από μόνος του δεν αρκεί για ασφαλή διάκριση. Οι γενικοί κανόνες που βασίζονται στον κωδικό, στην περιγραφή και στην αιτιολογία εξακολουθούν να αποτελούν το κύριο επίπεδο ταξινόμησης.

Ο μηχανισμός ταξινόμησης δεν επιστρέφει μόνο το όνομα της κατηγορίας. Για κάθε γραμμή δημιουργεί ένα σύνολο συμπληρωματικών πεδίων: *line_type*, *expense_category*, *expense_subcategory*, *included_in_expenses*, *exclude_reason*, *rule_matched* και *confidence*. Για παράδειγμα, μία γραμμή ΦΠΑ αναγνωρίζεται μέσω του προθέματος 54, λαμβάνει *line_type=vat*, κατηγορία ΦΠΑ, εξαιρείται από το άθροισμα των λειτουργικών εξόδων και διατηρεί τον κανόνα της ταξινόμησης.

Το *rule_matched* έχει ιδιαίτερη σημασία στη συγκεκριμένη εφαρμογή. Αντί το αποτέλεσμα να δηλώνει μόνο ότι μία εγγραφή ανήκει στο marketing, αποθηκεύεται και ο κανόνας που οδήγησε στην απόφαση. Αυτό επιτρέπει να εξεταστεί μετέπειτα αν η κατηγορία προέκυψε από συγκεκριμένο πρόθεμα, από λέξη-κλειδί στην περιγραφή ή από κανόνα. Το *confidence* δεν αποτελεί πιθανότητα ενός εκπαιδευμένου μοντέλου, αλλά μεταδεδομένο του μηχανισμού ταξινόμησης που βασίζεται σε κανόνες και εκφράζει τον βαθμό βεβαιότητας που έχει αποδοθεί στον κανόνα.

Η παραπάνω λογική υλοποιείται, ώστε κάθε κανόνας να επιστρέφει μαζί με την κατηγορία και τις πληροφορίες που αιτιολογούν την απόφαση:

```

def classification_result(
    category,
    subcategory,

```

```

        included,
        line_type,
        rule,
        confidence,
        exclude_reason=None,
    ):
        return {
            "expense_category": category,
            "expense_subcategory": subcategory,
            "included_in_expenses": included,
            "line_type": line_type,
            "exclude_reason": exclude_reason,
            "rule_matched": rule,
            "confidence": confidence,
        }

```

Για παράδειγμα, διαφορετικά λογιστικά προθέματα μπορούν να οδηγήσουν τόσο σε διαφορετική κατηγορία όσο και σε διαφορετική απόφαση συμμετοχής στα έξοδα:

```

if prefix_2 == "54":
    return classification_result(
        "λοιπα εξοδα",
        "φπα",
        False,
        "vat",
        "prefix_54_vat",
        "high",
        "Γραμμή ΦΠΑ",
    )

if prefix_2 in ["04", "08"]:
    return classification_result(
        "λοιπα εξοδα",
        "λογαριασμοί τάξεως / reverse charge",
        False,
        "excluded",
        f"prefix_{prefix_2}_excluded",
        "high",
        "Πληροφοριακός λογαριασμός, όχι πραγματικό έξοδο",
    )

```

Στα παραπάνω παραδείγματα η διαδικασία δεν επιστρέφει μόνο μία κατηγορία εξόδου. Καθορίζει ταυτόχρονα αν η γραμμή συμμετέχει στα λειτουργικά έξοδα, καταγράφει τον κανόνα που εφαρμόστηκε και, όπου απαιτείται, αποθηκεύει την αιτία εξαίρεσης.

Στους λογαριασμούς που δεν θεωρούνται άμεσα λειτουργικά έξοδα περιλαμβάνονται, μεταξύ άλλων, λογαριασμοί προμηθευτών, ταμείου και τραπεζών, μεταβατικοί λογαριασμοί και λογιστικές τακτοποιήσεις. Ο μηχανισμός ταξινόμησης διαχωρίζει αυτές τις εγγραφές και αποδίδει κατάλληλο *exclude_reason*. Αυτό αποτρέπει ένα συνηθισμένο πρόβλημα λογιστικών εξαγωγών: την πολλαπλή μέτρηση της ίδιας οικονομικής εγγραφής.

Οι βασικοί κανόνες ανά λογιστικό πρόθεμα που χρησιμοποιούνται στη συγκεκριμένη περίπτωση συνοψίζονται στον Πίνακα 4.6.

Πίνακας 4.6: Βασική λογική κατηγοριοποίησης των εξόδων ανά λογιστικό πρόθεμα.

Πρόθεμα / συν- θήκη	Τελική λογική
10-16	Πάγια και εξοπλισμός, με ειδικό χειρισμό αντίθετων λογαριασμών και αποσβέσεων.
24	F&B ή αναλώσιμα, ανάλογα με τον υποκωδικό και την περιγραφή.
25, 28	ΔΕΚΟ/καύσιμα ή λοιπές αγορές, ανάλογα με το περιεχόμενο της εγγραφής.
54	ΦΠΑ, ο οποίος δεν αντιμετωπίζεται ως αυτόνομο λειτουργικό έξοδο.
60	Μισθοδοσία, εισφορές και σχετικές παροχές.
61	Τραπεζικά, marketing, διοικητικές, λογιστικές, τεχνικές ή λοιπές αμοιβές τρίτων, ανάλογα με τον υποκωδικό και τις λέξεις-κλειδιά.
62	ΔΕΚΟ, ενοίκια, ασφάλειες, συντήρηση, F&B ή λοιπές παροχές.
63	Φόροι και τέλη.
64	Ιδιόχρηση, marketing, αναλώσιμα, ΔΕΚΟ, F&B ή λοιπές δαπάνες, με βάση υποκωδικό και αιτιολογία.
65	Τραπεζικές και χρηματοοικονομικές δαπάνες.
66	Αποσβέσεις.

Για τις πραγματικές δαπάνες χρησιμοποιήθηκαν κατηγορίες όπως μισθοδοσία, F&B, αναλώσιμα, ΔΕΚΟ, marketing, τραπεζικά έξοδα, ενοίκια, ασφάλειες, συντήρηση και επισκευές, φόροι και τέλη, αμοιβές τρίτων και πάγια. Η απόφαση βασίζεται στον συνδυασμό λογιστικού κωδικού, περιγραφής και αιτιολογίας. Για παράδειγμα, η ομάδα των λέξεων-κλειδίων marketing περιλαμβάνει ενδείξεις σχετικές με Booking, Expedia, Airbnb, Agoda, Google, commissions ενώ η ομάδα υπηρεσιών κοινής ωφέλειας περιλαμβάνει λέξεις σχετικές με ηλεκτρική ενέργεια, ύδρευση, τηλεφωνία, Internet και καύσιμα.

Η διάκριση αυτή ήταν ιδιαίτερα σημαντική για λογαριασμούς όπως η ομάδα 64, όπου η γενική λογιστική κατηγορία είναι πολύ ευρεία για επιχειρησιακή ανάλυση. Η αιτιολογία και οι υποκωδικοί χρησιμοποιήθηκαν ώστε διαφορετικές εγγραφές της ίδιας γενικής ομάδας να μην παρουσιάζονται όλες ως «λοιπά έξοδα».

Ιδιαίτερη αντιμετώπιση απαιτήθηκε επίσης για τα πάγια και τις αποσβέσεις. Οι αγορές εξοπλισμού μπορούν να αναγνωριστούν από τους αντίστοιχους λογαριασμούς και να αποκτήσουν υποκατηγορίες όπως εξοπλισμός, σκεύη, ιματισμός, κτίρια/εγκαταστάσεις ή ηλεκτρονικός εξοπλισμός. Ταυτόχρονα, οι αντίθετοι λογαριασμοί και οι σωρευμένες αποσβέσεις διακρίνονται από την αρχική αγορά ώστε να μην αντιμετωπίζονται ως το ίδιο είδος δαπάνης.

4.5.1 Κατανομή του ΦΠΑ

Το *vat_amount* των εξόδων δεν προέκυψε απλώς με αντιγραφή της γραμμής 54. Η γραμμή ΦΠΑ αποτελεί ξεχωριστή λογιστική γραμμή του παραστατικού και, εάν αθροίζοταν ως αυτόνομο expense, το reporting θα εμφάνιζε αφενός την καθαρή δαπάνη στην κατηγορία της και αφετέρου έναν δεύτερο ανεξάρτητο λογαριασμό ΦΠΑ.

Για τον λόγο αυτό τα δεδομένα οργανώθηκαν ανά παραστατικό και υπολογίστηκαν ποσά όπως *document_net_amount*, *document_vat_amount* και ο κατανεμητέος ΦΠΑ. Ο ΦΠΑ κατανέμεται στις πραγματικές included expense γραμμές ανάλογα με τη συμμετοχή της καθαρής τους αξίας στη συνολική καθαρή αξία του παραστατικού.

Έτσι η τελική εγγραφή εξόδου περιλαμβάνει τόσο την καθαρή αξία όσο και τον ΦΠΑ που αντιστοιχεί στην επιχειρησιακή κατηγορία, χωρίς να χάνεται η αρχική λογιστική γραμμή που χρησιμοποιήθηκε για τον υπολογισμό.

Το αποτέλεσμα του μηχανισμού καθαρισμού περιλαμβάνει την προέλευση της γραμμής, ημερομηνία, παραστατικό, λογιστικό κωδικό και περιγραφή, αιτιολογία, κατηγορία και υποκατηγορία εξόδου, ένδειξη συμμετοχής, λόγο εξαίρεσης, τον κανόνα ταξινόμησης, το *signed_amount*, τις χρεώσεις και πιστώσεις και τα *net_amount*, *vat_amount* και *gross_amount*. Τα πεδία αυτά αποτελούν το επιχειρησιακό αποτέλεσμα του καθαρισμού και είναι αυτά που στη συνέχεια μεταφέρονται στο επίπεδο της βάσης.

Για πληρέστερη αποτύπωση, τα βασικά καθαρισμένα πεδία που προκύπτουν για τα έξοδα είναι:

- **Προέλευση:** *source_file*, *source_year*, *source_row_number*, *transaction_date*, *document_number*, *document_group_key*.
- **Λογιστική πληροφορία:** *account_code*, *account_prefix_2*, *account_prefix_5*, *account_description*, *cost_center*, *reason*, *debit*, *credit*, *signed_amount*.

- **Ταξινόμηση:** *line_type, expense_category, expense_subcategory, included_in_expenses, exclude_reason, rule_matched, confidence.*
- **Οικονομικά ποσά:** *net_amount, vat_amount, gross_amount, vat_line_amount, document_net_amount, document_vat_amount, document_allocable_vat_amount, document_gross_amount.*

4.6 Από το αποτέλεσμα του μηχανισμού καθαρισμού στην εγγραφή της βάσης δεδομένων

Το σημείο αυτό είναι κρίσιμο για την κατανόηση της υλοποίησης, επειδή η βάση δεν σχεδιάστηκε ως έναν μεγάλο ενιαίο επίπεδο πίνακα με ξεχωριστή PostgreSQL στήλη για κάθε πεδίο κρατήσεων, εσόδων και εξόδων. Μία τέτοια προσέγγιση θα οδηγούσε σε μεγάλο αριθμό στηλών οι οποίες θα ήταν *NULL* στις περισσότερες κατηγορίες. Για παράδειγμα, το *booking_code* δεν έχει νόημα σε μια εγγραφή εξόδου, ενώ το *expense_category* δεν έχει νόημα σε μία κράτηση. Αν όλες οι πιθανές έννοιες αποθηκεύονταν στον ίδιο πίνακα ως κανονικές στήλες, το σχήμα θα ήταν ιδιαίτερα αραιό και στενά συνδεδεμένο με την τρέχουσα μορφή των μηχανισμών καθαρισμού.

Στη συγκεκριμένη ροή οι βασικές εγγραφές της βάσης έχουν διακριτούς ρόλους, οι οποίοι συνοψίζονται στον Πίνακα 4.7.

Πίνακας 4.7: Κύριες δομές της βάσης που συμμετέχουν στην αποθήκευση των επεξεργασμένων δεδομένων.

Πίνακας	Ρόλος στη συγκεκριμένη εφαρμογή	Κύρια σύνδεση
<i>hotels</i>	Διατηρεί την ταυτότητα και τα σταθερά μεταδεδομένα της ξενοδοχειακής μονάδας.	Το <i>id</i> συνδέεται με το <i>hotel_id</i> της μεταφόρτωσης.
<i>uploaded_files</i>	Αποτελεί το μητρώο σε επίπεδο αρχείου κάθε μεταφόρτωσης και συνδέει το αρχικό και το καθαρισμένο αρχείο με τα μεταδεδομένα της επεξεργασίας.	Το <i>id</i> αποτελεί το κοινό <i>file_id</i> της ροής.
<i>file_rows</i>	Αποθηκεύει το αποτέλεσμα σε επίπεδο γραμμής, διατηρώντας παράλληλα αρχική και καθαρισμένη αναπαράσταση.	Το <i>source_file_id</i> αναφέρεται στο <i>uploaded_files.id</i> .
<i>processing_logs</i>	Διατηρεί τα γεγονότα που σχετίζονται με την επεξεργασία της ίδιας μεταφόρτωσης.	Το <i>source_file_id</i> αναφέρεται στο <i>uploaded_files.id</i> .

Για τον λόγο αυτό η αποθήκευση χωρίστηκε σε δύο διαφορετικά επίπεδα. Οι πληροφορίες που είναι κοινές για κάθε επεξεργασμένη εγγραφή αποθηκεύονται ως πραγματικές σχεσιακές στήλες του *file_rows*, ενώ το περιεχόμενο που διαφέρει ανά κατηγορία αποθηκεύεται στα *raw_data* και *cleaned_data* τύπου JSONB. Ο πίνακας *file_rows* περιλαμβάνει πραγματικές στήλες όπως *id*, *source_file_id*, *category*, *row_number*, *raw_data*, *cleaned_data* και *created_at*. Τα *raw_data* και *cleaned_data* είναι JSONB και μπορούν επομένως να περιέχουν διαφορετικό σύνολο κλειδιών ανά κατηγορία χωρίς να μεταβάλλεται το schema του πίνακα.

Πίνακας 4.8: Βασικές φυσικές στήλες του πίνακα *file_rows*.

Στήλη	Τύπος / μορφή	Ρόλος
<i>id</i>	UUID	Μοναδική ταυτότητα της εγγραφής σε επίπεδο γραμμής.
<i>source_file_id</i>	UUID	Σύνδεση της εγγραφής με το upload από το οποίο προέκυψε.
<i>category</i>	κατηγορία αρχείου	Διάκριση μεταξύ κρατήσεων, εσόδων και εξόδων.
<i>row_number</i>	integer	Σειρά της καθαρισμένης εγγραφής μέσα στο αποτέλεσμα του συγκεκριμένου αρχείου.
<i>raw_data</i>	JSONB	Αρχική πληροφορία ή οργανωμένη προέλευση της εγγραφής.
<i>cleaned_data</i>	JSONB	Η επιχειρησιακά καθαρισμένη και εμπλουτισμένη εγγραφή.
<i>validation_status</i>	text	Τεχνική κατάσταση της αποθήκευσης σε επίπεδο γραμμής.
<i>validation_errors</i>	JSONB	Τυχόν προειδοποιήσεις που συνοδεύουν την επεξεργασία· δεν αποτελεί ξεχωριστή αναφορά επικύρωσης.
<i>created_at</i>	timestamp	Χρόνος δημιουργίας της εγγραφής.

Αυτό σημαίνει ότι ένα πεδίο όπως το *expense_category* δεν δημιουργείται αρχικά ως στήλη PostgreSQL στον *file_rows*. Ενδεικτικά, στο *cleaned_data* μιας γραμμής εξόδου περιλαμβάνονται τα πεδία *account_code*, *expense_category*, *expense_subcategory*, *included_in_expenses*, *exclude_reason*, *rule_matched*, *net_amount*, *vat_amount* και *gross_amount*. Αντίστοιχα, μία εγγραφή εσόδου διαθέτει μέσα στο *cleaned_data* διαφορετικό σύνολο κλειδιών, όπως *transaction_date*, *document_no*, *customer_category*, *include_in_turnover*, *net_amount*, *vat_amount*, *total_turnover*, *account_codes*, *line_types* και *source_excel_rows*. Η δομή αυτή προκύπτει απευθείας από τη συγκεντρωτική εγγραφή που δημιουργεί ο μηχανισμός καθαρισμού εσόδων.

Μετά την ολοκλήρωση του μηχανισμού καθαρισμού, η κεντρική διαδικασία προσθέτει στις καθαρισμένες εγγραφές τα κοινά στοιχεία της ξενοδοχειακής μονάδας. Επομένως, τα στοιχεία του ξενοδοχείου δεν χρειάζεται να δημιουργούνται ξεχωριστά από κάθε μηχανισμό καθαρισμού. Η ίδια διαδικασία εφαρμόζεται κεντρικά στις τρεις κατηγορίες. Στη συνέχεια δημιουργείται το καθαρισμένο CSV, το οποίο αποθηκεύεται στο Storage, ενώ

το σχεσιακό αποτέλεσμα οδηγείται στον *file_rows*.

Πριν από την εισαγωγή των νέων εγγραφών διαγράφονται από τον *file_rows* τυχόν προηγούμενες εγγραφές που έχουν το ίδιο *source_file_id*. Η επιλογή αυτή είναι σημαντική σε περίπτωση επανεκτέλεσης της ίδιας μεταφόρτωσης: ο μηχανισμός καθαρισμού δεν προσθέτει δεύτερο αντίγραφο των ίδιων εγγραφών, αλλά αντικαθιστά το προηγούμενο αποτέλεσμα σε επίπεδο γραμμής του συγκεκριμένου αρχικού αρχείου.

Η εγγραφή στη βάση δημιουργείται μία προς μία σε λογικό επίπεδο. Για κάθε *cleaned_record* επιλέγεται το αντίστοιχο *raw_record* και κατασκευάζεται η εγγραφή που θα εισαχθεί στον *file_rows*. Τα βασικά στοιχεία της εγγραφής είναι το *source_file_id*, η κατηγορία, ο αύξων *row_number*, το *raw_data* και το *cleaned_data*. Ο κώδικας συγκεντρώνει τις εγγραφές σε *rows_to_insert* και τις αποστέλλει στη βάση σε ομάδες των 500 εγγραφών.

```
rows_to_insert.append({
    "source_file_id": file_id,
    "category": category,
    "row_number": index,
    "raw_data": raw_record,
    "cleaned_data": cleaned_record,
    "validation_status": "cleaned",
    "validation_errors": summary.get("warnings", []),
})
```

Η επιλογή της ομαδικής εισαγωγής είναι σημαντική σε αρχεία με πολλές χιλιάδες γραμμές. Αν γινόταν ξεχωριστή HTTP κλήση για κάθε εγγραφή, το συνολικό κόστος επικοινωνίας με το Supabase θα ήταν σημαντικά μεγαλύτερο. Με την ομαδοποίηση των εισαγωγών, η εφαρμογή διατηρεί αποθήκευση σε επίπεδο γραμμής χωρίς να απαιτεί αντίστοιχο αριθμό ανεξάρτητων αιτημάτων.

4.6.1 Διαφορά *raw_data* και *cleaned_data*

Η παράλληλη αποθήκευση των δύο αναπαραστάσεων είναι ιδιαίτερα σημαντική για τη συγκεκριμένη εφαρμογή. Το *raw_data* απαντά στο ερώτημα «από ποια πληροφορία προέκυψε αυτή η εγγραφή;», ενώ το *cleaned_data* απαντά στο «ποια είναι η τελική αναλυτική της μορφή;».

Στα έξοδα η σχέση είναι άμεση, καθώς μία καθαρισμένη γραμμή συνδέεται με την αντίστοιχη λογιστική γραμμή. Στα έσοδα όμως η σχέση είναι διαφορετική. Επειδή μία τελική εγγραφή παραστατικού μπορεί να προέρχεται από περισσότερες αρχικές γραμμές, η νεότερη υλοποίηση δημιουργεί ομαδοποιημένη αρχική εγγραφή. Για κάθε συνδυασμό *transaction_date* και *document_no* συγκεντρώνονται οι αρχικές λογιστικές γραμμές, διατηρείται η λίστα *source_excel_rows*, ο αριθμός *source_record_count* και τα ίδια τα *source_records*.

Επομένως, μία εγγραφή εσόδου στον *file_rows* διαθέτει, σε επίπεδο σταθερών στηλών, το *source_file_id* και το *row_number*. Παράλληλα, μέσα στο *raw_data* διατηρούνται στοιχεία όπως τα *transaction_date*, *document_no*, *source_excel_rows*, *source_record_count* και η λίστα των *source_records*, ενώ στο *cleaned_data* αποθηκεύονται η τελική μορφή της εγγραφής και βασικά πεδία όπως *customer_category*, *include_in_turnover*, *net_amount*, *vat_amount*, *total_turnover*, *account_codes* και *line_types*. Με τον τρόπο αυτό η βάση αποθηκεύει ταυτόχρονα την προέλευση και την ερμηνεία της οικονομικής εγγραφής. Η συγκεκριμένη επιλογή είναι ιδιαίτερα χρήσιμη για οικονομικά δεδομένα, επειδή μία λανθασμένη κατηγορία ή ένα μη αναμενόμενο ποσό μπορεί να ελεγχθεί μέχρι τις πραγματικές γραμμές της λογιστικής εξαγωγής.

4.6.2 Σύνδεση με το *uploaded_files*

Ο *file_rows* δεν αποτελεί ανεξάρτητο σύνολο δεδομένων. Το *source_file_id* είναι ξένο κλειδί προς το *uploaded_files.id*. Έτσι κάθε καθαρισμένη εγγραφή συνδέεται με την αρχική μεταφόρτωση. Από το *uploaded_files* είναι δυνατό να ανακτηθούν πληροφορίες όπως το αρχικό όνομα αρχείου, η κατηγορία, το *storage_path*, το *cleaned_storage_path*, το έτος, ο μήνας, η ημερομηνία επεξεργασίας και το πλήθος των γραμμών. Η εγγραφή του *uploaded_files* λειτουργεί επομένως ως εγγραφή μεταδεδομένων σε επίπεδο αρχείου, ενώ ο *file_rows* ως εγγραφή του περιεχομένου.

Συνοπτικά, τα βασικά μεταδεδομένα που χρησιμοποιούνται σε αυτό το επίπεδο είναι:

- το μοναδικό *id* της μεταφόρτωσης,
- το *hotel_id* και η κατηγορία του αρχείου,
- το *original_filename* και το *storage_path*,
- τα *file_month* και *file_year*,
- η κατάσταση επεξεργασίας και οι χρόνοι μεταφόρτωσης/ολοκλήρωσης,
- το *cleaned_storage_path* και το τελικό *rows_count*.

Η σχέση αυτή είναι σημαντική και για τα SQL views. Δεν είναι απαραίτητο να αντιγραφούν σε κάθε εγγραφή το αρχικό όνομα αρχείου και οι χρονικές στιγμές της μεταφόρτωσης ως μόνιμες φυσικές στήλες. Τα SQL views μπορούν να ενώσουν τον *file_rows* με το *uploaded_files* μέσω του *source_file_id* και να παρουσιάσουν ενιαία τόσο τις πληροφορίες της εγγραφής όσο και τις πληροφορίες του αρχείου.

Μετά την επιτυχή αποθήκευση των γραμμών, το *uploaded_files* ενημερώνεται με την ολοκλήρωση της επεξεργασίας, τη διαδρομή του καθαρισμένου αρχείου και τον αριθμό

των εγγραφών που δημιουργήθηκαν. Έτσι η εγγραφή σε επίπεδο αρχείου δεν δηλώνει απλώς ότι ένα αρχείο μεταφορτώθηκε, αλλά διατηρεί και τη σύνδεση με το αποτέλεσμα.

4.7 Δημιουργία των SQL views και του τελικού αναλυτικού σχήματος

Η χρήση JSONB στον *file_rows* παρέχει ευελιξία στην αποθήκευση, αλλά δεν είναι η καταλληλότερη μορφή για χρήση από εργαλεία επιχειρηματικής ευφυΐας. Για παράδειγμα, μία αριθμητική τιμή μέσα στο JSON πρέπει πρώτα να εξαχθεί και στη συνέχεια να μετατραπεί σε *numeric*. Αν αυτή η λογική επαναλαμβανόταν μέσα στο Power BI, η επιχειρησιακή σημασία των πεδίων θα μεταφερόταν από τη βάση στο επίπεδο της αναφοράς. Για τον λόγο αυτό δημιουργήθηκαν SQL views, δηλαδή προβολές, οι οποίες λειτουργούν ως ενδιάμεσο αναλυτικό στάδιο.

Η βασική αρχή μπορεί να συνοψιστεί ως εξής: *JSONB αποθήκευση* → *εξαγωγή πεδίων* → *μετατροπή τύπων* → *αντιστοιχίσεις* → *αναλυτική προβολή*.

Τα SQL views δεν δημιουργούν δεύτερο αντίγραφο των εγγραφών. Παρουσιάζουν τα δεδομένα του *file_rows* σε σταθερό σχεσιακό σχήμα. Αυτό επιτρέπει στο επίπεδο αποθήκευσης να παραμένει ευέλικτο και στο επίπεδο ανάλυσης να υπάρχει μία αυστηρά ορισμένη δομή.

Πίνακας 4.9: Κύρια SQL views και ρόλος τους στο αναλυτικό επίπεδο.

SQL view	Βασική πηγή	Κύριος ρόλος
<i>v_bookings_clean</i>	<i>file_rows</i> κρατήσεων και σχετικές αντιστοιχίσεις	Τυποποιημένες κρατήσεις, ημερομηνίες, δωμάτια, οικονομικά πεδία και μεταδεδομένα ξενοδοχείου.
<i>v_income_clean</i>	<i>file_rows</i> εσόδων και μεταδεδομένα της μεταφόρτωσης	Μετατροπή του καθαρισμένου JSON σε παραστατικά με αριθμητικά ποσά, ένδειξη συμμετοχής και τυποποιημένες κατηγορίες πελατών.
<i>v_expenses_clean</i>	<i>file_rows</i> εξόδων και μεταδεδομένα της μεταφόρτωσης	Παρουσίαση κατηγοριών, κανόνων και οικονομικών ποσών ως κανονικών σχεσιακών στηλών.
<i>v_final_bookings_monthly</i>	Καθαρισμένο επίπεδο κρατήσεων	Μηνιαία συγκέντρωση ενεργών κρατήσεων για αναλυτική και προβλεπτική χρήση.
<i>v_forecast_monthly_series</i>	Μηνιαίες συγκεντρώσεις κρατήσεων, εσόδων και εξόδων	Κοινό σχήμα <i>hotel_id, period_month, target_name, actual_value</i> για την πρόβλεψη.
<i>v_hotel_operating_calendar</i>	Ημερολόγιο λειτουργίας και δεδομένα διαμονής	Χρονικό πλαίσιο λειτουργίας και βάση για υπολογισμούς δυναμικότητας/πληρότητας.

4.7.1 Δημιουργία της προβολής εσόδων (*v_income_clean*)

Η προβολή *v_income_clean* αρχικά εντοπίζει τις κατάλληλες ολοκληρωμένες μεταφορτώσεις της κατηγορίας *income*. Στη συγκεκριμένη υλοποίηση χρησιμοποιείται *DISTINCTON(original_filename)* και ταξινόμηση κατά *uploaded_atDESC*, ώστε, όταν το ίδιο λογικό αρχείο έχει μεταφορτωθεί ξανά, να επιλέγεται η νεότερη καθαρισμένη έκδοσή του.

Κατόπιν πραγματοποιείται σύνδεση του *uploaded_files* με τον *file_rows* μέσω της συνθήκης *uf.id=fr.source_file_id*. Στη συνέχεια τα κλειδιά του *cleaned_data* μετατρέπονται σε κανονικές στήλες της προβολής. Για παράδειγμα, η έκφραση *fr.cleaned_data->'document_no'* δημιουργεί το πεδίο *document_no*, ενώ η έκφραση *nullif(fr.cleaned_data->'net_amount', '')::numeric* μετατρέπει την αποθηκευμένη JSON τιμή σε πραγμα-

τικό PostgreSQL *numeric*. Η ίδια λογική εφαρμόζεται στα *vat_amount*, *total_turnover*, *tips_amount* και στα λογιστικά αθροίσματα.

Ιδιαίτερη μετατροπή εφαρμόζεται στο *include_in_turnover*. Η προβολή αναγνωρίζει τιμές όπως *true*, *t*, *1* και *yes* ως *TRUE* και *false*, *f*, *0* και *no* ως *FALSE*. Έτσι το Power BI λαμβάνει πραγματικό λογικό πεδίο Boolean και δεν χρειάζεται να ερμηνεύσει συμβολοσειρές.

Στο επόμενο επίπεδο η προβολή εσόδων συνδέεται και με την αντιστοίχιση των εμφανιζόμενων ονομάτων πελατών. Το *display_customer_name* επιλέγεται από την αντιστοίχιση όταν υπάρχει ενεργή αντιστοίχιση και, σε διαφορετική περίπτωση, χρησιμοποιείται η αρχική αιτιολογία. Παράλληλα το *customer_category* μετατρέπεται σε αναλυτικό *revenue_channel*: *individual_customer* σε *Direct*, *ota* σε *OTA*, *ta* σε *TA*, *business* σε *Business* και *other* σε *Other*.

Αυτό σημαίνει ότι στην περίπτωση των εσόδων υπάρχουν δύο διακριτές μορφές αντιστοίχισης. Ο μηχανισμός καθαρισμού σε Python αποφασίζει την **κατηγορία** του πελάτη, ενώ το αναλυτικό επίπεδο SQL μπορεί να τυποποιήσει το **εμφανιζόμενο όνομα**. Η διάκριση επιτρέπει, για παράδειγμα, διαφορετικές αρχικές γραφές της ίδιας οντότητας να εμφανίζονται με κοινό όνομα χωρίς να απαιτείται αλλαγή του αρχικού αρχείου.

4.7.2 Δημιουργία της προβολής εξόδων (*v_expenses_clean*)

Αντίστοιχη διαδικασία χρησιμοποιείται στα έξοδα. Από το *cleaned_data* εξάγονται *document_no*, *account_code*, *account_description*, *supplier_name*, *vat_number*, *reason*, *expense_category*, *expense_subcategory*, *line_type*, *included_in_expenses*, *exclude_reason*, *rule_matched* και *confidence*. Οι οικονομικές τιμές μετατρέπονται σε αριθμητικό τύπο και η τελική προβολή περιλαμβάνει *signed_amount*, *debit_amount*, *credit_amount*, *net_amount*, *vat_amount* και *gross_amount*.

Η μετατροπή των αριθμών στην προβολή υποστηρίζει περισσότερες από μία λεκτικές μορφές. Υποστηρίζονται απλές δεκαδικές τιμές, ευρωπαϊκές τιμές με τελεία χιλιάδων και κόμμα δεκαδικών, καθώς και τιμές με μόνο δεκαδικό κόμμα. Στο τέλος όλες οι περιπτώσεις παρουσιάζονται ως *numeric*.

Αυτό είναι ιδιαίτερα σημαντικό επειδή η τελική αριθμητική στήλη της προβολής έχει σταθερό τύπο δεδομένων ανεξάρτητα από το πώς είχε καταγραφεί η τιμή στην αρχική πηγή. Το *v_expenses_clean*, για παράδειγμα, διαθέτει *signed_amount*, *debit_amount*, *credit_amount*, *net_amount*, *vat_amount* και *gross_amount* ως *numeric*, ενώ εξακολουθεί να διατηρεί *full_cleaned_data* και *full_raw_data* ως JSONB για τον έλεγχο και την ιχνηλασιμότητα.

4.7.3 Δημιουργία της προβολής κρατήσεων (*v_bookings_clean*)

Στις κρατήσεις η ίδια λογική χρησιμοποιείται για μεγαλύτερο αριθμό πεδίων. Η προβολή μετατρέπει τα κλειδιά JSON σε σχεσιακές στήλες για ημερομηνίες κράτησης και διαμονής, κατάσταση, αριθμό επισκεπτών, δωμάτια και οικονομικές αξίες. Παράλληλα συνδέει τις εγγραφές με την αντιστοίχιση των τύπων δωματίων και προσθέτει την τυποποιημένη ονομασία, τη σύντομη ονομασία και τη σειρά εμφάνισης.

Στο τελικό αποτέλεσμα υπάρχουν επίσης *hotel_id*, *hotel_name* και *hotel_code*, ώστε οι ίδιες αναλυτικές δομές να μπορούν να χρησιμοποιηθούν όταν στην εφαρμογή προστεθούν περισσότερες ξενοδοχειακές μονάδες.

Πάνω από το καθαρισμένο επίπεδο δημιουργούνται στη συνέχεια πιο εξειδικευμένες SQL views. Το *v_final_bookings_monthly*, για παράδειγμα, μετατρέπει τα δεδομένα σε επίπεδο γραμμής κρατήσεων σε μηνιαία παρατήρηση και διαθέτει *period_month* και *actual_bookings*. Τα πεδία του έχουν ήδη καθορισμένους SQL τύπους, με το *period_month* ως *date* και το *actual_bookings* ως *integer*.

4.7.4 Από τις SQL views στους τελικούς δείκτες

Μετά τη δημιουργία των καθαρισμένων προβολών, η εφαρμογή μπορεί να διαχωρίσει καθαρά το «τι υπάρχει στη λογιστική πηγή» από το «τι χρησιμοποιείται στον αναλυτικό δείκτη». Για τα έσοδα το βασικό φίλτρο είναι το *include_in_turnover*, ενώ για τα έξοδα το *included_in_expenses*. Έτσι είναι δυνατό να υπάρχουν ταυτόχρονα στη βάση τα συνολικά λογιστικά μεγέθη *Revenue_{all}* και *Expenses_{all}*, αλλά και τα αναλυτικά *included* μεγέθη:

$$\begin{aligned} Revenue_{included} &= \sum net_amount_i, & \text{για } include_in_turnover_i &= TRUE, \\ Expenses_{included} &= \sum net_amount_i, & \text{για } included_in_expenses_i &= TRUE. \end{aligned}$$

Η διατήρηση και των δύο επιπέδων είναι ουσιαστική για τον έλεγχο των αποτελεσμάτων. Η διαφορά ανάμεσα στο συνολικό λογιστικό ποσό και στο *included* ποσό δεν αντιμετωπίζεται ως απόκλιση και μπορεί να εξηγηθεί μέσω των εξαιρούμενων εγγραφών και των αντίστοιχων *exclude_reason*.

Τα βασικά μεγέθη που προκύπτουν από το τελικό αναλυτικό επίπεδο και μπορούν να μετατραπούν σε μηνιαίες χρονοσειρές είναι:

- *active_bookings*, δηλαδή το πλήθος των ενεργών κρατήσεων στην αντίστοιχη χρονική περίοδο,
- *included_net_revenue*, δηλαδή το άθροισμα της καθαρής αξίας των εγγραφών εσόδων που συμμετέχουν στον τζίρο,

- *included_operating_expenses*, δηλαδή το άθροισμα της καθαρής αξίας των εγγραφών εξόδων που έχουν χαρακτηριστεί ως *included* λειτουργικές δαπάνες.

Η συνολική δομή των δεδομένων που προέκυψε στην εφαρμογή μπορεί επομένως να θεωρηθεί ως ακολουθία τεσσάρων επιπέδων. Το πρώτο επίπεδο είναι το πραγματικό αρχείο στο Storage. Το δεύτερο είναι το ιστορικό σε επίπεδο γραμμής στον *file_rows*, όπου συνυπάρχουν αρχική και καθαρισμένη αναπαράσταση. Το τρίτο είναι το σχεσιακό αναλυτικό επίπεδο των SQL views, στο οποίο τα JSON μετατρέπονται σε σταθερές και σωστά τυποποιημένες στήλες. Το τέταρτο είναι το επίπεδο των συγκεντρώσεων και των χρονοσειρών, στο οποίο δημιουργούνται οι μεταβλητές που χρησιμοποιούνται στην επιχειρηματική ανάλυση και στην πρόβλεψη.

Η προσέγγιση αυτή επέτρεψε να διατηρηθούν ταυτόχρονα δύο ιδιότητες που συχνά βρίσκονται σε σύγκρουση: η ευελιξία απέναντι σε διαφορετικές πηγές και η σταθερότητα του τελικού αναλυτικού σχήματος. Οι μηχανισμοί καθαρισμού μπορούν να εξελίσσονται και να δημιουργούν νέα κλειδιά χωρίς να απαιτείται άμεσος ανασχεδιασμός ενός μεγάλου ενιαίου πίνακα, ενώ οι προβολές παρέχουν προς το Power BI και το υποσύστημα πρόβλεψης μια σταθερή διεπαφή δεδομένων με σαφείς ονομασίες και τύπους δεδομένων. Επιπλέον, επειδή διατηρούνται το *source_file_id*, οι αρχικές εγγραφές, οι καθαρισμένες εγγραφές, οι αρχικές γραμμές του Excel και τα πεδία που περιγράφουν τους κανόνες ταξινόμησης, τα τελικά ποσά μπορούν να αναλυθούν μέχρι το επίπεδο της εγγραφής από την οποία προέκυψαν.

Με αυτή τη δομή, ο καθαρισμός και η βάση δεδομένων δεν λειτουργούν ως δύο ανεξάρτητα στάδια. Ο μηχανισμός καθαρισμού καθορίζει τη σημασιολογία της εγγραφής, ο *file_rows* τη διατηρεί με την προέλευσή της και τα SQL views μετατρέπουν τη σημασιολογία αυτή σε σταθερό αναλυτικό μοντέλο. Αυτό είναι το βασικό σημείο στο οποίο τα αρχικά δεδομένα μετατρέπονται σε οργανωμένη πληροφορία κατάλληλη για τις αναλύσεις που παρουσιάζονται στα επόμενα κεφάλαια.

4.8 Δημιουργία των μηνιαίων χρονοσειρών και υπολογισμός της πληρότητας

Μετά τον καθαρισμό και τη δημιουργία των τυποποιημένων προβολών, το επόμενο βήμα στη συγκεκριμένη περίπτωση ήταν η μετατροπή των εγγραφών σε επίπεδο γραμμής σε μηνιαία μεγέθη. Το στάδιο αυτό είναι διαφορετικό από τον καθαρισμό: ο μηχανισμός καθαρισμού αποφασίζει τι σημαίνει κάθε εγγραφή, ενώ η μηνιαία συγκέντρωση αποφασίζει πώς οι έγκυρες εγγραφές συνδυάζονται σε μία παρατήρηση χρονοσειράς. Για τον λόγο αυτό οι χρονοσειρές δημιουργούνται πάνω από τις καθαρισμένες προβολές SQL και όχι απευθείας από τα αρχικά Excel.

Για την πρόβλεψη χρησιμοποιείται κοινό αναλυτικό σχήμα με βασικά πεδία *hotel_id*, *hotel_name*, *hotel_code*, *period_month*, *target_name* και *actual_value*. Η κοινή αυτή μορφή επιτρέπει στον ίδιο Python κώδικα να επεξεργάζεται διαφορετικές μεταβλητές-στόχους χωρίς να γνωρίζει αν η αρχική πληροφορία προήλθε από κράτηση, παραστατικό εσόδου ή λογιστική γραμμή εξόδου. Η διαφοροποίηση έχει ήδη ολοκληρωθεί στο επίπεδο των SQL views.

Στη συγκεκριμένη εφαρμογή οι τρεις κύριες μηνιαίες σειρές σχηματίζονται ως εξής:

Πίνακας 4.10: Δημιουργία των μηνιαίων μεταβλητών που χρησιμοποιούνται στην πρόβλεψη.

Μεταβλητή-στόχος	Χρονική βάση	Υπολογισμός
<i>active_bookings</i>	Μήνας άφιξης	Πλήθος διακριτών <i>booking_code</i> για κρατήσεις με <i>booking_status=Ενεργή</i> .
<i>included_net_revenue</i>	Μήνας <i>transaction_date</i>	Άθροισμα <i>net_amount</i> μόνο για εγγραφές με <i>include_in_turnover=TRUE</i> .
<i>included_operating_expenses</i>	Μήνας <i>transaction_date</i>	Άθροισμα <i>net_amount</i> μόνο για εγγραφές με <i>included_in_expenses=TRUE</i> .

Για τις κρατήσεις χρησιμοποιείται το πλήθος κωδικών και όχι απλό πλήθος γραμμών. Η επιλογή αυτή αποτρέπει μία επαναλαμβανόμενη εγγραφή να αυξήσει το πλήθος των κρατήσεων. Το *v_final_bookings_monthly* αποτυπώνει αυτή τη λογική σε επίπεδο μήνα μέσω του *period_month* και του *actual_bookings*. Για τα οικονομικά μεγέθη, αντίστοιχα, δεν αθροίζονται όλες οι λογιστικές εγγραφές, αλλά μόνο εκείνες που έχουν ήδη περάσει από την απόφαση συμμετοχής/εξαίρεσης του μηχανισμού καθαρισμού. Επομένως, το σύστημα πρόβλεψης εκπαιδεύεται πάνω στα ίδια μεγέθη που χρησιμοποιούνται και στην αναλυτική παρουσίαση.

Η πληρότητα αντιμετωπίστηκε ξεχωριστά από τις τρεις μεταβλητές-στόχους πρόβλεψης. Δεν χρησιμοποιείται ως κύρια μεταβλητή-στόχος της πρόβλεψης, αλλά αποτελεί βασικό λειτουργικό KPI και παράγεται από τις καθαρισμένες κρατήσεις σε συνδυασμό με το ημερολόγιο λειτουργίας. Η λογική βασίζεται στα πραγματικά πεδία *arrival_date* και *departure_date* και στις κρατήσεις που έχουν χαρακτηριστεί ως *Ενεργή*. Για κάθε ημερομηνία λειτουργίας προσδιορίζεται πόσα δωμάτια καταλαμβάνονται από ενεργές διαμονές και οι ημερήσιες τιμές συγκεντρώνονται στη συνέχεια σε μήνα.

Στη συγκεκριμένη μονάδα η χρησιμοποιούμενη δυναμικότητα είναι 16 δωμάτια. Επομένως, όταν ένας μήνας περιλαμβάνει D_m ημέρες λειτουργίας, οι διαθέσιμες διανυκτερεύσεις δωματίων (available room-nights) είναι:

$$AvailableRoomNights_m = 16 \times D_m.$$

Για πλήρη μήνα 30 ημερών το μέγεθος είναι $16 \times 30 = 480$, ενώ για πλήρη μήνα 31 ημερών είναι $16 \times 31 = 496$. Η χρήση των *ημερών λειτουργίας* και όχι μηχανικά όλων των ημερών του ημερολογιακού μήνα επιτρέπει η ίδια λογική να επεκταθεί και σε περιόδους κατά τις οποίες η μονάδα λειτουργεί μόνο για μέρος του μήνα.

Οι κατειλημμένες διανυκτερεύσεις δωματίων προκύπτουν από την ημερήσια ανάπτυξη των ενεργών κρατήσεων στις πραγματικές ημερομηνίες διαμονής:

$$OccupiedRoomNights_m = \sum_{d \in m} OccupiedRooms_d.$$

Από τα δύο προηγούμενα μεγέθη υπολογίζονται οι διαθέσιμες διανυκτερεύσεις που δεν χρησιμοποιήθηκαν και το ποσοστό πληρότητας:

$$RemainingRoomNights_m = AvailableRoomNights_m - OccupiedRoomNights_m,$$

$$OccupancyRate_m = 100 \times \frac{OccupiedRoomNights_m}{AvailableRoomNights_m}.$$

Το αποτέλεσμα της συγκεκριμένης υλοποίησης περιλαμβάνει επιπλέον πεδία ελέγχου. Το *zero_occupancy_days* μετρά τις ημέρες λειτουργίας χωρίς κατειλημμένο δωμάτιο, το *full_days* τις ημέρες στις οποίες η ημερήσια κάλυψη φθάνει τη διαθέσιμη δυναμικότητα και το *over_capacity_days* τις ημέρες στις οποίες τα δεδομένα θα οδηγούσαν σε αριθμό κατειλημμένων δωματίων μεγαλύτερο από τη δυναμικότητα. Το τελευταίο λειτουργεί ουσιαστικά ως έλεγχος συνέπειας των δεδομένων.

Ο Πίνακας 4.11 παρουσιάζει ενδεικτικές μηνιαίες τιμές που παράγονται από τον υπολογισμό. Οι σχέσεις μεταξύ των πεδίων φαίνονται άμεσα. Τον Μάιο του 2024, για παράδειγμα, οι 31 ημέρες αντιστοιχούν σε 496 διαθέσιμες διανυκτερεύσεις δωματίων, από τις οποίες 441 ήταν κατειλημμένες. Απομένουν 55 και το ποσοστό πληρότητας είναι $441/496 \times 100 = 88,91\%$.

Πίνακας 4.11: Ενδεικτικός μηνιαίος υπολογισμός πληρότητας.

Μήνας	Ημέρες λειτουργ- γίας	Διαθέσιμες διανυκτερεύ- σεις	Κατειλημμένες διανυκτερεύσεις	Υπόλοιπο	Πληρότητα
04/2024	30	480	204	276	42,50%
05/2024	31	496	441	55	88,91%
06/2024	30	480	421	59	87,71%
08/2024	31	496	452	44	91,13%

Στο σύνολο υπάρχουν επίσης τα *first_operating_date*, *last_operating_date*, *expected_available_room_nights*, *recalculated_remaining_room_nights* και *recalculated_occupancy_rate_pct*. Τα επαναυπολογιζόμενα πεδία επιτρέπουν να ελέγχεται ότι η δηλωμένη δυναμικότητα, οι κατειλημμένες διανυκτερεύσεις δωματίων και το τελικό ποσοστό συμφωνούν αριθμητικά. Για παράδειγμα, όταν οι *available_room_nights* και *expected_available_room_nights* είναι ίσες και το *recalculated_occupancy_rate_pct* ταυτίζεται με το *occupancy_rate_pct*, επιβεβαιώνεται ότι ο παρονομαστής της πληρότητας έχει εφαρμοστεί όπως αναμένεται.

Η διάκριση μεταξύ της χρονοσειράς κρατήσεων και της πληρότητας είναι σημαντική. Το *active_bookings* μετρά κρατήσεις, ενώ το *occupied_room_nights* μετρά την δυναμικότητα μέσα στις ημέρες διαμονής. Μία κράτηση επτά διανυκτερεύσεων εξακολουθεί να είναι μία κράτηση στη μηνιαία χρονοσειρά, αλλά συνεισφέρει επτά διανυκτερεύσεις δωματίων στον υπολογισμό της πληρότητας. Για τον λόγο αυτό οι δύο δείκτες δεν είναι εναλλάξιμοι και χρησιμοποιούνται για διαφορετικό σκοπό: οι κρατήσεις ως μεταβλητή-στόχος πρόβλεψης και η πληρότητα ως λειτουργικός δείκτης ελέγχου δυναμικότητας.

Τέλος, το *v_hotel_operating_calendar* δηλώνεται στη ρύθμιση της πρόβλεψης ως η πηγή του ημερολογίου λειτουργίας. Αυτό επιτρέπει η περίοδος λειτουργίας να αποτελεί δεδομένο του συστήματος και όχι υπόθεση μέσα στα μοντέλα. Η μηνιαία χρονοσειρά και το ημερολόγιο λειτουργίας συνδέονται επομένως στο αναλυτικό επίπεδο, αλλά διατηρούν διαφορετικό ρόλο: το πρώτο παρέχει τις πραγματικές τιμές των μεταβλητών-στόχων και το δεύτερο το χρονικό πλαίσιο μέσα στο οποίο ερμηνεύονται οι κρατήσεις, η εποχικότητα και η πληρότητα.

4.8.1 Ρύθμιση υπερπαραμέτρων των SARIMA και LSTM και διαχωρισμός από τη ροή παραγωγής

Η δημιουργία της μηνιαίας χρονοσειράς καθορίζει ποια δεδομένα θα χρησιμοποιηθούν από την πρόβλεψη, δεν καθορίζει όμως από μόνη της τις παραμέτρους των πιο σύνθετων μοντέλων. Στη συγκεκριμένη εφαρμογή ιδιαίτερη προσοχή δόθηκε στα SARIMA και LSTM, επειδή η συμπεριφορά τους εξαρτάται ουσιαστικά από την επιλογή των παραμέτρων και υπερπαραμέτρων τους. Για τον λόγο αυτό δεν χρησιμοποιήθηκε μία ενιαία ρύθμιση για όλες τις μεταβλητές πρόβλεψης. Πριν από την ένταξη των μοντέλων στην διαδικασία πρόβλεψης πραγματοποιήθηκε ξεχωριστή διαδικασία ρύθμισης, κατά την οποία συγκρίθηκαν εναλλακτικές ρυθμίσεις πάνω στις ίδιες αρχές ιστορικής αξιολόγησης (backtesting) που χρησιμοποιούνται και στην τελική αξιολόγηση.

Η απόφαση να πραγματοποιείται η ρύθμιση υπερπαραμέτρων εκτός της κανονικής ροής της εφαρμογής ήταν σκόπιμη. Η λειτουργία που εκτελεί ο τελικός χρήστης πρέπει να παραμένει προβλέψιμη ως προς τον χρόνο εκτέλεσης και να έχει συγκεκριμένο ρόλο: να ανακτήσει την κατάλληλη χρονοσειρά, να εκπαιδεύσει τα διαθέσιμα μοντέλα με τις ήδη εγκεκριμένες παραμέτρους τους, να τα αξιολογήσει στα καθορισμένα folds και να παράγει την πρόβλεψη. Αν σε κάθε κλήση του `/forecast/monthly/run` επαναλαμβανόταν παράλληλα και αναζήτηση υπερπαραμέτρων, μία μόνο πρόβλεψη θα απαιτούσε την εκπαίδευση πολλών παραλλαγών του ίδιου μοντέλου. Αυτό θα αύξανε σημαντικά τον χρόνο και το υπολογιστικό κόστος της εκτέλεσης.

Η ρύθμιση υπερπαραμέτρων αντιμετωπίστηκε επομένως ως ξεχωριστή διαδικασία. Εκτελείται όταν δημιουργείται αρχικά ένα μοντέλο, όταν αλλάζει ουσιαστικά η διαθέσιμη ιστορία ή όταν υπάρχει λόγος να επανεξεταστεί η προηγούμενη ρύθμιση. Ο PYTHON κώδικας ρύθμισης υπερπαραμέτρων χρησιμοποιεί την ίδια μηνιαία πηγή δεδομένων και την ίδια λογική rolling-origin, αλλά παράγει αναλυτικά αποτελέσματα ανά υποψήφια ρύθμιση: κατάσταση εκτέλεσης, μετρικές ανά fold, συνολική κατάταξη και τελική επιλογή. Οι δοκιμαστικές εκτελέσεις μπορούν να πραγματοποιούνται χωρίς άμεση εγγραφή στη βάση, σε λειτουργία δοκιμαστικής εκτέλεσης ώστε να ελεγχθούν πρώτα τα αποτελέσματα ελέγχου και η σταθερότητα της επιλογής. Μόνο η ρύθμιση που έχει ελεγχθεί χρειάζεται να αποτελέσει παράμετρο της διαδικασίας.

Στο SARIMA η ρύθμιση υπερπαραμέτρων αφορά πρωτίστως τις δύο ομάδες παραμέτρων

$$(p, d, q) \quad \text{και} \quad (P, D, Q, 12),$$

όπου το 12 παραμένει σταθερό λόγω της μηνιαίας εποχικότητας. Αντί να θεωρηθεί ότι ο ίδιος συνδυασμός είναι κατάλληλος για κρατήσεις, έσοδα και έξοδα, δημιουργείται σύνολο υποψήφιων ρυθμίσεων με διαφορετικούς συνδυασμούς όρων, διαφοροποίησης και

όρων κινητού μέσου, τόσο στο μη εποχικό όσο και στο εποχικό μέρος. Κάθε υποψήφια ρύθμιση εφαρμόζεται στα ίδια διαδοχικά folds χρονικής αξιολόγησης και αξιολογείται με MAE, RMSE, sMAPE και Bias. Παράλληλα πραγματοποιούνται έλεγχοι εγκυρότητας και σταθερότητας, όπως η συμφωνία της εποχικής περιόδου με τους 12 μήνες και ο έλεγχος προσαρμογής του μοντέλου.

Η ανάγκη αυτής της διαδικασίας γίνεται σαφέστερη αν εξεταστούν οι τρεις χρονοσειρές. Το *active_bookings* είναι σειρά διακριτών μετρήσεων και παρουσιάζει έντονη εποχική συγκέντρωση στους μήνες λειτουργίας. Το *included_net_revenue* είναι χρηματική σειρά της οποίας το ύψος επηρεάζεται όχι μόνο από το πλήθος των κρατήσεων αλλά και από την αξία των συναλλαγών. Τα *included_operating_expenses* έχουν διαφορετική χρονική συμπεριφορά, καθώς περιλαμβάνουν κατηγορίες κόστους που δεν ακολουθούν υποχρεωτικά την ίδια εποχικότητα με τη ζήτηση. Επομένως, η επιβολή του ίδιου *order* και *seasonal_order* και στις τρεις σειρές θα ήταν τεχνικά απλούστερη, αλλά θα υπέθετε χωρίς αξιολόγηση ότι οι τρεις κατηγορίες λειτουργούν παρόμοια.

Στο LSTM η ανάγκη ρύθμισης υπερπαραμέτρων είναι ακόμη μεγαλύτερη, επειδή η ρύθμιση καθορίζει τόσο το μέγεθος της πληροφορίας που εισέρχεται στο δίκτυο όσο και την πολυπλοκότητα της εκπαίδευσης. Στην επίσημη διαδικασία *All History* εξετάστηκε πλέγμα 18 διαφορετικών υποψήφιων ρυθμίσεων. Μεταβλήθηκε το *lookback*, ο αριθμός των *units*, ο *learning_rate*, το *batch_size*, το *dropout_rate* και ο μέγιστος αριθμός *epochs*. Η *validation_fraction* διατηρήθηκε στο 20% και η *patience* στην τιμή 8, ώστε οι συγκρίσεις να επικεντρώνονται στις βασικές εκπαιδευτικές επιλογές και να χρησιμοποιείται κοινός μηχανισμός πρόωρης διακοπής.

Επιπλέον, η αξιολόγηση του LSTM δεν βασίστηκε σε μία μόνο τυχαία αρχικοποίηση. Για κάθε υποψήφια ρύθμιση χρησιμοποιήθηκαν οι τιμές αρχικοποίησης (*seeds*) 42, 123 και 2026. Η επιλογή αυτή είναι σημαντική επειδή η εκπαίδευση νευρωνικού δικτύου περιλαμβάνει στοχαστικά στοιχεία και δύο εκτελέσεις της ίδιας αρχιτεκτονικής μπορούν να καταλήξουν σε ελαφρώς διαφορετικό αποτέλεσμα. Η χρήση περισσότερων τιμών αρχικοποίησης επιτρέπει η κατάταξη να βασίζεται σε ρύθμιση που εμφανίζει επαναλαμβανόμενα καλή συμπεριφορά και όχι σε μία τυχαία καλή εκτέλεση.

Πίνακας 4.12: Κύριες παράμετροι που εξετάζονται κατά τη ρύθμιση υπερπαραμέτρων των SARIMA και LSTM.

Μοντέλο	Παράμετροι που μεταβάλλονται	Σκοπός
SARIMA	$order=(p,d,q)$, $seasonal_order=(P,D,Q,12)$	Επιλογή της κατάλληλης μη εποχικής και εποχικής δυναμικής για κάθε μεταβλητή-στόχο.
LSTM	$lookback$, $units$, $learning_rate$, $batch_size$, $dropout_rate$, $epochs$	Έλεγχος του ιστορικού βάθους, της χωρητικότητας του δικτύου, της ταχύτητας μάθησης και της κανονικοποίησης.
LSTM -- κοι- νές ρυθμίσεις	$validation_fraction=0.20$, $patience=8$, τιμές αρχικοποίησης 42/123/2026	Συγκρίσιμη αξιολόγηση των υποψήφιων ρυθμίσεων, πρόωρη διακοπή και περιορισμός της επίδρασης μιας μόνο τυχαίας αρχικοποίησης.

Το βασικό όφελος της ρύθμισης υπερπαραμέτρων είναι ότι επιτρέπει να εξετάζεται ποιο μοντέλο είναι καταλληλότερο για κάθε χρονοσειρά αλλά και ποια παραμετροποίηση αποδίδει καλύτερα. Η επιλογή του SARIMA ως οικογένειας μοντέλου δεν συνεπάγεται ότι υπάρχει ένας μοναδικός συνδυασμός παραμέτρων που είναι βέλτιστος για όλες τις μεταβλητές-στόχους. Αντίστοιχα, η επιλογή του LSTM δεν συνεπάγεται ότι το ίδιο $lookback$ και ο ίδιος αριθμός $units$ είναι κατάλληλα για κρατήσεις, έσοδα και έξοδα.

Η διαφοροποίηση αυτή έχει ενσωματωθεί και στο σχεδιασμό της βάσης. Η στρατηγική εκπαίδευσης αποτελεί ξεχωριστό κομμάτι της διαδικασίας και στην τρέχουσα υλοποίηση οι προβλέψεις μπορούν να αξιολογηθούν με τις στρατηγικές *All History* και *Last 30 Months*. Οι επιλεγμένες παράμετροι αναζητούνται με συνδυασμό $hotel_id$, $target_name$ και $training_strategy$. Για το SARIMA ανακτώνται το $order$ και το $seasonal_order$, ενώ για το LSTM ανακτώνται το $lookback_months$, τα $units$, το $learning_rate$, το $batch_size$, το $dropout_rate$, τα $epochs$, η $patience$, η $validation_fraction$ και οι τιμές αρχικοποίησης που χρησιμοποιήθηκαν στη ρύθμιση υπερπαραμέτρων. Η λογική αυτή σημαίνει ότι η υπηρεσία δεν χρειάζεται να γνωρίζει μία σταθερή ρύθμιση για όλες τις περιπτώσεις. Φορτώνει τις παραμέτρους που αντιστοιχούν στη συγκεκριμένη μονάδα, στη συγκεκριμένη μεταβλητή-στόχο και στην ενεργή στρατηγική εκπαίδευσης.

Ενδεικτικά, στο επίπεδο υπηρεσίας η λογική ανάκτησης έχει τη μορφή:

```
row = repository.fetch_sarima_parameters(
    hotel_id,
    target_name,
    training_strategy,
)
```

```
row = repository.fetch_lstm_parameters(  
    hotel_id,  
    target_name,  
    training_strategy,  
)
```

Η επιλογή αυτή προσφέρει και καλύτερη ιχνηλασιμότητα. Μαζί με τις ρυθμισμένες παραμέτρους μπορούν να διατηρούνται η *tuning_version*, το χρονικό εύρος των δεδομένων, ο αριθμός των υποψήφιων ρυθμίσεων και των επιτυχημένων folds και οι συγκεντρωτικές μετρικές της τελικής επιλογής. Επομένως, αν στο μέλλον επαναληφθεί η ρύθμιση υπερπαραμέτρων μετά την προσθήκη νέου ιστορικού, μπορεί να διακριθεί η νέα ρύθμιση από εκείνη που χρησιμοποιήθηκε σε παλαιότερο στάδιο. Παράλληλα, η ίδια αρχιτεκτονική επιτρέπει, όταν προστεθεί νέο ξενοδοχείο, να ελεγχθεί αν οι ήδη χρησιμοποιούμενες παράμετροι παραμένουν κατάλληλες ή αν πρέπει να δημιουργηθούν διαφορετικές τελικές επιλογές για τη νέα μονάδα.

Συνολικά, η ρύθμιση υπερπαραμέτρων δεν προστέθηκε για να αυξηθεί απλώς ο αριθμός των δοκιμών. Χρησιμοποιήθηκε για να αποφευχθεί η αυθαίρετη υπόθεση ότι ένα ενιαίο σύνολο παραμέτρων μπορεί να περιγράψει εξίσου καλά διαφορετικές οικονομικές και λειτουργικές χρονοσειρές. Η εκτέλεσή της εκτός της παραγωγικής ροής επιτρέπει να γίνει αναλυτική και υπολογιστικά βαρύτερη αναζήτηση όταν πραγματικά χρειάζεται, ενώ η καθημερινή λειτουργία της εφαρμογής χρησιμοποιεί μόνο το ελεγμένο αποτέλεσμα αυτής της διαδικασίας. Με αυτόν τον τρόπο διατηρούνται ταυτόχρονα η προσαρμοστικότητα των μοντέλων και η σταθερότητα της τελικής διαδικασίας πρόβλεψης.

Κεφάλαιο 5

Ανάλυση και Οπτικοποίηση στο Power BI

5.1 Διασύνδεση του Power BI με τη βάση PostgreSQL του Supabase

Μετά την ολοκλήρωση του καθαρισμού, της αποθήκευσης και της δημιουργίας των SQL views, το επόμενο επίπεδο της εφαρμογής είναι η παρουσίαση και η ανάλυση των δεδομένων στο Power BI. Η σύνδεση δεν πραγματοποιείται με ανάγνωση των αρχικών Excel ή CSV αρχείων και δεν χρησιμοποιεί κάποια σύνδεση με το Python backend. Αντίθετα, το Power BI συνδέεται απευθείας με τη βάση PostgreSQL του Supabase μέσω του PostgreSQL connector του Power Query. Με αυτόν τον τρόπο το Power BI βασίζεται στα ίδια καθαρισμένα και τυποποιημένα δεδομένα που έχουν ήδη δημιουργηθεί από την εφαρμογή και δεν δημιουργείται δεύτερη διαδικασία καθαρισμού μέσα στο Power BI.

Η επιλογή της απευθείας σύνδεσης με τη PostgreSQL παρουσιάζει δύο βασικά πλεονεκτήματα. Πρώτον, διατηρεί μία ενιαία πηγή για τα δεδομένα, καθώς όλα τα dashboards διαβάζουν τα ίδια views που τροφοδοτούνται από τον backend μηχανισμό καθαρισμού και όχι αντιγραφές αρχείων σε ξεχωριστό περιβάλλον. Δεύτερον, μεταφέρει το πιο απαιτητικό μέρος της τυποποίησης στο επίπεδο της βάσης, όπου τα πεδία έχουν ήδη σταθερό τύπο, κοινή ονοματολογία και συγκεκριμένη σημασία. Έτσι το Power BI εστιάζει στην ανάλυση, στα measures και στην οπτικοποίηση, αντί να επαναλαμβάνει περίπλοκους μετασχηματισμούς δεδομένων.

Για τη δημιουργία της σύνδεσης επιλέχθηκε στο Power BI Desktop η πηγή *PostgreSQL database*. Η σύνδεση απαιτεί πέντε βασικά στοιχεία: τη διεύθυνση του PostgreSQL server, τη θύρα σύνδεσης, το όνομα της βάσης, το όνομα χρήστη και τον αντίστοιχο κωδικό πρόσβασης. Η διεύθυνση και η θύρα λαμβάνονται από τις ρυθμίσεις σύνδεσης της βάσης στο Supabase και στο πεδίο *Server* του Power BI εισάγονται στη μορφή `host:port`. Το συγκεκριμένο host της βάσης δεν πρέπει να συγχέεται

με τη μεταβλητή SUPABASE_URL που χρησιμοποιεί το backend για τις κλήσεις. Στην περίπτωση του Power BI χρησιμοποιείται PostgreSQL connection endpoint και πιστοποίηση χρήστη βάσης δεδομένων.

Για λόγους ασφάλειας, οι πραγματικές τιμές του host, της θύρας, του ονόματος της βάσης και κυρίως του κωδικού πρόσβασης δεν καταγράφονται στη διπλωματική. Η πληροφορία που είναι απαραίτητη για την αναπαραγωγή της διαδικασίας είναι ο τρόπος με τον οποίο χρησιμοποιούνται τα στοιχεία και όχι τα πραγματικά στοιχεία της βάσης. Η λογική της σύνδεσης μπορεί να αποδοθεί με την ακόλουθη μορφή:

Πίνακας 5.1: Στοιχεία της σύνδεσης Power BI--Supabase.

Στοιχείο	Μορφή στο Power BI	Χρήση
Server	<postgres-host>:<port>	PostgreSQL endpoint και θύρα που παρέχονται από τις ρυθμίσεις σύνδεσης του Supabase.
Database	<database-name>	Η PostgreSQL βάση στην οποία βρίσκονται οι πίνακες και τα reporting views.
Authentication	Database	Πιστοποίηση με λογαριασμό PostgreSQL και όχι με Supabase API key.
Username	powerbi_reader	Ξεχωριστός λογαριασμός που χρησιμοποιείται για την αναλυτική σύνδεση.
Password	<secret>	Ξεχωριστός κωδικός του database role, ο οποίος δεν καταγράφεται στο κείμενο της εργασίας.
Data connectivity	Import	Τα δεδομένα φορτώνονται στο semantic model και ανανεώνονται μέσω refresh.

Για την σύνδεση και την πιστοποίηση δεν χρησιμοποιήθηκε ο κύριος λογαριασμός της PostgreSQL ούτε το service_role key του backend. Δημιουργήθηκε ξεχωριστός ρόλος, ο powerbi_reader, με δυνατότητα σύνδεσης και με ανεξάρτητο password. Η επιλογή αυτή εφαρμόζει διαχωρισμό δικαιωμάτων μεταξύ των διαφορετικών τμημάτων της εφαρμογής. Το backend χρειάζεται αυξημένα δικαιώματα, επειδή εισάγει, ενημερώνει και διαγράφει εγγραφές κατά την επεξεργασία. Το Power BI, αντίθετα, έχει ως σκοπό την ανάγνωση και ανάλυση των ήδη επεξεργασμένων δεδομένων και επομένως δεν απαιτεί δικαίωμα επεξεργασίας.

Ο έλεγχος των δικαιωμάτων της βάσης επιβεβαιώνει ότι ο powerbi_reader διαθέτει SELECT στα βασικά καθαρισμένα views v_bookings_clean, v_income_clean και v_expenses_clean. Η ίδια αρχή read-only πρόσβασης εφαρμόζεται και στα forecasting

views που χρησιμοποιούνται από το τελικό μοντέλο. Έτσι η αναφορά μπορεί να εκτελεί queries και refresh χωρίς να αποκτά δυνατότητα INSERT, UPDATE ή DELETE στα δεδομένα. Ο διαχωρισμός αυτός είναι ουσιαστικός, επειδή ένα dashboard δεν πρέπει να μπορεί να τροποποιήσει την πηγή από την οποία αντλεί τα αποτελέσματά του.

Η λειτουργία σύνδεσης επιλέχθηκε ως *Import*. Συνεπώς, κατά το refresh το Power Query συνδέεται στη PostgreSQL, εκτελεί τα queries των επιλεγμένων πηγών και φορτώνει το αποτέλεσμα στο μοντέλο του `.pbix`. Οι οπτικοποιήσεις και τα DAX measures υπολογίζονται στη συνέχεια πάνω στο εισαγμένο μοντέλο. Η επιλογή *Import* κρίθηκε καταλληλότερη από ένα *DirectQuery*, επειδή το μοντέλο περιλαμβάνει αρκετά υπολογιστικά measures, κοινά φίλτρα μεταξύ πολλών σελίδων και συνδυαστικές οπτικοποιήσεις που ωφελούνται από την τοπική φόρτωση των δεδομένων στη μνήμη του Power BI. Όταν προστεθούν νέα αρχεία, επανεκτελεστεί ο μηχανισμός καθαρισμού ή δημιουργηθούν νέα forecasts, απαιτείται refresh του Power BI ώστε η τοπική έκδοση να ενημερωθεί με την τελευταία κατάσταση της βάσης.

5.2 Πηγές δεδομένων και οργάνωση του μοντέλου

Το Power BI δεν διαβάζει απευθείας τις γενικές JSONB εγγραφές του `file_rows`. Η επιλογή αυτή είναι σκόπιμη. Στα Κεφάλαια 3 και 4 παρουσιάστηκε ότι ο `file_rows` αποτελεί το επίπεδο αποθήκευσης, ενώ τα SQL views μετατρέπουν το `cleaned_data` σε PostgreSQL πεδία με σαφείς τύπους δεδομένων. Το Power BI συνδέεται κυρίως με αυτό το επίπεδο, επειδή εκεί οι ημερομηνίες, τα ποσά, οι κατηγορίες και τα δεδομένα των ξενοδοχείων έχουν ήδη αποκτήσει σταθερή μορφή.

Στο τελικό μοντέλο τα τρία βασικά clean views εμφανίζονται με ελληνικά ονόματα για να είναι ευκολότερη η δημιουργία measures και visuals: το `v_income_clean` έχει εισαχθεί ως ΕΣΟΔΑ, το `v_expenses_clean` ως ΕΞΟΔΑ και το `v_bookings_clean` ως ΚΡΑΤΗΣΕΙΣ. Παράλληλα φορτώνονται το μηνιαίο view πληρότητας, ο πίνακας ξενοδοχείων και τα views που έχουν δημιουργηθεί ειδικά για την παρουσίαση των forecasts.

Πίνακας 5.2: Κύριες πηγές του Supabase που χρησιμοποιούνται στο Power BI.

Αντικείμενο Supabase	Όνομα στο μοντέλο	Ρόλος
v_income_clean	ΕΣΟΔΑ	Καθαρά ποσά εσόδων, inclusion flag, κανάλι εσόδου, customer mappings και στοιχεία παραστατικού.
v_expenses_clean	ΕΞΟΔΑ	Κατηγορίες και υποκατηγορίες εξόδων, included ποσά, κανόνες κατηγοριοποίησης και οικονομικά πεδία.
v_bookings_clean	ΚΡΑΤΗΣΕΙΣ	Ενεργές και ακυρωμένες κρατήσεις, ημερομηνίες, δωμάτια, τύποι δωματίων, άτομα, διαμονή και booking revenue.
v_hotel_occupancy_monthly	publicv_hotel_occupancy_monthly	Διαθέσιμα, κατειλημμένα και υπολειπόμενα room-nights και μηνιαία πληρότητα.
hotels	publichotels	Διάσταση ξενοδοχείου και βάση του hotel slicer.
v_powerbi_forecast_predictions	publicv_powerbi_forecast_predictions	Μηνιαίες προβλέψεις ανά target, model, strategy και horizon.
v_powerbi_forecast_evaluation	publicv_powerbi_forecast_evaluation	MAE, RMSE, sMAPE και Bias των αξιολογημένων configurations.
v_powerbi_forecast_selections	publicv_powerbi_forecast_selections	Επίσημες επιλογές μοντέλου και αποτελέσματα της διαδικασίας selection.
v_powerbi_latest_forecast_runs	publicv_powerbi_latest_forecast_runs	Metadata των τελευταίων ολοκληρωμένων forecasting runs που χρησιμοποιούνται στο reporting.

Η παραπάνω δομή διατηρεί σαφή διαχωρισμό ανάμεσα στην αποθήκευση και στην παρουσίαση. Για παράδειγμα, το v_income_clean αντλεί τις καθαρισμένες εγγραφές από τον file_rows και τις συνδέει με τον uploaded_files μέσω του source_file_id, ώστε το Power BI να λαμβάνει μαζί με το οικονομικό αποτέλεσμα και πληροφορίες προέλευσης του αρχείου. Αντίστοιχη λογική εφαρμόζεται στις κρατήσεις και στα έξοδα. Οι πίνακες file_rows και uploaded_files παραμένουν σημαντικοί για την καταγραφή της προέλευσης και της

πορείας των δεδομένων. Ωστόσο, το Power BI χρησιμοποιεί τα ήδη επεξεργασμένα SQL views, οπότε δεν χρειάζεται να επαναλαμβάνει τον μετασχηματισμό των πεδίων.

Εκτός από τις πηγές του Supabase, το μοντέλο περιλαμβάνει βοηθητικούς πίνακες που έχουν δημιουργηθεί στο Power BI. Το ΗΜΕΡΟΛΟΓΙΟ αποτελεί την κοινή χρονική διάσταση και παρέχει πεδία όπως ημερομηνία, έτος, μήνας και αρχή μήνα. Η ύπαρξή του είναι κρίσιμη, επειδή εξασφαλίζει ότι όλες οι σελίδες χρησιμοποιούν το ίδιο ημερολογιακό πλαίσιο και ότι τα γραφήματα, οι χρονοσειρές και οι συγκρίσεις μεταξύ ετών στηρίζονται στην ίδια χρονική λογική. Οι DimForecastModel, DimForecastStrategy και DimForecastHorizon χρησιμοποιούνται για τον έλεγχο της σελίδας forecasting, ενώ ο ForecastMeasures λειτουργεί ως συγκεντρωτικός πίνακας για measures. Ο ExpenseGroups χρησιμοποιείται για την ομαδοποίηση των αναλυτικών expense categories σε μεγαλύτερες επιχειρησιακές ομάδες, όπως Payroll, Operating Costs, Marketing και Food & Beverage.

Οι βασικοί πίνακες του Power BI είναι οι ΕΣΟΔΑ, ΕΞΟΔΑ, ΚΡΑΤΗΣΕΙΣ και τα forecasting views. Οι πίνακες αυτοί συνδέονται με κοινές διαστάσεις, όπως ο χρόνος, το ξενοδοχείο και οι παράμετροι του forecasting. Έτσι, όταν ο χρήστης επιλέγει ένα έτος, έναν μήνα ή ένα ξενοδοχείο, όλα τα σχετικά γραφήματα, οι δείκτες και οι πίνακες ενημερώνονται ταυτόχρονα. Με αυτόν τον τρόπο, τα φίλτρα εφαρμόζονται με ενιαίο τρόπο σε όλη τη σελίδα και ο χρήστης βλέπει πάντα συνεπή αποτελέσματα.

5.3 Φίλτρα, navigation buttons και εναλλαγή οπτικοποιήσεων

Η διαδραστικότητα του dashboard σχεδιάστηκε ώστε ο χρήστης να μπορεί να μετακινείται ανάμεσα στις διαφορετικές αναλύσεις με εύκολο τρόπο. Στο πάνω μέρος των κύριων σελίδων υπάρχει κοινή περιοχή πλοήγησης με action buttons. Τα κουμπιά χρησιμοποιούν τη λειτουργία *Page navigation* και οδηγούν στις σελίδες *Overview*, *Revenue Analysis*, *Expenses Analysis*, *Bookings Analysis*, *Room Performance*, *Stay Length Analysis*, *Occupancy Analysis* και *Forecasting*. Με αυτόν τον τρόπο ο χρήστης μπορεί να ξεκινήσει από τη συνολική εικόνα και να μεταβεί άμεσα στην επιμέρους κατάσταση που θέλει να διερευνήσει, χωρίς να αλλάζει αρχείο ή περιβάλλον.



Σχήμα 5.1: Κουμπιά πλοήγησης μεταξύ των σελίδων του Power BI dashboard.

Στις βασικές σελίδες ανάλυσης επαναλαμβάνονται τρεις κύριοι slicers: *Year*, *Month*

και *Hotel*. Ο χρονικός έλεγχος βασίζεται στον πίνακα ΗΜΕΡΟΛΟΓΙΟ, ενώ το φίλτρο ξενοδοχείου βασίζεται στη διάσταση *publichotels*. Όταν τα Year και Month παραμένουν στην επιλογή *All*, τα γραφήματα παρουσιάζουν ολόκληρο το διαθέσιμο ιστορικό. Με την επιλογή συγκεκριμένου έτους ή μήνα, τα cards, οι γραμμές τάσης, οι κατανομές και οι πίνακες επανυπολογίζονται στο ίδιο φίλτρο. Η επιλογή του ξενοδοχείου λειτουργεί με την ίδια λογική και επιτρέπει η ίδια αναφορά να χρησιμοποιείται για περισσότερες μονάδες χωρίς ξεχωριστό *.pbix*.

Σχήμα 5.2: Επιλογή έτους.

Σχήμα 5.3: Επιλογή μήνα.

Σχήμα 5.4: Επιλογή ξενοδοχείου.

Εκτός από την πλοήγηση μεταξύ σελίδων, χρησιμοποιήθηκαν buttons που ενεργοποιούν *Bookmarks*. Τα bookmarks δεν ανοίγουν νέα σελίδα, αλλά αλλάζουν ποια visuals εμφανίζονται στον ίδιο χώρο. Η επιλογή αυτή είναι σημαντική για την ανάγνωση του dashboard, επειδή επιτρέπει στον χρήστη να μεταβαίνει από ένα συνοπτική γράφημα σε ένα πιο αναλυτικό επίπεδο χωρίς να χάνει τα ενεργά φίλτρα του. Επομένως, η εναλλαγή δεν αποτελεί διαφορετικό υπολογισμό των δεδομένων αλλά διαφορετικό τρόπο παρουσίασης του ίδιου συνόλου.

Στη σελίδα *Revenue Analysis* τα buttons *Overview Chart*, *Online Travel Agency*, *Travel Agency*, *Business* και *Direct* μεταβάλλουν το επίπεδο ανάλυσης των εσόδων. Η αρχική κατάσταση παρουσιάζει τη συνολική εξέλιξη και την κατανομή ανά κανάλι. Με

την επιλογή ενός καναλιού ο χρήστης περνά στη σύνθεση του συγκεκριμένου τμήματος, ώστε να μπορεί να εξετάσει όχι μόνο πόσο συμμετέχει το κανάλι στο συνολικό revenue αλλά και από ποιους συνεργάτες προέρχεται. Αντίστοιχα, στη σελίδα *Expenses Analysis* τα *Total*, *Operating Expenses* και *Operating Expenses Detail* μετακινούν την ανάλυση από τη συνολική δαπάνη προς τις ομάδες και στη συνέχεια στις αναλυτικότερες κατηγορίες. Τα *View Full Table* και *Back to Chart* επιτρέπουν εναλλαγή ανάμεσα στη γρήγορη οπτική σύγκριση και στις ακριβείς τιμές του πίνακα.

Παρόμοια λογική εφαρμόζεται και στις υπόλοιπες σελίδες. Στο *Room Performance* τα *View Full Table* και *Back to Charts* εναλλάσσουν τη σύγκριση των τύπων δωματίου με την αναλυτική παρουσίαση των KPIs. Στο *Stay Length Analysis* τα *Monthly Trend* και *Back to Distribution* επιτρέπουν μετάβαση από την κατανομή της διάρκειας διαμονής στη χρονική της εξέλιξη. Στη σελίδα κρατήσεων τα *View all countries* επεκτείνουν την ανάλυση από το συγκεντρωτικό γράφημα σε έναν πιο αναλυτικό πίνακα. Έτσι τα buttons λειτουργούν ως μηχανισμός διερεύνησης και όχι απλώς ως αισθητικό στοιχείο της αναφοράς.

Πίνακας 5.3: Διαδραστικά στοιχεία ανά βασική σελίδα του report.

Σελίδα	Κύρια φίλτρα	Πρόσθετα buttons / bookmarks
Overview	Year, Month, Hotel	Κοινό navigation προς όλες τις αναλυτικές σελίδες.
Revenue Analysis	Year, Month, Hotel	Overview Chart, Online Travel Agency, Travel Agency, Business, Direct.
Expenses Analysis	Year, Month, Hotel	Total, Operating Expenses, Operating Expenses Detail, View Full Table, Back to Chart.
Bookings Analysis	Year, Month, Hotel	View all countries και επιστροφή στις βασικές απεικονίσεις.
Room Performance	Year, Month, Hotel	View Full Table, Back to Charts.
Stay Length Analysis	Year, Month, Hotel	Monthly Trend, Back to Distribution.
Occupancy Analysis	Year, Month, Hotel	Κοινή πλοήγηση και συνδυαστικές απεικονίσεις πληρότητας/ADR.
Forecasting	Hotel, Model, Strategy, Horizon	Δυναμική επιλογή μοντέλου, στρατηγικής και χρονικού ορίζοντα.

Η σελίδα forecasting διαφοροποιείται από τις προηγούμενες σελίδες, επειδή το βασικό ερώτημα δεν είναι μόνο "τι συνέβη σε ένα έτος ή μήνα", αλλά "πώς αλλάζει η εκτίμηση όταν αλλάζει η μέθοδος πρόβλεψης". Για τον λόγο αυτό χρησιμοποιούνται οι slicers

Hotel, *Forecast Model*, *Training Strategy* και *Forecast Horizon*. Ο *Forecast Model* επιτρέπει επιλογή μεταξύ ETS Additive, LSTM, SARIMA, Seasonal Naive, Seasonal Ridge και Weighted Seasonal Mean. Ο *Training Strategy* επιτρέπει σύγκριση των δύο στρατηγικών που χρησιμοποιούνται στην παρούσα υλοποίηση: *All History*, όπου αξιοποιείται όλο το διαθέσιμο ιστορικό μέχρι το αντίστοιχο forecast origin, και *Last 30 Months*, όπου η εκπαίδευση περιορίζεται στις τελευταίες 30 διαθέσιμες μηνιαίες παρατηρήσεις. Τέλος, ο *Forecast Horizon* επιτρέπει επιλογή 1, 3, 6 ή 12 μηνών. Οι επιλογές αυτές δεν αλλάζουν τις πραγματικές ιστορικές τιμές, αλλά καθορίζουν ποιοι μέθοδοι πρόβλεψης και ποια αποτελέσματα αξιολόγησης θα παρουσιαστούν.

5.4 Υπολογιστικό επίπεδο DAX και βασικοί δείκτες

Πάνω από τις εισαγόμενες πηγές δημιουργήθηκε ξεχωριστό υπολογιστικό επίπεδο με DAX measures. Στο τελικό μοντέλο υπάρχουν περίπου εβδομήντα measures, τα οποία οργανώνονται σε ομάδες κρατήσεων, εσόδων, εξόδων, πληρότητας και forecasting. Η χρήση measures αντί αποθήκευσης στατικών αποτελεσμάτων επιτρέπει κάθε KPI να υπολογίζεται δυναμικά όταν αλλάζει το Year, Month, Hotel ή κάποιο άλλο φίλτρο.

Στις κρατήσεις ο βασικός δείκτης TotalBookings δεν αποτελεί απλό COUNTROWS. Χρησιμοποιείται διακριτό πλήθος του booking_code και ταυτόχρονα εφαρμόζονται οι συνθήκες ότι η κράτηση είναι ενεργή και ότι υπάρχει πραγματική ημερομηνία άφιξης:

```
Total Bookings =
CALCULATE (
    DISTINCTCOUNT ( 'ΚΡΑΤΗΣΕΙΣ' [booking_code] ),
    KEEPFILTERS (
        'ΚΡΑΤΗΣΕΙΣ' [booking_status] = "Ενεργή"
    ),
    KEEPFILTERS (
        'ΚΡΑΤΗΣΕΙΣ' [arrival_date] <> BLANK ()
    )
)
```

Η λογική είναι αντίστοιχη με εκείνη που χρησιμοποιήθηκε στη δημιουργία της μηνιαίας σειράς κρατήσεων στο Κεφάλαιο 4. Το measure προστατεύει τα αποτελέσματα από διπλότυπη μέτρηση του ίδιου booking code και από εγγραφές που δεν αντιπροσωπεύουν ενεργή διαμονή.

Για τα λογιστικά έσοδα χρησιμοποιείται το TotalIncludedRevenue(Net). Το measure αθροίζει μόνο το net_amount των εγγραφών που έχουν ήδη χαρακτηριστεί από τον cleaner ως κατάλληλες για τον αναλυτικό τζίρο:

```
Total Included Revenue (Net) =
```

```

CALCULATE (
    SUM('ΕΣΟΔΑ'[net_amount]),
    KEEPFILTERS (
        'ΕΣΟΔΑ'[include_in_turnover] = TRUE()
    )
)

```

Αντίστοιχα, το βασικό measure εξόδων εφαρμόζει το included_in_expenses:

```

Total Included Expenses (Net) =
CALCULATE (
    SUM('ΕΞΟΔΑ'[net_amount]),
    KEEPFILTERS (
        'ΕΞΟΔΑ'[included_in_expenses] = TRUE()
    )
)

```

Με αυτόν τον τρόπο τα included/excluded ποσά που πραγματοποιήθηκαν στον Python cleaner και στα SQL views συνεχίζουν να ισχύουν μέχρι το τελικό dashboard. Το Power BI δεν επιχειρεί να αποφασίσει ξανά ποια λογιστική γραμμή είναι πραγματικό λειτουργικό έσοδο ή έξοδο. Χρησιμοποιεί την ήδη επιχειρησιακή ταξινόμηση και περιορίζεται στην αναλυτική συγκέντρωση των αποτελεσμάτων.

Για τη σύγκριση περιόδων έχει δημιουργηθεί και μέτρο ετήσιας μεταβολής των εσόδων. Ενδεικτικά, το RevenueGrowthYoY% υπολογίζει τη μεταβολή του included revenue έναντι της αντίστοιχης περιόδου του προηγούμενου έτους:

```

Revenue Growth YoY % =
VAR OneYearSelected =
    HASONEVALUE('ΗΜΕΡΟΛΟΓΙΟ'[Έτος])
VAR CurrentRevenue =
    [Total Included Revenue (Net)]
VAR PreviousYearRevenue =
    CALCULATE (
        [Total Included Revenue (Net)],
        SAMEPERIODLASTYEAR('ΗΜΕΡΟΛΟΓΙΟ'[Date])
    )
RETURN
IF(
    OneYearSelected,
    DIVIDE (
        CurrentRevenue - PreviousYearRevenue,
        PreviousYearRevenue
    ),
    BLANK()
)

```

Για την πληρότητα χρησιμοποιούνται τα ήδη υπολογισμένα μηνιαία μεγέθη του v_hotel_occupancy_monthly. Το DAX measure είναι λόγος των συνολικών occupied room-nights προς τα available room-nights :

```
Occupancy % =  
DIVIDE(  
    SUM(  
        'public v_hotel_occupancy_monthly'[occupied_room_nights]  
    ),  
    SUM(  
        'public v_hotel_occupancy_monthly'[available_room_nights]  
    ),  
    0  
)
```

Άρα ο υπολογισμός της δυναμικότητας των 16 δωματίων και η ημερήσια ανάπτυξη των κρατήσεων δεν επαναλαμβάνονται στο Power BI. Έχουν ήδη ολοκληρωθεί στο περιβάλλον της βάσης και το DAX πραγματοποιεί τη σωστή συγκέντρωση για το επιλεγμένο διάστημα. Πάνω σε αυτό το measure δημιουργείται ο RevPAR ως γινόμενο ADR και Occupancy%. Επιπλέον έχουν δημιουργηθεί οι δείκτες GOP, GOPPAR και CPOR, ώστε η πληρότητα να συνδέεται με οικονομικούς δείκτες απόδοσης και κόστους.

Το forecasting χρησιμοποιεί measures που φιλτράρουν το v_powerbi_forecast_predictions ως προς το target και το prediction scope. Για παράδειγμα, το forecast των εσόδων ορίζεται ως:

```
Forecast Revenue =  
CALCULATE(  
    SUM(  
        'public v_powerbi_forecast_predictions'[predicted_value]  
    ),  
    'public v_powerbi_forecast_predictions'[prediction_scope]  
    = "base_model",  
    'public v_powerbi_forecast_predictions'[target_name]  
    = "included_net_revenue"  
)
```

Το συγκεκριμένο measure δεν περιέχει μέσα στον τύπο του σταθερό όνομα μοντέλου ή ορίζοντα. Αυτά προκύπτουν από το filter context των DimForecastModel, DimForecastStrategy και DimForecastHorizon. Η ίδια λογική χρησιμοποιείται για τα ForecastBookings και ForecastExpenses. Αντίστοιχα, για την εμφάνιση της ενεργής στρατηγικής εκπαίδευσης σε τίτλους ή cards μπορεί να χρησιμοποιηθεί ένα απλό measure επιλογής, όπως το ακόλουθο:

```

Selected Training Strategy =
SELECTEDVALUE (
    DimForecastStrategy[training_strategy_label],
    "Select Strategy"
)

```

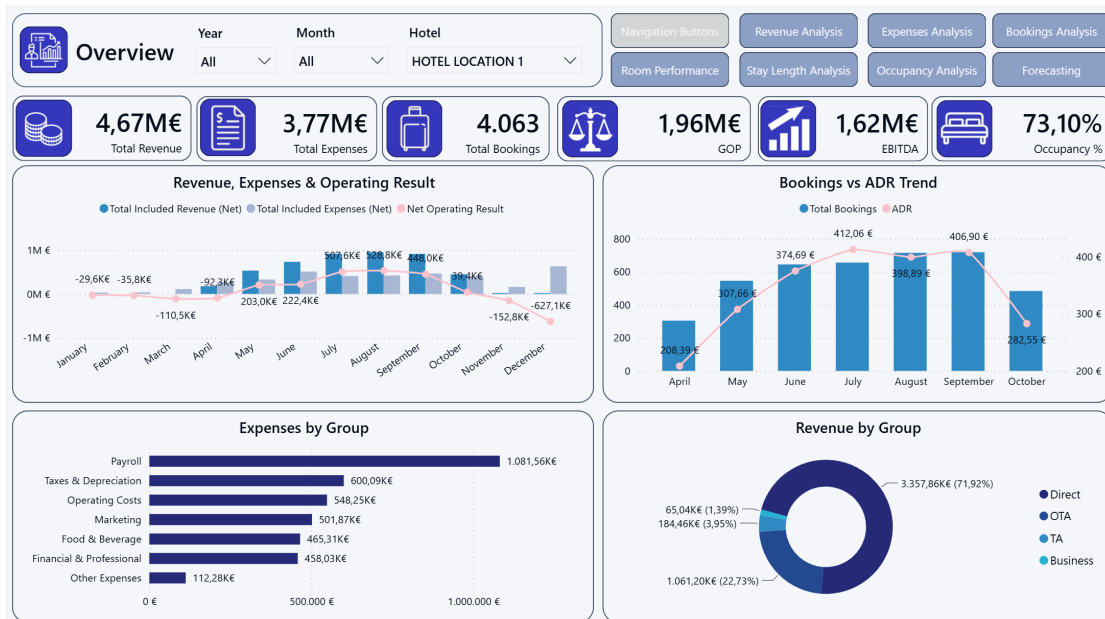
Οι MAE, RMSE, sMAPE και Bias διαβάζονται από το `v_powerbi_forecast_evaluation` και προσαρμόζονται στα φίλτρα που έχουν επιλεγεί στο dashboard.

Ιδιαίτερη περίπτωση αποτελούν τα measures `ActualBookingsLine`, `ForecastBookingsLine`, `ActualRevenueLine`, `ForecastRevenueLine`, `ActualExpensesLine` και `ForecastExpensesLine`. Τα measures αυτά χρησιμοποιούνται για τη σύνδεση των πραγματικών με τις προβλεπόμενες τιμές στα line charts. Εντοπίζουν το πρώτο forecast month και το τελευταίο actual month και διασφαλίζουν ότι οι δύο σειρές ενώνονται οπτικά στο σωστό χρονικό σημείο, χωρίς να παρουσιάζονται forecast τιμές μέσα στην ιστορική περίοδο.

5.5 Overview και οικονομική ανάλυση εσόδων--εξόδων

Η πρώτη σελίδα του report λειτουργεί ως συνοπτικό επίπεδο πληροφόρησης. Τα cards συγκεντρώνουν σε ένα σημείο τους βασικούς δείκτες της λειτουργίας, όπως included καθαρά έσοδα, included έξοδα, ενεργές κρατήσεις, GOP, EBITDA και πληρότητα. Σκοπός τους δεν είναι να αποτελέσουν από μόνα τους την ανάλυση, αλλά να δώσουν στον χρήστη ένα γρήγορο σημείο αναφοράς πριν εξετάσει τις πιο συγκεκριμένες κατηγορίες. Η συνολική διάταξη της σελίδας, τα κοινά φίλτρα Year--Month--Hotel και η περιοχή πλοήγησης παρουσιάζονται στο Σχήμα 5.5.

Ο δείκτης *GOP (Gross Operating Profit)* εκφράζει το μικτό λειτουργικό κέρδος, ενώ ο δείκτης *EBITDA (Earnings Before Interest, Taxes, Depreciation and Amortization)* εκφράζει τα κέρδη πριν από τόκους, φόρους και αποσβέσεις.



Σχήμα 5.5: Κεντρική σελίδα επισκόπησης (Overview) του Power BI report.

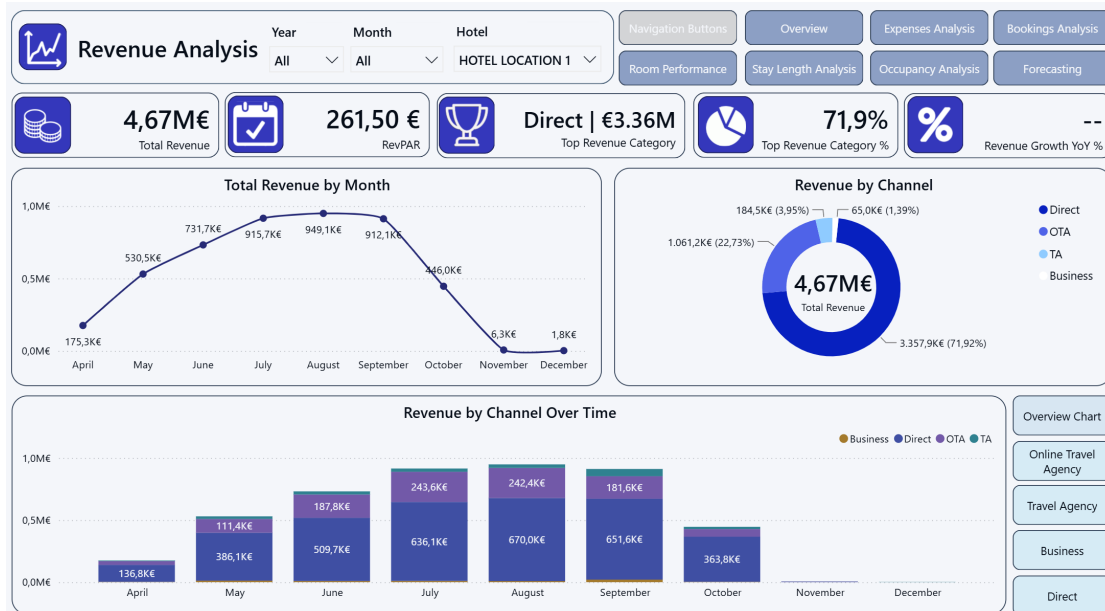
Το γράφημα *Revenue, Expenses & Operating Result* είναι το βασικό οικονομικό γράφημα της σελίδας. Παρουσιάζει στην ίδια χρονική κλίμακα τα included έσοδα, τα included έξοδα και το λειτουργικό αποτέλεσμα. Η ταυτόχρονη παρουσία των τριών μεγεθών επιτρέπει να διαπιστωθεί αν μία αύξηση των εσόδων συνοδεύεται από αντίστοιχη αύξηση του κόστους. Αντίστοιχα, σε μήνες χαμηλότερης δραστηριότητας γίνεται ορατό αν τα έξοδα μειώνονται με τον ίδιο ρυθμό ή αν παραμένει ένα σχετικά σταθερό επίπεδο κόστους. Επομένως, το γράφημα δεν δείχνει μόνο το ύψος των οικονομικών μεγεθών αλλά τη μεταξύ τους σχέση μέσα στον χρόνο.

Το *Bookings vs ADR Trend* συνδέει δύο διαφορετικές πλευρές της εμπορικής λειτουργίας: τον όγκο των κρατήσεων και τη μέση ημερήσια τιμή. Η κοινή χρονική απεικόνιση επιτρέπει να εξεταστεί αν οι περίοδοι αυξημένης ζήτησης συμπίπτουν με υψηλότερο ADR ή αν η τιμή μεταβάλλεται ανεξάρτητα από τον αριθμό των κρατήσεων. Τα γραφήματα σύνθεσης στο κάτω μέρος λειτουργούν συμπληρωματικά: παρουσιάζουν από ποιες κατηγορίες προέρχονται τα έσοδα και σε ποιες ομάδες κατανέμονται τα έξοδα. Με αυτόν τον τρόπο η σελίδα Overview απαντά αρχικά στο "ποια είναι η συνολική εικόνα" και στη συνέχεια κατευθύνει τον χρήστη στις αναλυτικότερες σελίδες μέσω των navigation buttons.

5.5.1 Revenue Analysis

Η σελίδα *Revenue Analysis* απομονώνει την πλευρά των included εσόδων και εξετάζει τόσο τη χρονική τους εξέλιξη όσο και την προέλευσή τους. Τα cards χρησιμοποιούνται κυρίως ως συνοπτικοί δείκτες, για παράδειγμα για το συνολικό included revenue, το RevPAR, τη μεταβολή RevenueGrowthYoY% και το κυρίαρχο revenue channel. Ο χρή-

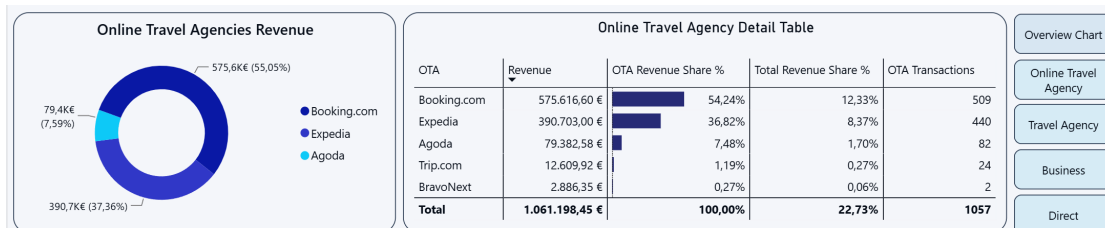
στης μπορεί έτσι να αποκτήσει γρήγορη εικόνα τόσο για το συνολικό ύψος του τζίρου όσο και για το αν η τρέχουσα περίοδος εμφανίζει βελτίωση ή υποχώρηση σε σχέση με την αντίστοιχη προηγούμενη.



Σχήμα 5.6: Σελίδα Revenue Analysis του Power BI με τα βασικά φίλτρα, τα navigation buttons και τη συνολική ανάλυση των καναλιών εσόδων.

Το κύριο μηνιαίο γράφημα της σελίδας αναδεικνύει την εποχικότητα των εσόδων και επιτρέπει τη σύγκριση των ιδίων περιόδων μεταξύ διαφορετικών ετών όταν εφαρμόζονται τα αντίστοιχα φίλτρα. Η χρησιμότητά του βρίσκεται στο ότι η εποχικότητα γίνεται ορατή ως επαναλαμβανόμενο χρονικό μοτίβο. Παράλληλα, το γράφημα κατανομής ανά κανάλι εξηγεί πώς σχηματίζεται το συνολικό revenue: ποιο μέρος προέρχεται από *Direct*, ποιο από *Online Travel Agencies*, ποιο από *Travel Agencies* και ποιο από *Business*. Έτσι η σελίδα δεν απαντά μόνο στο "πόσο" είναι το έσοδο, αλλά και στο "από πού" προέρχεται.

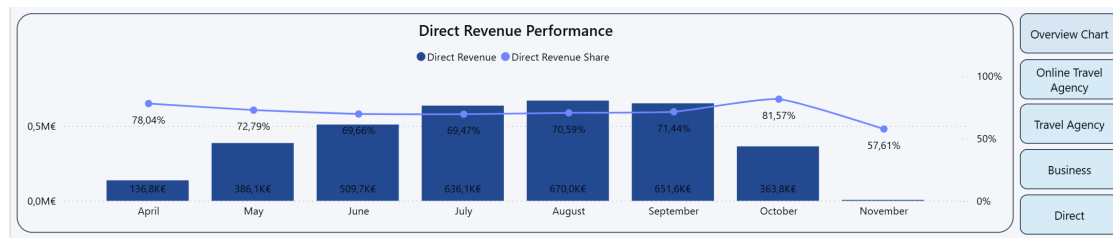
Ιδιαίτερη σημασία έχουν τα bookmark buttons *Overview Chart*, *Online Travel Agency*, *Travel Agency*, *Business* και *Direct*. Με το *Overview Chart* ο χρήστης βλέπει τη συνολική κατανομή των καναλιών, ενώ με τα υπόλοιπα buttons μεταβαίνει σε ανάγνωση του κάθε τμήματος.



Σχήμα 5.7: Ανάλυση του καναλιού Online Travel Agency στη σελίδα Revenue Analysis.

Στην κατάσταση *Online Travel Agency*, το donut chart δείχνει πώς κατανέμεται το revenue μεταξύ των OTA συνεργατών, ενώ ο αναλυτικός πίνακας δείχνει ταυτόχρονα το έσοδο, το ποσοστό συμμετοχής του κάθε OTA μέσα στο ίδιο το κανάλι, το ποσοστό συμμετοχής του στα συνολικά έσοδα και τον αριθμό συναλλαγών. Το συγκεκριμένο visual είναι ιδιαίτερα χρήσιμο επειδή επιτρέπει δύο επίπεδα ανάγνωσης: ποιος OTA είναι ισχυρότερος μεταξύ των συνεργατών και πόσο σημαντικός είναι συνολικά για το ξενοδοχείο. Έτσι, για παράδειγμα, ένας συνεργάτης μπορεί να κυριαρχεί μέσα στο OTA κανάλι αλλά το ίδιο το κανάλι να αποτελεί μικρότερο μέρος του συνολικού revenue, ή αντίστροφα.

Αντίστοιχη λογική ακολουθεί και η κατάσταση *Travel Agency*, όπου ο χρήστης βλέπει τη σύνθεση των πρακτορείων, καθώς και η κατάσταση *Business*, όπου αποτυπώνεται η χρονική εξέλιξη και η συμμετοχή των εταιρικών συνεργατών. Οι εικόνες αυτές επιτρέπουν να εξεταστεί η εξάρτηση από συγκεκριμένους συνεργάτες.



Σχήμα 5.8: Ανάλυση του καναλιού Direct στη σελίδα Revenue Analysis.

Στην κατάσταση *Direct*, το γράφημα *Direct Revenue Performance* δείχνει όχι μόνο το μηνιαίο direct revenue αλλά και το ποσοστό συμμετοχής του direct channel στα συνολικά έσοδα. Ο συνδυασμός ράβδων και γραμμής είναι κρίσιμος, επειδή διαχωρίζει το απόλυτο μέγεθος από την σχετική βαρύτητα. Ένας μήνας μπορεί να έχει υψηλό direct revenue αλλά χαμηλότερο direct share εάν τα υπόλοιπα κανάλια αυξάνονται ακόμη περισσότερο. Αντίστροφα, μπορεί να υπάρχει μικρότερο direct revenue αλλά υψηλότερο ποσοστό συμμετοχής σε μια περίοδο συνολικά χαμηλότερης δραστηριότητας. Άρα το συγκεκριμένο γράφημα εξηγεί πότε το direct channel είναι απλώς ισχυρό σε αξία και πότε είναι στρατηγικά κυρίαρχο στο revenue.

5.5.2 Expenses Analysis

Η σελίδα *Expenses Analysis* ακολουθεί την ίδια λογική, αλλά μεταφέρει το ενδιαφέρον από την προέλευση στη δομή του λειτουργικού κόστους. Τα cards συνοψίζουν το επίπεδο των included εξόδων και δείκτες όπως CPOR και Expense Ratio, ενώ το βασικό ζητούμενο είναι να γίνει κατανοητό πώς σχηματίζεται το συνολικό κόστος και πώς μεταβάλλεται μέσα στον χρόνο.

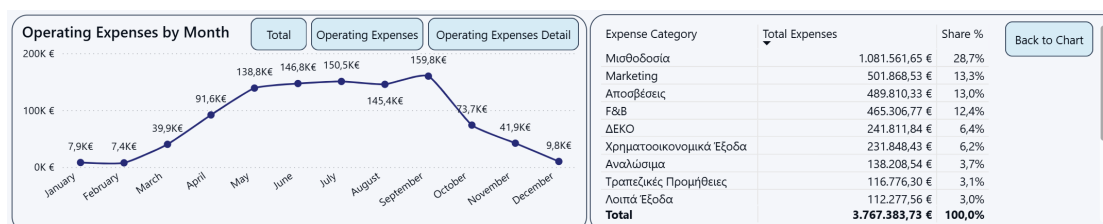
Ο δείκτης *CPOR (Cost per Occupied Room)* εκφράζει το κόστος ανά κατειλημμένο

δωμάτιο, ενώ το *Expense Ratio* εκφράζει τον λόγο των included καθαρών εξόδων προς τα included καθαρά έσοδα.



Σχήμα 5.9: Σελίδα Expenses Analysis του Power BI με τη χρονική εξέλιξη και την ανάλυση των κατηγοριών εξόδων.

Το μηνιαίο γράφημα συνολικού κόστους επιτρέπει τον εντοπισμό περιόδων με αυξημένες δαπάνες, ενώ το γράφημα κατανομής ή ομαδοποίησης αναλύει το ίδιο ύψος κόστους στις επιχειρησιακές ομάδες. Η δεύτερη απεικόνιση είναι ιδιαίτερα σημαντική, επειδή μία αύξηση του συνολικού ποσού μπορεί να έχει διαφορετική ερμηνεία αν οφείλεται κυρίως στη μισθοδοσία, στο Food & Beverage, στο marketing ή σε περισσότερες κατηγορίες ταυτόχρονα.



Σχήμα 5.10: Εναλλαγή της σελίδας Expenses Analysis σε αναλυτικότερη προβολή κατηγοριών εξόδων.

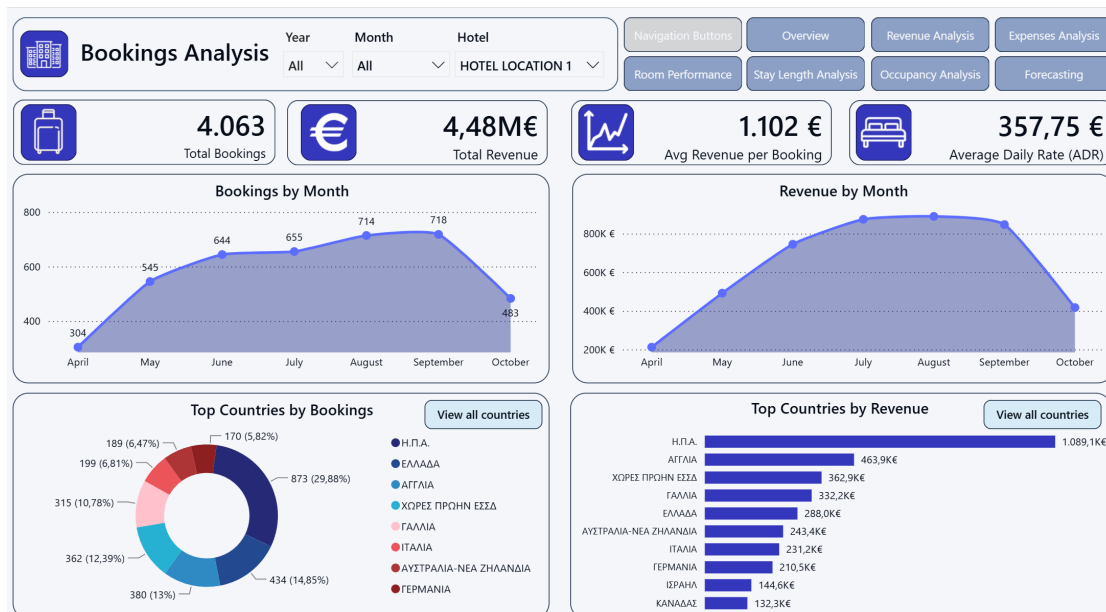
Τα buttons *Total*, *Operating Expenses* και *Operating Expenses Detail* επιτρέπουν τη μετάβαση από τη γενική εικόνα σε μεγαλύτερη λεπτομέρεια. Όταν εμφανίζεται ο αναλυτικός πίνακας, ο χρήστης μπορεί να δει όχι μόνο την ονομασία κάθε κατηγορίας αλλά και το συνολικό ποσό και το ποσοστό συμμετοχής της στο κόστος. Αυτή η δυνατότητα είναι πολύ χρήσιμη για την διοικητική ανάγνωση του dashboard, επειδή επιτρέπει γρήγορο

εντοπισμό των βασικών κατηγοριών κόστους χωρίς να απαιτείται εξαγωγή των δεδομένων σε ξεχωριστό αρχείο.

5.6 Κρατήσεις, δωμάτια, διάρκεια διαμονής και πληρότητα

5.6.1 Bookings Analysis

Η σελίδα *Bookings Analysis* εξετάζει την πλευρά της ζήτησης. Είναι σημαντικό να διαχωρίζεται το revenue των κρατήσεων από το included λογιστικό revenue της σελίδας *Revenue Analysis*. Το πρώτο βασίζεται στην αξία των ενεργών διαμονών ενώ το δεύτερο προέρχεται από τις λογιστικές εγγραφές. Η ύπαρξη των δύο μεγεθών δεν αποτελεί ασυνέπεια αλλά δύο διαφορετικές οπτικές της λειτουργίας: την λειτουργική αξία των διαμονών και την καθαρή λογιστική αναγνώριση του εσόδου.



Σχήμα 5.11: Σελίδα Bookings Analysis του Power BI με τους βασικούς δείκτες κρατήσεων και τις αναλύσεις ανά χρόνο και προέλευση πελατών.

Τα μηνιαία γραφήματα παρουσιάζουν πότε συγκεντρώνεται ο μεγαλύτερος αριθμός κρατήσεων και πώς μεταβάλλεται η αξία των διαμονών. Η σύγκριση αυτή επιτρέπει να διαπιστωθεί αν ένας μήνας εμφανίζει υψηλή συνολική αξία επειδή έχει περισσότερες κρατήσεις ή επειδή οι κρατήσεις έχουν υψηλότερη μέση αξία. Τα country visuals δείχνουν από ποιες αγορές προέρχεται η ζήτηση. Η αρχική προβολή περιορίζεται στις σημαντικότερες χώρες ώστε το γράφημα να παραμένει ευανάγνωστο, ενώ τα buttons *View all countries* εμφανίζουν την πλήρη κατανομή όταν απαιτείται μεγαλύτερη λεπτομέρεια.

Country	Total Bookings	Back
Η.Π.Α.	873	
ΕΛΛΑΔΑ	434	
ΑΓΓΛΙΑ	380	
ΧΩΡΕΣ ΠΡΩΗΝ ΕΣΣΔ	362	
ΓΑΛΛΙΑ	315	
ΙΤΑΛΙΑ	199	
ΑΥΣΤΡΑΛΙΑ-ΝΕΑ ΖΗΛΑΝΔΙΑ	189	
ΓΕΡΜΑΝΙΑ	170	
ΙΣΡΑΗΛ	128	
.....	---	
Total	4063	

Country	Total Revenue	Back
Η.Π.Α.	1,089,111.29 €	
ΑΓΓΛΙΑ	463,868.06 €	
ΧΩΡΕΣ ΠΡΩΗΝ ΕΣΣΔ	362,886.11 €	
ΓΑΛΛΙΑ	332,156.14 €	
ΕΛΛΑΔΑ	288,014.05 €	
ΑΥΣΤΡΑΛΙΑ-ΝΕΑ ΖΗΛΑΝΔΙΑ	243,435.98 €	
ΙΤΑΛΙΑ	231,213.47 €	
ΓΕΡΜΑΝΙΑ	210,469.33 €	
ΙΣΡΑΗΛ	144,649.93 €	
ΚΑΝΑΔΑΣ	132,326.50 €	
Total	4,477,272.75 €	

Σχήμα 5.12: Αναλυτικότερη προβολή της γεωγραφικής κατανομής στη σελίδα Bookings Analysis.

Η προβολή της λίστας των χωρών είναι χρήσιμη επειδή επιτρέπει να αξιολογηθεί η διαφοροποίηση των πελατών. Ένα dashboard που εμφανίζει μόνο τις κορυφαίες χώρες είναι επαρκές για γρήγορη επισκόπηση, αλλά η πλήρης λίστα είναι απαραίτητη όταν ο στόχος είναι να διαπιστωθεί αν το ξενοδοχείο εξαρτάται από λίγες αγορές ή αν διατηρεί ευρύτερη διασπορά.

5.6.2 Room Performance

Η σελίδα *Room Performance* μεταφέρει την ανάλυση στο επίπεδο του τύπου δωματίου. Τα βασικά visuals συγκρίνουν τον αριθμό κρατήσεων, το revenue και το ADR ανά room_type. Με αυτόν τον τρόπο δεν αξιολογείται ένας τύπος δωματίου μόνο από το συνολικό έσοδο, αλλά από τον συνδυασμό όγκου και τιμής. Ένας τύπος δωματίου μπορεί να εμφανίζει υψηλή συνεισφορά επειδή πωλείται συχνότερα, ενώ ένας άλλος μπορεί να έχει μικρότερο όγκο αλλά υψηλότερο ADR.

Ο δείκτης *ADR (Average Daily Rate)* εκφράζει τη μέση ημερήσια τιμή ανά κατειλημένο δωμάτιο και χρησιμοποιείται για τη σύγκριση της τιμολογιακής απόδοσης μεταξύ των διαφορετικών τύπων δωματίου.



Σχήμα 5.13: Σελίδα Room Performance του Power BI με σύγκριση κρατήσεων, εσόδων και ADR ανά τύπο δωματίου.

Το scatter chart *ADR vs Bookings by Room Type* είναι το πιο χρήσιμο γράφημα για τη συνδυαστική ανάγνωση της σελίδας. Στον έναν άξονα αποτυπώνεται η μέση τιμή και στον άλλο ο όγκος κρατήσεων, επομένως οι τύποι δωματίων τοποθετούνται σε διαφορετικές θέσεις ανάλογα με τον τρόπο που αποδίδουν. Η απεικόνιση βοηθά να εντοπιστούν δωμάτια με υψηλή ζήτηση αλλά χαμηλότερη τιμή, δωμάτια με υψηλή τιμή αλλά περιορισμένο όγκο και κατηγορίες που συνδυάζουν και τα δύο.

room_type_short_name	Total Bookings	Bookings %	Total Revenue	Revenue %	ADR	Revenue per Booking	Back to Charts
Deluxe Suite Outdoor HT	1194	29,39%	1.510.768,04 €	33,74%	387,58 €	1.265,80 €	
Junior Suite Caldera	1075	26,46%	944.865,39 €	21,10%	299,10 €	878,94 €	
Honeymoon Suite Outdoor HT	578	14,23%	810.525,02 €	18,10%	464,48 €	1.402,29 €	
Deluxe Suite Indoor HT	462	11,37%	528.009,56 €	11,79%	356,76 €	1.142,88 €	
Superior Suite Outdoor HT	479	11,79%	488.805,40 €	10,92%	324,79 €	1.020,47 €	
Standard Suite Indoor HT	275	6,77%	194.299,34 €	4,34%	266,89 €	706,54 €	
Total	4063	100,00%	4.477.272,75 €	100,00%	357,75 €	1.101,96 €	

Σχήμα 5.14: Αναλυτικός πίνακας δεικτών στη σελίδα Room Performance.

Το *View Full Table* παρουσιάζει τις ακριβείς τιμές των Total Bookings, Bookings %, Total Revenue, Revenue %, ADR και Revenue per Booking, ενώ το *Back to Charts* επιστρέφει στη συγκριτική οπτική παρουσίαση. Ο πίνακας αυτός είναι χρήσιμος όταν απαιτείται ακριβής σύγκριση μεταξύ δωματίων με παρόμοια εικόνα στα charts ή όταν ο χρήστης επιθυμεί να εξηγήσει ποσοτικά ένα συμπέρασμα που αρχικά προέκυψε οπτικά.

5.6.3 Stay Length Analysis

Η σελίδα *Stay Length Analysis* εξετάζει δύο συμπεριφορές που δεν είναι άμεσα ορατές από τα οικονομικά μεγέθη: πόσο διαρκεί η διαμονή και πόσο νωρίς πραγματοποιείται η κράτηση. Η κατανομή του length of stay δείχνει πού συγκεντρώνεται το μεγαλύτερο μέρος των κρατήσεων, ενώ ο πίνακας *Length of Stay & Booking Lead Time* συνδυάζει τη διάρκεια με το booking window και επιτρέπει να εξεταστεί αν οι μεγαλύτερες διαμονές κλείνονται νωρίτερα ή αν διαφορετικοί χρόνοι διαμονής ακολουθούν διαφορετική συμπεριφορά.



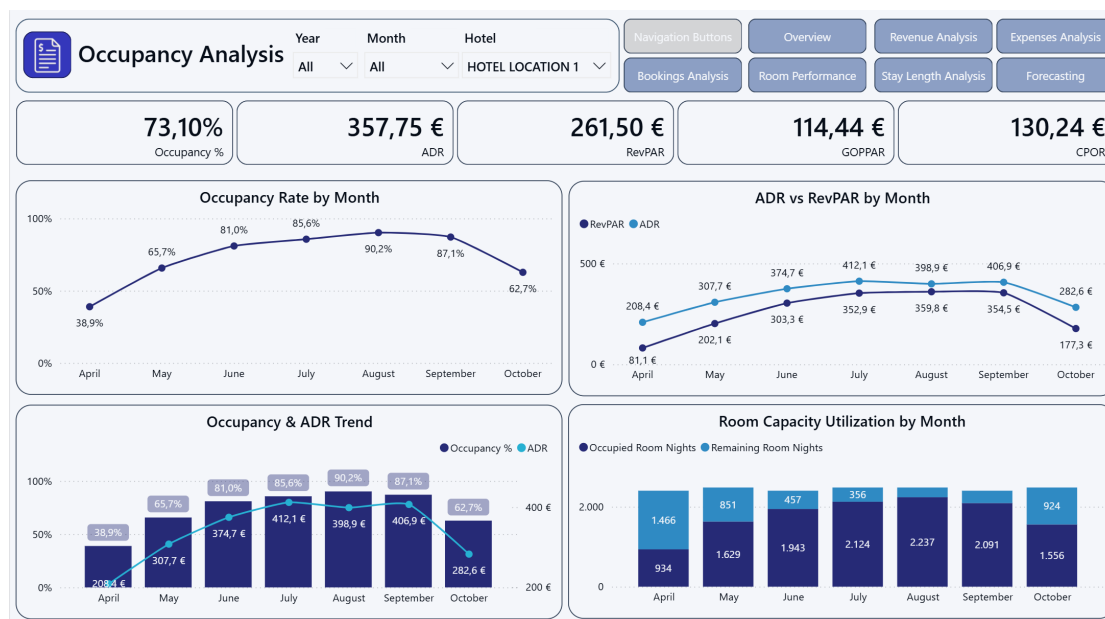
Σχήμα 5.15: Σελίδα Stay Length Analysis του Power BI με ανάλυση διάρκειας διαμονής και booking lead time.

Τα buttons *Monthly Trend* και *Back to Distribution* επιτρέπουν στον χρήστη να εξετάζει τη διάρκεια διαμονής με δύο διαφορετικούς τρόπους. Η κατανομή δείχνει ποια διάρκεια εμφανίζεται συχνότερα στις κρατήσεις, ενώ το μηνιαίο γράφημα παρουσιάζει πώς μεταβάλλεται η συμπεριφορά αυτή μέσα στην σεζόν. Με αυτόν τον τρόπο είναι δυνατό να εντοπιστούν περίοδοι στις οποίες οι επισκέπτες πραγματοποιούν μικρότερες ή μεγαλύτερες διαμονές και να εξεταστεί αν υπάρχει κάποιο εποχικό μοτίβο.

5.6.4 Occupancy Analysis

Η σελίδα *Occupancy Analysis* συνδέει τη ζήτηση με τη διαθέσιμη δυναμικότητα. Τα cards περιλαμβάνουν δείκτες όπως Occupancy, ADR, RevPAR, GOPPAR και CPOR, όμως η κύρια πληροφορία προέρχεται από τον τρόπο με τον οποίο αυτοί οι δείκτες εξελίσσονται και συνδυάζονται στα γραφήματα.

Το *Occupancy* εκφράζει το ποσοστό πληρότητας της διαθέσιμης δυναμικότητας. Ο δείκτης *RevPAR* (*Revenue per Available Room*) εκφράζει το έσοδο ανά διαθέσιμο δωμάτιο, ενώ ο *GOPPAR* (*Gross Operating Profit per Available Room*) εκφράζει το μικτό λειτουργικό κέρδος ανά διαθέσιμο δωμάτιο. Οι δείκτες *ADR* (*Average Daily Rate*) και *CPOR* (*Cost per Occupied Room*), οι οποίοι παρουσιάστηκαν προηγουμένως, αποτυπώνουν αντίστοιχα τη μέση ημερήσια τιμή και το κόστος ανά κατειλημμένο δωμάτιο.

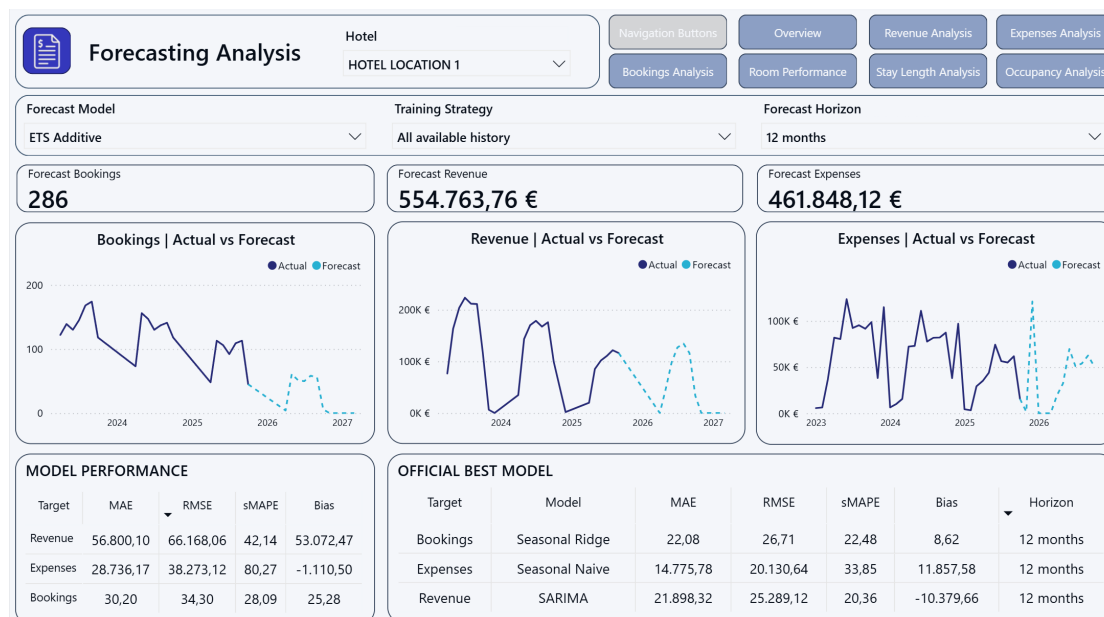


Σχήμα 5.16: Σελίδα Occupancy Analysis του Power BI με πληρότητα, ADR, RevPAR, GOPPAR, CPOR και αξιοποίηση της διαθέσιμης δυναμικότητας.

Το *Occupancy Rate by Month* δείχνει πώς μεταβάλλεται η πληρότητα του ξενοδοχείου ανά μήνα και σε ποιες περιόδους χρησιμοποιείται περισσότερο η διαθέσιμη δυναμικότητα. Από το γράφημα μπορεί να εντοπιστεί η εποχικότητα της ζήτησης και οι μήνες στους οποίους η μονάδα πλησιάζει περισσότερο στη μέγιστη δυναμικότητά της. Το *ADR vs RevPAR by Month* συγκρίνει τη μέση τιμή δωματίου με το έσοδο ανά διαθέσιμο δωμάτιο. Η σύγκριση δείχνει ότι η υψηλή τιμή από μόνη της δεν αρκεί για υψηλή απόδοση, καθώς σημαντικό ρόλο έχει και το ποσοστό των δωματίων που τελικά πωλούνται. Το *Occupancy & ADR Trend* παρουσιάζει μαζί την πληρότητα και το ADR κατά τη διάρκεια της σεζόν. Από την κοινή τους πορεία μπορεί να εξεταστεί αν οι περίοδοι αυξημένης ζήτησης συνοδεύονται και από υψηλότερες τιμές, δίνοντας καλύτερη εικόνα για τη σχέση μεταξύ ζήτησης και τιμολογιακής πολιτικής. Τέλος, το *Room Capacity Utilization by Month* παρουσιάζει τα κατειλημμένα και τα διαθέσιμα room-nights ανά μήνα. Το γράφημα κάνει πιο εύκολα αντιληπτό πόσο μέρος της συνολικής δυναμικότητας αξιοποιήθηκε και σε ποιες περιόδους υπήρχε μεγαλύτερο περιθώριο για επιπλέον κρατήσεις.

5.7 Forecasting Analysis και επιλογή μοντέλου

Η σελίδα *Forecasting Analysis* αποτελεί το σημείο στο οποίο η ιστορική ανάλυση επεκτείνεται στη μελλοντική περίοδο. Ο σχεδιασμός της δεν έχει ως στόχο να εμφανίσει μία μόνο "τελική πρόβλεψη", αλλά να επιτρέπει στον χρήστη να εξετάζει διαφορετικές μεθόδους πρόβλεψης, με διαφορετικά training strategies και να γνωρίζει για τα επιλεγμένα φίλτρα ποιο μοντέλο ήταν το καλύτερο.



Σχήμα 5.17: Σελίδα Forecasting Analysis του Power BI με επιλογή ξενοδοχείου, μοντέλου, training strategy και χρονικού ορίζοντα, καθώς και σύγκριση actual και forecast τιμών.

Ο πρώτος βασικός έλεγχος είναι ο slicer *Forecast Model*. Περιλαμβάνει ETS Additive, LSTM, SARIMA, Seasonal Naive, Seasonal Ridge και Weighted Seasonal Mean. Με την αλλαγή του μοντέλου ενημερώνονται οι προβλέψεις του επιλεγμένου μοντέλου και οι αντίστοιχες μετρικές αξιολόγησης. Ο δεύτερος έλεγχος είναι ο *Training Strategy*. Η *All History* χρησιμοποιεί όλο το διαθέσιμο ιστορικό πριν από κάθε πρόβλεψη, ενώ η *Last 30 Months* περιορίζει την εκπαίδευση στις 30 πιο πρόσφατες μηνιαίες παρατηρήσεις. Η σύγκριση των δύο στρατηγικών είναι σημαντική: αν η *All History* υπερέχει, το παλαιότερο ιστορικό εξακολουθεί να προσθέτει χρήσιμη εποχική πληροφορία· αν η *Last 30 Months* παρουσιάζει την καλύτερη επίδοση, αυτό αποτελεί ένδειξη ότι η πιο πρόσφατη συμπεριφορά της σειράς είναι περισσότερο αντιπροσωπευτική από τις παλαιότερες περιόδους. Ο τρίτος έλεγχος είναι ο *Forecast Horizon*, με επιλογές 1, 3, 6 και 12 μηνών. Με αυτόν τον τρόπο μπορεί να εξεταστεί αν ένα μοντέλο παραμένει αξιόπιστο μόνο σε βραχυπρόθεσμη πρόβλεψη ή διατηρεί την απόδοσή του και σε μεγαλύτερο ορίζοντα.

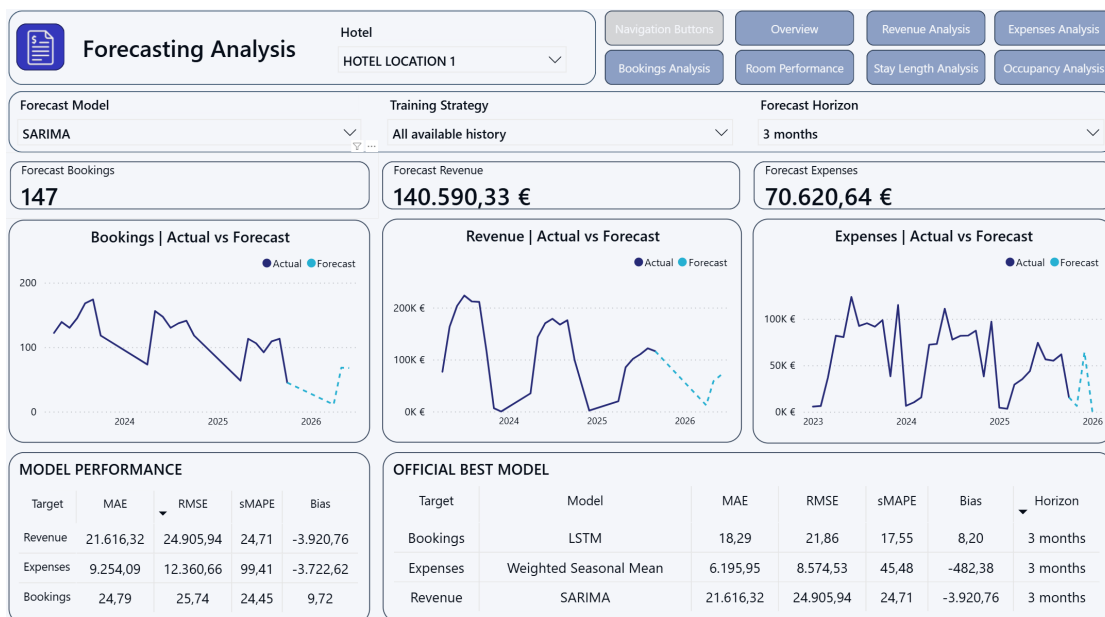
Τα τρία forecast cards για Bookings, Revenue και Expenses λειτουργούν ως συνοπτική

ένδειξη του επιλεγμένου μοντέλου και του χρονικού ορίζοντα. Η ουσιαστική ανάγνωση όμως πραγματοποιείται στα τρία line charts *Actual vs Forecast*, ένα για κάθε target: ενεργές κρατήσεις, included καθαρά έσοδα και included έξοδα. Η ιστορική σειρά εμφανίζεται μέχρι τον τελευταίο πραγματικά διαθέσιμο μήνα και η forecast σειρά συνεχίζει από το επόμενο χρονικό σημείο. Έτσι ο χρήστης μπορεί να εξετάσει αν το μοντέλο συνεχίζει την γνωστή εποχικότητα, αν προβλέπει υψηλότερο ή χαμηλότερο επίπεδο από τα πρόσφατα έτη και αν η ένταση των κορυφών και των χαμηλών περιόδων παραμένει ρεαλιστική σε σχέση με το ιστορικό.

Ο πίνακας *MODEL PERFORMANCE* παρουσιάζει τις MAE, RMSE, sMAPE και Bias για την μοντέλο που έχει επιλέξει ο χρήστης. Οι τέσσερις μετρικές έχουν συμπληρωματικούς ρόλους. Το MAE εκφράζει τη μέση απόλυτη απόκλιση και είναι εύκολο να ερμηνευθεί στις ίδιες μονάδες με το target. Το RMSE δίνει μεγαλύτερη βαρύτητα σε μεγάλες αστοχίες, επομένως η διαφορά του από το MAE αποτελεί ένδειξη ότι υπάρχουν μεγαλύτερα forecast errors. Το sMAPE εκφράζει το σφάλμα σε σχετική ποσοστιαία κλίμακα και επιτρέπει ευκολότερη σύγκριση μεταξύ των διαφορετικών μεγεθών. Τέλος, το Bias δείχνει την κατεύθυνση του σφάλματος: θετική τιμή υποδηλώνει τάση υπερεκτίμησης και αρνητική τιμή τάση υποεκτίμησης.

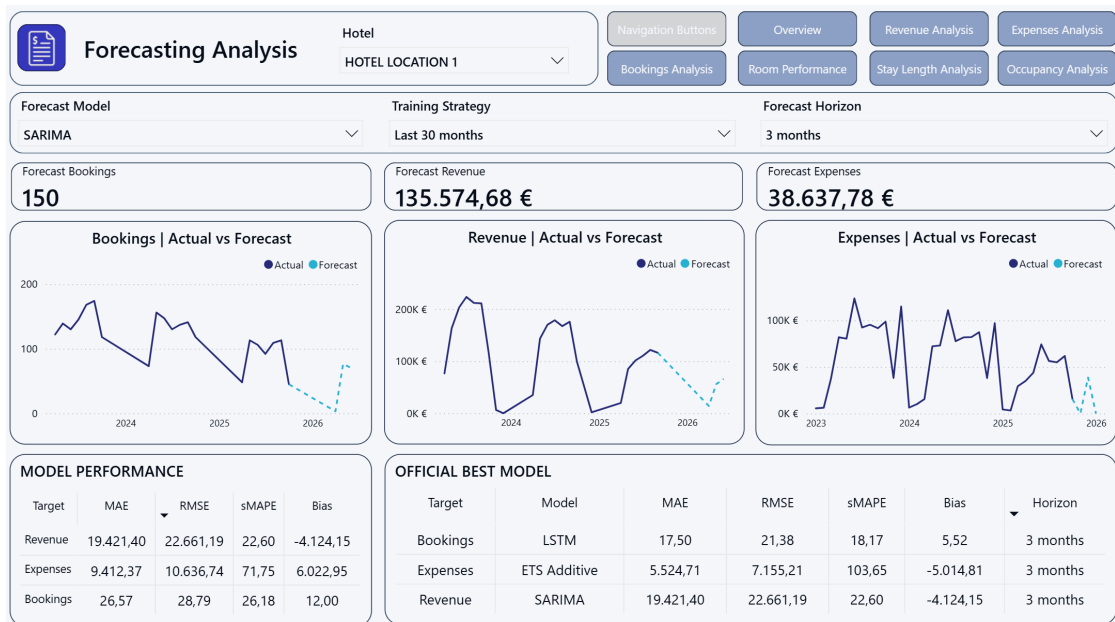
Ο πίνακας *OFFICIAL BEST MODEL* έχει διαφορετικό ρόλο από τον *MODEL PERFORMANCE*. Ο πρώτος ακολουθεί τους slicers και επιτρέπει διερεύνηση οποιουδήποτε μοντέλου και οποιασδήποτε από τις δύο training strategies. Ο δεύτερος εμφανίζει την επίσημη επιλογή που έχει αποθηκευτεί μετά το backtesting και επομένως διατηρεί ορατό το αποτέλεσμα της διαδικασίας επιλογής. Η διάκριση αυτή είναι βασική: το dashboard επιτρέπει πειραματική σύγκριση, αλλά ταυτόχρονα εμφανίζει ποιο μοντέλο κρίθηκε καταλληλότερο από το σύστημα.

Για να γίνει πιο σαφής η συμπεριφορά των μοντέλων, ακολουθούν επιπλέον παραδείγματα με διαφορετικές στρατηγικές και forecast horizons.



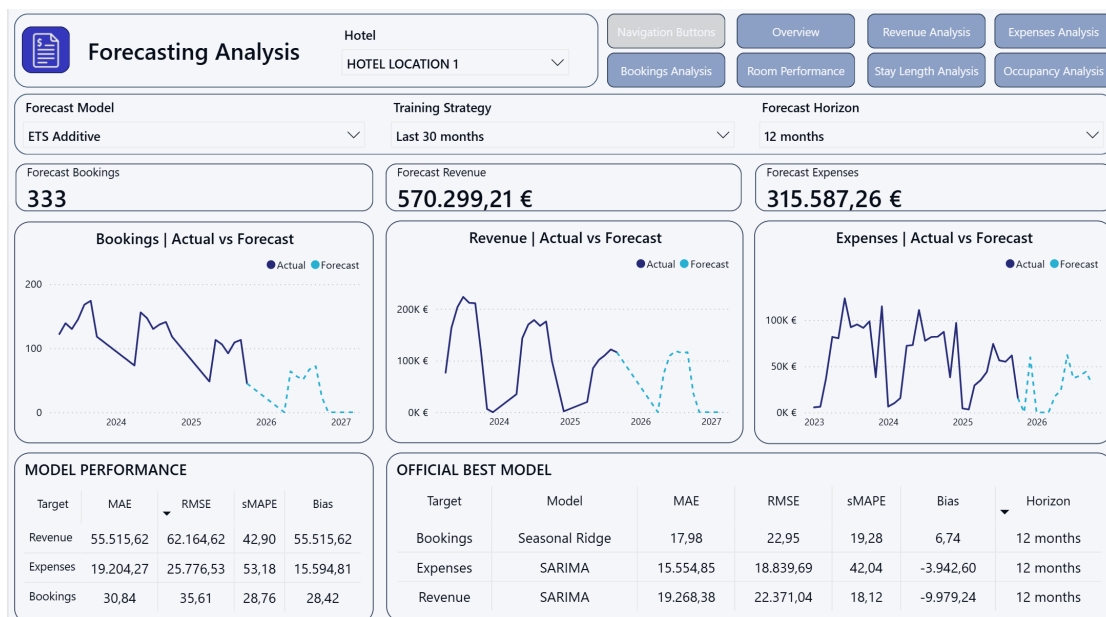
Σχήμα 5.18: Forecasting Analysis για horizon 3 μηνών με strategy *All available history*.

Στο Σχήμα 5.18 παρατηρείται ότι για horizon τριών μηνών η *All History* τείνει να διατηρεί εποχική δομή των σειρών. Στα revenue forecasts το SARIMA παραμένει ιδιαίτερα σταθερό και δίνει καμπύλες που ακολουθούν σχετικά ομαλά το ιστορικό μοτίβο. Στις bookings, η επίσημη επιλογή παραμένει LSTM, κάτι που υποδηλώνει ότι στις βραχυπρόθεσμες προβλέψεις η πρόσφατη δυναμική και οι μη γραμμικότητες της σειράς μπορούν να αποτυπωθούν ικανοποιητικά από πιο ευέλικτο μοντέλο. Στα expenses, αντίθετα, εμφανίζεται διαφορετική επιλογή μοντέλου, γεγονός που δείχνει ότι η συγκεκριμένη χρονοσειρά είναι πιο ασταθής και πιο ευαίσθητη στον τρόπο εκπαίδευσης.



Σχήμα 5.19: Forecasting Analysis για horizon 3 μηνών με strategy *Last 30 months*.

Στο Σχήμα 5.19, όπου η εκπαίδευση περιορίζεται στους τελευταίους 30 μήνες, η μορφή των προβλέψεων παραμένει κοντά στο πρόσφατο ιστορικό. Η διαφορά είναι μικρότερη στα bookings και στα revenue, όπου η βασική εποχική συμπεριφορά παραμένει ορατή ανεξάρτητα από τη στρατηγική. Ωστόσο, στα expenses η αλλαγή strategy επηρεάζει περισσότερο το αποτέλεσμα, επειδή οι σειρές εξόδων περιλαμβάνουν συχνότερα απότομες μεταβολές και λιγότερο σταθερή εποχικότητα. Το εύρημα αυτό είναι ουσιαστικό: η *Last 30 Months* δεν βελτιώνει απαραίτητα όλες τις σειρές, αλλά είναι χρήσιμη όταν θέλουμε το μοντέλο να προσανατολιστεί περισσότερο στην πρόσφατη συμπεριφορά.



Σχήμα 5.20: Forecasting Analysis για horizon 12 μηνών με strategy *Last 30 months*.

Στον ορίζοντα των 12 μηνών, όπως φαίνεται στο Σχήμα 5.20, η δυσκολία της πρόβλεψης αυξάνεται αναμενόμενα. Οι forecast γραμμές επεκτείνονται πλέον σε ολόκληρη μελλοντική περίοδο και επομένως δοκιμάζεται περισσότερο η ικανότητα του μοντέλου να διατηρήσει την εποχικότητα. Σε αυτό το πλαίσιο, το revenue συνεχίζει να εμφανίζει πιο σταθερή συμπεριφορά και το SARIMA παραμένει ισχυρή επιλογή, ενώ στα bookings εμφανίζονται συχνότερα μοντέλα όπως το Seasonal Ridge για μεγαλύτερους horizons. Στα expenses, η εικόνα είναι πιο ευμετάβλητη, επειδή το επίπεδο και οι κορυφές της σειράς επηρεάζονται περισσότερο από ειδικές λειτουργικές συνθήκες.

Συνολικά, από τη ανάγνωση των forecasting σελίδων προκύπτουν τρία βασικά συμπεράσματα. Πρώτον, δεν υπάρχει ένα μοντέλο που να υπερέχει σε όλα τα targets. Το revenue παρουσιάζει πιο σταθερή και "κανονική" εποχικότητα, γι' αυτό και το SARIMA εμφανίζεται επανειλημμένα ως επίσημη επιλογή. Δεύτερον, τα bookings ακολουθούν πιο σύνθετη συμπεριφορά: στους μικρούς horizons αποδίδουν καλά πιο ευέλικτα μοντέλα όπως το LSTM, ενώ στους μεγαλύτερους horizons εμφανίζονται συχνότερα λύσεις όπως το Seasonal Ridge, που αποτυπώνουν ικανοποιητικά τη γενικότερη εποχική δομή χωρίς να προσπαθούν να αναπαράγουν κάθε διακύμανση. Τρίτον, τα expenses είναι η πιο ευαίσθητη χρονοσειρά τόσο ως προς την επιλογή μοντέλου όσο και ως προς την επιλογή strategy. Η παρατήρηση αυτή είναι λογική, επειδή τα έξοδα δεν καθορίζονται μόνο από εποχικότητα αλλά και από λειτουργικές αποφάσεις και έκτακτες δαπάνες.

Η σύγκριση *All History* και *Last 30 Months* προσθέτει ένα ακόμη επίπεδο ερμηνείας. Όταν οι δύο στρατηγικές οδηγούν σε πολύ παρόμοια συμπεριφορά, αυτό σημαίνει ότι η σειρά έχει αρκετά σταθερό εποχικό μοτίβο και ότι το παλαιότερο ιστορικό δεν αλλοιώνει

ουσιαστικά την πρόβλεψη. Όταν όμως εμφανίζονται μεγαλύτερες διαφορές, κυρίως στα expenses, αυτό αποτελεί ένδειξη ότι η πρόσφατη περίοδος έχει διαφορετική πληροφοριακή αξία από το μακρύτερο ιστορικό. Επομένως, οι δύο στρατηγικές αντιμετωπίζονται ως ουσιαστικό μέρος της διαδικασίας αξιολόγησης.

5.8 Συμπεράσματα από την ανάλυση των dashboards

Η συνολική εικόνα των σελίδων δείχνει ότι το dashboard δεν έχει σχεδιαστεί ως σειρά ανεξάρτητων γραφημάτων, αλλά ως διαδοχική διαδικασία διερεύνησης. Η σελίδα Overview δίνει το γενικό επίπεδο λειτουργίας, οι σελίδες Revenue και Expenses εξηγούν από πού προέρχονται τα οικονομικά αποτελέσματα, οι Bookings, Room Performance και Stay Length περιγράφουν τη συμπεριφορά της ζήτησης και των κρατήσεων, ενώ η Occupancy συνδέει τη ζήτηση με τη διαθέσιμη δυναμικότητα. Τα φίλτρα και τα bookmarks επιτρέπουν η ίδια λογική να επαναλαμβάνεται για διαφορετικά έτη, μήνες και ξενοδοχεία χωρίς να δημιουργούνται νέα reports.

Από την οικονομική ανάλυση προκύπτει έντονη εποχικότητα, σημαντική συμβολή του direct channel και δυνατή συμμετοχή των OTA, ενώ η λεπτομερέστερη ανάλυση του revenue επιτρέπει να φανεί και ο βαθμός συγκέντρωσης μέσα σε κάθε κανάλι. Στα έξοδα, η ανάλυση δείχνει ότι δεν αρκεί η παρακολούθηση του συνολικού ποσού, αλλά απαιτείται διαχωρισμός των κατηγοριών που το διαμορφώνουν. Στις κρατήσεις και στα δωμάτια γίνεται επίσης εμφανές ότι η απόδοση δεν εξαρτάται αποκλειστικά από τον όγκο: η σύγκριση ADR, bookings και revenue επιτρέπει να εξεταστούν οι αποδόσεις ανά room type.

Η ανάλυση πληρότητας δείχνει επιπλέον ότι οι δείκτες τιμής και ζήτησης πρέπει να εξετάζονται μαζί. Το ADR περιγράφει την τιμή των πωλημένων δωματίων, ενώ το RevPAR και η πληρότητα ενσωματώνουν την συνολική διαθέσιμη δυναμικότητας. Επομένως, η αξιολόγηση της απόδοσης μόνο με την τιμή μπορεί να οδηγήσει σε ελλιπή συμπεράσματα. Η παράλληλη παρουσία Occupied και Remaining Room Nights κάνει ορατό και το μέρος της δυναμικότητας που δεν αξιοποιήθηκε.

Τέλος, το forecasting δείχνει ότι δεν υπάρχει ένα μοντέλο ή μία στρατηγική που να θεωρείται εκ των προτέρων καλύτερη για όλες τις σειρές. Η διαφοροποίηση των επίσημων επιλογών μεταξύ κρατήσεων, εσόδων και εξόδων υποστηρίζει την επιλογή ανά target, ενώ η σύγκριση *All History* και *Last 30 Months* επιτρέπει να ελεγχθεί αν η μακροχρόνια εποχικότητα ή η πρόσφατη συμπεριφορά είναι περισσότερο χρήσιμη για κάθε περίπτωση. Οι MAE, RMSE, sMAPE και Bias παρέχουν διαφορετικές συγκρίσεις της ποιότητας της πρόβλεψης και πρέπει να εξετάζονται μαζί με την πρόβλεψη. Συνεπώς, το Power BI λειτουργεί όχι μόνο ως μέσο παρουσίασης των αποτελεσμάτων αλλά και ως εργαλείο ελέγχου και ερμηνείας ολόκληρης της διαδικασίας.

Κεφάλαιο 6

Συμπεράσματα και Μελλοντικές Επεκτάσεις

6.1 Συνολική αποτίμηση της εργασίας

Η παρούσα εργασία είχε ως βασικό στόχο να εξετάσει κατά πόσο τα επιχειρησιακά και οικονομικά δεδομένα μιας ξενοδοχειακής μονάδας μπορούν να μετατραπούν σε ένα ενιαίο και αξιοποιήσιμο πληροφοριακό σύστημα. Το πρόβλημα δεν περιοριζόταν στην παραγωγή ορισμένων γραφημάτων ή στην εφαρμογή ενός μοντέλου πρόβλεψης πάνω σε ήδη καθαρά δεδομένα. Αντίθετα, το μεγαλύτερο μέρος της δυσκολίας βρισκόταν σε προηγούμενα στάδια: στην ανάγνωση αρχείων διαφορετικής μορφής, στην αναγνώριση των πραγματικών επικεφαλίδων, στην αντιστοίχιση διαφορετικών ονομασιών στην ίδια κατηγορία, στον διαχωρισμό των λογιστικών κινήσεων και τελικά στην δημιουργία ενός σταθερού αναλυτικού σχήματος το οποίο θα μπορούσε να χρησιμοποιηθεί τόσο για περιγραφική ανάλυση όσο και για πρόβλεψη.

Η τελική εφαρμογή καλύπτει αυτή τη διαδρομή από το αρχικό αρχείο έως το αναλυτικό αποτέλεσμα. Τα δεδομένα κρατήσεων, εσόδων και εξόδων εισάγονται από κοινή διαδικτυακή διεπαφή, αποθηκεύονται σε ελεγχόμενη δομή, επεξεργάζονται από διαφορετικούς μηχανισμούς καθαρισμού και συνδέονται με την αντίστοιχη ξενοδοχειακή μονάδα. Η επεξεργασία δεν δημιουργεί μόνο ένα νέο αρχείο εξόδου αλλά αποθηκεύει την αρχική και την καθαρισμένη μορφή τους και διατηρεί το αναγνωριστικό του αρχείου από το οποίο προήλθαν. Πάνω σε αυτό το επίπεδο δημιουργούνται SQL views, τα οποία μετατρέπουν τις JSONB δομές σε τυποποιημένες στήλες κατάλληλες για την αναλυτική χρήση.

Στα οικονομικά δεδομένα, η συμβολή της επεξεργασίας είναι ιδιαίτερα σημαντική. Στα έσοδα η λογιστική γραμμή μετατρέπεται σε επιχειρησιακή εγγραφή παραστατικού, ενώ διαχωρίζονται η καθαρή αξία, ο ΦΠΑ, ο συνολικός τζίρος και οι ειδικές λογιστικές κινήσεις. Παράλληλα, μέσω mappings και γενικών κανόνων δημιουργείται μια σταθερή

κατηγορία πελάτη, ώστε οι διαφορετικές ονομασίες μιας πηγής να μην μεταφέρονται αυτούσιες στο επίπεδο αναφοράς. Στα έξοδα, ο λογιστικός κωδικός, η περιγραφή και η αιτιολογία χρησιμοποιούνται συνδυαστικά για τον χαρακτηρισμό της γραμμής, τη δημιουργία κατηγορίας και υποκατηγορίας και την απόφαση αν η εγγραφή πρέπει να συμμετέχει στα έξοδα. Η διάκριση μεταξύ συνολικών και included ποσών επιτρέπει η αναλυτική εικόνα να μην ταυτίζεται αποκλειστικά με το άθροισμα όλων των λογιστικών κινήσεων.

Το forecasting αποτέλεσε το επόμενο επίπεδο της ροής στην εφαρμογή μας. Οι χρονοσειρές δημιουργούνται πάνω στα καθαρισμένα δεδομένα και οι βασικές μεταβλητές πρόβλεψης είναι οι ενεργές κρατήσεις, τα included καθαρά έσοδα και τα included έξοδα. Η τρέχουσα υλοποίηση χρησιμοποιεί κοινή μηνιαία χρονοσειρά και συγκρίνει διαφορετικές οικογένειες μοντέλων, όπως Seasonal Naive, σταθμισμένο εποχικό μέσο, ETS, SARIMA, Ridge και LSTM. Παράλληλα, για την εκπαίδευση και αξιολόγηση των μοντέλων χρησιμοποιούνται δύο διαφορετικές στρατηγικές ιστορικού: η AllHistory, στην οποία αξιοποιείται όλο το διαθέσιμο ιστορικό μέχρι το αντίστοιχο forecast origin, και η Last30Months, στην οποία η εκπαίδευση περιορίζεται στους τελευταίους 30 διαθέσιμους μήνες πριν από κάθε χρονικό σημείο πρόβλεψης. Με τον τρόπο αυτό εξετάζεται τόσο η περίπτωση στην οποία η συνολική ιστορική πληροφορία παραμένει χρήσιμη όσο και η περίπτωση στην οποία οι πιο πρόσφατες παρατηρήσεις περιγράφουν καλύτερα την τρέχουσα συμπεριφορά της χρονοσειράς. Η επιλογή πραγματοποιείται με μια κοινή διαδικασία backtesting και μετρικές σφάλματος, ώστε η τελική πρόβλεψη να μην στηρίζεται σε ένα μοντέλο ή μία στρατηγική εκπαίδευσης που έχει επιλεγεί εκ των προτέρων ανεξάρτητα από τη συμπεριφορά των δεδομένων.

Συνολικά, η εργασία κατέληξε σε ένα σύστημα στο οποίο η συλλογή, ο καθαρισμός, η επιχειρησιακή ερμηνεία, η αποθήκευση, η συγκέντρωση, η πρόβλεψη και η οπτικοποίηση αποτελούν διαδοχικά τμήματα της ίδιας ροής. Το τελικό αποτέλεσμα δεν είναι μόνο η αυτοματοποίηση των ενεργειών, αλλά η δημιουργία μιας κοινής λογικής με την οποία τα δεδομένα μπορούν να επανεπεξεργαστούν, να ελεγχθούν και να αξιοποιηθούν ξανά όταν προστεθούν νέα αρχεία.

6.2 Σημασία της ολοκληρωμένης ανάλυσης για τη λήψη αποφάσεων

Η σημασία της προσέγγισης γίνεται περισσότερο εμφανής αν εξεταστεί η διαφορά μεταξύ της απλής αποθήκευσης δεδομένων και της δημιουργίας πληροφορίας που μπορεί να υποστηρίξει την λήψη αποφάσεων. Ένα λογιστικό αρχείο μπορεί να περιέχει μεγάλο αριθμό εγγραφών χωρίς να παρέχει από μόνο του σαφή εικόνα για την επιχείρηση. Για να μετατραπεί σε χρήσιμη πληροφορία απαιτείται να είναι γνωστό ποια εγγραφή αντιπροσωπεύει πώληση, ποια είναι λογιστική τακτοποίηση, σε ποια κατηγορία ανήκει μία δαπάνη,

ποιο κανάλι δημιουργεί τον τζίρο και ποια χρονική περίοδο αφορά μία κράτηση.

Η ανάλυση που αναπτύχθηκε επιτρέπει την κοινή παρακολούθηση διαφορετικών πλευρών της λειτουργίας. Οι κρατήσεις παρέχουν εικόνα της ζήτησης και της μελλοντικής δραστηριότητας, τα έσοδα αποτυπώνουν την οικονομική απόδοση της επιχείρησης, ενώ τα έξοδα επιτρέπουν να εξεταστεί η δομή του συνολικού και του λειτουργικού κόστους. Η σύνδεση αυτών των κατηγοριών δημιουργεί καλύτερη εικόνα από μία μεμονωμένη ανάλυση των λογιστικών αρχείων.

Ιδιαίτερη αξία έχει η δυνατότητα να εξετάζονται παράλληλα τα συνολικά και τα included ποσά. Η διαφορά τους δεν αποτελεί απλώς λογιστική απόκλιση, αλλά πληροφορία η οποία μπορεί να αναλυθεί μέσω των εγγραφών που έχουν εξαιρεθεί και των λόγων εξαίρεσής τους. Έτσι η εφαρμογή επιτρέπει να απαντηθούν ερωτήματα όπως ποιο μέρος των κινήσεων δεν θεωρήθηκε πραγματικός τζίρος, ποιες κατηγορίες εξόδων επιβαρύνουν περισσότερο τη λειτουργία και ποιοι κανόνες οδήγησαν στην τελική ταξινόμηση.

Εξίσου σημαντική είναι η χρονική διάσταση της ανάλυσης. Όταν τα καθαρισμένα δεδομένα οργανώνονται ανά μήνα, μπορούν να συγκριθούν διαφορετικές περιόδους λειτουργίας, να εντοπιστούν εποχικά πρότυπα και να εξεταστεί η σχέση μεταξύ κρατήσεων, εσόδων και εξόδων. Το forecasting προσθέτει σε αυτή την εικόνα μια μελλοντική διάσταση. Η πρόβλεψη δεν αντικαθιστά την υπάρχουσα επιχειρησιακή ανάλυση, αλλά μπορεί να λειτουργήσει ως πρόσθετο σημείο αναφοράς για προγραμματισμό προσωπικού, προμήθειες, διαχείριση αναγκών, στόχους εσόδων και έλεγχο αποκλίσεων.

Η ύπαρξη ιχνηλασιμότητας ενισχύει επίσης τη χρησιμότητα της ανάλυσης. Ένας συγκεντρωτικός δείκτης μπορεί να οδηγήσει πίσω στις εγγραφές που τον δημιούργησαν και, όπου χρειάζεται, στις αρχικές γραμμές του αρχείου. Αυτό είναι ιδιαίτερα σημαντικό σε οικονομικά δεδομένα, όπου η αξιοπιστία ενός αποτελέσματος δεν εξαρτάται μόνο από το τελικό άθροισμα αλλά και από τη δυνατότητα να εξηγηθεί η προέλευσή του. Η εφαρμογή επομένως δεν λειτουργεί μόνο ως ένα μέσο παραγωγής δεικτών, αλλά και ως μηχανισμός ελέγχου της διαδρομής από την αρχική εγγραφή έως το τελικό KPI.

6.3 Περιορισμοί της παρούσας υλοποίησης

Παρά τα παραπάνω αποτελέσματα, η παρούσα υλοποίηση διαθέτει περιορισμούς οι οποίοι πρέπει να ληφθούν υπόψη στην ερμηνεία των αποτελεσμάτων και αποτελούν αφετηρία για μελλοντική ανάπτυξη. Η αρχιτεκτονική του συστήματος και οι βασικοί μηχανισμοί καθαρισμού έχουν σχεδιαστεί ώστε να λειτουργούν γενικά και να μην εξαρτώνται από μία συγκεκριμένη ονομασία εταιρείας, μία σταθερή θέση στήλης ή ένα μόνο αρχείο. Ωστόσο, η εισαγωγή δεδομένων από μία νέα πηγή ή από διαφορετική ξενοδοχειακή μονάδα πρέπει να συνοδεύεται από έναν αρχικό έλεγχο συμβατότητας. Ο έλεγχος αυτός

είναι απαραίτητος για να διαπιστωθεί αν εμφανίζονται νέες ονομασίες στηλών, διαφορετικοί λογιστικοί κωδικοί, νέες αιτιολογίες, διαφορετικές κατηγορίες δωματίων ή λογιστικές περιπτώσεις που δεν καλύπτονται ήδη από τους κανόνες. Όπου εντοπίζονται τέτοιες διαφοροποιήσεις, η προσαρμογή μπορεί να πραγματοποιείται με προσθήκη νέων aliases, mappings ή ειδικών κανόνων, χωρίς να απαιτείται αλλαγή της συνολικής αρχιτεκτονικής ή ανακατασκευή του βασικού μηχανισμού επεξεργασίας και καθαρισμού.

Ένας δεύτερος περιορισμός συνδέεται με το μέγεθος και το χρονικό βάθος των διαθέσιμων δεδομένων. Σε εποχικές ξενοδοχειακές επιχειρήσεις, μερικά έτη ιστορικού αντιστοιχούν σε σχετικά μικρό αριθμό μηνιαίων παρατηρήσεων. Αυτό περιορίζει την αξιοπιστία ιδιαίτερα σύνθετων μοντέλων με μεγάλο αριθμό παραμέτρων και αποτελεί έναν από τους λόγους για τους οποίους η σύγκριση με απλούστερα εποχικά μοντέλα είναι απαραίτητη. Ένα μοντέλο μεγαλύτερης πολυπλοκότητας δεν είναι εξ ορισμού καλύτερο όταν το διαθέσιμο δείγμα είναι περιορισμένο.

Επιπλέον, η κύρια συχνότητα πρόβλεψης είναι μηνιαία. Η επιλογή αυτή είναι κατάλληλη για τους επιχειρησιακούς δείκτες της εργασίας και περιορίζει τον ημερήσιο θόρυβο, δεν καλύπτει όμως όλες τις πιθανές λειτουργικές ανάγκες της επιχείρησης. Αποφάσεις που σχετίζονται με βραχυπρόθεσμη πληρότητα, προσωπικό ή τιμολόγηση μπορεί να απαιτούν εβδομαδιαίο ή ακόμη και ημερήσιο επίπεδο δεδομένων.

Τέλος, η ποιότητα του τελικού αποτελέσματος εξακολουθεί να εξαρτάται από την ποιότητα της πηγής. Ο μηχανισμός καθαρισμού μπορεί να αναγνωρίσει πολλές διαφορετικές παραλλαγές, να τυποποιήσει τιμές και να εφαρμόσει κανόνες, δεν μπορεί όμως να ανακατασκευάσει με βεβαιότητα πληροφορία που δεν υπάρχει στο αρχικό σύστημα ή έχει καταχωριστεί με λανθασμένη επιχειρησιακή σημασία. Για τον λόγο αυτό η αυτοματοποίηση πρέπει να συνοδεύεται από μηχανισμούς παρακολούθησης της ποιότητας και από δυνατότητα ελέγχου των διαφορετικών περιπτώσεων.

6.4 Επέκταση της επεξεργασίας σε περισσότερες ξενοδοχειακές μονάδες

Μία από τις σημαντικότερες μελλοντικές επεκτάσεις είναι η χρήση της ίδιας πλατφόρμας από περισσότερες ξενοδοχειακές μονάδες. Η βασική αρχιτεκτονική έχει ήδη σχεδιαστεί προς αυτή την κατεύθυνση, καθώς τα uploads συνδέονται με `hotel_id` και τα cleaned records εμπλουτίζονται με `hotel_name` και `hotel_code`. Η προσθήκη μιας νέας μονάδας δεν προϋποθέτει ότι τα δεδομένα της θα έχουν ακριβώς την ίδια μορφή με όσα έχουν ήδη επεξεργαστεί. Αντίθετα, πριν από την πλήρη ένταξή της στη ροή της εφαρμογής απαιτείται έλεγχος των αρχείων της για τυχόν πρόσθετες επικεφαλίδες, διαφορετικό λογιστικό σχέδιο, νέους τύπους παραστατικών, ειδικές αιτιολογίες, διαφορετικές κατηγορίες δωματίων

ή άλλες ιδιαιτερότητες. Οι διαφοροποιήσεις αυτές μπορούν να ενσωματώνονται στους υπάρχοντες μηχανισμούς μέσω νέων aliases, mappings και, όπου απαιτείται, με πρόσθετους κανόνες ειδικής περίπτωσης.

Μία κατάλληλη εξέλιξη θα ήταν η δημιουργία δύο επιπέδων κανόνων. Το πρώτο θα περιλαμβάνει τις κοινές αντιστοιχίσεις που μπορούν να εφαρμοστούν ανεξάρτητα από τη μονάδα, όπως βασικά aliases ημερομηνιών, χρέωσης, πίστωσης και παραστατικού, καθώς και γενικούς κανόνες αναγνώρισης λογιστικών ομάδων. Το δεύτερο θα περιλαμβάνει προαιρετικές ρυθμίσεις που ενεργοποιούνται μόνο όταν μία πηγή παρουσιάζει ιδιαίτερη συμπεριφορά, όπως διαφορετικούς λογιστικούς κωδικούς, ειδικές ονομασίες δωματίων, νέες αιτιολογίες ή mappings συνεργατών. Με αυτή τη λογική, η προσθήκη νέων δεδομένων δεν απαιτεί δημιουργία νέου cleaner από την αρχή, αλλά έλεγχο και εμπλουτισμό του υπάρχοντος συνόλου κανόνων όπου αυτό είναι πραγματικά αναγκαίο.

Η λογική αυτή θα μπορούσε να υλοποιηθεί με μεταφορά περισσότερων mappings από τον Python κώδικα στη βάση. Για παράδειγμα, αντί ορισμένες ομάδες aliases ή ταξινομήσεων να βρίσκονται αποκλειστικά στον μηχανισμό καθαρισμού, θα μπορούσαν να αποθηκεύονται σε tables με πεδία όπως hotel_id, τύπος κανόνα, τιμή αντιστοίχισης, τελικό label, προτεραιότητα και κατάσταση ενεργοποίησης. Έτσι η προσαρμογή μιας νέας μονάδας θα γινόταν κυρίως με παραμετροποίηση και όχι με αλλαγή του βασικού κώδικα.

Παράλληλα, θα ήταν χρήσιμη η δημιουργία διαδικασίας αρχικού ελέγχου και χαρτογράφησης κάθε νέας πηγής δεδομένων. Μετά τα πρώτα uploads, το σύστημα θα μπορούσε να εντοπίζει άγνωστες επικεφαλίδες, νέους λογιστικούς κωδικούς, μη αντιστοιχισμένους τύπους δωματίων, νέες αιτιολογίες και εγγραφές που καταλήγουν σε γενικούς κανόνες. Οι περιπτώσεις αυτές θα μπορούσαν να παρουσιάζονται σε ξεχωριστή οθόνη διαχείρισης, ώστε να ελέγχεται αν απαιτείται νέα αντιστοίχιση ή ειδικός κανόνας. Οι επιβεβαιωμένες επιλογές θα αποθηκεύονται ως mappings ή tables και θα εφαρμόζονται αυτόματα στις επόμενες μεταφορτώσεις. Με τον τρόπο αυτό ο γενικός μηχανισμός καθαρισμού διατηρείται σταθερός, αλλά μπορεί να επεκτείνεται ελεγχόμενα όταν εμφανίζονται νέες επιχειρησιακές περιπτώσεις.

Σε περιβάλλον με περισσότερα ξενοδοχεία θα αποκτούσε επίσης μεγαλύτερη σημασία η δημιουργία κοινών αλλά και συγκριτικών KPIs. Θα μπορούσαν να δημιουργηθούν αναλύσεις ανά μονάδα, συνολικά σε επίπεδο ομίλου και ανά κατηγορία ξενοδοχείου. Η ίδια βάση θα μπορούσε να υποστηρίξει συγκρίσεις κόστους, κατανομής εσόδων, καναλιών πώλησης και εποχικότητας, εφόσον οι βασικές επιχειρησιακές έννοιες έχουν προηγουμένως κατηγοριοποιηθεί με κοινό τρόπο.

6.5 Εξέλιξη του υποσυστήματος πρόβλεψης και των στρατηγικών εκπαίδευσης

Το forecasting αποτελεί έναν από τους τομείς με τις περισσότερες δυνατότητες περαιτέρω ανάπτυξης. Η τρέχουσα προσέγγιση έχει το πλεονέκτημα ότι συγκρίνει μοντέλα διαφορετικής πολυπλοκότητας με κοινή διαδικασία αξιολόγησης. Σε επόμενη έκδοση θα μπορούσε να διευρυνθεί το σύνολο των υποψήφιων μοντέλων και κάθε νέο μοντέλο θα πρέπει να αξιολογείται με την ίδια χρονικά συνεπή διαδικασία και να παραμένει μόνο εφόσον προσφέρει πραγματική βελτίωση έναντι των απλούστερων baselines.

Ενδεικτικές μελλοντικές κατευθύνσεις είναι:

- μοντέλα gradient boosting, όπως XGBoost, LightGBM ή CatBoost, με κατάλληλη δημιουργία χρονικών και εποχικών χαρακτηριστικών,
- μοντέλα Prophet ή TBATS για εναλλακτική αντιμετώπιση τάσης και εποχικότητας,
- νευρωνικά μοντέλα ακολουθιών, όπως MLP ή GRU, όταν το διαθέσιμο ιστορικό γίνει επαρκές,
- ensembles, στα οποία η τελική πρόβλεψη προκύπτει από σταθμισμένο συνδυασμό περισσότερων μοντέλων αντί από έναν μόνο νικητή.

Η εκπαίδευση των μοντέλων στην παρούσα υλοποίηση δεν βασίζεται σε μία μόνο επιλογή χρονικού ιστορικού. Για κάθε διαδικασία αξιολόγησης εξετάζονται δύο διαφορετικές στρατηγικές εκπαίδευσης, η AllHistory και η Last30Months. Στην πρώτη περίπτωση χρησιμοποιούνται όλες οι ιστορικές παρατηρήσεις που είναι διαθέσιμες μέχρι το αντίστοιχο χρονικό σημείο του backtesting, ενώ στη δεύτερη περίπτωση το training set περιορίζεται στις τελευταίες 30 διαθέσιμες μηνιαίες παρατηρήσεις πριν από το ίδιο χρονικό σημείο.

Η διάκριση αυτή επιτρέπει να εξετάζεται αν η μεγαλύτερη ποσότητα ιστορικής πληροφορίας βελτιώνει την προβλεπτική απόδοση ή αν, αντίθετα, ένα μικρότερο και περισσότερο πρόσφατο παράθυρο εκπαίδευσης προσαρμόζεται καλύτερα σε αλλαγές της συμπεριφοράς της επιχείρησης. Και οι δύο στρατηγικές αξιολογούνται μέσα στο ίδιο χρονικό πλαίσιο backtesting, ώστε η σύγκρισή τους να πραγματοποιείται με κοινές συνθήκες και κοινές μετρικές.

Σε μελλοντική επέκταση, το σύνολο των training strategies θα μπορούσε να διευρυνθεί με επιπλέον χρονικά παράθυρα ή διαφορετικούς τρόπους αξιοποίησης του ιστορικού. Οι πρόσθετες στρατηγικές θα μπορούσαν να αξιολογούνται με την ίδια διαδικασία, ώστε η επιλογή τους να βασίζεται στην πραγματική προβλεπτική τους απόδοση και όχι σε έναν προκαθορισμένο κανόνα.

Η αξιολόγηση θα μπορούσε επίσης να επεκταθεί πέρα από την επιλογή ενός μοντέλου με βάση μία κύρια μέση μετρική. Θα μπορούσε να εξετάζεται η σταθερότητα της επίδοσης μεταξύ διαφορετικών folds, η συμπεριφορά σε μήνες υψηλής ζήτησης και η ύπαρξη συστηματικής υπερεκτίμησης ή υποεκτίμησης. Για παράδειγμα, δύο μοντέλα μπορεί να έχουν παρόμοιο MAE, αλλά το ένα να εμφανίζει πολύ μεγαλύτερη διακύμανση μεταξύ των folds. Σε επιχειρησιακή εφαρμογή η σταθερότητα αυτή μπορεί να είναι εξίσου σημαντική με τη μέση ακρίβεια.

Μία ακόμη ουσιαστική επέκταση είναι η παραγωγή διαστημάτων πρόβλεψης αντί μόνο σημειακής εκτίμησης. Η παρουσίαση ενός εύρους πιθανών τιμών θα μπορούσε να υποστηρίξει καλύτερα αποφάσεις υπό αβεβαιότητα. Για παράδειγμα, αντί να παρουσιάζεται μόνο μία εκτίμηση εσόδων, θα μπορούσε να δίνεται βασικό σενάριο και πιθανό καλύτερο και ανώτερο όριο. Η ίδια λογική θα μπορούσε να επεκταθεί σε αισιόδοξο, βασικό και συντηρητικό επιχειρησιακό σενάριο.

Πίνακας 6.1: Ενδεικτικές κατευθύνσεις εξέλιξης του υποσυστήματος πρόβλεψης.

Περιοχή	Προσέγγιση	Μελλοντική επέκταση
Μοντέλα	Seasonal Naive, σταθμισμένος εποχικός μέσος, ETS, SARIMA, Ridge και LSTM.	Gradient boosting, Prophet/TBATS, MLP, GRU και ensembles.
Εκπαίδευση	Δύο στρατηγικές: All History και Last 30 Months.	Πρόσθετες στρατηγικές εκπαίδευσης και εναλλακτικά χρονικά παράθυρα.
Αξιολόγηση	Χρονικά συνεπές backtesting με κοινές μετρικές αξιολόγησης.	Έλεγχος σταθερότητας μεταξύ των folds και ειδική αξιολόγηση ανά περίοδο.
Έξοδος	Σημειακή πρόβλεψη ανά χρονικό ορίζοντα.	Εκτίμηση της αβεβαιότητας και παρουσίαση εύρους πιθανών μελλοντικών τιμών μέσω διαστημάτων πρόβλεψης.
Συχνότητα	Μηνιαία κύρια συχνότητα.	Συμπληρωματική εβδομαδιαία πρόβλεψη για βραχυπρόθεσμες λειτουργικές ανάγκες.

6.6 Περαιτέρω αναλυτικές και επιχειρησιακές επεκτάσεις

Μία σημαντική μελλοντική επέκταση θα μπορούσε να είναι η αξιοποίηση των αποτελεσμάτων της ανάλυσης και του forecasting για πιο άμεση υποστήριξη επιχειρησιακών

αποφάσεων. Για παράδειγμα, μία περίοδος χαμηλής ζήτησης θα μπορούσε να συνδέεται με προτάσεις για ενίσχυση συγκεκριμένων καναλιών πώλησης ή προσαρμογή της τιμολογιακής πολιτικής. Με αυτόν τον τρόπο, το σύστημα θα μπορούσε να εξελιχθεί από εργαλείο ανάλυσης και πρόβλεψης σε πιο ολοκληρωμένο εργαλείο υποστήριξης αποφάσεων.

Μία δεύτερη κατεύθυνση είναι η ενσωμάτωση εξωγενών μεταβλητών στο forecasting. Η μελλοντική ζήτηση ενός ξενοδοχείου δεν εξαρτάται αποκλειστικά από το ιστορικό των κρατήσεων. Οι αργίες, τοπικές εκδηλώσεις, τιμές δωματίων, ακυρώσεις, η δυναμικότητα της επιχείρησης και, όπου είναι διαθέσιμα, εξωτερικοί δείκτες αγοράς θα μπορούσαν να χρησιμοποιηθούν ως πρόσθετα χαρακτηριστικά. Η ενσωμάτωση τέτοιων μεταβλητών θα επέτρεπε τη μετάβαση από καθαρά μονομεταβλητές χρονοσειρές σε περισσότερο επιχειρησιακά ενημερωμένα μοντέλα.

Για τη λειτουργική διαχείριση θα μπορούσε επίσης να προστεθεί συμπληρωματική εβδομαδιαία πρόβλεψη κρατήσεων. Η μηνιαία πρόβλεψη παραμένει καταλληλότερη για οικονομικό προγραμματισμό και συγκρίσεις μεγαλύτερου ορίζοντα, ενώ μία εβδομαδιαία σειρά θα μπορούσε να λειτουργεί ως εργαλείο για πληρότητα, προσωπικό και λειτουργικές ανάγκες. Οι δύο συχνότητες δεν χρειάζεται να αντικαθιστούν η μία την άλλη καθώς εξυπηρετούν διαφορετικό επίπεδο απόφασης.

Τέλος, σε περίπτωση επέκτασης της εφαρμογής σε περισσότερες ξενοδοχειακές μονάδες, θα μπορούσε να εξεταστεί η χρήση ιεραρχικών προβλέψεων. Το σύστημα θα μπορούσε να παράγει προβλέψεις ξεχωριστά για κάθε ξενοδοχείο και ταυτόχρονα μία συνολική πρόβλεψη για ολόκληρο τον όμιλο, με τρόπο ώστε τα επιμέρους αποτελέσματα να παραμένουν συνεπή με το συνολικό επίπεδο. Η ίδια λογική θα μπορούσε να εφαρμοστεί και σε χαμηλότερα επίπεδα ανάλυσης, όπως στην πρόβλεψη εσόδων ανά κανάλι πώλησης ή λειτουργικών εξόδων ανά κατηγορία. Με αυτόν τον τρόπο, το forecasting θα μπορούσε να υποστηρίξει όχι μόνο τη λειτουργία μιας μεμονωμένης μονάδας αλλά και τη συγκριτική και συνολική παρακολούθηση ενός μεγαλύτερου ξενοδοχειακού σχήματος.

6.7 Τελικό συμπέρασμα

Η βασική διαπίστωση της εργασίας είναι ότι η αξία της ανάλυσης επιχειρησιακών δεδομένων δεν εξαρτάται μόνο από το τελευταίο στάδιο του dashboard ή του μοντέλου πρόβλεψης. Η αξιοπιστία των αποτελεσμάτων καθορίζεται σε μεγάλο βαθμό από το πώς έχουν αναγνωριστεί, καθαριστεί, κατηγοριοποιηθεί και συνδεθεί οι αρχικές εγγραφές. Ένα πιο σύνθετο μοντέλο δεν μπορεί να αντισταθμίσει δεδομένα των οποίων η επιχειρησιακή σημασία δεν έχει οριστεί σωστά.

Για τον λόγο αυτό η παρούσα προσέγγιση έδωσε έμφαση σε ολόκληρη τη διαδρομή των δεδομένων. Η διατήρηση του αρχικού αρχείου, η αποθήκευση raw και cleaned εγγρα-

φών, η χρήση aliases και mappings, η διάκριση included και excluded ποσών, η δημιουργία σταθερών SQL views και η εφαρμογή κοινής διαδικασίας αξιολόγησης των προβλέψεων δημιουργούν ένα πλαίσιο στο οποίο το τελικό αποτέλεσμα είναι όχι μόνο υπολογίσιμο αλλά και ερμηνεύσιμο.

Η σημασία της εφαρμογής βρίσκεται συνεπώς στην ενοποίηση δύο διαφορετικών αναγκών. Από τη μία πλευρά μειώνει την χειροκίνητη εργασία που απαιτείται για την προετοιμασία αρχείων και από την άλλη δημιουργεί μία βάση για ανάλυση και πρόβλεψη. Η ίδια αρχιτεκτονική μπορεί να επεκταθεί σε νέα έτη, νέες μονάδες, νέα mappings, νέα μοντέλα και διαφορετικές αναλυτικές ανάγκες χωρίς να απαιτείται ανακατασκευή ολόκληρης της ροής.

Οι μελλοντικές επεκτάσεις που παρουσιάστηκαν δεν αλλάζουν τη βασική αρχή της εργασίας την ενισχύουν. Περισσότερα μοντέλα, πρόσθετες στρατηγικές εκπαίδευσης, πολυ-ξενοδοχειακή λειτουργία, αυτόματη παρακολούθηση ποιότητας και πλουσιότερα επιχειρησιακά σενάρια μπορούν να προστεθούν πάνω στην ίδια δομή. Η γενική λογική της εφαρμογής μπορεί να διατηρηθεί και σε νέα σύνολα δεδομένων, με την προϋπόθεση ότι κάθε νέα πηγή ελέγχεται αρχικά για πρόσθετες ιδιαιτερότητες και, όπου απαιτείται, εμπλουτίζεται το επίπεδο aliases, mappings και ειδικών κανόνων. Με αυτόν τον τρόπο το σύστημα μπορεί να εξελιχθεί σε γενικότερη πλατφόρμα υποστήριξης αποφάσεων για ξενοδοχειακές επιχειρήσεις, χωρίς να θεωρείται ότι όλες οι μελλοντικές πηγές ακολουθούν εκ των προτέρων την ίδια ακριβώς δομή.

Βιβλιογραφία

Batini, C., & Scannapieco, M. (2006). *Data Quality: Concepts, Methodologies and Techniques*. Springer.

Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (1994). *Time Series Analysis: Forecasting and Control* (3rd ed.). Prentice Hall.

Dowlut, N., & Gobin-Rahimbux, B. (2023). Forecasting resort hotel tourism demand using deep learning techniques: A systematic literature review. *Heliyon*, 9(7), e18385.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann.

Raisinghani, M. S. (2025). *Foundations of Information Systems*. OpenStax.

Weatherford, L. R., & Kimes, S. E. (2003). A comparison of forecasting methods for hotel revenue management. *International Journal of Forecasting*, 19(3), 401--415.