



Σχολή Θετικών Επιστημών και Τεχνολογίας
Μεταπτυχιακή Εξειδίκευση στα Πληροφοριακά Συστήματα

Διπλωματική Εργασία
Δυναμική Τιμολόγηση με Ελαχιστοποίηση της Σπατάλης
Τροφίμων μέσω Πρόβλεψης της Ζήτησης, ABC Ταξινόμησης και
Ενισχυτικής Μάθησης
Δημήτριος Πάλλας

Επιβλέπουσα καθηγήτρια: Ροδάνθη Τσώνη

Πάτρα, Σεπτέμβριος 2026

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

Δυναμική Τιμολόγηση με Ελαχιστοποίηση της Σπατάλης
Τροφίμων μέσω Πρόβλεψης της Ζήτησης, ABC Ταξινόμησης και
Ενισχυτικής Μάθησης
Δημήτριος Πάλλας

Επιτροπή Επίβλεψης Διπλωματικής Εργασίας

Επιβλέπουσα Καθηγήτρια:

Ροδάνθη Τσώνη

Μέλος ΣΕΠ, ΕΑΠ

Συν-Επιβλέπων Καθηγητής:

Αθανάσιος Σύψας

Μέλος ΣΕΠ, ΕΑΠ

Πάτρα, Σεπτέμβριος 2026

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στην επιβλέπουσα καθηγήτριά μου, κ. Τσώνη Ροδάνθη, για την πολύτιμη καθοδήγηση, την επιστημονική υποστήριξη και τις ουσιαστικές παρατηρήσεις της καθ' όλη τη διάρκεια εκπόνησης της παρούσας διπλωματικής εργασίας. Η εμπιστοσύνη, η διαθεσιμότητα και οι εύστοχες υποδείξεις της συνέβαλαν καθοριστικά τόσο στην ολοκλήρωση της εργασίας όσο και στη βελτίωση της ποιότητάς της.

Παράλληλα, θα ήθελα να ευχαριστήσω θερμά και τα μέλη της τριμελούς εξεταστικής επιτροπής για τον χρόνο που αφιέρωσαν στη μελέτη και αξιολόγηση της εργασίας, καθώς και για τις πολύτιμες παρατηρήσεις και υποδείξεις τους.

Τέλος, ένα ιδιαίτερο ευχαριστώ οφείλω στη σύντροφό μου για την αμέριστη υπομονή, την κατανόηση και τη συνεχή στήριξή της καθ' όλη τη διάρκεια της εκπόνησης της παρούσας εργασίας.

1. Περίληψη

Η σπατάλη τροφίμων αποτελεί μία από τις σημαντικότερες προκλήσεις του σύγχρονου λιανικού εμπορίου, καθώς επηρεάζει αρνητικά τόσο την οικονομική αποδοτικότητα των επιχειρήσεων όσο και τη βιωσιμότητα της εφοδιαστικής αλυσίδας. Επίσης, ιδιαίτερα αρνητικές είναι οι επιπτώσεις της σπατάλης των τροφίμων στο περιβάλλον. Ιδιαίτερα στα ευπαθή προϊόντα, η αβεβαιότητα της ζήτησης και η περιορισμένη διάρκεια ζωής καθιστούν κρίσιμη τη λήψη αποτελεσματικών αποφάσεων σχετικά με την τιμολόγηση και τη διαχείριση αποθεμάτων.

Σκοπός της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη ενός ολοκληρωμένου πλαισίου δυναμικής τιμολόγησης με στόχο τη μείωση της σπατάλης τροφίμων και τη βελτίωση της κερδοφορίας στο λιανικό εμπόριο. Η προτεινόμενη μεθοδολογία συνδυάζει πρόβλεψη ζήτησης με το μοντέλο NeuralProphet, πολυκριτηριακή ABC ανάλυση για την ταξινόμηση των προϊόντων και αλγόριθμο Ενισχυτικής μάθησης (Q-Learning) για τη λήψη αποφάσεων δυναμικής τιμολόγησης. Η αξιολόγηση της προσέγγισης πραγματοποιήθηκε σε περιβάλλον προσομοίωσης με χρήση συνθετικών δεδομένων που αναπαριστούν ρεαλιστικές συνθήκες λειτουργίας καταστήματος λιανικής.

Τα αποτελέσματα έδειξαν ότι η πολιτική Reinforcement Learning υπερέιχε συστηματικά μιας heuristic πολιτικής αναφοράς, επιτυγχάνοντας αύξηση της κερδοφορίας κατά 112,20%, αύξηση των πωλήσεων κατά 20,63% και μείωση των απορρίψεων κατά 40,69%. Παράλληλα, παρατηρήθηκε αύξηση των χαμένων πωλήσεων κατά 26,74%, γεγονός που αποδόθηκε στους περιορισμούς της σταθερής πολιτικής αναπλήρωσης αποθέματος και αναδείχθηκε ως σημαντικός συμβιβασμός μεταξύ κερδοφορίας και διαθεσιμότητας προϊόντων.

Η εργασία συμβάλλει στη βιβλιογραφία προτείνοντας μία ολοκληρωμένη προσέγγιση που συνδυάζει πρόβλεψη ζήτησης, πολυκριτηριακή ταξινόμηση αποθεμάτων και Reinforcement Learning για τη διαχείριση ευπαθών προϊόντων. Τα ευρήματα υποδεικνύουν ότι η δυναμική τιμολόγηση μπορεί να αποτελέσει αποτελεσματικό εργαλείο για τη μείωση της σπατάλης τροφίμων και τη βελτίωση της επιχειρησιακής απόδοσης, ενώ παράλληλα αναδεικνύουν την ανάγκη συνδυασμένης βελτιστοποίησης της τιμολόγησης και της πολιτικής αναπλήρωσης αποθεμάτων.

Λέξεις – Κλειδιά

Πολυκριτηριακή ABC ταξινόμηση, Δυναμική τιμολόγηση, Μηχανική μάθηση, Ενισχυτική μάθηση, Ευπαθή προϊόντα, σπατάλη τροφίμων.

Dynamic Pricing for Food Waste Reduction through Demand Forecasting, ABC Classification, and Reinforcement Learning

Dimitrios Pallas

Abstract

Food waste is one of the most significant challenges in modern food retailing, affecting both business profitability and the sustainability of supply chains. In the case of perishable products, demand uncertainty and limited shelf life make pricing and inventory management critical decision-making processes.

The aim of this thesis is to develop an integrated dynamic pricing framework for reducing food waste while improving profitability in the retail sector. The proposed methodology combines demand forecasting using the NeuralProphet model, multi-criteria ABC inventory classification, and a Reinforcement Learning (Q-Learning) algorithm for dynamic pricing decisions. The proposed approach was evaluated in a simulation environment using synthetic data designed to represent realistic retail operating conditions. The experimental results demonstrated that the Reinforcement Learning policy consistently outperformed a heuristic baseline policy. More specifically, profitability increased by 112.20%, sales increased by 20.63%, and product waste was reduced by 40.69%. At the same time, lost sales increased by 26.74%, highlighting an important trade-off between profitability and product availability. This outcome is mainly attributed to the fixed inventory replenishment policy adopted in the simulation, where replenishment occurs every seven days and the learning agent is not allowed to modify replenishment decisions. Consequently, higher demand generated by dynamic pricing strategies may lead to more frequent stockouts.

This thesis contributes to the existing literature by proposing an integrated framework that combines demand forecasting, multi-criteria ABC inventory classification, and Reinforcement Learning for the dynamic pricing of perishable food products. The findings

indicate that Reinforcement Learning can effectively improve operational performance by increasing profitability and reducing food waste. At the same time, the study highlights the importance of jointly optimizing pricing and inventory replenishment policies in future research to achieve a better balance between profitability, food waste reduction, and product availability.

Keywords

Multi-criteria ABC classification, Dynamic pricing, Machine learning, Reinforcement learning, Perishable food products, Food waste.

2. Περιεχόμενα

1. Περίληψη.....	v
2. Περιεχόμενα.....	ix
3. Κατάλογος Εικόνων / Σχημάτων	xv
4. Κατάλογος Πινάκων	xvi
5. Συντομογραφίες & Ακρωνύμια.....	xviii
1. Εισαγωγή.....	1
1.1. Αναγκαιότητα και σημασία της προτεινόμενης έρευνας	2
1.2. Στόχος της εργασίας.....	3
1.3. Δομή της εργασίας	3
2. Θεωρητικό υπόβαθρο και βιβλιογραφική ανασκόπηση.....	5
2.1. Σπατάλη τροφίμων και διαχείριση ευπαθών αποθεμάτων.....	5
2.1.1. Ορισμός και χαρακτηριστικά της σπατάλης τροφίμων.....	5
2.1.2. Αίτια σπατάλης τροφίμων στο λιανικό εμπόριο	6
2.1.3. Ευπαθή προϊόντα και διαχείριση αποθεμάτων.....	8
2.1.4. Οικονομικές και περιβαλλοντικές επιπτώσεις της σπατάλης τροφίμων.....	10
2.2. Μηχανική μάθηση.....	11
2.2.1. Ορισμός και ιστορική εξέλιξη.....	11
2.2.2. Βασικές κατηγορίες μηχανικής μάθησης.....	12
2.2.3. Η σημασία της μηχανικής μάθησης στο λιανικό εμπόριο τροφίμων.....	13
2.2.4. Ενισχυτική μάθηση (Reinforcement learning).....	14
2.2.5. Βασικές προσεγγίσεις αλγορίθμων ενισχυτικής μάθησης	15
2.2.6. Η επιλογή του αλγορίθμου Q – Learning.....	17
2.3. Μοντέλα πρόβλεψης	19
2.3.1. Ορισμός, σκοπός και ιστορική αναδρομή των μοντέλων πρόβλεψης	19
2.3.2. Κατηγορίες μοντέλων πρόβλεψης.....	20
2.3.3. Αναλυτική παρουσίαση βασικών μοντέλων	21
2.3.3.1. Γραμμική Παλινδρόμηση.....	21
2.3.3.2. Λογιστική Παλινδρόμηση	22
2.3.3.3. Μοντέλα χρονοσειρών	23

2.3.3.4. Τεχνητά νευρωνικά δίκτυα.....	24
2.3.4. Κριτήρια αξιολόγησης των μοντέλων.....	25
2.3.5. Πλεονεκτήματα και μειονεκτήματα των μοντέλων πρόβλεψης	26
2.3.6. Σημαντικότητα της εφαρμογής μοντέλων πρόβλεψης στην διπλωματική εργασία	27
2.4. Ορισμός και βασικές αρχές της ABC ανάλυσης.....	27
2.4.1. ABC ανάλυση με πολλαπλά κριτήρια σε περιβάλλοντα εμπορίου.....	28
2.4.2. Μέθοδοι πολυκριτηριακής ABC ανάλυσης	30
2.5. Δυναμική τιμολόγηση	31
2.5.1. Ορισμός και βασικές αρχές της δυναμικής τιμολόγησης.....	31
2.5.2. Κατηγορίες δυναμικής τιμολόγησης.....	32
2.5.3. Περιορισμοί και προκλήσεις της δυναμικής τιμολόγησης.....	33
2.5.4. Δυναμική τιμολόγηση και διαχείριση αποθεμάτων σε ευπαθή προϊόντα	33
2.6. Ενισχυτική μάθηση για πολιτικές τιμολόγησης & αναπλήρωσης προϊόντων.	34
2.7. Μείωση σπατάλης στο λιανικό εμπόριο μέσω βελτιστοποίησης λειτουργιών	36
3. Δεδομένα.....	38
3.1. Χρήση συνθετικών δεδομένων	38
3.2. Δομή συνόλου δεδομένων.....	39
3.3. Χρονικό εύρος και επίπεδο ανάλυσης	40
3.4. Προϊόντα και κατηγορίες	40
3.5. Τιμολόγηση προϊόντων	42
3.6. Ημερολόγιο λειτουργίας και εμπορικά γεγονότα.....	42
3.7. Μοντελοποίηση ζήτησης	44
3.8. Κλειστές ημέρες και μεταφορά ζήτησης.....	45
3.9. Διαχείριση αποθεμάτων και αλλοίωση προϊόντων	45
3.10. Σύνοψη	46
4. Μεθοδολογία.....	47
4.1. Εργαλεία και λογισμικό	47
4.1.1. Συσκευή.....	47
4.1.2. Γλώσσες προγραμματισμού και βιβλιοθήκες που χρησιμοποιήθηκαν	48
4.1.2.1. Περιβάλλον ανάπτυξης λογισμικού (IDE).....	48

4.1.2.2. Python 3.13.13	48
4.1.2.3. Pandas 2.3.3	49
4.1.2.4. Numpy 1.26.4.....	49
4.1.2.5. Matplotlib 3.10.8.....	49
4.1.2.6. NeuralProphet 0.8.0	49
4.1.2.7. Scipy 1.18.0.....	51
4.1.2.8. Ο αλγόριθμος Q – Learning	51
4.2. Πορεία που ακολουθήθηκε	52
4.2.1. Πρόβλεψη ζήτησης	52
4.2.1.1. Προεπεξεργασία δεδομένων	53
4.2.1.2. Διαχωρισμός Train/Test	53
4.2.1.3. Μοντέλο NeuralProphet.....	54
4.2.1.4. Αξιολόγηση του μοντέλου	54
4.2.1.5. Οπτικοποίηση των αποτελεσμάτων	55
4.2.2. Πολυκριτηριακή ABC Ανάλυση και ταξινόμηση προϊόντων.....	55
4.2.2.1. Συλλογή και ενοποίηση δεδομένων	56
4.2.2.2. Υπολογισμός πραγματικής ζήτησης	56
4.2.2.3. Υπολογισμός βασικών δεικτών απόδοσης (KPIs)	57
4.2.2.4. Τόνωση ζήτησης λόγω γεγονότος.....	58
4.2.2.5. Ανάλυση συσχέτισης μεταβλητών	58
4.2.2.6. Κανονικοποίηση δεδομένων	59
4.2.2.7. Σύστημα βαρών.....	59
4.2.2.8. Υπολογισμός ABC Score.....	60
4.2.2.9. Κατηγοριοποίηση A/B/C	61
4.2.3. Δημιουργία συνόλου δεδομένων για ενισχυτική μάθηση.....	61
4.2.3.1. Ενοποίηση δεδομένων.....	62

4.2.3.2. Εμπλουτισμός χαρακτηριστικών.....	62
4.2.3.3. Υπολογισμός Χαρακτηριστικών κατάστασης και κατασκευή νέων μεταβλητών	63
4.2.3.4. Μοντελοποίηση Ελαστικότητας ζήτησης ως προς την Τιμή	63
4.2.3.5. Μετασχηματισμός δεικτών γεγονότων	64
4.2.3.6. Καθαρισμός και επικύρωση δεδομένων	65
4.2.3.7. Δημιουργία τελικού σετ δεδομένων για την ενισχυτική μάθηση	65
4.2.4. Σχεδιασμός Περιβάλλοντος Ενισχυτικής μάθησης.....	66
4.2.4.1. Χώρος ενεργειών (Action Space)	67
4.2.4.2. Επιχειρησιακές παραδοχές	68
4.2.4.3. Αρχικοποίηση και δυναμική αποθέματος	69
4.2.4.4. Αναπαράσταση κατάστασης (State representation).....	69
4.2.4.5. Διακριτοποίηση καταστάσεων (State encoding).....	71
4.2.4.6. Προσομοίωση ζήτησης	75
4.2.4.7. Διαχείριση αποθέματος και μετάβαση καταστάσεων	78
4.2.4.8. Συνάρτηση ανταμοιβής (Reward function).....	80
4.2.4.9. Εκτέλεση βήματος περιβάλλοντος (Environment step).....	84
4.2.5. Πειραματική διαδικασία εκπαίδευσης του πράκτορα και παραγωγή αποτελεσμάτων	85
4.2.5.1. Επιλογή προϊόντων εκπαίδευσης	86
4.2.5.2. Παράμετροι εκπαίδευσης	86
4.2.5.3. Πολιτική εξερεύνησης Epsilon – Greedy.....	89
4.2.5.4. Αναπαράσταση και ενημέρωση του πίνακα Q (Q - table).....	91
4.2.5.5. Πειραματικός σχεδιασμός τυχαίας αρχικοποίησης.....	93
4.2.5.6. Πολιτική αναφοράς (Heuristic Baseline Policy).....	94
4.2.5.7. Διαδικασία αξιολόγησης και δείκτες απόδοσης	95

4.2.5.8. Οπτικοποίηση και ερμηνεία πολιτικών.....	97
4.2.5.9. Συνοπτική περιγραφή του πειραματικού σεναρίου.....	98
5. Αποτελέσματα και συζήτηση.....	100
5.1. Αποτελέσματα δεδομένων	101
5.1.1. Γεγονότα ημερολογίου.....	102
5.1.2. Προϊόντα	103
5.1.3. Ημερήσιες συναλλαγές	104
5.1.4. Ημερήσιες απορρίψεις	105
5.2. Αποτελέσματα της πρόβλεψης της ζήτησης	105
5.2.1. Αξιολόγηση του μοντέλου NeuralProphet.....	105
5.2.2. Μετρικά δεδομένα ανά κωδικό προϊόντος (SKUs).....	107
5.2.3. Οπτική αξιολόγηση προβλέψεων.....	108
5.2.4. Σχολιασμός αποτελεσμάτων της πρόβλεψης της ζήτησης.....	110
5.3. Αποτελέσματα της πολυκριτηριακής ABC ανάλυσης.....	111
5.3.1. Ταξινόμηση προϊόντων και σχόλια.....	111
5.3.2. Πίνακες ABC	112
5.3.3. Αποτελέσματα διερεύνησης σχέσεων μεταξύ κριτηρίων για την ABC ανάλυση	121
5.4. Αποτελέσματα δεδομένων για την χρήση της ενισχυτικής μάθησης.....	122
5.5. Αποτελέσματα της ενισχυτικής μάθησης.....	123
5.5.1. Οπτικοποίηση αποτελεσμάτων	126
6. Συμπεράσματα	133
6.1. Περιορισμοί της έρευνας.....	134
6.2. Μελλοντικές προεκτάσεις.....	135
7. Βιβλιογραφία.....	136
8. Παράρτημα Α: Ο κώδικας της εργασίας και τα αποτελέσματα.....	147
8.1. Ο κώδικας για την παραγωγή των δεδομένων της εργασίας	147
8.2. Ο κώδικας για την πρόβλεψη της ζήτησης	174
8.3. Ο κώδικας για την ABC ανάλυση.....	187
8.4. Ο κώδικας της τελικής μορφής των δεδομένων για το RL	199
8.5. Ο κώδικας για το περιβάλλον του RL.....	212

8.6. Ο κώδικας για την εκπαίδευση του πράκτορα	230
8.7. Τελικά αποτελέσματα του πράκτορα	262

3. Κατάλογος Εικόνων / Σχημάτων

<i>Εικόνα 4.1 Διάγραμμα ροής μεθοδολογίας της παρούσας εργασίας</i>	<i>52</i>
<i>Εικόνα 4.2 Καμπύλη μάθησης του πράκτορα για το προϊόν P002 με 10000 επεισόδια</i>	<i>88</i>
<i>Εικόνα 4.3 Καμπύλη μάθησης του πράκτορα για το προϊόν P002 με 3000 επεισόδια</i>	<i>88</i>
<i>Εικόνα 4.4 Καμπύλη μάθησης του πράκτορα για το προϊόν P002 με 2200 επεισόδια</i>	<i>89</i>
<i>Εικόνα 5.1 Αποτελέσματα πρόβλεψης για 4 αντιπροσωπευτικά προϊόντα σε ιστόγραμμα..</i>	<i>109</i>
<i>Εικόνα 5.2 Αποτελέσματα πρόβλεψης για 4 αντιπροσωπευτικά προϊόντα σε line plot.....</i>	<i>110</i>
<i>Εικόνα 5.3 Ραβδόγραμμα μέσων τιμών ανά ABC κατηγορία</i>	<i>112</i>
<i>Εικόνα 5.4 Heatmap συσχέτιση μεταβλητών των βαρών.....</i>	<i>121</i>
<i>Εικόνα 5.5 Heatmap P002_seed_1</i>	<i>127</i>
<i>Εικόνα 5.6 Heatmap P016_seed_7</i>	<i>127</i>
<i>Εικόνα 5.7 Heatmap P117_seed_7</i>	<i>128</i>
<i>Εικόνα 5.8 Improvement distribution.....</i>	<i>128</i>
<i>Εικόνα 5.9 Products reward RL vs Heuristic.....</i>	<i>130</i>
<i>Εικόνα 5.10 Improvement per SKU RL vs Heuristic.....</i>	<i>131</i>
<i>Εικόνα 5.11 Ραβδόγραμμα RL vs Baseline σε % κέρδους, πωλήσεων, απορρίψεων ανά προϊόν</i>	<i>132</i>
<i>Εικόνα 8.1 RL all results seed 1</i>	<i>263</i>
<i>Εικόνα 8.2 RL all results seed 7</i>	<i>263</i>
<i>Εικόνα 8.3 RL all results seed 21</i>	<i>263</i>
<i>Εικόνα 8.4 RL all results seed 42</i>	<i>264</i>
<i>Εικόνα 8.5 RL all results seed 99</i>	<i>264</i>
<i>Εικόνα 8.6 RL Business summary</i>	<i>265</i>

4. Κατάλογος Πινάκων

Πίνακας 4.1 Βάρη μεταβλητών για την ABC ανάλυση	60
Πίνακας 4.2 Ελαστικότητα ζήτησης ως προς την τιμή για τις κατηγορίες προϊόντων	64
Πίνακας 4.3 Επιχειρησιακές παραδοχές του περιβάλλοντος μάθησης του πράκτορα.....	69
Πίνακας 4.4 Μεταβλητές κατάστασης.....	70
Πίνακας 4.5 Διακριτοποίηση τιμών καταστάσεων	72
Πίνακας 4.6 Διακριτοποίηση αποθέματος καταστάσεων	72
Πίνακας 4.7 Διακριτοποίηση υπολειπόμενης διάρκειας ζωής καταστάσεων	73
Πίνακας 4.8 Διακριτοποίηση ρυθμού αλλοίωσης καταστάσεων	73
Πίνακας 4.9 Διακριτοποίηση αύξησης της ζήτησης λόγω ειδικών γεγονότων καταστάσεων.....	74
Πίνακας 4.10 Διακριτοποίηση ελαστικότητας τιμής καταστάσεων	74
Πίνακας 4.11 Διακριτοποίηση κρισιμότητας προϊόντος καταστάσεων.....	74
Πίνακας 4.12 Μεταβλητές εκπαίδευσης του πράκτορα	87
Πίνακας 5.1 αρχικών δεδομένων.....	102
Πίνακας 5.2 Δεδομένων γεγονότων	102
Πίνακας 5.3 Σύνοψης προϊόντων	104
Πίνακας 5.4 Περιγραφικής στατιστικής ημερήσιων συναλλαγών.....	105
Πίνακας 5.5 Πειραματικής ρύθμισης NeuralProphet	106
Πίνακας 5.6 Αξιολόγησης σφαλμάτων της πρόβλεψης.....	106
Πίνακας 5.7 Οι δέκα καλύτερες εκτιμήσεις MAE / RMSE	108
Πίνακας 5.8 Οι δέκα χειρότερες εκτιμήσεις MAE / RMSE	108
Πίνακας 5.9 Δεδομένα ανά ABC κατηγορία προϊόντος.....	111
Πίνακας 5.10 Κατηγορία A	114
Πίνακας 5.11 Κατηγορία B	116
Πίνακας 5.12 Κατηγορία C	120
Πίνακας 5.13 Δομή του RL συνόλου δεδομένων.....	123
Πίνακας 5.14 Κύρια αποτελέσματα του πράκτορα σε σχέση με την ευρετική μέθοδο.....	125
Πίνακας 5.15 RL vs Baseline.....	126

5. Συντομογραφίες & Ακρωνύμια

ΔΕ	Διπλωματική Εργασία
ΕΑΠ	Ελληνικό Ανοικτό Πανεπιστήμιο
ΠΕ	Πτυχιακή Εργασία
ΣΥΝ	Συντονιστής/Συντονίστρια
DQN	Deep Q-Networks
KPIs	Key Performance Indicators
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MSE	Mean Squared Error
LSTM	Long Short-Term Memory
RMSE	Root Mean Squared Error
RL	Reinforcement Learning
SKUs	Stock Keeping Units
sMAPE	Symmetric Mean Absolute Percentage Error

1. Εισαγωγή

Η σπατάλη τροφίμων αποτελεί ένα από τα σημαντικότερα προβλήματα της σύγχρονης εφοδιαστικής αλυσίδας, καθώς εκτιμάται ότι περίπου το 19% της παγκόσμιας παραγόμενης τροφής δεν καταναλώνεται και καταλήγει ως απόβλητο, με σημαντικές οικονομικές και περιβαλλοντικές επιπτώσεις ενώ 820 εκατομμύρια συνανθρώπων μας υποφέρουν από την πείνα σε καθημερινή βάση (FAO, 2019). Το πρόβλημα είναι ιδιαίτερα έντονο στο λιανικό εμπόριο τροφίμων, όπου η αβεβαιότητα της ζήτησης και η περιορισμένη διάρκεια ζωής των προϊόντων οδηγούν σε υπεραποθέματα και απορρίψεις (Riesenegger & Hübner, 2022).

Σε θεσμικό επίπεδο, η Ευρωπαϊκή Ένωση έχει αναγνωρίσει τη σπατάλη τροφίμων ως κρίσιμο ζήτημα δημόσιας πολιτικής, θέτοντας μέσω της στρατηγικής Farm to Fork και της αναθεώρησης της Οδηγίας για τα Απόβλητα δεσμευτικό στόχο μείωσης της σπατάλης τροφίμων κατά 30% έως το 2030, με ιδιαίτερη έμφαση στα στάδια του λιανικού εμπορίου και της κατανάλωσης (*European Parliament Backs New 2030 EU Food Waste Targets*, 2025). Η παρέμβαση αυτή αντανακλά την ανάγκη διόρθωσης μιας αποτυχίας της αγοράς, όπου το κοινωνικό κόστος της σπατάλης δεν ενσωματώνεται πλήρως στις τιμές. Το φαινόμενο αυτό μπορεί να ερμηνευθεί μέσω της θεωρίας των αρνητικών εξωτερικοτήτων του Ρίγου, σύμφωνα με την οποία, όταν το ιδιωτικό κόστος αποκλίνει από το κοινωνικό κόστος, η αγορά οδηγείται σε μη αποδοτικά αποτελέσματα (Cannan & Pigou, 1921). Στην περίπτωση των τροφίμων, οι περιβαλλοντικές επιπτώσεις και η απώλεια φυσικών πόρων δεν αντικατοπτρίζονται στις αποφάσεις τιμολόγησης και αποθεματοποίησης, γεγονός που ενθαρρύνει τη σπατάλη. Ωστόσο, πρόσφατες εμπειρικές μελέτες δείχνουν ότι μηχανισμοί αγοράς, όπως η δυναμική τιμολόγηση, μπορούν να λειτουργήσουν ως μηχανισμοί τιμολόγησης του κινδύνου σπατάλης, ενσωματώνοντας το κόστος της αλλοίωσης στις αποφάσεις αγοράς και πώλησης, μειώνοντας τα απόβλητα χωρίς να θυσιάζεται η κερδοφορία των επιχειρήσεων (Sanders, 2024).

Στο πλαίσιο αυτό, καθίσταται επιτακτική η ανάγκη ανάπτυξης μεθοδολογιών που μειώνουν τη σπατάλη τροφίμων διατηρώντας παράλληλα το οικονομικό αποτέλεσμα σε βιώσιμα

επίπεδα. Η βιβλιογραφία υποδεικνύει ότι ο συνδυασμός πολυκριτηριακής ABC ταξινόμησης, πρόβλεψης ζήτησης και δυναμικής τιμολόγησης επιτρέπει πιο στοχευμένες αποφάσεις διαχείρισης αποθεμάτων ευπαθών προϊόντων (Chu et al., 2008; Flores & Whybark, 1987). Περαιτέρω, η αξιοποίηση τεχνικών Ενισχυτικής Μάθησης (Reinforcement Learning) παρέχει τη δυνατότητα εκμάθησης δυναμικών πολιτικών τιμολόγησης και αναπλήρωσης μέσω αλληλεπίδρασης με το περιβάλλον, λαμβάνοντας υπόψη τόσο την αβεβαιότητα της ζήτησης όσο και τις μελλοντικές επιπτώσεις των αποφάσεων, και έχει αποδειχθεί αποτελεσματική στη μείωση αποβλήτων και τη διατήρηση της κερδοφορίας σε συστήματα ευπαθών αποθεμάτων (Kara & Dogan, 2018; Yavuz & Kaya, 2024).

1.1. Αναγκαιότητα και σημασία της προτεινόμενης έρευνας

Η σπατάλη τροφίμων αποτελεί ένα από τα σημαντικότερα προβλήματα της σύγχρονης εφοδιαστικής αλυσίδας τροφίμων, καθώς συνδέεται άμεσα με οικονομικές απώλειες, περιβαλλοντικές επιπτώσεις και αναποτελεσματική αξιοποίηση φυσικών πόρων. Το πρόβλημα είναι ιδιαίτερα έντονο στο λιανικό εμπόριο, όπου η αβεβαιότητα της ζήτησης, η περιορισμένη διάρκεια ζωής των ευπαθών προϊόντων και οι ακατάλληλες αποφάσεις αποθεματοποίησης και τιμολόγησης οδηγούν συχνά σε αυξημένες απορρίψεις προϊόντων και χαμένες πωλήσεις. Παράλληλα, η Ευρωπαϊκή Ένωση έχει θέσει ως στρατηγικό στόχο τη σημαντική μείωση της σπατάλης τροφίμων έως το 2030, γεγονός που καθιστά αναγκαία την ανάπτυξη νέων εργαλείων λήψης αποφάσεων για τις επιχειρήσεις λιανικής.

Η υφιστάμενη βιβλιογραφία τεκμηριώνει εκτενώς τη χρήση τεχνικών δυναμικής τιμολόγησης και διαχείρισης αποθεμάτων για ευπαθή προϊόντα, καθώς και την αποτελεσματικότητα αλγορίθμων Ενισχυτικής Μάθησης (Reinforcement Learning) στη βελτιστοποίηση κερδοφορίας και επιπέδων αποθέματος υπό αβεβαιότητα ζήτησης (Kara & Dogan, 2018; Yavuz & Kaya, 2024). Ωστόσο, οι περισσότερες σχετικές μελέτες επικεντρώνονται κυρίως σε δείκτες οικονομικής απόδοσης, όπως το αναμενόμενο κέρδος ή το συνολικό κόστος, και δεν εξετάζουν ρητά τη σπατάλη τροφίμων ως πρωτεύοντα στόχο βελτιστοποίησης. Επιπλέον, η βιβλιογραφία συνήθως προσεγγίζει μεμονωμένα τα

προβλήματα της πρόβλεψης ζήτησης, της ταξινόμησης αποθεμάτων και της δυναμικής τιμολόγησης, χωρίς να τα ενσωματώνει σε ένα ενιαίο πλαίσιο λήψης αποφάσεων.

Η παρούσα εργασία επιχειρεί να καλύψει αυτό το ερευνητικό κενό προτείνοντας μία ολοκληρωμένη μεθοδολογία που συνδυάζει πρόβλεψη ζήτησης, πολυκριτηριακή ABC ταξινόμηση και δυναμική τιμολόγηση μέσω Reinforcement Learning. Η προσέγγιση αυτή στοχεύει όχι μόνο στη βελτίωση της κερδοφορίας αλλά και στη μείωση της σπατάλης τροφίμων, αντιμετωπίζοντας ταυτόχρονα τις επιχειρησιακές και περιβαλλοντικές διαστάσεις του προβλήματος. Με τον τρόπο αυτό, η έρευνα συμβάλλει τόσο στη βιβλιογραφία της διαχείρισης αποθεμάτων και της δυναμικής τιμολόγησης όσο και στην ανάπτυξη πρακτικών λύσεων που μπορούν να υποστηρίξουν πιο βιώσιμα συστήματα λιανικού εμπορίου τροφίμων.

1.2. Στόχος της εργασίας

Ο στόχος της διπλωματικής εργασίας είναι η μελέτη και εφαρμογή μιας μεθοδολογίας που συνδυάζει πρόβλεψη ζήτησης, πολυκριτηριακή ABC ανάλυση και δυναμική τιμολόγηση με τεχνικές ενισχυτικής μάθησης (Reinforcement Learning), με απώτερο σκοπό τη μείωση της σπατάλης τροφίμων στο λιανικό εμπόριο και τη βελτιστοποίηση των αποθεμάτων προϊόντων μικρής διάρκειας ζωής.

1.3. Δομή της εργασίας

Η παρούσα διπλωματική εργασία οργανώνεται σε οκτώ κεφάλαια.

Στο κεφάλαιο 1 περιγράφεται η αναγκαιότητα της εργασίας, ο στόχος και η δομή της καθώς και μια εισαγωγή που προϋδεάζει τον αναγνώστη για αυτά που θα ακολουθήσουν.

Στο κεφάλαιο 2 παρουσιάζεται το θεωρητικό υπόβαθρο και η βιβλιογραφική ανασκόπηση που σχετίζονται με τη διαχείριση ευπαθών προϊόντων, τη δυναμική τιμολόγηση, την πολυκριτηριακή ABC ανάλυση, την πρόβλεψη ζήτησης και τις τεχνικές Reinforcement Learning.

Στο κεφάλαιο 3 περιγράφονται τα δεδομένα που χρησιμοποιήθηκαν στην έρευνα, καθώς και η διαδικασία δημιουργίας και προεπεξεργασίας του συνθετικού συνόλου δεδομένων.

Στο κεφάλαιο 4 παρουσιάζεται αναλυτικά η προτεινόμενη μεθοδολογία. Περιγράφονται η διαδικασία πρόβλεψης της ζήτησης, η εφαρμογή της πολυκριτηριακής ABC ανάλυσης, η διαμόρφωση του περιβάλλοντος Reinforcement Learning και η ανάπτυξη του αλγορίθμου δυναμικής τιμολόγησης.

Στο κεφάλαιο 5 παρουσιάζονται και αναλύονται τα πειραματικά αποτελέσματα, καθώς και η αξιολόγηση της προτεινόμενης προσέγγισης σε σύγκριση με τη heuristic πολιτική αναφοράς.

Στο κεφάλαιο 6 συνοψίζονται τα βασικά συμπεράσματα της έρευνας, αναλύονται οι περιορισμοί της μελέτης και προτείνονται πιθανές κατευθύνσεις για μελλοντική έρευνα.

Στο κεφάλαιο 7 παρατίθεται η βιβλιογραφία που χρησιμοποιήθηκε για τη θεωρητική τεκμηρίωση και την υλοποίηση της εργασίας.

Τέλος, στο κεφάλαιο 8 παρουσιάζονται τα παραρτήματα, τα οποία περιλαμβάνουν τον κώδικα που αναπτύχθηκε και τα τελικά αποτελέσματα του πειράματος.

2. Θεωρητικό υπόβαθρο και βιβλιογραφική ανασκόπηση

Σε αυτή την ενότητα θα γίνει μια προσπάθεια να κατατοπιστεί ο αναγνώστης σχετικά με την υπάρχουσα κατάσταση και το θεωρητικό υπόβαθρο που θα στηριχτεί η παρούσα εργασία.

2.1. Σπατάλη τροφίμων και διαχείριση ευπαθών αποθεμάτων

2.1.1. Ορισμός και χαρακτηριστικά της σπατάλης τροφίμων

Η σπατάλη τροφίμων (food waste) αποτελεί ένα από τα σημαντικότερα προβλήματα των σύγχρονων συστημάτων παραγωγής και διανομής τροφίμων, καθώς συνδέεται άμεσα με οικονομικές απώλειες, περιβαλλοντικές επιπτώσεις και ζητήματα βιωσιμότητας. Σύμφωνα με τον Οργανισμό Τροφίμων και Γεωργίας των Ηνωμένων Εθνών (Food and Agriculture Organization – FAO), ως σπατάλη τροφίμων ορίζεται η απόρριψη ή μη αξιοποίηση τροφίμων που είναι κατάλληλα για ανθρώπινη κατανάλωση και λαμβάνει χώρα κυρίως στα τελικά στάδια της εφοδιαστικής αλυσίδας, δηλαδή στο λιανικό εμπόριο και στην κατανάλωση (FAO, 2019).

Στη βιβλιογραφία γίνεται συχνά διάκριση μεταξύ των εννοιών «food loss» και «food waste». Ο όρος food loss αναφέρεται στις απώλειες που προκύπτουν στα αρχικά στάδια της εφοδιαστικής αλυσίδας, όπως η παραγωγή, η συγκομιδή, η αποθήκευση και η μεταφορά, ενώ ο όρος food waste αφορά κυρίως την απόρριψη τροφίμων σε επίπεδο λιανικού εμπορίου και τελικής κατανάλωσης (FAO, 2019). Η διάκριση αυτή είναι ιδιαίτερα σημαντική, καθώς τα αίτια και οι μέθοδοι αντιμετώπισης διαφοροποιούνται σημαντικά ανάλογα με το στάδιο στο οποίο εμφανίζονται οι απώλειες.

Στο λιανικό εμπόριο τροφίμων, η σπατάλη συνδέεται κυρίως με την αβεβαιότητα της ζήτησης, τη μικρή διάρκεια ζωής των ευπαθών προϊόντων, τις πολιτικές αποθεματοποίησης και τις στρατηγικές τιμολόγησης. Τα καταστήματα λιανικής συχνά διατηρούν υψηλά επίπεδα αποθέματος ώστε να αποφεύγονται ελλείψεις προϊόντων και απώλειες πωλήσεων. Ωστόσο, η πρακτική αυτή αυξάνει σημαντικά τον κίνδυνο λήξης και απόρριψης προϊόντων, ιδιαίτερα σε κατηγορίες όπως τα φρούτα, τα λαχανικά, τα γαλακτοκομικά και τα προϊόντα κρέατος (Riesenegger & Hübner, 2022).

Παράλληλα, η σπατάλη τροφίμων δεν αφορά αποκλειστικά την απώλεια προϊόντων αλλά και τη σπατάλη των φυσικών πόρων που απαιτήθηκαν για την παραγωγή τους, όπως νερό, ενέργεια, καλλιεργήσιμη γη και ανθρώπινη εργασία. Για τον λόγο αυτό, η μείωση της σπατάλης τροφίμων αποτελεί βασικό στόχο διεθνών οργανισμών και πολιτικών βιώσιμης ανάπτυξης. Ειδικότερα, η Ευρωπαϊκή Ένωση έχει θέσει ως στρατηγικό στόχο τη σημαντική μείωση της σπατάλης τροφίμων έως το 2030, στο πλαίσιο της στρατηγικής Farm to Fork και των Στόχων Βιώσιμης Ανάπτυξης των Ηνωμένων Εθνών (*From Farm to Fork - Consilium*, 2020).

Η αυξανόμενη πολυπλοκότητα των εφοδιαστικών αλυσίδων και η μεταβλητότητα της καταναλωτικής ζήτησης καθιστούν αναγκαία την ανάπτυξη προηγμένων συστημάτων διαχείρισης αποθεμάτων και δυναμικής λήψης αποφάσεων. Στο πλαίσιο αυτό, τεχνικές μηχανικής μάθησης και ενισχυτικής μάθησης αξιοποιούνται ολοένα και περισσότερο για την πρόβλεψη ζήτησης, τη βελτιστοποίηση αποθεμάτων και τη δυναμική προσαρμογή των τιμών, με στόχο τον περιορισμό της σπατάλης και τη βελτίωση της λειτουργικής αποδοτικότητας.

2.1.2. Αίτια σπατάλης τροφίμων στο λιανικό εμπόριο

Η σπατάλη τροφίμων στο λιανικό εμπόριο αποτελεί αποτέλεσμα πολλών αλληλεπιδρώντων παραγόντων που σχετίζονται με τη διαχείριση αποθεμάτων, την αβεβαιότητα της ζήτησης, τις στρατηγικές τιμολόγησης και τα χαρακτηριστικά των ευπαθών προϊόντων. Η πολυπλοκότητα των σύγχρονων εφοδιαστικών αλυσίδων καθιστά ιδιαίτερα δύσκολη τη διατήρηση ισορροπίας μεταξύ διαθεσιμότητας προϊόντων και περιορισμού των απορρίψεων, οδηγώντας συχνά σε αυξημένα επίπεδα σπατάλης.

Ένα από τα σημαντικότερα αίτια είναι η δημιουργία υπεραποθεμάτων (overstocking). Οι επιχειρήσεις λιανικού εμπορίου διατηρούν συχνά υψηλά επίπεδα αποθεμάτων προκειμένου να αποφύγουν ελλείψεις προϊόντων και απώλειες πωλήσεων. Ωστόσο, στα ευπαθή προϊόντα η πρακτική αυτή αυξάνει σημαντικά την πιθανότητα λήξης και αλλοίωσης προϊόντων πριν αυτά διατεθούν στους καταναλωτές (Blackburn & Scudder, 2009). Το πρόβλημα γίνεται ακόμη πιο έντονο σε περιόδους χαμηλής ή απρόβλεπτης ζήτησης.

Ιδιαίτερα σημαντικό ρόλο παίζουν επίσης τα σφάλματα πρόβλεψης ζήτησης (forecasting errors). Η ζήτηση στο λιανικό εμπόριο τροφίμων επηρεάζεται από πολλούς παράγοντες, όπως η εποχικότητα, οι καιρικές συνθήκες, οι προωθητικές ενέργειες και οι μεταβολές στις καταναλωτικές προτιμήσεις. Η αδυναμία ακριβούς πρόβλεψης της μελλοντικής ζήτησης οδηγεί είτε σε υπερβολικές παραγγελίες είτε σε ανεπαρκή επίπεδα αποθέματος, δημιουργώντας αντίστοιχα αυξημένη σπατάλη ή απώλειες πωλήσεων (Janiesch et al., 2021).

Παράλληλα, η περιορισμένη διάρκεια ζωής των ευπαθών προϊόντων (shelf life) αποτελεί κρίσιμο παράγοντα δημιουργίας αποβλήτων. Προϊόντα όπως φρούτα, λαχανικά, γαλακτοκομικά και προϊόντα κρέατος παρουσιάζουν σταδιακή υποβάθμιση της ποιότητάς τους με την πάροδο του χρόνου. Ως αποτέλεσμα, η οικονομική αξία των προϊόντων μειώνεται όσο πλησιάζει η ημερομηνία λήξης τους, αυξάνοντας την πιθανότητα απόρριψης πριν την πώληση (Nahmias, 1982).

Επιπλέον, η αβεβαιότητα της ζήτησης (demand uncertainty) δυσχεραίνει σημαντικά τη λήψη αποτελεσματικών αποφάσεων αποθεματοποίησης και τιμολόγησης. Σε πραγματικά περιβάλλοντα λιανικής, η συμπεριφορά των καταναλωτών μεταβάλλεται συνεχώς, ενώ η ζήτηση συχνά εμφανίζει στοχαστική και μη γραμμική συμπεριφορά. Η αδυναμία προσαρμογής των επιχειρήσεων στις μεταβολές αυτές οδηγεί σε μη αποδοτική κατανομή αποθεμάτων και αυξημένα επίπεδα σπατάλης (Kara & Dogan, 2018).

Σημαντικό αίτιο αποτελεί επίσης ο χρόνος και ο τρόπος εφαρμογής των markdown πολιτικών τιμολόγησης. Σε πολλές περιπτώσεις, οι μειώσεις τιμών πραγματοποιούνται πολύ αργά, όταν η υπολειπόμενη διάρκεια ζωής των προϊόντων είναι ήδη περιορισμένη. Ως αποτέλεσμα, η μείωση της τιμής δεν επαρκεί για να αυξήσει σημαντικά τη ζήτηση και μεγάλο μέρος των αποθεμάτων τελικά απορρίπτεται. Αντίθετα, έγκαιρες και δυναμικά προσαρμοζόμενες πολιτικές markdown μπορούν να μειώσουν σημαντικά τη σπατάλη

τροφίμων και να βελτιώσουν την αποδοτικότητα του συστήματος (Talluri & Van Ryzin, 2004).

Τέλος, η συμπεριφορά των καταναλωτών επηρεάζει επίσης σημαντικά τη δημιουργία σπατάλης στο λιανικό εμπόριο. Οι καταναλωτές συχνά προτιμούν προϊόντα με μεγαλύτερη υπολειπόμενη διάρκεια ζωής ή καλύτερη οπτική ποιότητα, ακόμη και όταν τα υπόλοιπα προϊόντα παραμένουν απολύτως ασφαλή προς κατανάλωση. Η πρακτική αυτή οδηγεί σε άνιση κατανάλωση αποθεμάτων και αυξημένες απορρίψεις προϊόντων μικρότερης διάρκειας ζωής.

Η αποτελεσματική αντιμετώπιση των παραπάνω προβλημάτων απαιτεί την ανάπτυξη δυναμικών και προσαρμοστικών συστημάτων λήψης αποφάσεων, τα οποία να συνδυάζουν πρόβλεψη ζήτησης, διαχείριση αποθεμάτων και δυναμική τιμολόγηση. Στο πλαίσιο αυτό, οι τεχνικές μηχανικής μάθησης και ενισχυτικής μάθησης παρέχουν τη δυνατότητα συνεχούς προσαρμογής των αποφάσεων στις πραγματικές συνθήκες της αγοράς, συμβάλλοντας στη μείωση της σπατάλης τροφίμων και στη βελτιστοποίηση της λειτουργικής αποδοτικότητας.

2.1.3. Ευπαθή προϊόντα και διαχείριση αποθεμάτων

Η διαχείριση αποθεμάτων ευπαθών προϊόντων αποτελεί μία από τις πιο απαιτητικές διαδικασίες στη διοίκηση εφοδιαστικής αλυσίδας και στο λιανικό εμπόριο τροφίμων. Τα ευπαθή προϊόντα χαρακτηρίζονται από περιορισμένη διάρκεια ζωής και σταδιακή υποβάθμιση της ποιότητάς τους με την πάροδο του χρόνου, γεγονός που διαφοροποιεί σημαντικά τη διαχείρισή τους σε σχέση με τα μη ευπαθή αγαθά (Nahmias, 1982). Η ιδιαιτερότητα αυτή δημιουργεί αυξημένη πολυπλοκότητα στη διαδικασία λήψης αποφάσεων σχετικά με την αποθεματοποίηση, την αναπλήρωση και την τιμολόγηση των προϊόντων.

Η έννοια της φθοράς (perishability) αναφέρεται στη σταδιακή μείωση της χρηστικής ή εμπορικής αξίας ενός προϊόντος λόγω του χρόνου, των συνθηκών αποθήκευσης ή φυσικών και βιολογικών διεργασιών. Προϊόντα όπως φρούτα, λαχανικά, γαλακτοκομικά και κρέας εμφανίζουν υψηλή ευαισθησία σε αλλοιώσεις και περιορισμένη χρονική περίοδο διάθεσης, γεγονός που αυξάνει τον κίνδυνο απόρριψης πριν από την πώλησή τους. Ως αποτέλεσμα,

οι επιχειρήσεις καλούνται να ισορροπήσουν μεταξύ υψηλής διαθεσιμότητας προϊόντων και περιορισμού των απωλειών λόγω λήξης (Blackburn & Scudder, 2009).

Κεντρικό ρόλο στη διαχείριση ευπαθών αποθεμάτων διαδραματίζει η ημερομηνία λήξης (expiration date) ή η υπολειπόμενη διάρκεια ζωής του προϊόντος. Καθώς ο διαθέσιμος χρόνος πώλησης μειώνεται, μειώνεται παράλληλα και η πιθανότητα διάθεσης του προϊόντος σε πλήρη τιμή. Για τον λόγο αυτό, οι επιχειρήσεις εφαρμόζουν συχνά πολιτικές προσαρμογής τιμών ή markdown pricing ώστε να αυξήσουν τη ζήτηση προϊόντων που πλησιάζουν στη λήξη τους και να περιορίσουν τις απορρίψεις.

Στη βιβλιογραφία έχουν αναπτυχθεί διάφορες στρατηγικές διαχείρισης αποθεμάτων για ευπαθή προϊόντα. Μία από τις πιο διαδεδομένες είναι η πολιτική FIFO (First In – First Out), σύμφωνα με την οποία τα προϊόντα που εισέρχονται πρώτα στην αποθήκη διατίθενται πρώτα προς πώληση. Η προσέγγιση αυτή επιδιώκει τη μείωση του χρόνου παραμονής των προϊόντων στην αποθήκη και τον περιορισμό των ληγμένων αποθεμάτων. Ωστόσο, σε περιβάλλοντα με διαφορετικές ημερομηνίες λήξης, η πολιτική FEFO (First Expired – First Out) θεωρείται συχνά αποτελεσματικότερη, καθώς προτεραιοποιεί τη διάθεση των προϊόντων με τη μικρότερη υπολειπόμενη διάρκεια ζωής (Akkerman et al., 2010).

Παράλληλα, η διαχείριση ευπαθών αποθεμάτων συνδέεται άμεσα με το κόστος αλλοίωσης και απόρριψης προϊόντων (spoilage cost). Το κόστος αυτό περιλαμβάνει όχι μόνο την απώλεια πιθανών εσόδων από μη πωληθέντα προϊόντα αλλά και πρόσθετα λειτουργικά έξοδα που σχετίζονται με τη μεταφορά, αποθήκευση και διαχείριση αποβλήτων. Σε πολλές περιπτώσεις, το συνολικό κόστος σπατάλης μπορεί να επηρεάσει σημαντικά την κερδοφορία των επιχειρήσεων λιανικού εμπορίου (Nahmias, 1982).

Η αποτελεσματική διαχείριση ευπαθών αποθεμάτων απαιτεί επομένως τη λήψη δυναμικών αποφάσεων που λαμβάνουν υπόψη πολλαπλές μεταβλητές, όπως η ζήτηση, η διάρκεια ζωής των προϊόντων, τα επίπεδα αποθέματος και οι στρατηγικές τιμολόγησης. Παραδοσιακές στατικές μέθοδοι διαχείρισης συχνά αδυνατούν να ανταποκριθούν στην πολυπλοκότητα και τη μεταβλητότητα των πραγματικών συνθηκών της αγοράς. Για τον λόγο αυτό, τα τελευταία χρόνια αυξάνεται σημαντικά το ενδιαφέρον για τη χρήση τεχνικών μηχανικής μάθησης και ενισχυτικής μάθησης, οι οποίες επιτρέπουν την ανάπτυξη προσαρμοστικών πολιτικών διαχείρισης αποθεμάτων και δυναμικής τιμολόγησης σε πραγματικό χρόνο (Akkerman et al., 2010; Kara & Dogan, 2018; Sutton & Barto, 2018).

Στο πλαίσιο της παρούσας εργασίας, η διαχείριση ευπαθών αποθεμάτων αποτελεί βασικό στοιχείο του προβλήματος βελτιστοποίησης, καθώς οι αποφάσεις τιμολόγησης επηρεάζουν άμεσα τόσο την κερδοφορία όσο και τα επίπεδα σπατάλης τροφίμων. Η ενσωμάτωση πληροφοριών σχετικών με το επίπεδο αποθέματος και την υπολειπόμενη διάρκεια ζωής των προϊόντων στο περιβάλλον ενισχυτικής μάθησης επιτρέπει στον πράκτορα (agent) της ενισχυτικής μάθησης (Reinforcement Learning) να προσαρμόζει δυναμικά τις αποφάσεις του στις πραγματικές συνθήκες λειτουργίας.

2.1.4. Οικονομικές και περιβαλλοντικές επιπτώσεις της σπατάλης τροφίμων

Η σπατάλη τροφίμων αποτελεί σημαντικό πρόβλημα τόσο για τις επιχειρήσεις όσο και για την κοινωνία συνολικά, καθώς συνδέεται με σοβαρές οικονομικές, περιβαλλοντικές και κοινωνικές επιπτώσεις. Στον τομέα του λιανικού εμπορίου τροφίμων, οι απορρίψεις προϊόντων οδηγούν σε άμεσες οικονομικές απώλειες λόγω μη πωληθέντων αποθεμάτων, ενώ παράλληλα αυξάνουν το λειτουργικό κόστος που σχετίζεται με την αποθήκευση, τη μεταφορά και τη διαχείριση αποβλήτων (FAO, 2019).

Οι οικονομικές επιπτώσεις της σπατάλης είναι ιδιαίτερα έντονες στα ευπαθή προϊόντα, όπου η υποβάθμιση της ποιότητας και η περιορισμένη διάρκεια ζωής αυξάνουν σημαντικά τον κίνδυνο απόρριψης πριν από την πώληση. Οι επιχειρήσεις συχνά καλούνται να ισορροπήσουν μεταξύ υψηλής διαθεσιμότητας προϊόντων και περιορισμού των απωλειών λόγω λήξης, γεγονός που καθιστά τη διαχείριση αποθεμάτων κρίσιμο παράγοντα κερδοφορίας. Επιπλέον, η σπατάλη επηρεάζει αρνητικά την αποδοτικότητα της εφοδιαστικής αλυσίδας, καθώς πόροι που επενδύθηκαν στην παραγωγή, αποθήκευση και διανομή προϊόντων τελικά δεν αξιοποιούνται εμπορικά (Blackburn & Scudder, 2009).

Παράλληλα, οι περιβαλλοντικές επιπτώσεις της σπατάλης τροφίμων είναι ιδιαίτερα σημαντικές. Η παραγωγή τροφίμων απαιτεί μεγάλες ποσότητες φυσικών πόρων, όπως νερό, ενέργεια, καλλιεργήσιμη γη και πρώτες ύλες. Όταν τα τρόφιμα απορρίπτονται χωρίς να καταναλωθούν, ουσιαστικά σπαταλώνονται και οι πόροι που χρησιμοποιήθηκαν για την παραγωγή τους. Σύμφωνα με τον FAO, η σπατάλη τροφίμων συμβάλλει σημαντικά στις παγκόσμιες εκπομπές αερίων του θερμοκηπίου, καθώς τα τρόφιμα που απορρίπτονται

οδηγούνται συχνά σε χώρους υγειονομικής ταφής, όπου παράγονται εκπομπές μεθανίου κατά τη διαδικασία αποσύνθεσης (FAO, 2019).

Επιπλέον, η μείωση της σπατάλης τροφίμων αποτελεί βασικό στόχο των διεθνών πολιτικών βιωσιμότητας και κυκλικής οικονομίας. Η Ευρωπαϊκή Ένωση, μέσω της στρατηγικής Farm to Fork, επιδιώκει τη σημαντική μείωση της σπατάλης τροφίμων έως το 2030, προωθώντας πιο βιώσιμα πρότυπα παραγωγής και κατανάλωσης (*From Farm to Fork - Consilium*, 2020). Παρόμοια, οι Στόχοι Βιώσιμης Ανάπτυξης των Ηνωμένων Εθνών (Sustainable Development Goals – SDGs) περιλαμβάνουν συγκεκριμένους στόχους για τη μείωση των αποβλήτων τροφίμων σε επίπεδο λιανικού εμπορίου και καταναλωτή, με χαρακτηριστικό παράδειγμα τον στόχο SDG 12.3, ο οποίος επιδιώκει τη μείωση κατά το ήμισυ της κατά κεφαλήν σπατάλης τροφίμων έως το 2030

(Transforming Our World: The 2030 Agenda for Sustainable Development | Department of Economic and Social Affairs, 2015).

Στο πλαίσιο αυτό, η αξιοποίηση προηγμένων τεχνολογιών ανάλυσης δεδομένων, μηχανικής μάθησης και δυναμικής τιμολόγησης αποκτά ιδιαίτερη σημασία. Μέσω της βελτιστοποίησης των αποφάσεων αποθεματοποίησης και τιμολόγησης, οι επιχειρήσεις μπορούν να μειώσουν σημαντικά τα επίπεδα απορρίψεων και να βελτιώσουν τη συνολική αποδοτικότητα των λειτουργιών τους. Ειδικότερα, η εφαρμογή τεχνικών ενισχυτικής μάθησης επιτρέπει την ανάπτυξη προσαρμοστικών πολιτικών που λαμβάνουν υπόψη τη μεταβλητότητα της ζήτησης και τη διάρκεια ζωής των προϊόντων, συμβάλλοντας ταυτόχρονα στη μεγιστοποίηση της κερδοφορίας και στη μείωση του περιβαλλοντικού αποτυπώματος.

Η αποτελεσματική αντιμετώπιση της σπατάλης τροφίμων απαιτεί επομένως συνδυασμό τεχνολογικών, επιχειρησιακών και στρατηγικών παρεμβάσεων. Η ανάπτυξη έξυπνων συστημάτων λήψης αποφάσεων μπορεί να συμβάλει ουσιαστικά στη δημιουργία πιο βιώσιμων και αποδοτικών εφοδιαστικών αλυσίδων τροφίμων, μειώνοντας τόσο τις οικονομικές απώλειες όσο και τις περιβαλλοντικές επιπτώσεις.

2.2. Μηχανική μάθηση

2.2.1. Ορισμός και ιστορική εξέλιξη

Η μηχανική μάθηση (Machine Learning – ML) είναι ένα υποπεδίο της τεχνητής νοημοσύνης και μπορεί να οριστεί ευρέως ως υπολογιστικές μέθοδοι που χρησιμοποιούν την εμπειρία για τη βελτίωση της απόδοσης ή για την πραγματοποίηση ακριβών προβλέψεων (Mohri et al., 2018). Ως εμπειρία νοούνται τα δεδομένα που χρησιμοποιούνται για την εκπαίδευση του μοντέλου. Ο όγκος αυτών των δεδομένων είναι καίριας σημασίας για τα αποτελέσματα της διαδικασίας, καθώς απαιτούνται μεγάλες ποσότητες δεδομένων για να λειτουργήσει αξιόπιστα η διαδικασία. Ένα πρόγραμμα μαθαίνει όταν η απόδοση του σε μια συγκεκριμένη διαδικασία βελτιώνεται όσο αυξάνεται η εμπειρία του.

Η έννοια της μηχανικής μάθησης εμφανίστηκε στα τέλη της δεκαετίας του 1950. Ο Arthur Samuel χρησιμοποίησε για πρώτη φορά τον όρο το 1959, περιγράφοντας ένα πρόγραμμα ντάμας το οποίο μπορούσε να βελτιώνει την απόδοσή του παίζοντας επαναλαμβανόμενα παιχνίδια. Το πρόγραμμα αυτό θεωρείται μία από τις πρώτες επιτυχημένες εφαρμογές αυτοματοποιημένης μάθησης σε υπολογιστή.

Κατά τις επόμενες δεκαετίες, η πρόοδος στη στατιστική, η αύξηση της υπολογιστικής ισχύος και η διαθεσιμότητα μεγάλων συνόλων δεδομένων συνέβαλαν στην ταχεία εξέλιξη του πεδίου. Από τη δεκαετία του 1990 και μετά, η μηχανική μάθηση άρχισε να χρησιμοποιείται ευρέως σε εφαρμογές όπως η αναγνώριση προτύπων, η πρόβλεψη χρονοσειρών, η ιατρική διάγνωση και η ανάλυση επιχειρησιακών δεδομένων. Τα τελευταία χρόνια, η ανάπτυξη της βαθιάς μάθησης (deep learning) και των νευρωνικών δικτύων έχει οδηγήσει σε σημαντικές βελτιώσεις στην ακρίβεια των προβλέψεων και στην ικανότητα επεξεργασίας πολύπλοκων δεδομένων (Janiesch et al., 2021).

2.2.2. Βασικές κατηγορίες μηχανικής μάθησης

Η μηχανική μάθηση διακρίνεται συνήθως σε τρεις βασικές κατηγορίες: την επιβλεπόμενη μάθηση (supervised learning), τη μη επιβλεπόμενη μάθηση (unsupervised learning) και την ενισχυτική μάθηση (reinforcement learning).

Στην επιβλεπόμενη μάθηση, ο αλγόριθμος εκπαιδεύεται χρησιμοποιώντας δεδομένα για τα οποία είναι γνωστή η σωστή έξοδος. Στόχος είναι η εκμάθηση της σχέσης μεταξύ των μεταβλητών εισόδου και της επιθυμητής εξόδου, ώστε το μοντέλο να μπορεί να πραγματοποιεί προβλέψεις σε νέα δεδομένα. Στην κατηγορία αυτή ανήκουν μέθοδοι όπως

η γραμμική παλινδρόμηση, η λογιστική παλινδρόμηση, τα δέντρα απόφασης και τα νευρωνικά δίκτυα.(Awad & Khanna, 2015)

Στη μη επιβλεπόμενη μάθηση, αντίθετα, δεν υπάρχουν προκαθορισμένες ετικέτες ή γνωστές απαντήσεις. Ο στόχος είναι η αναγνώριση κρυφών προτύπων και σχέσεων μέσα στα δεδομένα. Οι πιο συνηθισμένες εφαρμογές περιλαμβάνουν την ομαδοποίηση δεδομένων (clustering), όπως ο αλγόριθμος k-means, καθώς και τις τεχνικές μείωσης διαστάσεων, όπως η Ανάλυση Κυρίων Συνιστωσών (Principal Component Analysis – PCA).

Η τρίτη κατηγορία είναι η ενισχυτική μάθηση που θα μας απασχολήσει ιδιαίτερα σε αυτή την εργασία. Στην ενισχυτική μάθηση ένας «πράκτορας» (agent) αλληλεπιδρά με ένα περιβάλλον και λαμβάνει αποφάσεις με βάση ένα σύστημα ανταμοιβών και ποινών. Ο πράκτορας επιδιώκει να μάθει ποια ακολουθία ενεργειών οδηγεί στη μέγιστη συνολική ανταμοιβή. Η προσέγγιση αυτή είναι ιδιαίτερα κατάλληλη για προβλήματα όπου οι αποφάσεις λαμβάνονται διαδοχικά και επηρεάζουν το μέλλον του συστήματος, όπως η διαχείριση αποθεμάτων και η δυναμική τιμολόγηση.

2.2.3. Η σημασία της μηχανικής μάθησης στο λιανικό εμπόριο τροφίμων

Η χρήση τεχνικών μηχανικής μάθησης στο λιανικό εμπόριο τροφίμων έχει αυξηθεί σημαντικά, κυρίως λόγω της ανάγκης για ακριβέστερη πρόβλεψη της ζήτησης και αποτελεσματικότερη διαχείριση αποθεμάτων. Η ζήτηση για ευπαθή προϊόντα επηρεάζεται από μεγάλο αριθμό παραγόντων, όπως η εποχικότητα, οι αργίες, οι προσφορές και οι μεταβολές στις καταναλωτικές προτιμήσεις. Οι παραδοσιακές στατιστικές μέθοδοι συχνά δυσκολεύονται να αποτυπώσουν αυτές τις σύνθετες και μη γραμμικές σχέσεις, ενώ τα μοντέλα μηχανικής μάθησης μπορούν να τις αναγνωρίσουν με μεγαλύτερη ακρίβεια (Janiesch et al., 2021).

Στο πλαίσιο της πρόβλεψης ζήτησης, χρησιμοποιούνται συχνά τεχνικές επιβλεπόμενης μάθησης και νευρωνικά δίκτυα, καθώς μπορούν να αξιοποιήσουν ιστορικά δεδομένα πωλήσεων, ημερολογιακά γεγονότα και χαρακτηριστικά προϊόντων. Η δυνατότητα αυτή είναι ιδιαίτερα σημαντική για προϊόντα μικρής διάρκειας ζωής, όπως φρούτα, λαχανικά,

γαλακτοκομικά και κρέας, όπου η λανθασμένη πρόβλεψη μπορεί να οδηγήσει είτε σε ελλείψεις είτε σε αυξημένη σπατάλη.

Παράλληλα, η ενισχυτική μάθηση μπορεί να χρησιμοποιηθεί για τη λήψη δυναμικών αποφάσεων τιμολόγησης και αναπλήρωσης αποθεμάτων. Σε αυτήν την περίπτωση, ο αλγόριθμος μαθαίνει σταδιακά ποιες ενέργειες οδηγούν στη μέγιστη κερδοφορία και στη μικρότερη δυνατή σπατάλη, λαμβάνοντας υπόψη τόσο το διαθέσιμο απόθεμα όσο και τον χρόνο που απομένει μέχρι τη λήξη των προϊόντων. Η προσέγγιση αυτή συνδέεται άμεσα με το αντικείμενο της παρούσας εργασίας, καθώς επιτρέπει τη δημιουργία πολιτικών δυναμικής τιμολόγησης που προσαρμόζονται συνεχώς στις συνθήκες της αγοράς.

2.2.4. Ενισχυτική μάθηση (Reinforcement learning)

Η ενισχυτική μάθηση (Reinforcement Learning – RL) αποτελεί μία από τις βασικές κατηγορίες της μηχανικής μάθησης και χρησιμοποιείται για την επίλυση προβλημάτων διαδοχικής λήψης αποφάσεων υπό συνθήκες αβεβαιότητας (Sutton & Barto, 2018). Στην ενισχυτική μάθηση, ένας πράκτορας (agent) αλληλεπιδρά με ένα περιβάλλον (environment), επιλέγοντας ενέργειες με στόχο τη μεγιστοποίηση μιας αθροιστικής ανταμοιβής (reward) σε βάθος χρόνου. Σε αντίθεση με την επιβλεπόμενη μάθηση, όπου το μοντέλο εκπαιδεύεται σε δεδομένα με γνωστές εξόδους, στην ενισχυτική μάθηση ο πράκτορας μαθαίνει μέσω δοκιμής και σφάλματος, αξιοποιώντας την ανατροφοδότηση που λαμβάνει από το περιβάλλον μετά από κάθε ενέργεια (Kaelbling et al., 1996).

Η θεωρητική βάση της ενισχυτικής μάθησης στηρίζεται στις Μαρκοβιανές Διαδικασίες Απόφασης (Markov Decision Processes – MDPs), οι οποίες περιγράφουν συστήματα στα οποία η επόμενη κατάσταση εξαρτάται αποκλειστικά από την τρέχουσα κατάσταση και την επιλεγμένη ενέργεια (Puterman, 2014). Ένα MDP ορίζεται από το σύνολο των καταστάσεων, το σύνολο των ενεργειών, τη συνάρτηση ανταμοιβής και τις πιθανότητες μετάβασης μεταξύ καταστάσεων. Στόχος του πράκτορα είναι η εκμάθηση μιας πολιτικής (policy), δηλαδή ενός μηχανισμού επιλογής ενεργειών που μεγιστοποιεί τη μακροχρόνια αναμενόμενη ανταμοιβή.

Η διαδικασία μάθησης πραγματοποιείται επαναληπτικά. Ο πράκτορας παρατηρεί την τρέχουσα κατάσταση του περιβάλλοντος, επιλέγει μία ενέργεια και στη συνέχεια λαμβάνει

ανταμοιβή και μεταβαίνει σε νέα κατάσταση. Μέσω διαδοχικών επαναλήψεων, ο αλγόριθμος προσαρμόζει σταδιακά τη συμπεριφορά του, επιδιώκοντας τη βελτιστοποίηση της συνολικής απόδοσης του συστήματος.

Ένα από τα σημαντικότερα ζητήματα στην ενισχυτική μάθηση είναι η ισορροπία μεταξύ εξερεύνησης (exploration) και εκμετάλλευσης (exploitation). Η εξερεύνηση αφορά τη δοκιμή νέων ενεργειών ώστε να εντοπιστούν πιθανώς αποδοτικότερες στρατηγικές, ενώ η εκμετάλλευση αφορά τη χρήση των ήδη γνωστών ενεργειών που οδηγούν σε υψηλή ανταμοιβή. Η αποτελεσματική ισορροπία μεταξύ των δύο αυτών μηχανισμών είναι κρίσιμη για τη σύγκλιση του αλγορίθμου σε βέλτιστες πολιτικές λήψης αποφάσεων (Sutton & Barto, 2018).

Στη βιβλιογραφία έχουν αναπτυχθεί διάφορες προσεγγίσεις ενισχυτικής μάθησης, όπως οι Q-Learning και SARSA, τα Deep Q-Networks (DQN), καθώς και μέθοδοι βασισμένες σε πολιτικές, όπως οι Policy Gradient Methods και οι Actor-Critic αλγόριθμοι. Οι μέθοδοι αυτές διαφοροποιούνται κυρίως ως προς τον τρόπο εκμάθησης της πολιτικής και την ικανότητά τους να διαχειρίζονται πολύπλοκους χώρους κατάστασης και ενεργειών (Mnih et al., 2015; Sutton & Barto, 2018).

Στο πλαίσιο της παρούσας εργασίας, η ενισχυτική μάθηση αξιοποιείται ως βασικός μηχανισμός λήψης αποφάσεων για τη δυναμική τιμολόγηση και τη διαχείριση αποθεμάτων ευπαθών προϊόντων. Η δυνατότητα του αλγορίθμου να μαθαίνει από την αλληλεπίδραση με το περιβάλλον και να προσαρμόζει τις αποφάσεις του σε πραγματικό χρόνο καθιστά την προσέγγιση αυτή ιδιαίτερα κατάλληλη για σύνθετα προβλήματα λιανικού εμπορίου με υψηλή αβεβαιότητα ζήτησης και περιορισμένη διάρκεια ζωής προϊόντων (W. Chen et al., 2018; Kara & Dogan, 2018).

2.2.5. Βασικές προσεγγίσεις αλγορίθμων ενισχυτικής μάθησης

Στη βιβλιογραφία έχουν αναπτυχθεί διαφορετικές προσεγγίσεις αλγορίθμων ενισχυτικής μάθησης, οι οποίες διαφοροποιούνται κυρίως ως προς τον τρόπο με τον οποίο εκτιμούν τη βέλτιστη πολιτική λήψης αποφάσεων, τον τρόπο αναπαράστασης της γνώσης και την ικανότητά τους να διαχειρίζονται πολύπλοκους χώρους καταστάσεων και ενεργειών. Οι

σημαντικότερες κατηγορίες περιλαμβάνουν τις value-based μεθόδους, τις policy-based μεθόδους και τις υβριδικές προσεγγίσεις Actor-Critic.

Μία από τις πιο γνωστές value-based μεθόδους είναι ο αλγόριθμος Q-Learning, ο οποίος προτάθηκε από τον (C. Watkins, 1989). Ο αλγόριθμος βασίζεται στην εκτίμηση της συνάρτησης αξίας Q, η οποία αποτυπώνει τη συνολική αναμενόμενη ανταμοιβή που συνδέεται με μία συγκεκριμένη ενέργεια σε μία δεδομένη κατάσταση. Το Q-Learning αποτελεί off-policy μέθοδο, καθώς ο πράκτορας μαθαίνει τη βέλτιστη πολιτική ανεξάρτητα από την πολιτική εξερεύνησης που ακολουθεί κατά την εκπαίδευση. Η μέθοδος αυτή χαρακτηρίζεται από σχετικά απλή υλοποίηση, χαμηλές υπολογιστικές απαιτήσεις και ικανοποιητική σταθερότητα σε προβλήματα με περιορισμένο και διακριτό χώρο καταστάσεων. Ωστόσο, η απόδοσή της περιορίζεται σημαντικά όταν ο χώρος καταστάσεων ή ενεργειών γίνεται πολύ μεγάλος, καθώς απαιτείται η αποθήκευση και συνεχής ενημέρωση μεγάλου πίνακα Q-values (Sutton & Barto, 2018).

Παρόμοια φιλοσοφία ακολουθεί και ο αλγόριθμος SARSA (State–Action–Reward–State–Action), ο οποίος επίσης ανήκει στις value-based μεθόδους. Η βασική διαφορά του σε σχέση με το Q-Learning είναι ότι πρόκειται για on-policy αλγόριθμο, καθώς η εκμάθηση πραγματοποιείται με βάση τις πραγματικές ενέργειες που επιλέγονται από τον πράκτορα κατά τη διάρκεια της εξερεύνησης. Ως αποτέλεσμα, ο SARSA λαμβάνει υπόψη τον κίνδυνο των εξερευνητικών ενεργειών και συχνά παρουσιάζει πιο συντηρητική συμπεριφορά κατά την εκπαίδευση. Η προσέγγιση αυτή μπορεί να οδηγήσει σε μεγαλύτερη σταθερότητα σε ορισμένα περιβάλλοντα, αν και συνήθως συγκλίνει πιο αργά σε σχέση με το Q-Learning (Rummery & Niranjan, 1994).

Η ανάπτυξη της βαθιάς μάθησης οδήγησε στη δημιουργία των Deep Q-Networks (DQN), τα οποία επεκτείνουν το Q-Learning μέσω της χρήσης βαθιών νευρωνικών δικτύων για την προσέγγιση της συνάρτησης αξίας. Στα DQN, η γνώση δεν αποθηκεύεται σε πίνακα αλλά αναπαρίσταται μέσω παραμέτρων νευρωνικού δικτύου, επιτρέποντας έτσι τη διαχείριση πολύ μεγάλων και πολύπλοκων χώρων κατάστασης. Η μέθοδος αυτή εφαρμόστηκε με ιδιαίτερη επιτυχία σε περιβάλλοντα όπως τα Atari Games, η ρομποτική και τα αυτόνομα συστήματα ελέγχου (Mnih et al., 2015). Παρά τα σημαντικά πλεονεκτήματά τους, τα DQN χαρακτηρίζονται από υψηλό υπολογιστικό κόστος, αυξημένη πολυπλοκότητα εκπαίδευσης

και πιθανή αστάθεια κατά τη διαδικασία μάθησης, ιδιαίτερα όταν τα δεδομένα παρουσιάζουν υψηλή μεταβλητότητα.

Μια διαφορετική κατηγορία αλγορίθμων αποτελούν οι μέθοδοι Policy Gradient, οι οποίες δεν βασίζονται στην εκτίμηση συνάρτησης αξίας αλλά μαθαίνουν απευθείας την πολιτική επιλογής ενεργειών. Οι μέθοδοι αυτές είναι ιδιαίτερα κατάλληλες για προβλήματα με συνεχείς χώρους ενεργειών και σύνθετα δυναμικά περιβάλλοντα, όπου οι value-based προσεγγίσεις δυσκολεύονται να εφαρμοστούν αποτελεσματικά. Επιπλέον, επιτρέπουν τη μοντελοποίηση στοχαστικών πολιτικών, γεγονός που τις καθιστά χρήσιμες σε προβλήματα ελέγχου και ρομποτικής. Ωστόσο, οι αλγόριθμοι αυτοί παρουσιάζουν συχνά υψηλή διακύμανση κατά την εκπαίδευση και απαιτούν μεγάλο αριθμό επαναλήψεων για τη σύγκλιση του μοντέλου (Sutton et al., 1999).

Τέλος, ιδιαίτερο ενδιαφέρον παρουσιάζουν οι Actor-Critic μέθοδοι, οι οποίες συνδυάζουν χαρακτηριστικά τόσο των «value-based» όσο και των «policy-based» προσεγγίσεων. Στις μεθόδους αυτές, το τμήμα Actor είναι υπεύθυνο για την επιλογή ενεργειών μέσω της πολιτικής, ενώ το τμήμα Critic αξιολογεί την ποιότητα των αποφάσεων εκτιμώντας τη συνάρτηση αξίας. Η υβριδική αυτή αρχιτεκτονική επιτρέπει μεγαλύτερη σταθερότητα και ταχύτερη μάθηση σε πολύπλοκα περιβάλλοντα, ενώ χρησιμοποιείται ευρέως σε σύγχρονες εφαρμογές βαθιάς ενισχυτικής μάθησης, όπως ο έλεγχος ρομποτικών συστημάτων και η δυναμική βελτιστοποίηση (Konda & Tsitsiklis, 1999). Παράλληλα, η μεγαλύτερη αρχιτεκτονική πολυπλοκότητα και οι αυξημένες απαιτήσεις παραμετροποίησης καθιστούν τις μεθόδους αυτές δυσκολότερες στην υλοποίηση και στη σταθερή εκπαίδευση.

Συνολικά, η επιλογή του κατάλληλου αλγορίθμου ενισχυτικής μάθησης εξαρτάται από τα χαρακτηριστικά του προβλήματος, το μέγεθος του χώρου καταστάσεων, τη διαθεσιμότητα δεδομένων και τις υπολογιστικές απαιτήσεις της εφαρμογής. Σε προβλήματα μικρότερης πολυπλοκότητας και διακριτών καταστάσεων, οι value-based μέθοδοι, όπως το Q-Learning, συχνά αποτελούν αποτελεσματική και υπολογιστικά αποδοτική λύση, ενώ σε περιβάλλοντα υψηλής πολυπλοκότητας προτιμώνται συνήθως μέθοδοι βαθιάς ενισχυτικής μάθησης και υβριδικές προσεγγίσεις.

2.2.6. Η επιλογή του αλγορίθμου Q – Learning

Στην παρούσα εργασία επιλέχθηκε η χρήση του αλγορίθμου Q-Learning για την ανάπτυξη της πολιτικής δυναμικής τιμολόγησης και διαχείρισης αποθεμάτων. Η επιλογή αυτή βασίστηκε τόσο στα χαρακτηριστικά του προβλήματος όσο και στα πλεονεκτήματα που παρουσιάζει ο συγκεκριμένος αλγόριθμος σε περιβάλλοντα διαδοχικής λήψης αποφάσεων με διακριτό χώρο καταστάσεων και ενεργειών.

Το Q-Learning αποτελεί μία από τις πλέον διαδεδομένες μεθόδους ενισχυτικής μάθησης χωρίς μοντέλο (model-free reinforcement learning), καθώς επιτρέπει την εκμάθηση βέλτιστης πολιτικής μέσω αλληλεπίδρασης με το περιβάλλον, χωρίς να απαιτείται εκ των προτέρων γνώση των πιθανοτήτων μετάβασης ή της κατανομής της ζήτησης (C. Watkins, 1989). Η ιδιότητα αυτή είναι ιδιαίτερα σημαντική στο λιανικό εμπόριο τροφίμων, όπου η ζήτηση χαρακτηρίζεται από υψηλή αβεβαιότητα και μεταβλητότητα.

Επιπλέον, το πρόβλημα που εξετάζεται στην παρούσα εργασία μπορεί να αναπαρασταθεί με σχετικά περιορισμένο και διακριτό χώρο καταστάσεων, ο οποίος περιλαμβάνει μεταβλητές όπως το επίπεδο αποθέματος, οι ημέρες μέχρι τη λήξη του προϊόντος και οι διαθέσιμες τιμές πώλησης. Στα προβλήματα αυτού του τύπου, το Q-Learning παρουσιάζει ικανοποιητική απόδοση χωρίς να απαιτείται η χρήση πολύπλοκων νευρωνικών δικτύων ή υψηλής υπολογιστικής ισχύος, όπως συμβαίνει σε μεθόδους βαθιάς ενισχυτικής μάθησης (Deep Reinforcement Learning).

Ένας επιπλέον λόγος επιλογής του Q-Learning είναι η αυξημένη ερμηνευσιμότητα του μοντέλου. Η δυνατότητα αποθήκευσης και ανάλυσης των Q-values επιτρέπει την καλύτερη κατανόηση της διαδικασίας λήψης αποφάσεων του πράκτορα, γεγονός ιδιαίτερα σημαντικό σε εφαρμογές διαχείρισης αποθεμάτων και τιμολόγησης, όπου η διαφάνεια των αποφάσεων αποτελεί κρίσιμο παράγοντα αξιολόγησης.

Παράλληλα, σε σύγκριση με πιο σύνθετες μεθόδους βαθιάς ενισχυτικής μάθησης, όπως τα Deep Q-Networks (DQN) και οι Actor-Critic αλγόριθμοι, το Q-Learning παρουσιάζει μικρότερη υπολογιστική πολυπλοκότητα και μεγαλύτερη σταθερότητα σε περιβάλλοντα με περιορισμένο και διακριτό χώρο καταστάσεων, καθώς δεν απαιτεί την εκπαίδευση βαθιών νευρωνικών δικτύων (Sutton & Barto, 2018). Η χρήση βαθιών νευρωνικών δικτύων κρίθηκε ότι θα αύξανε σημαντικά την πολυπλοκότητα της υλοποίησης χωρίς αντίστοιχα σημαντική βελτίωση της απόδοσης, δεδομένου του μεγέθους και της δομής των διαθέσιμων δεδομένων.

Τέλος, η επιλογή του Q-Learning συνάδει με τη βιβλιογραφία που αφορά εφαρμογές δυναμικής τιμολόγησης και διαχείρισης ευπαθών αποθεμάτων, όπου ο αλγόριθμος έχει χρησιμοποιηθεί επιτυχώς για την εκμάθηση πολιτικών markdown και αναπλήρωσης υπό αβεβαιότητα ζήτησης (W. Chen et al., 2018).

2.3. Μοντέλα πρόβλεψης

2.3.1. Ορισμός, σκοπός και ιστορική αναδρομή των μοντέλων πρόβλεψης

Τα μοντέλα πρόβλεψης αποτελούν μεθόδους ανάλυσης που χρησιμοποιούνται για την εκτίμηση μελλοντικών τιμών ή γεγονότων με βάση ιστορικά δεδομένα. Στόχος τους είναι να εντοπίσουν πρότυπα, τάσεις και σχέσεις μεταξύ μεταβλητών, ώστε να μπορούν να χρησιμοποιηθούν για τη λήψη αποφάσεων.

Η χρήση τους είναι ιδιαίτερα διαδεδομένη σε τομείς όπως η οικονομία, η ιατρική, η ενέργεια, οι μεταφορές και η επιστήμη δεδομένων. Για παράδειγμα, μπορούν να χρησιμοποιηθούν για την πρόβλεψη της ζήτησης ηλεκτρικής ενέργειας, της πορείας μιας ασθένειας ή της εξέλιξης των πωλήσεων μιας επιχείρησης. Με την πάροδο των χρόνων και την εξέλιξη της επιστήμης παρατηρείται ότι αναπτύσσονται μοντέλα πρόβλεψης σε κάθε πιθανό κλάδο που έχει να κάνει με την λήψη αποφάσεων.

Η εξέλιξη των μοντέλων πρόβλεψης ξεκίνησε στις αρχές του 20ού αιώνα με απλές στατιστικές μεθόδους, όπως ο κινούμενος μέσος όρος (moving average), ο οποίος χρησιμοποιήθηκε για την εξομάλυνση των διακυμάνσεων των δεδομένων και για να αποτυπώσει τη γενική τάση. Στη συνέχεια, τη δεκαετία του 1950 αναπτύχθηκε η εκθετική εξομάλυνση (exponential smoothing) από τους Brown και Holt (Holt, 2004), ενώ λίγο αργότερα το μοντέλο Holt–Winters επέτρεψε την ενσωμάτωση τάσης και εποχικότητας. (Winters, 1960)

Κατά τη δεκαετία του 1970, τα μοντέλα ARIMA των Box και Jenkins κυριάρχησαν στην ανάλυση χρονοσειρών, καθώς μπορούσαν να περιγράψουν πιο σύνθετα πρότυπα ζήτησης (Box et al., 2016). Ωστόσο, η αυξανόμενη πολυπλοκότητα των αγορών και η ανάγκη για

μεγαλύτερη ακρίβεια οδήγησαν, από τη δεκαετία του 1990 και μετά, στην ανάπτυξη μοντέλων μηχανικής μάθησης, όπως τα Support Vector Machines και τα Random Forest (Breiman, 2001; Vapnik, 1995). Τα τελευταία χρόνια με την εξέλιξη της υπολογιστικής ισχύος αλλά και χάρη στην αποθήκευση μεγάλων όγκων δεδομένων, ιδιαίτερη σημασία έχουν αποκτήσει τα νευρωνικά δίκτυα και ειδικά τα μοντέλα LSTM, τα οποία είναι κατάλληλα για δεδομένα με έντονη μεταβλητότητα και εποχικότητα (Hochreiter & Schmidhuber, 1997).

2.3.2. Κατηγορίες μοντέλων πρόβλεψης

Τα μοντέλα πρόβλεψης χωρίζονται συνήθως σε τρεις βασικές κατηγορίες: τα στατιστικά μοντέλα, τα μοντέλα μηχανικής μάθησης και τα υβριδικά μοντέλα.

Τα στατιστικά μοντέλα βασίζονται σε μαθηματικές σχέσεις μεταξύ των μεταβλητών και χρησιμοποιούνται κυρίως όταν οι σχέσεις αυτές μπορούν να περιγραφούν με σαφή τρόπο. Τα πιο γνωστά στατιστικά μοντέλα είναι η γραμμική παλινδρόμηση (linear regression), η λογιστική παλινδρόμηση (logistic regression) και τα μοντέλα χρονοσειρών (time series models). Η γραμμική παλινδρόμηση χρησιμοποιείται όταν στόχος είναι η πρόβλεψη μιας συνεχούς μεταβλητής, ενώ η λογιστική παλινδρόμηση εφαρμόζεται σε προβλήματα ταξινόμησης, όπου το αποτέλεσμα ανήκει σε συγκεκριμένες κατηγορίες. Τα μοντέλα χρονοσειρών χρησιμοποιούνται όταν τα δεδομένα μεταβάλλονται με την πάροδο του χρόνου (Hyndman & Athanasopoulos, 2021).

Τα μοντέλα μηχανικής μάθησης είναι πιο σύνθετα, καθώς μπορούν να εντοπίσουν ακόμη και μη γραμμικές σχέσεις μεταξύ των δεδομένων. Σε αυτή την κατηγορία ανήκουν τα δέντρα απόφασης (decision trees), τα τυχαία δάση (random forests), οι μηχανές διανυσμάτων υποστήριξης (support vector machines – SVM) και τα τεχνητά νευρωνικά δίκτυα (artificial neural networks). Τα μοντέλα αυτά εμφανίζουν συχνά μεγαλύτερη ακρίβεια, ιδιαίτερα όταν υπάρχει μεγάλος όγκος δεδομένων και σύνθετα πρότυπα που δεν μπορούν να περιγραφούν εύκολα με τις παραδοσιακές στατιστικές μεθόδους (Berberich, 2020).

Τέλος, τα υβριδικά μοντέλα (hybrid models) συνδυάζουν δύο ή περισσότερες μεθόδους, συνήθως ένα στατιστικό μοντέλο και ένα μοντέλο μηχανικής μάθησης, ώστε να αξιοποιούν τα πλεονεκτήματα κάθε προσέγγισης. Για παράδειγμα, ένα μοντέλο ARIMA μπορεί να

χρησιμοποιηθεί για την περιγραφή της γραμμικής συμπεριφοράς μιας χρονοσειράς, ενώ ένα νευρωνικό δίκτυο ή ένα μοντέλο Δικτύου Μακράς και Βραχυπρόθεσμης Διάρκειας (Long Short-Term Memory - LSTM) μπορεί να χρησιμοποιηθεί για την ανίχνευση μη γραμμικών προτύπων. Με αυτόν τον τρόπο, τα υβριδικά μοντέλα επιτυγχάνουν συχνά μεγαλύτερη ακρίβεια πρόβλεψης σε σχέση με τις επιμέρους μεθόδους (Berberich, 2020).

2.3.3. Αναλυτική παρουσίαση βασικών μοντέλων

Η επιλογή του κατάλληλου μοντέλου πρόβλεψης εξαρτάται από τη φύση των δεδομένων, τον χρονικό ορίζοντα της πρόβλεψης και το επίπεδο πολυπλοκότητας του προβλήματος. Στην παρούσα ενότητα παρουσιάζονται τα σημαντικότερα μοντέλα που χρησιμοποιούνται στη βιβλιογραφία, με ιδιαίτερη έμφαση στα νευρωνικά δίκτυα, καθώς αυτά σχετίζονται άμεσα με τα μοντέλα NeuralProphet και τις τεχνικές Reinforcement Learning που θα μας απασχολήσουν στην παρούσα εργασία.

2.3.3.1. Γραμμική Παλινδρόμηση

Η γραμμική παλινδρόμηση (linear regression) αποτελεί ένα από τα απλούστερα και πιο διαδεδομένα μοντέλα πρόβλεψης. Στόχος της είναι να περιγράψει τη σχέση μεταξύ μίας εξαρτημένης μεταβλητής και μίας ή περισσότερων ανεξάρτητων μεταβλητών.

Στην απλή γραμμική παλινδρόμηση η σχέση περιγράφεται από την εξίσωση:

$$Y = \alpha + \beta X$$

Όπου Y είναι η μεταβλητή που θέλουμε να προβλέψουμε, X η ανεξάρτητη μεταβλητή, α η σταθερά και β ο συντελεστής κλίσης. Το μοντέλο υπολογίζει τις τιμές των παραμέτρων έτσι ώστε η ευθεία να προσαρμόζεται όσο το δυνατόν καλύτερα στα δεδομένα. Η προσαρμογή αυτή πραγματοποιείται συνήθως με τη μέθοδο των ελαχίστων τετραγώνων (least squares),

η οποία ελαχιστοποιεί το συνολικό σφάλμα μεταξύ πραγματικών και προβλεπόμενων τιμών (James et al., 2021).

Η γραμμική παλινδρόμηση παρουσιάζει αρκετά πλεονεκτήματα, όπως η απλότητα και η εύκολη ερμηνεία των αποτελεσμάτων. Έχει μικρές υπολογιστικές απαιτήσεις και είναι κατάλληλο όταν η σχέση μεταξύ μεταβλητών είναι περίπου γραμμική. Παρ' όλα αυτά, εμφανίζει και περιορισμούς, καθώς δεν μπορεί να περιγράψει αποτελεσματικά μη γραμμικές σχέσεις και επηρεάζεται σημαντικά από ακραίες τιμές. Για τον λόγο αυτό χρησιμοποιείται κυρίως ως βασικό ή συγκριτικό μοντέλο (James et al., 2021).

Μερικά παραδείγματα χρήσης είναι η εκτίμηση τιμών των ακινήτων, η πρόβλεψη κατανάλωσης ενέργειας και η οικονομική ανάλυση στις χρηματοοικονομικές προβλέψεις (Han et al., 2012).

2.3.3.2. Λογιστική Παλινδρόμηση

Η λογιστική παλινδρόμηση (logistic regression) αποτελεί στατιστικό μοντέλο που χρησιμοποιείται σε προβλήματα ταξινόμησης, όπου η εξαρτημένη μεταβλητή λαμβάνει διακριτές τιμές, συνήθως δύο κατηγορίες, όπως «ναι/όχι» ή «0/1». Σε αντίθεση με τη γραμμική παλινδρόμηση, η οποία προβλέπει συνεχείς τιμές, η λογιστική παλινδρόμηση εκτιμά την πιθανότητα εμφάνισης ενός γεγονότος.

Η πιθανότητα αυτή υπολογίζεται μέσω της λογιστικής συνάρτησης:

$$P(Y = 1) = \frac{1}{1 + e^{-(\alpha + bX)}}$$

Όπου X είναι η ανεξάρτητη μεταβλητή, α η σταθερά και b ο συντελεστής του μοντέλου. Το αποτέλεσμα λαμβάνει τιμές από 0 έως 1 και ερμηνεύεται ως πιθανότητα.

Η λογιστική παλινδρόμηση χρησιμοποιείται ευρέως στην ιατρική για την πρόβλεψη εμφάνισης μιας ασθένειας, στη χρηματοοικονομική ανάλυση για την εκτίμηση πιστωτικού κινδύνου και στο μάρκετινγκ για την πρόβλεψη της πιθανότητας αγοράς ενός προϊόντος.

Παρουσιάζει το πλεονέκτημα ότι είναι εύκολη στην ερμηνεία, ωστόσο η απόδοσή της περιορίζεται όταν οι σχέσεις μεταξύ των μεταβλητών είναι ιδιαίτερα μη γραμμικές (James et al., 2021).

2.3.3.3. Μοντέλα χρονοσειρών

Τα μοντέλα χρονοσειρών (time series models) χρησιμοποιούνται όταν τα δεδομένα καταγράφονται διαδοχικά μέσα στον χρόνο. Στόχος τους είναι να αναγνωρίσουν πρότυπα όπως η τάση, η εποχικότητα και οι τυχαίες διακυμάνσεις, ώστε να είναι δυνατή η πρόβλεψη μελλοντικών τιμών (Hyndman & Athanasopoulos, 2021).

Τα αυτοπαλίνδρομα μοντέλα (autoregressive models – AR) βασίζονται στην υπόθεση ότι η τρέχουσα τιμή μιας μεταβλητής εξαρτάται από προηγούμενες τιμές της ίδιας μεταβλητής. Ένα μοντέλο AR τάξης p εκφράζεται ως:

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + \varepsilon_t$$

όπου η τιμή στη χρονική στιγμή t εξαρτάται από τις προηγούμενες χρονικές στιγμές. Τα μοντέλα αυτά είναι χρήσιμα όταν οι παρελθοντικές τιμές περιέχουν επαρκή πληροφορία για την πρόβλεψη των μελλοντικών τιμών (Box et al., 2016).

Μια δεύτερη κατηγορία είναι τα μοντέλα κινούμενου μέσου όρου (moving average – MA), τα οποία χρησιμοποιούν προηγούμενα σφάλματα πρόβλεψης για να υπολογίσουν τη νέα τιμή.

Η βασική μορφή ενός μοντέλου MA τάξης q είναι:

$$Y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q}$$

Στην πράξη, πολύ συχνά χρησιμοποιούνται συνδυαστικά μοντέλα, όπως τα ARMA και ARIMA. Το μοντέλο αυτοπαλίνδρομης ολοκληρωμένης κινητής μέσης τιμής (Autoregressive Integrated Moving Average – ARIMA) συνδυάζει αυτοπαλίνδρομα

στοιχεία και στοιχεία κινούμενου μέσου όρου, ενώ μπορεί να εφαρμοστεί και σε μη στάσιμες χρονοσειρές μέσω διαφοροποίησης. Η δυνατότητα αυτή καθιστά το ARIMA ιδιαίτερα χρήσιμο σε πραγματικά δεδομένα, όπου οι χρονοσειρές συχνά εμφανίζουν μεταβαλλόμενη τάση ή διακύμανση. Όταν υπάρχουν και εποχικά πρότυπα, χρησιμοποιείται συνήθως το μοντέλο SARIMA (Seasonal ARIMA) (Box et al., 2016).

Τα μοντέλα χρονοσειρών χρησιμοποιούνται ευρέως στην πρόβλεψη πωλήσεων, στη διαχείριση αποθεμάτων, στην πρόβλεψη ενεργειακής κατανάλωσης και στη χρηματοοικονομική ανάλυση.

Παρότι τα στατιστικά μοντέλα χρονοσειρών παραμένουν ιδιαίτερα χρήσιμα, συχνά αδυνατούν να αποτυπώσουν πολύπλοκες μη γραμμικές σχέσεις. Για τον λόγο αυτό, τα τελευταία χρόνια αυξάνεται το ενδιαφέρον για μοντέλα βασισμένα σε νευρωνικά δίκτυα και βαθιά μάθηση, όπως τα LSTM και το NeuralProphet, το οποίο θα παρουσιαστεί σε επόμενη ενότητα.

2.3.3.4. Τεχνητά νευρωνικά δίκτυα

Τα τεχνητά νευρωνικά δίκτυα (artificial neural networks – ANN) αποτελούν μοντέλα μηχανικής μάθησης που είναι εμπνευσμένα από τον τρόπο λειτουργίας του ανθρώπινου εγκεφάλου. Αποτελούνται από διασυνδεδεμένους τεχνητούς νευρώνες, οι οποίοι οργανώνονται συνήθως σε ένα επίπεδο εισόδου, ένα ή περισσότερα κρυφά επίπεδα και ένα επίπεδο εξόδου.

Τα νευρωνικά δίκτυα έχουν τη δυνατότητα να μαθαίνουν πολύπλοκες και μη γραμμικές σχέσεις μεταξύ των δεδομένων, γεγονός που τα καθιστά ιδιαίτερα αποτελεσματικά σε προβλήματα πρόβλεψης. Σε αντίθεση με τα παραδοσιακά στατιστικά μοντέλα, δεν απαιτούν αυστηρές υποθέσεις σχετικά με τη μορφή των δεδομένων ή τη γραμμικότητα των σχέσεων. Η διαδικασία εκπαίδευσης πραγματοποιείται μέσω της προσαρμογής των βαρών των συνδέσεων μεταξύ των νευρώνων, συνήθως με τη μέθοδο της οπισθοδιάδοσης του σφάλματος (backpropagation) και τη χρήση αλγορίθμων βελτιστοποίησης, όπως ο gradient descent (Goodfellow et al., 2016).

Τα τεχνητά νευρωνικά δίκτυα χρησιμοποιούνται σε εφαρμογές όπως η αναγνώριση προτύπων, η επεξεργασία εικόνων και ήχου, η πρόβλεψη χρηματοοικονομικών δεδομένων και η ανάλυση χρονοσειρών.

Παρότι τα στατιστικά μοντέλα χρονοσειρών παραμένουν ιδιαίτερα χρήσιμα, συχνά αδυνατούν να αποτυπώσουν πολύπλοκες μη γραμμικές σχέσεις. Για τον λόγο αυτό, τα τελευταία χρόνια αυξάνεται το ενδιαφέρον για υβριδικά μοντέλα πρόβλεψης που συνδυάζουν στοιχεία νευρωνικών δικτύων και παραδοσιακών στατιστικών μεθόδων. Ένα χαρακτηριστικό παράδειγμα αποτελεί το NeuralProphet, το οποίο θα χρησιμοποιηθεί σε αυτή την εργασία.

2.3.4. Κριτήρια αξιολόγησης των μοντέλων

Η αξιολόγηση της απόδοσης ενός μοντέλου πρόβλεψης αποτελεί βασικό στάδιο της διαδικασίας ανάπτυξης και επιλογής του. Στόχος είναι να προσδιοριστεί κατά πόσο οι προβλέψεις του μοντέλου προσεγγίζουν τις πραγματικές τιμές και ποιο μοντέλο παρουσιάζει την καλύτερη απόδοση.

Στα προβλήματα πρόβλεψης συνεχών τιμών, όπως η πρόβλεψη πωλήσεων ή ενεργειακής κατανάλωσης, χρησιμοποιούνται συνήθως μέτρα σφάλματος όπως το μέσο απόλυτο σφάλμα (Mean Absolute Error – MAE), το μέσο τετραγωνικό σφάλμα (Mean Squared Error – MSE), η τετραγωνική ρίζα του μέσου τετραγωνικού σφάλματος (Root Mean Squared Error – RMSE) και το μέσο απόλυτο ποσοστιαίο σφάλμα (Mean Absolute Percentage Error – MAPE) (Hyndman & Athanasopoulos, 2021).

Το MAE εκφράζει τη μέση απόλυτη διαφορά μεταξύ των πραγματικών και των προβλεπόμενων τιμών. Μικρότερες τιμές του δείκτη υποδηλώνουν καλύτερη απόδοση του μοντέλου (James et al., 2021).

Το MSE δίνει μεγαλύτερη βαρύτητα στα μεγάλα σφάλματα, καθώς βασίζεται στο τετράγωνο της διαφοράς μεταξύ πραγματικών και προβλεπόμενων τιμών. Για τον λόγο αυτό είναι ιδιαίτερα χρήσιμο όταν τα μεγάλα σφάλματα θεωρούνται περισσότερο σημαντικά (James et al., 2021).

Το RMSE αποτελεί την τετραγωνική ρίζα του MSE και εκφράζεται στις ίδιες μονάδες με τα δεδομένα, γεγονός που διευκολύνει την ερμηνεία του. Συχνά χρησιμοποιείται ως βασικό

κριτήριο σύγκρισης διαφορετικών μοντέλων πρόβλεψης (Hyndman & Athanasopoulos, 2021).

Το MAPE εκφράζει το σφάλμα σε ποσοστιαία μορφή και επιτρέπει τη σύγκριση μεταξύ διαφορετικών συνόλων δεδομένων. Ωστόσο, παρουσιάζει περιορισμούς όταν οι πραγματικές τιμές είναι πολύ μικρές ή ίσες με μηδέν (Hyndman & Athanasopoulos, 2021). Εκτός από τα μέτρα σφάλματος, συχνά χρησιμοποιείται και ο συντελεστής προσδιορισμού R^2 , ο οποίος δείχνει ποιο ποσοστό της μεταβλητότητας των δεδομένων εξηγείται από το μοντέλο. Τιμές του R^2 κοντά στο 1 υποδηλώνουν καλύτερη προσαρμογή (James et al., 2021).

Στην παρούσα εργασία, η σύγκριση των μοντέλων θα βασιστεί κυρίως στους δείκτες RMSE, MAE και MAPE, οι οποίοι χρησιμοποιούνται ευρέως στη βιβλιογραφία για την αξιολόγηση μοντέλων πρόβλεψης καθώς επιτρέπουν την άμεση σύγκριση της ακρίβειας μεταξύ διαφορετικών προσεγγίσεων.

2.3.5. Πλεονεκτήματα και μειονεκτήματα των μοντέλων πρόβλεψης

Τα μοντέλα πρόβλεψης παρουσιάζουν σημαντικά πλεονεκτήματα, καθώς μπορούν να συμβάλουν στη βελτίωση της διαδικασίας λήψης αποφάσεων μέσω της εκτίμησης μελλοντικών εξελίξεων. Επιπλέον, επιτρέπουν τον έγκαιρο εντοπισμό τάσεων και προτύπων στα δεδομένα, μειώνοντας την αβεβαιότητα και διευκολύνοντας τον προγραμματισμό. Παράλληλα, η χρήση τους επιτρέπει την αυτοματοποίηση της ανάλυσης, γεγονός που μειώνει τον χρόνο και το κόστος επεξεργασίας των δεδομένων (Hyndman & Athanasopoulos, 2021).

Ωστόσο, τα μοντέλα πρόβλεψης παρουσιάζουν και ορισμένους περιορισμούς. Η αποτελεσματικότητά τους εξαρτάται σε μεγάλο βαθμό από την ποιότητα και την πληρότητα των διαθέσιμων δεδομένων, καθώς λανθασμένα ή ελλιπή δεδομένα μπορούν να οδηγήσουν σε ανακριβείς προβλέψεις (Kasar et al., 2025).

Επιπλέον, υπάρχει ο κίνδυνος υπερπροσαρμογής (overfitting), κατά την οποία το μοντέλο προσαρμόζεται υπερβολικά στα δεδομένα εκπαίδευσης και αποτυγχάνει να γενικεύσει σε νέα δεδομένα.

Τέλος, ορισμένα σύνθετα μοντέλα, ιδιαίτερα εκείνα που βασίζονται σε τεχνικές μηχανικής μάθησης ή νευρωνικά δίκτυα, χαρακτηρίζονται από περιορισμένη ερμηνευσιμότητα, γεγονός που δυσκολεύει την κατανόηση του τρόπου με τον οποίο παράγονται οι προβλέψεις (James et al., 2021).

2.3.6. Σημαντικότητα της εφαρμογής μοντέλων πρόβλεψης στην διπλωματική εργασία

Στην παρούσα εργασία, η πρόβλεψη ζήτησης αποτελεί το πρώτο βήμα της προτεινόμενης μεθοδολογίας. Πριν εξεταστεί η ταξινόμηση των προϊόντων και η δυναμική τιμολόγηση, είναι απαραίτητο να εκτιμηθεί με όσο το δυνατόν μεγαλύτερη ακρίβεια η μελλοντική ζήτηση των φρέσκων προϊόντων.

Η ακριβής πρόβλεψη είναι ιδιαίτερα σημαντική στο λιανικό εμπόριο τροφίμων, επειδή τα φρέσκα προϊόντα έχουν περιορισμένη διάρκεια ζωής και η λανθασμένη εκτίμηση της ζήτησης οδηγεί είτε σε ελλείψεις είτε σε αυξημένη σπατάλη (Kasar et al., 2025).

Για αυτό το λόγο, πρέπει η πρόβλεψη να έχει όσο το δυνατόν μεγαλύτερη ακρίβεια χωρίς όμως να κινδυνεύει να αλλοιωθεί από την υπερπροσαρμογή (overfitting) στα δεδομένα.

2.4. Ορισμός και βασικές αρχές της ABC ανάλυσης

Η ABC ανάλυση είναι μια μέθοδος ταξινόμησης στοιχείων σε κατηγορίες διαφορετικής σημασίας, με στόχο την ιεράρχηση και την αποτελεσματικότερη διαχείρισή τους. Η βασική ιδέα της μεθόδου είναι ότι όλα τα στοιχεία ενός συνόλου δεν έχουν την ίδια βαρύτητα ή επίδραση και, επομένως, δεν απαιτούν τον ίδιο βαθμό προσοχής. Μέσω της ABC ανάλυσης, τα στοιχεία διαχωρίζονται συνήθως σε τρεις κατηγορίες: την κατηγορία A, που περιλαμβάνει τα πιο σημαντικά στοιχεία, την κατηγορία B, που περιλαμβάνει στοιχεία μέσης σημασίας, και την κατηγορία C, που περιλαμβάνει τα λιγότερο σημαντικά στοιχεία. Η θεωρητική βάση της μεθόδου προέρχεται από την αρχή Pareto, η οποία διατυπώθηκε από τον Ιταλό οικονομολόγο Vilfredo Pareto στα τέλη του 19ου αιώνα. Ο Pareto παρατήρησε ότι περίπου το 80% του πλούτου στην Ιταλία κατείχε το 20% του πληθυσμού. Αργότερα, η

παρατήρηση αυτή γενικεύθηκε και εφαρμόστηκε σε πολλά διαφορετικά πεδία, οδηγώντας στον γνωστό «κανόνα 80/20», σύμφωνα με τον οποίο ένα μικρό ποσοστό αιτιών είναι υπεύθυνο για το μεγαλύτερο μέρος των αποτελεσμάτων. Η λογική αυτή χρησιμοποιείται ευρέως στη διοίκηση, στην οικονομία, στη διαχείριση έργων και στην ανάλυση δεδομένων (Craft & Leake, 2002).

Η εφαρμογή της αρχής Pareto στη διοίκηση αποθεμάτων και στη διαχείριση πόρων εμφανίστηκε στα μέσα του 20ού αιώνα. Ο όρος «ABC analysis» χρησιμοποιήθηκε αρχικά για την ταξινόμηση προϊόντων ή υλικών ανάλογα με τη σχετική σημασία τους, επιτρέποντας στις επιχειρήσεις να επικεντρώνουν τις προσπάθειές τους στα στοιχεία που έχουν τη μεγαλύτερη επίδραση στη λειτουργία τους. Με την πάροδο του χρόνου, η μέθοδος επεκτάθηκε και χρησιμοποιήθηκε όχι μόνο για αποθέματα, αλλά και για ταξινόμηση πελατών, προμηθευτών, κινδύνων και δραστηριοτήτων.

Στη γενική της μορφή, η ABC ανάλυση εφαρμόζεται όταν απαιτείται η ταξινόμηση ενός μεγάλου αριθμού στοιχείων με βάση κάποιο κριτήριο σημαντικότητας. Έτσι, η κατηγορία A περιλαμβάνει ένα μικρό ποσοστό στοιχείων με ιδιαίτερα υψηλή σημασία, η κατηγορία B περιλαμβάνει στοιχεία ενδιάμεσης σημασίας και η κατηγορία C περιλαμβάνει μεγάλο αριθμό στοιχείων με περιορισμένη επίδραση. Παρότι η ακριβής κατανομή μπορεί να διαφέρει ανάλογα με την εφαρμογή, συχνά παρατηρείται ότι περίπου το 10–20% των στοιχείων ανήκει στην κατηγορία A, το 20–30% στην κατηγορία B και το υπόλοιπο στην κατηγορία C.

2.4.1. ABC ανάλυση με πολλαπλά κριτήρια σε περιβάλλοντα εμπορίου

Η πολυκριτηριακή ABC ανάλυση αποτελεί εξέλιξη της παραδοσιακής ταξινόμησης αποθεμάτων, καθώς επιτρέπει την ενσωμάτωση μη κοστολογικών κριτηρίων, όπως η κρισιμότητα, η συχνότητα ζήτησης κ.λπ. Οδηγ(Flores & Whybark, 1987)ιεράρχηση αποθεμάτων.

Οι (Flores & Whybark, 1987) ήταν οι πρώτοι που έδειξαν ότι η παραδοσιακή ABC ανάλυση, η οποία βασίζεται αποκλειστικά στην ετήσια αξία κατανάλωσης, δεν είναι αρκετή για να αποτυπώσει τη συνολική σημασία ενός προϊόντος. Έτσι πρότειναν την ενσωμάτωση δεύτερου κριτηρίου, όπως η κρισιμότητα ή η συχνότητα χρήσης, και απέδειξαν ότι έτσι

προκύπτει πιο ρεαλιστική ταξινόμηση αποθεμάτων. Συγκεκριμένα, έδειξαν ότι ορισμένα προϊόντα χαμηλής αξίας αλλά υψηλής επιχειρησιακής σημασίας θα έπρεπε να αντιμετωπίζονται ως κατηγορία A ή B και όχι ως C.

Αργότερα, ο (Ramanathan, 2006) έδειξε ότι η πολυκριτηριακή ABC μπορεί να πραγματοποιηθεί με πιο αντικειμενικό τρόπο μέσω ενός μοντέλου σταθμισμένης γραμμικής βελτιστοποίησης. Η βασική συμβολή του ήταν ότι τα βάρη των κριτηρίων δεν χρειάζεται να επιλέγονται αυθαίρετα από τον ερευνητή ή τον διαχειριστή, αλλά μπορούν να υπολογίζονται αυτόματα από το ίδιο το μοντέλο. Η μελέτη έδειξε ότι η μέθοδος αυτή οδηγεί σε πιο σταθερή και ακριβή ταξινόμηση όταν χρησιμοποιούνται πολλά κριτήρια ταυτόχρονα. Σημαντική είναι επίσης η συνεισφορά των (Chu et al., 2008) που έδειξαν ότι σε περιπτώσεις όπου τα κριτήρια δεν μπορούν να εκφραστούν με απόλυτη ακρίβεια, η χρήση «fuzzy logic» οδηγεί σε πιο κατάλληλη ταξινόμηση. Για παράδειγμα, κριτήρια όπως η υψηλή κρισιμότητα ή η μέτρια σταθερότητα ζήτησης δεν είναι εύκολο να αποδοθούν με ακριβείς αριθμούς. Η μελέτη έδειξε ότι ο συνδυασμός ασαφούς ταξινόμησης και ABC ανάλυσης επιτρέπει την καλύτερη διαχείριση αβεβαιότητας αυτό την καθιστά ιδιαίτερα χρήσιμη σε περιβάλλοντα όπως το λιανικό εμπόριο και τα ευπαθή προϊόντα.

Στο λιανικό εμπόριο τροφίμων, η εφαρμογή της ABC ανάλυσης πραγματοποιείται συχνά έμμεσα μέσω πληροφοριακών συστημάτων διαχείρισης αποθεμάτων, γεγονός που αναδεικνύει τη σημασία της σωστής κατανόησης και προσαρμογής της μεθοδολογίας από το ανθρώπινο δυναμικό για την αποφυγή κακής διαχείρισης και διαφυγόντων κερδών. Οι (Jayakumaran et al., 2020) υποστηρίζουν πως η ABC ανάλυση στο λιανικό εμπόριο τροφίμων πρέπει να συνδέεται με λειτουργικά χαρακτηριστικά, όπως η συχνότητα ζήτησης και ο χρόνος αναπλήρωσης, και όχι μόνο με την οικονομική αξία των προϊόντων.

Τέλος, οι (Wang & Ning, 2026) έδειξαν ότι η σταθερότητα ή μεταβλητότητα της ζήτησης αποτελεί κρίσιμο κριτήριο για την ταξινόμηση αποθεμάτων. Μέχρι τότε, οι περισσότερες πολυκριτηριακές ABC μελέτες έδιναν έμφαση κυρίως στην οικονομική αξία και στην κρισιμότητα. Η συγκεκριμένη μελέτη έδειξε ότι προϊόντα με πολύ ασταθή ζήτηση πρέπει να αντιμετωπίζονται διαφορετικά, ακόμη και αν η οικονομική τους αξία είναι χαμηλή, επειδή δημιουργούν μεγαλύτερο κίνδυνο ελλείψεων ή υπεραποθέματος. Αυτό είναι ιδιαίτερα σημαντικό για ευπαθή προϊόντα, όπου η αβεβαιότητα της ζήτησης συνδέεται άμεσα με τη σπατάλη.

Συνολικά, η πολυκριτηριακή ABC ανάλυση αποτελεί ένα πιο ολοκληρωμένο εργαλείο ταξινόμησης σε σχέση με την παραδοσιακή προσέγγιση, καθώς επιτρέπει την ενσωμάτωση πολλαπλών παραγόντων που επηρεάζουν τη διαχείριση αποθεμάτων, ιδιαίτερα σε περιβάλλοντα με υψηλή αβεβαιότητα και ευπαθή προϊόντα.

2.4.2. Μέθοδοι πολυκριτηριακής ABC ανάλυσης

Στη βιβλιογραφία έχουν προταθεί διάφορες μέθοδοι για την υλοποίηση της πολυκριτηριακής ABC ανάλυσης, με στόχο την ενσωμάτωση πολλαπλών ποσοτικών και ποιοτικών κριτηρίων στη διαδικασία ταξινόμησης. Οι μέθοδοι αυτές διαφοροποιούνται κυρίως ως προς τον τρόπο με τον οποίο προσδιορίζονται τα βάρη των κριτηρίων και υπολογίζεται η συνολική σημαντικότητα κάθε προϊόντος.

Μία από τις πιο διαδεδομένες προσεγγίσεις είναι η Ανάλυση Ιεραρχικής Διαδικασίας (Analytic Hierarchy Process – AHP), η οποία επιτρέπει τη συστηματική σύγκριση των κριτηρίων ανά ζεύγη και τον υπολογισμό των σχετικών βαρών τους. Η μέθοδος αυτή είναι ιδιαίτερα χρήσιμη όταν εμπλέκονται ποιοτικά κριτήρια, όπως η επιχειρησιακή κρισιμότητα ή η δυνατότητα υποκατάστασης ενός προϊόντος

Μια απλούστερη προσέγγιση είναι η μέθοδος της σταθμισμένης βαθμολόγησης (weighted scoring), κατά την οποία κάθε προϊόν αξιολογείται ως προς ένα σύνολο κριτηρίων και λαμβάνει μια συνολική βαθμολογία ως γραμμικό συνδυασμό των επιμέρους τιμών και των αντίστοιχων βαρών. Η μέθοδος αυτή είναι εύκολη στην εφαρμογή και χρησιμοποιείται ευρέως στην πράξη, ιδιαίτερα σε πληροφοριακά συστήματα διαχείρισης αποθεμάτων.

Για την αντιμετώπιση της αβεβαιότητας και της ασάφειας που χαρακτηρίζει ορισμένα κριτήρια, έχουν αναπτυχθεί προσεγγίσεις βασισμένες στη θεωρία ασαφών συνόλων (fuzzy logic). Οι μέθοδοι αυτές επιτρέπουν την αναπαράσταση εννοιών όπως «υψηλή ζήτηση» ή «μέτρια κρισιμότητα» μέσω γλωσσικών μεταβλητών, οδηγώντας σε πιο ρεαλιστική αξιολόγηση των προϊόντων σε περιβάλλοντα με αβεβαιότητα (Chu et al., 2008).

Τέλος, πιο σύνθετες προσεγγίσεις βασίζονται σε μοντέλα βελτιστοποίησης, όπως η μέθοδος που προτάθηκε από τον (Ramanathan, 2006), όπου τα βάρη των κριτηρίων προσδιορίζονται ενδογενώς μέσω γραμμικού προγραμματισμού. Οι μέθοδοι αυτές μειώνουν την

υποκειμενικότητα στην επιλογή των βαρών και μπορούν να οδηγήσουν σε πιο αντικειμενική ταξινόμηση των αποθεμάτων.

Συνολικά, η επιλογή της κατάλληλης μεθόδου εξαρτάται από τη φύση των δεδομένων, τη διαθεσιμότητα πληροφοριών και το επίπεδο πολυπλοκότητας που είναι αποδεκτό από το εκάστοτε σύστημα διαχείρισης.

2.5. Δυναμική τιμολόγηση

2.5.1. Ορισμός και βασικές αρχές της δυναμικής τιμολόγησης

Η δυναμική τιμολόγηση (dynamic pricing) αποτελεί μια στρατηγική τιμολόγησης κατά την οποία οι τιμές των προϊόντων ή υπηρεσιών προσαρμόζονται δυναμικά με βάση τις συνθήκες της αγοράς και τα διαθέσιμα δεδομένα. Αν και η σύγχρονη μορφή της συνδέεται με την ανάπτυξη της τεχνολογίας και των πληροφοριακών συστημάτων, οι βασικές αρχές της δυναμικής τιμολόγησης έχουν τις ρίζες τους σε παλαιότερες πρακτικές διαφοροποίησης τιμών.

Το πρόβλημα της δυναμικής τιμολόγησης μπορεί να διατυπωθεί ως πρόβλημα βελτιστοποίησης, όπου στόχος είναι η μεγιστοποίηση των αναμενόμενων εσόδων σε έναν χρονικό ορίζοντα. Η τιμή σε κάθε χρονική περίοδο επηρεάζει άμεσα τη ζήτηση και, κατ' επέκταση, τα συνολικά έσοδα.

$$\max E \left[\sum_{t=1}^T p_t \cdot q_t(p_t) \right]$$

όπου p_t είναι η τιμή στο χρόνο t και $q_t(p_t)$ η ζήτηση ως συνάρτηση της τιμής. Η στοχαστική φύση της ζήτησης καθιστά αναγκαία τη χρήση προσδοκώμενων τιμών (Talluri & Van Ryzin, 2004).

Στην πράξη, το πρόβλημα υπόκειται σε περιορισμούς όπως το διαθέσιμο απόθεμα και η διάρκεια ζωής των προϊόντων. Σε περιβάλλοντα ευπαθών προϊόντων, η χρονική διάσταση

παίζει καθοριστικό ρόλο, καθώς η αξία του προϊόντος μειώνεται όσο πλησιάζει η ημερομηνία λήξης.

Οι πρώτες συστηματικές εφαρμογές δυναμικής τιμολόγησης εμφανίστηκαν στη βιομηχανία των αερομεταφορών κατά τη δεκαετία του 1970 και του 1980, στο πλαίσιο των συστημάτων διαχείρισης εσόδων (revenue management). Οι εταιρείες άρχισαν να προσαρμόζουν τις τιμές των εισιτηρίων ανάλογα με τη ζήτηση, τον χρόνο κράτησης και τη διαθεσιμότητα θέσεων, με στόχο τη μεγιστοποίηση των εσόδων από περιορισμένη χωρητικότητα (Talluri & Van Ryzin, 2004).

Κατά τις επόμενες δεκαετίες, η δυναμική τιμολόγηση επεκτάθηκε σε άλλους τομείς, όπως τα ξενοδοχεία, οι μεταφορές και το ηλεκτρονικό εμπόριο, καθώς η αύξηση της υπολογιστικής ισχύος και η διαθεσιμότητα δεδομένων επέτρεψαν την ανάπτυξη πιο σύνθετων μοντέλων τιμολόγησης. Ιδιαίτερα από τις αρχές του 21ου αιώνα, η ενσωμάτωση αλγοριθμικών μεθόδων και τεχνικών ανάλυσης δεδομένων κατέστησε δυνατή την εφαρμογή δυναμικής τιμολόγησης σε πραγματικό χρόνο (Phillips, 2005).

Στη σύγχρονη μορφή της, η δυναμική τιμολόγηση συνδέεται στενά με τη μηχανική μάθηση και την ανάλυση μεγάλων δεδομένων, επιτρέποντας στις επιχειρήσεις να προσαρμόζουν τις τιμές με βάση πολύπλοκα πρότυπα ζήτησης και συμπεριφοράς καταναλωτών. Στο πλαίσιο αυτό, η τιμή αντιμετωπίζεται ως μεταβλητή απόφασης που μπορεί να προσαρμόζεται δυναμικά με στόχο τη βελτιστοποίηση των εσόδων ή του κέρδους.

Οι βασικοί παράγοντες που επηρεάζουν τη δυναμική τιμολόγηση περιλαμβάνουν τη ζήτηση, το διαθέσιμο απόθεμα, τον χρόνο και τον ανταγωνισμό, οι οποίοι αλληλεπιδρούν μεταξύ τους και καθορίζουν τη βέλτιστη στρατηγική τιμολόγησης.

2.5.2. Κατηγορίες δυναμικής τιμολόγησης

Στη βιβλιογραφία διακρίνονται διάφορες μορφές δυναμικής τιμολόγησης, ανάλογα με το βασικό κριτήριο προσαρμογής της τιμής.

Η τιμολόγηση με βάση τον χρόνο (time-based pricing) αφορά την προσαρμογή των τιμών ανάλογα με χρονικούς παράγοντες, όπως η εποχικότητα ή η χρονική στιγμή της αγοράς. Η προσέγγιση αυτή χρησιμοποιείται ευρέως σε κλάδους όπως οι μεταφορές και ο τουρισμός (Talluri & Van Ryzin, 2004).

Η τιμολόγηση με βάση τη ζήτηση (demand-based pricing) προσαρμόζει τις τιμές ανάλογα με το επίπεδο ζήτησης, αυξάνοντάς τις σε περιόδους υψηλής ζήτησης και μειώνοντάς τις όταν η ζήτηση είναι χαμηλή (Phillips, 2005).

Η τιμολόγηση με βάση το απόθεμα (inventory-based pricing) λαμβάνει υπόψη τη διαθεσιμότητα των προϊόντων και χρησιμοποιείται ιδιαίτερα σε περιβάλλοντα με περιορισμένο ή ευπαθές απόθεμα. Σε περιπτώσεις υπεραποθέματος, εφαρμόζονται μειώσεις τιμών για την επιτάχυνση των πωλήσεων (Talluri & Van Ryzin, 2004).

Τέλος, η εξατομικευμένη τιμολόγηση (personalized pricing) αφορά την προσαρμογή των τιμών σε επίπεδο καταναλωτή, με βάση δεδομένα συμπεριφοράς και προτιμήσεων (Y. Chen & Zhang, 2009). Ένα παρακλάδι που είναι άξιο αναφοράς αυτής της κατηγορίας είναι η τιμολόγηση βάσει τμήματος η οποία κατηγοριοποιεί τους πελάτες με διαφορετικές τιμές για κάθε κατηγορία.

2.5.3. Περιορισμοί και προκλήσεις της δυναμικής τιμολόγησης

Παρά τα σημαντικά πλεονεκτήματά της, η δυναμική τιμολόγηση συνοδεύεται από προκλήσεις. Η αβεβαιότητα της ζήτησης αποτελεί έναν από τους βασικούς περιορισμούς, καθώς η συμπεριφορά των καταναλωτών δεν μπορεί να προβλεφθεί με απόλυτη ακρίβεια. Επιπλέον, η συχνή μεταβολή των τιμών μπορεί να δημιουργήσει ζητήματα αντιληπτής δικαιοσύνης (perceived fairness), επηρεάζοντας αρνητικά την εμπιστοσύνη των καταναλωτών (Bolton et al., 2002). Οι επιχειρήσεις οφείλουν να ισορροπήσουν μεταξύ βελτιστοποίησης των εσόδων και διατήρησης θετικής εικόνας στην αγορά.

Τέλος, η εφαρμογή δυναμικής τιμολόγησης απαιτεί κατάλληλες τεχνολογικές υποδομές και επεξεργασία μεγάλου όγκου δεδομένων, γεγονός που αυξάνει την πολυπλοκότητα και το κόστος υλοποίησης.

2.5.4. Δυναμική τιμολόγηση και διαχείριση αποθεμάτων σε ευπαθή προϊόντα

Η πρόσφατη βιβλιογραφία καταδεικνύει ότι η δυναμική τιμολόγηση αποτελεί αποτελεσματικό μηχανισμό διαχείρισης ευπαθών προϊόντων και μείωσης της σπατάλης στο

λιανικό εμπόριο. Οι (Kayıkcı et al., 2022) προτείνουν μια στρατηγική βασισμένη σε δεδομένα που αξιοποιεί δεδομένα πραγματικού χρόνου από IoT (Internet of Things) αισθητήρες για την εκτίμηση της φρεσκάδας και την προσαρμογή των τιμών, επιτυγχάνοντας ταυτόχρονα αύξηση κερδοφορίας και μείωση της σπατάλης τροφίμων.

Σε παρόμοιο πλαίσιο, οι (Yang et al., 2022) ενσωματώνουν βαθιά ενισχυτική μάθηση (deep reinforcement learning) και στρατηγικές αποκάλυψης πληροφορίας ποιότητας, δείχνοντας ότι η ποιοτικά προσαρμοσμένη τιμολόγηση μπορεί να ευθυγραμμίσει τη ζήτηση με τη φθίνουσα ποιότητα των προϊόντων και να περιορίσει τις απώλειες.

Παράλληλα, ο (Sanders, 2024) δείχνει εμπειρικά ότι η δυναμική τιμολόγηση μειώνει τη σπατάλη τροφίμων πιο αποτελεσματικά από κανονιστικά μέτρα, όπως οι απαγορεύσεις οργανικών αποβλήτων, ενώ ταυτόχρονα αυξάνει τα κέρδη και την ευημερία των καταναλωτών.

Επιπλέον, οι (Pradhan et al., 2025) επισημαίνουν ότι τα συστήματα τιμολόγησης πραγματικού χρόνου καθοδηγούμενα από τεχνητή νοημοσύνη αποτελούν την εξέλιξη της δυναμικής τιμολόγησης, αναδεικνύοντας τόσο τα οικονομικά τους οφέλη όσο και τις προκλήσεις διαφάνειας, δικαιοσύνης και κανονιστικής συμμόρφωσης, ιδιαίτερα κρίσιμες για αγορές τροφίμων.

Τέλος, σύμφωνα με τους (Khanorkar & Kane, 2023) η ταξινόμηση αποθεμάτων ευπαθών προϊόντων με τη μέθοδο ABC έχει εφαρμοστεί σε περιβάλλοντα όπου τα είδη έχουν σταθερό χρόνο ζωής στο ράφι, δείχνοντας ότι η κατηγοριοποίηση αυτή παρέχει σημαντική πληροφόρηση για προτεραιοποίηση προϊόντων και διαφοροποιημένη διαχείριση πολιτικών τιμολόγησης και αποθεμάτων.

2.6. Ενισχυτική μάθηση για πολιτικές τιμολόγησης & αναπλήρωσης προϊόντων.

Η ενισχυτική μάθηση (Reinforcement Learning – RL) έχει αναδειχθεί ως μία ιδιαίτερα ισχυρή μεθοδολογία για την αντιμετώπιση προβλημάτων δυναμικής τιμολόγησης και αναπλήρωσης αποθεμάτων, όπου οι αποφάσεις λαμβάνονται διαδοχικά υπό αβεβαιότητα και οι επιπτώσεις τους επηρεάζουν τη μελλοντική εξέλιξη του συστήματος. Η σχετική

βιβλιογραφία βασίζεται θεωρητικά σε κλασικά μοντέλα δυναμικής τιμολόγησης αποθεμάτων με στοχαστική και τιμοεξαρτώμενη ζήτηση, τα οποία δείχνουν ότι οι βέλτιστες τιμές εξαρτώνται τόσο από το επίπεδο αποθέματος όσο και από τον εναπομείναντα χρονικό ορίζοντα πώλησης (Gallego & van Ryzin, 1994).

Σε αντίθεση με τις παραδοσιακές αναλυτικές προσεγγίσεις, η μηχανική μάθηση επιτρέπει την εκμάθηση πολιτικών απευθείας από την αλληλεπίδραση με το περιβάλλον, χωρίς την ανάγκη πλήρους γνώσης της κατανομής της ζήτησης, γεγονός που την καθιστά ιδιαίτερα κατάλληλη για πραγματικά συστήματα λιανικής (Tesauro & Kephart, 2002).

Πέρα από τις κλασσικές προσεγγίσεις, στα προβλήματα ευπαθών προϊόντων, η δυναμική τιμολόγηση μέσω RL έχει εφαρμοστεί επιτυχώς με αλγορίθμους Q-learning, όπου οι ενέργειες αντιστοιχούν σε διακριτές μεταβολές τιμών ή ποσοστά markdown, λαμβάνοντας υπόψη αβεβαιότητα ζήτησης, προτιμήσεις καταναλωτών και ανταγωνισμό (W. Chen et al., 2018). Παράλληλα, η χρήση βαθιών νευρωνικών δικτύων σε αλγορίθμους όπως τα Deep Q-Networks (DQN) επέτρεψε τη διαχείριση μεγάλων χώρων κατάστασης, όπως αναπαραστάσεις αποθέματος ανά ηλικία προϊόντος, και κατέδειξε ότι η RL μπορεί να μειώσει σημαντικά τα απόβλητα σε συστήματα ευπαθών αποθεμάτων, σε σύγκριση με κλασσικές ευρετικές πολιτικές (Kara & Dogan, 2018).

Πιο πρόσφατα, αλγόριθμοι βασισμένοι σε πολιτικές, και ειδικότερα ο Proximal Policy Optimization (PPO), έχουν χρησιμοποιηθεί για την από κοινού βελτιστοποίηση τιμολόγησης και αναπλήρωσης, επιτρέποντας συνεχή χώρο ενεργειών και σταθερότερη εκπαίδευση σε πολύπλοκα περιβάλλοντα με χρονικές υστερήσεις και φθορά προϊόντων (Nomura et al., 2025; Schulman et al., 2017).

Ένα κρίσιμο ζήτημα στα προβλήματα αυτά είναι ότι τα απόβλητα και οι ελλείψεις αποτελούν σπάνια αλλά ιδιαίτερα κοστοβόρα γεγονότα, τα οποία δεν αποτυπώνονται επαρκώς όταν η RL στοχεύει αποκλειστικά στη μεγιστοποίηση της αναμενόμενης ανταμοιβής. Για τον λόγο αυτό, έχει προταθεί η distributional reinforcement learning, η οποία μοντελοποιεί ολόκληρη την κατανομή της μελλοντικής απόδοσης και όχι μόνο τη μέση τιμή, προσφέροντας μεγαλύτερη ευαισθησία σε κινδύνους και ακραία γεγονότα (Bellemare et al., 2017).

Η προσέγγιση αυτή έχει υλοποιηθεί πρακτικά μέσω αλγορίθμων όπως το Quantile Regression DQN, οι οποίοι έχουν αποδειχθεί ιδιαίτερα αποτελεσματικοί σε περιβάλλοντα με υψηλή αβεβαιότητα και ασύμμετρα κόστη (Dabney et al., 2017).

Στο πλαίσιο της λιανικής πώλησης ευπαθών προϊόντων, η ενσωμάτωση προβλέψεων ζήτησης, διανυσμάτων αποθέματος ανά ημέρα έως τη λήξη και ρητών ποινών για απόβλητα και stockouts στο σήμα ανταμοιβής έχει οδηγήσει στη διαμόρφωση ρεαλιστικών RL περιβαλλόντων, στα οποία οι distributional μέθοδοι υπερέχουν έναντι κλασικών προσεγγίσεων τόσο σε κερδοφορία όσο και σε μείωση αποβλήτων (Jullien et al., 2023).

Τέλος, η βιβλιογραφία επισημαίνει ότι σε ανταγωνιστικά περιβάλλοντα τιμολόγησης, οι RL αλγόριθμοι ενδέχεται να αναπτύξουν μη αναμενόμενες στρατηγικές, όπως σιωπηρή σύμπραξη, γεγονός που καθιστά αναγκαία την προσεκτική ερμηνεία και αξιολόγηση των μαθημένων πολιτικών (Calvano et al., 2020).

2.7. Μείωση σπατάλης στο λιανικό εμπόριο μέσω βελτιστοποίησης λειτουργιών

Η μείωση της σπατάλης τροφίμων στο λιανικό εμπόριο αποτελεί κατεξοχήν πρόβλημα λειτουργικής βελτιστοποίησης, καθώς προκύπτει από αλληλεξαρτώμενες αποφάσεις που αφορούν τον σχεδιασμό του product assortment, δηλαδή τη σύνθεση και το εύρος των προϊόντων που διατίθενται στο κατάστημα όπως επίσης και τα επίπεδα εξυπηρέτησης, τις πολιτικές αναπλήρωσης και τις στρατηγικές διάθεσης προϊόντων κοντά στη λήξη τους.

Η σχετική βιβλιογραφία υπογραμμίζει ότι η σπατάλη δεν είναι απλώς αποτέλεσμα αβεβαιότητας ζήτησης, αλλά συστηματική συνέπεια λειτουργικών επιλογών που συχνά ευνοούν τη διαθεσιμότητα και τη φρεσκάδα εις βάρος της αποδοτικότητας (Riesenegger & Hübner, 2022).

Σε επίπεδο αναπλήρωσης, μελέτες σε πραγματικά περιβάλλοντα σούπερ μάρκετ δείχνουν ότι κατάλληλα σχεδιασμένες πολιτικές παραγγελίας μπορούν να μειώσουν σημαντικά τα απόβλητα χωρίς να υποβαθμίζεται το επίπεδο εξυπηρέτησης, ακόμη και όταν η ηλικιακή κατανομή των προϊόντων στο ράφι δεν είναι πλήρως γνωστή, όπως συμβαίνει συχνά στην πράξη (Hendrix et al., 2023).

Παράλληλα, η δυναμική τιμολόγηση και οι πολιτικές markdown αναδεικνύονται ως κρίσιμοι λειτουργικοί μοχλοί μείωσης σπατάλης, ιδίως όταν συνδυάζονται με προσαρμοζόμενη διάρκεια ζωής προϊόντων (dynamic shelf life), καθώς επιτρέπουν την ευθυγράμμιση της αντιλαμβανόμενης ποιότητας με την τιμή, μειώνοντας τόσο τα απούλητα αποθέματα όσο και τις ελλείψεις (Buisman et al., 2019) .

Όταν η πρόληψη της σπατάλης δεν είναι εφικτή, η βιβλιογραφία δείχνει ότι η βελτιστοποίηση των αποφάσεων διάθεσης αποθεμάτων, και ειδικότερα ο χρονισμός και η ποσότητα δωρεών τροφίμων, μπορεί να λειτουργήσει συμπληρωματικά, μειώνοντας τα απόβλητα και ταυτόχρονα βελτιώνοντας τόσο την οικονομική όσο και την κοινωνική απόδοση των επιχειρήσεων λιανικής (Mallidis et al., 2022).

Συνολικά, τα παραπάνω έργα συγκλίνουν στο ότι η αποτελεσματική μείωση της σπατάλης στο λιανικό εμπόριο δεν επιτυγχάνεται μέσω μεμονωμένων παρεμβάσεων, αλλά απαιτεί ολιστική βελτιστοποίηση λειτουργιών σε επίπεδο καταστήματος, όπου οι αποφάσεις αναπλήρωσης, τιμολόγησης και διάθεσης αποθεμάτων αντιμετωπίζονται ως ενιαίο δυναμικό σύστημα.

3. Δεδομένα

3.1. Χρήση συνθετικών δεδομένων

Η πρόσβαση σε πραγματικά δεδομένα επιχειρήσεων λιανικού εμπορίου αποτελεί σημαντική πρόκληση για ερευνητικές μελέτες. Τα δεδομένα πωλήσεων, τιμολόγησης και αποθεμάτων θεωρούνται συνήθως ιδιαίτερα ευαίσθητα, καθώς αποκαλύπτουν εμπορικές στρατηγικές και επιχειρησιακές πρακτικές. Για τον λόγο αυτό, οι περισσότερες επιχειρήσεις δεν δημοσιοποιούν τέτοιου είδους δεδομένα, με αποτέλεσμα η δημόσια διαθεσιμότητα ολοκληρωμένων συνόλων δεδομένων να είναι περιορισμένη.

Η περιορισμένη πρόσβαση σε πραγματικές βάσεις δεδομένων έχει οδηγήσει σε αυξανόμενη χρήση συνθετικών δεδομένων σε ερευνητικές εφαρμογές. Τα συνθετικά δεδομένα επιτρέπουν τη δημιουργία μεγάλων και πλήρως ελεγχόμενων συνόλων δεδομένων, χωρίς περιορισμούς εμπιστευτικότητας ή ιδιωτικότητας. Παράλληλα, δίνουν τη δυνατότητα ενσωμάτωσης συγκεκριμένων χαρακτηριστικών της πραγματικής λειτουργίας ενός συστήματος, όπως εποχικότητα ζήτησης, ημερολογιακά γεγονότα και περιορισμοί αποθέματος.

Στο πλαίσιο της παρούσας εργασίας δημιουργήθηκε ένα συνθετικό σύνολο δεδομένων που προσομοιώνει τη λειτουργία ενός καταστήματος σούπερ μάρκετ. Η παραγωγή των δεδομένων πραγματοποιήθηκε μέσω προσαρμοσμένου αλγορίθμου σε γλώσσα Python, ο οποίος ενσωματώνει βασικές λειτουργίες του λιανικού εμπορίου τροφίμων, όπως η εποχικότητα της ζήτησης, οι επιδράσεις αργιών και εμπορικών γεγονότων, η δυναμική τιμολόγηση, οι περιορισμοί αποθέματος και η αλλοίωση προϊόντων.

Η χρήση συνθετικών δεδομένων για την ανάπτυξη και αξιολόγηση μοντέλων μηχανικής μάθησης αποτελεί πλέον ενεργό ερευνητικό πεδίο. Πρόσφατες μελέτες δείχνουν ότι η δημιουργία συνθετικών σχεσιακών βάσεων δεδομένων μπορεί να υποστηρίξει την εκπαίδευση μοντέλων ανάλυσης δεδομένων και να επιτρέψει την κλιμάκωση της εκπαίδευσης σε περιβάλλοντα όπου τα πραγματικά δεδομένα είναι περιορισμένα (Kothapalli et al., 2026).

3.2. Δομή συνόλου δεδομένων

Το σύνολο δεδομένων που δημιουργήθηκε στο πλαίσιο της παρούσας εργασίας αποτελείται από τέσσερα βασικά αρχεία, τα οποία αποτυπώνουν συμπληρωματικές πτυχές της λειτουργίας ενός καταστήματος σούπερ μάρκετ και χρησιμοποιούνται συνδυαστικά στα επόμενα στάδια της ανάλυσης.

Το αρχείο `products.csv` περιλαμβάνει τα στατικά χαρακτηριστικά των προϊόντων (SKUs), όπως η κατηγορία, η υποκατηγορία, η μονάδα μέτρησης, η διάρκεια ζωής, η βασική τιμή πώλησης και το κόστος μονάδας. Επιπλέον, περιλαμβάνονται παράμετροι κόστους που σχετίζονται με την αποθήκευση και την απόσυρση προϊόντων. Το αρχείο αυτό αποτελεί τη βάση για την οικονομική ανάλυση, την ανάλυση ABC και τη μοντελοποίηση της αλλοίωσης.

Το αρχείο `calendar_events.csv` περιλαμβάνει το ημερολόγιο λειτουργίας του καταστήματος και καταγράφει όλες τις επίσημες ελληνικές αργίες, τις Κυριακές, καθώς και εμπορικά γεγονότα και περιόδους αυξημένης ζήτησης. Το ημερολόγιο αυτό χρησιμοποιείται τόσο στη μοντελοποίηση της ζήτησης όσο και στον καθορισμό της λειτουργίας του καταστήματος (ανοιχτό ή κλειστό).

Το αρχείο `transactions_daily.csv` αποτελεί το βασικό δυναμικό σύνολο δεδομένων και περιλαμβάνει τις ημερήσιες πωλήσεις ανά προϊόν και ημέρα. Για κάθε εγγραφή καταγράφονται οι πωληθείσες ποσότητες, η τελική τιμή πώλησης, η εφαρμογή προσφορών, καθώς και οικονομικά μεγέθη όπως τα έσοδα, το κόστος πωληθέντων και το μικτό κέρδος. Επιπλέον, περιλαμβάνονται δείκτες που σχετίζονται με την εμφάνιση ελλείψεων αποθέματος (stockouts) και την καταγραφή χαμένων πωλήσεων. Το αρχείο αυτό αποτελεί την κύρια είσοδο για τα μοντέλα πρόβλεψης ζήτησης, την ανάλυση ABC και την εκπαίδευση του μοντέλου Reinforcement Learning.

Τέλος, το αρχείο `waste_daily.csv` καταγράφει την ημερήσια αλλοίωση και σπατάλη αποθέματος ανά προϊόν. Περιλαμβάνει την ποσότητα που αποσύρεται, τον λόγο απόσυρσης (π.χ. λήξη ή φθορά) και το αντίστοιχο κόστος. Τα δεδομένα αυτά χρησιμοποιούνται για την

αξιολόγηση της σπατάλης τροφίμων και τη μελέτη της επίδρασης διαφορετικών πολιτικών αποθέματος και τιμολόγησης.

Η παραπάνω διάρθρωση επιτρέπει τον σαφή διαχωρισμό μεταξύ στατικών δεδομένων προϊόντων και δυναμικών δεδομένων λειτουργίας, διευκολύνοντας την ανάλυση και την αξιοποίηση του συνόλου δεδομένων στα επόμενα στάδια της εργασίας.

3.3. Χρονικό εύρος και επίπεδο ανάλυσης

Το σύνολο δεδομένων καλύπτει χρονική περίοδο δύο ετών, από την 1η Ιανουαρίου 2024 έως την 31η Δεκεμβρίου 2025. Τα δεδομένα έχουν ημερήσια χρονική ανάλυση.

Η επιλογή ημερήσιας ανάλυσης είναι σύμφωνη με τις πρακτικές του λιανικού εμπορίου, όπου οι αποφάσεις που αφορούν την αναπλήρωση αποθεμάτων, την εφαρμογή προσφορών και τη διαχείριση αλλοιώσιμων προϊόντων λαμβάνονται συνήθως σε καθημερινή βάση. Επιπλέον, η ημερήσια ζήτηση αποτελεί βασική μονάδα ανάλυσης στη σχετική βιβλιογραφία για προβλήματα πρόβλεψης ζήτησης και διαχείρισης αποθεμάτων (Silver et al., 2016).

3.4. Προϊόντα και κατηγορίες

Το σύνολο δεδομένων περιλαμβάνει συνολικά 130 προϊόντα (Stock Keeping Units – SKUs). Τα προϊόντα κατανέμονται σε οκτώ βασικές κατηγορίες τροφίμων:

- | | |
|-------------------------------------|--------|
| • Κρέας (Meat) | 18/130 |
| • Ψάρι (Fish) | 10/130 |
| • Γαλακτοκομικά (Dairy) | 18/130 |
| • Προϊόντα παντοπωλείου (Groceries) | 28/130 |
| • Φρούτα (Fruits) | 12/130 |
| • Λαχανικά (Vegetables) | 24/130 |
| • Προϊόντα αρτοποιείου (Bakery) | 10/130 |
| • Προϊόντα Delicatessen (Deli) | 10/130 |

Η κατανομή των προϊόντων μεταξύ των κατηγοριών αντανακλά τη δομή ενός τυπικού σούπερ μάρκετ. Για παράδειγμα, κατηγορίες όπως τα προϊόντα παντοπωλείου ή τα λαχανικά περιλαμβάνουν μεγαλύτερη ποικιλία προϊόντων, ενώ κατηγορίες όπως τα ψάρια περιλαμβάνουν μικρότερο αριθμό προϊόντων.

Κάθε προϊόν χαρακτηρίζεται από ένα σύνολο στατικών χαρακτηριστικών, τα οποία αποθηκεύονται στο αρχείο products.csv.

Τα χαρακτηριστικά αυτά περιλαμβάνουν:

- μοναδικό αναγνωριστικό προϊόντος (id)
- όνομα
- κατηγορία και υποκατηγορία προϊόντος
- μονάδα μέτρησης (kg, pack, pcs, L)
- βασική τιμή πώλησης
- κόστος μονάδας
- μικτό περιθώριο κέρδους
- διάρκεια ζωής προϊόντος
- κόστος απόσυρσης προϊόντος ανά μονάδα
- ημερήσιο κόστος αποθήκευσης ανά μονάδα
- δείκτης επιχειρησιακής κρισιμότητας (service criticality)

Η διάρκεια ζωής των προϊόντων διαφοροποιείται ανά κατηγορία. Για παράδειγμα, τα ψάρια έχουν διάρκεια ζωής από μία έως τρεις ημέρες, ενώ προϊόντα παντοπωλείου μπορούν να έχουν διάρκεια ζωής έως και ένα έτος.

Επιπλέον, σε κάθε προϊόν αποδίδεται ένας δείκτης επιχειρησιακής κρισιμότητας (service criticality), ο οποίος εκφράζει τη σημασία της συνεχούς διαθεσιμότητας του προϊόντος για τη λειτουργία του καταστήματος και την εξυπηρέτηση των πελατών. Η φιλοσοφία στην συγκεκριμένη μεταβλητή βρίσκεται στην βιβλιογραφία και ως “loss leader strategy” και είναι προϊόντα που και με αρνητικό κέρδος συμβάλουν στην κερδοφορία της εκάστοτε επιχείρησης επειδή συμβάλλουν ενεργά στην αύξηση της συνολικής αγοράς του «καλαθιού» ανά πελάτη (Z. Chen & Rey, 2012). Ο δείκτης λαμβάνει τιμές 0, 0.5 ή 1, όπου υψηλότερες τιμές αντιστοιχούν σε βασικά προϊόντα καθημερινής κατανάλωσης, όπως το

γάλα, το ψωμί, τα ζυμαρικά και τα βασικά λαχανικά, ενώ χαμηλότερες τιμές αντιστοιχούν σε προϊόντα μικρότερης στρατηγικής σημασίας ή πιο εξειδικευμένης κατανάλωσης. Ο δείκτης αυτός χρησιμοποιείται μεταγενέστερα ως ένα από τα κριτήρια της πολυκριτηριακής ανάλυσης ABC.

3.5. Τιμολόγηση προϊόντων

Η τιμή πώλησης κάθε προϊόντος δημιουργείται μέσω διαδικασίας που προσομοιώνει πρακτικές τιμολόγησης του λιανικού εμπορίου. Για τον σκοπό αυτό ορίζονται εύρη τιμών σε επίπεδο υποκατηγορίας προϊόντος.

Τα εύρη τιμών προσδιορίστηκαν βάσει ενδεικτικών τιμών που παρατηρούνται σε ελληνικά σούπερ μάρκετ και online retail catalogs, με στόχο τη δημιουργία ρεαλιστικών συνθηκών προσομοίωσης.

Οι τελικές τιμές δεν αποσκοπούν στην ακριβή αναπαράσταση πραγματικών εμπορικών δεδομένων, αλλά στη δημιουργία ρεαλιστικών και διαφοροποιημένων προτύπων τιμολόγησης.

Για παράδειγμα, προϊόντα όπως ο σολομός ή οι γαρίδες έχουν υψηλότερα εύρη τιμών σε σύγκριση με άλλα είδη ψαριών. Αντίστοιχα, προϊόντα όπως το γάλα ή τα ζυμαρικά έχουν χαμηλότερα εύρη τιμών.

Σε περιπτώσεις όπου δεν υπάρχει εξειδικευμένο εύρος τιμών για συγκεκριμένη υποκατηγορία, χρησιμοποιείται ένα γενικό εύρος τιμών σε επίπεδο κατηγορίας προϊόντος.

Η τελική τιμή κάθε προϊόντος προκύπτει μέσω τυχαίας επιλογής τιμής μέσα από το αντίστοιχο εύρος και προσαρμόζεται με έναν μικρό τυχαίο συντελεστή διαφοροποίησης. Ο συντελεστής αυτός προσομοιώνει διαφορές που οφείλονται σε παράγοντες όπως η μάρκα, ο προμηθευτής ή η ποιότητα του προϊόντος

Το κόστος μονάδας υπολογίζεται με βάση το μικτό περιθώριο κέρδους, το οποίο επιλέγεται από εύρη τιμών ανά κατηγορία προϊόντων.

3.6. Ημερολόγιο λειτουργίας και εμπορικά γεγονότα

Το σύνολο δεδομένων περιλαμβάνει ένα αναλυτικό ημερολόγιο λειτουργίας του καταστήματος, το οποίο αποθηκεύεται στο αρχείο `calendar_events.csv`. Το ημερολόγιο αυτό καταγράφει όλες τις ημέρες κατά τις οποίες το κατάστημα παραμένει κλειστό, καθώς και γεγονότα που επηρεάζουν τη ζήτηση.

Οι ημέρες κατά τις οποίες το κατάστημα θεωρείται κλειστό περιλαμβάνουν όλες τις Κυριακές, καθώς και τις επίσημες ελληνικές αργίες. Οι αργίες χωρίζονται σε σταθερές και κινητές.

Στις σταθερές αργίες περιλαμβάνονται:

- Πρωτοχρονιά
- Θεοφάνια
- 25η Μαρτίου
- Πρωτομαγιά
- Δεκαπενταύγουστος
- 28η Οκτωβρίου
- Χριστούγεννα
- δεύτερη ημέρα Χριστουγέννων

Οι κινητές αργίες υπολογίζονται με βάση το Ορθόδοξο Πάσχα και περιλαμβάνουν:

- Καθαρά Δευτέρα
- Μεγάλη Παρασκευή
- Κυριακή του Πάσχα
- Δευτέρα του Πάσχα
- Αγίου Πνεύματος

Εκτός από τις αργίες, στο ημερολόγιο ενσωματώνονται και διάφορα εμπορικά γεγονότα τα οποία επηρεάζουν τη ζήτηση. Μεταξύ αυτών περιλαμβάνονται:

- Η ημέρα μισθοδοσίας (payday), η οποία εμφανίζεται κάθε μήνα στις 25 του μήνα.
- Η περίοδος επιστροφής στα σχολεία (Back to School).

- Η Black Friday.
- Η Τσικνοπέμπτη, η οποία συνδέεται με αυξημένη κατανάλωση κρέατος.
- Περίοδοι αυξημένης ζήτησης πριν από μεγάλες εορτές.

Συγκεκριμένα, για την περίοδο πριν από τα Χριστούγεννα ορίζεται ένα διάστημα αυξημένης ζήτησης από τις 20 έως τις 24 Δεκεμβρίου. Αντίστοιχα, για την περίοδο πριν από το Πάσχα ορίζεται αυξημένη ζήτηση κατά τις τρεις ημέρες που προηγούνται της Κυριακής του Πάσχα.

Για κάθε γεγονός ορίζεται ένας συντελεστής αύξησης της ζήτησης, ο οποίος εφαρμόζεται στα προϊόντα που επηρεάζονται περισσότερο από το συγκεκριμένο γεγονός.

3.7. Μοντελοποίηση ζήτησης

Η ημερήσια ζήτηση κάθε προϊόντος δημιουργείται με τρόπο που προσεγγίζει τη συμπεριφορά των καταναλωτών σε ένα πραγματικό περιβάλλον λιανικού εμπορίου.

Αρχικά, σε κάθε προϊόν αντιστοιχίζεται μια βασική ζήτηση, η οποία εκφράζει τον μέσο αριθμό τεμαχίων που αναμένεται να πωλούνται ημερησίως. Η τιμή αυτή διαφοροποιείται ανά κατηγορία προϊόντος, ώστε να αντικατοπτρίζονται διαφορές στη δημοτικότητα και τη συχνότητα κατανάλωσης.

Στη συνέχεια, η ζήτηση μεταβάλλεται καθημερινά με βάση διάφορους παράγοντες.

Πρώτον, λαμβάνεται υπόψη η ημέρα της εβδομάδας. Παρατηρείται ότι οι πωλήσεις τείνουν να αυξάνονται προς το τέλος της εβδομάδας, ιδιαίτερα το Σάββατο, ενώ η ζήτηση της Κυριακής μεταφέρεται στις προηγούμενες ημέρες λόγω μη λειτουργίας του καταστήματος. Δεύτερον, ενσωματώνεται ετήσια εποχικότητα, ώστε να αποτυπώνονται σταδιακές μεταβολές της ζήτησης κατά τη διάρκεια του έτους. Για παράδειγμα, ορισμένα προϊόντα εμφανίζουν αυξημένη ζήτηση σε συγκεκριμένες περιόδους, όπως τα φρούτα το καλοκαίρι. Τρίτον, λαμβάνονται υπόψη ημερολογιακά και εμπορικά γεγονότα. Για παράδειγμα, η Τσικνοπέμπτη οδηγεί σε αύξηση της ζήτησης για προϊόντα κρέατος, ενώ η Καθαρά Δευτέρα επηρεάζει περισσότερο προϊόντα όπως τα θαλασσινά και τα λαχανικά. Αντίστοιχα, περίοδοι

όπως τα Χριστούγεννα ή το Πάσχα συνοδεύονται από γενικότερη αύξηση της κατανάλωσης.

Επιπλέον, για κάθε προϊόν εισάγεται μια μακροχρόνια τάση ζήτησης, η οποία μπορεί να είναι είτε αυξητική είτε μειωτική, αποτυπώνοντας σταδιακές μεταβολές στη συμπεριφορά των καταναλωτών κατά τη διάρκεια της εξεταζόμενης περιόδου.

Τέλος, εισάγεται μια μικρή τυχαία μεταβολή στη ζήτηση (θόρυβος) με τυπική απόκλιση 0.22, ώστε να αποτυπώνεται η φυσική αβεβαιότητα της καταναλωτικής συμπεριφοράς. Στην πράξη, η ζήτηση δεν είναι ποτέ απολύτως σταθερή, ακόμη και υπό τις ίδιες συνθήκες, καθώς επηρεάζεται από πολλούς μη παρατηρήσιμους παράγοντες, όπως οι προτιμήσεις των καταναλωτών ή εξωτερικά γεγονότα. Η εισαγωγή αυτής της τυχαίας διακύμανσης αποτρέπει τη δημιουργία τεχνητά ομαλών δεδομένων και οδηγεί σε πιο ρεαλιστικές χρονοσειρές, γεγονός που συνάδει με τη βιβλιογραφία πρόβλεψης ζήτησης, η οποία αναδεικνύει τη μεταβλητότητα και την αβεβαιότητα της καταναλωτικής συμπεριφοράς.

3.8. Κλειστές ημέρες και μεταφορά ζήτησης

Κατά τις ημέρες που το κατάστημα θεωρείται κλειστό δεν πραγματοποιούνται πωλήσεις. Ωστόσο, η ζήτηση των ημερών αυτών δεν χάνεται πλήρως αλλά μεταφέρεται στις προηγούμενες ημέρες λειτουργίας.

Η μεταφορά της ζήτησης κατανέμεται στις δύο προηγούμενες ανοιχτές ημέρες με συντελεστές 70% και 30%. Με τον τρόπο αυτό προσομοιώνεται η συμπεριφορά των καταναλωτών να πραγματοποιούν αγορές πριν από ημέρες κατά τις οποίες τα καταστήματα παραμένουν κλειστά.

3.9. Διαχείριση αποθεμάτων και αλλοίωση προϊόντων

Η διαχείριση αποθεμάτων προσομοιώνεται μέσω πολιτικής αναπλήρωσης τύπου order-up-to. Κάθε προϊόν διατηρεί απόθεμα που αντιστοιχεί σε συγκεκριμένο αριθμό ημερών ζήτησης.

Σε τακτά χρονικά διαστήματα πραγματοποιούνται παραλαβές προϊόντων με στόχο την επαναφορά του αποθέματος στο επιθυμητό επίπεδο. Σε περίπτωση που μια προγραμματισμένη παραλαβή συμπίπτει με ημέρα κατά την οποία το κατάστημα είναι κλειστό, η παραλαβή μεταφέρεται στην επόμενη ημέρα λειτουργίας.

Όταν η ζήτηση υπερβαίνει το διαθέσιμο απόθεμα καταγράφονται χαμένες πωλήσεις, οι οποίες αποτελούν ένδειξη εμφάνισης stockout.

Παράλληλα, στο σύστημα ενσωματώνεται μηχανισμός αλλοίωσης προϊόντων. Η αλλοίωση μοντελοποιείται ως καθημερινή διαδικασία, με διαφορετικούς ρυθμούς για κάθε κατηγορία προϊόντος. Προϊόντα με μικρή διάρκεια ζωής, όπως τα ψάρια ή τα προϊόντα αρτοποιείου, εμφανίζουν υψηλότερο ρυθμό αλλοίωσης σε σύγκριση με προϊόντα μεγάλης διάρκειας ζωής.

Η ποσότητα προϊόντων που αλλοιώνεται υπολογίζεται μέσω στοχαστικής διαδικασίας και αφαιρείται από το διαθέσιμο απόθεμα. Για κάθε περίπτωση αλλοίωσης καταγράφεται η ποσότητα προϊόντων που αποσύρεται, ο λόγος απόσυρσης και το αντίστοιχο κόστος.

Οι πληροφορίες αυτές αποθηκεύονται στο αρχείο `waste_daily.csv`, επιτρέποντας τη σύνδεση της αλλοίωσης με τα δεδομένα πωλήσεων και αποθέματος και τη συνολική αξιολόγηση της λειτουργίας του συστήματος.

3.10. Σύνοψη

Το συνθετικό σύνολο δεδομένων που δημιουργήθηκε στο πλαίσιο της παρούσας εργασίας προσομοιώνει τη λειτουργία ενός καταστήματος σούπερ μάρκετ σε ημερήσια χρονική ανάλυση. Η ενσωμάτωση χαρακτηριστικών όπως εποχικότητα ζήτησης, επιδράσεις αργιών και εμπορικών γεγονότων, δυναμική τιμολόγηση, περιορισμοί αποθέματος και αλλοίωση προϊόντων δημιουργεί ένα ρεαλιστικό περιβάλλον προσομοίωσης για την ελληνική αγορά. Το συγκεκριμένο σύνολο συνθετικών δεδομένων χρησιμοποιείται στα επόμενα στάδια της εργασίας για την εφαρμογή μεθόδων πρόβλεψης ζήτησης, πολυκριτηριακής ανάλυσης ABC

και τεχνικών Reinforcement Learning με στόχο τη μείωση σπατάλης τροφίμων και τη βελτιστοποίηση των πολιτικών διαχείρισης αποθεμάτων.

4. Μεθοδολογία

Στην παρούσα ενότητα περιγράφονται τα εργαλεία και το λογισμικό που χρησιμοποιήθηκαν για την υλοποίηση του πειράματος καθώς και η αναλυτική πορεία που ακολουθήθηκε.

4.1. Εργαλεία και λογισμικό

Αυτή η ενότητα αφορά τα εργαλεία και το λογισμικό που χρησιμοποιήθηκαν για την υλοποίηση της παρούσας εργασίας και θα τα περιγράψει συνοπτικά σε υποενότητες.

4.1.1. Συσκευή

Για την αποπεράτωση του πειράματος και την όλη την συγγραφή της παρούσας εργασίας χρησιμοποιήθηκε φορητός υπολογιστής (Laptop) μάρκας Lenovo με τα εξής χαρακτηριστικά:

- Επεξεργαστής: AMD Ryzen 5 5500U με Radeon Graphics 2.10 GHz

- Εγκατεστημένη RAM: 8,00 GB
- Χώρος αποθήκευσης: 238 GB SSD SAMSUNG MZALQ256HBJD-00BL2
- Κάρτα γραφικών: AMD Radeon(TM) Graphics (2 GB)
- Τύπος συστήματος: Λειτουργικό σύστημα 64 bit , Windows 10 Pro , με έκδοση: 22H2

4.1.2. Γλώσσες προγραμματισμού και βιβλιοθήκες που χρησιμοποιήθηκαν

4.1.2.1. Περιβάλλον ανάπτυξης λογισμικού (IDE)

Για την ανάπτυξη, εκτέλεση και αποσφαλμάτωση (debugging) του κώδικα χρησιμοποιήθηκε το ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment – IDE) PyCharm Community Edition της JetBrains.

Το PyCharm παρέχει εργαλεία διαχείρισης έργων Python, αυτόματη συμπλήρωση κώδικα, έλεγχο συντακτικών σφαλμάτων, δυνατότητες αποσφαλμάτωσης και ενσωματωμένη διαχείριση βιβλιοθηκών, διευκολύνοντας σημαντικά την ανάπτυξη σύνθετων εφαρμογών ανάλυσης δεδομένων και μηχανικής μάθησης.

Στην παρούσα εργασία το PyCharm χρησιμοποιήθηκε για την υλοποίηση των σεναρίων δημιουργίας συνθετικών δεδομένων, της πρόβλεψης ζήτησης με NeuralProphet, της πολυκριτηριακής ABC ανάλυσης καθώς και της ανάπτυξης και εκπαίδευσης του περιβάλλοντος RL. Η χρήση ενός ολοκληρωμένου περιβάλλοντος ανάπτυξης συνέβαλε στη βελτίωση της παραγωγικότητας και στην αποτελεσματικότερη διαχείριση του πηγαίου κώδικα.

4.1.2.2. Python 3.13.13

Η ανάπτυξη και η εκτέλεση της παρούσας εργασίας πραγματοποιήθηκε στη γλώσσα προγραμματισμού Python έκδοσης 3.13.13, η οποία επιλέχθηκε λόγω της εκτεταμένης υποστήριξης που παρέχει σε εφαρμογές επιστήμης δεδομένων, μηχανικής μάθησης και ενισχυτικής μάθησης. Η Python αποτελεί μία από τις πλέον διαδεδομένες γλώσσες στον χώρο της ανάλυσης δεδομένων και της τεχνητής νοημοσύνης, καθώς διαθέτει μεγάλο

αριθμό βιβλιοθηκών για επεξεργασία δεδομένων, forecasting και ανάπτυξη αλγορίθμων μηχανικής μάθησης.

4.1.2.3. Pandas 2.3.3

Για την επεξεργασία και διαχείριση των δεδομένων χρησιμοποιήθηκε η βιβλιοθήκη Pandas έκδοσης 2.3.3. Η βιβλιοθήκη αξιοποιήθηκε για την παραγωγή δεδομένων, τη φόρτωση των αρχείων CSV, τη συγχώνευση πινάκων δεδομένων, τον μετασχηματισμό χρονοσειρών και τον υπολογισμό των επιχειρησιακών δεικτών απόδοσης (KPIs). Η Pandas επιλέχθηκε επειδή παρέχει αποδοτικές δομές δεδομένων και εργαλεία κατάλληλα για ανάλυση μεγάλου όγκου επιχειρησιακών δεδομένων.

4.1.2.4. Numpy 1.26.4

Για αριθμητικές πράξεις και στοχαστική προσομοίωση χρησιμοποιήθηκε η βιβλιοθήκη NumPy έκδοσης 1.26.4. Η συγκεκριμένη βιβλιοθήκη χρησιμοποιήθηκε κυρίως για τη δημιουργία τυχαίων μεταβλητών, τον υπολογισμό seasonal multipliers και την υλοποίηση μαθηματικών συναρτήσεων που σχετίζονται με την προσομοίωση της ζήτησης. Η χρήση της NumPy κρίθηκε απαραίτητη λόγω της υψηλής υπολογιστικής απόδοσης που παρέχει σε vectorized numerical operations.

4.1.2.5. Matplotlib 3.10.8

Η οπτικοποίηση των αποτελεσμάτων πραγματοποιήθηκε μέσω της βιβλιοθήκης Matplotlib έκδοσης 3.8.4. Η βιβλιοθήκη χρησιμοποιήθηκε για τη δημιουργία learning curves, forecasting plots και comparative charts μεταξύ πραγματικής και προβλεπόμενης ζήτησης.

4.1.2.6. NeuralProphet 0.8.0

Για την πρόβλεψη της ζήτησης χρησιμοποιήθηκε η βιβλιοθήκη NeuralProphet έκδοσης 0.8.0. Το NeuralProphet αποτελεί ένα σύγχρονο υβριδικό μοντέλο πρόβλεψης

χρονοσειρών, το οποίο αναπτύχθηκε ως εξέλιξη του Prophet της Meta (Facebook), ενσωματώνοντας παράλληλα τεχνικές βαθιάς μάθησης (Triebe et al., 2021). Ο βασικός στόχος του είναι η ακριβής πρόβλεψη μελλοντικών τιμών χρονοσειρών μέσω της συνδυασμένης μοντελοποίησης τάσεων, εποχικότητας και αυτοπαλινδρόμησης.

Η αρχιτεκτονική του NeuralProphet βασίζεται σε τρεις κύριους μηχανισμούς. Αρχικά, μοντελοποιεί τη μακροχρόνια τάση (trend) της χρονοσειράς, επιτρέποντας την αποτύπωση σταδιακών μεταβολών στη ζήτηση κατά τη διάρκεια του χρόνου. Στη συνέχεια, ενσωματώνει πολλαπλά εποχικά μοτίβα (seasonality), όπως εβδομαδιαίες ή ετήσιες επαναλήψεις, τα οποία εμφανίζονται συχνά σε δεδομένα λιανικού εμπορίου. Τέλος, χρησιμοποιεί μηχανισμό αυτοπαλινδρόμησης (autoregression), μέσω του οποίου οι προηγούμενες τιμές της χρονοσειράς αξιοποιούνται για την πρόβλεψη των μελλοντικών τιμών (Triebe et al., 2021).

Η επιλογή του NeuralProphet πραγματοποιήθηκε επειδή συνδυάζει τα πλεονεκτήματα των παραδοσιακών μοντέλων χρονοσειρών με τις δυνατότητες των νευρωνικών δικτύων. Σε αντίθεση με τα κλασικά στατιστικά μοντέλα, όπως τα ARIMA και Exponential Smoothing, το NeuralProphet μπορεί να αποτυπώσει μη γραμμικές σχέσεις στα δεδομένα. Παράλληλα, σε σύγκριση με πιο σύνθετες αρχιτεκτονικές βαθιάς μάθησης, όπως τα LSTM και GRU δίκτυα, διατηρεί υψηλό βαθμό ερμηνευσιμότητας και απαιτεί μικρότερη διαδικασία παραμετροποίησης (Triebe et al., 2021).

Η βιβλιοθήκη επιλέχθηκε επίσης επειδή είναι ιδιαίτερα κατάλληλη για δεδομένα λιανικού εμπορίου, όπου η ζήτηση επηρεάζεται από εποχικότητα, ημερολογιακά γεγονότα, προωθητικές ενέργειες και τυχαίες διακυμάνσεις. Η δυνατότητα ενσωμάτωσης πολλαπλών συνιστωσών σε ένα ενιαίο μοντέλο επιτρέπει τη δημιουργία πιο αξιόπιστων προβλέψεων σε σχέση με απλούστερες προσεγγίσεις.

Στην παρούσα εργασία το NeuralProphet χρησιμοποιήθηκε για την πρόβλεψη της ημερήσιας ζήτησης κάθε προϊόντος (SKU) ξεχωριστά. Για κάθε χρονοσειρά πραγματοποιήθηκε χρονικός διαχωρισμός των δεδομένων σε σύνολο εκπαίδευσης (80%) και σύνολο ελέγχου (20%), ώστε η αξιολόγηση να πραγματοποιείται σε μελλοντικές χρονικές περιόδους που δεν είχαν χρησιμοποιηθεί κατά την εκπαίδευση του μοντέλου. Η διαδικασία αυτή προσομοιώνει ρεαλιστικά τις πραγματικές συνθήκες λειτουργίας ενός συστήματος πρόβλεψης ζήτησης.

Οι προβλέψεις που παρήχθησαν αξιολογήθηκαν με κατάλληλους δείκτες σφάλματος και χρησιμοποιήθηκαν για τη μελέτη της συμπεριφοράς της ζήτησης των προϊόντων. Η ακριβής πρόβλεψη της μελλοντικής ζήτησης αποτελεί κρίσιμο παράγοντα για τη βελτιστοποίηση της διαχείρισης αποθεμάτων, καθώς συμβάλλει στη μείωση των ελλείψεων προϊόντων, στον περιορισμό των υπεραποθεμάτων και τελικά στη μείωση της σπατάλης τροφίμων (Hyndman, R. J., & Athanasopoulos, G. 2021).

4.1.2.7. Scipy 1.18.0

Για τον υπολογισμό της στατιστικής σημαντικότητας των συσχετίσεων χρησιμοποιήθηκε η βιβλιοθήκη SciPy. Συγκεκριμένα, αξιοποιήθηκε η συνάρτηση «pearsonr» του πακέτου scipy.stats για τον υπολογισμό του συντελεστή συσχέτισης Pearson και των αντίστοιχων p-values μεταξύ των μεταβλητών που χρησιμοποιήθηκαν στην πολυκριτηριακή ABC ανάλυση. Η χρήση της SciPy επέτρεψε την αξιολόγηση όχι μόνο της έντασης των συσχετίσεων αλλά και της στατιστικής σημαντικότητάς τους.

4.1.2.8. Ο αλγόριθμος Q – Learning

Για την υλοποίηση της ενισχυτικής μάθησης χρησιμοποιήθηκε ο αλγόριθμος Q-Learning. Το Q-Learning αποτελεί έναν από τους πιο διαδεδομένους αλγορίθμους ενισχυτικής μάθησης και ανήκει στην κατηγορία των model free μεθόδων, καθώς δεν απαιτεί εκ των προτέρων γνώση του περιβάλλοντος στο οποίο δρα ο πράκτορας (C. J. C. H. Watkins & Dayan, 1992).

Η βασική αρχή λειτουργίας του αλγορίθμου είναι η εκμάθηση μιας συνάρτησης αξίας ενεργειών (Q-function), η οποία εκτιμά το αναμενόμενο σωρευτικό όφελος που προκύπτει από την επιλογή μιας συγκεκριμένης ενέργειας σε μια δεδομένη κατάσταση. Μέσω επαναλαμβανόμενων αλληλεπιδράσεων με το περιβάλλον, ο πράκτορας ενημερώνει σταδιακά τις τιμές του πίνακα Q και βελτιώνει την πολιτική λήψης αποφάσεων που ακολουθεί.

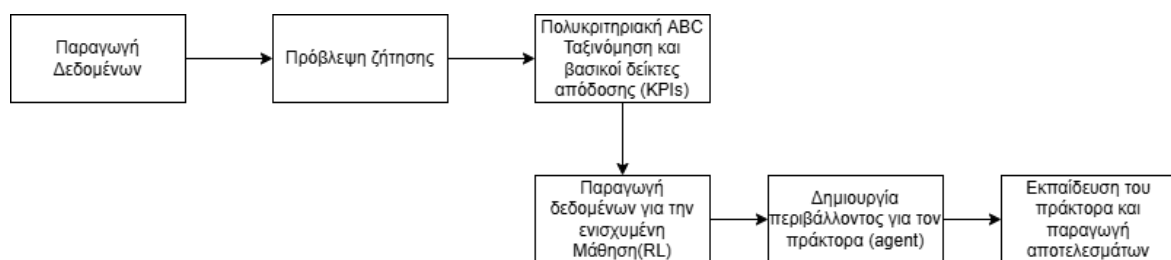
Η επιλογή του Q-Learning πραγματοποιήθηκε επειδή αποτελεί μία ώριμη και καλά τεκμηριωμένη προσέγγιση για προβλήματα διαδοχικής λήψης αποφάσεων, ενώ παράλληλα

προσφέρει υψηλό βαθμό ερμηνευσιμότητας των αποτελεσμάτων. Επιπλέον, είναι κατάλληλος για περιβάλλοντα με διακριτό χώρο καταστάσεων και ενεργειών, όπως το περιβάλλον δυναμικής τιμολόγησης που αναπτύχθηκε στην παρούσα εργασία.

Στο πλαίσιο της έρευνας, ο αλγόριθμος χρησιμοποιήθηκε για την εκμάθηση πολιτικών δυναμικής τιμολόγησης προϊόντων με στόχο τη μεγιστοποίηση της κερδοφορίας και τη μείωση της σπατάλης τροφίμων. Ο πράκτορας εκπαιδεύτηκε μέσω επαναλαμβανόμενων επεισοδίων αλληλεπίδρασης με ένα προσομοιωμένο περιβάλλον λιανικού εμπορίου, στο οποίο οι αποφάσεις τιμολόγησης επηρεάζουν τη ζήτηση, τις πωλήσεις, τις ελλείψεις αποθέματος και τις απώλειες λόγω αλλοίωσης προϊόντων.

4.2. Πορεία που ακολουθήθηκε

Σε αυτή την ενότητα θα αναλύεται η πορεία που ακολουθήθηκε. Στο παρακάτω σχήμα φαίνονται συνοπτικά τα βήματα που έγιναν με τη σειρά, τα οποία θα αναλυθούν περαιτέρω σε υποενότητες πλην της παραγωγής δεδομένων που αναλύθηκε παραπάνω.



Εικόνα 4.1 Διάγραμμα ροής μεθοδολογίας της παρούσας εργασίας

4.2.1. Πρόβλεψη ζήτησης

Η πρόβλεψη ζήτησης πραγματοποιήθηκε σε επίπεδο προϊόντος (SKU-level forecasting), με στόχο την εκτίμηση των μελλοντικών πωλήσεων για κάθε κωδικό προϊόντος ξεχωριστά.

Για την ανάπτυξη του μοντέλου χρησιμοποιήθηκε η βιβλιοθήκη NeuralProphet, η οποία αποτελεί επέκταση του Prophet και συνδυάζει κλασικές τεχνικές χρονοσειρών με νευρωνικά δίκτυα.

Το μοντέλο εκπαιδεύτηκε χρησιμοποιώντας ημερήσια δεδομένα πωλήσεων, όπου κάθε εγγραφή αντιστοιχεί στις πωλήσεις ενός προϊόντος για μία συγκεκριμένη ημερομηνία. Η μεταβλητή στόχος ήταν η ποσότητα πωλήσεων (`units_sold`), ενώ κάθε προϊόν αναγνωριζόταν μέσω του μοναδικού αναγνωριστικού (`product_id`).

4.2.1.1. Προεπεξεργασία δεδομένων

Αρχικά πραγματοποιήθηκε ανάγνωση του αρχείου συναλλαγών και έλεγχος εγκυρότητας των βασικών πεδίων, όπως:

- Ημερομηνία (`date`)
- Κωδικός προϊόντος (`product_id`)
- Πωλήσεις (`units_sold`)

Οι ημερομηνίες μετατράπηκαν σε μορφή `datetime`, ενώ οι τιμές πωλήσεων μετατράπηκαν σε αριθμητική μορφή. Εγγραφές με ελλιπή ή μη έγκυρα δεδομένα απορρίφθηκαν.

Στη συνέχεια δημιουργήθηκαν ημερήσιες χρονοσειρές για κάθε SKU ξεχωριστά. Για τη διατήρηση συνεχούς χρονικού άξονα εφαρμόστηκε διαδικασία επαναδεικτοδότησης (`reindexing`), ώστε να περιλαμβάνονται όλες οι ημερομηνίες του χρονικού διαστήματος ανάλυσης. Οι ημέρες χωρίς πωλήσεις συμπληρώθηκαν με τιμή μηδέν.

4.2.1.2. Διαχωρισμός Train/Test

Για την αξιολόγηση του μοντέλου χρησιμοποιήθηκε χρονικός διαχωρισμός `train/test` (εκπαίδευσης και δοκιμής) σε αναλογία 80/20. Συγκεκριμένα, το 80% των παλαιότερων ημερομηνιών χρησιμοποιήθηκε για εκπαίδευση ενώ το υπόλοιπο 20% χρησιμοποιήθηκε για έλεγχο και αξιολόγηση.

Ο διαχωρισμός πραγματοποιήθηκε χρονολογικά και όχι τυχαία, ώστε να προσομοιώνεται ρεαλιστικά η διαδικασία πρόβλεψης μελλοντικής ζήτησης.

Το forecast horizon (ορίζοντας πρόβλεψης) του μοντέλου ορίστηκε ίσο με το μήκος του test set, ώστε να αξιολογηθεί πλήρως η ικανότητα πρόβλεψης σε όλη την περίοδο ελέγχου.

4.2.1.3. Μοντέλο NeuralProphet

Για την πρόβλεψη χρησιμοποιήθηκε το global NeuralProphet model, το οποίο εκπαιδεύτηκε ταυτόχρονα σε όλες τις χρονοσειρές προϊόντων. Η προσέγγιση αυτή επιτρέπει στο μοντέλο να εντοπίζει κοινά μοτίβα συμπεριφοράς μεταξύ διαφορετικών προϊόντων.

Το μοντέλο παραμετροποιήθηκε με:

- Yearly seasonality = True
- Weekly seasonality = True
- Daily seasonality = False
- Learning rate = 0.01
- Epochs = 200

Η εκπαίδευση πραγματοποιήθηκε σε ημερήσια συχνότητα (freq="D").

4.2.1.4. Αξιολόγηση του μοντέλου

Η αξιολόγηση του μοντέλου πραγματοποιήθηκε συγκρίνοντας τις πραγματικές τιμές πωλήσεων με τις προβλεπόμενες τιμές για το σύνολο του test set.

Υπολογίστηκαν οι ακόλουθες μετρικές:

- Μέσο απόλυτο σφάλμα (Mean Absolute Error, MAE)
- Ρίζα μέσου τετραγωνικού σφάλματος (Root Mean Squared Error, RMSE)
- Μέσο απόλυτο ποσοστιαίο σφάλμα (Mean Absolute Percentage Error, MAPE)
- Συμμετρικό Μέσο Απόλυτο Ποσοστιαίο Σφάλμα (Symmetric Mean Absolute Percentage Error, sMAPE)

Οι μετρικές υπολογίστηκαν τόσο ανά προϊόν όσο και συνολικά για όλα τα SKU. Επιπλέον υπολογίστηκε ο συντελεστής μεταβλητότητας (Coefficient of Variation - CV), ο οποίος ορίζεται ως:

$$CV = \frac{\sigma}{\mu}$$

όπου:

- σ είναι η τυπική απόκλιση της ζήτησης
- μ είναι ο μέσος όρος της ζήτησης

Ο δείκτης CV χρησιμοποιήθηκε για την αξιολόγηση της σταθερότητας της ζήτησης κάθε προϊόντος.

4.2.1.5. Οπτικοποίηση των αποτελεσμάτων

Για την οπτική αξιολόγηση των αποτελεσμάτων δημιουργήθηκαν γραφήματα σύγκρισης πραγματικής και προβλεπόμενης ζήτησης. Τα γραφήματα περιλαμβάνουν:

- actual vs forecast plots
- line-to-line comparison plots

Τα διαγράμματα αποθηκεύτηκαν σε μορφή PNG για περαιτέρω ανάλυση και παρουσίαση των αποτελεσμάτων.

4.2.2. Πολυκριτηριακή ABC Ανάλυση και ταξινόμηση προϊόντων

Συνοπτικά όπως ειπώθηκε και νωρίτερα, η παραδοσιακή ABC ανάλυση βασίζεται συνήθως αποκλειστικά στην αξία πωλήσεων ή στη συμβολή κάθε προϊόντος στον συνολικό κύκλο εργασιών. Ωστόσο, σε περιβάλλοντα λιανικής τροφίμων και ευπαθών προϊόντων, η σημασία ενός προϊόντος δεν καθορίζεται μόνο από τις πωλήσεις του, αλλά και από παράγοντες όπως η μεταβλητότητα της ζήτησης, η πιθανότητα αλλοίωσης, η συχνότητα stockout και η ευαισθησία σε ειδικά γεγονότα (events).

Για τον λόγο αυτό αναπτύχθηκε ένα πολυκριτηριακό σύστημα ABC κατηγοριοποίησης, το οποίο συνδυάζει πολλαπλούς δείκτες απόδοσης (Key Performance Indicators - KPIs), με στόχο την ακριβέστερη αξιολόγηση της επιχειρησιακής σημαντικότητας κάθε προϊόντος. Σε αυτή την υποενότητα, θα αναλυθεί η διαδικασία και η λογική που αφορά την Πολυκριτηριακή ταξινόμηση και ανάλυση των προϊόντων.

4.2.2.1. Συλλογή και ενοποίηση δεδομένων

Για την ανάπτυξη του μοντέλου χρησιμοποιήθηκαν τέσσερα βασικά σύνολα δεδομένων:

- δεδομένα προϊόντων (products.csv)
- ημερήσιες συναλλαγές (transactions_daily.csv)
- δεδομένα απορρίψεων και αλλοιώσεων (waste_daily.csv)
- ημερολόγιο ειδικών γεγονότων (calendar_events.csv)

Τα δεδομένα ενοποιήθηκαν μέσω των πεδίων «product_id» και «date», επιτρέποντας τη δημιουργία ενός ολοκληρωμένου πίνακα χαρακτηριστικών ανά προϊόν.

Οι ημερομηνίες μετατράπηκαν σε μορφή datetime ώστε να καταστεί δυνατή η χρονική συσχέτιση των δεδομένων συναλλαγών με τα ειδικά γεγονότα του ημερολογίου.

4.2.2.2. Υπολογισμός πραγματικής ζήτησης

Η πραγματική ζήτηση (real demand) υπολογίστηκε ως το άθροισμα των πραγματοποιημένων πωλήσεων και των χαμένων πωλήσεων λόγω έλλειψης αποθέματος (lost sales).

Η σχέση αυτή δίνεται από:

$$Real\ Demand = Units\ Sold + Lost\ Sales$$

Η προσέγγιση αυτή επιτρέπει την καλύτερη αποτύπωση της πραγματικής καταναλωτικής ζήτησης, καθώς λαμβάνει υπόψη περιπτώσεις όπου η ζήτηση δεν καλύφθηκε εξαιτίας καταστάσεων εξάντλησης αποθέματος (stockout).

4.2.2.3. Υπολογισμός βασικών δεικτών απόδοσης (KPIs)

Για κάθε προϊόν υπολογίστηκε ένα σύνολο βασικών δεικτών απόδοσης (KPIs), μερικοί από τους οποίους χρησιμοποιήθηκαν ως είσοδοι στην πολυκριτηριακή ABC ανάλυση.

Οι βασικοί δείκτες περιλαμβάνουν:

- συνολική αξία πωλήσεων (sales value)
- μικτό κέρδος (gross profit)
- μέση ζήτηση (mean demand)
- τυπική απόκλιση ζήτησης (std demand)
- συνολική ζήτηση (total demand)
- συχνότητα stockout (stockout frequency)
- ποσοστό αλλοίωσης (spoilage rate)
- κόστος απορρίψεων (waste cost)
- διάρκεια ζωής προϊόντος (shelf life days)
- επιχειρησιακή κρισιμότητα (service criticality)

Ιδιαίτερη σημασία δόθηκε στη μεταβλητότητα της ζήτησης μέσω του συντελεστή μεταβλητότητας (Coefficient of Variation - CV), ο οποίος ορίζεται ως:

$$CV = \frac{\mu}{\sigma}$$

όπου:

- σ είναι η τυπική απόκλιση της ζήτησης,
- μ είναι ο μέσος όρος της ζήτησης.

Υψηλές τιμές CV υποδηλώνουν ασταθή ή δυσκολότερα προβλέψιμη ζήτηση.

4.2.2.4. Τόνωση ζήτησης λόγω γεγονότος

Για την αποτύπωση της επίδρασης ειδικών γεγονότων στη ζήτηση, υπολογίστηκε δείκτης τόνωσης ζήτησης ανά υποκατηγορία προϊόντων (*subcategory_event_uplift*).

Ο δείκτης αυτός εκφράζει το ποσοστό αύξησης της μέσης ζήτησης κατά τη διάρκεια ειδικών γεγονότων σε σχέση με τις κανονικές ημέρες λειτουργίας. Παραδείγματος χάρη τα κρέατα έχουν μεγάλη αύξηση την τσικνοπέμπτη, ενώ τα θαλασσινά την Καθαρά Δευτέρα.

Η σχέση υπολογισμού δίνεται από:

$$Uplift = \frac{Demand_{event}}{Demand_{normal}}$$

Για κάθε υποκατηγορία διατηρήθηκε η μέγιστη παρατηρούμενη τιμή τόνωσης (*uplift*) μεταξύ όλων των ειδικών γεγονότων. Η προσέγγιση αυτή επιτρέπει την αναγνώριση προϊόντων που εμφανίζουν έντονες αυξήσεις ζήτησης σε περιόδους αυξημένης κατανάλωσης, όπως εορτές ή εποχικά γεγονότα.

4.2.2.5. Ανάλυση συσχέτισης μεταβλητών

Πριν την τελική εφαρμογή της πολυκριτηριακής ανάλυσης πραγματοποιήθηκε έλεγχος συσχέτισης μεταξύ των μεταβλητών μέσω του συντελεστή Pearson. Παράλληλα υπολογίστηκαν τα αντίστοιχα *p-values* με σκοπό τον έλεγχο της στατιστικής σημαντικότητας των παρατηρούμενων συσχετίσεων.

Στόχος της διαδικασίας ήταν η αποφυγή υπερβολικά συσχετισμένων χαρακτηριστικών, τα οποία θα μπορούσαν να οδηγήσουν σε διπλή καταμέτρηση της ίδιας πληροφορίας κατά τον υπολογισμό του τελικού score.

Μεταβλητές με πολύ υψηλή συσχέτιση (correlation > 0.90) θεωρήθηκαν υποψήφιες για αφαίρεση ή επαναξιολόγηση.

4.2.2.6. Κανονικοποίηση δεδομένων

Λόγω της διαφορετικής κλίμακας των KPIs εφαρμόστηκε διαδικασία κανονικοποίησης (Min-Max normalization), ώστε όλες οι μεταβλητές να μετατραπούν σε κοινή κλίμακα τιμών μεταξύ 0 και 1.

Η κανονικοποίηση δίνεται από τον τύπο:

$$X_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Για τη μεταβλητή διάρκειας ζωής (shelf_life_days) εφαρμόστηκε αντίστροφη κανονικοποίηση, καθώς τα προϊόντα με μικρότερη διάρκεια ζωής θεωρούνται επιχειρησιακά πιο κρίσιμα.

4.2.2.7. Σύστημα βαρών

Μετά την κανονικοποίηση εφαρμόστηκε σύστημα βαρών για την αποτύπωση της σχετικής σημαντικότητας κάθε KPI στον τελικό υπολογισμό του ABC score (Πίνακας 4.1).

Τα βάρη καθορίστηκαν εμπειρικά με βάση επιχειρησιακά κριτήρια και τη σημασία κάθε μεταβλητής στη διαχείριση αποθεμάτων ευπαθών προϊόντων και στη λήψη αποφάσεων αναπλήρωσης.

KPIs	Βάρος
------	-------

Μικτό κέρδος (Gross profit)	0,23
Συντελεστής μεταβλητότητας ζήτησης (Cv demand)	0,20
Επίδραση ειδικών γεγονότων (Event Uplift)	0,15
Ποσοστό αλλοίωσης (Spoilage rate)	0,15
Διάρκεια ζωής προϊόντος (Shelf life)	0,10
Κόστος απορρίψεων (Waste cost)	0,10
Επιχειρησιακή κρισιμότητα (Service criticality)	0,05
Συχνότητα εξάντλησης αποθέματος (Stockout frequency)	0,02

Πίνακας 4.1 Βάρη μεταβλητών για την ABC ανάλυση

Το μεγαλύτερο βάρος αποδόθηκε στο μικτό κέρδος (23%), καθώς αποτελεί τον σημαντικότερο οικονομικό δείκτη συνεισφοράς ενός προϊόντος στην κερδοφορία της επιχείρησης. Παράλληλα, ιδιαίτερη έμφαση δόθηκε στη μεταβλητότητα της ζήτησης (20%), καθώς τα προϊόντα με υψηλή διακύμανση απαιτούν μεγαλύτερη προσοχή στον σχεδιασμό αποθεμάτων και στις διαδικασίες πρόβλεψης.

Τα βάρη ειδικών γεγονότων και ποσοστού αλλοίωσης έλαβαν βάρος 15% έκαστος, επειδή επηρεάζουν άμεσα τόσο τη διαθεσιμότητα όσο και τις απώλειες προϊόντων. Η διάρκεια ζωής και το κόστος απορρίψεων συμμετέχουν με βάρος 10%, αντανακλώντας τη σημασία τους στη διαχείριση ευπαθών προϊόντων.

Τέλος, η επιχειρησιακή κρισιμότητα και η συχνότητα εμφάνισης stockout έλαβαν μικρότερα βάρη (5% και 2% αντίστοιχα), καθώς χρησιμοποιούνται κυρίως ως συμπληρωματικοί δείκτες διαφοροποίησης μεταξύ προϊόντων.

4.2.2.8. Υπολογισμός ABC Score

Το τελικό ABC score κάθε προϊόντος υπολογίστηκε ως σταθμισμένο άθροισμα των κανονικοποιημένων KPIs.

Η γενική μορφή του υπολογισμού είναι:

$$ABC_{score} = \sum_{i=1}^n w_i x_i$$

όπου:

- w_i είναι το βάρος του KPI,
- x_i η κανονικοποιημένη τιμή του KPI.

Στη συνέχεια τα προϊόντα ταξινομήθηκαν κατά φθίνουσα σειρά score και υπολογίστηκε το αθροιστικό ποσοστό συμμετοχής κάθε προϊόντος στο συνολικό score.

4.2.2.9. Κατηγοριοποίηση A/B/C

Η τελική κατηγοριοποίηση πραγματοποιήθηκε μέσω ποσοστημορίων (quantiles) του ABC score.

Συγκεκριμένα, τα προϊόντα με score άνω του 80ού ποσοστημορίου ταξινομήθηκαν στην κατηγορία A, τα προϊόντα μεταξύ 50ού και 80ού ποσοστημορίου στην κατηγορία B, ενώ τα υπόλοιπα προϊόντα ταξινομήθηκαν στην κατηγορία C.

Η προσέγγιση αυτή επιτρέπει την πολυδιάστατη αξιολόγηση των προϊόντων, λαμβάνοντας υπόψη όχι μόνο οικονομικά αλλά και λειτουργικά χαρακτηριστικά της εφοδιαστικής αλυσίδας.

4.2.3. Δημιουργία συνόλου δεδομένων για ενισχυτική μάθηση

Μετά την ολοκλήρωση της πολυκριτηριακής ABC ανάλυσης, δημιουργήθηκε ένα ολοκληρωμένο σύνολο δεδομένων με στόχο τη χρήση του σε περιβάλλον ενισχυτικής μάθησης (RL). Σκοπός του συγκεκριμένου σταδίου ήταν η ενοποίηση των διαθέσιμων πληροφοριών και η μετατροπή τους σε χαρακτηριστικά κατάλληλα για την αναπαράσταση της κατάστασης (state representation) του πράκτορα.

Το τελικό σύνολο δεδομένων συνδυάζει πληροφορίες πωλήσεων, απωλειών, απορρίψεων, ειδικών γεγονότων και αποτελεσμάτων της ABC κατηγοριοποίησης, παρέχοντας μια ολοκληρωμένη εικόνα της λειτουργικής κατάστασης κάθε προϊόντος.

Σε αυτή την υποενότητα θα αναλυθεί η διαδικασία της δημιουργίας του συγκεκριμένου σετ δεδομένων.

4.2.3.1. Ενοποίηση δεδομένων

Για τη δημιουργία του RL συνόλου δεδομένων πραγματοποιήθηκε συγχώνευση δεδομένων από πολλαπλές πηγές:

- προϊόντα (products.csv)
- συναλλαγές (transactions_daily.csv)
- απορρίψεις και αλλοιώσεις (waste_daily.csv)
- ημερολόγιο ειδικών γεγονότων (calendar_events.csv)
- αποτελέσματα πολυκριτηριακής ABC ανάλυσης (abc_results.csv)

Οι συγχωνεύσεις πραγματοποιήθηκαν μέσω των πεδίων «product_id» και «date», δημιουργώντας μία ενιαία εγγραφή για κάθε προϊόν και ημερομηνία ώστε να είναι δυνατή η χρονική επεξεργασία των δεδομένων.

4.2.3.2. Εμπλουτισμός χαρακτηριστικών

Το βασικό σύνολο συναλλαγών εμπλουτίστηκε μέσω διαδοχικών συγχωνεύσεων (joins) με:

- την κατηγορία ABC κάθε προϊόντος,
- την κατηγορία κάθε προϊόντος,
- τον ρυθμό αλλοίωσης (spoilage rate),
- την επιχειρησιακή κρισιμότητα (service criticality),
- τη διάρκεια ζωής (shelf life),
- τους δείκτες επίδρασης γεγονότων (event uplift),
- τις ποσότητες αποβλήτων και το αντίστοιχο κόστος για κάθε προϊόν ,

- τα ημερολογιακά γεγονότα.

Οι ελλείπουσες τιμές στα δεδομένα αποβλήτων αντικαταστάθηκαν με μηδενικές τιμές, θεωρώντας ότι η απουσία εγγραφής αντιστοιχεί σε μηδενική σπατάλη.

4.2.3.3. Υπολογισμός Χαρακτηριστικών κατάστασης και κατασκευή νέων μεταβλητών

Δημιουργήθηκαν επιπλέον χαρακτηριστικά για την αναπαράσταση της κατάστασης του περιβάλλοντος:

- Event Flag: δυαδική μεταβλητή που υποδεικνύει την ύπαρξη ειδικού γεγονότος.
- Day of Week: αριθμητική αναπαράσταση της ημέρας της εβδομάδας.

Αλλά και υπολογίστηκε η πραγματική ζήτηση :

$$RealDemand = UnitsSold + LostSales$$

ώστε να ληφθεί υπόψη τόσο η πραγματοποιηθείσα ζήτηση όσο και οι χαμένες πωλήσεις λόγω έλλειψης αποθέματος.

Οι μεταβλητές αυτές επιτρέπουν στο περιβάλλον RL να αναγνωρίζει χρονικά και λειτουργικά μοτίβα που επηρεάζουν τη συμπεριφορά της ζήτησης.

4.2.3.4. Μοντελοποίηση Ελαστικότητας ζήτησης ως προς την Τιμή

Ένα από τα βασικά χαρακτηριστικά που προστέθηκαν στο RL dataset είναι η ελαστικότητα ζήτησης ως προς την τιμή (price elasticity).

Η ελαστικότητα αποτυπώνει τον βαθμό μεταβολής της ζήτησης σε σχέση με μεταβολές της τιμής και ορίστηκε σε επίπεδο κατηγορίας προϊόντων. Για κάθε κατηγορία αποδόθηκε προκαθορισμένη τιμή ελαστικότητας με βάση τα χαρακτηριστικά της αγοράς και τη σχετική ευαισθησία των καταναλωτών στις μεταβολές τιμών.

Οι υψηλότερες απόλυτες τιμές αποδόθηκαν σε κατηγορίες όπως το κρέας και τα ψάρια, ενώ μικρότερες τιμές χρησιμοποιήθηκαν για βασικά είδη παντοπωλείου.

Η ελαστικότητα έχει αρνητικές τιμές επειδή υπάρχει αντίστροφη σχέση μεταξύ τιμής και ζητούμενης ποσότητας.

Ειδικότερα οι τιμές απεικονίζονται στον πίνακα (Πίνακας 4.2) παρακάτω:

Είδος	Ελαστικότητα
Κρέας, ψάρια	-1,5
Φρούτα, λαχανικά	-1,2
Γαλακτοκομικά και προϊόντα αρτοποιείου	-0,8
Προϊόντα Delicatessen	-0,6
Βασικά είδη παντοπωλείου	-0,5

Πίνακας 4.2 Ελαστικότητα ζήτησης ως προς την τιμή για τις κατηγορίες προϊόντων

Αυτό σημαίνει ότι για το Κρέας (-1.5) αν η τιμή αυξηθεί 1%, η ζήτηση αναμένεται να μειωθεί περίπου 1.5% και για τα Βασικά είδη παντοπωλείου (-0.5) αν η τιμή αυξηθεί 1%, η ζήτηση αναμένεται να μειωθεί μόνο 0.5% και αντιστρόφως ανάλογα.

Άρα τα κρέατα και τα ψάρια θεωρούνται πιο ευαίσθητα στις μεταβολές τιμών, ενώ τα βασικά είδη παντοπωλείου λιγότερο ευαίσθητα.

Η προσέγγιση αυτή επιτρέπει στο RL περιβάλλον να προσομοιώνει πιο ρεαλιστικά τις επιπτώσεις των αποφάσεων τιμολόγησης στη ζήτηση.

4.2.3.5. Μετασχηματισμός δεικτών γεγονότων

Ο δείκτης επίδρασης γεγονότων μετατράπηκε σε σήμα ενίσχυσης ζήτησης:

$$\text{Event Uplift Signal} = \text{Uplift Factor} - 1$$

Έτσι όταν η τιμή είναι 1 δεν έχει καμία επίδραση αλλά σε περίπτωση που έχει την τιμή 1.3 γίνεται 0.3 και η ζήτηση ενισχύεται κατά 30%.

4.2.3.6. Καθαρισμός και επικύρωση δεδομένων

Πριν τη δημιουργία του τελικού συνόλου δεδομένων πραγματοποιήθηκαν διαδικασίες ελέγχου ποιότητας και επικύρωσης.

Συγκεκριμένα τα δεδομένα ελέγχθηκαν για ελλιπείς τιμές (NaN), για αρνητικές τιμές ζήτησης και για μη έγκυρες τιμές τιμών πώλησης. Η ύπαρξη οποιασδήποτε μη αποδεκτής τιμής οδηγούσε σε διακοπή της διαδικασίας και παραγωγή κατάλληλου μηνύματος σφάλματος.

Επιπλέον εφαρμόστηκαν τεχνικές οριοθέτησης (clipping) στο ποσοστό αλλοίωσης, στην επιχειρησιακή κρισιμότητα και στην επίδραση ειδικών γεγονότων ώστε να διασφαλιστούν ρεαλιστικά όρια τιμών και μεγαλύτερη σταθερότητα κατά την εκπαίδευση του πράκτορα.

Πιο συγκεκριμένα, στην επιχειρησιακή κρισιμότητα στο διάστημα $[0, 0.5]$, στον ρυθμό αλλοίωσης στο διάστημα $[0, 1]$, και στην επιχειρησιακή κρισιμότητα στο διάστημα $[0, 1]$.

4.2.3.7. Δημιουργία τελικού σετ δεδομένων για την ενισχυτική μάθηση

Οι μεταβλητές επιλέχθηκαν ώστε να περιγράφουν ταυτόχρονα οικονομικές, λειτουργικές, χρονικές και επιχειρησιακές διαστάσεις κάθε προϊόντος. Το τελικό σύνολο δεδομένων περιλαμβάνει 17 μεταβλητές που θα χρησιμοποιηθούν ως είσοδος στο περιβάλλον ενισχυτικής μάθησης:

- ημερομηνία,
- κωδικό προϊόντος,
- κατηγορία ABC,
- κατηγορία προϊόντος,
- τιμή πώλησης,
- τις πραγματοποιημένες πωλήσεις,
- τις χαμένες πωλήσεις,
- την πραγματική ζήτηση,
- την ποσότητα απορρίψεων,
- το κόστος απορρίψεων,
- την ημέρα της εβδομάδας,
- τον δείκτη ειδικών γεγονότων,

- τη διάρκεια ζωής του προϊόντος,
- το ποσοστό αλλοίωσης,
- τον δείκτη επίδρασης ειδικών γεγονότων,
- την επιχειρησιακή κρισιμότητα,
- την ελαστικότητα τιμής.

Τέλος, οι εγγραφές ταξινομήθηκαν χρονικά ανά προϊόν ώστε να διατηρείται η χρονική συνέπεια των παρατηρήσεων, στοιχείο ιδιαίτερα σημαντικό για την εκπαίδευση αλγορίθμων ενισχυτικής μάθησης που βασίζονται σε αλληλουχίες αποφάσεων και καταστάσεων.

Το παραγόμενο σύνολο δεδομένων αποτέλεσε τη βάση για την κατασκευή του περιβάλλοντος ενισχυτικής μάθησης, στο οποίο κάθε εγγραφή χρησιμοποιείται για την αναπαράσταση της κατάστασης του συστήματος και τη λήψη αποφάσεων τιμολόγησης και διαχείρισης αποθεμάτων.

4.2.4. Σχεδιασμός Περιβάλλοντος Ενισχυτικής μάθησης

Για την αξιολόγηση στρατηγικών δυναμικής τιμολόγησης από τον πράκτορα αναπτύχθηκε ένα προσομοιωτικό περιβάλλον ενισχυτικής μάθησης (Reinforcement Learning Environment). Η ανάπτυξη του συγκεκριμένου περιβάλλοντος πραγματοποιήθηκε μέσω μιας ειδικής υλοποίησης (custom implementation) χωρίς τη χρήση εξειδικευμένων RL frameworks όπως «OpenAI Gym» ή «Stable Baselines». Η επιλογή αυτή έγινε ώστε να υπάρχει πλήρης έλεγχος της συνάρτησης ανταμοιβής, της προσομοίωσης διαχείρισης αποθεμάτων και της δυναμικής συμπεριφοράς του supermarket περιβάλλοντος. Το περιβάλλον αναπαριστά τη λειτουργία ενός καταστήματος λιανικής πώλησης (supermarket), όπου η τιμή των προϊόντων προσαρμόζεται δυναμικά με στόχο τη μεγιστοποίηση της κερδοφορίας, τη μείωση των απωλειών λόγω αλλοίωσης προϊόντων και τη διατήρηση ικανοποιητικού επιπέδου εξυπηρέτησης των πελατών.

Η λειτουργία του περιβάλλοντος βασίζεται σε διαδοχικά χρονικά βήματα, όπου κάθε βήμα αντιστοιχεί σε μία χρονική περίοδο παρατήρησης (π.χ. μία ημέρα). Σε κάθε χρονική στιγμή ο πράκτορας (agent) παρατηρεί την κατάσταση του συστήματος, επιλέγει μία ενέργεια

τιμολόγησης και λαμβάνει ανταμοιβή (reward) ανάλογα με την επίδοση της απόφασης που έλαβε.

Σε αυτό το κεφάλαιο θα αναλυθεί η διαδικασία για την υλοποίηση αυτού του περιβάλλοντος.

4.2.4.1. Χώρος ενεργειών (Action Space)

Στο πλαίσιο του προβλήματος δυναμικής τιμολόγησης, η ενέργεια του πράκτορα αντιστοιχεί στην επιλογή μίας ποσοστιαίας μεταβολής της τρέχουσας τιμής του προϊόντος. Αντί να επιτρέπονται απεριόριστες μεταβολές τιμών, ο χώρος ενεργειών ορίστηκε ως ένα διακριτό σύνολο προκαθορισμένων επιλογών, το οποίο αντανακλά τις πραγματικές πρακτικές που εφαρμόζονται στη λιανική αγορά.

Συγκεκριμένα, στην παρούσα εργασία ο πράκτορας μπορεί να επιλέξει μία από τις παρακάτω 8 ενέργειες:

1. -30%
2. -20%
3. -10%
4. -5%
5. 0%
6. 5%
7. 10%
8. 20%

Οι αρνητικές μεταβολές αντιπροσωπεύουν εκπτώσεις, ενώ οι θετικές μεταβολές αντιστοιχούν σε αυξήσεις τιμών. Η ενέργεια μηδενικής μεταβολής επιτρέπει στον πράκτορα να διατηρήσει την υφιστάμενη τιμολογιακή πολιτική όταν κρίνει ότι δεν απαιτείται παρέμβαση.

Η επιλογή ενός διακριτού χώρου ενεργειών έγινε για δύο βασικούς λόγους. Πρώτον, επιτρέπει την άμεση εφαρμογή αλγορίθμων tabular Reinforcement Learning, όπως το Q-Learning, χωρίς την ανάγκη χρήσης πιο σύνθετων τεχνικών συνεχούς βελτιστοποίησης. Δεύτερον, προσεγγίζει ρεαλιστικά τον τρόπο με τον οποίο τα τμήματα τιμολόγησης

λαμβάνουν αποφάσεις στην πράξη, καθώς οι τιμές συνήθως μεταβάλλονται σε συγκεκριμένα ποσοστιαία βήματα και όχι σε αυθαίρετες συνεχείς τιμές.

Ιδιαίτερη σημασία έχει η συμπερίληψη ισχυρών αρνητικών μεταβολών, όπως οι εκπτώσεις 20% και 30%. Οι ενέργειες αυτές επιτρέπουν στον πράκτορα να αντιδρά αποτελεσματικά σε περιπτώσεις όπου το προϊόν πλησιάζει στο τέλος του κύκλου ζωής του ή όταν το διαθέσιμο απόθεμα είναι υψηλό σε σχέση με την αναμενόμενη ζήτηση. Σε τέτοιες περιπτώσεις, η μείωση της τιμής μπορεί να λειτουργήσει ως μηχανισμός επιτάχυνσης των πωλήσεων και περιορισμού των απωλειών λόγω αλλοίωσης.

Αντίστοιχα, οι θετικές μεταβολές τιμών δίνουν τη δυνατότητα αξιοποίησης περιόδων αυξημένης ζήτησης. Παραδείγματα αποτελούν οι ημέρες με ειδικά γεγονότα ή εορταστικές περίοδοι, κατά τις οποίες η αναμενόμενη ζήτηση αυξάνεται και η επιχείρηση μπορεί να επιτύχει υψηλότερα περιθώρια κέρδους χωρίς σημαντική μείωση του όγκου πωλήσεων. Η δυνατότητα αύξησης της τιμής λειτουργεί επίσης ως εργαλείο ελέγχου της κατανάλωσης αποθέματος σε περιπτώσεις περιορισμένης διαθεσιμότητας προϊόντων.

Επομένως, ο σχεδιασμός του χώρου ενεργειών επιτρέπει στον πράκτορα να ισορροπεί μεταξύ δύο αντικρουόμενων επιχειρησιακών στόχων: της μεγιστοποίησης του κέρδους μέσω υψηλότερων τιμών και της ελαχιστοποίησης της σπατάλης μέσω στρατηγικών εκπτώσεων. Η τελική πολιτική που προκύπτει από τη διαδικασία εκπαίδευσης αποτελεί αποτέλεσμα της δυναμικής αλληλεπίδρασης αυτών των δύο στόχων με την κατάσταση του συστήματος σε κάθε χρονική στιγμή.

4.2.4.2. Επιχειρησιακές παραδοχές

Για την προσομοίωση ενός ρεαλιστικού περιβάλλοντος λιανικής πώλησης ορίστηκαν συγκεκριμένες επιχειρησιακές παράμετροι οι οποίες επηρεάζουν άμεσα την αξιολόγηση των αποφάσεων του πράκτορα. Το κόστος προμήθειας θεωρήθηκε ίσο με το 70% της τιμής πώλησης του προϊόντος, επιτρέποντας τον υπολογισμό του μικτού κέρδους από κάθε πώληση. Παράλληλα, στη συνάρτηση ανταμοιβής ενσωματώθηκαν μηχανισμοί που λαμβάνουν υπόψη τις απώλειες προϊόντων λόγω αλλοίωσης, το οικονομικό κόστος των μη ικανοποιημένων πωλήσεων και την υπερβολική χρήση εκπτώσεων. Η προσέγγιση αυτή

σχεδιάστηκε ώστε η διαδικασία μάθησης να λαμβάνει υπόψη όχι μόνο τη μεγιστοποίηση του κέρδους, αλλά και τη διαχείριση της σπατάλης και της διαθεσιμότητας των προϊόντων.

Πιο συγκεκριμένα οι τιμές αυτές αναδεικνύονται στον πίνακα (Πίνακας 4.3) παρακάτω:

Επιχειρησιακές παραδοχές	Τιμές
Cost ratio	70%
Waste penalty	1,2
Discount penalty factor	0,15

Πίνακας 4.3 Επιχειρησιακές παραδοχές του περιβάλλοντος μάθησης του πράκτορα

Οι τιμές αυτές θα χρησιμοποιηθούν αργότερα στην συνάρτηση επιβράβευσης (reward function) του πράκτορα.

4.2.4.3. Αρχικοποίηση και δυναμική αποθέματος

Για την αποφυγή μη ρεαλιστικών αρχικών συνθηκών, το αρχικό απόθεμα κάθε προϊόντος δεν ορίστηκε σταθερά αλλά υπολογίζεται δυναμικά βάσει της μέσης ιστορικής ζήτησης του συγκεκριμένου SKU.

Συγκεκριμένα, το επίπεδο αρχικού αποθέματος αντιστοιχεί σε περίπου πέντε ημέρες μέσης ζήτησης, με ελάχιστο όριο τις είκοσι μονάδες. Το κατώτατο όριο 20 μονάδων εφαρμόστηκε ώστε να αποφεύγονται μη ρεαλιστικά χαμηλά επίπεδα αποθέματος σε προϊόντα με πολύ μικρή ζήτηση και να διασφαλίζεται η σταθερότητα της προσομοίωσης. Η προσέγγιση αυτή επιτρέπει την προσαρμογή του περιβάλλοντος στα διαφορετικά πρότυπα κατανάλωσης κάθε προϊόντος και εξασφαλίζει συγκρίσιμες συνθήκες εκπαίδευσης μεταξύ προϊόντων με διαφορετικούς όγκους πωλήσεων.

4.2.4.4. Αναπαράσταση κατάστασης (State representation)

Στο πλαίσιο του Reinforcement Learning, η κατάσταση (state) αντιπροσωπεύει το σύνολο των πληροφοριών που είναι διαθέσιμες στον πράκτορα κατά τη λήψη μιας απόφασης. Η

επιλογή κατάλληλων μεταβλητών κατάστασης αποτελεί κρίσιμο παράγοντα για την αποτελεσματικότητα της διαδικασίας μάθησης, καθώς καθορίζει το επίπεδο πληροφόρησης που διαθέτει ο πράκτορας σχετικά με το περιβάλλον λειτουργίας του.

Στο προτεινόμενο περιβάλλον, η κατάσταση σχεδιάστηκε ώστε να αποτυπώνει τόσο τα χαρακτηριστικά του προϊόντος όσο και τις επιχειρησιακές συνθήκες που επηρεάζουν τις αποφάσεις δυναμικής τιμολόγησης. Για κάθε χρονικό βήμα συλλέγεται ένα σύνολο μεταβλητών που περιγράφουν την τρέχουσα κατάσταση του συστήματος και χρησιμοποιούνται από τον πράκτορα για την επιλογή της κατάλληλης ενέργειας.

Οι μεταβλητές που συνθέτουν την κατάσταση αποτυπώνονται στον πίνακα (Πίνακας 4.4) παρακάτω:

Μεταβλητές	Περιγραφή
Price	Τρέχουσα τιμή προϊόντος
Stock	Διαθέσιμο απόθεμα
Day of week	Ημέρα εβδομάδας
Event flag	Ένδειξη ύπαρξης ειδικού γεγονότος
Shelf life days	Υπολειπόμενες ημέρες ζωής προϊόντος
Spoilage rate	Ρυθμός αλλοίωσης προϊόντος
Event uplift signal	Αναμενόμενη ποσοστιαία αύξηση της ζήτησης λόγω γεγονότος
Elasticity	Ελαστικότητα ζήτησης ως προς την τιμή
Service criticality	Βαθμός επιχειρησιακής κρισιμότητας προϊόντος

Πίνακας 4.4 Μεταβλητές κατάσταση

Η τρέχουσα τιμή και το διαθέσιμο απόθεμα παρέχουν πληροφορίες σχετικά με την υφιστάμενη επιχειρησιακή κατάσταση του προϊόντος. Οι δύο αυτές μεταβλητές είναι άμεσα συνδεδεμένες με τις αποφάσεις τιμολόγησης, καθώς καθορίζουν τόσο το δυνητικό κέρδος όσο και τον κίνδυνο εμφάνισης ελλείψεων ή υπερβολικού αποθέματος.

Οι μεταβλητές που σχετίζονται με την αλλοίωση των προϊόντων (shelf life και spoilage rate) ενσωματώνουν τη φύση των ευπαθών αγαθών στο περιβάλλον. Η συμπερίληψή τους επιτρέπει στον πράκτορα να λαμβάνει αποφάσεις λαμβάνοντας υπόψη τον κίνδυνο μελλοντικής σπατάλης, ο οποίος αυξάνεται όσο μειώνεται ο διαθέσιμος χρόνος ζωής του προϊόντος.

Παράλληλα, η ύπαρξη ειδικών γεγονότων, η αναμενόμενη αύξηση της ζήτησης λόγω αυτών και η ελαστικότητα τιμής επιτρέπουν στο περιβάλλον να αποτυπώνει εξωτερικούς παράγοντες που επηρεάζουν τη συμπεριφορά των καταναλωτών. Με τον τρόπο αυτό, ο πράκτορας δεν βασίζεται αποκλειστικά σε ιστορικά δεδομένα πωλήσεων, αλλά μπορεί να προσαρμόζει τη στρατηγική του ανάλογα με τις αναμενόμενες συνθήκες της αγοράς.

Τέλος, η μεταβλητή service criticality εκφράζει τη σχετική σημασία κάθε προϊόντος για τη λειτουργία του καταστήματος. Η πληροφορία αυτή επιτρέπει στο περιβάλλον να διαφοροποιεί τις συνέπειες μιας έλλειψης αποθέματος ανάλογα με τη στρατηγική αξία του προϊόντος, ενσωματώνοντας επιχειρησιακά κριτήρια πέρα από την απλή μεγιστοποίηση του κέρδους.

Η συγκεκριμένη αναπαράσταση κατάστασης παρέχει μια ολοκληρωμένη εικόνα του επιχειρησιακού περιβάλλοντος, συνδυάζοντας πληροφορίες ζήτησης, αποθέματος, τιμολόγησης και χαρακτηριστικών προϊόντος. Ωστόσο, η άμεση χρήση συνεχών μεταβλητών θα οδηγούσε σε ιδιαίτερα μεγάλο χώρο καταστάσεων, καθιστώντας δυσκολότερη την εφαρμογή του αλγορίθμου Q-Learning. Για τον λόγο αυτό εφαρμόζεται στη συνέχεια διαδικασία διακριτοποίησης των μεταβλητών κατάστασης, η οποία περιγράφεται στην επόμενη ενότητα.

4.2.4.5. Διακριτοποίηση καταστάσεων (State encoding)

Παρόλο που οι μεταβλητές που περιγράφουν το περιβάλλον παρέχουν μια ολοκληρωμένη αναπαράσταση της επιχειρησιακής κατάστασης, η άμεση χρήση των αρχικών συνεχών τιμών θα οδηγούσε στη δημιουργία ενός εξαιρετικά μεγάλου χώρου καταστάσεων. Σε ένα τέτοιο περιβάλλον, η εφαρμογή του αλγορίθμου Q-Learning θα καθίστατο ιδιαίτερα δύσκολη, καθώς ο αριθμός των πιθανών συνδυασμών καταστάσεων θα αυξανόταν εκθετικά.

Για την αντιμετώπιση αυτού του προβλήματος εφαρμόστηκε διαδικασία διακριτοποίησης (state discretization), κατά την οποία οι συνεχείς μεταβλητές μετατρέπονται σε ένα περιορισμένο σύνολο κατηγοριών (bins). Η προσέγγιση αυτή μειώνει σημαντικά την πολυπλοκότητα του προβλήματος και επιτρέπει την αποδοτική εκπαίδευση του πράκτορα μέσω πίνακα Q-values.

Η διακριτοποίηση σχεδιάστηκε με γνώμονα την επιχειρησιακή ερμηνευσιμότητα των καταστάσεων, έτσι ώστε κάθε κατηγορία να αντιστοιχεί σε συγκεκριμένες συνθήκες λειτουργίας του καταστήματος.

Ακολουθούν οι πίνακες (Πίνακες 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.11) με τις διακριτοποιήσεις των τιμών ανά κατηγορία.

Διακριτοποίηση τιμής	
Κατηγορία	Εύρος τιμών
0	Τιμή < 2
1	$2 \leq \text{Τιμή} < 5$
2	$5 \leq \text{Τιμή} < 9$
3	Τιμή ≥ 9

Πίνακας 4.5 Διακριτοποίηση τιμών καταστάσεων

Η συγκεκριμένη διάκριση επιτρέπει στον πράκτορα να αναγνωρίζει διαφορετικά επίπεδα τιμολόγησης χωρίς να απαιτείται γνώση της ακριβούς τιμής κάθε προϊόντος.

Διακριτοποίηση αποθέματος	
Κατηγορία	Ποσοστό αποθέματος
0	< 25%
1	25% - < 50%
2	50% - < 80%
3	$\geq 80\%$

Πίνακας 4.6 Διακριτοποίηση αποθέματος καταστάσεων

Το διαθέσιμο απόθεμα εκφράζεται ως ποσοστό του μέγιστου αποθέματος και στη συνέχεια κατηγοριοποιείται σε τέσσερα επίπεδα. Η χρήση σχετικού αντί απόλυτου αποθέματος επιτρέπει την εφαρμογή της ίδιας πολιτικής σε προϊόντα με διαφορετικούς όγκους πωλήσεων.

Διακριτοποίηση υπολειπόμενης διάρκειας ζωής	
Κατηγορία	Υπολοιπόμενες ημέρες
0	≤ 3 ημέρες
1	4 – 10 ημέρες
2	> 10 ημέρες

Πίνακας 4.7 Διακριτοποίηση υπολειπόμενης διάρκειας ζωής καταστάσεων

Η διάρκεια ζωής των προϊόντων ομαδοποιείται σε τρεις κατηγορίες. Η μεταβλητή αυτή είναι ιδιαίτερα σημαντική για τα ευπαθή προϊόντα, καθώς συνδέεται άμεσα με τον κίνδυνο μελλοντικής αλλοίωσης και απωλειών αποθέματος.

Διακριτοποίηση ρυθμού αλλοίωσης	
Κατηγορία	Ρυθμός αλλοίωσης
0	$< 1\%$
1	1% - 2%
2	$> 2\%$

Πίνακας 4.8 Διακριτοποίηση ρυθμού αλλοίωσης καταστάσεων

Ο ρυθμός αλλοίωσης ταξινομείται σε τρία επίπεδα. Η μεταβλητή αυτή επιτρέπει στον πράκτορα να διαφοροποιεί τη στρατηγική του ανάλογα με τον κίνδυνο απωλειών λόγω φθοράς προϊόντων.

Διακριτοποίηση event uplift	
Κατηγορία	Uplift
0	$< 10\%$
1	10% - 30%

2	> 30%
---	-------

Πίνακας 4.9 Διακριτοποίηση αύξησης της ζήτησης λόγω ειδικών γεγονότων καταστάσεων

Η αναμενόμενη αύξηση ζήτησης λόγω ειδικών γεγονότων (event uplift) ομαδοποιείται σε τρεις κατηγορίες. Η κατηγοριοποίηση αυτή επιτρέπει στον πράκτορα να διακρίνει περιόδους χαμηλής, μέτριας και υψηλής αναμενόμενης αύξησης της ζήτησης.

Διακριτοποίηση ελαστικότητας τιμής	
Κατηγορία	Απόλυτη ελαστικότητα
0	<0,7
1	0,7 – 1,2
2	> 1,2

Πίνακας 4.10 Διακριτοποίηση ελαστικότητας τιμής καταστάσεων

Η απόλυτη τιμή της ελαστικότητας χρησιμοποιείται για τη δημιουργία τριών επιπέδων ευαισθησίας των καταναλωτών. Με τον τρόπο αυτό ο πράκτορας μπορεί να διακρίνει προϊόντα χαμηλής, μέσης και υψηλής ευαισθησίας στις μεταβολές τιμών.

Διακριτοποίηση κρισιμότητας προϊόντος	
Κατηγορία	Κρισιμότητα
0	<0,3
1	0,3 – 0,7
2	> 0,7

Πίνακας 4.11 Διακριτοποίηση κρισιμότητας προϊόντος καταστάσεων

Η μεταβλητή service criticality ταξινομείται επίσης σε τρεις κατηγορίες. Η μεταβλητή αυτή χρησιμοποιείται αργότερα στη συνάρτηση ανταμοιβής για τη διαφοροποίηση των συνεπειών που προκαλούν οι ελλείψεις αποθέματος.

Μετά τη διαδικασία διακριτοποίησης, κάθε κατάσταση αναπαρίσταται ως ένα σύνολο εννέα διακριτών χαρακτηριστικών:

$$State =$$
$$(Price, Stock, ShelfLife, Spoilage, Event, Uplift, Elasticity, Criticality, DayOfWeek)$$

Η τελική αυτή αναπαράσταση επιτρέπει στον πράκτορα να αναγνωρίζει επαναλαμβανόμενα μοτίβα του περιβάλλοντος και να αποθηκεύει αποτελεσματικά τις αντίστοιχες τιμές Q στον πίνακα μάθησης. Παράλληλα, διατηρεί την απαραίτητη πληροφορία για τη λήψη αποφάσεων χωρίς να αυξάνει υπερβολικά τη διάσταση του προβλήματος. Ειδικότερα, ο χώρος της κατάστασης (state space) ανέρχεται σε :

$$4 \cdot 4 \cdot 3 \cdot 3 \cdot 2 \cdot 3 \cdot 3 \cdot 3 \cdot 7 = 54432$$

Πέρα από τη διαδικασία διακριτοποίησης, υλοποιήθηκε και μηχανισμός αντίστροφης απεικόνισης των καταστάσεων (state decoding), ο οποίος επιτρέπει τη μετατροπή των διακριτών κατηγοριών σε αντιπροσωπευτικές επιχειρησιακές τιμές.

Ο μηχανισμός αυτός δεν χρησιμοποιείται κατά την εκπαίδευση του αλγορίθμου ενισχυτικής μάθησης, αλλά αξιοποιείται κατά τη φάση ανάλυσης και ερμηνείας των αποτελεσμάτων. Συγκεκριμένα, κάθε κατηγορία τιμής, αποθέματος, διάρκειας ζωής, ελαστικότητας και λοιπών χαρακτηριστικών αντιστοιχίζεται σε μία αντιπροσωπευτική τιμή, επιτρέποντας τη μετάφραση των καταστάσεων του Q-table σε όρους κατανοητούς από τους χρήστες και τους υπεύθυνους λήψης αποφάσεων.

Η διαδικασία αυτή διευκολύνει την ερμηνευσιμότητα του μοντέλου και επιτρέπει τη δημιουργία οπτικοποιήσεων πολιτικής (policy visualizations), μέσω των οποίων μπορούν να αναλυθούν οι στρατηγικές τιμολόγησης που έμαθε ο πράκτορας σε διαφορετικά επιχειρησιακά σενάρια.

4.2.4.6. Προσομοίωση ζήτησης

Η προσομοίωση της ζήτησης αποτελεί τον βασικό μηχανισμό μετάβασης του περιβάλλοντος από μία κατάσταση στην επόμενη. Κάθε απόφαση τιμολόγησης που λαμβάνει ο πράκτορας επηρεάζει τη ζήτηση του προϊόντος, η οποία στη συνέχεια καθορίζει τις πωλήσεις, την κατανάλωση αποθέματος, τις χαμένες πωλήσεις και τελικά την ανταμοιβή

που λαμβάνει ο αλγόριθμος. Για τον λόγο αυτό αναπτύχθηκε ένας μηχανισμός προσομοίωσης που συνδυάζει ιστορικά δεδομένα ζήτησης, επιδράσεις εξωτερικών γεγονότων και μεταβολές που προκύπτουν από την ελαστικότητα τιμής.

Αφετηρία του υπολογισμού αποτελεί η μεταβλητή «real demand», για την οποία μιλήσαμε παραπάνω και η οποία εκφράζει την πραγματική ζήτηση της αγοράς για το συγκεκριμένο προϊόν, επιτρέποντας στο περιβάλλον να προσεγγίζει τη συνολική αγοραστική πρόθεση των καταναλωτών ανεξάρτητα από περιορισμούς αποθέματος.

Η χρήση της πραγματικής ζήτησης αντί των απλών πωλήσεων είναι ιδιαίτερα σημαντική, καθώς αποτρέπει την υποεκτίμηση της καταναλωτικής ζήτησης σε περιπτώσεις εξάντλησης αποθέματος.

Η πραγματική ζήτηση προσαρμόζεται σύμφωνα με την ύπαρξη ειδικών γεγονότων πριν από την εφαρμογή της επίδρασης της τιμής,. Τα γεγονότα αυτά μπορεί να αντιπροσωπεύουν εορταστικές περιόδους, αργίες ή άλλες συνθήκες που επηρεάζουν προσωρινά τη συμπεριφορά των καταναλωτών.

Όταν ενεργοποιείται ένα γεγονός, η ζήτηση πολλαπλασιάζεται με έναν συντελεστή αύξησης που βασίζεται στο χαρακτηριστικό event uplift signal. Η προσαρμοσμένη ζήτηση υπολογίζεται ως:

$$D_{event} = D_{real}(1 + U)$$

Όπου:

- D_{real} είναι η πραγματική ζήτηση,
- U είναι ο συντελεστής αύξησης ζήτησης λόγω γεγονότος

Για λόγους σταθερότητας και αποφυγής ακραίων μεταβολών, η τιμή του uplift περιορίζεται στο διάστημα $[0,0.5]$, επιτρέποντας μέγιστη αύξηση ζήτησης 50%.

Μετά την ενσωμάτωση των εξωτερικών γεγονότων, η ζήτηση προσαρμόζεται σύμφωνα με την επιλεγμένη μεταβολή τιμής και την ελαστικότητα του προϊόντος.

Η ελαστικότητα εκφράζει τον βαθμό ευαισθησίας των καταναλωτών στις μεταβολές της τιμής. Προϊόντα με υψηλή ελαστικότητα παρουσιάζουν μεγαλύτερες μεταβολές στη ζήτηση όταν αλλάζει η τιμή, ενώ προϊόντα χαμηλής ελαστικότητας επηρεάζονται λιγότερο.

Ο συντελεστής μεταβολής της ζήτησης υπολογίζεται ως:

$$M = 1 + (E \cdot A)$$

Όπου:

- E είναι η ελαστικότητα του προϊόντος,
- A είναι η ποσοστιαία μεταβολή της τιμής που επέλεξε ο πράκτορας.

Η τελική ζήτηση υπολογίζεται από τη σχέση:

$$D = D_{event} \cdot M$$

Με τον τρόπο αυτό οι μειώσεις τιμών οδηγούν σε αύξηση της ζήτησης, ενώ οι αυξήσεις τιμών προκαλούν μείωσή της, με ένταση που εξαρτάται από την κατηγορία του προϊόντος και την αντίστοιχη ελαστικότητα.

Σε πραγματικές συνθήκες αγοράς, η ζήτηση δεν μπορεί να αυξάνεται ή να μειώνεται απεριόριστα ως αποτέλεσμα αλλαγών στην τιμή. Για τον λόγο αυτό εφαρμόζεται μηχανισμός περιορισμού του συντελεστή μεταβολής ζήτησης.

Συγκεκριμένα, ο συντελεστής M περιορίζεται στο διάστημα $0.5 \leq M \leq 1.8$, έτσι, ακόμη και σε ακραίες μεταβολές τιμών, η ζήτηση δεν μπορεί να μειωθεί κάτω από το 50% της αρχικής ούτε να αυξηθεί πάνω από το 180% της βασικής ζήτησης. Ο περιορισμός αυτός συμβάλλει στη σταθερότητα της εκπαίδευσης και στην παραγωγή επιχειρησιακά ρεαλιστικών αποτελεσμάτων.

Τέλος, η προσομοιωμένη ζήτηση αποτελεί τη βασική μεταβλητή που συνδέει τις αποφάσεις του πράκτορα με την εξέλιξη του περιβάλλοντος. Η ποσότητα που προκύπτει από τη διαδικασία προσομοίωσης χρησιμοποιείται άμεσα στον υπολογισμό των πωλήσεων, των χαμένων πωλήσεων και της κατανάλωσης αποθέματος στα επόμενα στάδια του περιβάλλοντος. Κατά συνέπεια, η ποιότητα της στρατηγικής τιμολόγησης που μαθαίνει ο πράκτορας εξαρτάται άμεσα από τον τρόπο με τον οποίο οι μεταβολές τιμών επηρεάζουν τη ζήτηση μέσα στο μοντέλο.

4.2.4.7. Διαχείριση αποθέματος και μετάβαση καταστάσεων

Μετά τον υπολογισμό της ζήτησης, το περιβάλλον προχωρά στην ενημέρωση της κατάστασης του αποθέματος και στον υπολογισμό των επιχειρησιακών αποτελεσμάτων που προκύπτουν από την επιλεγμένη πολιτική τιμολόγησης. Η διαδικασία αυτή αποτελεί τον βασικό μηχανισμό μετάβασης μεταξύ διαδοχικών καταστάσεων και συνδέει άμεσα τις αποφάσεις του πράκτορα με την εξέλιξη του συστήματος.

Όσον αφορά την ποσότητα που τελικά πωλείται δεν μπορεί να υπερβαίνει το διαθέσιμο απόθεμα. Για τον λόγο αυτό οι πραγματοποιημένες πωλήσεις υπολογίζονται ως το ελάχιστο μεταξύ διαθέσιμου αποθέματος και προσομοιωμένης ζήτησης.

$$Sales = \min(Stock, Demand)$$

Έτσι ακόμη και αν η ζήτηση είναι υψηλή, οι πωλήσεις περιορίζονται από τη διαθεσιμότητα του προϊόντος.

Αντιθέτως, όταν η ζήτηση υπερβαίνει το διαθέσιμο απόθεμα, δημιουργούνται χαμένες πωλήσεις (lost sales). Η μεταβλητή αυτή εκφράζει τη ζήτηση που δεν μπόρεσε να ικανοποιηθεί λόγω ανεπαρκούς αποθέματος.

Ο υπολογισμός πραγματοποιείται ως:

$$LostSales = \max(0, Demand - Sales)$$

Οι χαμένες πωλήσεις αποτελούν κρίσιμο επιχειρησιακό δείκτη, καθώς συνδέονται με απώλεια εσόδων, μείωση επιπέδου εξυπηρέτησης πελατών και πιθανή μελλοντική απώλεια πιστότητας καταναλωτών. Για τον λόγο αυτό επηρεάζουν αργότερα τη συνάρτηση ανταμοιβής του πράκτορα.

Μετά την ολοκλήρωση των πωλήσεων, το διαθέσιμο απόθεμα μειώνεται κατά την ποσότητα που πωλήθηκε.

$$Stock_{after\ sales} = Stock - Sales$$

Επειδή το περιβάλλον αφορά ευπαθή προϊόντα, μέρος του εναπομείναντος αποθέματος ενδέχεται να χαθεί λόγω αλλοίωσης. Η απώλεια αυτή υπολογίζεται με βάση τον αντίστοιχο ρυθμό αλλοίωσης του προϊόντος.

$$Waste = Stock_{after\ sales} \cdot SpoilageRate$$

Η ποσότητα που αλλοιώνεται αφαιρείται από το διαθέσιμο απόθεμα και καταγράφεται ως επιχειρησιακή απώλεια. Μετά τον υπολογισμό των απωλειών λόγω αλλοίωσης το απόθεμα στρογγυλοποιείται στην πλησιέστερη ακέραιη μονάδα ώστε να αποφεύγονται κλασματικές ποσότητες προϊόντων. Με τον τρόπο αυτό το περιβάλλον ενσωματώνει το βασικό χαρακτηριστικό των ευπαθών αγαθών, δηλαδή την πιθανότητα απώλειας προϊόντων που δεν διατέθηκαν εγκαίρως στην αγορά. Η ενσωμάτωση αυτού του μηχανισμού διαφοροποιεί την παρούσα εργασία από άλλες μελέτες δυναμικής τιμολόγησης όπου τα προϊόντα θεωρούνται συνήθως μη φθαρτά και μπορούν να παραμείνουν στο απόθεμα χωρίς οικονομικές συνέπειες.

Τέλος, για την προσομοίωση πραγματικών διαδικασιών εφοδιασμού εφαρμόζεται μηχανισμός περιοδικής αναπλήρωσης αποθέματος. Συγκεκριμένα, κάθε επτά χρονικά βήματα το απόθεμα επαναφέρεται στο προκαθορισμένο επίπεδο στόχο που έχει υπολογιστεί κατά την αρχικοποίηση του περιβάλλοντος.

Η πολιτική αυτή προσομοιώνει ένα απλοποιημένο σύστημα εβδομαδιαίου ανεφοδιασμού και επιτρέπει στον πράκτορα να λαμβάνει αποφάσεις τιμολόγησης γνωρίζοντας ότι η διαθεσιμότητα αποθέματος μεταβάλλεται δυναμικά κατά τη διάρκεια του επεισοδίου. Η ύπαρξη περιοδικής αναπλήρωσης αποτρέπει επίσης τη μόνιμη εξάντληση αποθεμάτων και διατηρεί τη λειτουργικότητα του περιβάλλοντος σε μακροχρόνιους κύκλους εκπαίδευσης. Μετά την ολοκλήρωση όλων των παραπάνω διαδικασιών, το περιβάλλον ενημερώνει το χρονικό βήμα και δημιουργεί τη νέα κατάσταση που θα παρατηρήσει ο πράκτορας στο επόμενο βήμα. Η νέα κατάσταση προκύπτει από τα επικαιροποιημένα χαρακτηριστικά του προϊόντος και του αποθέματος και κωδικοποιείται μέσω της διαδικασίας διακριτοποίησης που παρουσιάστηκε στην προηγούμενη ενότητα. Με τον τρόπο αυτό ολοκληρώνεται ένας

πλήρης κύκλος αλληλεπίδρασης μεταξύ πράκτορα και περιβάλλοντος: επιλογή ενέργειας, μεταβολή ζήτησης, πωλήσεις, αλλοίωση αποθέματος, αναπλήρωση και δημιουργία νέας κατάστασης. Η αλληλουχία αυτή αποτελεί τη βάση πάνω στην οποία πραγματοποιείται η διαδικασία μάθησης του αλγορίθμου ενισχυτικής μάθησης.

4.2.4.8. Συνάρτηση ανταμοιβής (Reward function)

Η συνάρτηση ανταμοιβής αποτελεί τον βασικό μηχανισμό μέσω του οποίου ο πράκτορας αξιολογεί την ποιότητα των αποφάσεών του. Σε προβλήματα Reinforcement Learning, η μορφή της ανταμοιβής καθορίζει ουσιαστικά τη συμπεριφορά που θα μάθει ο αλγόριθμος κατά τη διάρκεια της εκπαίδευσης. Για τον λόγο αυτό η σχεδίασή της θεωρείται ένα από τα σημαντικότερα στάδια ανάπτυξης του περιβάλλοντος.

Στο προτεινόμενο μοντέλο η ανταμοιβή δεν βασίζεται αποκλειστικά στην κερδοφορία. Αντίθετα, σχεδιάστηκε ώστε να ενσωματώνει πολλαπλούς επιχειρησιακούς στόχους που χαρακτηρίζουν τη διαχείριση ευπαθών προϊόντων στη λιανική αγορά. Συγκεκριμένα, ο πράκτορας επιβραβεύεται για την επίτευξη κέρδους, ενώ παράλληλα λαμβάνει αρνητική ανταμοιβή για το οικονομικό κόστος των αλλοιωμένων προϊόντων, το οικονομικό κόστος των μη ικανοποιημένων πωλήσεων και την υπερβολική χρήση μειώσεων τιμών.

Η συνολική ανταμοιβή υπολογίζεται ως:

$$\text{Reward} = \text{Profit} - \text{Waste penalty} \cdot \text{Waste cost} - \\ \text{Lost Profit Cost} \cdot (1 + \text{Criticality}) - \text{Discount penalty}$$

Όπου:

$$\text{Lost Profit Cost} = \text{Lost Sales} \times \max(0, \text{New Price} - \text{Unit Cost})$$

Και:

$$\text{Unit Cost} = 0.70 \times \text{Base Price}$$

Η εξίσωση αυτή εκφράζει την ισορροπία μεταξύ κερδοφορίας και λειτουργικής αποτελεσματικότητας, επιτρέποντας στον πράκτορα να αξιολογεί τις αποφάσεις του με βάση πολλαπλά κριτήρια.

Η βασική θετική συνιστώσα της ανταμοιβής είναι το κέρδος που προκύπτει από τις πωλήσεις του προϊόντος.

Το κέρδος υπολογίζεται ως:

$$Profit = Revenue - Cost$$

Όπου:

$$Revenue = Units\ sold \cdot Price$$

Και

$$Cost = Cost\ ratio \cdot Base\ price \cdot Units\ sold$$

Στο περιβάλλον θεωρείται ότι το κόστος προμήθειας αντιστοιχεί στο 70% της βασικής τιμής του προϊόντος. Η παραδοχή αυτή επιτρέπει την εκτίμηση ενός ρεαλιστικού περιθωρίου κέρδους χωρίς να απαιτούνται λεπτομερή λογιστικά δεδομένα ανά προϊόν.

Η μεγιστοποίηση του κέρδους αποτελεί τον πρωταρχικό στόχο του πράκτορα, ωστόσο δεν είναι ο μοναδικός παράγοντας που επηρεάζει την ανταμοιβή.

Η ποινή που είναι ανάλογη του οικονομικού κόστους των αλλοιωμένων μονάδων εισάγεται για τον λόγο ότι στα ευπαθή προϊόντα, η μη έγκαιρη διάθεση αποθέματος μπορεί να οδηγήσει σε αλλοίωση και καταστροφή προϊόντων.

Το κόστος αλλοίωσης υπολογίζεται ως:

$$Waste\ cost = Waste \cdot Cost\ ratio \cdot Base\ price$$

Η συγκεκριμένη ποινή ενθαρρύνει τον πράκτορα να λαμβάνει αποφάσεις που περιορίζουν τη συσσώρευση αποθέματος όταν αυξάνεται ο κίνδυνος αλλοίωσης. Με τον τρόπο αυτό το

περιβάλλον δεν επιβραβεύει μόνο τις υψηλές τιμές, αλλά και τη σωστή διαχείριση των ευπαθών προϊόντων.

Όσον αφορά τις χαμένες πωλήσεις, αυτές προκύπτουν όταν η ζήτηση υπερβαίνει το διαθέσιμο απόθεμα και ένα μέρος της ζήτησης δεν μπορεί να ικανοποιηθεί. Οι χαμένες πωλήσεις αποτελούν σημαντικό επιχειρησιακό πρόβλημα, καθώς αντιστοιχούν σε πωλήσεις και αντίστοιχο μικτό κέρδος που δεν πραγματοποιήθηκαν.

Για τον λόγο αυτό, αντί να χρησιμοποιείται ένας σταθερός χρηματικός συντελεστής ανά χαμένη μονάδα, το οικονομικό κόστος των χαμένων πωλήσεων υπολογίζεται με βάση το δυνητικό μικτό κέρδος που θα προέκυπτε εάν οι συγκεκριμένες μονάδες είχαν πωληθεί.

Αρχικά υπολογίζεται το μοναδιαίο κόστος προμήθειας ως:

$$Unit\ Cost = Cost\ Ratio \cdot Base\ Price$$

όπου $Cost\ Ratio = 0,70$ και $Base\ Price$ είναι η βασική τιμή πώλησης του προϊόντος.

Στη συνέχεια υπολογίζεται το δυνητικό μικτό κέρδος ανά χαμένη μονάδα ως:

$$LostProfitPerUnit = \max(0, NewPrice - UnitCost)$$

όπου $NewPrice$ είναι η τιμή που προκύπτει από την ενέργεια τιμολόγησης του πράκτορα.

Το συνολικό οικονομικό κόστος των χαμένων πωλήσεων υπολογίζεται επομένως ως:

$$LostProfitCost = Lost\ Sales \cdot LostProfitPerUnit$$

Με αυτόν τον τρόπο, η επίδραση των χαμένων πωλήσεων στη συνάρτηση ανταμοιβής εκφράζεται πλέον σε χρηματικούς όρους και είναι συνεπής με τον τρόπο υπολογισμού του κέρδους.

Επιπλέον, το οικονομικό κόστος των χαμένων πωλήσεων προσαρμόζεται βάσει της μεταβλητής *service criticality*, ώστε να διαφοροποιείται η επίδραση μιας έλλειψης ανάλογα

με τη σημασία του προϊόντος για την επιχείρηση. Η αντίστοιχη συνιστώσα της ανταμοιβής υπολογίζεται ως:

$$LostSalesPenalty = LostProfitCost \times (1 + Criticality)$$

Η προσέγγιση αυτή επιτρέπει στο περιβάλλον να αποδίδει μεγαλύτερη βαρύτητα στις μη ικανοποιημένες πωλήσεις προϊόντων υψηλής κρισιμότητας. Επομένως, το service criticality λειτουργεί ως συντελεστής στάθμισης του οικονομικού κόστους και όχι ως σταθερό χρηματικό κόστος ανά χαμένη μονάδα.

Σχετικά με τις εκπτώσεις, αν και μπορούν να αυξήσουν τη ζήτηση και να περιορίσουν την αλλοίωση αποθέματος, η υπερβολική χρήση τους ενδέχεται να μειώσει σημαντικά την κερδοφορία. Για τον λόγο αυτό εισάγεται ένας πρόσθετος μηχανισμός ποινής που ενεργοποιείται μόνο όταν ο πράκτορας επιλέγει αρνητικές μεταβολές τιμών.

Η ποινή υπολογίζεται ως:

$$Discount\ penalty = |Action| \cdot Revenue \cdot \lambda$$

Όπου $\lambda = 0.15$ είναι ο συντελεστής ποινής εκπτώσεων.

Η συγκεκριμένη επιλογή αποτρέπει τον πράκτορα από το να καταφεύγει συστηματικά σε μεγάλες εκπτώσεις προκειμένου να αυξήσει τεχνητά τη ζήτηση, εξαναγκάζοντάς τον να αναζητήσει μια πιο ισορροπημένη στρατηγική τιμολόγησης.

Οι συντελεστές της συνάρτησης ανταμοιβής χρησιμοποιούνται για τη στάθμιση των διαφορετικών επιχειρησιακών στόχων. Ο συντελεστής $WastePenalty = 1,2$ αυξάνει την επίδραση του οικονομικού κόστους αλλοίωσης, ενώ το service criticality διαφοροποιεί την επίδραση του οικονομικού κόστους των χαμένων πωλήσεων ανάλογα με τη σημασία του προϊόντος. Τέλος, ο συντελεστής $\lambda = 0,15$ χρησιμοποιείται για την ποινή των μειώσεων τιμών.

Συνοψίζοντας, η συνάρτηση ανταμοιβής σχεδιάστηκε ώστε να μετατρέπει ένα πολυκριτηριακό επιχειρησιακό πρόβλημα σε ένα ενιαίο σήμα μάθησης για τον πράκτορα. Μέσω της ταυτόχρονης επιβράβευσης της κερδοφορίας και της αρνητικής στάθμισης του οικονομικού κόστους της σπατάλης, των χαμένων πωλήσεων και των υπερβολικών εκπτώσεων, ο πράκτορας οδηγείται στην εκμάθηση μιας πολιτικής που επιδιώκει ισορροπία μεταξύ οικονομικής απόδοσης και λειτουργικής αποτελεσματικότητας. Η φιλοσοφία αυτή αντανακλά τις πραγματικές συνθήκες λειτουργίας ενός σύγχρονου σούπερ μάρκετ, όπου η βέλτιστη τιμολόγηση δεν συνδέεται αποκλειστικά με τη μεγιστοποίηση των πωλήσεων ή του κέρδους, αλλά με τη συνολική βελτιστοποίηση της διαχείρισης αποθεμάτων και της εξυπηρέτησης πελατών.

4.2.4.9. Εκτέλεση βήματος περιβάλλοντος (Environment step)

Η αλληλεπίδραση μεταξύ πράκτορα και περιβάλλοντος υλοποιείται μέσω της συνάρτησης «step», η οποία αποτελεί τον βασικό μηχανισμό μετάβασης του συστήματος από μία κατάσταση στην επόμενη. Σε κάθε χρονικό βήμα, η συνάρτηση δέχεται ως είσοδο την ενέργεια που επέλεξε ο πράκτορας και επιστρέφει τη νέα κατάσταση του περιβάλλοντος, την αντίστοιχη ανταμοιβή και πληροφορίες σχετικά με την απόδοση του συστήματος.

Η διαδικασία εκτέλεσης ενός βήματος ακολουθεί μια συγκεκριμένη αλληλουχία ενεργειών. Αρχικά εφαρμόζεται η επιλεγμένη μεταβολή τιμής στο προϊόν και υπολογίζεται η νέα τιμή πώλησης. Στη συνέχεια ενεργοποιείται ο μηχανισμός προσομοίωσης ζήτησης, ο οποίος λαμβάνει υπόψη τόσο τις εξωτερικές επιδράσεις όσο και την ελαστικότητα της ζήτησης ως προς την τιμή.

Με βάση τη ζήτηση που προκύπτει, υπολογίζονται οι πραγματοποιημένες πωλήσεις και οι πιθανές χαμένες πωλήσεις λόγω ανεπαρκούς αποθέματος. Ακολούθως ενημερώνεται το διαθέσιμο απόθεμα και αφαιρούνται οι ποσότητες που χάνονται λόγω αλλοίωσης. Η διαδικασία αυτή επιτρέπει στο περιβάλλον να προσομοιώνει ρεαλιστικά την καθημερινή λειτουργία ενός συστήματος λιανικής πώλησης ευπαθών προϊόντων.

Μετά την ενημέρωση του αποθέματος υπολογίζονται τα βασικά οικονομικά μεγέθη του συστήματος, όπως τα έσοδα, το κόστος και το κέρδος. Τα μεγέθη αυτά χρησιμοποιούνται από τη συνάρτηση ανταμοιβής για την παραγωγή του σήματος μάθησης που λαμβάνει ο πράκτορας. Παράλληλα, το περιβάλλον διατηρεί συγκεντρωτικά στατιστικά στοιχεία για ολόκληρο το επεισόδιο, συμπεριλαμβανομένων του συνολικού κέρδους, των συνολικών εσόδων, των ποσοτήτων που αλλοιώθηκαν και των χαμένων πωλήσεων. Οι δείκτες αυτοί χρησιμοποιούνται αργότερα για την αξιολόγηση της απόδοσης της πολιτικής που έμαθε ο πράκτορας. Με την ολοκλήρωση των παραπάνω βημάτων, ο μετρητής χρόνου του περιβάλλοντος προχωρά στην επόμενη χρονική περίοδο και παράγεται η νέα κατάσταση. Εφόσον έχουν επεξεργαστεί όλες οι διαθέσιμες χρονικές εγγραφές του προϊόντος, το επεισόδιο τερματίζεται και το περιβάλλον επιστρέφει ένδειξη ολοκλήρωσης (`done = True`). Διαφορετικά, η διαδικασία συνεχίζεται μέχρι να εξεταστούν όλα τα διαθέσιμα δεδομένα του συγκεκριμένου προϊόντος.

Συνολικά, η συνάρτηση `step` υλοποιεί τον πλήρη κύκλο αλληλεπίδρασης μεταξύ πράκτορα και περιβάλλοντος, συνδέοντας τις αποφάσεις τιμολόγησης με τις μεταβολές στη ζήτηση, στο απόθεμα και στην οικονομική απόδοση του συστήματος. Μέσω αυτής της διαδικασίας δημιουργείται το απαραίτητο πλαίσιο μάθησης πάνω στο οποίο εκπαιδεύεται ο αλγόριθμος ενισχυτικής μάθησης.

4.2.5. Πειραματική διαδικασία εκπαίδευσης του πράκτορα και παραγωγή αποτελεσμάτων

Μετά την ανάπτυξη του περιβάλλοντος προσομοίωσης, ακολουθεί η εκπαίδευση ενός πράκτορα RL με στόχο την εκμάθηση μιας αποτελεσματικής πολιτικής δυναμικής τιμολόγησης. Η εκπαίδευση πραγματοποιήθηκε μέσω του αλγορίθμου Q-Learning, ο οποίος ανήκει στην κατηγορία των `model-free` RL αλγορίθμων και επιτρέπει την εκμάθηση βέλτιστων πολιτικών αλληλεπίδρασης με το περιβάλλον χωρίς προηγούμενη γνώση της δυναμικής του συστήματος.

Ο αλγόριθμος επιλέχθηκε λόγω της διακριτής φύσης του χώρου καταστάσεων και ενεργειών που προέκυψε από τη διαδικασία διακριτοποίησης του περιβάλλοντος. Η χρήση

διακριτών καταστάσεων καθιστά εφικτή την αποθήκευση και ενημέρωση των τιμών Q σε μορφή πίνακα, χωρίς να απαιτείται η χρήση νευρωνικών δικτύων ή άλλων τεχνικών προσέγγισης συναρτήσεων.

Η υλοποίηση του Q-Learning πραγματοποιήθηκε εξ ολοκλήρου σε γλώσσα Python χωρίς χρήση εξειδικευμένων βιβλιοθηκών Reinforcement Learning (π.χ. Stable-Baselines3). Η επιλογή αυτή έγινε ώστε να υπάρχει πλήρης έλεγχος της διαδικασίας μάθησης, της αναπαράστασης καταστάσεων, της συνάρτησης ανταμοιβής και της δυναμικής του περιβάλλοντος. Επιπλέον, η χειροκίνητη υλοποίηση διευκολύνει την κατανόηση και ερμηνεία του αλγορίθμου, καθώς και την προσαρμογή του στις ιδιαιτερότητες του προβλήματος διαχείρισης ευπαθών προϊόντων.

Σε αυτή την ενότητα θα δούμε αναλυτικά πως έγινε η εκπαίδευση του πράκτορα και η παραγωγή αποτελεσμάτων για την αξιολόγηση της.

4.2.5.1. Επιλογή προϊόντων εκπαίδευσης

Η εκπαίδευση πραγματοποιήθηκε ξεχωριστά για κάθε προϊόν υψηλής επιχειρησιακής σημασίας (ABC Class A), επιτρέποντας στον πράκτορα να προσαρμόζει τη στρατηγική τιμολόγησής του στα ιδιαίτερα χαρακτηριστικά κάθε SKU. Για κάθε μοναδικό προϊόν της κατηγορίας A δημιουργείται ξεχωριστό περιβάλλον RL και εκτελείται ανεξάρτητη διαδικασία εκπαίδευσης. Τα δεδομένα εκπαίδευσης προέρχονται από το RL dataset που παρουσιάστηκε σε προηγούμενη ενότητα και οργανώνονται σε διαδοχικά επεισόδια μάθησης.

4.2.5.2. Παράμετροι εκπαίδευσης

Για την τελική εκπαίδευση του πράκτορα χρησιμοποιήθηκαν οι παρακάτω μεταβλητές με τις αντίστοιχες τιμές οι οποίες οριστικοποιήθηκαν μετά από πληθώρα εμπειρικών δοκιμών (Πίνακας 4.12).

Μεταβλητές	Τιμές
Alpha (Ρυθμός μάθησης)	0.10
Gamma (συντελεστής προεξόφλησης)	0.95
Epsilon start (Ρυθμός εξερεύνησης)	1.0
Epsilon min (ελάχιστος ρυθμός εξερεύνησης)	0.05
Episodes (Επεισόδια εκπαίδευσης)	2200

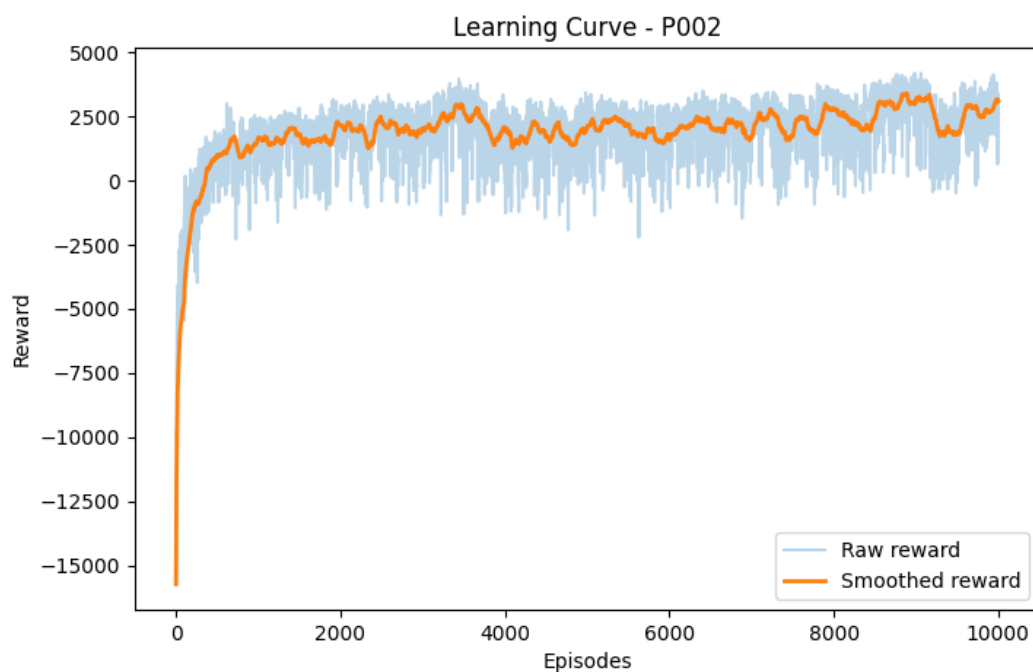
Πίνακας 4.12 Μεταβλητές εκπαίδευσης του πράκτορα

Ειδικότερα, το alpha καθορίζει τον βαθμό στον οποίο οι νέες εμπειρίες επηρεάζουν τη γνώση που έχει ήδη αποκτήσει ο πράκτορας. Στην παρούσα εργασία χρησιμοποιήθηκε τιμή $\alpha = 0.10$, γεγονός που σημαίνει ότι κάθε νέα εμπειρία συμβάλλει κατά 10% στην ενημέρωση της αντίστοιχης τιμής Q, ενώ το υπόλοιπο 90% προέρχεται από την προηγούμενη γνώση του συστήματος. Το gamma είναι υψηλό επειδή έχουμε έντονο ενδιαφέρον για τις μελλοντικές ανταμοιβές καθώς η αλλοίωση των προϊόντων, οι ελλείψεις και η ανανέωση των αποθεμάτων εμφανίζονται αργότερα και δεν έχουμε κάποιο βραχυπρόθεσμο πρόβλημα. Όσον αφορά τις παραμέτρους εξερεύνησης κατά την έναρξη της εκπαίδευσης η τιμή της παραμέτρου epsilon start ορίστηκε ίση με 1.0, επιτρέποντας στον πράκτορα να εξερευνήσει εκτενώς τον χώρο ενεργειών.

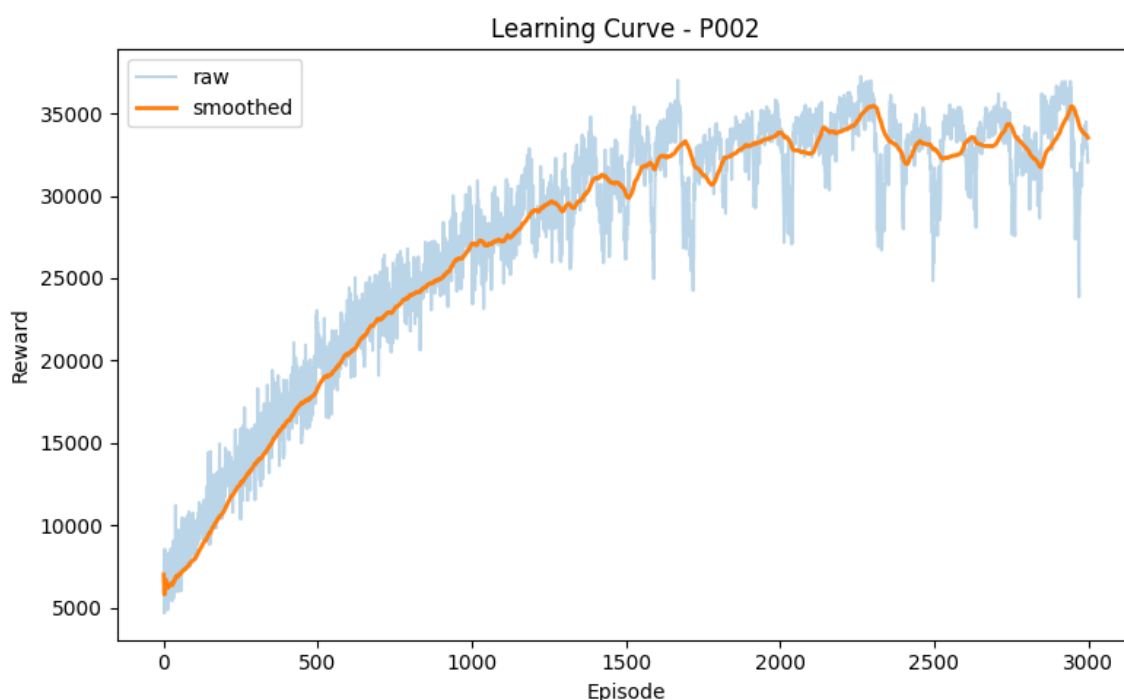
Παράλληλα ορίστηκε κατώτατο όριο ίσο με 0.05, ώστε ακόμη και στα τελικά στάδια της εκπαίδευσης να διατηρείται ένα μικρό ποσοστό τυχαίων ενεργειών. Η προσέγγιση αυτή περιορίζει τον κίνδυνο πρόωρης σύγκλισης σε τοπικά βέλτιστες λύσεις και επιτρέπει τη συνεχή διερεύνηση εναλλακτικών στρατηγικών.

Η ακριβής διαδικασία μείωσης της παραμέτρου epsilon παρουσιάζεται στην επόμενη ενότητα.

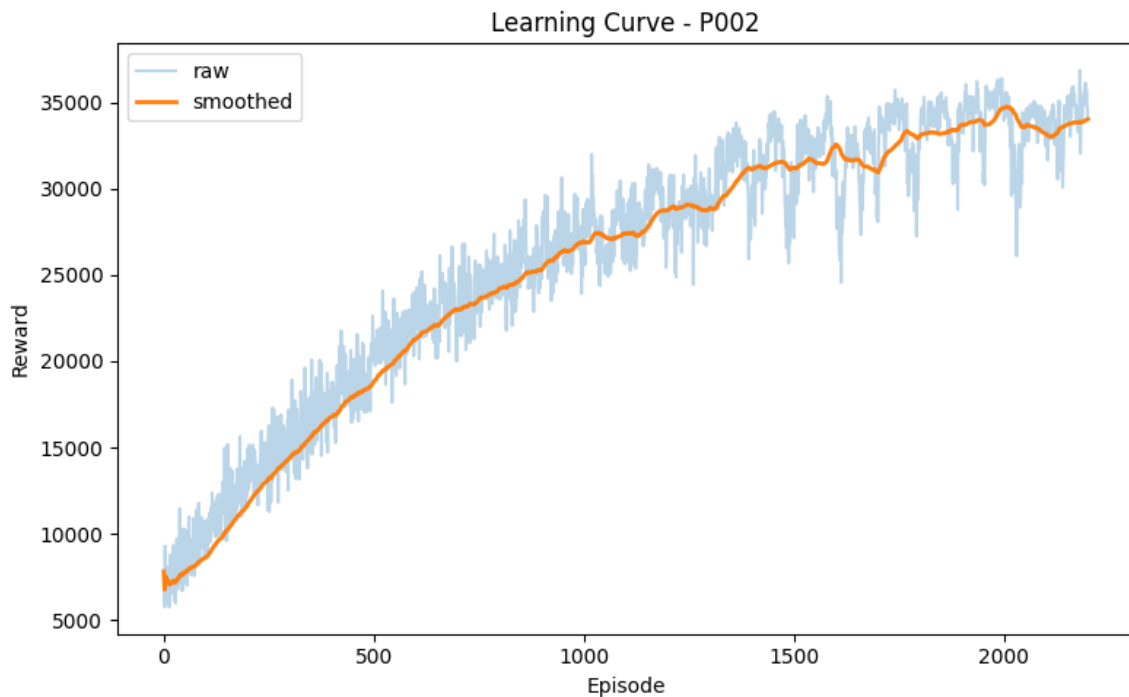
Σχετικά με τον αριθμό επεισοδίων, μετά από πειραματικές δοκιμές και αξιολόγηση των καμπυλών μάθησης, με στόχο την επίτευξη σταθερής σύγκλισης και ικανοποιητικής απόδοσης οι βέλτιστη τιμή ορίστηκε στα 2200 επεισόδια. Όπως μπορούμε να δούμε στις εικόνες (Εικόνες 4.2, 4.3, 4.4) παρακάτω, μετά τα 2000 επεισόδια ο ρυθμός της ανταμοιβής φθίνει και η καμπύλη σταθεροποιείται.



Εικόνα 4.2 Καμπύλη μάθησης του πράκτορα για το προϊόν P002 με 10000 επεισόδια



Εικόνα 4.3 Καμπύλη μάθησης του πράκτορα για το προϊόν P002 με 3000 επεισόδια



Εικόνα 4.4 Καμπύλη μάθησης του πράκτορα για το προϊόν P002 με 2200 επεισόδια

Η σχετική μεταβλητότητα που παρατηρείται στις καμπύλες μάθησης μπορεί να αποδοθεί εν μέρει στη δομή του συνόλου δεδομένων. Συγκεκριμένα, η ανάλυση περιορίστηκε αποκλειστικά στα προϊόντα κατηγορίας A, τα οποία χαρακτηρίζονται από υψηλή αξία και αυξημένη μεταβλητότητα στη ζήτηση. Επιπλέον, δεν συμπεριλήφθηκαν προϊόντα χαμηλότερης προτεραιότητας (κατηγορίες B και C), τα οποία θα μπορούσαν να λειτουργήσουν εξομαλυντικά στην κατανομή των παρατηρήσεων. Ως αποτέλεσμα, το περιβάλλον εκπαίδευσης παρουσιάζει αυξημένο θόρυβο και μεγαλύτερη διακύμανση στις ανταμοιβές.

4.2.5.3. Πολιτική εξερεύνησης Epsilon – Greedy

Ένα από τα βασικότερα ζητήματα στους αλγόριθμους RL είναι η ισορροπία μεταξύ εξερεύνησης (exploration) και εκμετάλλευσης (exploitation). Ο πράκτορας πρέπει αφενός να δοκιμάζει νέες ενέργειες ώστε να ανακαλύπτει πιθανώς καλύτερες στρατηγικές και

αφετέρου να αξιοποιεί τη γνώση που έχει ήδη αποκτήσει για τη μεγιστοποίηση της ανταμοιβής.

Για την αντιμετώπιση αυτού του προβλήματος χρησιμοποιήθηκε η πολιτική epsilon-greedy, μία από τις πιο διαδεδομένες στρατηγικές επιλογής ενεργειών σε προβλήματα Q-Learning. Σύμφωνα με την προσέγγιση αυτή, σε κάθε χρονικό βήμα ο πράκτορας επιλέγει μία τυχαία ενέργεια με πιθανότητα ϵ , ενώ με πιθανότητα $1 - \epsilon$ επιλέγει την ενέργεια που αντιστοιχεί στη μεγαλύτερη εκτιμώμενη τιμή Q για την τρέχουσα κατάσταση.

Η διαδικασία αυτή μπορεί να εκφραστεί ως:

$$a = \begin{cases} \text{τυχαία ενέργεια, με πιθανότητα } \epsilon \\ \arg \max_a Q(s, a), \text{ με πιθανότητα } 1 - \epsilon \end{cases}$$

Κατά τα αρχικά στάδια της εκπαίδευσης ο πράκτορας διαθέτει περιορισμένη γνώση σχετικά με το περιβάλλον. Για τον λόγο αυτό η τιμή της παραμέτρου ϵ αρχικοποιείται στην τιμή 1, επιτρέποντας σχεδόν πλήρη εξερεύνηση του χώρου ενεργειών. Με τον τρόπο αυτό ο πράκτορας αποκτά εμπειρία από διαφορετικές καταστάσεις και στρατηγικές τιμολόγησης. Καθώς η εκπαίδευση εξελίσσεται, η τιμή της παραμέτρου ϵ μειώνεται προοδευτικά μέσω εκθετικής απομείωσης (exponential decay):

$$\epsilon = \max(\epsilon_{\min}, \epsilon_{\text{start}} e^{-0.0015 \cdot \text{episode}})$$

Η προσέγγιση αυτή επιτρέπει στον πράκτορα να δίνει έμφαση στην εξερεύνηση κατά τα πρώτα επεισόδια και σταδιακά να μεταβαίνει σε εκμετάλλευση της γνώσης που έχει αποκτήσει. Παράλληλα, η ύπαρξη κατώτατου ορίου ίσου με 0.05 εξασφαλίζει ότι η εξερεύνηση δεν μηδενίζεται ποτέ πλήρως, διατηρώντας τη δυνατότητα ανακάλυψης εναλλακτικών στρατηγικών ακόμη και στα τελικά στάδια της εκπαίδευσης.

Η χρήση της epsilon-greedy πολιτικής θεωρείται ιδιαίτερα κατάλληλη για το συγκεκριμένο πρόβλημα, καθώς το περιβάλλον περιλαμβάνει μεγάλο αριθμό συνδυασμών καταστάσεων που σχετίζονται με διαφορετικά επίπεδα αποθέματος, διάρκειας ζωής προϊόντων, ελαστικότητας και εξωτερικών γεγονότων. Η συστηματική εξερεύνηση αυτών των

συνδυασμών επιτρέπει στον πράκτορα να αναπτύξει πιο αποτελεσματικές πολιτικές δυναμικής τιμολόγησης σε σχέση με μια αποκλειστικά εκμεταλλευτική προσέγγιση.

4.2.5.4. Αναπαράσταση και ενημέρωση του πίνακα Q (Q - table)

Ο πυρήνας του αλγορίθμου Q-Learning είναι η εκτίμηση της αναμενόμενης μακροχρόνιας ανταμοιβής που συνδέεται με κάθε πιθανό συνδυασμό κατάστασης και ενέργειας. Η γνώση αυτή αποθηκεύεται σε έναν πίνακα Q-values (Q-table), ο οποίος ενημερώνεται προοδευτικά κατά τη διάρκεια της εκπαίδευσης.

Στην παρούσα εργασία ο Q-table υλοποιείται ως δομή τύπου defaultdict, όπου κάθε στοιχείο αντιστοιχεί σε ένα ζεύγος κατάστασης-ενέργειας (state-action pair) και αποθηκεύει την αντίστοιχη τιμή Q . Κατά την αρχικοποίηση, όλες οι τιμές Q λαμβάνουν μηδενική τιμή, γεγονός που σημαίνει ότι ο πράκτορας ξεκινά χωρίς καμία προηγούμενη γνώση για το περιβάλλον.

Η κατάσταση s περιγράφεται από το διακριτοποιημένο διάνυσμα χαρακτηριστικών που παρουσιάστηκε στην προηγούμενη ενότητα, ενώ η ενέργεια a αντιστοιχεί σε μία από τις οκτώ διαθέσιμες μεταβολές τιμής. Συνεπώς, κάθε τιμή $Q(s,a)$ εκφράζει την αναμενόμενη συνολική ανταμοιβή που αναμένεται να λάβει ο πράκτορας εάν επιλέξει την ενέργεια a ενώ βρίσκεται στην κατάσταση s .

Μετά την εκτέλεση μιας ενέργειας, το περιβάλλον επιστρέφει την επόμενη κατάσταση « s' » και την αντίστοιχη ανταμοιβή « r ». Ο πράκτορας αξιολογεί τη μελλοντική προοπτική της νέας κατάστασης υπολογίζοντας τη μέγιστη διαθέσιμη τιμή Q μεταξύ όλων των πιθανών ενεργειών.

Η ποσότητα αυτή εκφράζεται ως:

$$\max_{a'} Q(s', a')$$

και αντιπροσωπεύει την καλύτερη αναμενόμενη μελλοντική απόδοση που μπορεί να επιτευχθεί από τη νέα κατάσταση. Στην περίπτωση που το επεισόδιο έχει ολοκληρωθεί και

δεν υπάρχει επόμενη κατάσταση, η μελλοντική αξία ορίζεται ίση με μηδέν, καθώς δεν αναμένονται πρόσθετες ανταμοιβές.

Όσον αφορά την ενημέρωση των τιμών Q , η μάθηση πραγματοποιείται μέσω της επαναληπτικής ενημέρωσης των Q -values. Μετά από κάθε αλληλεπίδραση με το περιβάλλον, η τιμή που αντιστοιχεί στο τρέχον ζεύγος κατάστασης-ενέργειας αναθεωρείται σύμφωνα με τον κανόνα ενημέρωσης του Q-Learning.

Η εξίσωση ενημέρωσης δίνεται από:

$$Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$$

Όπου:

- $Q(s, a)$ είναι η τρέχουσα εκτίμηση της αξίας της ενέργειας,
- r είναι η άμεση ανταμοιβή που επέστρεψε το περιβάλλον,
- α είναι ο συντελεστής μάθησης,
- γ είναι ο παράγοντας προεξόφλησης,
- s' είναι η επόμενη κατάσταση.

Η εξίσωση αυτή επιτρέπει στον πράκτορα να συνδυάζει την εμπειρία που μόλις απέκτησε με τη γνώση που έχει ήδη αποθηκεύσει στον πίνακα Q . Καθώς αυξάνεται ο αριθμός των επεισοδίων, οι τιμές Q συγκλίνουν σταδιακά προς ακριβέστερες εκτιμήσεις της πραγματικής αξίας κάθε ενέργειας.

Κατά τη διάρκεια της εκπαίδευσης, ο πράκτορας επαναλαμβάνει συνεχώς τον ίδιο κύκλο ενεργειών και η διαδικασία μάθησης είναι η εξής:

- 1) Παρατηρεί την τρέχουσα κατάσταση.
- 2) Επιλέγει μία ενέργεια μέσω της πολιτικής epsilon-greedy.
- 3) Αλληλεπιδρά με το περιβάλλον.
- 4) Λαμβάνει ανταμοιβή και νέα κατάσταση.
- 5) Ενημερώνει την αντίστοιχη τιμή Q .
- 6) Μεταβαίνει στην επόμενη κατάσταση.

Η επαναλαμβανόμενη εφαρμογή αυτής της διαδικασίας επιτρέπει στον πράκτορα να ανακαλύπτει σταδιακά ποιες στρατηγικές τιμολόγησης οδηγούν σε υψηλότερη συνολική ανταμοιβή υπό διαφορετικές επιχειρησιακές συνθήκες.

Στο πλαίσιο της παρούσας εργασίας, η Q-table λειτουργεί ως μηχανισμός αποθήκευσης της εμπειρίας που αποκτά ο πράκτορας σχετικά με τη σχέση μεταξύ τιμολόγησης, ζήτησης και διαχείρισης αποθέματος. Μέσω της συνεχούς ενημέρωσης των τιμών Q, ο πράκτορας μαθαίνει ποιες μεταβολές τιμών είναι περισσότερο αποτελεσματικές για συγκεκριμένους συνδυασμούς αποθέματος, διάρκειας ζωής προϊόντων, ελαστικότητας και συνθηκών αγοράς.

Με τον τρόπο αυτό η γνώση που αποκτάται κατά τη διάρκεια της εκπαίδευσης μετατρέπεται σταδιακά σε μια ολοκληρωμένη πολιτική δυναμικής τιμολόγησης, η οποία μπορεί να εφαρμοστεί σε νέες καταστάσεις του περιβάλλοντος χωρίς να απαιτείται περαιτέρω αναζήτηση ή βελτιστοποίηση.

4.2.5.5. Πειραματικός σχεδιασμός τυχαίας αρχικοποίησης

Η απόδοση των αλγορίθμων RL επηρεάζεται σημαντικά από στοχαστικούς παράγοντες, όπως η αρχικοποίηση του περιβάλλοντος, η διαδικασία εξερεύνησης και οι τυχαίες επιλογές ενεργειών κατά τη διάρκεια της εκπαίδευσης. Για τον περιορισμό της επίδρασης της τυχαιότητας και την αύξηση της αξιοπιστίας των αποτελεσμάτων, η εκπαίδευση δεν πραγματοποιήθηκε μία μόνο φορά αλλά επαναλήφθηκε για πολλαπλές τιμές τυχαίας αρχικοποίησης (random seed).

Συγκεκριμένα, χρησιμοποιήθηκαν πέντε διαφορετικοί τυχαίοι «σπόροι»:

$$Seeds = \{1, 7, 21, 42, 99\}$$

Για κάθε τιμή του random seed πραγματοποιείται ανεξάρτητη εκτέλεση ολόκληρης της διαδικασίας εκπαίδευσης και αξιολόγησης. Πριν από την έναρξη κάθε εκτέλεσης αρχικοποιούνται εκ νέου οι γεννήτριες τυχαίων αριθμών των βιβλιοθηκών Python και NumPy, εξασφαλίζοντας ότι τα αποτελέσματα κάθε πειράματος είναι αναπαραγώγιμα.

Η χρήση πολλαπλών ανεξάρτητων εκτελέσεων επιτρέπει την εκτίμηση της σταθερότητας του αλγορίθμου και μειώνει την πιθανότητα τα συμπεράσματα να επηρεάζονται από μία ευνοϊκή ή δυσμενή τυχαία αρχικοποίηση. Αντί να αξιολογείται η απόδοση του συστήματος από μία μόνο εκπαίδευση, τα τελικά αποτελέσματα προκύπτουν από τη συγκέντρωση και στατιστική ανάλυση των αποτελεσμάτων όλων των εκτελέσεων.

Η προσέγγιση αυτή θεωρείται ιδιαίτερα σημαντική σε πειράματα RL, καθώς παρέχει μεγαλύτερη αξιοπιστία στα συμπεράσματα σχετικά με την αποτελεσματικότητα της προτεινόμενης στρατηγικής δυναμικής τιμολόγησης (Henderson et al., 2018).

4.2.5.6. Πολιτική αναφοράς (Heuristic Baseline Policy)

Για την αξιολόγηση της αποτελεσματικότητας του προτεινόμενου αλγορίθμου RL αναπτύχθηκε μία πολιτική αναφοράς (heuristic baseline policy), η οποία βασίζεται σε προκαθορισμένους επιχειρησιακούς κανόνες και δεν περιλαμβάνει διαδικασία μάθησης. Η πολιτική αυτή λειτουργεί ως σημείο σύγκρισης, επιτρέποντας την εκτίμηση της προστιθέμενης αξίας που προκύπτει από τη χρήση του Reinforcement Learning έναντι μιας συμβατικής rule-based προσέγγισης.

Η heuristic πολιτική λαμβάνει αποφάσεις χρησιμοποιώντας πληροφορίες σχετικά με το διαθέσιμο απόθεμα, την υπολειπόμενη διάρκεια ζωής του προϊόντος, τον ρυθμό αλλοίωσης, την ύπαρξη ειδικών γεγονότων και τη στρατηγική κρισιμότητα του προϊόντος. Οι αποφάσεις εφαρμόζονται μέσω μιας σειράς διαδοχικών κανόνων προτεραιότητας.

Πρώτον, όταν η υπολειπόμενη διάρκεια ζωής ενός προϊόντος είναι ίση ή μικρότερη από δύο ημέρες, εφαρμόζεται έκπτωση 30%. Η επιλογή αυτή στοχεύει στη γρήγορη απορρόφηση του αποθέματος πριν από την πιθανή αλλοίωση του προϊόντος.

Δεύτερον, όταν ο ρυθμός αλλοίωσης υπερβαίνει το 5%, εφαρμόζεται έκπτωση 20%, με στόχο τη μείωση του κινδύνου δημιουργίας αποβλήτων και οικονομικών απωλειών λόγω φθοράς των προϊόντων.

Τρίτον, σε περιπτώσεις όπου υπάρχει ενεργό γεγονός και η αναμενόμενη αύξηση ζήτησης υπερβαίνει το 15%, η πολιτική εφαρμόζει αύξηση τιμής κατά 10%. Η απόφαση αυτή βασίζεται στην παραδοχή ότι η αυξημένη καταναλωτική ζήτηση επιτρέπει υψηλότερα περιθώρια κέρδους χωρίς σημαντική μείωση του όγκου πωλήσεων.

Τέταρτον, όταν το διαθέσιμο απόθεμα βρίσκεται κάτω από το 25% της μέγιστης χωρητικότητας αποθήκευσης, εφαρμόζεται αύξηση τιμής κατά 20%. Η στρατηγική αυτή επιδιώκει να επιβραδύνει την κατανάλωση αποθέματος και να περιορίσει την πιθανότητα εμφάνισης ελλείψεων προϊόντων.

Τέλος, για προϊόντα με υψηλή επιχειρησιακή κρισιμότητα ($\text{criticality} > 0.7$), καθώς και για όλες τις υπόλοιπες περιπτώσεις, η πολιτική διατηρεί αμετάβλητη την τιμή του προϊόντος. Η επιλογή αυτή αντανακλά μια συντηρητική προσέγγιση διαχείρισης τιμών για προϊόντα στρατηγικής σημασίας.

Η πολιτική αναφοράς αξιολογείται στο ίδιο περιβάλλον και με τους ίδιους δείκτες απόδοσης που χρησιμοποιούνται για τον πράκτορα Reinforcement Learning. Κατά συνέπεια, οι διαφορές που παρατηρούνται στα τελικά αποτελέσματα μπορούν να αποδοθούν αποκλειστικά στις διαφορετικές στρατηγικές λήψης αποφάσεων και όχι σε μεταβολές του πειραματικού πλαισίου.

4.2.5.7. Διαδικασία αξιολόγησης και δείκτες απόδοσης

Μετά την ολοκλήρωση της διαδικασίας εκπαίδευσης, η απόδοση του πράκτορα αξιολογείται στο ίδιο περιβάλλον χρησιμοποιώντας την πολιτική που προέκυψε από τον τελικό πίνακα Q-table. Κατά τη φάση αξιολόγησης δεν πραγματοποιείται πλέον εξερεύνηση του χώρου ενεργειών. Για κάθε κατάσταση επιλέγεται η ενέργεια με τη μεγαλύτερη εκτιμώμενη τιμή Q, υλοποιώντας έτσι την πολιτική που έχει μάθει ο πράκτορας κατά τη διάρκεια της εκπαίδευσης.

Η διαδικασία αξιολόγησης εκτελείται για κάθε προϊόν ξεχωριστά και επαναλαμβάνεται για όλες τις τιμές random seed που χρησιμοποιήθηκαν κατά την εκπαίδευση. Παράλληλα, η ίδια διαδικασία εφαρμόζεται και στην ευρετική πολιτική αναφοράς, ώστε οι δύο προσεγγίσεις να αξιολογηθούν υπό ακριβώς τις ίδιες συνθήκες περιβάλλοντος.

Για κάθε εκτέλεση συλλέγονται τέσσερις βασικοί επιχειρησιακοί δείκτες απόδοσης:

- Συνολικό Κέρδος (Profit)
- Συνολικά Έσοδα (Revenue)

- Συνολικές Απώλειες λόγω Αλλοίωσης (Waste)
- Συνολικές Χαμένες Πωλήσεις (Lost Sales)

Οι δείκτες αυτοί προέρχονται απευθείας από το περιβάλλον και αποτυπώνουν διαφορετικές διαστάσεις της επιχειρησιακής απόδοσης της στρατηγικής τιμολόγησης.

Το κέρδος και τα έσοδα χρησιμοποιούνται για την αξιολόγηση της οικονομικής αποτελεσματικότητας της πολιτικής, ενώ οι δείκτες waste και lost sales επιτρέπουν την εκτίμηση της επίδρασης της στρατηγικής στη διαχείριση αποθέματος και στο επίπεδο εξυπηρέτησης πελατών.

Για κάθε προϊόν υπολογίζεται η διαφορά απόδοσης μεταξύ του Reinforcement Learning και της heuristic πολιτικής.

Η απόλυτη βελτίωση ορίζεται ως:

$$Improvement = Reward_{RL} - Reward_{Heuristic}$$

Θετική τιμή υποδηλώνει ότι η πολιτική Reinforcement Learning απέδωσε καλύτερα από την πολιτική αναφοράς, ενώ αρνητική τιμή υποδηλώνει υπεροχή της heuristic προσέγγισης.

Παράλληλα υπολογίζεται και η ποσοστιαία βελτίωση:

$$Improvement \% = \frac{Reward_{RL} - Reward_{Heuristic}}{|Reward_{Heuristic}| + 10^{-6}} \cdot 100$$

Η χρήση της μικρής σταθεράς 10^{-6} αποτρέπει προβλήματα αριθμητικής αστάθειας σε περιπτώσεις όπου η ανταμοιβή της ευρετικής πολιτικής προσεγγίζει το μηδέν.

Πέρα από τη συνολική ανταμοιβή, υπολογίζονται και δείκτες βελτίωσης για τα βασικά επιχειρησιακά μεγέθη. Για τον σκοπό αυτό χρησιμοποιείται ένας συμμετρικός δείκτης μεταβολής (Symmetric Improvement Index), ο οποίος επιτρέπει δίκαιη σύγκριση μεταξύ διαφορετικών προϊόντων και διαφορετικών κλιμάκων μεγεθών.

Ο δείκτης εφαρμόζεται ξεχωριστά για:

- Βελτίωση των κερδών (Profit Improvement)
- Βελτίωση πωλήσεων (Revenue Improvement)
- Μείωση απορρίψεων (Waste Reduction)
- Μείωση χαμένων πωλήσεων (Lost Sales Reduction)

και εκφράζει τη σχετική βελτίωση του RL σε σχέση με την ευρετική (heuristic) πολιτική ανεξάρτητα από το απόλυτο μέγεθος των μετρικών.

Μετά την ολοκλήρωση όλων των εκτελέσεων, τα αποτελέσματα συγκεντρώνονται σε ενιαίο πίνακα δεδομένων. Για κάθε προϊόν υπολογίζονται ο μέσος όρος και η τυπική απόκλιση των δεικτών απόδοσης μεταξύ των διαφορετικών τυχαίων αρχικοποιήσεων (random seeds). Επιπλέον, υπολογίζονται διαστήματα εμπιστοσύνης 95% (Confidence Intervals), χρησιμοποιώντας την κλασική προσέγγιση:

$$CI_{95} = 1.96 \cdot \frac{\sigma}{\sqrt{n}}$$

Όπου:

- σ είναι η τυπική απόκλιση των αποτελεσμάτων
- n είναι ο αριθμός των ανεξάρτητων εκτελέσεων

Η διαδικασία αυτή επιτρέπει την εκτίμηση όχι μόνο της μέσης απόδοσης του αλγορίθμου αλλά και της σταθερότητας των αποτελεσμάτων μεταξύ διαφορετικών εκπαιδεύσεων.

4.2.5.8. Οπτικοποίηση και ερμηνεία πολιτικών

Πέρα από την ποσοτική αξιολόγηση μέσω των επιχειρησιακών δεικτών, πραγματοποιήθηκε και οπτική ανάλυση της πολιτικής που προέκυψε από τη διαδικασία εκπαίδευσης. Στόχος

της ανάλυσης αυτής είναι η ερμηνεία των αποφάσεων του πράκτορα και η σύγκρισή τους με την πολιτική αναφοράς που βασίζεται σε προκαθορισμένους επιχειρησιακούς κανόνες. Για κάθε προϊόν δημιουργήθηκε ένα σύνολο τριών χαρτών θερμότητας (heatmaps), οι οποίοι παρουσιάζονται σε κοινό διάγραμμα. Ο πρώτος χάρτης απεικονίζει την πολιτική που έμαθε ο πράκτορας RL, ο δεύτερος την πολιτική αναφοράς (heuristic baseline), ενώ ο τρίτος τη συχνότητα επίσκεψης των καταστάσεων κατά τη διάρκεια της εκπαίδευσης.

Οι άξονες των χαρτών αντιστοιχούν στις διακριτοποιημένες κατηγορίες αποθέματος και υπολειπόμενης διάρκειας ζωής προϊόντος, δύο μεταβλητές που επηρεάζουν άμεσα τις αποφάσεις δυναμικής τιμολόγησης. Κάθε κελί του χάρτη περιέχει την ενέργεια που επιλέγεται από την αντίστοιχη πολιτική, εκφρασμένη ως ποσοστιαία μεταβολή της τιμής. Για τη διευκόλυνση της ερμηνείας, οι διακριτοποιημένες καταστάσεις μετατρέπονται σε επιχειρησιακά κατανοητές κατηγορίες μέσω της διαδικασίας state decoding.

Η σύγκριση μεταξύ των δύο πολιτικών επιτρέπει την αναγνώριση περιοχών όπου ο πράκτορας RL ακολουθεί διαφορετικές στρατηγικές από εκείνες που προκύπτουν από τους εμπειρικούς κανόνες της heuristic προσέγγισης. Με τον τρόπο αυτό καθίσταται δυνατή η ερμηνεία των διαφορών που παρατηρούνται στους δείκτες απόδοσης και η διερεύνηση των συνθηκών υπό τις οποίες η διαδικασία μάθησης οδηγεί σε εναλλακτικές αποφάσεις τιμολόγησης.

Παράλληλα, ο χάρτης συχνότητας επισκέψεων παρέχει πληροφορίες σχετικά με την κάλυψη του χώρου καταστάσεων κατά τη διάρκεια της εκπαίδευσης. Η οπτικοποίηση αυτή επιτρέπει την αναγνώριση περιοχών στις οποίες ο πράκτορας έχει αποκτήσει περισσότερη εμπειρία, συμβάλλοντας στην καλύτερη ερμηνεία της αξιοπιστίας των αντίστοιχων αποφάσεων.

Οι συγκεκριμένες οπτικοποιήσεις χρησιμοποιούνται συμπληρωματικά προς τους ποσοτικούς δείκτες αξιολόγησης και συμβάλλουν στην κατανόηση της συμπεριφοράς του μοντέλου, παρέχοντας μια πιο ολοκληρωμένη εικόνα της πολιτικής δυναμικής τιμολόγησης που προέκυψε από τη διαδικασία της ενισχυτικής μάθησης.

4.2.5.9. Συνοπτική περιγραφή του πειραματικού σεναρίου

Συνοψίζοντας, η παρούσα μελέτη εξετάζει το πρόβλημα της δυναμικής τιμολόγησης ευπαθών προϊόντων λιανικής μέσω τεχνικών RL. Για τον σκοπό αυτό αναπτύχθηκε ένα προσομοιωμένο περιβάλλον λήψης αποφάσεων, στο οποίο ένας πράκτορας καλείται να επιλέγει καθημερινές μεταβολές τιμών με στόχο τη βελτιστοποίηση της συνολικής επιχειρησιακής απόδοσης.

Κάθε προϊόν αντιμετωπίζεται ως ανεξάρτητο πρόβλημα λήψης αποφάσεων. Σε κάθε χρονικό βήμα ο πράκτορας παρατηρεί την κατάσταση του προϊόντος, η οποία περιλαμβάνει πληροφορίες σχετικά με το διαθέσιμο απόθεμα, την τιμή, την υπολειπόμενη διάρκεια ζωής, τον ρυθμό αλλοίωσης, την ελαστικότητα της ζήτησης και πιθανούς εξωτερικούς παράγοντες που επηρεάζουν τη συμπεριφορά των καταναλωτών. Με βάση τις πληροφορίες αυτές επιλέγει μία από τις διαθέσιμες ενέργειες μεταβολής τιμής. Η απόφαση επηρεάζει τη ζήτηση, τις πωλήσεις, τις απώλειες αποθέματος και τελικά την ανταμοιβή που λαμβάνει από το περιβάλλον.

Η διαδικασία εκπαίδευσης πραγματοποιείται μέσω του αλγορίθμου Q-Learning, ο οποίος ενημερώνει σταδιακά τις εκτιμήσεις του σχετικά με τη μακροχρόνια αξία κάθε ενέργειας σε διαφορετικές καταστάσεις. Η εκπαίδευση επαναλαμβάνεται για πολλαπλές τιμές random seed ώστε να περιοριστεί η επίδραση της τυχαιότητας και να ενισχυθεί η αξιοπιστία των αποτελεσμάτων.

Μετά την ολοκλήρωση της εκπαίδευσης, η πολιτική που προκύπτει από τον πράκτορα αξιολογείται στο ίδιο περιβάλλον και συγκρίνεται με μία ευρετική (heuristic) πολιτική αναφοράς βασισμένη σε προκαθορισμένους επιχειρησιακούς κανόνες. Η σύγκριση πραγματοποιείται με βάση δείκτες κερδοφορίας, εσόδων, απωλειών λόγω αλλοίωσης και χαμένων πωλήσεων, ενώ συμπληρώνεται από οπτικοποιήσεις των πολιτικών λήψης αποφάσεων και της κάλυψης του χώρου καταστάσεων.

Η συνολική πειραματική διαδικασία αποσκοπεί στη διερεύνηση του κατά πόσο μία πολιτική δυναμικής τιμολόγησης που προκύπτει μέσω της ενισχυτικής μάθησης μπορεί να υπερβεί την απόδοση παραδοσιακών rule-based στρατηγικών στη διαχείριση ευπαθών προϊόντων λιανικής.

5. Αποτελέσματα και συζήτηση

Σε αυτή την ενότητα θα παρουσιαστούν και θα σχολιαστούν τα συνολικά αποτελέσματα της παρούσας εργασίας. Ειδικότερα θα γίνει παρουσίαση των αποτελεσμάτων των αρχικών δεδομένων, της πρόβλεψης της ζήτησης, της ABC ανάλυσης, του συνόλου δεδομένων για

το RL περιβάλλον αλλά και τα τελικά αποτελέσματα του πράκτορα με την σύγκριση της heuristic baseline μεθόδου.

5.1. Αποτελέσματα δεδομένων

Σε αυτή την υποενότητα θα παρουσιαστούν τα αρχικά δεδομένα ώστε ο αναγνώστης να έχει μια πρώτη εικόνα για τα δεδομένα πάνω στα οποία υλοποιήθηκε το πείραμα (Πίνακας 5.1).

Όνομα αρχείου	Πλήθος εγγραφών	Βασικές μεταβλητές	Ρόλος του συνόλου δεδομένων στην ανάλυση
calendar _events.csv	80	Date, event name, event type, holiday flag, promo flag, payday flag, uplift multiplier, notes	Βασικό ημερολόγιο αργιών και γεγονότων που επηρεάζει την λειτουργία του καταστήματος και παίζει ρόλο στην ζήτηση των προϊόντων.
products. csv	130	Product id, name, category, sub category, unit, service criticality, base price, unit cost, gross margin %, shelf life days, disposal cost per unit	Βασικό σύνολο δεδομένων με όλα τα προϊόντα και τα στοιχεία τους που αργότερα ταξινομούνται και επεξεργάζονται σε όλα τα στάδια της εργασίας.
transacti ons_dail y.csv	95030	Date, product id, units sold, price, discount %, promo flag, revenue, cost of all units sold,	Βασικό σύνολο δεδομένων με τις συναλλαγές για όλα τα προϊόντα σε όλες τις ημερομηνίες με στοιχεία κόστους και πωλήσεων.

		gross profit, stockout flag, lost sales unit	
waste_data_ily.csv	36768	Date, product id, quantity waste, waste cost	Βασικό σύνολο δεδομένων των απορρίψεων και του κόστους σε κάθε ημερομηνία που χρησιμοποιείτε για τον υπολογισμό του κέρδους και του κόστους απορρίψεων .

Πίνακας 5.1 αρχικών δεδομένων

5.1.1. Γεγονότα ημερολογίου

Σε αυτή την υποενότητα θα αναφερθεί ένα παράδειγμα δεδομένων του αρχείου calendar_events.csv και λίγα λόγια για τις μεταβλητές και τον ρόλο τους (Πίνακας 5.2).

Date	Event name	Event type	Holiday flag	Promo flag	Payday flag	Uplift multiplier	notes
1/1/2024	New Year	holiday	1	0	0	1.1	Fixed-date event
6/1/2024	Epiphany	holiday	1	0	0	1.04	Fixed-date event
14/2/2024	Valentine's	event	0	0	0	1.03	Fixed-date event
25/3/2024	25 March	holiday	1	0	0	1.08	Fixed-date event

Πίνακας 5.2 Δεδομένων γεγονότων

- **Date:** Η ημερομηνία εκφράζει την κάθε ημέρα που υπάρχει ένα συγκεκριμένο γεγονός, είτε αυτό είναι αργία ή κάποιο ειδικό γεγονός που έχει αντίκτυπο στην ζήτηση, για παράδειγμα Χριστούγεννα.
- **Event name:** Το όνομα του γεγονότος.
- **Event type:** Το είδος του γεγονότος με πιθανές τιμές: Event / Holiday / payday / seasonal.
- **Holiday flag:** Δυαδική μεταβλητή που εκφράζεται με τις τιμές 0 ή 1 και σηματοδοτεί την ύπαρξη αργίας.
- **Promo flag:** Δυαδική μεταβλητή που εκφράζεται με τις τιμές 0 ή 1 και σηματοδοτεί την ύπαρξη έκπτωσης.
- **Payday flag:** Δυαδική μεταβλητή που εκφράζεται με τις τιμές 0 ή 1 και σηματοδοτεί την ύπαρξη γεγονότος πληρωμής των καταναλωτών που εκφράζεται σε αυξημένη ζήτηση.
- **Uplift multiplier:** συντελεστής αύξησης της ζήτησης.
- **Notes:** σημειώσεις για κάθε γεγονός.

5.1.2. Προϊόντα

Τα συνολικά προϊόντα ανέρχονται σε 130 διαφορετικούς κωδικούς. Οι κωδικοί ομαδοποιούνται στον παρακάτω πίνακα (Πίνακας 5.3) με τα εξής χαρακτηριστικά.

Κατηγορία προϊόντος	Πλήθος προϊόντων	Εύρος αρχικών τιμών	Διάρκεια ζωής σε ημέρες	Εύρος κόστους απόρριψης ανά μονάδα
Κρέας	18	6,28 - 12,73	2 - 5	0,05 – 0,76
Ψάρι	10	9,17 – 18,47	1 - 3	0,10 – 0,74
Γαλακτοκομικά	18	0,93 – 6,07	5 - 13	0,07 – 0,76
Προϊόντα φούρνου	10	0,82 – 1,95	3 - 5	0,10 – 0,73
Delicatessen	10	2,18 – 4,41	4 - 10	0,10 – 0,80
Φρούτα	12	1,18 – 6,05	4 - 9	0,10 – 0,80
Προϊόντα παντοπωλείου	28	1,05 – 9,80	136 - 355	0,07 – 0,79
Λαχανικά	24	0,94 – 3,37	3 - 10	0,06 – 0,72

Πίνακας 5.3 Σύνοψη προϊόντων

5.1.3. Ημερήσιες συναλλαγές

Σε αυτή την υποενότητα θα γίνει η στατιστική περιγραφή του αρχείου transactions_daily.csv (Πίνακας 5.4) που αποτελείται από 95030 εγγραφές. Το αρχείο περιλαμβάνει τις ημερήσιες πωλήσεις για όλα τα 130 προϊόντα καθημερινά ανάμεσα στο διάστημα 1/1/2024 – 31/12/2025.

Στήλη	Μέσος όρος	Διάμεση τιμή	Ελάχιστη τιμή	Μέγιστη τιμή
Ημερήσιες μονάδες πώλησης	19,75	16	0	199
Τιμές πώλησης	4,3	2,33	0,62	18,5
Αξία πωλήσεων προϊόντων ανά ημέρα (revenue)	70,40	41,82	0	940,22

Κόστος πωληθέντων ανά πωληθείσα μονάδα(cogs)	53,85	32,16	0	695,5
Μεικτό κέρδος (revenue – cogs)	16,54	8,96	-44,2	280,9
Χαμένες πωλήσεις	3,35	0	0	192

Πίνακας 5.4 Περιγραφικής στατιστικής ημερήσιων συναλλαγών

5.1.4. Ημερήσιες απορρίψεις

Το αρχείο waste_daily.csv αποτελείται από 36768 εγγραφές για το διάστημα 1/1/2024 – 31/12/2025. Σε αυτό το διάστημα οι συνολικές απορρίψεις ανήλθαν σε 73303 μονάδες και το συνολικό κόστος σε 273460,36€. Ο μέσος όρος μονάδων απορρίψεων ανά προϊόν ανά ημέρα ανέρχεται σε 1,99 ενώ ο μέσος όρος του κόστους ανά προϊόν ανά ημέρα ανέρχεται σε 7,43€.

5.2. Αποτελέσματα της πρόβλεψης της ζήτησης

Σε αυτή την ενότητα θα παρουσιαστούν και θα σχολιαστούν τα αποτελέσματα από την πρόβλεψη της ζήτησης που έγινε στα παραπάνω δεδομένα.

5.2.1. Αξιολόγηση του μοντέλου NeuralProphet

Για την αξιολόγηση της προγνωστικής ικανότητας του μοντέλου NeuralProphet εφαρμόστηκε χρονικός διαχωρισμός των δεδομένων σε σύνολο εκπαίδευσης (80%) και σύνολο ελέγχου (20%). Η εκπαίδευση πραγματοποιήθηκε σε 584 ημέρες ιστορικών δεδομένων, ενώ οι τελευταίες 147 ημέρες χρησιμοποιήθηκαν αποκλειστικά για την αξιολόγηση των προβλέψεων. Το μοντέλο εκπαιδεύτηκε σε συνολικά 130 διαφορετικά

προϊόντα (SKU), αξιοποιώντας κοινό μοντέλο πολλαπλών χρονοσειρών. Τα στοιχεία και τα αποτελέσματα αποτυπώνονται παρακάτω στους πίνακες (Πίνακες 5.5 , 5.6).

Ποσοστιαίος διαχωρισμός εκπαίδευσης	Ημέρες εκπαίδευσης	Ημέρες για την αξιολόγηση της πρόβλεψης	Αριθμός προϊόντων
80-20	584	147	130

Πίνακας 5.5 Πειραματικής ρύθμισης NeuralProphet

Δείκτης	Τιμή
Μέσο απόλυτο σφάλμα (MAE)	6,898
Τετραγωνική ρίζα του μέσου απόλυτου σφάλματος (RMSE)	9,728
Μέσο απόλυτο ποσοστιαίο σφάλμα (MAPE %)	166,413 %
Συμμετρικό μέσο απόλυτο ποσοστιαίο σφάλμα (sMAPE %)	69,103 %

Πίνακας 5.6 Αξιολόγησης σφαλμάτων της πρόβλεψης

Οι δείκτες MAE και RMSE υποδηλώνουν ότι το μοντέλο πέτυχε ικανοποιητική ακρίβεια σε απόλυτους όρους. Συγκεκριμένα, η μέση απόκλιση των προβλέψεων από τις πραγματικές τιμές ανέρχεται σε περίπου 7 μονάδες προϊόντος ανά ημέρα. Η υψηλότερη τιμή του RMSE σε σχέση με το MAE υποδεικνύει την ύπαρξη ορισμένων περιπτώσεων με μεγαλύτερα σφάλματα πρόβλεψης, οι οποίες επηρεάζουν δυσανάλογα τον συγκεκριμένο δείκτη.

Οι ποσοστιαίες μετρικές MAPE και sMAPE εμφανίζουν σημαντικά υψηλότερες τιμές. Το γεγονός αυτό οφείλεται κυρίως στην παρουσία προϊόντων με χαμηλή ή μηδενική ημερήσια ζήτηση, όπου ακόμη και μικρές αποκλίσεις οδηγούν σε πολύ υψηλά ποσοστιαία σφάλματα. Επίσης, το μοντέλο δεν λαμβάνει υπόψιν τις ημέρες που το κατάστημα δεν λειτουργεί. Για τον λόγο αυτό, οι δείκτες MAE και RMSE θεωρούνται περισσότερο αντιπροσωπευτικοί για την αξιολόγηση του μοντέλου στη συγκεκριμένη εφαρμογή.

Τέλος, τα αποτελέσματα που προκύπτουν από το αρχείο actual_vs_forecast_full.csv θεωρούνται επαρκή για τη συνέχεια του πειράματος, καθώς το μοντέλο αποτυπώνει ικανοποιητικά τη γενική τάση της ζήτησης. Συγκεκριμένα, η μέση προβλεπόμενη ζήτηση

($\hat{y}$) ανήλθε σε 21,37 μονάδες, ενώ η μέση πραγματική ζήτηση ήταν 19,47 μονάδες. Το μέσο σφάλμα πρόβλεψης ήταν -1,91 μονάδες, γεγονός που υποδηλώνει ελαφρά υπερεκτίμηση της ζήτησης από το μοντέλο. Επιπλέον, το μέσο απόλυτο σφάλμα (MAE) διαμορφώθηκε στις 6,90 μονάδες, ενώ ο συντελεστής μεταβλητότητας (CV) ήταν 0,75, υποδεικνύοντας σχετικά υψηλή διακύμανση της ζήτησης.

5.2.2. Μετρικά δεδομένα ανά κωδικό προϊόντος (SKUs)

Πέρα από την συνολική αξιολόγηση του μοντέλου, έγινε μια μέτρηση και για το κάθε προϊόν (SKU) ξεχωριστά. Οι δέκα καλύτερες και οι δέκα χειρότερες εκτιμήσεις αποτυπώνονται στους παρακάτω πίνακες (Πίνακες 5.7, 5.8).

product _id	test_da ys	MAE	RMSE	MAPE_ %	sMAPE _%	y_mean_t est	y_sum_tr ain
P050	147	1,985842	2,5919 62	107,2191	87,7670 3	3,142857	1844
P051	147	2,109372	2,8881 53	97,17491	81,1060 7	3,768707	2357
P066	147	2,149104	2,7884 75	85,52622	84,0766 8	3,877551	2220
P064	147	2,342853	2,9126 06	101,8553	83,0648 4	4,210884	2642
P069	147	2,651182	3,4442 18	90,25729	78,7232 6	4,891156	3003
P017	147	2,799487	4,0942 16	110,0159	78,3522 7	5,986395	3648
P023	147	3,128811	4,4226 86	106,6938	77,4155 2	6,44898	3948
P019	147	3,1321	4,3549 3	108,7149	76,7312 1	7,734694	4594

P054	147	3,146994	4,1675 72	107,3142	82,2177 7	5,734694	3602
P056	147	3,151781	4,5758 79	98,40605	73,5167 3	7,306122	4298

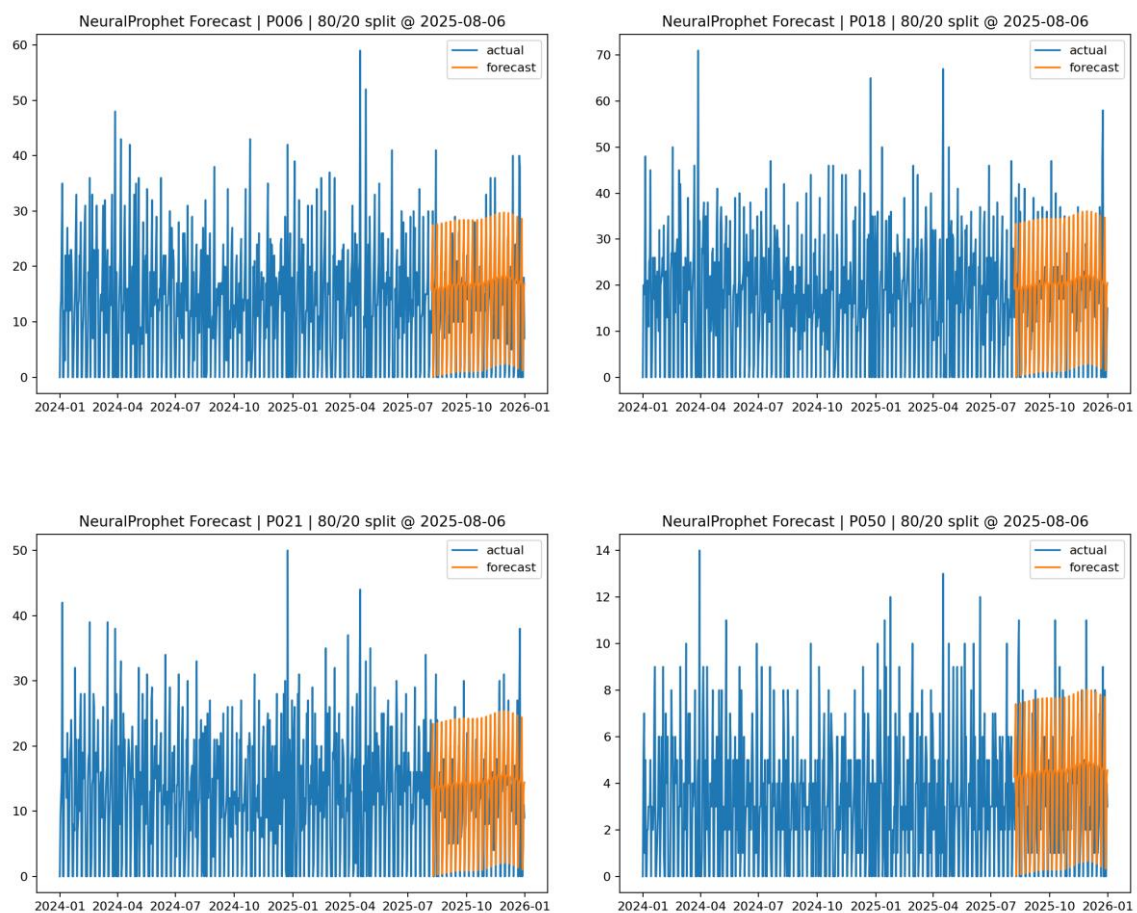
Πίνακας 5.7 Οι δέκα καλύτερες εκτιμήσεις MAE / RMSE

product_ id	test_da ys	MAE	RMSE	MAPE_ %	sMAPE_ %	y_mean_t est	y_sum_tra in
P102	147	16,376 56	22,130 45	354,407 4	69,67859	43,98639	26180
P088	147	16,021 71	21,409 66	280,979 1	65,10767	43,17687	26113
P096	147	15,29	21,425 47	312,178 1	63,39871	45,83673	27379
P083	147	14,560 07	20,916 66	351,971 5	63,20533	49,29932	29638
P094	147	14,428 71	23,722 47	386,493 2	63,00993	47,18367	27509
P097	147	14,105 24	20,784 78	348,423 7	67,85923	43,14966	25083
P109	147	14,033 37	20,861 54	302,840 8	64,73925	43,46939	25816
P095	147	13,797 76	18,163 79	319,125 9	69,7116	39,68027	23483
P110	147	12,350 15	17,208 86	273,806 2	66,86096	36,02721	21616
P089	147	12,024 03	18,220 13	292,057	63,21535	40,19048	23819

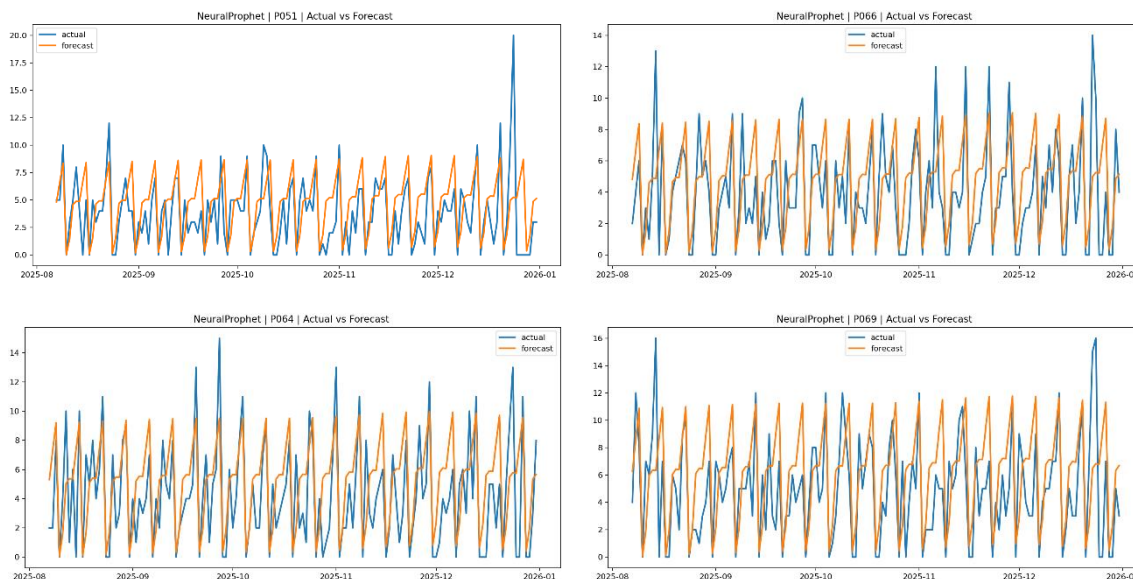
Πίνακας 5.8 Οι δέκα χειρότερες εκτιμήσεις MAE / RMSE

5.2.3. Οπτική αξιολόγηση προβλέψεων

Στις εικόνες 5.1 και 5.2 παρουσιάζονται ενδεικτικά παραδείγματα σύγκρισης μεταξύ πραγματικής ζήτησης (y) και προβλεπόμενης ζήτησης ($\hat{y}_{hat1}$) για επιλεγμένα προϊόντα σε δύο διαφορετικά είδη γραφήματος για την καλύτερη κατανόηση του αναγνώστη.



Εικόνα 5.1 Αποτελέσματα πρόβλεψης για 4 αντιπροσωπευτικά προϊόντα σε ιστόγραμμα



Εικόνα 5.2 Αποτελέσματα πρόβλεψης για 4 αντιπροσωπευτικά προϊόντα σε line plot

Παρατηρείται ότι το μοντέλο ακολουθεί με ικανοποιητική ακρίβεια τη γενική τάση της ζήτησης, ενώ καταφέρνει να αναπαραστήσει τα περισσότερα εποχικά μοτίβα που εμφανίζονται στη χρονοσειρά. Οι μεγαλύτερες αποκλίσεις εμφανίζονται κυρίως σε περιόδους αιχμής, όπου η ζήτηση μεταβάλλεται απότομα. Παρά τις αποκλίσεις αυτές, η συνολική συμπεριφορά της χρονοσειράς αποτυπώνεται ικανοποιητικά.

Παρά τις ακραίες διακυμάνσεις και τον έντονο θόρυβο, το μοντέλο καταφέρνει να αναγνωρίσει τόσο την υποκείμενη τάση όσο και τα επαναλαμβανόμενα εποχικά μοτίβα της ζήτησης.

5.2.4. Σχολιασμός αποτελεσμάτων της πρόβλεψης της ζήτησης

Τα αποτελέσματα δείχνουν ότι το NeuralProphet μπορεί να αποτυπώσει αποτελεσματικά τη συμπεριφορά της ζήτησης σε επίπεδο προϊόντος. Η αξιοποίηση εβδομαδιαίας και ετήσιας εποχικότητας επέτρεψε στο μοντέλο να αναγνωρίσει επαναλαμβανόμενα μοτίβα που σχετίζονται με τις ημέρες της εβδομάδας, τις εορταστικές περιόδους και τις γενικότερες τάσεις κατανάλωσης.

Παρότι ορισμένα προϊόντα παρουσίασαν αυξημένα ποσοστιαία σφάλματα λόγω χαμηλού όγκου πωλήσεων, η συνολική απόδοση του μοντέλου κρίθηκε επαρκής για τις ανάγκες της

παρούσας εργασίας. Οι προβλέψεις που παρήχθησαν αξιοποιούνται στα επόμενα στάδια της μελέτης για την υποστήριξη των αποφάσεων διαχείρισης αποθεμάτων και δυναμικής τιμολόγησης. Πιο συγκεκριμένα η πρόβλεψη της ζήτησης παίζει καθοριστικό ρόλο στην εξίσωση για την εύρεση της ελαστικότητας που είναι σημαντική εισροή στην ABC ανάλυση και αργότερα στην RL.

5.3. Αποτελέσματα της πολυκριτηριακής ABC ανάλυσης

Σε αυτή την υποενότητα αυτή παρουσιάζονται τα αποτελέσματα της πολυκριτηριακής ABC ανάλυσης των προϊόντων.

5.3.1. Ταξινόμηση προϊόντων και σχόλια

Η εφαρμογή της πολυκριτηριακής ABC ανάλυσης οδήγησε στην ταξινόμηση των 130 προϊόντων σε τρεις κατηγορίες επιχειρησιακής σημασίας. Συγκεκριμένα, 26 προϊόντα (20%) κατατάχθηκαν στην κατηγορία A, 39 προϊόντα (30%) στην κατηγορία B και 65 προϊόντα (50%) στην κατηγορία C.

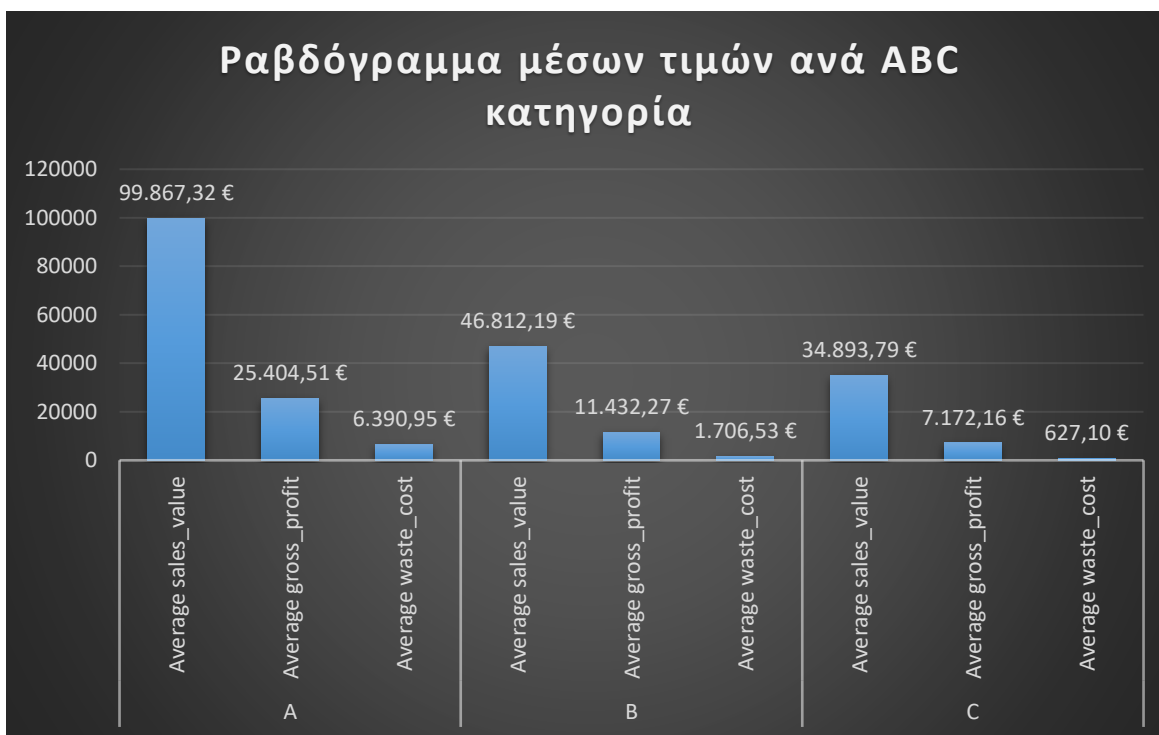
	Κατηγορία A	Κατηγορία B	Κατηγορία C
Πλήθος προϊόντων	26	39	65
Ποσοστό των προϊόντων	20%	30%	50%
Μέση αξία πωλήσεων	99867,31	46812,19	34893,78
Μέσο μεικτό κέρδος	25404,51	11432,27	7172,16
Μέσο κόστος απορρίψεων	6390,94	1706,53	627,09

Πίνακας 5.9 Δεδομένα ανά ABC κατηγορία προϊόντος

Όπως διακρίνεται και στον πίνακα παραπάνω (Πίνακας 5.9) τα προϊόντα της κατηγορίας Α παρουσίασαν τις υψηλότερες μέσες τιμές πωλήσεων και κερδοφορίας. Συγκεκριμένα, η μέση αξία πωλήσεων ανήλθε σε περίπου 99.867 €, ενώ το μέσο μικτό κέρδος διαμορφώθηκε στις 25.405 €. Παράλληλα, τα προϊόντα αυτά εμφάνισαν και το υψηλότερο κόστος απωλειών (6.391 € κατά μέσο όρο), γεγονός που υποδηλώνει ότι απαιτούν ιδιαίτερη προσοχή στη διαχείριση αποθεμάτων.

Αντίστοιχα, τα προϊόντα της κατηγορίας Β παρουσίασαν μέση αξία πωλήσεων 46.812 € και μέσο μικτό κέρδος 11.432 €, καταλαμβάνοντας ενδιάμεση θέση μεταξύ των κατηγοριών Α και Γ.

Τέλος, τα προϊόντα της κατηγορίας Γ εμφάνισαν τις χαμηλότερες επιδόσεις, με μέση αξία πωλήσεων 34.894 € και μέσο μικτό κέρδος 7.172 €, ενώ παράλληλα παρουσίασαν σημαντικά χαμηλότερο κόστος απωλειών.



Εικόνα 5.3 Ραβδόγραμμα μέσων τιμών ανά ABC κατηγορία

5.3.2. Πίνακες ABC

Παρακάτω δίνονται οι πίνακες για τα προϊόντα Α (Πίνακας 5.10) ,Β (Πίνακας 5.11) και C (Πίνακας 5.12).

product_id	name	category	abc_score	score_share	cumulative_share	ABC_class
P024	Cod kg	Fish	0,66525214	0,01425213	0,01425213	A
P022	Cod pack	Fish	0,638716478	0,013683639	0,027935768	A
P008	Beef Mince pack	Meat	0,606876531	0,013001511	0,040937279	A
P026	Sea Bream kg	Fish	0,603710903	0,012933692	0,053870971	A
P012	Pork Chops pack	Meat	0,589784949	0,012635347	0,066506318	A
P021	Sea Bream kg	Fish	0,56972726	0,012205638	0,078711956	A
P003	Sausages pack	Meat	0,560227316	0,012002114	0,09071407	A
P023	Shrimps kg	Fish	0,551827054	0,01182215	0,10253622	A
P028	Cod pack	Fish	0,551448464	0,011814039	0,11435026	A
P002	Sausages pack	Meat	0,536079846	0,011484787	0,125835047	A
P020	Cod pack	Fish	0,512922398	0,01098867	0,136823718	A
P027	Salmon kg	Fish	0,511393994	0,010955926	0,147779644	A
P009	Chicken Breast pack	Meat	0,503222051	0,010780854	0,158560498	A
P007	Turkey Fillet pack	Meat	0,499094882	0,010692435	0,169252932	A
P017	Pork Chops kg	Meat	0,496335592	0,01063332	0,179886253	A

P018	Beef Mince kg	Meat	0,4913627 88	0,0105267 85	0,190413037	A
P010	Beef Mince pack	Meat	0,4884349 56	0,0104640 6	0,200877097	A
P015	Sausages kg	Meat	0,4868262 85	0,0104295 96	0,211306694	A
P006	Sausages pack	Meat	0,4857297 85	0,0104061 05	0,221712799	A
P005	Sausages pack	Meat	0,4808419 12	0,0103013 89	0,232014188	A
P016	Pork Chops kg	Meat	0,4758905 37	0,0101953 13	0,242209501	A
P019	Tuna pack	Fish	0,4754565 33	0,0101860 15	0,252395516	A
P025	Sea Bream pack	Fish	0,4702508 33	0,0100744 9	0,262470006	A
P116	Pita pack	Bakery	0,4686806 98	0,0100408 52	0,272510857	A
P117	Muffin pack	Bakery	0,4622179 31	0,0099023 96	0,282413253	A
P011	Beef Mince pack	Meat	0,4598049 09	0,0098507	0,292263953	A

Πίνακας 5.10 Κατηγορία A

product_ id	name	category	abc_score	score_sh are	cumulative_sh are	ABC_cla ss
P001	Sausages kg	Meat	0,4568685 51	0,0097877 92	0,302051745	B
P014	Chicken Breast kg	Meat	0,4560177 51	0,0097695 65	0,31182131	B
P046	Cheese L	Dairy	0,4420794 1	0,0094709 55	0,321292265	B
P004	Pork Chops pack	Meat	0,4286133 95	0,0091824 64	0,330474729	B

P118	Bread pack	Bakery	0,4282652 21	0,0091750 05	0,339649733	B
P119	Bread pcs	Bakery	0,4240090 1	0,0090838 21	0,348733555	B
P013	Sausages kg	Meat	0,4151710 11	0,0088944 79	0,357628033	B
P113	Baguette pack	Bakery	0,4130400 63	0,0088488 26	0,366476859	B
P121	Feta Pack pack	Deli	0,4002086 04	0,0085739 29	0,375050789	B
P076	Apples kg	Fruits	0,3941918	0,0084450 28	0,383495816	B
P112	Bread pack	Bakery	0,3917821 96	0,0083934 05	0,391889221	B
P036	Cheese pack	Dairy	0,3909907 06	0,0083764 48	0,40026567	B
P092	Lettuce kg	Vegetabl es	0,3901361 08	0,0083581 4	0,40862381	B
P088	Tomatoes kg	Vegetabl es	0,3876424 23	0,0083047 16	0,416928525	B
P122	Feta Pack pack	Deli	0,3838356 79	0,0082231 62	0,425151687	B
P103	Lettuce kg	Vegetabl es	0,3770114 83	0,0080769 62	0,433228649	B
P115	Croissant pack	Bakery	0,3758189 68	0,0080514 14	0,441280064	B
P035	Cheese L	Dairy	0,3734538 87	0,0080007 46	0,44928081	B
P097	Onions kg	Vegetabl es	0,3723872 27	0,0079778 94	0,457258704	B
P101	Lettuce kg	Vegetabl es	0,3677329 29	0,0078781 82	0,465136885	B
P098	Onions kg	Vegetabl es	0,3594728 6	0,0077012 21	0,472838106	B
P108	Potatoes kg	Vegetabl es	0,3590330 48	0,0076917 99	0,480529905	B

P078	Apples kg	Fruits	0,3573471 36	0,0076556 8	0,488185585	B
P099	Lettuce kg	Vegetabl es	0,3546346 61	0,0075975 69	0,495783155	B
P080	Apples kg	Fruits	0,3538662 05	0,0075811 06	0,503364261	B
P086	Strawberries kg	Fruits	0,3538495 88	0,0075807 5	0,510945011	B
P084	Apples kg	Fruits	0,3535724 5	0,0075748 13	0,518519823	B
P100	Tomatoes kg	Vegetabl es	0,3507079 26	0,0075134 44	0,526033268	B
P110	Carrots kg	Vegetabl es	0,3504094 44	0,0075070 5	0,533540317	B
P130	Ready Salad pack	Deli	0,3491864 71	0,0074808 49	0,541021166	B
P093	Cucumbers kg	Vegetabl es	0,3465814 53	0,0074250 4	0,548446206	B
P081	Bananas kg	Fruits	0,3455674 29	0,0074033 16	0,555849522	B
P104	Lettuce kg	Vegetabl es	0,3455013 81	0,0074019 01	0,563251423	B
P102	Potatoes kg	Vegetabl es	0,3449644 7	0,0073903 98	0,570641822	B
P082	Grapes kg	Fruits	0,3422443 08	0,0073321 23	0,577973944	B
P126	Olives pack	Deli	0,3420609 22	0,0073281 94	0,585302138	B
P095	Carrots kg	Vegetabl es	0,3392689 08	0,0072683 79	0,592570517	B
P114	Croissant pcs	Bakery	0,3392397 28	0,0072677 54	0,59983827	B
P087	Potatoes kg	Vegetabl es	0,3390025 76	0,0072626 73	0,607100943	B

Πίνακας 5.11 Κατηγορία Β

product_id	name	category	abc_score	score_share	cumulative_share	ABC_class
P129	Ready Salad pack	Deli	0,33854617	0,007252895	0,614353838	C
P107	Onions kg	Vegetables	0,337277875	0,007225723	0,621579561	C
P091	Tomatoes kg	Vegetables	0,336523566	0,007209563	0,628789125	C
P077	Bananas kg	Fruits	0,334349195	0,00716298	0,635952105	C
P123	Salami pack	Deli	0,333735529	0,007149833	0,643101939	C
P073	Olive Oil pack	Groceries	0,331674261	0,007105674	0,650207612	C
P105	Tomatoes kg	Vegetables	0,33130747	0,007097816	0,657305428	C
P124	Olives pack	Deli	0,330670126	0,007084161	0,664389589	C
P037	Milk L	Dairy	0,330611609	0,007082908	0,671472497	C
P044	Yogurt pack	Dairy	0,330363948	0,007077602	0,678550099	C
P085	Bananas kg	Fruits	0,328604195	0,007039902	0,68559	C
P125	Salami pack	Deli	0,327426151	0,007014664	0,692604664	C
P111	Croissant pack	Bakery	0,32636145	0,006991854	0,699596518	C
P109	Onions kg	Vegetables	0,325810045	0,006980041	0,706576558	C
P094	Tomatoes kg	Vegetables	0,325681516	0,006977287	0,713553846	C

P039	Milk pack	Dairy	0,3256384 77	0,0069763 65	0,720530211	C
P079	Oranges kg	Fruits	0,3228269 97	0,0069161 33	0,727446343	C
P096	Carrots kg	Vegetabl es	0,3228195 18	0,0069159 73	0,734362316	C
P045	Yogurt L	Dairy	0,3225062 64	0,0069092 62	0,741271578	C
P075	Bananas kg	Fruits	0,3221694 56	0,0069020 46	0,748173624	C
P066	Rice pack	Grocerie s	0,3219189 11	0,0068966 78	0,755070302	C
P040	Cream L	Dairy	0,3218673 11	0,0068955 73	0,761965875	C
P043	Milk pack	Dairy	0,3199681 36	0,0068548 86	0,768820761	C
P042	Butter pack	Dairy	0,3182810 97	0,0068187 43	0,775639504	C
P032	Milk L	Dairy	0,3178730 11	0,00681	0,782449504	C
P034	Milk L	Dairy	0,3175618 38	0,0068033 34	0,789252838	C
P054	Olive Oil pack	Grocerie s	0,3171112 21	0,0067936 8	0,796046518	C
P058	Olive Oil pack	Grocerie s	0,3163044 48	0,0067763 96	0,802822914	C
P120	Croissant pack	Bakery	0,3149206 74	0,0067467 51	0,809569665	C
P038	Milk L	Dairy	0,3143624 66	0,0067347 92	0,816304457	C
P050	Canned Beans pack	Grocerie s	0,3141586 97	0,0067304 26	0,823034883	C
P031	Milk pack	Dairy	0,3139156 26	0,0067252 19	0,829760102	C
P106	Potatoes kg	Vegetabl es	0,3134685 43	0,0067156 41	0,836475742	C
P030	Cheese pack	Dairy	0,3094858 45	0,0066303 17	0,843106059	C

P128	Ham Slices pack	Deli	0,3079078	0,0065965 09	0,849702568	C
P127	Ham Slices pack	Deli	0,3074134 29	0,0065859 18	0,856288486	C
P033	Cream pack	Dairy	0,3071260 1	0,0065797 6	0,862868247	C
P065	Olive Oil pack	Groceries	0,3050507 25	0,0065353	0,869403547	C
P029	Butter pack	Dairy	0,3022020 69	0,0064742 72	0,875877818	C
P041	Milk L	Dairy	0,3002988 92	0,0064334 99	0,882311317	C
P089	Cucumbers kg	Vegetables	0,2888234 26	0,0061876 52	0,888498969	C
P090	Cucumbers kg	Vegetables	0,2883452 76	0,0061774 09	0,894676378	C
P083	Oranges kg	Fruits	0,2865536 51	0,0061390 25	0,900815403	C
P069	Flour pack	Groceries	0,2834107 2	0,0060716 92	0,906887096	C
P072	Olive Oil pack	Groceries	0,2792332 3	0,0059821 95	0,912869291	C
P067	Olive Oil pack	Groceries	0,2708314 63	0,0058021 99	0,91867149	C
P051	Flour pack	Groceries	0,2707626 59	0,0058007 25	0,924472214	C
P068	Coffee pack	Groceries	0,2475405 73	0,0053032 23	0,929775437	C
P052	Rice pack	Groceries	0,2392506	0,0051256 21	0,934901058	C
P061	Flour pack	Groceries	0,2342726 04	0,0050189 74	0,939920033	C

P060	Coffee pack	Groceries	0,222375956	0,004764105	0,944684138	C
P059	Canned Beans pack	Groceries	0,211431834	0,004529642	0,94921378	C
P056	Flour pack	Groceries	0,210969014	0,004519727	0,953733506	C
P070	Pasta pack	Groceries	0,207084848	0,004436513	0,958170019	C
P062	Flour pack	Groceries	0,206870671	0,004431925	0,962601945	C
P048	Rice pack	Groceries	0,195693453	0,004192468	0,966794413	C
P057	Rice pack	Groceries	0,19420955	0,004160678	0,97095509	C
P049	Pasta pack	Groceries	0,193726002	0,004150318	0,975105409	C
P064	Tea pack	Groceries	0,192417147	0,004122278	0,979227686	C
P053	Tea pack	Groceries	0,189609492	0,004062128	0,983289814	C
P063	Canned Beans pack	Groceries	0,177921257	0,003811723	0,987101537	C
P047	Cereal pack	Groceries	0,177039866	0,00379284	0,990894377	C
P074	Pasta pack	Groceries	0,152512602	0,003267377	0,994161754	C
P055	Canned Beans pack	Groceries	0,141564804	0,003032835	0,997194589	C
P071	Canned Beans pack	Groceries	0,130949257	0,002805411	1	C

Πίνακας 5.12 Κατηγορία C

5.3.3. Αποτελέσματα διερεύνησης σχέσεων μεταξύ κριτηρίων για την ABC ανάλυση

Για την αποφυγή της μέτρησης διπλότυπων μεταβλητών και την δίκαιη κατανομή των βαρών πραγματοποιήθηκε η διερεύνηση σχέσεων μεταξύ των μεταβλητών.

Τα αποτελέσματα αυτής της διερεύνησης αποτυπώνονται στο στιγμιότυπο (Εικόνα 5.4) παρακάτω.

	A	B	C	D	E	F	G	H	I	J
1		sales_value	gross_profit	cv_demand	spoilage_rate	shelf_life_days	waste_cost	stockout_frequency	service_criticality	subcategory_event_uplift
2	sales_value	1	0,949	-0,223	0,392	-0,134	0,736	0,128	-0,175	0,089
3	gross_profit	0,949	1	-0,222	0,449	-0,193	0,746	0,185	-0,163	0,099
4	cv_demand	-0,223	-0,222	1	-0,098	0,373	-0,084	-0,315	-0,169	-0,027
5	spoilage_rate	0,392	0,449	-0,098	1	-0,64	0,804	0,58	-0,32	0,016
6	shelf_life_days	-0,134	-0,193	0,373	-0,64	1	-0,378	-0,538	-0,006	-0,224
7	waste_cost	0,736	0,746	-0,084	0,804	-0,378	1	0,4	-0,31	-0,132
8	stockout_frequency	0,128	0,185	-0,315	0,58	-0,538	0,4	1	-0,079	-0,097
9	service_criticality	-0,175	-0,163	-0,169	-0,32	-0,006	-0,31	-0,079	1	0,006
10	subcategory_event_uplift	0,089	0,099	-0,027	0,016	-0,224	-0,132	-0,097	0,006	1

Εικόνα 5.4 Heatmap συσχέτιση μεταβλητών των βαρών

Η ανάλυση των συσχετίσεων ανέδειξε ορισμένες ιδιαίτερα ισχυρές σχέσεις μεταξύ των μεταβλητών που χρησιμοποιήθηκαν στο πολυκριτηριακό μοντέλο.

Η ισχυρότερη θετική συσχέτιση παρατηρήθηκε μεταξύ της αξίας πωλήσεων και του μικτού κέρδους ($r = 0,949$, $p < 0,001$). Το αποτέλεσμα αυτό υποδηλώνει ότι τα προϊόντα με υψηλότερη εμπορική δραστηριότητα τείνουν να συνεισφέρουν σημαντικά και στην κερδοφορία του καταστήματος. Η παρατήρηση αυτή επιβεβαιώνει τη σημασία των οικονομικών δεικτών στην αξιολόγηση των προϊόντων, χωρίς όμως να δικαιολογεί την αποκλειστική χρήση τους ως κριτήρια ταξινόμησης. Για τον λόγο αυτό αφαιρέθηκε από την εξίσωση των κριτηρίων μετά την παραπάνω ανάλυση.

Παράλληλα, παρατηρήθηκε πολύ ισχυρή θετική συσχέτιση μεταξύ του ποσοστού αλλοίωσης και του κόστους σπατάλης ($r = 0,804$, $p < 0,001$). Το εύρημα αυτό οδήγησε στην δικαιότερη κατανομή των βαρών ανάμεσα στις δύο αυτές μεταβλητές.

Ιδιαίτερο ενδιαφέρον παρουσιάζει και η ισχυρή αρνητική συσχέτιση μεταξύ της διάρκειας ζωής των προϊόντων και του ποσοστού αλλοίωσης ($r = -0,640$, $p < 0,001$). Το αποτέλεσμα αυτό επιβεβαιώνει ότι τα προϊόντα με μικρότερο shelf life εμφανίζουν σημαντικά αυξημένο

κίνδυνο απωλειών λόγω αλλοίωσης, γεγονός που τα καθιστά κρίσιμα για την εφαρμογή στρατηγικών δυναμικής τιμολόγησης.

Τέλος, η αρνητική συσχέτιση μεταξύ διάρκειας ζωής και συχνότητας ελλείψεων αποθέματος ($r = -0,538$, $p < 0,001$) υποδηλώνει ότι τα προϊόντα με μεγαλύτερη δυνατότητα αποθήκευσης προσφέρουν μεγαλύτερη ευελιξία στη διαχείριση αποθεμάτων και συνεπώς μικρότερο λειτουργικό κίνδυνο.

5.4. Αποτελέσματα δεδομένων για την χρήση της ενισχυτικής μάθησης

Σε αυτό το υποκεφάλαιο παρουσιάζεται η δομή του συνόλου δεδομένων που χρησιμοποιήθηκε για την εκπαίδευση και αξιολόγηση του αλγορίθμου Reinforcement Learning (Πίνακας 5.13).

Το RL dataset περιλαμβάνει 96.330 εγγραφές και 17 μεταβλητές, οι οποίες περιγράφουν λειτουργικά, εμπορικά και ποιοτικά χαρακτηριστικά των προϊόντων. Κάθε εγγραφή αντιστοιχεί σε μία χρονική παρατήρηση συγκεκριμένου προϊόντος και περιλαμβάνει πληροφορίες σχετικά με τη ζήτηση, τις απώλειες λόγω αλλοίωσης, τις πωλήσεις, καθώς και εξωγενείς παράγοντες που επηρεάζουν τη συμπεριφορά της αγοράς.

Κατηγορία	Μεταβλητές	Περιγραφή
Αναγνωριστικά	Product id, category, ABC class	Ταυτοποίηση προϊόντος
Μεταβλητές κατάστασης	price, shelf life days, spoilage rate, event flag, event uplift signal, service criticality, elasticity, day of week	Κατάσταση περιβάλλοντος που παρατηρεί ο agent

Μεταβλητές ζήτησης	Units sold, real demand, lost sales	Στοιχεία πραγματικής και ανεκπλήρωτης ζήτησης
Μεταβλητές απορρίψεων	Qty wasted, waste cost	Απώλειες λόγω αλλοίωσης
Ενέργειες του πράκτορα	-30%, -20%, -10%, -5%, 0%, 5%, 10%, 20%	Δυνατές αποφάσεις του πράκτορα
Reward components	Profit, Waste Cost, Lost Profit Cost	Συνιστώσες της συνάρτησης ανταμοιβής

Πίνακας 5.13 Δομή του RL συνόλου δεδομένων

5.5. Αποτελέσματα της ενισχυτικής μάθησης

Η καταγραφή των αποτελεσμάτων του πράκτορα έγινε σε δύο αρχεία το πρώτο αρχείο είναι η γενική καταγραφή των τιμών που αποτυπώνει όλες τι αρχικές τιμές από το πείραμα για κάθε seed. Ενδεικτικά αποτυπώνονται τα κύρια αποτελέσματα για 2 seeds στον παρακάτω πίνακα (Πίνακας 5.14).

product_id	rl_reward	heuristic_reward	improvement	improvement_pct	rl_profit
Seed 1					
P002	28400,25	-2562,77	30963,02	1208,19	44141,12
P003	26245,9	-1907,36	28153,26	1476,03	39284,42
P005	18238,52	-432,23	18670,75	4319,67	26401,49
P006	12657,83	-5491,28	18149,11	330,51	21830,78
P007	13099,06	-8572,01	21671,07	252,81	20837,47
P008	35775,23	-12525,71	48300,94	385,61	55172,31
P009	21393,81	-8482,16	29875,97	352,22	30958,89
P010	15955,42	-1308,74	17264,16	1319,14	26057,27
P011	15317,44	-5669,42	20986,85	370,18	23209,61
P012	25647,84	-11635,56	37283,4	320,43	38586,48
P015	21826,65	-1218,06	23044,71	1891,92	31802,05
P016	11224,77	-4612,97	15837,74	343,33	17683,46

P017	7878,71	-6566,11	14444,82	219,99	12074,23
P018	24712,85	-2282,5	26995,36	1182,71	36415,68
P019	8604,21	-9262,44	17866,65	192,89	21394,71
P020	6558,93	-11764,22	18323,15	155,75	17762,08
P021	8721,89	-11741,16	20463,04	174,28	24583,98
P022	12652,09	-15329,78	27981,87	182,53	33616,41
P023	7386,99	-9921,03	17308,02	174,46	23153,94
P024	12660,35	-17247,44	29907,79	173,4	39461,93
P025	8989,16	-8620,08	17609,24	204,28	22416,16
P026	10311,49	-15277,59	25589,08	167,49	39660,43
P027	11551,49	-13644,74	25196,22	184,66	30513,58
P028	13022,29	-19837,38	32859,67	165,65	31926,8
P116	8765,14	-5663,36	14428,5	254,77	15249,27
P117	1828,34	-980,02	2808,36	286,56	3601,19
Seed 7					
P002	27477,51	-2562,77	30040,27	1172,18	43582,59
P003	26562,65	-1907,36	28470,01	1492,64	39533,01
P005	18466,03	-432,23	18898,26	4372,31	26634,61
P006	14415,94	-5491,28	19907,22	362,52	22004,18
P007	13206,89	-8572,01	21778,9	254,07	20853,84
P008	35539,65	-12525,71	48065,36	383,73	51747,44
P009	21100,65	-8482,16	29582,81	348,77	30825,38
P010	18936,62	-1308,74	20245,36	1546,93	27186,64
P011	14898,31	-5669,42	20567,73	362,78	22591,82
P012	27066,05	-11635,56	38701,61	332,61	40792
P015	18639,95	-1218,06	19858,01	1630,3	28376,84
P016	13134,18	-4612,97	17747,15	384,72	19189,94
P017	7461,46	-6566,11	14027,57	213,64	11485,11
P018	20222,21	-2282,5	22504,72	985,97	35258,51
P019	7723,75	-9262,44	16986,18	183,39	20253,3

P020	6792,66	-11764,22	18556,88	157,74	17924,05
P021	8764,27	-11741,16	20505,42	174,65	23398,72
P022	13033,33	-15329,78	28363,12	185,02	34630,63
P023	9220,05	-9921,03	19141,08	192,93	22711,85
P024	16126,83	-17247,44	33374,27	193,5	40457,39
P025	9195,25	-8620,08	17815,33	206,67	22635,53
P026	13470,67	-15277,59	28748,26	188,17	38111,69
P027	11203,4	-13644,74	24848,14	182,11	27594,26
P028	8138,83	-19837,38	27976,21	141,03	33357,14
P116	8955,5	-5663,36	14618,86	258,13	14703,77
P117	2077,22	-980,02	3057,23	311,96	3494,36

Πίνακας 5.14 Κύρια αποτελέσματα του πράκτορα σε σχέση με την ευρετική μέθοδο

Για την καλύτερη κατανόηση του αναγνώστη, δημιουργήθηκε ακόμα ένα αρχείο με ποσοστά που συγκρίνει αυτές τις δύο πολιτικές συγκεντρωτικά (Πίνακας 5.15).

product_id	profit_improvement_pct_mean	profit_improvement_pct_std	revenue_improvement_pct_mean	revenue_improvement_pct_std
P002	9,85	6,71	1,35	1,98
P003	10,16	2,65	1,62	0,67
P005	17,69	2,16	3,53	0,66
P006	200	0	37,2	0,6
P007	112,15	1,73	23,39	0,61
P008	200	0	37,16	1,21
P009	12,73	3,17	2,29	0,83
P010	15,26	1,77	2,38	0,35
P011	200	0	37,04	0,92
P012	12,99	4,06	2,28	1,15
P015	14,31	5,48	2,72	1,54

P016	200	0	36,62	0,79
P017	112,59	2,11	23,57	0,72
P018	14,67	3,47	2,73	0,86
P019	108,74	2,53	20,91	0,74
P020	105,64	0,85	19,08	0,29
P021	108,79	2,23	20,08	1,02
P022	200	0	30,97	0,31
P023	200	0	31,7	0,58
P024	200	0	31,49	0,64
P025	200	0	34,49	0,79
P026	200	0	32,95	1,63
P027	109,34	3,34	20,79	1,08
P028	109,75	3,14	20,77	1,25
P116	120,02	0,94	28,78	0,49
P117	122,74	0,82	30,52	0,51

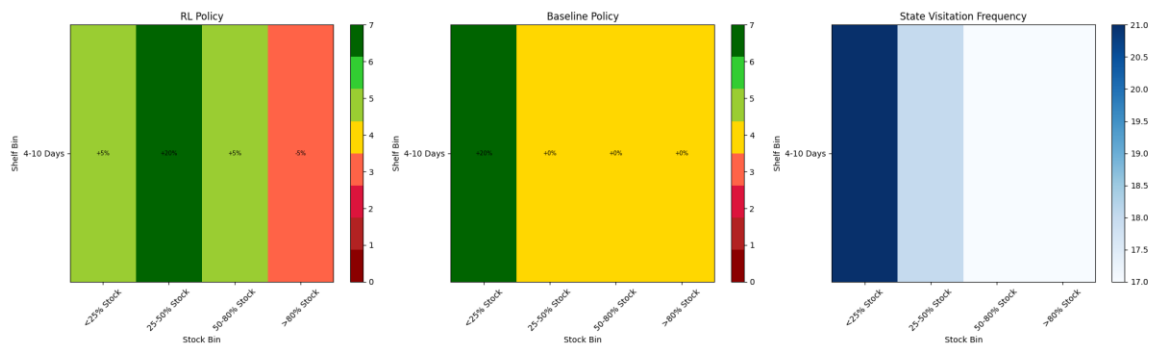
Πίνακας 5.15 RL vs Baseline

Οι δείκτες βελτίωσης εκφράζονται σε όρους συμμετρικής ποσοστιαίας μεταβολής, καθώς η κλασική ποσοστιαία μεταβολή μπορεί να οδηγήσει σε υπερβολικά μεγάλες τιμές όταν το baseline λαμβάνει πολύ μικρές ή αρνητικές τιμές. Η χρήση της συμμετρικής μορφής εξομαλύνει το πρόβλημα αυτό, παρέχοντας πιο σταθερά και συγκρίσιμα αποτελέσματα. Επιπλέον, η μέγιστη θεωρητική τιμή του δείκτη περιορίζεται στο 200%, γεγονός που βελτιώνει σημαντικά την αναγνωσιμότητα και την ερμηνεία των αποτελεσμάτων. Ειδικότερα αυτό εξηγείται από την σχέση:

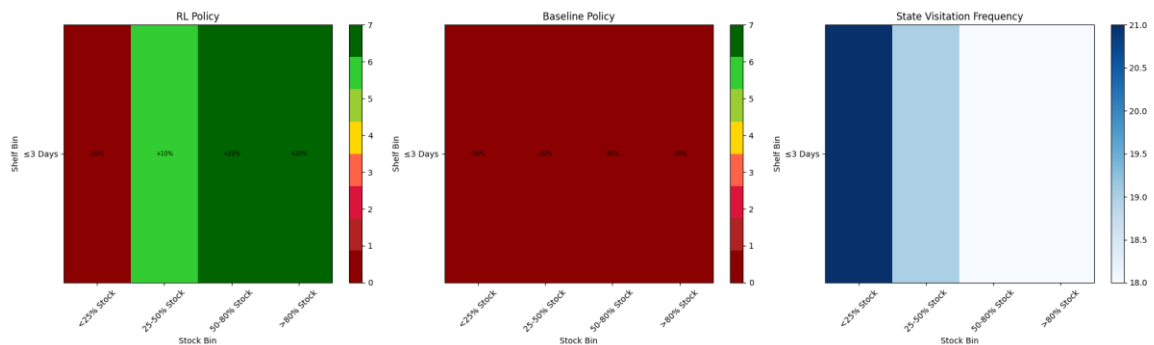
$$profit\ improvement\ pct = \frac{2 \cdot (RL_{profit} - Heur_{profit})}{|RL_{profit}| + |Heur_{profit}| + 10^{-6}} \cdot 100$$

5.5.1. Οπτικοποίηση αποτελεσμάτων

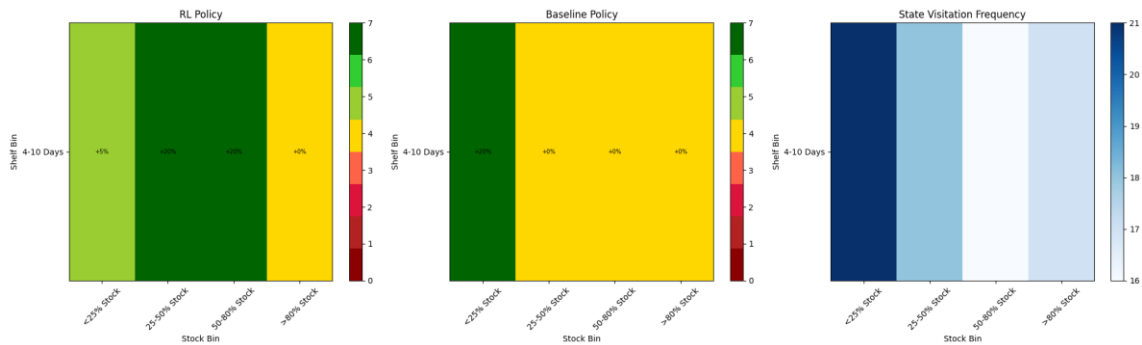
Για την ερμηνεία της πολιτικής του πράκτορα δημιουργήθηκαν οπτικοποιήσεις που αποτυπώνουν τις αποφάσεις τιμολόγησης και τη συχνότητα επίσκεψης καταστάσεων. Ειδικότερα, για κάθε προϊόν σε κάθε seed δημιουργήθηκαν χάρτες θερμότητας που αναδεικνύουν τις επιλογές του πράκτορα σε σχέση με το baseline όσον αφορά την πολιτική των εκπτώσεων σε συνάρτηση με το απόθεμα και τη διάρκεια ζωής καθώς και την συχνότητα επίσκεψης του πράκτορα σε κάθε κατάσταση. Ενδεικτικά οι παρακάτω εικόνες (Εικόνες 5.5 , 5.6 , 5.7) αποτυπώνουν τους συγκεκριμένους θερμικούς χάρτες σε δύο διαφορετικά seeds.



Εικόνα 5.5 Heatmap P002_seed_1

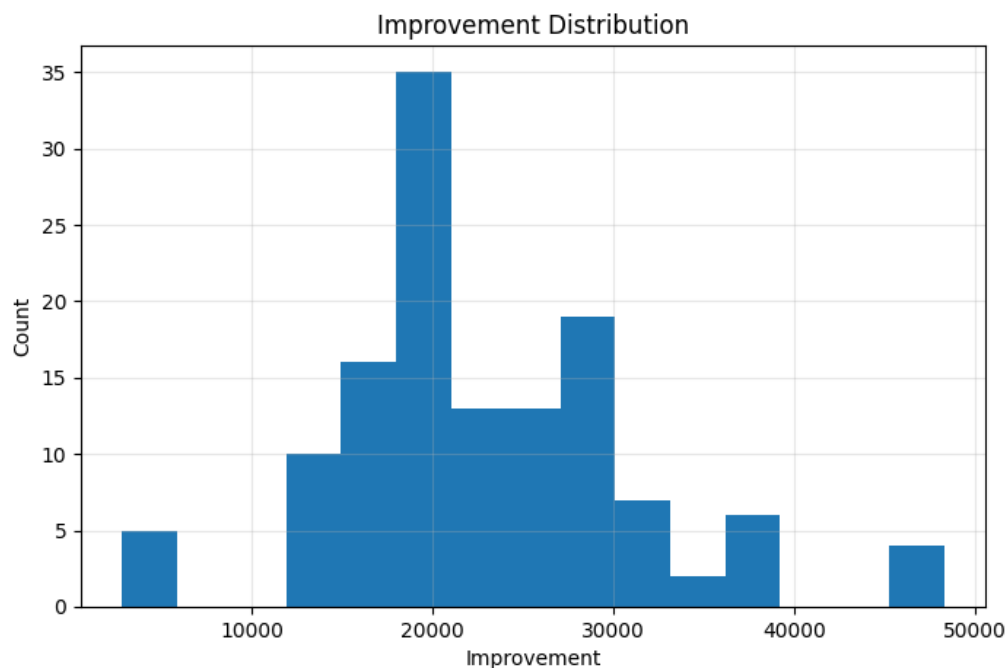


Εικόνα 5.6 Heatmap P016_seed_7



Εικόνα 5.7 Heatmap P117_seed_7

Επίσης το παρακάτω ιστόγραμμα (Εικόνα 5.8) παρουσιάζει την κατανομή της απόλυτης βελτίωσης (RL ανταμοιβή – Baseline ανταμοιβή) για όλα τα SKU και όλα τα seeds.

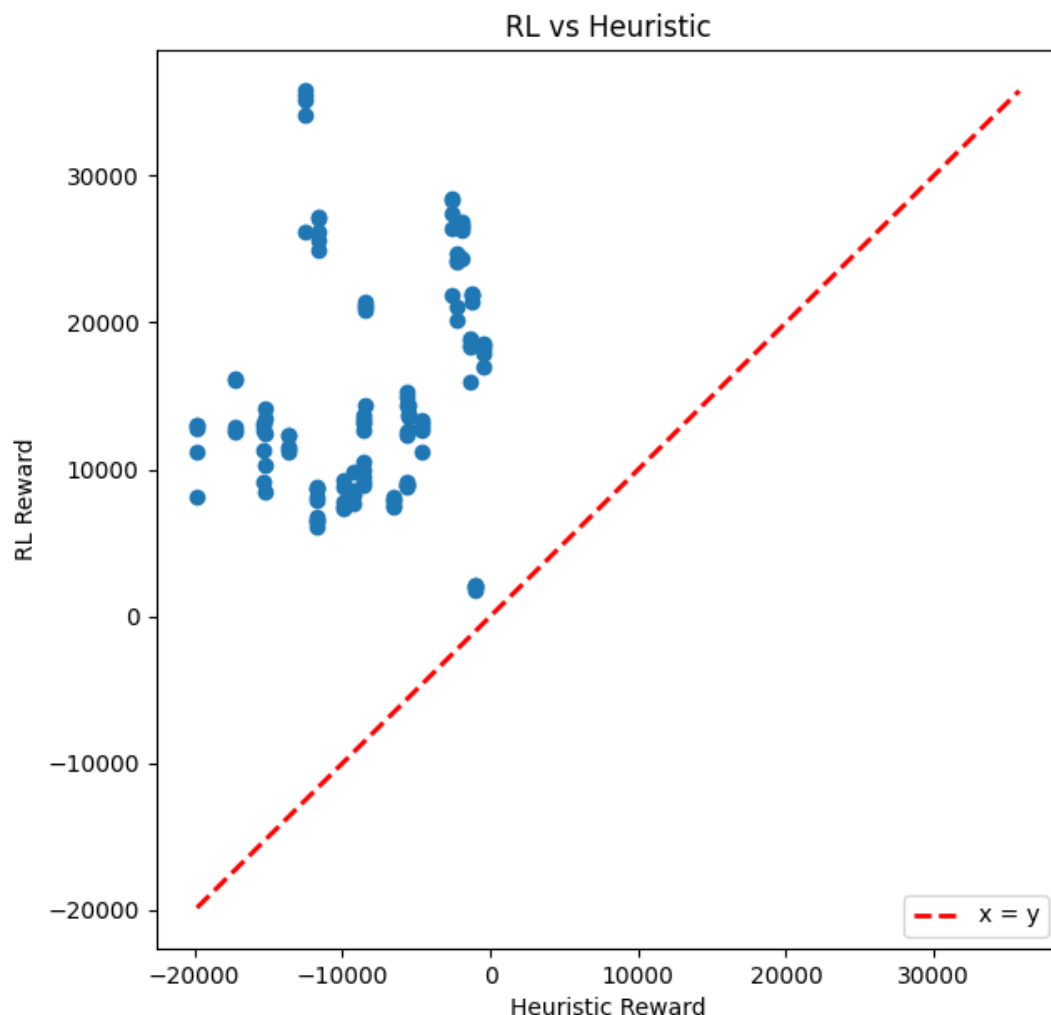


Εικόνα 5.8 Improvement distribution

Παρατηρείται ότι το μεγαλύτερο μέρος των παρατηρήσεων συγκεντρώνεται σε θετικές τιμές βελτίωσης, γεγονός που υποδηλώνει ότι το RL παράγαγε υψηλότερες ανταμοιβές στις περισσότερες περιπτώσεις. Παράλληλα, η παρουσία δεξιάς ουράς στην κατανομή δείχνει

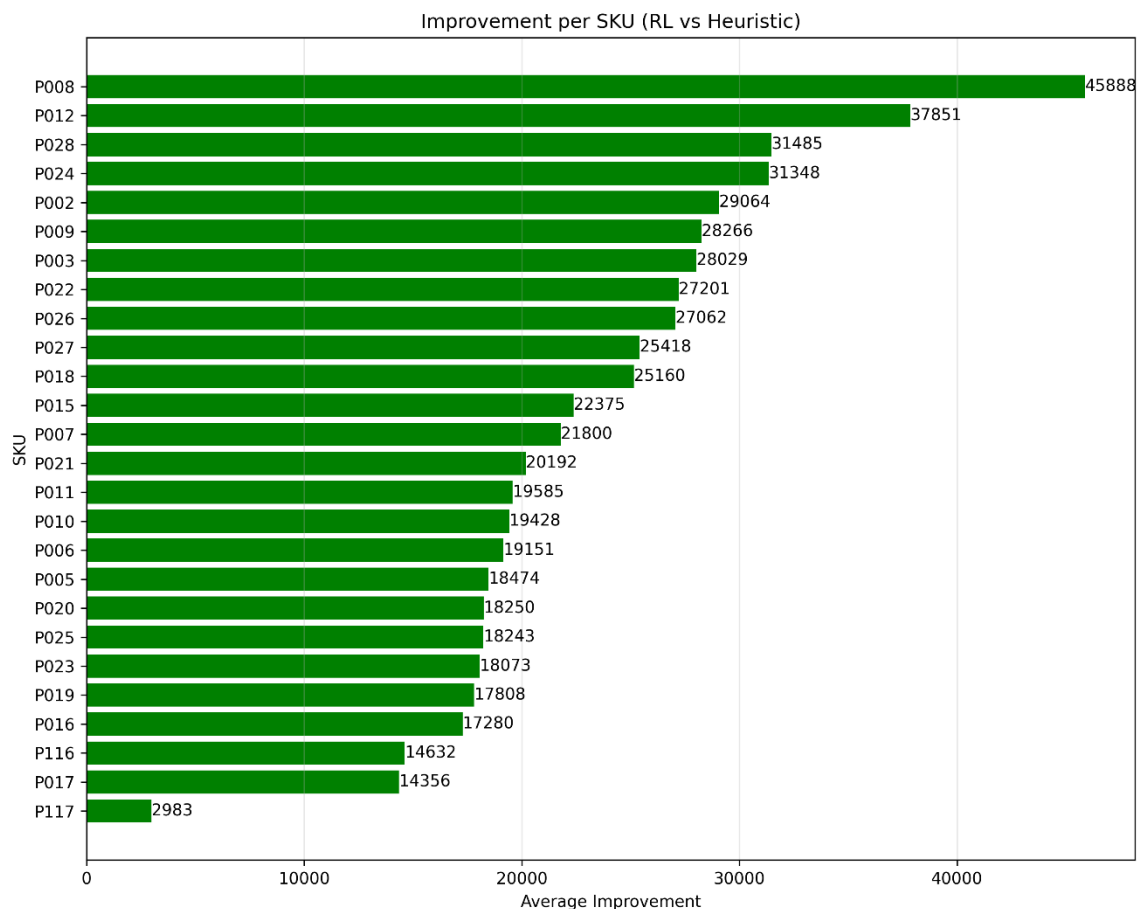
ότι ορισμένα προϊόντα παρουσίασαν ιδιαίτερα σημαντικά οφέλη από την εφαρμογή της προτεινόμενης μεθόδου.

Επιπλέον, το παρακάτω σχήμα (Εικόνα 5.9) συγκρίνει τις συνολικές ανταμοιβές της πολιτικής RL με τις αντίστοιχες ανταμοιβές της heuristic πολιτικής για κάθε seed. Η διακεκομμένη γραμμή αντιπροσωπεύει την περίπτωση ισοδύναμης απόδοσης των δύο μεθόδων. Παρατηρείται ότι το σύνολο των σημείων βρίσκεται πάνω από τη γραμμή αναφοράς, γεγονός που υποδηλώνει συστηματική υπεροχή της προσέγγισης του RL. Διακρίνονται επίσης μερικά ακραία σημεία, το (35000,-12000) και το (2000,-1000), πρόκειται για τα προϊόντα P008(Beef Mince pack) και P117 (Muffin pack). Το P008 είναι το προϊόν με το μεγαλύτερο μεικτό κέρδος και για αυτό αποφέρει και τα μεγαλύτερα κέρδη όταν αξιοποιείται σωστά η πολιτική τιμολόγησης από τον πράκτορα με αποτέλεσμα να έχει μεγάλη ανταμοιβή. Όσον αφορά το προϊόν P117 είναι ένα προϊόν με μεγάλο σχετικά κόστος απόρριψης και μικρό κέρδος, για αυτό και συγκεντρώνει τη μικρότερη ανταμοιβή.



Εικόνα 5.9 Products reward RL vs Heuristic

Επίσης, παρήχθη ένα οριζόντιο ιστόγραμμα (Εικόνα 5.10) για κάθε προϊόν ξεχωριστά με τη μέση βελτίωση. Παρατηρείται ότι η επίδραση της RL πολιτικής δεν είναι ομοιόμορφη σε όλα τα προϊόντα. Το προϊόν P008 παρουσίασε τη μεγαλύτερη βελτίωση, ενώ ακόμη και τα προϊόντα με τη μικρότερη βελτίωση εμφάνισαν θετικό αποτέλεσμα.

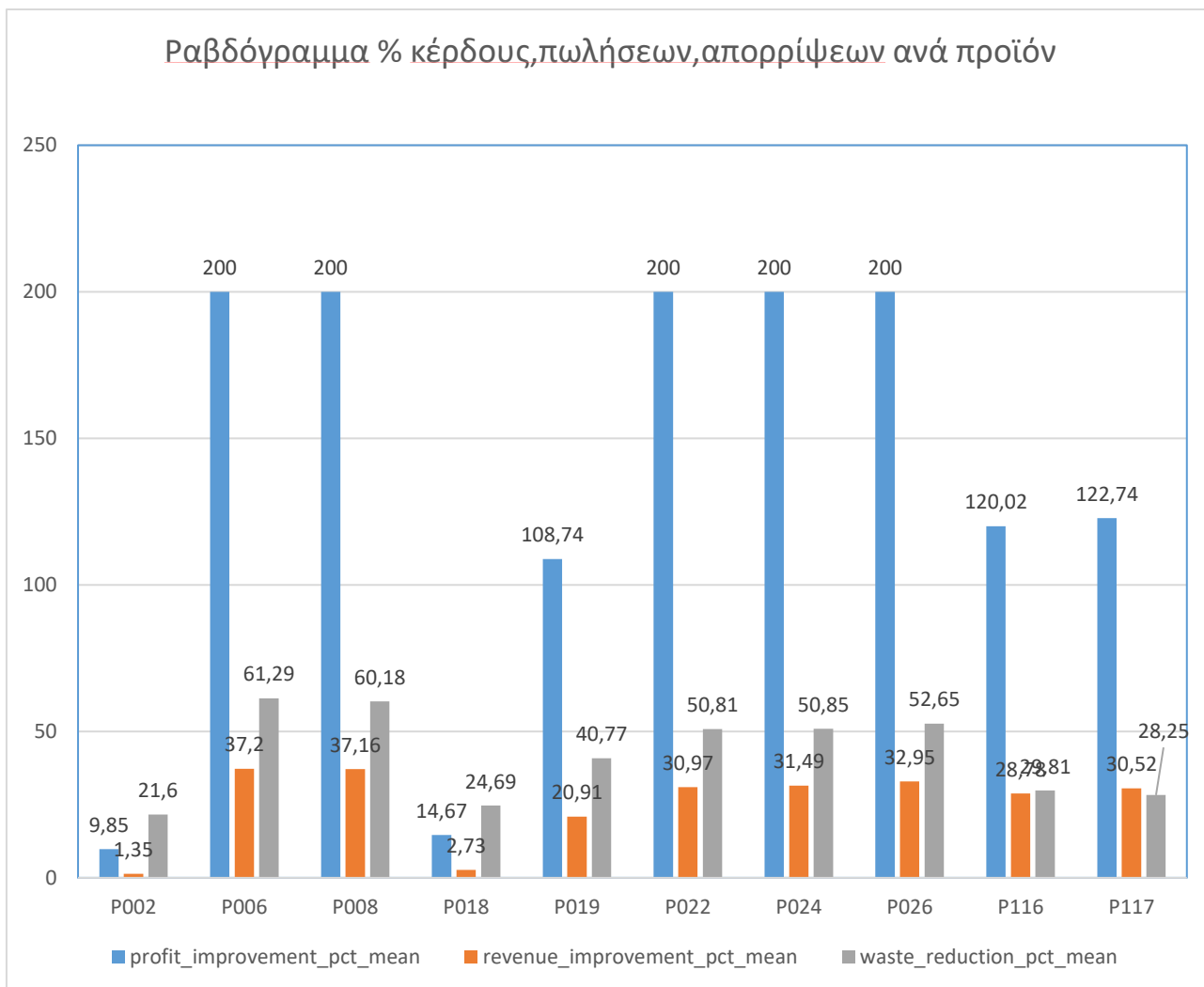


Εικόνα 5.10 Improvement per SKU RL vs Heuristic

Τα προϊόντα που παρουσίασαν τη μεγαλύτερη βελτίωση μέσω της προσέγγισης του RL χαρακτηρίζονται από υψηλή εμπορική αξία, υψηλή κερδοφορία, περιορισμένη διάρκεια ζωής και αυξημένη μεταβλητότητα ζήτησης. Ιδιαίτερα τα προϊόντα της κατηγορίας κρέατος (P008, P012 και P028) εμφανίζουν υψηλό κόστος απόρριψης, γεγονός που δημιουργεί σημαντικές ευκαιρίες για βελτιστοποίηση μέσω δυναμικών αποφάσεων τιμολόγησης. Τα ευρήματα υποδηλώνουν ότι τα μεγαλύτερα οφέλη της Reinforcement Learning προκύπτουν σε προϊόντα υψηλής αξίας και αυξημένου κινδύνου απορρίψεων.

Αντίθετα, τα προϊόντα που παρουσίασαν τη μικρότερη βελτίωση μέσω της πολιτικής Reinforcement Learning χαρακτηρίζονται από μεγαλύτερη διάρκεια ζωής (4–5 ημέρες) και χαμηλότερα ποσοστά αλλοίωσης (περίπου 4–5%). Η μειωμένη επιχειρησιακή πίεση για άμεση εκκαθάριση αποθεμάτων περιορίζει τα περιθώρια βελτιστοποίησης μέσω δυναμικής τιμολόγησης.

Τέλος, στην παρακάτω εικόνα (Εικόνα 5.11) φαίνεται ένα ραβδόγραμμα για 10 ενδεικτικά προϊόντα όπου συγκρίνεται σε όρους ποσοστιαίας συμμετρικής μεταβολής το κέρδος, οι πωλήσεις και οι απορρίψεις ανά προϊόν μεταξύ του πράκτορα και της ευρετικής μεθόδου.



Εικόνα 5.11 Ραβδόγραμμα RL vs Baseline σε % κέρδους, πωλήσεων, απορρίψεων ανά προϊόν

6. Συμπεράσματα

Η παρούσα εργασία διερεύνησε τη δυνατότητα μείωσης της σπατάλης τροφίμων στο λιανικό εμπόριο μέσω της εφαρμογής δυναμικής τιμολόγησης σε συνδυασμό με ABC διαχείριση αποθεμάτων και τεχνικές Reinforcement Learning. Για τον σκοπό αυτό αναπτύχθηκε ένα ολοκληρωμένο πλαίσιο που συνδύασε πρόβλεψη ζήτησης, πολυκριτηριακή ABC ανάλυση και εκπαίδευση ενός πράκτορα Q-Learning για τη λήψη αποφάσεων τιμολόγησης σε περιβάλλον ευπαθών προϊόντων.

Τα αποτελέσματα υποδεικνύουν σαφή υπεροχή της πολιτικής Reinforcement Learning έναντι της heuristic baseline ως προς την κερδοφορία, τις πωλήσεις και τη μείωση των απορρίψεων. Συγκεκριμένα, τα κέρδη αυξήθηκαν κατά 112,20%, οι πωλήσεις κατά 20,63%, ενώ οι απώλειες λόγω απορρίψεων μειώθηκαν κατά 40,69%, επιβεβαιώνοντας την αποτελεσματικότητα της προτεινόμενης προσέγγισης. Επιπλέον, η ανάλυση ανά προϊόν έδειξε ότι τα μεγαλύτερα οφέλη προέκυψαν σε προϊόντα υψηλής εμπορικής αξίας και αυξημένου κινδύνου αλλοίωσης, γεγονός που υποδηλώνει ότι η αποτελεσματικότητα της δυναμικής τιμολόγησης επηρεάζεται σημαντικά από τα λειτουργικά χαρακτηριστικά των προϊόντων.

Παράλληλα, παρατηρήθηκε αύξηση των χαμένων πωλήσεων κατά 26,74%. Το εύρημα αυτό δεν αποτελεί απλώς αδυναμία του μοντέλου, αλλά αντανακλά έναν σημαντικό συμβιβασμό (trade off) μεταξύ κερδοφορίας και διαθεσιμότητας προϊόντων. Η πολιτική Reinforcement Learning επιδιώκει τη μεγιστοποίηση της συνολικής ανταμοιβής μέσω επιθετικότερων στρατηγικών τιμολόγησης που αυξάνουν τις πωλήσεις και μειώνουν τις απορρίψεις.

Ωστόσο, το σύστημα λειτουργεί υπό σταθερή πολιτική αναπλήρωσης αποθέματος, όπου η ανανέωση πραγματοποιείται ανά επτά ημέρες και ο πράκτορας δεν έχει τη δυνατότητα να παραγγείλει πρόσθετο απόθεμα όταν αυτό εξαντλείται. Ως αποτέλεσμα, η αυξημένη ζήτηση που δημιουργείται από ορισμένες τιμολογιακές αποφάσεις οδηγεί συχνότερα σε καταστάσεις έλλειψης αποθέματος και συνεπώς σε αυξημένες χαμένες πωλήσεις.

Επομένως, τα αποτελέσματα αναδεικνύουν ότι η βελτιστοποίηση της τιμολόγησης από μόνη της μπορεί να αυξήσει σημαντικά την οικονομική απόδοση, αλλά ταυτόχρονα αποκαλύπτει τους περιορισμούς που επιβάλλει μια στατική πολιτική αναπλήρωσης αποθέματος. Το εύρημα αυτό βρίσκεται σε συμφωνία με τη διεθνή βιβλιογραφία, η οποία υποστηρίζει ότι οι τεχνικές δυναμικής τιμολόγησης και Reinforcement Learning μπορούν να βελτιώσουν την κερδοφορία και να περιορίσουν τις απώλειες λόγω αλλοίωσης σε προϊόντα μικρής διάρκειας ζωής. Παράλληλα, η παρούσα εργασία αναδεικνύει ότι η βελτιστοποίηση της τιμολόγησης δεν αρκεί από μόνη της για τη συνολική βελτιστοποίηση του συστήματος, καθώς οι αποφάσεις αναπλήρωσης αποθέματος επηρεάζουν άμεσα την τελική απόδοση.

Τέλος, οι τιμές των τυπικών αποκλίσεων και των διαστημάτων εμπιστοσύνης κρίνονται ικανοποιητικές, γεγονός που υποδηλώνει σταθερότητα των αποτελεσμάτων μεταξύ των διαφορετικών εκτελέσεων και ενισχύει την αξιοπιστία των συμπερασμάτων της μελέτης.

6.1. Περιορισμοί της έρευνας

Παρά τα θετικά αποτελέσματα, η παρούσα εργασία παρουσιάζει ορισμένους περιορισμούς που πρέπει να ληφθούν υπόψη κατά την ερμηνεία των ευρημάτων.

Ο σημαντικότερος περιορισμός αφορά τη χρήση συνθετικών δεδομένων, τα οποία δημιουργήθηκαν με στόχο την προσομοίωση ρεαλιστικών συνθηκών λειτουργίας ενός περιβάλλοντος λιανικής πώλησης τροφίμων. Παρότι τα δεδομένα σχεδιάστηκαν ώστε να αναπαράγουν βασικά χαρακτηριστικά της πραγματικής αγοράς, δεν μπορούν να αντικαταστήσουν πλήρως την πολυπλοκότητα και την αβεβαιότητα πραγματικών επιχειρησιακών δεδομένων.

Επιπλέον, η πολιτική αναπλήρωσης αποθεμάτων θεωρήθηκε σταθερή και δεν βελτιστοποιήθηκε από τον πράκτορα Reinforcement Learning. Ως αποτέλεσμα, το μοντέλο δεν μπορούσε να αντιδράσει δυναμικά σε περιπτώσεις αυξημένης ζήτησης μέσω πρόσθετων παραγγελιών ή τροποποίησης των επιπέδων αποθέματος.

Τέλος, η αξιολόγηση πραγματοποιήθηκε με χρήση ενός συγκεκριμένου περιβάλλοντος προσομοίωσης και μίας συγκεκριμένης heuristic πολιτικής αναφοράς. Εναλλακτικά θα μπορούσαν να χρησιμοποιηθούν διαφορετικές baseline πολιτικές ή διαφορετικοί αλγόριθμοι ενισχυτικής μάθησης.

6.2. Μελλοντικές προεκτάσεις

Μία φυσική συνέχεια της παρούσας έρευνας είναι η ανάπτυξη ολοκληρωμένων μοντέλων που θα βελτιστοποιούν ταυτόχρονα τόσο τη δυναμική τιμολόγηση όσο και την πολιτική αναπλήρωσης αποθεμάτων. Η συνδυασμένη αντιμετώπιση των δύο προβλημάτων αναμένεται να περιορίσει τις χαμένες πωλήσεις που παρατηρήθηκαν στην παρούσα εργασία και να βελτιώσει περαιτέρω τη συνολική απόδοση του συστήματος.

Επιπλέον, θα μπορούσε να διερευνηθεί η χρήση πιο προηγμένων αλγορίθμων Reinforcement Learning, όπως Deep Q-Networks (DQN), Proximal Policy Optimization (PPO) και Actor-Critic προσεγγίσεις, οι οποίοι έχουν τη δυνατότητα να διαχειρίζονται μεγαλύτερους χώρους καταστάσεων και πιο σύνθετα περιβάλλοντα λήψης αποφάσεων.

Τέλος, ιδιαίτερο ενδιαφέρον παρουσιάζει η εφαρμογή της προτεινόμενης μεθοδολογίας σε πραγματικά δεδομένα λιανικής πώλησης, καθώς και η αξιολόγηση της σε διαφορετικές κατηγορίες προϊόντων, διαφορετικές πολιτικές αναπλήρωσης και εναλλακτικά επιχειρησιακά περιβάλλοντα.

7. Βιβλιογραφία

- Akkerman, R., Farahani, P. & Grunow, M. (2010). Quality, safety and sustainability in food distribution: A review of quantitative operations management approaches and challenges. *OR Spectrum*, 32(4), 863–904. <https://doi.org/10.1007/S00291-010-0223-2>
- Ameet Talwalkar Mehryar Mohri, A. R. (2018). Foundation of Machine Learning 2nd. *The MIT Press*, 4(1), 88–100.
https://www.hlevkin.com/hlevkin/45MachineDeepLearning/ML/Foundations_of_Machine_Learning.pdf
- Awad, M. & Khanna, R. (2015). Machine Learning. *Efficient Learning Machines*, 1–18.
https://doi.org/10.1007/978-1-4302-5990-9_1

- Bellemare, M. G., Dabney, W. & Munos, R. (2017). A Distributional Perspective on Reinforcement Learning. *34th International Conference on Machine Learning, ICML 2017*, 1, 693–711. <https://doi.org/10.48550/arXiv.1707.06887>
- Berberich, D. (2020). *Hybrid Methods for Time Series Forecasting*.
<https://doi.org/10.5445/IR/1000134512>
- Blackburn, J. & Scudder, G. (2009). Supply chain strategies for perishable products: The case of fresh produce. *Production and Operations Management*, 18(2), 129–137.
<https://doi.org/10.1111/J.1937-5956.2009.01016.X>
- Bolton, L. E., Warlop, L. & Alba, J. W. (2002). Consumer perceptions of price (un)fairness. *Journal of Consumer Research*, 29(4), 474–491.
<https://doi.org/10.1086/346244>
- Box, G. E. P., Jenkins, G. M., Reinsel, G. C. & Ljung, G. M. (2016). Journal of time series analysis book review time series analysis: Forecasting and control, 5th edition. *J. Time. Ser. Anal*, 37, 709–711. <https://www.wiley.com/en-us/Time+Series+Analysis%3A+Forecasting+and+Control%2C+5th+Edition-p-9781118675021>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.
<https://doi.org/10.1023/A:1010933404324/METRICS>
- Buisman, M. E., Haijema, R. & Bloemhof-Ruwaard, J. M. (2019). Discounting and dynamic shelf life to reduce fresh food waste at retailers. *International Journal of Production Economics*, 209, 274–284. <https://doi.org/10.1016/j.ijpe.2017.07.016>
- Calvano, E., Calzolari, G., Denicolò, V. & Pastorello, S. (2020). Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review*, 110(10), 3267–3297.
<https://doi.org/10.1257/aer.20190623>

- Chen, W., Liu, H. & Xu, D. (2018). Dynamic Pricing Strategies for Perishable Product in a Competitive Multi-Agent Retailers Market. *2017:4:3, 21(2)*.
<https://doi.org/10.18564/JASSS.3710>
- Chen, Y. & Zhang, Z. J. (2009). Dynamic targeted pricing with strategic consumers. *International Journal of Industrial Organization*, 27(1), 43–50.
<https://doi.org/10.1016/J.IJINDORG.2008.03.002>
- Chen, Z. & Rey, P. (2012). Loss Leading as an Exploitative Practice. *American Economic Review*, 102(7), 3462–3482. <https://doi.org/10.1257/aer.102.7.3462>
- Chu, C. W., Liang, G. S. & Liao, C. T. (2008). Controlling inventory by combining ABC analysis and fuzzy classification. *Computers and Industrial Engineering*, 55(4), 841–851. <https://doi.org/10.1016/j.cie.2008.03.006>
- Craft, R. C. & Leake, C. (2002). The Pareto principle in organizational decision making. *Management Decision*, 40(8), 729–733. <https://doi.org/10.1108/00251740210437699>
- Dabney, W., Rowland, M., Bellemare, M. G. & Munos, R. (2017). Distributional Reinforcement Learning with Quantile Regression. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 2892–2901.
<https://doi.org/10.1609/aaai.v32i1.11791>
- FAO. (2019). The State of Food and Agriculture 2019. *The State of Food and Agriculture 2019*. <https://doi.org/10.4060/CA6030EN>
- Flores, B. E. & Whybark, D. C. (1987). Implementing multiple criteria ABC analysis. *Journal of Operations Management*, 7(1–2), 79–85. [https://doi.org/10.1016/0272-6963\(87\)90008-8](https://doi.org/10.1016/0272-6963(87)90008-8)

- From farm to fork - Consilium*. (2020). <https://www.consilium.europa.eu/en/policies/from-farm-to-fork/>
- Gallego, G. & van Ryzin, G. (1994). Optimal Dynamic Pricing of Inventories with Stochastic Demand over Finite Horizons. *Https://Doi.Org/10.1287/Mnsc.40.8.999*, 40(8), 999–1020. <https://doi.org/10.1287/MNSC.40.8.999>
- Goodfellow Ian, Bengio Yoshua & Courville Aaron. (2016). Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning. In MIT Press (Ed.), *The MIT Press*. <https://www.deeplearningbook.org/>
- Han, J., Kamber, M., Techniques, J. P.-, Morgan, W. & 2012, undefined. (2012). Data mining: Concepts and. *Homes.Di.Unimi.It*. <http://homes.di.unimi.it/ceselli/IM/2012-13/slides/02-KnowYourData.pdf>
- Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D. & Meger, D. (2018). Deep Reinforcement Learning That Matters. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 3207–3214. <https://doi.org/10.1609/AAAI.V32I1.11694>
- Hendrix, E. M. T., Pauls-Worm, K. G. J. & de Jong, M. V. (2023). On Order Policies for a Perishable Product in Retail. *Informatica (Netherlands)*, 34(2), 271–283. <https://doi.org/10.15388/23-INFOR520>
- Hochreiter, S. & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/NECO.1997.9.8.1735>
- Holt, C. C. (2004). Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, 20(1), 5–10. <https://doi.org/10.1016/J.IJFORECAST.2003.09.015>

Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting Principles and Practice* (3rd ed.). OTexts: Melbourne, Australia. OTexts.com/fpp3. Accessed on April 12, 2026.

(n.d.). Retrieved April 12, 2026, from <https://otexts.com/fpp3/>

Hyndman, R. J. . & Athanasopoulos, George. (2021). Time series cross-validation.

Forecasting: Principles and Practice, 440.

<https://books.google.com/books/about/Forecasting.html?hl=en&id=gZB-zgEACAAJ>

James, G., Witten, D., Hastie, T. & Tibshirani, R. (2021). *An Introduction to Statistical Learning*. <https://doi.org/10.1007/978-1-0716-1418-1>

Janiesch, C., Zschech, P. & Heinrich, K. (2021). Machine learning and deep learning.

Electronic Markets, 31(3), 685–695. <https://doi.org/10.1007/s12525-021-00475-2>

Jayakumaran, S., Shan, W. Z. & Daud, D. (2020). ABC Analysis: A Qualitative Case Study on Inventory Management in Giant Superstore Taman Connaught, An Outlet Of GCH Retail (Malaysia) SDN. BHD. *IOP Conference Series: Materials Science and Engineering*, 780(7), 072016. <https://doi.org/10.1088/1757-899X/780/7/072016>

Jullien, S., Ariannezhad, M., Groth, P. & de Rijke, M. (2023). *A Simulation Environment and Reinforcement Learning Method for Waste Reduction*.

<https://doi.org/10.48550/arXiv.2205.15455>

Kaelbling, L. P., Littman, M. L. & Moore, A. W. (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, 4, 237–285.

<https://doi.org/10.1613/JAIR.301>

Kara, A. & Dogan, I. (2018). Reinforcement learning approaches for specifying ordering policies of perishable inventory systems. *Expert Systems with Applications*, 91, 150–158. <https://doi.org/10.1016/J.ESWA.2017.08.046>

- Kasar, Aditya, Tripathi, Mukund, Indolia, Asha & Rawat. (2025). Data-Driven Decision Making for Perishable Food Supply Chains: Insights from Demand Forecasting Models. *International Journal of Supply Chain Management*.
<https://doi.org/10.13140/RG.2.2.12142.93761>
- Kayikci, Y., Demir, S., Mangla, S. K., Subramanian, N. & Koc, B. (2022). Data-driven optimal dynamic pricing strategy for reducing perishable food waste at retailers. *Journal of Cleaner Production*, 344. <https://doi.org/10.1016/j.jclepro.2022.131068>
- Khanorkar, Y. & Kane, P. V. (2023). Selective inventory classification using ABC classification, multi-criteria decision making techniques, and machine learning techniques. *Materials Today: Proceedings*, 72, 1270–1274.
<https://doi.org/10.1016/J.MATPR.2022.09.298>
- Konda, V. R. & Tsitsiklis, J. N. (1999). *Actor-Critic Algorithms*.
https://proceedings.neurips.cc/paper_files/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf
- Kothapalli, V., Ranjan, R., Hudovernik, V., Dwivedi, V. P., Hoffart, J., Guestrin, C. & Leskovec, J. (2026). *PluRel: Synthetic Data unlocks Scaling Laws for Relational Foundation Models*. <http://arxiv.org/abs/2602.04029>
- Mallidis, I., Sariannidis, N., Vlachos, D., Yakavenka, V., Aifadopoulou, G. & Zopounidis, K. (2022). Optimal inventory control policies for avoiding food waste. *Operational Research*, 22(1), 685–701. <https://doi.org/10.1007/s12351-020-00554-w>
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533. <https://doi.org/10.1038/NATURE14236>

- Nahmias, S. (1982). PERISHABLE INVENTORY THEORY: A REVIEW. *Operations Research*, 30(4), 680–708. <https://doi.org/10.1287/OPRE.30.4.680>
- Nomura, Y., Liu, Z. & Nishi, T. (2025). Deep Reinforcement Learning for Dynamic Pricing and Ordering Policies in Perishable Inventory Management. *Applied Sciences (Switzerland)*, 15(5). <https://doi.org/10.3390/app15052421>
- Phillips, R. (2005). *Pricing and Revenue Optimization*. Stanford University Press. <https://doi.org/10.1515/9780804781640>
- Pradhan, T., L, M. K., Chandrakantrao Bhaganagare, R. & Kumar, S. (2025). Dynamic Pricing 2.0: How AI Is Revolutionizing Real-Time Pricing Strategies. In *International Journal of Environmental Sciences* (Vol. 11, Issue 14). <https://doi.org/10.64252/ffbrmg88>
- Puterman, M. L. . (2014). *Markov decision processes : discrete stochastic dynamic programming*. https://books.google.com/books/about/Markov_Decision_Processes.html?id=VvBjBAAAQBAJ
- Ramanathan, R. (2006). ABC inventory classification with multiple-criteria using weighted linear optimization. *Computers and Operations Research*, 33(3), 695–700. <https://doi.org/10.1016/J.COR.2004.07.014>
- Riesenegger, L. & Hübner, A. (2022). Reducing Food Waste at Retail Stores—An Explorative Study. *Sustainability (Switzerland)*, 14(5). <https://doi.org/10.3390/su14052494>
- Rob J Hyndman and George Athanasopoulos. (2021). *Forecasting: Principles and Practice (3rd ed)*. <https://otexts.com/fpp3/>

- Rummery, G. A. and N. M. & Niranjana, M. (1994). *Rummery, G.A. and Niranjana, M. (1994) On-Line Q-Learning Using Connectionist Systems. Department of Engineering, University of Cambridge, Cambridge. - References - Scientific Research Publishing.* https://www.researchgate.net/profile/Mahesan-Niranjana/publication/2500611_On-Line_Q-Learning_Using_Connectionist_Systems/links/5438d5db0cf204cab1d6db0f/On-Line-Q-Learning-Using-Connectionist-Systems.pdf
- Sanders, R. E. (2024). Dynamic Pricing and Organic Waste Bans: A Study of Grocery Retailers' Incentives to Reduce Food Waste. *Marketing Science*, 43(2), 289–316. <https://doi.org/10.1287/mksc.2020.0214>
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. (2017). *Proximal Policy Optimization Algorithms.* <https://doi.org/10.48550/arXiv.1707.06347>
- Silver, E. A., Pyke, D. F. & Thomas, D. J. (2016). Inventory and Production Management in Supply Chains, Fourth Edition. *Inventory and Production Management in Supply Chains, Fourth Edition*, 1–781. <https://doi.org/10.1201/9781315374406>
- Sutton, R. S. . & Barto, A. G. . (2018). *Reinforcement learning : an introduction.* 526. <https://www.andrew.cmu.edu/course/10-703/textbook/BartoSutton.pdf>
- Sutton, R. S., McAllester, D., Singh, S. & Mansour, Y. (1999). Policy Gradient Methods for Reinforcement Learning with Function Approximation. *Advances in Neural Information Processing Systems*, 12. https://papers.nips.cc/paper_files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf
- Talluri, K. T. & Van Ryzin, G. J. (2004). *The Theory and Practice of Revenue Management.* 68. <https://doi.org/10.1007/B139000>

- Tesauro, G. & Kephart, J. O. (2002). Pricing in Agent Economies Using Multi-Agent Q-Learning. *Autonomous Agents and Multi-Agent Systems*, 5(3), 289–304.
<https://doi.org/10.1023/A:1015504423309>
- Transforming our world: the 2030 Agenda for Sustainable Development* | Department of Economic and Social Affairs. (n.d.). Retrieved May 16, 2026, from
<https://sdgs.un.org/2030agenda>
- Triebe, O., Hewamalage, H., Pilyugina, P., Laptev, N., Bergmeir, C. & Rajagopal, R. (2021). *NeuralProphet: Explainable Forecasting at Scale*.
<https://arxiv.org/pdf/2111.15397>
- Vapnik, V. N. (1995). The Nature of Statistical Learning Theory. *The Nature of Statistical Learning Theory*. <https://doi.org/10.1007/978-1-4757-2440-0>
- Wang, C. & Ning, G. (2026). Multi-criteria inventory classification considering demand stability. *Scientific Reports 2026 16:1*, 16(1), 10664-. <https://doi.org/10.1038/s41598-026-42590-0>
- Watkins, C. (1989). *Learning from delayed rewards*.
https://www.academia.edu/download/50360235/Learning_from_delayed_rewards_20161116-28282-v2pwvq.pdf
- Watkins, C. J. C. H. & Dayan, P. (1992). Q-learning. *Machine Learning 1992 8:3*, 8(3), 279–292. <https://doi.org/10.1007/BF00992698>
- Winters, P. R. (1960). Forecasting Sales by Exponentially Weighted Moving Averages. *https://doi.org/10.1287/Mnsc.6.3.324*, 6(3), 324–342.
<https://doi.org/10.1287/MNSC.6.3.324>

Yang, C., Feng, Y. & Whinston, A. (2022). Dynamic Pricing and Information Disclosure for Fresh Produce: An Artificial Intelligence Approach. *Production and Operations Management*, 31(1), 155–171. <https://doi.org/10.1111/POMS.13525>

8. Παράρτημα Α: Ο κώδικας της εργασίας και τα αποτελέσματα

Σε αυτή την ενότητα θα παρατεθεί ο κώδικας που αναπτύχθηκε στη γλώσσα Python για την περάτωση της παρούσας εργασίας αλλά και τα τελικά αποτελέσματα σε μορφή στιγμιότυπων.

8.1. Ο κώδικας για την παραγωγή των δεδομένων της εργασίας

```
# make_supermarket_data.py

import os
import numpy as np
import pandas as pd
from datetime import date, datetime, timedelta
from dateutil.easter import easter # Ορθόδοξο Πάσχα

# ΠΥΘΜΙΣΕΙΣ

SEED = 7
STORE_ID = "S01"

START_DATE = date(2024, 1, 1)
END_DATE = date(2025, 12, 31) # 2 χρόνια
OUT_DIR = "synthetic_out"

N_SKU_TOTAL = 130

# Κατανομή SKUs ανά κατηγορία (σύνολο = 130)
```

```
SKU_SPLIT = {  
    "Meat": 18,  
    "Fish": 10,  
    "Dairy": 18,  
    "Groceries": 28,  
    "Fruits": 12,  
    "Vegetables": 24,  
    "Bakery": 10,  
    "Deli": 10,  
}  
assert sum(SKU_SPLIT.values()) == N_SKU_TOTAL
```

Baseline ζήτηση/ημέρα ανά κατηγορία (τεμάχια)

```
BASE_DEMAND_RANGES = {  
    "Meat": (6, 28),  
    "Fish": (4, 18),  
    "Dairy": (10, 45),  
    "Groceries": (3, 22),  
    "Fruits": (12, 55),  
    "Vegetables": (10, 50),  
    "Bakery": (8, 40),  
    "Deli": (6, 30),  
}
```

Shelf life (ημέρες) ανά κατηγορία

```
SHELF_LIFE_RANGES = {  
    "Meat": (2, 5),  
    "Fish": (1, 3),  
    "Dairy": (5, 14),  
    "Groceries": (120, 365),  
    "Fruits": (3, 9),
```

```
"Vegetables": (3, 10),  
"Bakery": (2, 5),  
"Deli": (4, 10),  
}
```

Εύρος τιμής ανά κατηγορία (€) (fallback αν δεν βρεθεί subcat range)

```
PRICE_RANGES = {  
    "Meat": (4.0, 12.0),  
    "Fish": (5.0, 14.0),  
    "Dairy": (0.9, 5.5),  
    "Groceries": (0.8, 8.0),  
    "Fruits": (0.8, 4.5),  
    "Vegetables": (0.8, 4.0),  
    "Bakery": (0.6, 4.0),  
    "Deli": (1.8, 7.5),  
}
```

Ρεαλιστικό εύρος τιμών ανά sub_category (ενδεικτικά)

Οι μονάδες είναι "λογικές" ανά μονάδα (kg/pack/L/pcs) με μικρό πολλαπλασιαστή
παρακάτω.

```
SUBCAT_PRICE = {  
    # MEAT  
    "Turkey Fillet": (8.0, 13.0),  
    "Chicken Breast": (6.0, 11.0),  
    "Beef Mince": (8.0, 14.0),  
    "Pork Chops": (6.5, 10.5),  
    "Sausages": (5.0, 9.0),  
  
    # FISH  
    "Salmon": (10.0, 18.0),  
    "Sea Bream": (9.0, 15.0),
```

"Cod": (8.0, 14.0),
"Tuna": (9.0, 16.0),
"Shrimps": (10.0, 20.0),

DAIRY

"Milk": (0.95, 1.90),
"Yogurt": (0.60, 1.60),
"Cheese": (2.20, 6.80),
"Butter": (1.60, 4.20),
"Cream": (1.10, 3.20),

GROCERIES

"Pasta": (0.85, 2.40),
"Rice": (1.20, 4.80),
"Canned Beans": (0.85, 2.20),
"Olive Oil": (5.80, 12.80),
"Cereal": (1.80, 5.90),
"Coffee": (3.20, 10.50),
"Tea": (1.60, 6.20),
"Flour": (0.85, 2.70),

FRUITS

"Bananas": (1.00, 2.30),
"Apples": (1.20, 3.20),
"Oranges": (1.00, 2.90),
"Grapes": (1.90, 4.80),
"Peaches": (1.60, 4.20),
"Strawberries": (3.20, 7.40),

VEGETABLES

"Tomatoes": (1.20, 3.70),

"Cucumbers": (1.00, 2.90),

"Potatoes": (0.60, 1.90),

"Onions": (0.70, 2.10),

"Lettuce": (0.80, 2.30),

"Carrots": (0.80, 2.30),

BAKERY

"Bread": (0.90, 2.60),

"Baguette": (0.80, 2.30),

"Croissant": (0.60, 2.10),

"Muffin": (0.85, 2.70),

"Pita": (0.85, 2.70),

DELI

"Ham Slices": (1.90, 4.80),

"Salami": (2.10, 5.30),

"Feta Pack": (2.30, 6.50),

"Olives": (1.70, 5.20),

"Ready Salad": (1.60, 4.80),

}

Μικρή προσαρμογή ανά μονάδα (ώστε pcs να βγαίνει πιο χαμηλό)

UNIT_PRICE_MULT = {

"kg": 1.00,

"pack": 1.00,

"pcs": 0.60,

"L": 1.00,

}

Μικτό περιθώριο (gross margin) ανά κατηγορία

MARGIN_RANGES = {

```
"Meat": (0.18, 0.32),  
"Fish": (0.18, 0.33),  
"Dairy": (0.15, 0.30),  
"Groceries": (0.10, 0.28),  
"Fruits": (0.18, 0.35),  
"Vegetables": (0.18, 0.35),  
"Bakery": (0.20, 0.40),  
"Deli": (0.18, 0.35),  
}
```

```
# Εβδομαδιακή εποχικότητα
```

```
SAT_UPLIFT = 0.22
```

```
SUN_UPLIFT = 0.16
```

```
# Noise: τυχαία διακύμανση ζήτησης (πολλαπλασιαστικό)
```

```
NOISE_SIGMA = 0.22
```

```
# πιθανότητα promo ανά SKU ανά μέρα (μόνο όταν ανοιχτό)
```

```
PROMO_PROB = 0.025
```

```
# Μεταφορά ζήτησης από κλειστές μέρες προς προηγούμενες ανοιχτές
```

```
SHIFT_WEIGHTS = (0.70, 0.30)
```

```
# Inventory proxy
```

```
TARGET_COVER_DAYS = 7
```

```
INIT_COVER_MIN = 4
```

```
INIT_COVER_MAX = 8
```

```
# Daily spoilage (χάλασμα αποθέματος) ως % του ON_HAND ανά μέρα (τρέχει και όταν  
κλειστό)
```

```
DAILY_SPOILAGE_RATE = {
```

```
"Fish": 0.030,  
"Meat": 0.015,  
"Bakery": 0.020,  
"Fruits": 0.012,  
"Vegetables": 0.010,  
"Dairy": 0.006,  
"Deli": 0.008,  
"Groceries": 0.0002,  
}
```

```
# HELPERS
```

```
def ensure_dir(path: str):  
    os.makedirs(path, exist_ok=True)  
  
def dmy(dt: date) -> str:  
    return dt.strftime("%d/%m/%Y")  
  
def daterange(d1: date, d2: date):  
    cur = d1  
    while cur <= d2:  
        yield cur  
        cur += timedelta(days=1)  
  
def weekly_mult(dt: date) -> float:  
    wd = dt.weekday() # Mon=0 ... Sun=6  
    if wd == 5:  
        return 1.0 + SAT_UPLIFT  
    if wd == 6:  
        return 1.0 + SUN_UPLIFT
```

```
return 1.0
```

```
def year_sine(dt: date) -> float:  
    doy = dt.timetuple().tm_yday  
    return 1.0 + 0.10 * np.sin(2 * np.pi * doy / 365.25)
```

```
def fruit_season_mult(dt: date) -> float:  
    if dt.month in (6, 7, 8):  
        return 1.15  
    if dt.month in (12, 1):  
        return 0.95  
    return 1.00
```

```
def veg_season_mult(dt: date) -> float:  
    if dt.month in (11, 12, 1, 2):  
        return 1.08  
    return 1.00
```

```
def orthodox_easter(year: int) -> date:  
    return easter(year, method=3)
```

```
def black_friday(year: int) -> date:  
    nov1 = date(year, 11, 1)  
    first_thu = nov1 + timedelta(days=(3 - nov1.weekday()) % 7)  
    thanksgiving = first_thu + timedelta(days=21)  
    return thanksgiving + timedelta(days=1)
```

```
# CLOSED DAYS (Αργίες + Κυριακές)
```

```
def holiday_dates_greece(start: date, end: date) -> set[date]:
```

""

Κλειστό όλες τις επίσημες αργίες Ελλάδας + όλες τις Κυριακές.

Fixed: 1/1, 6/1, 25/3, 1/5, 15/8, 28/10, 25/12, 26/12

Movable (Orthodox): Καθαρά Δευτέρα, Μ. Παρασκευή, Κυριακή Πάσχα, Δευτέρα
Πάσχα, Αγ. Πνεύματος

""

closed = set()

```
for y in range(start.year, end.year + 1):
```

```
    fixed = [
```

```
        date(y, 1, 1),
```

```
        date(y, 1, 6),
```

```
        date(y, 3, 25),
```

```
        date(y, 5, 1),
```

```
        date(y, 8, 15),
```

```
        date(y, 10, 28),
```

```
        date(y, 12, 25),
```

```
        date(y, 12, 26),
```

```
    ]
```

```
    for d in fixed:
```

```
        if start <= d <= end:
```

```
            closed.add(d)
```

```
pascha = orthodox_easter(y)
```

```
movable = [
```

```
    pascha - timedelta(days=48), # Clean Monday
```

```
    pascha - timedelta(days=2), # Good Friday
```

```
    pascha, # Easter Sunday
```

```
    pascha + timedelta(days=1), # Easter Monday
```

```
    pascha + timedelta(days=50), # Holy Spirit Monday
```

```
]
```

```
for d in movable:
    if start <= d <= end:
        closed.add(d)

for dt in daterange(start, end):
    if dt.weekday() == 6: # Sundays
        closed.add(dt)

return closed

# CALENDAR EVENTS

def make_calendar(start: date, end: date) -> pd.DataFrame:
    rows = []

    for y in range(start.year, end.year + 1):
        fixed = [
            (date(y, 1, 1), "New Year", "holiday", 1.10),
            (date(y, 1, 6), "Epiphany", "holiday", 1.04),
            (date(y, 2, 14), "Valentine's", "event", 1.03),
            (date(y, 3, 25), "25 March", "holiday", 1.08),
            (date(y, 5, 1), "Labour Day", "holiday", 1.05),
            (date(y, 8, 15), "Assumption 15Aug", "holiday", 1.10),
            (date(y, 10, 28), "28 October", "holiday", 1.06),
            (date(y, 12, 25), "Christmas Day", "holiday", 1.15),
            (date(y, 12, 26), "2nd Christmas Day", "holiday", 1.08),
        ]
        for dt, name, typ, mult in fixed:
            if start <= dt <= end:
                rows.append({
```

```
"date": dmy(dt),  
"event_name": name,  
"event_type": typ,  
"store_id": np.nan,  
"holiday_flag": 1 if typ == "holiday" else 0,  
"promo_flag": 0,  
"payday_flag": 0,  
"uplift_multiplier (για υπολογισμό ζήτησης/ διόρθωση forecasting)": mult,  
"notes": "Fixed-date event",  
})
```

```
# Payday κάθε 25 (ανοιχτό)
```

```
for m in range(1, 13):
```

```
    dt = date(y, m, 25)
```

```
    if start <= dt <= end:
```

```
        rows.append({
```

```
            "date": dmy(dt),
```

```
            "event_name": "Payday",
```

```
            "event_type": "payday",
```

```
            "store_id": np.nan,
```

```
            "holiday_flag": 0,
```

```
            "promo_flag": 0,
```

```
            "payday_flag": 1,
```

```
            "uplift_multiplier (για υπολογισμό ζήτησης/ διόρθωση forecasting)": 1.05,
```

```
            "notes": "Monthly salary uplift",
```

```
        })
```

```
# Back to School
```

```
for dt in [date(y, 8, 25), date(y, 9, 1), date(y, 9, 8)]:
```

```
    if start <= dt <= end:
```

```
        rows.append({
```

```
"date": dmy(dt),  
"event_name": "Back to School",  
"event_type": "event",  
"store_id": np.nan,  
"holiday_flag": 0,  
"promo_flag": 0,  
"payday_flag": 0,  
"uplift_multiplier (για υπολογισμό ζήτησης/ διόρθωση forecasting)": 1.05,  
"notes": "Snacks/groceries uplift",  
})
```

```
# Black Friday
```

```
bf = black_friday(y)
```

```
if start <= bf <= end:
```

```
    rows.append({  
        "date": dmy(bf),  
        "event_name": "Black Friday",  
        "event_type": "event",  
        "store_id": np.nan,  
        "holiday_flag": 0,  
        "promo_flag": 1,  
        "payday_flag": 0,  
        "uplift_multiplier (για υπολογισμό ζήτησης/ διόρθωση forecasting)": 1.04,  
        "notes": "Small uplift (bulk groceries)",  
    })
```

```
# Orthodox Easter
```

```
pascha = orthodox_easter(y)
```

```
clean_monday = pascha - timedelta(days=48)
```

```
tsiknopempti = clean_monday - timedelta(days=11)
```

```
good_friday = pascha - timedelta(days=2)
```

```
holy_saturday = pascha - timedelta(days=1)
easter_monday = pascha + timedelta(days=1)
holy_spirit = pascha + timedelta(days=50)

movable = [
    (tsiknopempti, "Tsiknopempti", "event", 1.08),
    (clean_monday, "Clean Monday", "holiday", 1.06),
    (good_friday, "Good Friday", "holiday", 1.05),
    (holy_saturday, "Holy Saturday", "event", 1.10),
    (pascha, "Easter Sunday", "holiday", 1.05),
    (easter_monday, "Easter Monday", "holiday", 1.06),
    (holy_spirit, "Holy Spirit Day", "holiday", 1.05),
]

for dt, name, typ, mult in movable:
    if start <= dt <= end:
        rows.append({
            "date": dmy(dt),
            "event_name": name,
            "event_type": typ,
            "store_id": np.nan,
            "holiday_flag": 1 if typ == "holiday" else 0,
            "promo_flag": 1 if name == "Tsiknopempti" else 0,
            "payday_flag": 0,
            "uplift_multiplier (για υπολογισμό ζήτησης/ διόρθωση forecasting)": mult,
            "notes": "Movable (Orthodox Easter based)",
        })

# Christmas rush 20-24/12
for day in (20, 21, 22, 23, 24):
    dt = date(y, 12, day)
    if start <= dt <= end:
```

```
rows.append({
    "date": dmy(dt),
    "event_name": "Pre-Christmas Rush",
    "event_type": "seasonal",
    "store_id": np.nan,
    "holiday_flag": 0,
    "promo_flag": 0,
    "payday_flag": 0,
    "uplift_multiplier (για υπολογισμό ζήτησης/ διόρθωση forecasting)": 1.18,
    "notes": "Holiday meals uplift window",
})

# Pre-Easter rush (3 μέρες πριν το Πάσχα)
for delta in (3, 2, 1):
    dt = pascha - timedelta(days=delta)
    if start <= dt <= end:
        rows.append({
            "date": dmy(dt),
            "event_name": "Pre-Easter Rush",
            "event_type": "seasonal",
            "store_id": np.nan,
            "holiday_flag": 0,
            "promo_flag": 0,
            "payday_flag": 0,
            "uplift_multiplier (για υπολογισμό ζήτησης/ διόρθωση forecasting)": 1.15,
            "notes": "Pre-Easter shopping rush",
        })

return pd.DataFrame(rows, columns=[
    "date", "event_name", "event_type", "store_id", "holiday_flag", "promo_flag", "payday_flag",
```

```
"uplift_multiplier (για υπολογισμό ζήτησης/ διόρθωση forecasting)", "notes"  
])
```

```
def build_uplift_map(calendar_df: pd.DataFrame) -> dict:  
    m = {}  
    for _, r in calendar_df.iterrows():  
        dt = datetime.strptime(r["date"], "%d/%m/%Y").date()  
        u = float(r["uplift_multiplier (για υπολογισμό ζήτησης/ διόρθωση forecasting)"])  
        m[dt] = m.get(dt, 1.0) * u  
    return m
```

```
def events_by_date(calendar_df: pd.DataFrame) -> dict:  
    d = {}  
    for _, r in calendar_df.iterrows():  
        dt = datetime.strptime(r["date"], "%d/%m/%Y").date()  
        d.setdefault(dt, []).append(str(r["event_name"]))  
    return d
```

```
# Category-specific multipliers per event_name
```

```
EVENT_CATEGORY_MULT = {  
    "Tsiknopempti": {  
        "Meat": 1.35, "Fish": 0.90, "Bakery": 1.08, "Deli": 1.10, "Groceries": 1.02,  
        "Dairy": 1.03, "Fruits": 1.00, "Vegetables": 0.98,  
    },  
    "Clean Monday": {  
        "Fish": 1.35, "Groceries": 1.08, "Vegetables": 1.05, "Fruits": 1.02,  
        "Meat": 0.85, "Dairy": 0.95, "Bakery": 1.05, "Deli": 0.95,  
    },  
    "Pre-Easter Rush": {  
        "Meat": 1.25, "Bakery": 1.12, "Deli": 1.10, "Dairy": 1.05, "Groceries": 1.05,  
        "Fish": 0.98, "Fruits": 1.02, "Vegetables": 1.05,  
    },  
}
```

```
},  
"Holy Saturday": {  
    "Meat": 1.20, "Bakery": 1.15, "Deli": 1.12, "Groceries": 1.06, "Dairy": 1.06,  
    "Fish": 1.00, "Fruits": 1.02, "Vegetables": 1.05,  
},  
"Pre-Christmas Rush": {  
    "Meat": 1.18, "Bakery": 1.10, "Deli": 1.10, "Dairy": 1.06, "Groceries": 1.06,  
    "Fish": 1.02, "Fruits": 1.03, "Vegetables": 1.03,  
},  
"Assumption 15Aug": {  
    "Meat": 1.10, "Bakery": 1.08, "Deli": 1.08, "Groceries": 1.05,  
    "Fish": 1.02, "Dairy": 1.03, "Fruits": 1.03, "Vegetables": 1.03,  
},  
"Black Friday": {  
    "Groceries": 1.08, "Dairy": 1.02, "Bakery": 1.01, "Meat": 1.01, "Fish": 1.00,  
    "Fruits": 1.00, "Vegetables": 1.00, "Deli": 1.01,  
},  
"Back to School": {  
    "Groceries": 1.10, "Dairy": 1.03, "Bakery": 1.03, "Fruits": 1.02, "Vegetables": 1.01,  
    "Meat": 1.01, "Fish": 1.00, "Deli": 1.01,  
},  
}
```

```
def category_event_multiplier(ev_names: list[str], category: str) -> float:  
    m = 1.0  
    for name in ev_names:  
        if name in EVENT_CATEGORY_MULT and category in  
EVENT_CATEGORY_MULT[name]:  
            m *= EVENT_CATEGORY_MULT[name][category]  
    return m
```

```
# DEMAND SHIFT FOR CLOSED DAYS
```

```
def prev_open_days(dt: date, CLOSED: set[date], k: int) -> list[date]:
```

```
    out = []
```

```
    cur = dt - timedelta(days=1)
```

```
    while cur >= START_DATE and len(out) < k:
```

```
        if cur not in CLOSED:
```

```
            out.append(cur)
```

```
            cur -= timedelta(days=1)
```

```
    return out
```

```
def build_shifted_demand(raw_demand: dict, CLOSED: set[date], weights=(0.7, 0.3)) ->  
dict:
```

```
    shifted = dict(raw_demand)
```

```
    for (dt, pid), dem in raw_demand.items():
```

```
        if dt in CLOSED and dem > 0:
```

```
            shifted[(dt, pid)] = 0
```

```
            prev_days = prev_open_days(dt, CLOSED, k=len(weights))
```

```
            for w, pdt in zip(weights, prev_days):
```

```
                shifted[(pdt, pid)] = shifted.get((pdt, pid), 0) + int(round(dem * w))
```

```
    return shifted
```

```
# PRODUCTS
```

```
def make_products(rng: np.random.Generator) -> pd.DataFrame:
```

```
    name_pool = {
```

```
        "Meat": ["Chicken Breast", "Beef Mince", "Pork Chops", "Turkey Fillet",
```

```
        "Sausages"],
```

```
        "Fish": ["Salmon", "Sea Bream", "Cod", "Tuna", "Shrimps"],
```

```
"Dairy": ["Milk", "Yogurt", "Cheese", "Butter", "Cream"],
"Groceries": ["Pasta", "Rice", "Canned Beans", "Olive Oil", "Cereal", "Coffee",
"Tea", "Flour"],
"Fruits": ["Bananas", "Apples", "Oranges", "Grapes", "Peaches", "Strawberries"],
"Vegetables": ["Tomatoes", "Cucumbers", "Potatoes", "Onions", "Lettuce",
"Carrots"],
"Bakery": ["Bread", "Baguette", "Croissant", "Muffin", "Pita"],
"Deli": ["Ham Slices", "Salami", "Feta Pack", "Olives", "Ready Salad"],
}
unit_pool = {
    "Meat": ["pack", "kg"],
    "Fish": ["pack", "kg"],
    "Dairy": ["L", "pack"],
    "Groceries": ["pack"],
    "Fruits": ["kg"],
    "Vegetables": ["kg"],
    "Bakery": ["pcs", "pack"],
    "Deli": ["pack"],
}

rows = []
i = 1
for cat, n in SKU_SPLIT.items():
    for _ in range(n):
        pid = f"P{i:03d}"
        sku = f"{cat[:2].upper()}{i:04d}"
        sub = rng.choice(name_pool[cat])
        unit = rng.choice(unit_pool[cat])

        shelf_min, shelf_max = SHELF_LIFE_RANGES[cat]
        shelf = int(rng.integers(shelf_min, shelf_max + 1))
```

```
# ΠΙΟ ΠΕΑΛΙΣΤΙΚΗ ΤΙΜΟΛΟΓΗΣΗ:
# 1) προτιμούμε sub_category range αν υπάρχει
# 2) αλλιώς επιστροφή στο category range
if sub in SUBCAT_PRICE:
    pmin, pmax = SUBCAT_PRICE[sub]
else:
    pmin, pmax = PRICE_RANGES[cat]

# 3) μικρή προσαρμογή ανά μονάδα (π.χ. pcs χαμηλότερο)
mult = UNIT_PRICE_MULT.get(unit, 1.0)
base_price = float(np.round(rng.uniform(pmin, pmax) * mult, 2))

# 4) brand factor για διαφοροποίηση μεταξύ ίδιων subcats
brand_factor = float(rng.uniform(0.95, 1.10))
base_price = float(np.round(base_price * brand_factor, 2))

# Κόστος μέσω margin
mmin, mmax = MARGIN_RANGES[cat]
margin = float(rng.uniform(mmin, mmax))
unit_cost = float(np.round(base_price * (1 - margin), 2))

disposal = float(np.round(rng.uniform(0.05, 0.80), 2))
holding = float(np.round(rng.uniform(0.005, 0.03), 3))

rows.append({
    "product_id": pid,
    "sku": sku,
    "name": f"{sub} {unit}",
    "category": cat,
    "sub_category": sub,
```

```
"unit": unit,
"service_criticality": SERVICE_CRITICALITY.get(sub, 0.5),
"base_price": base_price,
"unit_cost": unit_cost,
"gross_margin_pct": float(np.round(margin, 4)),
"shelf_life_days": shelf,
"disposal_cost_per_unit": disposal,
"holding_cost_per_unit_day": holding,
})
i += 1

return pd.DataFrame(rows)

# SERVICE CRITICALITY

SERVICE_CRITICALITY = {
    # High criticality
    "Milk": 1.0,
    "Bread": 1.0,
    "Yogurt": 1.0,
    "Cheese": 1.0,
    "Bananas": 1.0,
    "Apples": 1.0,
    "Tomatoes": 1.0,
    "Potatoes": 1.0,
    "Onions": 1.0,
    "Lettuce": 1.0,
    "Chicken Breast": 1.0,
    "Pork Chops": 1.0,
    "Pasta": 1.0,
    "Rice": 1.0,
```

"Olive Oil": 1.0,

Medium criticality

"Butter": 0.5,

"Cream": 0.5,

"Cereal": 0.5,

"Flour": 0.5,

"Ham Slices": 0.5,

"Feta Pack": 0.5,

"Cucumbers": 0.5,

"Carrots": 0.5,

"Oranges": 0.5,

"Baguette": 0.5,

Low criticality

"Salmon": 0.0,

"Shrimps": 0.0,

"Sea Bream": 0.0,

"Tuna": 0.0,

"Grapes": 0.0,

"Strawberries": 0.0,

"Croissant": 0.0,

"Muffin": 0.0,

"Salami": 0.0,

"Ready Salad": 0.0,

"Coffee": 0.0,

"Tea": 0.0,

}

MAIN

```
def generate():  
    rng = np.random.default_rng(SEED)  
    ensure_dir(OUT_DIR)  
  
    # Products  
    products = make_products(rng)  
  
    # Calendar + uplift maps  
    calendar = make_calendar(START_DATE, END_DATE)  
    u_map = build_uplift_map(calendar)  
    ev_by_dt = events_by_date(calendar)  
  
    # Closed days  
    CLOSED = holiday_dates_greece(START_DATE, END_DATE)  
  
    # Demand parameters per SKU  
    demand_base = {}  
    demand_trend = {}  
    for _, p in products.iterrows():  
        pid = p["product_id"]  
        cat = p["category"]  
        dmin, dmax = BASE_DEMAND_RANGES[cat]  
        demand_base[pid] = float(rng.uniform(dmin, dmax))  
        demand_trend[pid] = float(rng.uniform(-0.06, 0.10))  
  
    total_days = (END_DATE - START_DATE).days + 1  
  
    #PASS 1: raw demand  
    raw_demand = {}  
    for day_idx, dt in enumerate(daterange(START_DATE, END_DATE)):
```

```
wk = weekly_mult(dt)
yr = year_sine(dt)
ev_global = u_map.get(dt, 1.0)
t = day_idx / max(1, total_days - 1)
ev_names = ev_by_dt.get(dt, [])

for _, p in products.iterrows():
    pid = p["product_id"]
    cat = p["category"]

    cat_mult = 1.0
    if cat == "Fruits":
        cat_mult = fruit_season_mult(dt)
    elif cat == "Vegetables":
        cat_mult = veg_season_mult(dt)

    cat_ev = category_event_multiplier(ev_names, cat)
    trend_mult = 1.0 + demand_trend[pid] * t

    lam = demand_base[pid] * trend_mult * wk * yr * ev_global * cat_mult * cat_ev
    lam = lam * np.exp(rng.normal(0, NOISE_SIGMA))

    raw_demand[(dt, pid)] = int(rng.poisson(max(0.0, lam)))

# SHIFT: κλειστές μέρες -> προηγούμενες ανοιχτές
shifted_demand = build_shifted_demand(raw_demand, CLOSED,
weights=SHIFT_WEIGHTS)

# Inventory proxy init (on_hand per SKU)
on_hand = {}
for _, p in products.iterrows():
```

```
pid = p["product_id"]
cover = float(rng.uniform(INIT_COVER_MIN, INIT_COVER_MAX))
on_hand[pid] = int(round(demand_base[pid] * cover))
```

```
# PASS 2: transactions + waste
```

```
txn_rows = []
```

```
waste_rows = []
```

```
# Delivery scheduling:
```

```
delivery_due = False
```

```
for dt in daterange(START_DATE, END_DATE):
```

```
    is_closed = dt in CLOSED
```

```
    # Προγραμματισμός παραλαβής κάθε Δευτέρα
```

```
    if dt.weekday() == 0: # Monday
```

```
        delivery_due = True
```

```
    # Εκτέλεση παραλαβής στην πρώτη επόμενη ανοιχτή μέρα
```

```
    deliver_today = (delivery_due and (not is_closed))
```

```
for _, p in products.iterrows():
```

```
    pid = p["product_id"]
```

```
    cat = p["category"]
```

```
    shelf = int(p["shelf_life_days"])
```

```
    base_price = float(p["base_price"])
```

```
    unit_cost = float(p["unit_cost"])
```

```
    disposal = float(p["disposal_cost_per_unit"])
```

```
    # Promo/discount μόνο αν είναι ανοιχτό
```

```
promo_flag = 0
discount_pct = 0
price = base_price
if not is_closed:
    promo_flag = 1 if rng.random() < PROMO_PROB else 0
    if promo_flag:
        discount_pct = int(rng.choice([10, 15, 20, 25]))
    else:
        if shelf <= 4 and rng.random() < 0.06:
            discount_pct = int(rng.choice([10, 15]))
        price = float(np.round(base_price * (1 - discount_pct / 100.0), 2))

# Demand after shifting
dem = int(shifted_demand.get((dt, pid), 0))

# Sales
if is_closed:
    units_sold = 0
    lost_sales = 0
    stockout_flag = 0
else:
    units_sold = min(dem, on_hand[pid])
    lost_sales = max(0, dem - units_sold)
    stockout_flag = 1 if lost_sales > 0 else 0
    on_hand[pid] -= units_sold

# Deliveries (order-up-to) if deliver_today
if deliver_today:
    target = int(round(demand_base[pid] * TARGET_COVER_DAYS))
    if on_hand[pid] < target:
        on_hand[pid] = target
```

```
# Οικονομικά
revenue = float(np.round(units_sold * price, 2))
cogs = float(np.round(units_sold * unit_cost, 2))
gp = float(np.round(revenue - cogs, 2))

txn_rows.append({
    "date": dmy(dt),
    "store_id": STORE_ID,
    "product_id": pid,
    "units_sold": int(units_sold),
    "price": float(price),
    "discount_pct": int(discount_pct),
    "promo_flag": int(promo_flag),
    "revenue (units sold*price)": revenue,
    "cogs (cost of unit sold)": cogs,
    "gross_profit (revenue-cogs)": gp,
    "stockout_flag": int(stockout_flag),
    "lost_sales_units (για RL)": int(lost_sales),
})

# DAILY SPOILAGE (τρέχει κάθε μέρα, ανοιχτό ή κλειστό)
spoil_rate = DAILY_SPOILAGE_RATE[cat]
if dem > demand_base[pid] * 1.6:
    spoil_rate += 0.003

qty_wasted = int(rng.binomial(max(on_hand[pid], 0), min(0.35, spoil_rate)))

if qty_wasted > 0:
    on_hand[pid] = max(0, on_hand[pid] - qty_wasted)
```

```
if cat in ("Fish", "Meat", "Fruits", "Vegetables", "Bakery", "Deli", "Dairy"):
```

```
    reason = "expired" if rng.random() < 0.80 else "damaged"
```

```
else:
```

```
    reason = "damaged" if rng.random() < 0.80 else "expired"
```

```
expired_qty = qty_wasted if reason == "expired" else 0
```

```
damaged_qty = qty_wasted if reason == "damaged" else 0
```

```
overstock_qty = 0
```

```
waste_cost = float(np.round(qty_wasted * (unit_cost + disposal), 2))
```

```
waste_rows.append({  
    "date": dmy(dt),  
    "store_id": STORE_ID,  
    "product_id": pid,  
    "qty_wasted": int(qty_wasted),  
    "waste_reason": reason,  
    "waste_cost": waste_cost,  
    "expired_qty": int(expired_qty),  
    "damaged_qty": int(damaged_qty),  
    "overstock_qty": int(overstock_qty),  
})
```

```
# Αν κάναμε παραλαβή σήμερα, κλείνει το pending
```

```
if deliver_today:
```

```
    delivery_due = False
```

```
# Save outputs
```

```
transactions_df = pd.DataFrame(txn_rows, columns=[  
    "date", "store_id", "product_id", "units_sold", "price", "discount_pct", "promo_flag",  
    "revenue (units sold*price)", "cogs (cost of unit sold)", "gross_profit (revenue-cogs)",
```

```
"stockout_flag","lost_sales_units (για RL)"
])
waste_df = pd.DataFrame(waste_rows, columns=[
    "date","store_id","product_id","qty_wasted","waste_reason","waste_cost",
    "expired_qty","damaged_qty","overstock_qty"
])

products.to_csv(f'{OUT_DIR}/products.csv', index=False, sep=";",
encoding="cp1253")
calendar.to_csv(f'{OUT_DIR}/calendar_events.csv', index=False, sep=";",
encoding="cp1253")
transactions_df.to_csv(f'{OUT_DIR}/transactions_daily.csv', index=False, sep=";",
encoding="cp1253")
waste_df.to_csv(f'{OUT_DIR}/waste_daily.csv', index=False, sep=";",
encoding="cp1253")

#Report
expected = ((END_DATE - START_DATE).days + 1) * len(products)
print("DONE ->", OUT_DIR)
print("SKUs:", len(products))
print("Days:", (END_DATE - START_DATE).days + 1)
print("Closed days:", len(CLOSED))
print("transactions rows:", len(transactions_df), f"(expected {expected})")
print("waste rows:", len(waste_df))
print("calendar rows:", len(calendar))

if __name__ == "__main__":
    generate()
```

8.2. Ο κώδικας για την πρόβλεψη της ζήτησης

```
# neuralprophet_sku_train_test_80_20.py

# NeuralProphet forecasting ανά SKU (product_id)
# - Χρονικό 80/20 split
# - Forecast horizon = μήκος test set (ώστε να αξιολογήσουμε ολο το test)
# - Metrics ανά SKU + συνολικά
# - Plots και Line plots σε .png για παρουσίαση
#
# Outputs στον φάκελο: "test prophet sku"


import os
import warnings
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from neuralprophet import NeuralProphet

warnings.filterwarnings("ignore")

# ΠΥΘΜΙΣΕΙΣ

TRANSACTIONS_FILE = "transactions_daily.csv"

DATE_COL = "date"
SKU_COL = "product_id"
TARGET = "units_sold"

TRAIN_RATIO = 0.80
```

```
EPOCHS = 200
```

```
LEARNING_RATE = 0.01
```

```
OUTPUT_DIR = "test prophet sku"
```

```
# Αν θέλουμε να τρέχουμε δοκιμαστικά σε λίγα SKUs:
```

```
# π.χ. ["P001", "P010", "P050"] ή None για όλα
```

```
ONLY_SKUS = None
```

```
# Πόσα plots να φτιάξει (για να μη βγάλει 130 png αν δεν θελουμε)
```

```
N_PLOTS = 130
```

```
# HELPERS
```

```
def read_csv_flexible(path: str) -> pd.DataFrame:
```

```
    """Διαβάζει CSV με διάφορα encodings/separators (τα δικά μου είναι συνήθως cp1253  
    + ';')."""
```

```
    for enc in ["utf-8", "utf-8-sig", "cp1253", "latin1"]:
```

```
        for sep in [";", ",", "\t"]:
```

```
            try:
```

```
                df = pd.read_csv(path, sep=sep, encoding=enc)
```

```
                # Αν διάβασε λάθος και έφερε 1 στήλη με ';' μέσα, πάμε στην επόμενη  
                # προσπάθεια.
```

```
                if df.shape[1] == 1 and ";" in str(df.iloc[0, 0]):
```

```
                    continue
```

```
                return df
```

```
            except Exception:
```

```
                pass
```

```
            raise RuntimeError(f'Δεν μπόρεσα να διαβάσω το αρχείο: {path}')
```

```
def safe_mape(y_true: np.ndarray, y_pred: np.ndarray) -> float:
    denom = np.maximum(1.0, y_true) # για να μη διαιρούμε με 0
    return float(np.mean(np.abs((y_true - y_pred) / denom)) * 100.0)

def smape(y_true: np.ndarray, y_pred: np.ndarray) -> float:
    denom = (np.abs(y_true) + np.abs(y_pred) + 1e-9)
    return float(np.mean(2.0 * np.abs(y_pred - y_true) / denom) * 100.0)

def ensure_daily_panel(df: pd.DataFrame, all_days: pd.DatetimeIndex) -> pd.DataFrame:
    """
    Παίρνει ένα df με columns [ds, y] για ένα SKU και το κάνει daily panel:
    - reindex σε όλα τα days
    - fill missing y με 0
    """
    df = df.set_index("ds").reindex(all_days).rename_axis("ds").reset_index()
    df["y"] = df["y"].fillna(0.0)
    return df

def make_plot(actual_df: pd.DataFrame, pred_df: pd.DataFrame, outpath: str, title: str):
    """
    actual_df: columns [ds, y]
    pred_df: columns [ds, yhat1]
    Κάνει plot actual vs forecast και το σώζει.
    """
    plt.figure()
    plt.plot(actual_df["ds"], actual_df["y"], label="actual")
    plt.plot(pred_df["ds"], pred_df["yhat1"], label="forecast")
    plt.title(title)
    plt.legend()
    plt.tight_layout()
    plt.savefig(outpath, dpi=160)
```

```
plt.close()
```

```
# Δύο γραμμές (Actual vs Forecast)
```

```
# 1) πραγματική ζήτηση
```

```
# 2) πρόβλεψη του μοντέλου
```

```
def make_double_line_plot(actual_df: pd.DataFrame, pred_df: pd.DataFrame, outpath:  
str, title: str):
```

```
# Merge actual + prediction με βάση την ημερομηνία
```

```
merged = actual_df.merge(pred_df, on="ds")
```

```
# Μέγεθος γραφήματος
```

```
plt.figure(figsize=(10, 5))
```

```
# 1η γραμμή = πραγματική ζήτηση
```

```
plt.plot(  
    merged["ds"],  
    merged["y"],  
    label="actual",  
    linewidth=2  
)
```

```
# 2η γραμμή = πρόβλεψη του NeuralProphet
```

```
plt.plot(  
    merged["ds"],  
    merged["yhat1"],
```

```
        label="forecast",
        linewidth=2
    )

plt.title(title)

plt.legend()

# Βελτίωση spacing
plt.tight_layout()

# Save σε png
plt.savefig(outpath, dpi=160)

# Κλείσιμο figure για να μην γεμίζει η μνήμη
plt.close()

# MAIN

def main():
    os.makedirs(OUTPUT_DIR, exist_ok=True)

    #Load
    df = read_csv_flexible(TRANSACTIONS_FILE)

    #Basic checks
    for col in [DATE_COL, SKU_COL, TARGET]:
        if col not in df.columns:
            raise ValueError(f'Λείπει η στήλη '{col}'. Columns: {list(df.columns)}')
```

```
#Parse date + numeric target
```

```
df[DATE_COL] = pd.to_datetime(df[DATE_COL], dayfirst=True, errors="coerce")
```

```
df = df.dropna(subset=[DATE_COL]).copy()
```

```
df[TARGET] = pd.to_numeric(df[TARGET], errors="coerce")
```

```
df = df.dropna(subset=[TARGET]).copy()
```

```
#Κρατάμε μόνο ό,τι χρειαζόμαστε
```

```
df = df[[DATE_COL, SKU_COL, TARGET]].copy()
```

```
df[SKU_COL] = df[SKU_COL].astype(str)
```

```
#Επιλογή SKUs (αν θέλουμε subset)
```

```
all_skus = sorted(df[SKU_COL].unique().tolist())
```

```
if ONLY_SKUS is not None:
```

```
    only = set(str(x) for x in ONLY_SKUS)
```

```
    all_skus = [s for s in all_skus if s in only]
```

```
if len(all_skus) == 0:
```

```
    raise ValueError("Δεν βρέθηκαν SKUs για εκπαίδευση. Έλεγξε ONLY_SKUS ή το  
αρχείο.")
```

```
#Φτιάχνουμε daily series ανά SKU (και βεβαιώνουμε ότι έχει όλες τις μέρες)
```

```
min_day = df[DATE_COL].min()
```

```
max_day = df[DATE_COL].max()
```

```
all_days = pd.date_range(min_day, max_day, freq="D")
```

```
#Χρονικό split point (κοινό για όλα τα SKUs)
```

```
split_idx = int(len(all_days) * TRAIN_RATIO)
```

```
split_day = all_days[split_idx - 1] # last day of train
```

```
train_days = all_days[:split_idx]
test_days = all_days[split_idx:]
test_horizon = len(test_days)
```

```
#Ετοιμάζουμε dataframe για NeuralProphet global model με ID
```

```
panels = []
for sku in all_skus:
    dsku = (
        df[df[SKU_COL] == sku]
        .groupby(DATE_COL, as_index=False)
        .agg({TARGET: "sum"})
        .rename(columns={DATE_COL: "ds", TARGET: "y"})
        .sort_values("ds")
    )
    dsku = ensure_daily_panel(dsku, all_days)
    dsku["ID"] = sku
    panels.append(dsku)
```

```
full_panel = pd.concat(panels, ignore_index=True)
```

```
train_df = full_panel[full_panel["ds"].isin(train_days)].copy()
test_df = full_panel[full_panel["ds"].isin(test_days)].copy()
```

```
# Save split (χρήσιμο για report)
```

```
train_df.to_csv(os.path.join(OUTPUT_DIR, "train_panel.csv"), index=False,
encoding="utf-8-sig")
test_df.to_csv(os.path.join(OUTPUT_DIR, "test_panel.csv"), index=False,
encoding="utf-8-sig")
```

```
#Μοντέλο NeuralProphet (global)
# Global model = μαθαίνει patterns κοινά μεταξύ SKUs + κρατάει ID normalization.
model = NeuralProphet(
    yearly_seasonality=True,
    weekly_seasonality=True,
    daily_seasonality=False,
    learning_rate=LEARNING_RATE,
    epochs=EPOCHS,
)

# Fit
model.fit(train_df, freq="D")

#Forecast ακριβώς για όλο το test horizon
future = model.make_future_dataframe(train_df, periods=test_horizon)
forecast = model.predict(future)

forecast.to_csv(os.path.join(OUTPUT_DIR, "forecast_full.csv"), index=False,
encoding="utf-8-sig", sep=";",
decimal=",")

#Evaluation: merge forecast με test για να αξιολογήσουμε ΟΛΟ το test (80/20 σωστά)
#To NeuralProphet επιστρέφει: ds, yhat1, και ID (για multi-series)
needed_cols = ["ds", "ID", "yhat1"]
for c in needed_cols:
    if c not in forecast.columns:
        raise RuntimeError(f"Στο forecast λείπει η στήλη '{c}'. Columns:
{list(forecast.columns)}")

fc_test = forecast[needed_cols].merge(
    test_df[["ds", "ID", "y"]],
```

```
on=["ds", "ID"],
how="inner"
)

if len(fc_test) == 0:
    raise RuntimeError("Δεν βρέθηκε overlap μεταξύ forecast και test. Κάτι πήγε στραβά
με το split.")

# Dataset για ανάλυση
# Δημιουργία αρχείου με:
# - πραγματική ζήτηση
# - πρόβλεψη
# - error
# - μεταβλητότητα (std (τυπική απόκλιση ζήτησης ανά SKU), mean(Μέσος όρος
ζήτησης ανά SKU),
# cv ( Coefficiency of Variation = Standard Deviation / Mean)

"""
CV < 0.5: Χαμηλή μεταβλητότητα (σταθερή ζήτηση)
CV 0.5-1.0: Μέτρια μεταβλητότητα
CV > 1.0: Υψηλή μεταβλητότητα (ακανόνιστη ζήτηση)
"""

comparison = fc_test.copy()

# error (signed)
comparison["error"] = comparison["y"] - comparison["yhat1"]

# absolute error
```

```
comparison["abs_error"] = np.abs(comparison["error"])

# υπολογισμός μεταβλητότητας ανά SKU
variability = (
    comparison.groupby("ID")["y"]
    .agg(["mean", "std"])
    .reset_index()
)

# coefficient of variation
variability["cv"] = variability["std"] / (variability["mean"] + 1e-9)

# merge πίσω στο dataset
comparison = comparison.merge(variability, on="ID", how="left")

# save αρχείου
comparison.to_csv(
    os.path.join(OUTPUT_DIR, "actual_vs_forecast_full.csv"),
    index=False,
    encoding="utf-8-sig",
    sep=";",
    decimal=","
)

# Metrics ανά SKU
metrics_rows = []
for sku, g in fc_test.groupby("ID"):
    y_true = g["y"].to_numpy(dtype=float)
    y_pred = g["yhat1"].to_numpy(dtype=float)

    mae = float(np.mean(np.abs(y_true - y_pred)))
    rmse = float(np.sqrt(np.mean((y_true - y_pred) ** 2)))
```

```
mape = safe_mape(y_true, y_pred)
s_mape = smape(y_true, y_pred)

metrics_rows.append({
    "product_id": sku,
    "test_days": len(g),
    "MAE": mae,
    "RMSE": rmse,
    "MAPE_%": mape,
    "sMAPE_%": s_mape,
    "y_mean_test": float(np.mean(y_true)),
    "y_sum_train": float(train_df[train_df["ID"] == sku]["y"].sum()),
})

metrics_df = pd.DataFrame(metrics_rows).sort_values("y_sum_train",
ascending=False)

metrics_df.to_csv(os.path.join(OUTPUT_DIR, "metrics_per_sku.csv"), index=False,
encoding="utf-8-sig",
sep=";", decimal=",")

# Συνολικά metrics (micro-average σε όλα τα SKU-days)
y_true_all = fc_test["y"].to_numpy(dtype=float)
y_pred_all = fc_test["yhat1"].to_numpy(dtype=float)

overall_mae = float(np.mean(np.abs(y_true_all - y_pred_all)))
overall_rmse = float(np.sqrt(np.mean((y_true_all - y_pred_all) ** 2)))
overall_mape = safe_mape(y_true_all, y_pred_all)
overall_smape = smape(y_true_all, y_pred_all)

with open(os.path.join(OUTPUT_DIR, "summary.txt"), "w", encoding="utf-8") as f:
    f.write("NEURALPROPHET SKU-LEVEL TRAIN/TEST SUMMARY\n")
```

```
f.write("-----\n")
f.write(f"File: {TRANSACTIONS_FILE}\n")
f.write(f"Target: {TARGET}\n")
f.write(f"SKUs: {len(all_skus)}\n")
f.write(f"Total days: {len(all_days)}\n")
f.write(f"Train ratio: {TRAIN_RATIO}\n")
f.write(f"Train days: {len(train_days)} | Test days: {len(test_days)}\n")
f.write(f"Train period: {train_days.min().date()} → {train_days.max().date()}\n")
f.write(f"Test period: {test_days.min().date()} → {test_days.max().date()}\n")
f.write("\nOVERALL METRICS (all SKU-days)\n")
f.write(f"MAE: {overall_mae:.3f}\n")
f.write(f"RMSE: {overall_rmse:.3f}\n")
f.write(f"MAPE(%): {overall_mape:.3f}\n")
f.write(f"sMAPE(%): {overall_smape:.3f}\n")

# Plots για παρουσίαση: παίρνουμε τα top SKUs σε train sales
os.makedirs(os.path.join(OUTPUT_DIR, "plots"), exist_ok=True)
# φάκελος για line + line plots
os.makedirs(os.path.join(OUTPUT_DIR, "double_line_plots"), exist_ok=True)

top_skus = metrics_df["product_id"].head(N_PLOTS).tolist()

for sku in top_skus:
    actual = full_panel[full_panel["ID"] == sku][["ds", "y"]].copy()
    pred = forecast[forecast["ID"] == sku][["ds", "yhat1"]].copy()

    # Θέλουμε plot σε όλο το διάστημα (train+test+forecast)
    # αλλά για να φαίνεται καθαρά δείχνουμε όλο range:
    outpath = os.path.join(OUTPUT_DIR, "plots", f"{sku}_actual_vs_forecast.png")
    title = f"NeuralProphet Forecast | {sku} | 80/20 split @ {split_day.date()}"
    make_plot(actual, pred, outpath, title)
```

```
# Line + Line plot
double_line_outpath = os.path.join(OUTPUT_DIR, "double_line_plots",
f"{sku}_double_line.png")

# Τίτλος plot
double_line_title = (f"NeuralProphet | {sku} | Actual vs Forecast ") #(Last 60 Days)
αν χρειαστεί

# Δημιουργία και save του νέου plot
make_double_line_plot(actual, pred, double_line_outpath, double_line_title)

print("Τέλος!")
print(f"Outputs: '{OUTPUT_DIR}'")
print(f"SKUs trained: {len(all_skus)}")
print(f"Split day (end of train): {split_day.date()}")
print(f"OVERALL: MAE={overall_mae:.2f} | RMSE={overall_rmse:.2f} |
MAPE={overall_mape:.2f}% | sMAPE={overall_smape:.2f}%")
print(f"Plots saved: {os.path.join(OUTPUT_DIR, 'plots')}")

if __name__ == "__main__":
    main()
```

8.3. Ο κώδικας για την ABC ανάλυση

```
# Multicriteria ABC – Προεπεξεργασία και Υπολογισμός KPI
```

```
import pandas as pd
import numpy as np
from scipy.stats import pearsonr
```

```
# Load δεδομένων
```

```
products = pd.read_csv("synthetic_out/products.csv", sep=";", encoding="cp1253")
transactions = pd.read_csv("synthetic_out/transactions_daily.csv", sep=";",
encoding="cp1253")
waste = pd.read_csv("synthetic_out/waste_daily.csv", sep=";", encoding="cp1253")
calendar = pd.read_csv("synthetic_out/calendar_events.csv", sep=";", encoding="cp1253")
```

```
# Μετατροπή ημερομηνιών για να μπορεί να γίνει merge
```

```
transactions["date"] = pd.to_datetime(transactions["date"], dayfirst=True)
calendar["date"] = pd.to_datetime(calendar["date"], dayfirst=True)
```

```
# Πραγματική ζήτηση
```

```
transactions["real_demand"] = (
    transactions["units_sold"] +
    transactions["lost_sales_units (για RL)"]
)
```

```
# Merge transactions με subcategory
```

```
trx = transactions.merge(
    products[["product_id", "sub_category"]],
    on="product_id",
    how="left"
)
```

```
# Merge με calendar ώστε κάθε γραμμή να ξέρει αν είναι event day
```

```
trx = trx.merge(
    calendar[["date", "event_name"]],
```

```
on="date",
how="left"
)

# Dictionary:
# key = sub_category
# value = μέγιστο uplift που παρατηρήθηκε σε κάποιο event
subcategory_uplift = {}

# Loop σε όλες τις υποκατηγορίες
for subcat in trx["sub_category"].dropna().unique():

    # Κρατάμε μόνο τις γραμμές της συγκεκριμένης υποκατηγορίας
    dsub = trx[trx["sub_category"] == subcat]

    # Μέση πραγματική ζήτηση σε normal μέρες (χωρίς event)
    normal_mean = dsub[
        dsub["event_name"].isna()
    ]["real_demand"].mean()

    # Default uplift = 1 (δηλαδή καμία αλλαγή)
    max_uplift = 1.0

    # Loop σε όλα τα διαφορετικά events αυτής της υποκατηγορίας
    for ev in dsub["event_name"].dropna().unique():

        # Μέση ζήτηση μόνο για το συγκεκριμένο event
        ev_mean = dsub[
            dsub["event_name"] == ev
        ]["real_demand"].mean()
```

```
# Αν υπάρχει normal demand, υπολογίζουμε uplift
# uplift = event demand / normal demand
if pd.notna(normal_mean) and normal_mean > 0:
    uplift = ev_mean / normal_mean

# Κρατάμε το μεγαλύτερο uplift
max_uplift = max(max_uplift, uplift)

# Save στο dictionary
subcategory_uplift[subcat] = max_uplift
# Μετατροπή dictionary σε dataframe
subcat_uplift_df = pd.DataFrame({
    "sub_category": list(subcategory_uplift.keys()),
    "subcategory_event_uplift": list(subcategory_uplift.values())
})

# KPI ανά SKU

# Aggregation από transactions
sku_txn = transactions.groupby("product_id").agg({
    "revenue (units sold*price)": "sum",
    "gross_profit (revenue-cogs)": "sum",
    "real_demand": ["mean", "std", "sum"],
    "stockout_flag": "mean"
})

sku_txn.columns = [
    "sales_value",
    "gross_profit",
    "mean_demand",
    "std_demand",
```

```
"total_demand",
"stockout_frequency"
]

sku_txn = sku_txn.reset_index()

# CV ζήτησης
sku_txn["cv_demand"] = (
    sku_txn["std_demand"] /
    sku_txn["mean_demand"].replace(0, np.nan)
)
sku_txn["cv_demand"] = sku_txn["cv_demand"].fillna(0)

# KPI από waste

sku_waste = waste.groupby("product_id").agg({
    "qty_wasted": "sum",
    "waste_cost": "sum"
}).reset_index()

# Merge με total demand για spoilage rate
sku_waste = sku_waste.merge(
    sku_txn[["product_id", "total_demand"]],
    on="product_id",
    how="left"
)

sku_waste["spoilage_rate"] = (
    sku_waste["qty_wasted"] /
    sku_waste["total_demand"].replace(0, np.nan)
)
```

```
)  
sku_waste["spoilage_rate"] = sku_waste["spoilage_rate"].fillna(0)
```

```
# Merge με products
```

```
abc_df = products[[  
    "product_id",  
    "name",  
    "category",  
    "sub_category",  
    "shelf_life_days",  
    "disposal_cost_per_unit",  
    "service_criticality"  
]].merge(sku_txn, on="product_id", how="left") \  
.merge(sku_waste[[  
    "product_id",  
    "qty_wasted",  
    "waste_cost",  
    "spoilage_rate"  
]], on="product_id", how="left")
```

```
abc_df = abc_df.fillna(0)
```

```
abc_df = abc_df.merge(  
    subcat_uplift_df,  
    on="sub_category",  
    how="left"  
)
```

```
# Αν κάποιο SKU δεν έχει uplift, βάζουμε 1.0 (= no effect)
```

```
abc_df["subcategory_event_uplift"] = (  
    abc_df["subcategory_event_uplift"]  
    .fillna(1.0)  
)
```

```
# Correlation matrix
```

```
correlation_cols = [  
    "sales_value",  
    "gross_profit",  
    "cv_demand",  
    "spoilage_rate",  
    "shelf_life_days",  
    "waste_cost",  
    "stockout_frequency",  
    "service_criticality",  
    "subcategory_event_uplift"  
]
```

```
# Correlation coefficients
```

```
corr_matrix = abc_df[correlation_cols].corr()
```

```
# P-values
```

```
p_matrix = pd.DataFrame(  
    index=correlation_cols,  
    columns=correlation_cols  
)
```

```
for col1 in correlation_cols:  
    for col2 in correlation_cols:
```

```
if col1 == col2:
    p_matrix.loc[col1, col2] = np.nan

else:
    r, p = pearsonr(
        abc_df[col1],
        abc_df[col2]
    )

    p_matrix.loc[col1, col2] = p

print("\nCorrelation Matrix")
print(corr_matrix.round(3))

print("\nP-value Matrix")
print(p_matrix.astype(float).round(4))

# Normalization

# Higher is more important
positive_cols = [
    "sales_value",
    "gross_profit",
    "cv_demand",
    "spoilage_rate",
    "waste_cost",
    "stockout_frequency",
    "service_criticality",
    "subcategory_event_uplift"
]
```

```
for col in positive_cols:
    abc_df[col + "_norm"] = (
        abc_df[col] - abc_df[col].min()
    ) / (
        abc_df[col].max() - abc_df[col].min()
    )

# Shelf life: όσο μικρότερο τόσο πιο κρίσιμο
abc_df["shelf_life_norm"] = 1 - (
    (abc_df["shelf_life_days"] - abc_df["shelf_life_days"].min()) /
    (abc_df["shelf_life_days"].max() - abc_df["shelf_life_days"].min())
)

# Βάρη

weights = {
    "gross_profit_norm": 0.23,
    "cv_demand_norm": 0.20,
    "subcategory_event_uplift_norm": 0.15,
    "spoilage_rate_norm": 0.15,
    "shelf_life_norm": 0.10,
    "waste_cost_norm": 0.10,
    "stockout_frequency_norm": 0.02,
    "service_criticality_norm": 0.05
}

# ABC score

abc_df["abc_score"] = 0
```

```
for col, w in weights.items():
    abc_df["abc_score"] += abc_df[col] * w

# Ταξινόμηση και cumulative score

abc_df = abc_df.sort_values("abc_score", ascending=False).reset_index(drop=True)

abc_df["score_share"] = abc_df["abc_score"] / abc_df["abc_score"].sum()
abc_df["cumulative_share"] = abc_df["score_share"].cumsum()

# Κατηγορίες A / B / C

q_a = abc_df["abc_score"].quantile(0.80)
q_b = abc_df["abc_score"].quantile(0.50)

def classify_abc(score):
    if score >= q_a:
        return "A"
    elif score >= q_b:
        return "B"
    return "C"

abc_df["ABC_class"] = abc_df["abc_score"].apply(classify_abc)

# Έλεγχος αποτελεσμάτων

print(abc_df[
```

```
"product_id",  
"name",  
"category",  
"abc_score",  
"ABC_class"  
]].head(20))  
  
print(abc_df["ABC_class"].value_counts())  
  
# Save αποτελεσμάτων  
  
import os  
  
output_dir = "Multi-ABC Preprocessing and KPIs"  
os.makedirs(output_dir, exist_ok=True)  
  
# Πλήρες αρχείο KPI + ABC  
abc_df.to_csv(  
    os.path.join(output_dir, "abc_results.csv"),  
    index=False,  
    sep=";",  
    encoding="cp1253",  
    decimal = ",",  
)  
  
# Correlation matrix  
corr_matrix.round(3).to_csv(  
    os.path.join(output_dir, "correlation_matrix.csv"),  
    sep=";",  
    encoding="cp1253",
```

```
decimal = ","
)

# P-value matrix
p_matrix.astype(float).round(6).to_csv(
    os.path.join(output_dir, "correlation_pvalues.csv"),
    sep=";",
    encoding="cp1253",
    decimal=","
)

# Top προϊόντα ανά score
abc_df[[
    "product_id",
    "name",
    "category",
    "abc_score",
    "ABC_class"
]].head(30).to_csv(
    os.path.join(output_dir, "top_30_products.csv"),
    index=False,
    sep=";",
    encoding="cp1253",
    decimal = ","
)

# Σύνοψη κατηγοριών ABC
abc_summary =
abc_df["ABC_class"].value_counts().rename_axis("ABC_class").reset_index(name="count")
abc_summary.to_csv(
```

```
os.path.join(output_dir, "abc_summary.csv"),  
index=False,  
sep=";",  
encoding="cp1253",  
decimal = ","  
)
```

```
# Εμφάνιση στην κονσόλα  
print("Files saved in:", output_dir)  
print("- abc_results.csv")  
print("- correlation_matrix.csv")  
print("- top_30_products.csv")  
print("- abc_summary.csv")
```

```
print("Top 20 προϊόντα:")  
print(abc_df[[  
    "product_id",  
    "name",  
    "category",  
    "abc_score",  
    "ABC_class"  
]].head(20))
```

```
print("ABC summary:")  
print(abc_summary)
```

8.4. Ο κώδικας της τελικής μορφής των δεδομένων για το RL

```
# RL_Dataset.py
```

```
import pandas as pd  
import numpy as np
```

```
# LOAD DATA
```

```
products = pd.read_csv(  
    "synthetic_out/products.csv",  
    sep=";",  
    encoding="cp1253"  
)
```

```
transactions = pd.read_csv(  
    "synthetic_out/transactions_daily.csv",  
    sep=";",  
    encoding="cp1253"  
)
```

```
waste = pd.read_csv(  
    "synthetic_out/waste_daily.csv",  
    sep=";",  
    encoding="cp1253"  
)
```

```
calendar = pd.read_csv(  
    "synthetic_out/calendar_events.csv",  
    sep=";",  
    encoding="cp1253"  
)
```

```
abc = pd.read_csv(  

```

```
"Multi-ABC Preprocessing and KPIs/abc_results.csv",  
sep=";",  
encoding="cp1253"  
)
```

```
# DATE CONVERSION
```

```
transactions["date"] = pd.to_datetime(  
    transactions["date"],  
    dayfirst=True  
)
```

```
waste["date"] = pd.to_datetime(  
    waste["date"],  
    dayfirst=True  
)
```

```
calendar["date"] = pd.to_datetime(  
    calendar["date"],  
    dayfirst=True  
)
```

```
# MERGE ABC FEATURES
```

```
base = transactions.merge(  
  
    abc[  
        "product_id",  
        "ABC_class",
```

```
"category",
"spoilage_rate",
"subcategory_event_uplift",
"service_criticality",
"shelf_life_days"
]],

on="product_id",

how="left"
)

# MERGE WASTE

base = base.merge(

waste[[
    "date",
    "product_id",
    "qty_wasted",
    "waste_cost"
]],

on=["date", "product_id"],

how="left"
)

base["qty_wasted"] = base["qty_wasted"].fillna(0)
```

```
base["waste_cost"] = base["waste_cost"].fillna(0)
```

```
# MERGE CALENDAR
```

```
base = base.merge(
```

```
    calendar[[  
        "date",  
        "event_name",  
        "uplift_multiplier (για υπολογισμό ζήτησης/ διόρθωση forecasting)"  
    ]],
```

```
    on="date",
```

```
    how="left"
```

```
)
```

```
# BASIC FEATURES
```

```
# binary event flag
```

```
base["event_flag"] = (  
    base["event_name"]  
    .notna()  
    .astype(int)  
)
```

```
# weekday
```

```
base["day_of_week"] = (  
    base["date"]
```

```
.dt.weekday  
)
```

```
# REAL DEMAND
```

```
base["real_demand"] = (  
  
    base["units_sold"]  
  
    +  
  
    base["lost_sales_units (για RL)"]  
)
```

```
# ELASTICITY MAP
```

```
ELASTICITY_MAP = {  
  
    "Meat": -1.5,  
  
    "Fish": -1.5,  
  
    "Fruits": -1.2,  
  
    "Vegetables": -1.2,  
  
    "Dairy": -0.8,
```

```
"Bakery": -0.8,  
  
"Deli": -0.6,  
  
"Groceries": -0.5  
}  
  
# ASSIGN ELASTICITY  
  
def assign_elasticity(category):  
  
    if category not in ELASTICITY_MAP:  
  
        raise ValueError(  
            f"Unknown category '{category}' "  
            f"in ELASTICITY_MAP"  
        )  
  
    return ELASTICITY_MAP[category]  
  
base["elasticity"] = (  
    base["category"]  
    .apply(assign_elasticity)  
)  
  
# EVENT UPLIFT SIGNAL
```

```
# signal = uplift - 1.0
```

```
# Example:
```

```
# 1.0 -> 0.0
```

```
# 1.3 -> 0.3
```

```
base["event_uplift_signal"] = (  
    base[  
        "uplift_multiplier (για υπολογισμό ζήτησης/ διόρθωση forecasting)"  
    ]  
    .fillna(1.0)  
    - 1.0  
)
```

```
# FEATURE CLIPPING
```

```
# uplift clipping
```

```
base["event_uplift_signal"] = np.clip(  
  
base["event_uplift_signal"],  
  
0.0,  
  
0.5  
)
```

```
# spoilage clipping
```

```
base["spoilage_rate"] = np.clip(  
  

```

```
base["spoilage_rate"],

0.0,

1.0
)

# criticality clipping
base["service_criticality"] = np.clip(

base["service_criticality"],

0.0,

1.0
)

# VALIDATION

required_columns = [

"price",

"real_demand",

"event_flag",

"shelf_life_days",
```

```
"spoilage_rate",

"event_uplift_signal",

"service_criticality",

"elasticity"
]

# CHECK NaNs

for col in required_columns:

    nan_count = base[col].isna().sum()

    if nan_count > 0:

        raise RuntimeError(

            f'Column '{col}' '
            f'contains {nan_count} NaN values"
        )

# CHECK NEGATIVE DEMAND

negative_demand = (
    base["real_demand"] < 0
).sum()
```

```
if negative_demand > 0:
```

```
    raise RuntimeError(  
        f'Found {negative_demand} "  
        f"negative demand rows"  
    )
```

```
# CHECK INVALID PRICES
```

```
invalid_prices = (  
    base["price"] <= 0  
) .sum()
```

```
if invalid_prices > 0:
```

```
    raise RuntimeError(  
        f'Found {invalid_prices} "  
        f"invalid price rows"  
    )
```

```
# FINAL RL DATASET
```

```
rl_df = base[  
  
    "date",  
  
    "product_id",  
  
    "ABC_class",
```

```
"category",  
  
"price",  
  
"units_sold",  
  
"lost_sales_units (για RL)",  
  
"real_demand",  
  
"qty_wasted",  
  
"waste_cost",  
  
"day_of_week",  
  
"event_flag",  
  
"shelf_life_days",  
  
"spoilage_rate",  
  
"event_uplift_signal",  
  
"service_criticality",  
  
"elasticity"  
]].copy()
```

```
# CLEAN COLUMN NAMES
```

```
rl_df = rl_df.rename(columns={  
  
    "lost_sales_units (για RL)": "lost_sales"  
})
```

```
# SORT DATA
```

```
rl_df = rl_df.sort_values(  
  
    ["product_id", "date"]  
  
).reset_index(drop=True)
```

```
# SAVE
```

```
output_path = ( "Multi-ABC Preprocessing and KPIs/rl_dataset.csv" )
```

```
rl_df.to_csv(  
  
    output_path,  
  
    index=False,  
  
    sep=";",  
  
    encoding="cp1253"  
)
```

```
# FINAL INFO
```

```
print("\n=====")
print("RL DATASET CREATED SUCCESSFULLY")
print("=====")
```

```
print(f"\nRows: {len(rl_df)}")
```

```
print(f"Products: {rl_df['product_id'].nunique()}")
```

```
print("\nElasticity values:")
print(sorted(rl_df["elasticity"].unique()))
```

```
print("\nABC distribution:")
print(
    rl_df["ABC_class"]
    .value_counts()
)
```

```
print("\nSaved to:")
print(output_path)
```

8.5. Ο κώδικας για το περιβάλλον του RL

```
# RL_Environment.py
```

```
import numpy as np
import pandas as pd

class SupermarketEnv:

    # INIT

    def __init__(self, df):

        self.df = df.reset_index(drop=True)

        self.n_steps = len(self.df)

    # ACTION SPACE

    # percentage price changes
    self.actions = [
        -0.30,
        -0.20,
        -0.10,
        -0.05,
        0.00,
        0.05,
        0.10,
        0.20
    ]
```

```
self.n_actions = len(self.actions)

# BUSINESS PARAMETERS

self.cost_ratio = 0.70

self.waste_penalty = 1.2

self.discount_penalty_factor = 0.15

# INVENTORY INITIALIZATION

avg_demand = self.df["real_demand"].mean()

self.initial_stock = max(
    20,
    int(round(avg_demand * 5))
)

self.max_stock = self.initial_stock

# STATE

self.current_step = 0

self.stock = self.initial_stock
```

```
# METRICS
```

```
self.total_profit = 0.0
```

```
self.total_revenue = 0.0
```

```
self.total_waste = 0.0
```

```
self.total_lost_sales = 0.0
```

```
# RESET
```

```
def reset(self):
```

```
    self.current_step = 0
```

```
    self.stock = self.initial_stock
```

```
    self.total_profit = 0.0
```

```
    self.total_revenue = 0.0
```

```
    self.total_waste = 0.0
```

```
    self.total_lost_sales = 0.0
```

```
    raw_state = self.get_raw_state()
```

```
    return self.encode_state(raw_state)
```

```
# RAW STATE
```

```
def get_raw_state(self):
```

```
    row = self.df.iloc[self.current_step]
```

```
    return {
```

```
        "price": float(row["price"]),
```

```
        "day_of_week": int(row["day_of_week"]),
```

```
        "event_flag": int(row["event_flag"]),
```

```
        "shelf_life_days": float(row["shelf_life_days"]),
```

```
        "spoilage_rate": float(row["spoilage_rate"]),
```

```
        "event_uplift": float(  
            row["event_uplift_signal"]  
        ),
```

```
        "elasticity": float(row["elasticity"]),
```

```
        "criticality": float(  
            row["service_criticality"]  
        ),
```

```
        "stock": float(self.stock)
```

```
    }
```

```
# STATE ENCODER
```

```
def encode_state(self, raw_state):
```

```
    # PRICE BUCKET
```

```
    price = raw_state["price"]
```

```
    if price < 2:
```

```
        price_b = 0
```

```
    elif price < 5:
```

```
        price_b = 1
```

```
    elif price < 9:
```

```
        price_b = 2
```

```
    else:
```

```
        price_b = 3
```

```
    # STOCK RATIO
```

```
    stock_ratio = raw_state["stock"] / self.max_stock
```

```
    if stock_ratio < 0.25:
```

```
        stock_b = 0
```

```
    elif stock_ratio < 0.50:
```

```
        stock_b = 1
```

```
    elif stock_ratio < 0.80:
```

```
        stock_b = 2
```

```
else:
    stock_b = 3

# SHELF LIFE

shelf = raw_state["shelf_life_days"]

if shelf <= 3:
    shelf_b = 0
elif shelf <= 10:
    shelf_b = 1
else:
    shelf_b = 2

# SPOILAGE

spoil = raw_state["spoilage_rate"]

if spoil < 0.01:
    spoil_b = 0
elif spoil < 0.02:
    spoil_b = 1
else:
    spoil_b = 2

# EVENT UPLIFT

uplift_signal = raw_state["event_uplift"]
```

```
if uplift_signal < 0.10:
    uplift_b = 0
elif uplift_signal < 0.30:
    uplift_b = 1
else:
    uplift_b = 2

# ELASTICITY

elasticity = abs(raw_state["elasticity"])

if elasticity < 0.7:
    elasticity_b = 0
elif elasticity < 1.2:
    elasticity_b = 1
else:
    elasticity_b = 2

# CRITICALITY

criticality = raw_state["criticality"]

if criticality < 0.30:
    criticality_b = 0
elif criticality < 0.70:
    criticality_b = 1
else:
    criticality_b = 2
```

```
# FINAL RL STATE
```

```
return (
```

```
    price_b,
```

```
    stock_b,
```

```
    shelf_b,
```

```
    spoil_b,
```

```
    int(raw_state["event_flag"]),
```

```
    uplift_b,
```

```
    elasticity_b,
```

```
    criticality_b,
```

```
    int(raw_state["day_of_week"])
```

```
)
```

```
def decode_state(self, state):
```

```
    price_b, stock_b, shelf_b, spoil_b, event_flag, uplift_b, elasticity_b, criticality_b,  
    dow = state
```

```
# PRICE (approximate midpoints)
```

```
price_map = {0: 1.0, 1: 3.5, 2: 7.0, 3: 12.0}
price = price_map[price_b]

# STOCK (convert bin → ratio → absolute)
stock_ratio_map = {0: 0.12, 1: 0.37, 2: 0.65, 3: 0.90}
stock = stock_ratio_map[stock_b] * self.max_stock

# SHELF LIFE
shelf_map = {0: 2, 1: 7, 2: 12}
shelf_life_days = shelf_map[shelf_b]

# SPOILAGE
spoil_map = {0: 0.005, 1: 0.02, 2: 0.05}
spoilage_rate = spoil_map[spoil_b]

# UPLIFT SIGNAL
uplift_map = {0: 0.05, 1: 0.15, 2: 0.35}
event_uplift = uplift_map[uplift_b]

# ELASTICITY
elasticity_map = {0: -0.5, 1: -1.0, 2: -1.5}
elasticity = elasticity_map[elasticity_b]

# CRITICALITY
criticality_map = {0: 0.15, 1: 0.50, 2: 0.85}
criticality = criticality_map[criticality_b]

return {
    "price": price,
    "stock": stock,
    "shelf_life_days": shelf_life_days,
```

```
"spoilage_rate": spoilage_rate,  
"event_flag": event_flag,  
"event_uplift": event_uplift,  
"elasticity": elasticity,  
"criticality": criticality,  
"day_of_week": dow  
}
```

```
# DEMAND SIMULATION
```

```
def simulate_demand(self, row, action):
```

```
    base_demand = row["real_demand"]
```

```
# EVENT EFFECT
```

```
if row["event_flag"] == 1:
```

```
    uplift_signal = max(  
        0.0,  
        row["event_uplift_signal"]  
    )
```

```
    uplift_signal = min(uplift_signal, 0.50)
```

```
    base_demand *= (1 + uplift_signal)
```

```
# PRICE ELASTICITY EFFECT
```

```
elasticity = row["elasticity"]

demand_multiplier = (
    1 + (elasticity * action)
)

# CLIPPING

demand_multiplier = np.clip(
    demand_multiplier,
    0.50,
    1.80
)

demand = base_demand * demand_multiplier

return max(0, demand)

# REWARD FUNCTION

def calculate_reward(
    self,
    revenue,
    profit,
    waste_cost,
    lost_profit_cost,
    action,
    criticality
):
```

```
# DISCOUNT PENALTY
discount_penalty = (
    max(0, -action)
    * revenue
    * self.discount_penalty_factor
)

reward = (
    profit
    - self.waste_penalty * waste_cost
    - lost_profit_cost * (1 + criticality)
    - discount_penalty
)

return reward

# REPLENISHMENT

def replenish_stock(self):

    # every 7 days
    if self.current_step % 7 == 0:

        target_stock = self.initial_stock

        replenishment = (
            target_stock - self.stock
        )

        replenishment = max(0, replenishment)
```

```
self.stock += replenishment

# STEP

def step(self, action_idx):

    row = self.df.iloc[self.current_step]

    # ACTION

    action = self.actions[action_idx]

    base_price = row["price"]

    new_price = base_price * (1 + action)

    # DEMAND

    demand = self.simulate_demand(
        row,
        action
    )

    # SALES

    units_sold = min(
```

```
self.stock,  
demand  
)  
  
lost_sales = max(  
    0,  
    demand - units_sold  
)  
  
self.stock -= units_sold  
  
# SPOILAGE / WASTE  
  
spoilage_rate = row["spoilage_rate"]  
  
waste = self.stock * spoilage_rate  
  
self.stock -= waste  
  
# integer inventory  
self.stock = max(  
    0,  
    int(round(self.stock))  
)  
  
# ECONOMICS  
  
revenue = units_sold * new_price
```

```
cost = (  
    self.cost_ratio  
    * base_price  
    * units_sold  
)  
  
profit = revenue - cost  
  
# LOST SALES COST  
unit_cost = self.cost_ratio * base_price  
  
lost_profit_per_unit = max(  
    0,  
    new_price - unit_cost  
)  
  
lost_profit_cost = (  
    lost_sales * lost_profit_per_unit  
)  
  
waste_cost = (  
    waste  
    * self.cost_ratio  
    * base_price  
)  
  
# REWARD  
  
reward = self.calculate_reward(
```

```
revenue=revenue,  
  
profit=profit,  
  
waste_cost=waste_cost,  
  
lost_profit_cost=lost_profit_cost,  
  
action=action,  
  
criticality=row[  
    "service_criticality"  
]  
)  
  
# METRICS UPDATE  
  
self.total_profit += profit  
  
self.total_revenue += revenue  
  
self.total_waste += waste  
  
self.total_lost_sales += lost_sales  
  
# NEXT STEP  
  
self.current_step += 1
```

```
done = self.current_step >= self.n_steps
```

```
# REPLENISHMENT
```

```
if not done:
```

```
    self.replenish_stock()
```

```
# NEXT STATE
```

```
if done:
```

```
    next_state = None
```

```
else:
```

```
    raw_state = self.get_raw_state()
```

```
    next_state = self.encode_state(  
        raw_state  
    )
```

```
info = {
```

```
    "profit": profit,
```

```
    "revenue": revenue,
```

```
    "waste": waste,
```

```
    "sales": units_sold,
```

```
    "lost_sales": lost_sales,
```

```
    "stockout": int(lost_sales > 0)
```

```
}
```

```
return next_state, reward, done , info
```

```
# METRICS
```

```
def get_metrics(self):
```

```
    return {
```

```
        "profit": self.total_profit,
```

```
        "revenue": self.total_revenue,
```

```
        "waste": self.total_waste,
```

```
        "lost_sales": self.total_lost_sales
```

```
    }
```

```
# CONFIG
```

```
def get_config(self):
```

```
    return {
```

```
        "actions": self.actions,
```

```
        "cost_ratio": self.cost_ratio,
```

```
        "waste_penalty": self.waste_penalty,
```

```
        "discount_penalty_factor": self.discount_penalty_factor,
```

```
        "initial_stock": self.initial_stock
```

```
    }
```

8.6. Ο κώδικας για την εκπαίδευση του πράκτορα

```
# Train_RL.py

import os
import random
from datetime import datetime
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
from collections import defaultdict
from RL_Environment import SupermarketEnv

# EXPERIMENT SETUP

experiment_name = "exp_clean_qlearning"

timestamp = datetime.now().strftime("%d%m%Y_%H%M%S")

output_dir = f"experiments/{experiment_name}_{timestamp}"

os.makedirs(output_dir, exist_ok=True)

# RANDOM SEEDS

seeds = [1, 7, 21, 42, 99]
```

```
# LOAD RL DATASET
```

```
df = pd.read_csv(  
    "Multi-ABC Preprocessing and KPIs/rl_dataset.csv",  
    sep=";",  
    encoding="cp1253"  
)
```

```
# KEEP ONLY A PRODUCTS
```

```
df_A = df[df["ABC_class"] == "A"].copy()
```

```
A_products = sorted(  
    df_A["product_id"].unique()  
)
```

```
print("\n=====")  
print("A PRODUCTS")  
print("=====")
```

```
print(f"Total A products: {len(A_products)}")
```

```
# Q-LEARNING PARAMETERS
```

```
alpha = 0.10
```

```
gamma = 0.95
```

```
epsilon_start = 1.0

epsilon_min = 0.05

episodes = 2200

# SAVE CONFIG

tmp_env = SupermarketEnv(
    df_A[df_A["product_id"] == A_products[0]]
)

with open(
    os.path.join(output_dir, "config.txt"),
    "w"
) as f:

    f.write("=== RL CONFIG ===\n")

    f.write(f'episodes={episodes}\n')

    f.write(f'alpha={alpha}\n')

    f.write(f'gamma={gamma}\n')

    f.write(f'epsilon_start={epsilon_start}\n')

    f.write(f'epsilon_min={epsilon_min}\n')

    f.write(f'seeds={seeds}\n')
```

```
f.write(f'actions={tmp_env.actions}\n')
```

```
f.write(f'ΣΗΜΕΙΩΣΕΙΣ:  ')
```

```
# HEURISTIC POLICY
```

```
def heuristic_policy(raw_state, env):
```

```
    stock = raw_state["stock"]
```

```
    shelf = raw_state["shelf_life_days"]
```

```
    spoilage = raw_state["spoilage_rate"]
```

```
    event_flag = raw_state["event_flag"]
```

```
    uplift = raw_state["event_uplift"]
```

```
    criticality = raw_state["criticality"]
```

```
# SHORT SHELF LIFE
```

```
if shelf <= 2:
```

```
    return env.actions.index(-0.30)
```

```
# HIGH SPOILAGE
```

```
if spoilage > 0.05:
```

```
    return env.actions.index(-0.20)
```

```
# EVENT DEMAND
```

```
if event_flag == 1 and uplift > 0.15:
```

```
    return env.actions.index(0.10)
```

```
# LOW STOCK
```

```
if stock < 0.25 * env.max_stock:
```

```
    return env.actions.index(0.20)
```

```
# CRITICAL PRODUCTS
```

```
if criticality > 0.7:
```

```
    return env.actions.index(0.00)
```

```
return env.actions.index(0.00)
```

```
# RESULTS
```

```
all_results = []
```

```
# MULTI-SEED TRAINING
```

```
for seed in seeds:
```

```
    print("\n=====")
    print(f"RUNNING SEED {seed}")
    print("=====")
```

```
    random.seed(seed)
```

```
    np.random.seed(seed)
```

```
    seed_results = []
```

```
# LOOP PRODUCTS
```

```
for product_id in A_products:
```

```
    print(f"\nTraining SKU: {product_id}")
```

```
    sku_df = df_A[
        df_A["product_id"] == product_id
    ].reset_index(drop=True)
```

```
# ENVIRONMENT
```

```
env = SupermarketEnv(sku_df)

# Q-TABLE

Q = defaultdict(float)

# TRAINING METRICS

episode_rewards = []

# TRAINING LOOP

for ep in range(episodes):

    # EXPONENTIAL EPSILON DECAY

    epsilon = max(
        epsilon_min,
        epsilon_start * np.exp(-0.0015 * ep)
    )

    state = env.reset()

    done = False

    total_reward = 0
```

while not done:

```
# EPSILON-GREEDY
```

```
if np.random.rand() < epsilon:
```

```
    action = np.random.choice(
        env.n_actions
    )
```

```
else:
```

```
    qs = [
        Q[(state, a)]
        for a in range(env.n_actions)
    ]
```

```
    action = int(np.argmax(qs))
```

```
# ENV STEP
```

```
next_state, reward, done, info = env.step(
    action
)
```

```
total_reward += reward
```

```
# FUTURE Q
```

```
if next_state is None:
```

```
future_q = 0

else:

    future_q = max([
        Q[(next_state, a)]
        for a in range(env.n_actions)
    ])

# Q UPDATE

old_q = Q[(state, action)]

new_q = old_q + alpha * (reward + gamma * future_q - old_q)

Q[(state, action)] = new_q

state = next_state

episode_rewards.append(total_reward)

# LEARNING CURVE

smoothed = (
    pd.Series(episode_rewards)
    .rolling(100, min_periods=1)
    .mean()
)
```

```
plt.figure(figsize=(8, 5))

plt.plot(
    episode_rewards,
    alpha=0.3,
    label="raw"
)

plt.plot(
    smoothed,
    linewidth=2,
    label="smoothed"
)

plt.xlabel("Episode")

plt.ylabel("Reward")

plt.title(
    f"Learning Curve - {product_id}"
)

plt.legend()

plt.tight_layout()

plt.savefig(
    os.path.join(
        output_dir,
        f"learning_curve_{product_id}_seed_{seed}.png"
    )
)
```

)

```
plt.close()
```

```
# RL EVALUATION
```

```
env = SupermarketEnv(sku_df)
```

```
state = env.reset()
```

```
done = False
```

```
rl_reward = 0
```

```
while not done:
```

```
    qs = [  
        Q[(state, a)]  
        for a in range(env.n_actions)  
    ]
```

```
    action = int(np.argmax(qs))
```

```
    next_state, reward, done, info = env.step(action)
```

```
    rl_reward += reward
```

```
    state = next_state
```

```
rl_metrics = env.get_metrics()
```

```
# HEURISTIC BASELINE
```

```
env = SupermarketEnv(sku_df)
```

```
env.reset()
```

```
done = False
```

```
heuristic_reward = 0
```

```
while not done:
```

```
    raw_state = env.get_raw_state()
```

```
    action = heuristic_policy(  
        raw_state,  
        env  
    )
```

```
    next_state, reward, done, info = env.step(  
        action  
    )
```

```
    heuristic_reward += reward
```

```
heuristic_metrics = env.get_metrics()
```

```
# IMPROVEMENT
```

```
improvement = (  
    rl_reward - heuristic_reward  
)
```

```
improvement_pct = (  
  
    improvement  
  
    /  
  
    (abs(heuristic_reward) + 1e-6)  
  
    ) * 100
```

```
print(f"RL reward: {rl_reward:.2f}")
```

```
print(  
    f"Heuristic reward: "  
    f"{heuristic_reward:.2f}"  
)
```

```
print(  
    f"Improvement: "  
    f"{improvement_pct:.2f}%"  
)
```

```
# SAVE RESULTS
```

```
seed_results.append({

    "seed": seed,

    "product_id": product_id,

    "rl_reward": rl_reward,

    "heuristic_reward": heuristic_reward,

    "improvement": improvement,

    "improvement_pct": improvement_pct,

    # RL metrics
    "rl_profit":
        rl_metrics["profit"],

    "rl_revenue":
        rl_metrics["revenue"],

    "rl_waste":
        rl_metrics["waste"],

    "rl_lost_sales":
        rl_metrics["lost_sales"],

    # Baseline metrics
    "heur_profit":
        heuristic_metrics["profit"],
```

```
"heur_revenue":  
    heuristic_metrics["revenue"],  
  
"heur_waste":  
    heuristic_metrics["waste"],  
  
"heur_lost_sales":  
    heuristic_metrics["lost_sales"]  
})
```

```
# VISUALIZATION
```

```
# LABELS
```

```
stock_labels = {  
    0: "<25% Stock",  
    1: "25-50% Stock",  
    2: "50-80% Stock",  
    3: ">80% Stock"  
}
```

```
shelf_labels = {  
    0: "≤3 Days",  
    1: "4-10 Days",  
    2: ">10 Days"  
}
```

```
# DISCRETE COLORMAP FOR 8 ACTIONS
```

```
action_colors = [
```

```
"#8b0000", # -30%
"#b22222", # -20%
"#dc143c", # -10%
"#ff6347", # -5%
"#ffd700", # 0%
"#9acd32", # +5%
"#32cd32", # +10%
"#006400" # +20%
]

cmap = mcolors.ListedColormap(action_colors)

visited_states = list({s for (s, a) in Q.keys()})

df_states = pd.DataFrame(visited_states, columns=[
    "price_bin",
    "stock_bin",
    "shelf_bin",
    "spoilage_bin",
    "event_flag",
    "uplift_bin",
    "elasticity_bin",
    "criticality_bin",
    "dow_bin"
])

stock_values = sorted(df_states["stock_bin"].unique())
shelf_values = sorted(df_states["shelf_bin"].unique())
```

```
rl_heatmap = []
baseline_heatmap = []
visit_heatmap = []

for shelf in shelf_values:
    rl_row = []
    base_row = []
    visit_row = []

    for stock in stock_values:

        subset = df_states[
            (df_states["stock_bin"] == stock) &
            (df_states["shelf_bin"] == shelf)
        ]

        # visit count
        visit_row.append(len(subset))

        if subset.empty:
            rl_row.append(None)
            base_row.append(None)
            continue

        s = tuple(subset.iloc[0].values)

        # RL best action (index)
        qs = [Q[(s, a)] for a in range(env.n_actions)]
        best_action_idx = int(np.argmax(qs))
        rl_row.append(best_action_idx)
```

```
# Baseline best action (index)
raw_state = env.decode_state(s)
base_best_idx = heuristic_policy(raw_state, env)
base_row.append(base_best_idx)

rl_heatmap.append(rl_row)
baseline_heatmap.append(base_row)
visit_heatmap.append(visit_row)

# 8. SIDE-BY-SIDE RL vs BASELINE

fig, axes = plt.subplots(1, 3, figsize=(20, 6))

# RL POLICY HEATMAP (action indices)

im1 = axes[0].imshow(rl_heatmap, cmap=cmap, aspect="auto", vmin=0, vmax=7)
axes[0].set_title("RL Policy")
axes[0].set_xlabel("Stock Bin")
axes[0].set_ylabel("Shelf Bin")
axes[0].set_xticks(range(len(stock_values)))
axes[0].set_xticklabels([stock_labels[x] for x in stock_values], rotation=45)
axes[0].set_yticks(range(len(shelf_values)))
axes[0].set_yticklabels([shelf_labels[y] for y in shelf_values])
fig.colorbar(im1, ax=axes[0])

# annotations
for i in range(len(shelf_values)):
    for j in range(len(stock_values)):
        val = rl_heatmap[i][j]
```

```
if val is not None:
    axes[0].text(j, i, f'{env.actions[val]*100:+.0f}%', ha="center", va="center",
fontsize=7)

# contour lines
# contour only if grid is at least 2x2
if len(shelf_values) >= 2 and len(stock_values) >= 2:
    axes[0].contour(rl_heatmap, levels=range(8), colors="black", linewidths=0.3)

# BASELINE POLICY HEATMAP (action indices)

im2 = axes[1].imshow(baseline_heatmap, cmap=cmap, aspect="auto", vmin=0,
vmax=7)
axes[1].set_title("Baseline Policy")
axes[1].set_xlabel("Stock Bin")
axes[1].set_ylabel("Shelf Bin")
axes[1].set_xticks(range(len(stock_values)))
axes[1].set_xticklabels([stock_labels[x] for x in stock_values], rotation=45)
axes[1].set_yticks(range(len(shelf_values)))
axes[1].set_yticklabels([shelf_labels[y] for y in shelf_values])
fig.colorbar(im2, ax=axes[1])

# annotations baseline
for i in range(len(shelf_values)):
    for j in range(len(stock_values)):
        val = baseline_heatmap[i][j]

        if val is not None:
            axes[1].text(
                j,
```

```
        i,  
        f'{env.actions[val]*100:+.0f}%',  
        ha="center",  
        va="center",  
        fontsize=7  
    )  
  
# STATE VISITATION HEATMAP  
  
im3 = axes[2].imshow(visit_heatmap, cmap="Blues", aspect="auto")  
axes[2].set_title("State Visitation Frequency")  
axes[2].set_xlabel("Stock Bin")  
axes[2].set_ylabel("Shelf Bin")  
axes[2].set_xticks(range(len(stock_values)))  
axes[2].set_xticklabels([stock_labels[x] for x in stock_values], rotation=45)  
axes[2].set_yticks(range(len(shelf_values)))  
axes[2].set_yticklabels([shelf_labels[y] for y in shelf_values])  
fig.colorbar(im3, ax=axes[2])  
  
plt.tight_layout()  
plt.savefig(os.path.join(output_dir,  
f'advanced_policy_{product_id}_seed_{seed}.png'))  
plt.close()  
  
# SEED RESULTS  
  
seed_df = pd.DataFrame(seed_results)  
  
all_results.append(seed_df)
```

```
# FINAL RESULTS
```

```
final_results = pd.concat(  
    all_results,  
    ignore_index=True  
)
```

```
# BUSINESS KPI PERCENTAGES
```

```
# SYMMETRIC IMPROVEMENT INDEX
```

```
# Profit improvement %
```

```
final_results["profit_improvement_pct"] = (  
    2 * (  
        final_results["rl_profit"]  
        - final_results["heur_profit"]  
    )  
    /  
    (  
        final_results["rl_profit"].abs()  
        + final_results["heur_profit"].abs()  
        + 1e-6  
    )  
) * 100
```

```
# Revenue improvement %
```

```
final_results["revenue_improvement_pct"] = (  
    2 * (  
        final_results["rl_revenue"]
```

```
- final_results["heur_revenue"]  
)  
/  
(  
    final_results["rl_revenue"].abs()  
    + final_results["heur_revenue"].abs()  
    + 1e-6  
)  
) * 100
```

Waste reduction %

```
final_results["waste_reduction_pct"] = (  
    2 * (  

```

```
        final_results["rl_waste"]  
        - final_results["heur_waste"]  
    )  
    /  
    (  
        final_results["rl_waste"].abs()  
        + final_results["heur_waste"].abs()  
        + 1e-6  
    )  
) * 100
```

Lost sales reduction %

```
final_results["lost_sales_reduction_pct"] = (  
    2 * (  

```

```
        final_results["rl_lost_sales"]  
        - final_results["heur_lost_sales"]  

```

```
)  
/  
(  
    final_results["rl_lost_sales"].abs()  
    + final_results["heur_lost_sales"].abs()  
    + 1e-6  
)  
) * 100  
  
# CLEAN BUSINESS SUMMARY  
  
business_summary = (  
    final_results  
    .groupby("product_id")  
    .agg({  
  
        "profit_improvement_pct": ["mean", "std"],  
        "revenue_improvement_pct": ["mean", "std"],  
        "waste_reduction_pct": ["mean", "std"],  
        "lost_sales_reduction_pct": ["mean", "std"]  
  
    })  
)  
  
# flatten column names  
business_summary.columns = [  
    "_".join(col)  
    for col in business_summary.columns  
]
```

```
business_summary = business_summary.reset_index()
```

```
# CI95 CALCULATION
```

```
num_seeds = len(seeds)
```

```
business_summary["profit_ci95"] = (  
    1.96  
    * business_summary["profit_improvement_pct_std"]  
    / np.sqrt(num_seeds)  
)
```

```
business_summary["revenue_ci95"] = (  
    1.96  
    * business_summary["revenue_improvement_pct_std"]  
    / np.sqrt(num_seeds)  
)
```

```
business_summary["waste_ci95"] = (  
    1.96  
    * business_summary["waste_reduction_pct_std"]  
    / np.sqrt(num_seeds)  
)
```

```
business_summary["lost_sales_ci95"] = (  
    1.96  
    * business_summary["lost_sales_reduction_pct_std"]  
    / np.sqrt(num_seeds)
```

)

```
# ROUND VALUES
```

```
business_summary = business_summary.round(2)
```

```
# SAVE CLEAN EXCEL
```

```
business_summary.to_csv(  
    os.path.join(output_dir, "business_summary.csv"),  
    index=False,  
    sep=";",  
    decimal=".",  
    encoding="cp1253"  
)
```

```
print("\nBusiness summary saved.")
```

```
# SUMMARY STATISTICS
```

```
summary_stats = (  
    final_results  
    .groupby("product_id")  
    ["improvement"]  
    .agg(["mean", "std"])  
    .reset_index()  
)
```

```
summary_stats["ci95"] = 1.96 * (  
    summary_stats["std"] / np.sqrt(len(seeds))
```

)

```
summary_stats = summary_stats.round(2)
summary_stats.to_csv(
    os.path.join(output_dir, "summary_statistics.csv"),
    index=False,
    sep=";",
    decimal=".",
    encoding="cp1253"
)
```

GLOBAL SUMMARY

```
global_summary = {
    "mean_improvement": final_results["improvement"].mean(),
    "std_improvement": final_results["improvement"].std(),
    "positive_ratio": (
        final_results["improvement"] > 0
    ).mean(),
    "max_improvement": final_results["improvement"].max(),
    "min_improvement": final_results["improvement"].min()
}
```

```
global_summary_df = pd.DataFrame(
    [global_summary]
).round(2)
```

```
global_summary_df.to_csv(
    os.path.join(output_dir, "global_summary.csv"),
    index=False,
```

```
sep=";",  
decimal="",  
encoding="cp1253"  
)  
  
# SKU IMPROVEMENT CHART  
  
sku_improvement = (  
    final_results  
    .groupby("product_id")["improvement"]  
    .mean()  
    .reset_index()  
)  
  
sku_improvement = sku_improvement.sort_values(  
    "improvement",  
    ascending=False  
)  
  
colors = [  
    "green" if x > 0 else "red"  
    for x in sku_improvement["improvement"]  
]  
  
plt.figure(figsize=(10, 8))  
  
plt.barh(  
    sku_improvement["product_id"],  
    sku_improvement["improvement"],  
    color=colors  
)
```

```
for i, value in enumerate(  
    sku_improvement["improvement"]  
):  
    plt.text(  
        value,  
        i,  
        f'{value:.0f}',  
        va="center"  
    )  
  
plt.gca().invert_yaxis()  
  
plt.xlabel("Average Improvement")  
plt.ylabel("SKU")  
  
plt.title(  
    "Improvement per SKU (RL vs Heuristic)"  
)  
  
plt.grid(  
    axis="x",  
    alpha=0.3  
)  
  
plt.tight_layout()  
  
plt.savefig(  
    os.path.join(  
        output_dir,  
        "improvement_per_sku.png"
```

```
),  
    dpi=300  
)  
  
plt.close()  
  
# 13. FINAL VISUALIZATIONS  
  
# Histogram  
plt.figure(figsize=(8, 5))  
  
plt.hist(final_results["improvement"], bins=15)  
  
plt.title("Improvement Distribution")  
plt.xlabel("Improvement")  
plt.ylabel("Count")  
  
plt.grid(alpha=0.3)  
  
plt.savefig(  
    os.path.join(output_dir, "hist_improvement.png")  
)  
  
plt.close()  
  
# RL vs Heuristic Scatter  
plt.figure(figsize=(7, 7))  
  
plt.scatter(  
    final_results["heuristic_reward"],  
    final_results["rl_reward"]
```

```
)  
min_val = min(  
    final_results["heuristic_reward"].min(),  
    final_results["rl_reward"].min()  
)  
  
max_val = max(  
    final_results["heuristic_reward"].max(),  
    final_results["rl_reward"].max()  
)  
  
plt.plot(  
    [min_val, max_val],  
    [min_val, max_val],  
    "r--",  
    linewidth=2,  
    label="x = y"  
)  
  
plt.legend()  
plt.xlabel("Heuristic Reward")  
plt.ylabel("RL Reward")  
plt.title("RL vs Heuristic")  
  
plt.savefig(  
    os.path.join(output_dir, "scatter_rl_vs_heuristic.png")  
)  
  
plt.close()
```

```
# SAVE RESULTS
```

```
results_path = os.path.join(  
    output_dir,  
    "all_results.csv"  
)
```

```
final_results_export = (  
    final_results.copy()  
    .round(2)  
)
```

```
final_results_export.to_csv(  
    results_path,  
    index=False,  
    sep=";",  
    decimal=".",  
    encoding="cp1253"  
)
```

```
# FINAL VISUALIZATION
```

```
plt.figure(figsize=(8, 5))
```

```
plt.hist(  
    final_results["improvement_pct"],  
    bins=20  
)
```

```
plt.xlabel("Improvement %")
```

```
plt.ylabel("Frequency")

plt.title(
    "RL Improvement Distribution"
)

plt.tight_layout()

plt.savefig(
    os.path.join(
        output_dir,
        "improvement_distribution.png"
    )
)

plt.close()

# FINISHED

print("\n=====")
print("EXPERIMENT COMPLETED")
print("=====")

print("\nResults saved in:")

print(output_dir)
```

8.7. Τελικά αποτελέσματα του πράκτορα

Δημήτριος Πάλλας, Δυναμική Τιμολόγηση με Ελαχιστοποίηση της Σπατάλης Τροφίμων μέσω Πρόβλεψης της Ζήτησης, ABC Ταξινόμησης και Ενισχυτικής Μάθησης

Σε αυτή την υποενότητα παρατίθενται τα τελικά αποτελέσματα που προέκυψαν από το πείραμα που διεξήχθη (Εικόνες 8.1, 8.2, 8.3, 8.4, 8.5 και 8.6).

i	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
seed	product_id	ri_reward	heuristic_reward	improvement	improvement_pct	ri_profit	ri_revenue	ri_waste	ri_lost_sales	heuristic_profit	heuristic_revenue	heuristic_waste	heuristic_lost_sales	profit_improvement_pct	revenue_improvement_pct	waste_reduction_pct	lost_sales_reduction_pct	
1	P002	28400.25	-2562.77	30963.02	1208.19	44141.12	122540.11	1878.14	7005.17	38081.33	118075.08	1485.15	5005.35	14.74	2.87	23.37	32.69	
3	P005	26245.9	-1907.36	28153.26	1476.05	39284.42	112992.52	1804.99	8039.48	35736.93	111350.89	1475.04	4909.88	9.46	1.46	20.12	48.34	
4	P003	18238.52	-432.23	18670.75	4319.67	26401.49	71615.05	1005.87	3951.39	21900.1	68865.83	805.1	3084.91	18.64	3.91	27.87	24.13	
5	P006	11657.83	-5491.28	18149.11	330.51	21807.78	60092.78	1080.82	3661.85	0	41125.8	575.02	9314.53	200	37.48	61.09	47.06	
6	P007	15099.06	-857.01	21671.07	252.81	20837.47	59472.61	674.52	2631.28	5882.89	47063.13	414.63	4586.4	111.93	23.3	47.72	-54.18	
7	P008	35775.23	-12525.71	48300.94	585.61	55172.31	149321.66	1610.4	5615.06	0	100895.04	830.39	14957.76	200	38.71	63.91	-90.83	
8	P009	21393.81	-4482.16	29875.97	352.22	30958.89	86589.52	1420.41	5806.17	26808.58	84118.46	1113.85	4033.78	14.37	2.9	24.19	36.02	
9	P010	15955.42	-1308.74	17264.16	1319.14	26057.27	73708.8	743.5	3054.38	22950.56	72265.34	587.95	2091.49	12.68	1.97	23.07	37.42	
10	P011	15317.44	-5669.42	20986.85	370.18	23209.61	62133.41	951.49	2947.56	0	42367.04	480.05	7739.95	200	37.83	65.86	-89.68	
11	P012	25647.84	-11635.56	37283.4	320.43	38586.48	111195.85	1617.56	6913.13	35265.16	109954.71	1289.21	4324.9	8.99	1.12	22.59	46.06	
12	P015	21826.65	-1218.06	23044.71	1891.92	31802.05	87205.7	524.56	5688.72	27076.64	84472.4	1176.89	4140.51	16.05	3.18	25.74	31.15	
13	P016	11224.77	-4612.98	15837.74	343.38	17683.46	52666.11	807.86	3961.8	0	36874.52	463.26	8175.04	200	35.27	54.21	-69.43	
14	P017	7878.71	-6566.11	14444.82	219.99	12074.23	33962.85	493.89	2071.6	3341.28	26730.25	305.78	3631.45	113.3	23.83	47.05	-54.7	
15	P018	24712.85	-2282.5	26995.36	1182.71	36415.68	98636.26	1338.14	4644.09	30333.13	93368.92	994.13	3507.33	18.23	3.37	29.5	27.89	
16	P019	8604.21	-9262.44	17866.65	199.89	21394.71	61379.21	957.95	3076.95	6182.05	49456.4	626.06	4801.88	110.33	21.51	41.91	-43.79	
17	P020	6558.93	-11764.22	18323.15	155.75	17762.08	52834.24	1377.01	4698.41	5445.53	43564.26	898.46	6625.7	106.14	19.23	42.06	-34.04	
18	P021	8721.89	-11741.16	20463.04	174.28	24583.88	70683.51	1535.01	5010.26	7181.66	57453.25	1004.41	7420.44	109.57	20.65	41.99	-38.78	
19	P022	12652.09	-15329.78	27981.87	182.53	33616.41	100763.91	2002.44	7050.15	0	71440.71	1194.89	11626.57	200	30.44	50.51	-49.01	
20	P023	1786.99	-9921.08	17108.07	174.46	23115.94	67647.48	574.09	2749.09	48130.64	501.89	500.66	51.8	10.73	51.54	-58.16	-54.13	
21	P024	12660.35	-17247.44	29907.79	173.4	39461.93	116909.17	1982.82	6888.95	0	84813.82	1196.7	1200.2	200	31.82	49.45	-54.13	
22	P025	8989.16	-8620.08	17609.24	204.28	22416.16	64183.15	1360.93	4414.93	0	45494.12	806.31	8617.77	200	34.08	51.18	-64.5	
23	P026	10311.49	-1527.59	25589.08	167.49	39660.43	107759.55	1951.66	5006.15	0	75785.07	1107.17	11278.06	200	34.84	55.26	-77.03	
24	P027	11551.49	-13644.74	25196.22	184.66	30513.58	87015.74	1778.78	3907.35	8768.44	70147.51	819.46	6162.86	110.71	21.47	43.86	-44.8	
25	P028	13022.29	-18837.38	32859.67	165.65	31926.8	80953.47	1600.33	5090.55	9228.24	73825.91	1017.49	7583.67	110.31	20.79	44.53	-39.34	
26	P116	8765.14	-5663.36	14428.5	254.77	15249.27	42024.36	3195.89	11756.38	3734.94	29879.53	2355.49	16267.56	121.3	29.46	30.28	-32.2	
27	P117	1828.34	-980.02	2808.36	286.56	3601.19	9187.26	1134.77	3745.67	845.33	6762.68	856.22	5607.45	123.96	31.25	27.98	-39.91	

Εικόνα 8.1 RL all results seed 1

i	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
seed	product_id	ri_reward	heuristic_reward	improvement	improvement_pct	ri_profit	ri_revenue	ri_waste	ri_lost_sales	heuristic_profit	heuristic_revenue	heuristic_waste	heuristic_lost_sales	profit_improvement_pct	revenue_improvement_pct	waste_reduction_pct	lost_sales_reduction_pct	
28	P 002	21477.51	-2562.77	30040.27	1172.18	43582.59	122147.48	1875.7	7177.97	38081.33	118075.08	1485.15	5005.35	13.47	2.55	23.13	35.09	
29	P0562	19070.36	-1907.36	28470.01	1492.64	39533.01	112959.93	1843.7	7844.09	35736.93	111350.89	1475.04	4909.88	10.09	1.43	22.22	46.04	
30	P005	18466.03	-432.23	18898.26	4372.31	26244.61	71688.9	1069.64	3803.38	21900.1	68865.83	805.1	3084.91	19.51	4.01	28.22	20.86	
31	P006	14415.94	-5491.28	19907.22	362.52	22004.18	60106.07	1109.64	3782.35	0	41125.8	575.02	9314.53	200	37.5	63.47	-84.48	
32	P007	13206.89	-857.01	21778.9	254.07	20853.84	59441.04	674.17	2638.89	5882.89	47063.13	414.63	4586.4	111.99	23.24	47.68	-53.91	
33	P008	35539.65	-12525.71	48065.36	583.73	51747.44	146594.96	1540.44	6600.8	0	100895.04	830.39	14957.76	200	36.93	59.9	-77.53	
34	P009	21100.65	-4482.16	29581.81	349.97	30815.38	86599.52	1437.33	5782.28	26808.58	84118.46	1113.85	4033.78	13.94	2.56	25.66	35.63	
35	P010	18936.62	-1308.74	20245.36	1546.93	21786.64	74361.3	763.87	2888.9	22950.56	72265.34	587.95	2091.49	16.9	2.86	26.03	32.02	
36	P011	14898.31	-5669.42	20567.73	362.78	25912.81	61953.75	913.39	3190.14	0	42367.04	480.05	7739.95	200	37.55	62.2	-83.25	
37	P012	27066.05	-12182.5	38701.61	332.61	40792	122370.19	1680.06	6345.12	95255.16	109954.71	1289.21	4324.9	14.53	2.71	26.53	54.74	
38	P015	18639.95	-1218.06	19980.01	1630.3	28376.84	84489.97	1420.74	7118.02	27076.64	84472.4	1176.89	4140.51	4.69	0.02	18.77	52.89	
39	P016	13134.18	-4612.97	17747.15	384.72	19189.94	53616.46	882.21	3377.23	0	36874.52	463.26	8175.04	200	37	62.28	-53.86	
40	P017	7461.46	-6566.11	14027.57	213.64	11485.11	33569.95	474.97	2210.74	3341.28	26730.25	305.78	3631.45	109.86	22.69	43.34	-48.64	
41	P018	20222.21	-2282.5	22504.72	985.97	35245.81	98408.08	1240.32	4876.65	30333.13	93368.92	994.13	3507.33	15.02	3.14	22.37	31.4	
42	P019	7723.75	-9262.44	16988.12	183.39	20253.3	60496.97	921.65	3311.64	6182.05	49456.4	626.06	4801.88	106.46	20.08	38.2	-36.73	
43	P020	6792.66	-11764.22	18556.88	157.74	17924.05	52904.62	1384.48	4660.96	5445.53	43564.26	898.46	6625.7	106.79	19.36	42.58	-34.82	
44	P021	8764.27	-11741.16	20505.42	174.65	23598.72	70647.54	1520.56	5246.45	7181.66	57453.25	1004.41	7420.44	106.06	18.93	41.45	-32.49	
45	P022	13053.38	-15329.78	28365.12	185.02	34630.63	101442.09	2039.19	6695.69	0	71410.71	1194.89	11626.57	200	31.1	52.91	-53.85	
46	P023	9220.05	-1907.36	19101.68	192.93	22711.85	67334.88	843.11	2932.1	0	49138.48	501.99	5005.68	200	31.25	50.72	-52.21	
47	P024	16126.83	-17247.44	33374.27	193.5	40457.39	116948.78	2084.98	6827.82	0	84813.82	1196.7	12002	200	31.85	54.14	-54.96	
48	P025	9195.25	-8620.08	17815.33	206.67	21655.53	64209.78	1385.08	4337.96	0	45494.12	806.31	8617.77	200	34.12	52.82	-66.07	
49	P026	13470.67	-1527.59	28748.26	188.17	38111.69	106053.05	1968.91	5632.14	0	75785.07	1107.17	11278.06	200	33.29	56.03	-66.78	
50	P027	11203.4	-13644.74	24848.14	182.11	27594.26	84954.44	1187.92	4548.87	8768.44	70154.1	819.46	6162.86	103.54	19.07	36.71	-30.13	
51	P028	8138.83	-18837.38	27976.21	141.05	33557.14	92551.79	1591.92	4540.97	9228.24	73825.91	1017.49	7587.67	113.52	22.51	44.03	-50.19	
52	P116	8955.5	-5663.36	14618.86	258.13	14703.77	39713.34	3154.67	12330.05	3734.94	29879.53	2355.49	16267.56	118.88	28.26	29.01	-28.15	
53	P117	2077.22	-980.02	3057.23	311.96	3494.36	9165.88	1136.1	4016.72	845.33	6762.68	856.22	5607.45	122.08	30.17	28.1	-33.06	

Εικόνα 8.2 RL all results seed 7

i	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
seed	product_id	ri_reward	heuristic_reward	improvement	improvement_pct	ri_profit	ri_revenue	ri_waste	ri_lost_sales	heuristic_profit	heuristic_revenue	heuristic_waste	heuristic_lost_sales	profit_improvement_pct	revenue_improvement_pct	waste_reduction_pct	lost_sales_reduction_pct	
54	P002	28333.06	-2562.77	30895.82	1205.57	42851.39	121034.86	1893.46	7433.27	38081.33	118075.08	1485.15	5005.35	11.79	1.63	24.17	38.46	
55	P003	26885.47	-1907.36	28793.84	1509.56	41177.4	114332.72	1897.37	7117.9	35736.93	111350.89	1475.04	4909.88	14.15	2.63	25.05	36.72	
56	P005	17934.16	-432.23	18366.39	4249.26	26436.16	71684.35	1025.86	3862.38	21800.1	68865.83	805.1	3084.91	11.79	1.63	24.17	38.46	
57	P008	13855.84	-5491.28	19347.12	350.32	21662.66	94787.07	1094.16	3805.59	0	41125.8	575.02	9314.53	200	36.94	62.2	64.02	
58	P008	12562.7	-21236.79	33804.494	5092.62	247.74	2796.484	5092.62	247.74	5882.8	47028.13	214.643	63.84	200	36.94	62.2	64.02	
59	P008	26195.02	-13252.71	39720.73	393.13	48322.05	159781.11	1428.97	7219.77	0	100805.04	830.39	4957.76	200	58.3	53.03	48.54	
60	P009	21843.59	-8482.16	29726.09	350.5	31225.49	86516.93	1478.56	5616.1	26808.58	84118.46	1113.85	4033.78	15.22	2.81	26.75	39.78	
61	P010	12245.87	-1308.74	20157.31	1540.2	27200.14	74139.34	1780.67	2851.87	22950.56	72265.54	587.95	2093.49	16.96	2.86	26.75	30.2	
62	P010	15960.42	-5660.42	21620.84	321.18	20797.62	60847.1	3659.54	0	0	43387.28	500	2500	200	58.3	53.03	48.54	
63	P012	26178.67	-12163.56	37814.23	349.9	41326.22	113395.58	1682.79	5947.33	35265.16	100954.71	1289.21	4524.9	15.83	3.08	26.89	31.58	
64	P015	2411.97	-16308.28	22689.57	886.77	31763.89	87299.32	1581.79	5764.74	20766.64	84472.4	1767.89	4104.51	15.83	3.29	25.43	32.59	
65	P015	11895.45	-1471.97	13366.46	3735.55	17828.66	54407.46	3735.55	17828.66	0	3687.52	83.25	21.54	200	58.3	53.03	48.54	
66	P017	8105.41	-6565.11	14671.52	223.44	12401.52	34151.72	126.26	1945.2	3341.28	26730.25	305.78	3631.45	11.51	2.48	24.88	24.82	
67	P018	24145.38	-8286.12	26427.88	1157.85	35485.22	93630.1	1033.47	4977.23	30333.13	93360.92	994.13	3507.33	11.51	2.48	24.88	60.45	
68	P019	8206.68	-9262.44	17469.12	18.8	20175.94	60033.44	913.57	3361.57	6182.05	40958.6	626.06	4801.88	106.18	20.26	37.55	-35.26	
69	P019	6339.38	-11746.22	18023.46	155.59	17427.41	52038.17	155.59	17427.41	4356.16	54545.93	602.61	602.61	10.77	1.07	40.44	-50.7	
70	P021	8105.41	-8286.12	26427.88	1157.85	35485.22	93630.1	1033.47	4977.23	30333.13	93360.92	994.13	3507.33	11.51	2.48	24.88	60.45	
71	P021	8105.41	-8286.12	26427.88	1157.85	35485.22	93630.1	1033.47	4977.23	30333.13	93360.92	994.13	3507.33	11.51	2.48	24.88	60.45	
72	P021	8105.41	-8286.12	26427.88	1157.85	35485.22	93630.1	1033.47	4977.23	30333.13	93360.92	994.13	3507.33	11.51	2.48	24.88	60.45	
73	P021	8105.41	-8286.12	26427.88	1157.85	35485.22	93630.1	1033.47	4977.23	30333.13	93360.92	994.13	3507.33	11.51	2.48	24.88	60.45	
74	P024	16160.23	-17247.44	33407.67	174.01	40268.55	110288.79	2095.8	6882.22	0	84811.82	1196.7	1200.2	200	58.3	53.03	48.54	
75	P024	16160.23	-17247.44	33407.67	174.01	40268.55	110288.79	2095.8	6882.22	0	84811.82	1196.7	1200.2	200	58.3	53.03	48.54	
76	P025	12696.95	-8620.08	19096.64	221.54	23508.02	64759.36	140.43	4067.12	0	45494.12	806.31	611.77	200	14.87	56	-71.5	
77	P025	12696.95	-8620.08	19096.64	221.54	23508.02	64759.36	140.43	4067.12	0	45494.12	806.31	611.77	200	14.87	56	-71.5	
78	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	
79	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	
80	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	
81	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	
82	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	
83	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	
84	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	
85	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	
86	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	
87	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	
88	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	
89	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	
90	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	
91	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	
92	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	
93	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	
94	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	
95	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	
96	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	
97	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	
98	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	
99	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	
100	P027	13699.98	-7460.74	26014.72	106.56	30254.34	86844.73	1297.1	4406.64	8768.44	70174.51	819.46	687.86	110.12	10.68	45.13	-46.1	

Δημήτριος Πάλλας, Δυναμική Τιμολόγηση με Ελαχιστοποίηση της Σπατάλης Τροφίμων μέσω Πρόβλεψης της Ζήτησης, ABC Ταξινόμησης και Ενισχυτικής Μάθησης

I	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	seed	product_id	rl_reward	heuristic_reward	improvement	improvement_pct	rl_profit	rl_revenue	rl_waste	rl_lost_sales	hour_profit	hour_revenue	hour_waste	hour_lost_sales	profit_improvement_pct	revenue_improvement_pct	waste_reduction_pct	lost_sales_reduction_pct
80	42	P002	21848.79	-2562.77	24411.56	952.55	37372.54	116633.95	1772.52	8809.63	38081.33	113075.08	1485.15	5035.35	-1.88	-2.07	17.84	54.52
81	42	P003	26588.27	-1907.36	28495.64	1493.87	39632.56	113317.43	1823.3	7734.03	35736.93	111350.89	1475.04	4909.88	10.34	1.75	21.12	44.67
82	42	P005	18584.13	-432.23	19016.35	4399.63	26101.59	71129.07	1076.16	4017.13	21900.11	68865.83	805.1	3084.91	17.5	3.23	28.82	26.25
83	42	P006	13725.68	-5491.28	19217.95	349.97	22151.16	60304.42	1092.99	3659.04	0	41125.8	575.02	9314.53	200	37.82	62.11	47.63
84	42	P007	13738.56	-857.01	22130.56	260.27	21739.22	60115.56	686.4	2422.02	5882.89	47063.13	414.63	4586.4	114.81	24.36	49.37	-61.77
85	42	P008	35142.89	-12525.71	47668.6	380.57	53274.66	147756.7	1585.86	6062.02	0	100895.04	830.39	14957.76	200	37.69	62.53	-84.64
86	42	P009	20831.82	-8481.16	29313.98	145.6	30483.53	86108.39	1423.15	6020.88	26808.58	84118.46	1113.85	4033.78	12.83	2.34	24.52	39.53
87	42	P010	18521.87	-1308.74	19830.61	1515.24	26699.14	73972.17	759.28	3041.94	22950.56	72265.34	587.95	2091.49	15.1	2.33	25.43	37.03
88	42	P011	14376.05	-5669.42	20045.46	353.57	22748.48	62034.27	919.57	3072.85	0	42387.04	480.05	7739.95	200	37.68	62.81	-86.33
89	42	P012	24964.7	-11635.56	36600.26	314.56	38340.38	111072.06	1605.93	7036.06	35365.16	109954.71	1289.21	4324.9	8.36	1.01	21.88	47.46
90	42	P015	21871.15	-1218.06	23091.21	1895.74	31890.37	87234.47	1540.1	5685.56	27076.64	84472.4	1176.89	4160.51	16.33	3.24	26.74	31.41
91	42	P016	12674.81	-4612.97	17287.77	374.76	19055.05	53591.39	866.89	3486.53	0	36874.52	463.26	8175.04	200	36.96	60.69	-80.17
92	42	P017	7545.76	-6566.11	14109.87	214.89	11682.57	33653.51	487	2177.76	3341.28	26730.25	305.78	3631.45	111.04	22.93	45.72	-50.05
93	42	P018	24212.05	-2182.5	26495.15	1160.79	35462.32	98199.42	1289.41	5008.7	30333.13	93368.92	994.13	3507.33	15.39	2.91	25.86	31.26
94	42	P019	9787.21	-9262.44	19049.65	205.67	21954.23	61517.92	1008.16	3012.02	6182.05	49456.4	626.06	4801.88	112.11	12.74	46.76	-45.81
95	42	P020	6458.4	-11764.22	18222.62	154.9	17456.2	52642.54	1358.73	4788.21	5445.53	43564.26	898.46	6625.7	104.89	18.87	40.78	-32.2
96	42	P021	8100.54	-11741.16	19841.7	168.99	23656.08	69521.87	1550.41	5269.89	7181.66	57453.25	1004.41	7420.44	106.79	19.01	42.74	-33.89
97	42	P022	13158.88	-13339.78	26488.67	185.84	34877.68	101564.47	1541.99	6981.77	0	74140.71	1194.89	11626.57	200	31.22	52.43	-53.99
98	42	P023	8852.92	-9921.03	18773.94	189.23	22727.76	67298.89	841.77	2938.1	0	49138.48	501.99	5003.68	200	31.19	50.57	-52.02
99	42	P024	12939.65	-17247.44	30187.1	175.02	39518.74	116628.36	2008.32	6914.47	0	84813.82	1196.7	12002.2	200	31.59	50.65	-53.79
100	42	P025	9965.35	-8620.08	18585.44	215.61	23829.04	65241.54	1402.92	3894.96	0	45494.12	806.31	8617.77	200	35.67	53.94	-75.49
101	42	P026	12455.39	-15277.59	27732.99	181.53	34282.85	103628.06	1821.3	6746.67	0	75785.07	1107.17	11278.06	200	31.04	48.77	-50.28
102	42	P027	12318.18	-19644.74	25962.92	190.28	31169.18	87427.85	1287.69	3809.35	8768.44	70147.51	819.46	6162.86	112.18	21.93	44.44	-47.2
103	42	P028	12821.49	-19897.38	32358.88	164.63	31173.65	90807.94	1543.91	5215.42	9218.24	73825.91	1017.49	7583.67	108.64	20.63	41.1	-36.82
104	42	P116	9040.63	-5663.36	14703.99	250.63	15093.12	42050.92	1393.45	1198.13	3734.94	23873.53	2355.49	16267.56	110.65	29.09	30.18	-50.35
105	42	P117	2016.54	-9802.02	2996.56	305.77	3547.7	9216.91	1136.9	3905.27	845.33	6762.68	856.22	5607.45	123.03	30.72	28.16	-35.79

Εικόνα 8.4 RL all results seed 42

I	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	seed	product_id	rl_reward	heuristic_reward	improvement	improvement_pct	rl_profit	rl_revenue	rl_waste	rl_lost_sales	hour_profit	hour_revenue	hour_waste	hour_lost_sales	profit_improvement_pct	revenue_improvement_pct	waste_reduction_pct	lost_sales_reduction_pct
106	99	P002	26446.76	-2562.77	29009.53	1131.96	42569.64	121176.91	1809.47	7633.99	38081.33	113075.08	1485.15	5035.35	11.13	1.75	19.69	41.02
107	99	P003	24329.89	-1907.36	26237.26	1375.58	38235.1	112245.9	1768.06	8049.48	35736.93	111350.89	1475.04	4909.88	6.75	0.8	18.07	48.45
108	99	P005	16984.87	-432.23	17417.09	4029.63	25206.85	70609.01	1027.45	4392.15	21900.11	68865.83	805.1	3084.91	14.04	2.5	24.27	34.97
109	99	P006	13842.35	-5491.28	19333.63	348.44	20800.45	59353.72	1039.85	4242.8	0	41125.8	575.02	9314.53	200	36.28	57.57	-74.82
110	99	P007	13430.92	-8572.01	22002.93	256.68	20877.67	59527.71	657.82	2705.2	5882.89	47063.13	414.63	4586.4	112.07	23.39	45.35	-51.6
111	99	P008	34161.02	-12525.71	46686.73	372.73	52161.38	146800.06	1587.95	6424.87	0	100895.04	830.39	14957.76	200	37.07	61.51	-79.81
112	99	P009	14347.06	-8482.16	22829.22	269.14	28831.7	84846.58	1354.04	6130.95	26808.58	84118.46	1113.85	4033.78	7.27	0.86	19.47	41.26
113	99	P010	18332.82	-1308.74	19641.07	1500.76	26582.64	73843.03	766.54	3050.44	29055.26	72265.34	587.95	2091.49	14.67	2.16	26.37	37.3
114	99	P011	12391.33	-5669.42	18060.74	318.56	21578.2	61186.1	884.4	3373.92	0	42387.04	480.05	7739.95	200	36.35	59.27	-78.57
115	99	P012	27221.34	-11635.56	38856.9	335.95	41917.92	113871.49	1706.81	5856.1	35265.16	109954.71	1289.21	4324.9	17.24	3.5	27.88	30.08
116	99	P015	21974.8	-1218.06	23192.86	1904.08	32609.39	87808.71	1553.91	5344.91	27076.64	84472.4	1176.89	4160.51	18.54	3.87	27.61	25.39
117	99	P016	13407.35	-4612.97	18020.31	390.64	19541.88	53752.54	908.58	3289.03	0	36874.52	463.26	8175.04	200	37.25	64.92	-85.24
118	99	P017	7961.64	-6566.11	14527.75	221.25	12135.65	34020.46	492.43	2078.84	3341.28	26730.25	305.78	3631.45	113.64	24	46.77	-54.38
119	99	P018	22094.62	-2182.5	23377.13	1024.19	33152.74	96336.86	1220.25	5603.81	30333.13	93368.92	994.13	3507.33	8.88	1.22	20.42	46.02
120	99	P019	8406.39	-9262.44	17668.83	190.76	20884.07	61047.91	935.38	3084.45	6182.05	49456.4	626.06	4801.88	108.64	20.98	46.87	-43.55
121	99	P020	6079.53	-11764.22	17843.75	151.68	17625.75	52834.64	1344.12	4752.58	5445.53	43564.26	898.46	6625.7	105.59	19.23	39.74	-32.92
122	99	P021	7963.09	-11741.16	19704.25	167.82	24968.3	70395.19	1548.68	4782.26	7181.66	57453.25	1004.41	7420.44	110.65	21	42.51	-43.22
123	99	P022	9168.1	-15329.78	24497.88	159.81	34478.12	101469.08	2018.88	6447.59	0	74140.71	1194.89	11626.57	200	31.12	51.28	-57.31
124	99	P023	7475.02	-9921.03	17396.04	175.35	23708	68310.19	840.56	2659.43	0	49138.48	501.99	5003.68	200	32.65	50.44	-61.8
125	99	P024	12617.03	-17247.44	29864.47	175.15	37284.82	115169.23	1923.32	7366.41	0	84813.82	1196.7	12002.2	200	30.36	46.58	-47.87
126	99	P025	9489.71	-8620.08	18109.8	210.09	21898.65	63925.99	1327.82	4544.14	0	45494.12	806.31	8617.77	200	33.69	48.87	-61.9
127	99	P026	14163.31	-15277.59	29440.9	192.71	35857.64	104127.99	1986.21	6470.28	0	75785.07	1107.17	11278.06	200	31.51	54.48	-54.18
128	99	P027	11421.11	-19644.74	25063.85	183.7	30257.29	86253.43	1316.73	3988.26	8768.44	70147.51	819.46	6162.86	110.13	20.6	46.56	-44.84
129	99	P028	11233.04	-19897.38	31070.42	156.63	29644.34	89707.55	1540.23	5530.32	9218.24	73825.91	1017.49	7583.67	105.04	18.98	40.88	-31.52
130	99	P116	9151.78	-5663.36	14815.14	261.6	14824.84	39798.47	3184.12	12205.57	3734.94	23873.53	2355.49	16267.56	119.5	28.47	29.92	-28.55
131	99	P117	2126.23	-9802.02	3106.25	316.96	3483.92	9143.09	1143.03	4076.58	845.33	6762.68	856.22	5607.45	121.9	29.93	28.69	-31.62

Εικόνα 8.5 RL all results seed 99

Δημήτριος Πάλλας, Δυναμική Τιμολόγηση με Ελαχιστοποίηση
της Σπατάλης Τροφίμων μέσω Πρόβλεψης της Ζήτησης, ABC
Ταξινόμησης και Ενισχυτικής Μάθησης

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	product_id	profit_improvement_pct_mean	profit_improvement_pct_std	revenue_improvement_pct_mean	revenue_improvement_pct_std	waste_reduction_pct_mean	waste_reduction_pct_std	lost_sales_reduction_pct_mean	lost_sales_reduction_pct_std	profit_ci95	revenue_ci95	waste_ci95	lost_sales_ci95
2	P002	9,85	6,71	1,35	1,98	21,6	2,8	40,36	8,53	5,88	1,74	2,46	7,48
3	P003	10,16	2,65	1,62	0,67	21,31	2,59	44,84	4,82	2,32	0,58	2,27	4,22
4	P005	17,69	2,16	3,53	0,66	27,16	1,81	25,72	5,55	1,9	0,58	1,58	4,86
5	P006	200	0	37,2	0,6	61,29	2,24	-83,6	5,15	0	0,53	1,97	4,52
6	P007	112,15	1,73	23,39	0,61	46,79	2,19	-54,01	4,88	1,52	0,54	1,92	4,28
7	P008	200	0	37,16	1,21	60,18	4,25	-80,47	7,96	0	1,06	3,73	6,97
8	P009	12,73	3,17	2,29	0,83	24,06	2,75	37,05	3,36	2,78	0,73	2,41	2,94
9	P010	15,26	1,77	2,38	0,35	25,9	1,98	34,91	3,23	1,55	0,3	1,74	2,83
10	P011	200	0	37,04	0,92	60,86	4,41	-81,91	7,01	0	0,81	3,86	6,14
11	P012	12,99	4,06	2,28	1,15	25,11	2,7	37,99	8,2	3,56	1,01	2,36	7,19
12	P015	14,31	5,48	2,72	1,54	24,86	3,51	34,8	10,51	4,8	1,35	3,08	9,22
13	P016	200	0	36,62	0,79	60,63	3,95	-79,54	6,08	0	0,69	3,46	5,33
14	P017	112,59	2,11	23,57	0,72	46,34	2,01	-53,65	4,65	1,85	0,64	1,77	4,07
15	P018	14,67	3,47	2,73	0,86	24,69	3,5	35,3	6,66	3,04	0,75	3,07	5,83
16	P019	108,74	2,53	20,91	0,74	40,77	3,77	-41,03	4,71	2,22	0,65	3,3	4,13
17	P020	105,64	0,85	19,08	0,29	41,12	1,17	-32,95	1,58	0,75	0,25	1,03	1,39
18	P021	108,79	2,23	20,08	1,02	42,62	1,22	-38,21	4,92	1,95	0,89	1,07	4,31
19	P022	200	0	30,97	0,31	50,81	1,95	-52,89	3,3	0	0,27	1,71	2,89
20	P023	200	0	31,7	0,58	50,88	0,45	-56,3	4,03	0	0,51	0,39	3,53
21	P024	200	0	31,49	0,64	50,85	3,07	-52,98	2,89	0	0,56	2,69	2,53
22	P025	200	0	34,49	0,79	52,56	2,71	-67,94	5,55	0	0,69	2,37	4,87
23	P026	200	0	32,95	1,63	52,65	3,62	-64,55	11,94	0	1,43	3,17	10,47
24	P027	109,34	3,34	20,79	1,08	43,34	3,84	-41,32	6,6	2,93	0,95	3,37	5,78
25	P028	109,75	3,14	20,77	1,25	43,56	2,65	-39,97	6,98	2,75	1,1	2,32	6,11
26	P116	120,02	0,94	28,78	0,49	29,81	0,51	-29,67	1,64	0,83	0,43	0,44	1,43
27	P117	122,74	0,82	30,52	0,51	28,25	0,27	-35,22	3,17	0,72	0,45	0,24	2,78
28													
29	AVERAGE	112,2084615	1,813846154	20,63115385	0,854615385	40,69230769	2,535384615	-26,74	5,534615385	1,590384615	0,749615385	2,222307692	4,85

Εικόνα 8.6 RL Business summary

Υπεύθυνη Δήλωση Συγγραφέα:

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