ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ»

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Συστήματα μηχανικής μάθησης στις ξενοδοχειακές κρατήσεις

Δήμου Αντιγόνη

Υποβλήθηκε ως προ-απαιτούμενο για την απόκτηση του
Μεταπτυχιακού Διπλώματος ειδίκευσης στα
Πληροφοριακά Συστήματα.

Επιβλέπων καθηγητής
Ευάγγελος Καλαμπόκης

ΘΕΣΣΑΛΟΝΙΚΗ

2024

Περίληψη

Η παρούσα διπλωματική εργασία φέρει τον τίτλο «Συστήματα Μηχανικής Μάθησης στις Ξενοδοχειακές Κρατήσεις» και αναλύει το πρόβλημα των ακυρώσεων κρατήσεων στα ξενοδοχεία, χρησιμοποιώντας μηχανική μάθηση, ειδικότερα τον αλγόριθμο XGBoost.

Οι διαδικτυακές πλατφόρμες κρατήσεων για ξενοδοχεία έχουν επιφέρει σημαντικές αλλαγές στον τρόπο που οι πελάτες πραγματοποιούν κρατήσεις και στη συμπεριφορά τους. Πολλές κρατήσεις ακυρώνονται λόγω αλλαγής σχεδίων ή διάφορων προγραμμάτων. Το ερώτημα που τίθεται είναι το εξής: "Μπορούμε να προβλέψουμε αν ένας πελάτης θα τηρήσει την κράτησή του ή θα την ακυρώσει;"

Η μελέτη χρησιμοποιεί ένα σύνολο δεδομένων από το Kaggle.com με πραγματικά δεδομένα από ξενοδοχεία. Για καλύτερη κατανόηση, πραγματοποιείται διερευνητική ανάλυση των δεδομένων με οπτικοποιήσεις και διαγράμματα με χρήση της Python. Στο στάδιο προεπεξεργασίας των δεδομένων, προστίθενται νέες ανεξάρτητες μεταβλητές για τη βελτιστοποίηση της ανάλυσης.

Δημιουργείται ένα προβλεπτικό μοντέλο XGBoost, το οποίο ανήκει στη βιβλιοθήκη μηχανικής εκμάθησης με ενισχυμένη κλίση δέντρων απόφασης, γνωστή ως gradient-boosted decision trees (GBDT). Το σύνολο δεδομένων διαιρείται σε δεδομένα εκπαίδευσης και δεδομένα δοκιμής, ενώ η τεχνική cross-validation βοηθά στον καλύτερο διαμοιρασμό των δεδομένων. Επίσης, εφαρμόζεται αναζήτηση πλέγματος (Grid Search) για τη ρύθμιση των υπερπαραμέτρων.

Για την αξιολόγηση του μοντέλου χρησιμοποιούνται οι καμπύλες ROC, AUC και η βιβλιοθήκη SHAP για μεγαλύτερη διαφάνεια των παραγόντων που επηρέασαν τη βελτιστοποίηση του μοντέλου.

Η μηχανική μάθηση είναι ένα εργαλείο, που μπορεί να φάνει πολύ χρήσιμο και επικερδές στον τομέα του τουρισμού.

Λέξεις κλειδιά: σύνολο δεδομένων, μηχανική μάθηση, τεχνητή νοημοσύνη, επιστήμη δεδομένων, αλγόριθμος μηχανικής μάθησης, λήψη αποφάσεων, επιχειρηματική ευφυία, python, jupyter notebook, kaggle, csv, grid search, xgboost, Shap, επεξεγησιμότητα, ξενοδοχειακές κρατήσεις

Abstract

This thesis is entitled "Machine Learning Systems in Hotel Reservations" and analyses the problem of hotel reservation cancellations using machine learning, in particular the XGBoost algorithm.

Online booking platforms for hotels have brought about significant changes in the way customers make reservations and in their behavior. Many bookings are cancelled due to change of plans or various schedules. The question that arises is this: "Can we predict whether a customer will keep their reservation or cancel it?"

The study uses a dataset from Kaggle.com with real data from hotels. For better understanding, exploratory analysis of the data is performed with visualizations and charts using Python. In the data pre-processing stage, new independent variables are added to optimize the analysis.

An XGBoost predictive model is created, which belongs to the machine learning library of gradient-boosted decision trees, known as gradient-boosted decision trees (GBDT). The dataset is divided into training data and test data, and the cross-validation technique helps in better data sharing. Grid Search is also applied to adjust the hyperparameters.

To evaluate the model, the ROC and AUC curves and the SHAP library are used for greater transparency of the factors that influenced the optimization of the model.

Machine learning is a tool, which can be very useful and profitable in the tourism sector.

Keywords: dataset, machine learning, Artificial Intelligence, data science, machine learning algorithm, decision making, data engineering , business intelligence, python, jupyter notebook, kaggle, csv, grid search, xgboost, Shap, explainability, hotel cancelation

Περιεχόμενα

Λίστα Εικόνων

1. Εισαγωγή

Η ιστορία των ξενοδοχείων συνδέεται άρρηκτα με την εξέλιξη του ανθρώπινου πολιτισμού. Αρχικά, ο ξένος ήταν ένα σπάνιο φαινόμενο και έτσι η παρέα του ήταν περιζήτητη. Οι ντόπιοι ανταγωνίζονταν για το ποιος θα φιλοξενούσε τον επισκέπτη, ενώ οι ξένοι παρείχαν πολύτιμες πληροφορίες σχετικά με θέματα όπως τα στρατιωτικά, πολιτιστικά και τεχνικά. η φιλοξενία εκείνη την εποχή προσέφερε κοινωνική αίγλη.

Καθώς ο αριθμός των ξένων αυξανόταν, οι πληροφορίες που μεταφέρονταν ήταν προκαταρκτικά γνωστές, καθώς άλλοι ξένοι επισκέπτες είχαν ήδη μοιραστεί τις εμπειρίες τους, και οι ντόπιοι έχασαν το ενδιαφέρον να φιλοξενήσουν τους νέους επισκέπτες. Η ανάγκη για διαμονή σε κάποιο κατάλυμα, επέβαλε πλέον στους ξένους την πληρωμή. Αυτό οδήγησε στην παρέμβαση του κράτους για την αντιμετώπιση του προβλήματος της διαμονής των ξένων, με αποτέλεσμα την εμφάνιση των πρώτων ξενοδοχείων. Το πρώτο ξενοδοχείο στην Ευρώπη, το "Αετός," εμφανίστηκε περίπου το 1302 στη Γαλλία.

Σήμερα, εκατομμύρια τουρίστες κινούνται κάθε χρόνο, ενώ η βιομηχανία της φιλοξενίας έχει αποκτήσει επιστημονικές διαστάσεις, αναδεικνύοντας τον τουρισμό ως σημαντικό κοινωνικό και οικονομικό φαινόμενο.

Από το 1302 μέχρι και σήμερα, ο τρόπος διεξαγωγής των κρατήσεων έχει αλλάξει ποικιλότροπος ανάλογα με τις τάσεις και την τεχνολογία της εκάστοτε εποχής. Πολύ πριν το Διαδίκτυο γίνει το νούμερο ένα εργαλείο του σύγχρονου εμπορίου και επικοινωνίας, η διαδικασία εύρεσης καταλύματος ήταν πολύ πιο περίπλοκη και αδιαφανής.

Αρχικά, ο επισκέπτης έπρεπε πρώτα να καταφθάσει στον προορισμό του και να αναζητήσει επιτόπου ξενοδοχείο, καθώς δεν υπήρχαν τα μέσα για να προνοήσει και να κάνει κράτηση νωρίτερα. Ακόμα και μετά την εφεύρεση του τηλεφώνου, η εύρεση ενός χώρου διαμονής απαιτούσε άμεση τοπική γνώση, μελέτη ενός βιβλίου οδηγού, λήψη ενός φυλλαδίου ή κάποιου άλλου είδους ερευνητικής εργασίας. Εκείνη την εποχή των αποκεντρωμένων πληροφοριών και των χάρτινων αρχείων, ο ταξιδιωτικός πράκτορας βασίλευε υπέρτατα.

Με την χρήση του διαδικτύου, η όλη διαδικασία έχει αλλάξει ριζικά. Σύγκριση τιμών και εγκαταστάσεων, φωτογραφίες των δωματίων, ειδικές προσφορές κ.α. είναι πλέον διαθέσιμες υπηρεσίες που καθιστούν εύκολη την εύρεση ενός καταλύματος. Οι ταξιδιώτες μπορούν να κάνουν κράτηση δωματίων χρησιμοποιώντας την ασφάλεια του διαδικτύου για να προστατεύσουν το απόρρητο και τις οικονομικές τους πληροφορίες.

1.1 Διατύπωση προβλήματος

Η διαδικασία των κρατήσεων ξενοδοχείων έχει υποστεί σημαντικές αλλαγές, κυρίως λόγω της αυξημένης χρήσης των διαδικτυακών ταξιδιωτικών πρακτορείων (OTAs Online Travel Agencies). Η δυνατότητα εύκολης και οικονομικής κράτησης δωματίου μέσω αυτών των πλατφορμών έχει καταστήσει πιο προσιτή τη διαδικασία για τους ταξιδιώτες. Εκτός από την άνεση που παρέχουν κατά τη διάρκεια της κράτησης, οι OTAs διευκολύνουν επίσης την ακύρωση των κρατήσεων, καθιστώντας τη διαδικασία αυτή πιο απλή.

Συγκριτικά, ο παραδοσιακός τρόπος κρατήσεων μέσω εκτός σύνδεσης (offline) ή απευθείας στην ρεσεψιόν φαίνεται πιο πιθανό να αποτρέψει έναν πελάτη από το να ακυρώσει την κράτηση. Αυτό οφείλεται ίσως στον υψηλότερο βαθμό προσπάθειας, χρόνου και χρημάτων που απαιτούνται για μια απευθείας κράτηση. Η τεχνολογική πρόοδος δεν είναι μόνο ο μοναδικός παράγοντας που επηρεάζει τις ακυρώσεις, καθώς ο ρυθμός της καθημερινής ζωής και οι απρόβλεπτες συνθήκες παίζουν επίσης σημαντικό ρόλο.

Είτε πρόκειται για ξαφνικές αλλαγές στα ταξιδιωτικά σχέδια, απρόβλεπτες καταστάσεις ή απλά αλλαγή γνώμης, οι ακυρώσεις κρατήσεων ξενοδοχείων έχουν καταντήσει συνηθισμένο φαινόμενο. Οι επιπτώσεις αυτών των ακυρώσεων είναι σημαντικές τόσο για τους ταξιδιώτες όσο και για τους ξενοδόχους.

1.2 Στόχος

Η παρούσα ερευνητική διαδικασία επιδιώκει τη δημιουργία ενός προβλεπτικού μοντέλου που θα είναι ικανό να αναγνωρίζει προκαταβολικά κρατήσεις που πιθανόν να ακυρωθούν από τον πελάτη. Η βάση αυτού του μοντέλου θα αποτελείται από δεδομένα ξενοδοχειακών κρατήσεων.

Τα δεδομένα αυτά θα υποβληθούν σε εκτεταμένη και ενδελεχής ανάλυσης. Κατά τη διάρκεια αυτής της διαδικασίας, θα αναδειχθούν νέες παράμετροι που θα ενισχύσουν το μοντέλο, με στόχο την βελτιστοποίηση της προβλεπτικής του ικανότητας.

Οι προσπάθειες εστιάζουν στην εξαγωγή σημαντικών συσχετίσεων από τα δεδομένα και στον προσδιορισμό κρίσιμων παραμέτρων που επηρεάζουν την πιθανότητα ακύρωσης. Η τελική επίτευξη είναι ένα μοντέλο που είναι ικανό να προβλέπει με υψηλή ακρίβεια τις ακυρώσεις κρατήσεων, με σκοπό τη βελτίωση της αποτελεσματικότητας και της ανταγωνιστικότητας στον τομέα της διαχείρισης κρατήσεων ξενοδοχείων

1.3 Περιεχόμενο της μελέτης

Η παρούσα εργασία ακολουθεί την παρακάτω δομή:

Στο κεφάλαιο 2 γίνεται αναφορά στο γνωστικό υπόβαθρο που απαιτείται για τη μελέτη και εισάγονται κάποιες βασικές έννοιες. Εισάγεται η έννοια της μηχανικής μάθησης, των συνιστωσών που περιλαμβάνει και των κατηγόριων που χωρίζεται . Αποκτάται μια πρώτη επαφή με την βιβλιοθήκη XGBoost, που αποτελεί τον πυρήνα της εργασίας, καθώς και με την βιβλιοθήκη SHAP, που βοηθάει στην καλύτερη κατανόηση και διαφάνεια των αποτελεσμάτων του μοντέλου XGBoost. Αναφέρονται επίσης χρήσιμα εργαλεία και περιβάλλοντα που θα βοηθήσουν στην παρούσα μελέτη.

Στο κεφάλαιο 3 παρουσιάζεται μια βιβλιογραφική ανασκόπηση με σκοπό να παρέχει πληροφορίες σχετικά με την χρονική διάρκεια που το πεδίο του προβλήματος μελετάται από την ερευνητική κοινότητα, καθώς και τους τρόπους που έχουν υιοθετηθεί για την αντιμετώπισή του.

Στο κεφάλαιο 4 παρουσιάζεται η μεθοδολογία που θα ακολουθηθεί στην συνέχεια.

Στο κεφάλαιο 5 γίνεται μια γνωριμία με το σύνολο δεδομένων, τα χαρακτηριστικά και το εύρος τιμών αυτών.

Ακολουθεί το κεφάλαιο 6, στο όποιο πραγματοποιείται η διερευνητική ανάλυση των δεδομένων. Με την εξερεύνηση και την οπτικοποίηση των δεδομένων μέσω της Python, προκύπτουν κάποιες πρώτες συσχετίσεις.

Στο κεφάλαιο 7, δημιουργούνται και εντάσσονται νέες μεταβλητές στο μοντέλο με σκοπό την βελτιστοποίηση αυτού.

Στο κεφάλαιο 8 υλοποιείται η μηχανική μάθηση. Αρχικά, γίνεται ο συντονισμός υπερπαραμέτρων με την Αναζήτηση Πλέγματος (Grid Search) και η δημιουργία του μοντέλου XGBoost μετά από αυτήν. Στην συνέχεια ακολουθούν κάποιοι εκτιμητές αξιολόγησης, καμπύλες ROC και AUC, διαγράμματα σπουδαιότητας βάρους και κέρδους. Τέλος παρουσιάζονται τα διαγράμματα της βιβλιοθήκης SHAP για την βέλτιστη ερμηνεία του μοντέλου.

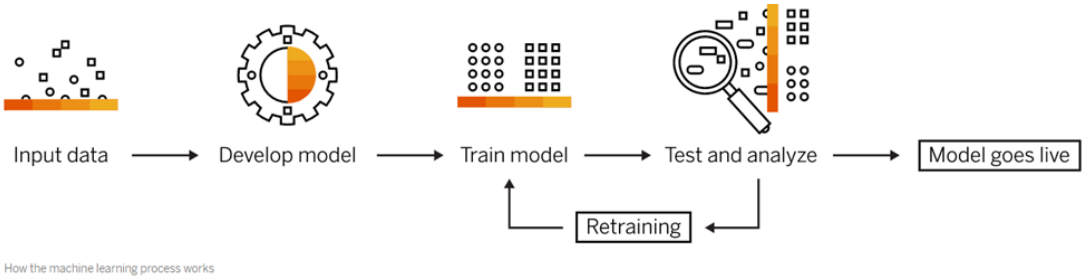
Στο κεφαλαιο 9 περιγράφονται τα συμπεράσματα.

2. Γνωστικό Υπόβαθρο

2.1 Μηχανική Μάθηση

2.1.1 Ορισμός Μηχανικής Μάθησης

Η μηχανική μάθηση είναι ένας κλάδος της τεχνητής νοημοσύνης (AI : Artificial Intelligence) και της επιστήμης των υπολογιστών που εστιάζει στη χρήση δεδομένων και αλγορίθμων για τη μίμηση του τρόπου με τον οποίο μαθαίνουν οι άνθρωποι, βελτιώνοντας σταδιακά την ακρίβειά της. (*What is Machine Learning?* | *IBM*, n.d.)



Εικόνα 1: Διαδικασία Μηχανικής Μάθησης

(Sharma A., 2022)

2.1.2 Συνιστώσες Μηχανικής μάθησης

Το σύστημα εκμάθησης ενός αλγόριθμου μηχανικής μάθησης αναλύεται σε τρείς κύριες συνιστώσες:

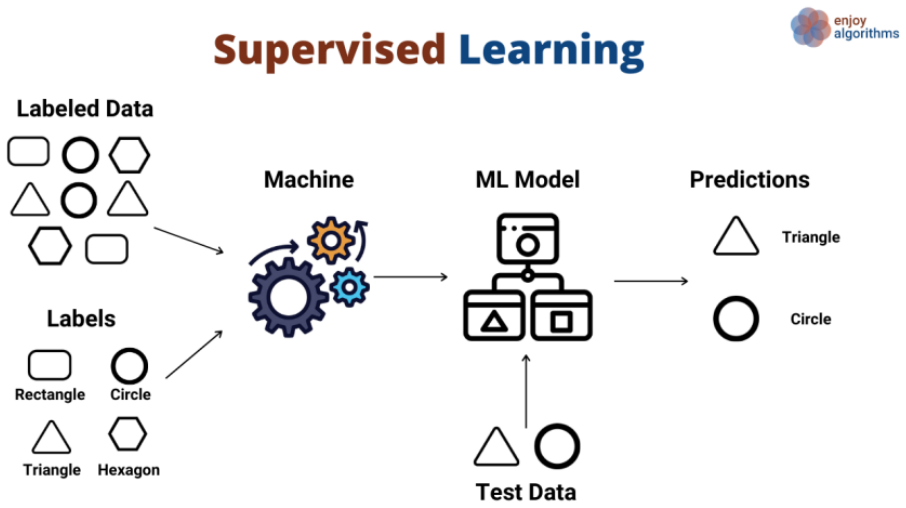
1. **Διαδικασία απόφασης**: Γενικώς, οι αλγόριθμοι της μηχανικής μάθησης αξιοποιούνται με σκοπό να προβλέπουν ή να κατηγοριοποιούν. Με βάση τα δεδομένα εισόδου, που μπορεί να περιλαμβάνουν ετικέτες ή να είναι χωρίς αυτές, ο αλγόριθμος παράγει μια εκτίμηση σχετικά με ένα μοτίβο που υπάρχει στα δεδομένα.
2. **Μια συνάρτηση σφάλματος:** Η αξιολόγηση της πρόβλεψης του μοντέλου γίνεται μέσω μιας συνάρτησης σφάλματος. Στην περίπτωση όπου υπάρχουν γνωστά παραδείγματα, αυτή η συνάρτηση σφάλματος χρησιμοποιείται για να συγκρίνει και να αξιολογήσει την ακρίβεια του μοντέλου.
3. **Διαδικασία βελτιστοποίησης μοντέλου:** Αν το μοντέλο έχει τη δυνατότητα να προσαρμοστεί καλύτερα στα δεδομένα εκπαίδευσης, τότε προσαρμόζονται τα βάρη του ώστε να μειωθεί η απόκλιση μεταξύ της πραγματικής τιμής και της πρόβλεψης του μοντέλου. Ο αλγόριθμος επαναλαμβάνει αυτήν τη διαδικασία "αξιολόγησης και βελτιστοποίησης", ενημερώνοντας αυτόματα τα βάρη μέχρις ότου επιτευχθεί ένα καθορισμένο επίπεδο ακρίβειας.

2.1.3 Κατηγορίες Μηχανικής Μάθησης & Αλγοριθμικές Τεχνικές

Ανάλογα με τη φύση των δεδομένων και το επιθυμητό αποτέλεσμα, επιλέγεται μια από τις τέσσερις βασικές κατηγορίες μηχανικής μάθησης: εποπτευόμενη, μη εποπτευόμενη, ημι-εποπτευόμενη, ή ενισχυτική, οι οποίες αναλύονται εκτενέστερα παρακάτω. Κάθε κατηγορία μπορεί να εφαρμοστεί χρησιμοποιώντας μία ή περισσότερες αλγοριθμικές τεχνικές.

Οι αλγοριθμικές τεχνικές μπορούν να χρησιμοποιηθούν ανεξάρτητα ή να συνδυαστούν για να επιτευχθεί η καλύτερη δυνατή ακρίβεια, ιδίως όταν αντιμετωπίζουμε σύνθετα και προβληματικά δεδομένα.

1. **Εποπτευόμενη Μηχανική Μάθηση:** Η εποπτευόμενη μάθηση αποτελεί ένα είδος Μηχανικής Μάθησης, κατά το οποίο ένας αλγόριθμος εκπαιδεύεται να λαμβάνει αποφάσεις ή να κάνει προβλέψεις βασιζόμενος σε πληροφορίες που έχουν ετικέτες. Τα δεδομένα με ετικέτα αποτελούνται από προηγουμένως γνωστές μεταβλητές εισόδου (γνωστές επίσης ως χαρακτηριστικά) και μεταβλητές εξόδου (γνωστές επίσης ως ετικέτες). Ο αλγόριθμος, αναλύοντας τις σχέσεις και τα μοτίβα μεταξύ αυτών των μεταβλητών εισόδου και εξόδου, μαθαίνει να παράγει προβλέψεις. Παραδείγματα τέτοιων εφαρμογών περιλαμβάνουν την αναγνώριση εικόνων και ομιλίας, τα συστήματα συστάσεων και την ανίχνευση απάτης.



Εικόνα 2. Μοντέλο Εποπτευόμενης Μηχανικής Μάθηση

(Kozan, M., 2021)

Η εποπτευόμενη μηχανική μάθηση κατηγοριοποιείτε σε δύο κύριους τύπους προβλημάτων, τους οποίους μπορούμε να περιγράψουμε ως εξής:

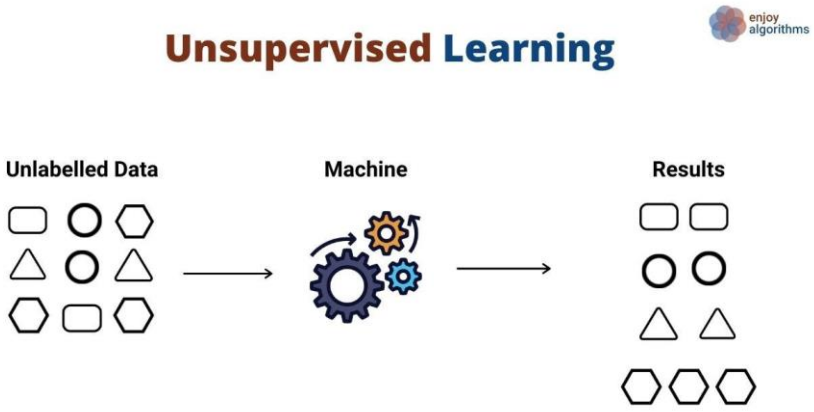
- Πρόβλημα Ταξινόμησης (classification)
- Πρόβλημα Παλινδρόμησης (regression)

Οι αλγοριθμικές τεχνικές που χρησιμοποιούνται σε προβλήματα εποπτευόμενης μηχανικής μάθησης είναι οι παρακάτω:

- Γραμμική Παλινδρόμηση (Linear Regression)
- Λογιστική Παλινδρόμηση (Logistic Regression)
- Κοντινότεροι Γείτονες (k-Nearest Neighbors - kNN)
- Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines - SVM)
- Δέντρα Απόφασης (Decision Trees)
- Τυχαίο Δάσος (Random Forest)
- Νευρωνικά Δίκτυα (Neural Networks)

2. **Μη Εποπτευόμενη Μηχανική Εκμάθηση:** Η μη εποπτευόμενη μηχανική μάθηση, όπως υποδηλώνει το όνομα της, δεν απαιτεί επίβλεψη κατά τη διάρκεια της εκπαίδευσης. Στην ουσία, στη μη εποπτευόμενη μηχανική μάθηση, το μηχάνημα εκπαιδεύεται χρησιμοποιώντας ένα σύνολο δεδομένων που δεν έχει ετικέτες, και το μηχάνημα προβλέπει αποτελέσματα χωρίς καμία επίβλεψη. Τα μοντέλα εκπαιδεύονται με δεδομένα που δεν είναι ούτε ταξινομημένα ούτε επισημασμένα και το μοντέλο ενεργεί σε αυτά τα δεδομένα χωρίς καθοδήγηση.

   Ο βασικός στόχος της μη εποπτευόμενης μηχανικής μάθησης είναι να ανακαλύψει δομές, μοτίβα, και σχέσεις μεταξύ των δεδομένων χωρίς προκαθορισμένες ετικέτες. Ο αλγόριθμος προσπαθεί να ομαδοποιήσει ή να κατηγοριοποιήσει τα δεδομένα ανάλογα με τις ομοιότητές τους, τα πρότυπα που παρουσιάζουν, και τις διαφορές τους. Το μηχάνημα αναλύει τα δεδομένα εισόδου προκειμένου να ανακαλύψει αόρατες δομές.



Εικόνα 3. Μοντέλο Μη Εποπτευόμενης Μηχανικής Μάθησης

(Kozan, M., 2021)

Η μη εποπτευόμενη μάθηση κατηγοριοποιείτε σε δύο κύριες κατηγορίες:

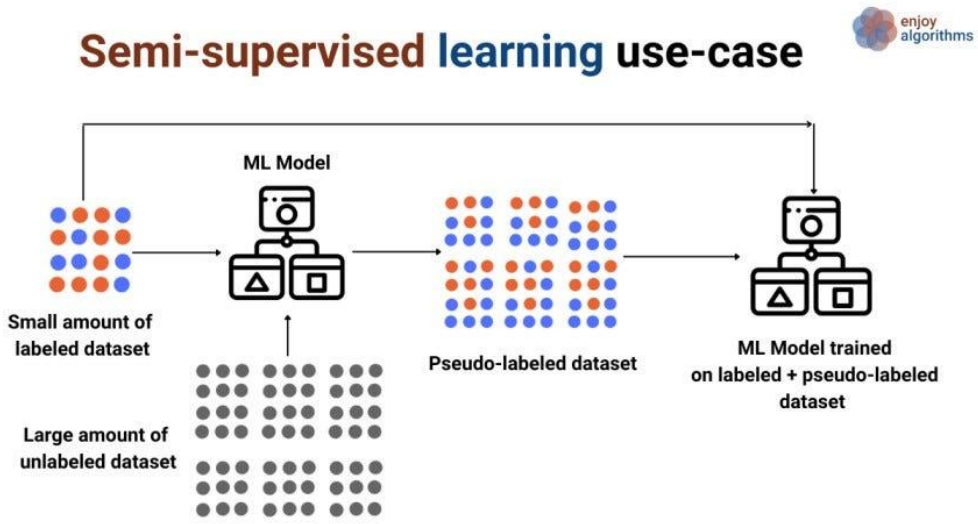
- Ομαδοποίηση (Clustering)
- Συσχέτιση (Association)

Οι αλγοριθμικές τεχνικές που χρησιμοποιούνται σε προβλήματα μη εποπτευόμενης μηχανικής μάθησης είναι οι παρακάτω:

- Κατηγοριοποίηση K-Means
- Ανάλυση Κύριων Συνιστωσών (Principal Component Analysis - PCA)
- Αποκαλυπτικός Κανόνας (Apriori)
- Κοινότητες σε Δίκτυα (Community Detection in Networks)

3. **Ημι-εποπτευόμενη μάθηση:** Η ημι-εποπτευόμενη μάθηση αναπαριστά ένα είδος μηχανικής μάθησης που βρίσκεται κάπου ανάμεσα στην εποπτευόμενη και την μη εποπτευόμενη μηχανική μάθηση. Αυτή η προσέγγιση αντιπροσωπεύει έναν ενδιάμεσο δρόμο μεταξύ αλγορίθμων μηχανικής εκμάθησης που απαιτούν πλήρες σύνολο δεδομένων με ετικέτες (εποπτευόμενη μάθηση) και αλγορίθμων που λειτουργούν χωρίς ετικέτες (μη εποπτευόμενη μάθηση). Κατά τη διάρκεια της φάσης εκπαίδευσης, χρησιμοποιεί τόσο δεδομένα που έχουν ετικέτες όσο και δεδομένα που δεν έχουν ετικέτες.

   Το κύριο χαρακτηριστικό αυτής της μεθόδου είναι ότι χρησιμοποιεί δεδομένα με ελάχιστες ετικέτες, καθώς αυτές μπορεί να είναι δαπανηρές. Συνεπώς, κατά τη διάρκεια της εκπαίδευσης, μπορεί να χρησιμοποιεί ένα περιορισμένο σύνολο ετικετών και να επωφελείται από δεδομένα χωρίς ετικέτες για τη βελτίωση της απόδοσής του.

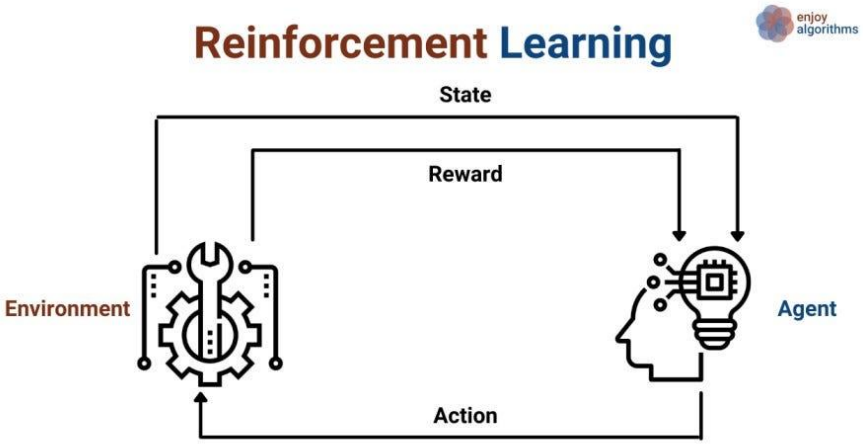


Εικόνα 4. Μοντέλο Ημι Εποπτευόμενης Μηχανικής Μάθησης

(Kozan, M., 2021)

4. **Ενισχυτική Μάθηση:** Η Ενισχυτική Μάθηση αποτελεί τον τομέα της επιστήμης που αφορά τη διαδικασία λήψης αποφάσεων. Σε αυτόν τον τομέα, ο στόχος είναι η εκμάθηση της βέλτιστης δυνατής συμπεριφοράς σε ένα περιβάλλον, με σκοπό την λήψη της μέγιστης δυνατής ανταμοιβής. Αυτή η βέλτιστη συμπεριφορά αποκτάται μέσω της αλληλεπίδρασης με το περιβάλλον και την παρακολούθηση των αποτελεσμάτων.

   Η διαδικασία της ενισχυτικής μάθησης έχει αρκετές ομοιότητες με την ανθρώπινη μάθηση. Για παράδειγμα, ένα παιδί μαθαίνει διάφορες δεξιότητες και γνώσεις μέσα από τις καθημερινές του εμπειρίες. Ένα παράδειγμα ενισχυτικής μάθησης μπορεί να είναι ένα παιχνίδι, όπου το παιχνίδι αποτελεί το περιβάλλον, οι ενέργειες ενός παίκτη σε κάθε γύρο καθορίζουν τις καταστάσεις και ο στόχος του παίκτη είναι να επιτύχει το υψηλότερο δυνατό σκορ. Ο παίκτης λαμβάνει ανατροφοδότηση σε μορφή ανταμοιβής και τιμωρίας καθώς παίζει το παιχνίδι. (*Types of Machine Learning - Javatpoint*, 2021)



Εικόνα 5. Μοντέλο Ενισχυτικής Μάθησης

(Kozan, M., 2021)

Οι αλγοριθμικές τεχνικές που χρησιμοποιούνται σε προβλήματα ενισχυτικής μηχανικής μάθησης είναι οι παρακάτω:

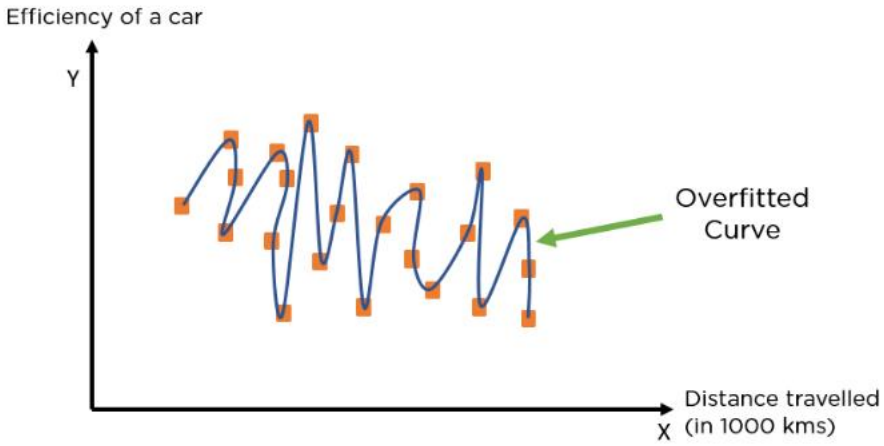
- Q-Learning
- Deep Q Network (DQN)
- Πολιτικές Κατά βάρος (Policy Gradients)

2.1.4 Προβλήματα Προσαρμογής στα μοντέλα μηχανικής μάθησης

Σε ένα μοντέλο μηχανικής μάθησης σκοπός είναι να πετύχουμε υψηλή ακρίβεια αποτελεσμάτων στις προβλέψεις και αποδοτικότητα. Η υπερπροσαρμογή (overfitting) και η υποπροσαρμογή (underfitting) αναδεικνύονται ως δύο κεντρικές έννοιες στον χώρο της μηχανικής μάθησης και αποτελούν τις κυρίαρχες πηγές ανεπαρκούς απόδοσης ενός μοντέλου.

2.1.4.1 Υπερπροσαρμογή (Overfitting)

Όταν ένα μοντέλο έχει εξαιρετική απόδοση στα δεδομένα εκπαίδευσης, αλλά παρουσιάζει χαμηλή απόδοση σε δεδομένα ελέγχου ή δοκιμής (νέα δεδομένα), βρίσκεται σε κατάσταση υπερπροσαρμογής. Αυτό συμβαίνει όταν το μοντέλο μάθει τα δεδομένα εκπαίδευσης τόσο καλά, που αδυνατεί να γενικεύσει αποτελεσματικά σε νέα δεδομένα. Συνήθως, αυτή η υπερπροσαρμογή προκαλείται από υπερβολική πολυπλοκότητα του μοντέλου.



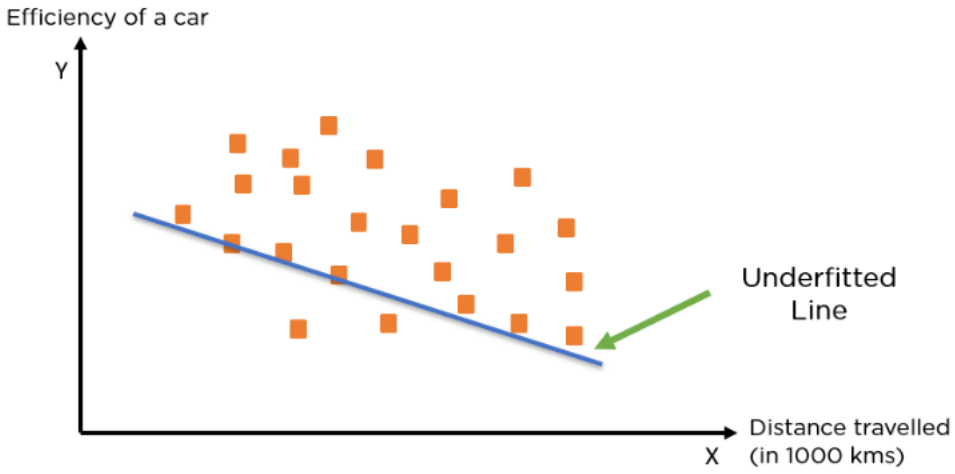
Εικόνα 6: Διάγραμμα Υπερπροσαρμογής του μοντέλου (Overfitting)

(Biswal, A. ,2021)

2.1.4.2 Υποπροσαρμογή (Underfittng)

Όταν ένα μοντέλο δεν έχει καταφέρει να μάθει αποτελεσματικά τα μοτίβα στα δεδομένα εκπαίδευσης και αποτυγχάνει να γενικεύσει καλά σε νέα δεδομένα, βρισκόμαστε μπροστά σε ένα πρόβλημα υποπροσαρμογής. Αυτό σημαίνει ότι το μοντέλο δεν έχει αποκτήσει επαρκή κατανόηση των παρατηρούμενων μοτίβων και δεν είναι σε θέση να προβλέψει αποτελεσματικά. Η υποπροσαρμογή είναι συνήθως αποτέλεσμα της χρήσης πολύ απλών μοντέλων που εφαρμόζουν υπερβολικά απλοποιημένες υποθέσεις.

Τα μοντέλα υποπροσαρμογής εμφανίζουν κακή απόδοση τόσο στα δεδομένα εκπαίδευσης όσο και στα νέα δεδομένα, καθιστώντας τα αναξιόπιστα στις προβλέψεις τους.



Εικόνα 7: Διάγραμμα Υποπροσαρμογής του μοντέλου (Underfitting)

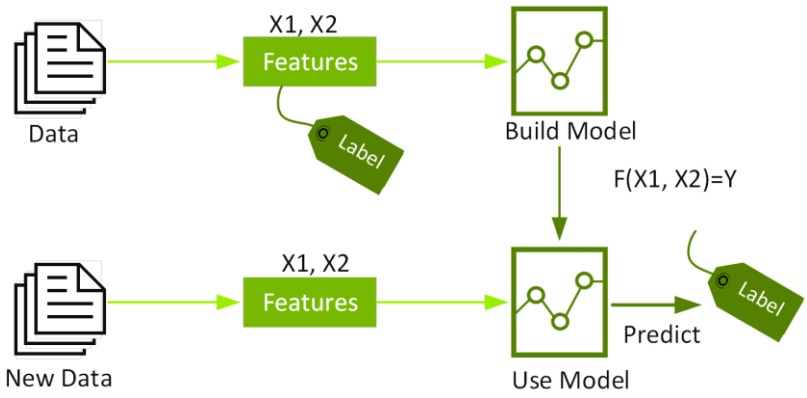
(Biswal, A. ,2021)

2.2 Ο αλγόριθμος XGBOOST

Ο αλγόριθμος XGBoost, που σημαίνει Extreme Gradient Boosting, είναι μια επεκτάσιμη και κατανεμημένη βιβλιοθήκη μηχανικής εκμάθησης με ενισχυμένη κλίση δέντρων απόφασης, γνωστή ως gradient-boosted decision trees (GBDT). Για την πλήρη κατανόηση του XGBoost, είναι σημαντική η εξοικείωσης με τις βασικές έννοιες και τους αλγορίθμους μηχανικής μάθησης που βασίζεται ο XGBoost:

- εποπτευόμενη μάθηση,
- δέντρα απόφασης,
- εκμάθηση συνόλου και
- ενίσχυση κλίσης.

Στην **εποπτευόμενη μηχανική** εκμάθηση, χρησιμοποιούνται αλγόριθμοι για την εκπαίδευση ενός μοντέλου, το οποίο αναζητά μοτίβα σε ένα σύνολο δεδομένων που έχει ετικέτες και χαρακτηριστικά. Αφού το μοντέλο εκπαιδευτεί, χρησιμοποιείται για την πρόβλεψη των ετικετών σε ένα νέο σύνολο δεδομένων βάσει των χαρακτηριστικών που περιέχει.

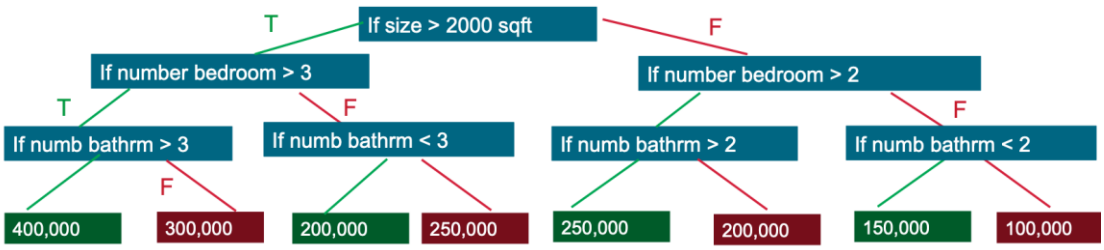


Εικόνα 8: Διαδικασία Εποπτευόμενης Μηχανικής Μάθησης

("What is XGBoost?", n.d.)

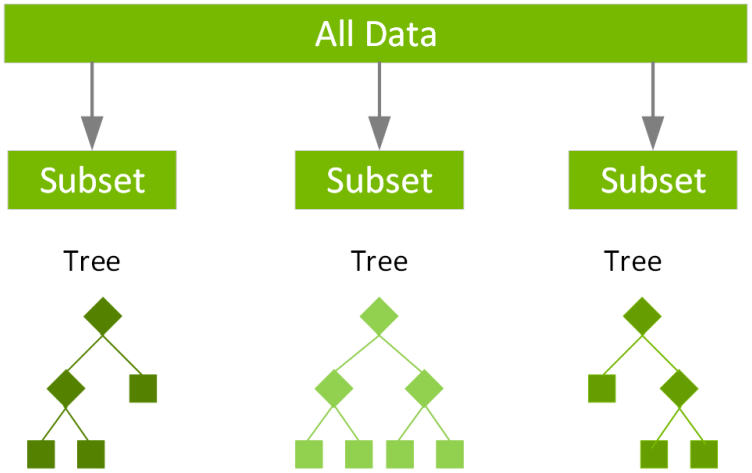
Τα **δέντρα αποφάσεων** δημιουργούν ένα μοντέλο που προβλέπει την ετικέτα αξιολογώντας ένα δέντρο ερωτήσεων σχετικές με τα χαρακτηριστικά if-then-else true/false και εκτιμώντας τον ελάχιστο αριθμό ερωτήσεων που απαιτούνται για την αξιολόγηση της πιθανότητας λήψης μιας σωστής απόφασης.

Η διαδικασία δημιουργίας του δέντρου ξεκινάει με μια βασική ερώτηση στην κορυφή και ακολουθεί ανακατεύθυνση προς τα κάτω, ακολουθώντας τα κλαδιά του δέντρου με κάθε νέα απάντηση. Κάθε ερώτηση παρέχει περισσότερες λεπτομέρειες για τα χαρακτηριστικά, καταλήγοντας σε έναν κόμβο (leaf node) όπου εκεί βρίσκεται η τελική απάντηση (ετικέτα/label).



Εικόνα 9: Παράδειγμα δέντρου απόφασης: εκτίμηση της τιμής του σπιτιού (η ετικέτα) με βάση το μέγεθος και τον αριθμό των υπνοδωματίων (τα χαρακτηριστικά). ("What is XGBoost?", n.d.)

Τα δέντρα απόφασης **ενισχυμένης κλίσης** (GBDT) είναι ένας αλγόριθμος εκμάθησης συνόλου δέντρων αποφάσεων παρόμοιος με το τυχαίο δάσος (Random Forest). Τόσο το Random Forest όσο και το GBDT δημιουργούν ένα μοντέλο που αποτελείται από πολλαπλά δέντρα απόφασης.



Εικόνα 10: Πολλαπλά δέντρα απόφασης

("What is XGBoost?", n.d.)

Η κύρια διαφορά είναι ο τρόπος με τον οποίο χτίζονται και συνδυάζονται τα δέντρα στο Random Forest και στα Gradient Boosted Decision Trees (GBDT). Στο Random Forest, τα δέντρα αποφάσεων αναπτύσσονται παράλληλα από τυχαία δείγματα του συνόλου δεδομένων, και η τελική πρόβλεψη προκύπτει από τον μέσο όρο των προβλέψεων όλων των δέντρων. Αντίθετα, στα GBDT, τα δέντρα αναπτύσσονται σειριακά, με κάθε νέο δέντρο που βοηθά στη διόρθωση των σφαλμάτων του προηγούμενου. Στην αρχή του GBDT, όλες οι περιπτώσεις έχουν ίσα βάρη. Ένα πρώτο μοντέλο εκπαιδεύεται και, ανάλογα με την απόδοσή του, ενημερώνονται τα βάρη των περιπτώσεων. Όσο καλύτερη είναι η πρόβλεψη, τόσο μικρότερο είναι το βάρος για αυτές τις περιπτώσεις στον επόμενο γύρο. Η τελική πρόβλεψη προκύπτει από ένα σταθμισμένο άθροισμα όλων των προβλέψεων των δέντρων.

2.3 Ερμηνευσιμότητα Μηχανικής Μάθησης

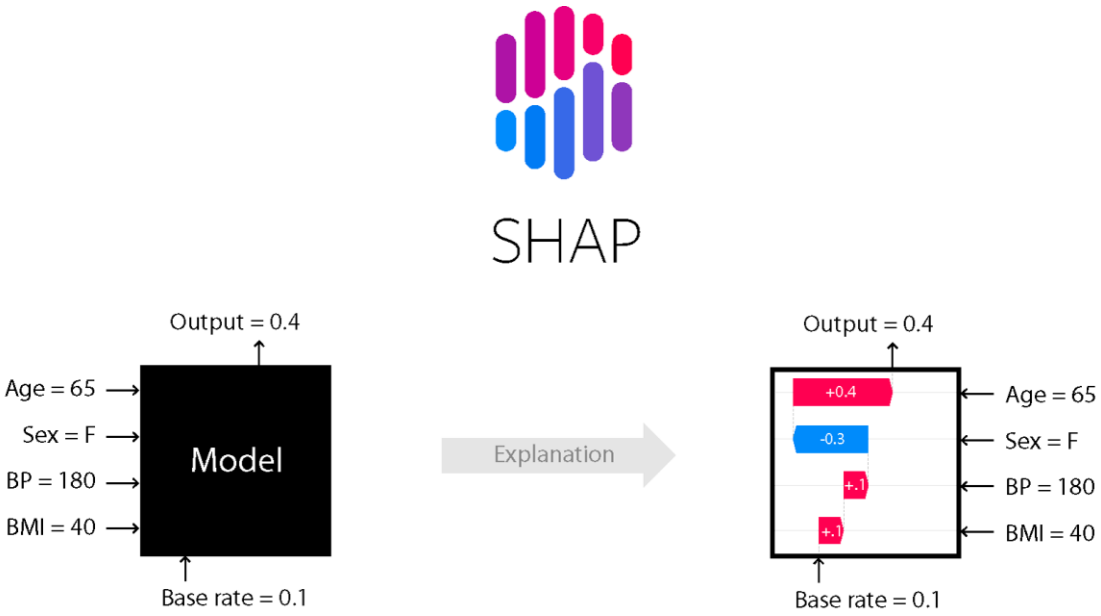
Τα μοντέλα μηχανικής μάθησης συχνά εμφανίζονται ως μαύρα κουτιά, δυσκολεύοντας την κατανόηση του πώς λαμβάνουν αποφάσεις. Για να αντιμετωπίσουμε αυτήν τη δυσκολία και να κατανοήσουμε τα κύρια χαρακτηριστικά που επηρεάζουν την απόδοση του μοντέλου, απαιτούνται εξηγήσιμες τεχνικές μηχανικής μάθησης που αναλύουν ορισμένες από τις ασάφειες.

Κάποιες από τις πιο γνωστές τεχνικές ερμηνευσιμότητας και κατανόησης μοντέλων μηχανικής μάθησης περιλαμβάνουν τη μέθοδο AI LIME, Integrated Gradient (IG) κ.α., και φυσικά τη μέθοδο SHAP, η οποία επιλέχθηκε για την παρούσα εργασία και θα εξηγηθεί αναλυτικά παρακάτω.

Με μια σύντομη αναφορά στην μέθοδο AI LIME, χαρακτηρίζεται ως ένα εργαλείο που επεξηγεί μια συγκεκριμένη περίπτωση, είναι δηλαδή κατάλληλη για τοπικές επεξηγήσεις. Το LIME χειρίζεται τα δεδομένα εισόδου και δημιουργεί εκ νέου μια σειρά τεχνητών δεδομένων που περιλαμβάνουν μόνο ένα μέρος των αρχικών χαρακτηριστικών. Συνεπώς, για παράδειγμα, στην περίπτωση δεδομένων κειμένου, από το αρχικό κείμενο αφαιρείται ένας συγκεκριμένος αριθμός διαφορετικών, τυχαία επιλεγμένων λέξεων και παράγονται διαφορετικές εκδόσεις του αρχικού κειμένου. Έτσι, μέσω της παρουσίας ή απουσίας ορισμένων λέξεων-κλειδιών, μπορούμε να δούμε την επίδρασή τους στην ταξινόμηση του επιλεγμένου κειμένου. Η μέθοδος εξηγήσιμης τεχνητής νοημοσύνης AI LIME είναι συμβατή με πολλούς διαφορετικούς ταξινομητές και μπορεί να χρησιμοποιηθεί με δεδομένα κειμένου, εικόνας και πινάκων.

Το Integrated Gradient (IG) αποτελεί μια τεχνική ερμηνείας ή εξήγησης για βαθιά νευρωνικά δίκτυα, παρουσιάζοντας τη σημασία του κάθε χαρακτηριστικού εισόδου στη συνολική πρόβλεψη του μοντέλου. Αυτή η μέθοδος εφαρμόζεται σε διάφορα είδη μοντέλων, όπως εικόνες, κείμενο ή δομημένα δεδομένα. Το Integrated Gradient βασίζεται σε δύο θεμελιώδη αξιώματα που πρέπει να τηρούνται: Ευαισθησία (Sensitivity) και Αμετάβλητο Εφαρμογής (Invariance to Baseline). Η ευαισθησία εξασφαλίζει ότι το IG ανταποκρίνεται στις μικρές μεταβολές των χαρακτηριστικών, ενώ η αμετάβλητο εφαρμογή εξασφαλίζει τη συνέπεια των εξηγήσεων ανεξάρτητα από το αρχικό σημείο αναφοράς.

Μια ακόμη από τις τεχνικές ερνηνευσιμότητας είναι η μέθοδος SHAP (**SH**apley **A**dditive ex**P**lanations). Βασίζεται στη θεωρία παιγνίων και χρησιμοποιείται για την αύξηση της διαφάνειας και την καλύτερη κατανόηση των μοντέλων μηχανικής μάθησης. Μια εύστοχη παρομοίωση είναι ένα παιχνίδι συνεργασίας με τον ίδιο αριθμό παικτών, όπου κάθε παίχτης αντιστοιχεί σε ένα χαρακτηριστικό. Το SHAP θα αποκαλύψει την ατομική συνεισφορά κάθε παίκτη (ή χαρακτηριστικό) στην έξοδο του μοντέλου, για κάθε παράδειγμα ή παρατήρηση.



Εικόνα 11: SHAP (Shapley Additive explanations)

(Welcome to the SHAP documentation — SHAP latest documentation)

Οι τιμές SHAP βασίζονται σε τιμές Shapley από τη θεωρία παιγνίων. Στη θεωρία παιγνίων, οι τιμές Shapley βοηθούν στον προσδιορισμό του πόσο έχει συνεισφέρει κάθε παίκτης σε ένα παιχνίδι συνεργασίας στη συνολική πληρωμή. Για ένα μοντέλο μηχανικής μάθησης, κάθε χαρακτηριστικό θεωρείται «παίκτης». Συγκεκριμένα, οι τιμές SHAP υπολογίζονται συγκρίνοντας τις προβλέψεις ενός μοντέλου με και χωρίς παρόν συγκεκριμένο χαρακτηριστικό. Αυτό γίνεται επαναληπτικά για κάθε χαρακτηριστικό και κάθε δείγμα στο σύνολο δεδομένων.

Η βιβλιοθήκη SHAP παρέχει ποικιλία γραφημάτων όπως Summary Plot (beeswarm, violin,) Waterfall plot, Force plot κα.

2.4 Η Γλώσσα Προγραμματισμού Python

Η Python αποτελεί μια γλώσσα προγραμματισμού που εφαρμόζεται ευρέως στη δημιουργία ιστοσελίδων, ανάπτυξη λογισμικού, αυτοματοποίηση εργασιών και ανάλυση δεδομένων. Πρόκειται για μια πολύπλευρη γλώσσα προγραμματισμού που μπορεί να χρησιμοποιηθεί για τη δημιουργία διαφορετικών ειδών προγραμμάτων και δεν περιορίζεται σε συγκεκριμένες εφαρμογές. Η ευελιξία αυτή, σε συνδυασμό με την φιλικότητά της για αρχάριους, έχει καταστήσει τη Python μία από τις πλέον δημοφιλείς γλώσσες προγραμματισμού στη σύγχρονη εποχή.

Παρακάτω επισημαίνονται μερικοί από τους παράγοντες που καθιστούν την Python ένα ισχυρό εργαλείο που η χρήση του επεκτείνεται σε πολλούς τομείς, από τον ακαδημαϊκό και τον επιχειρησιακό χώρο, καθιστώντας της προνομιούχα επιλογή για πολλούς χρηστές.

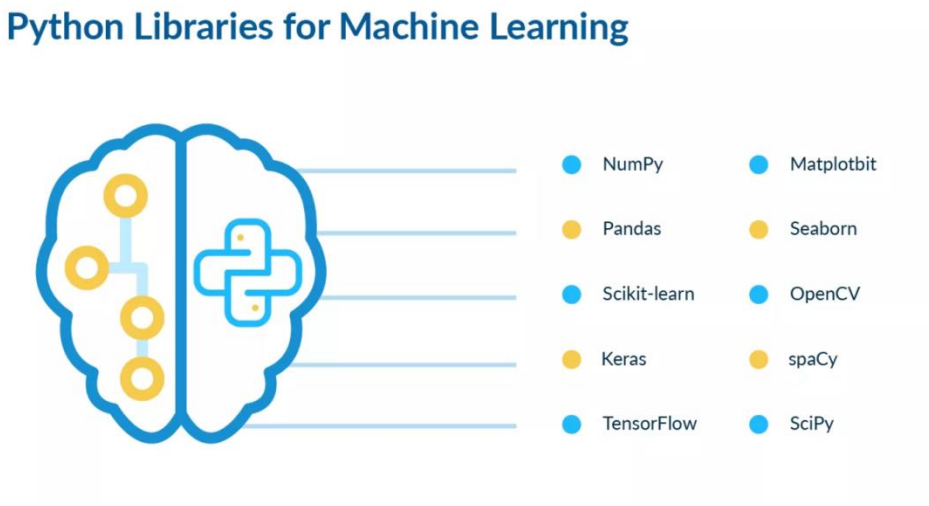
- **Ευκολία σύνταξης και ανάγνωσης:** Η Python είναι απλή γλώσσα και μοιάζει με τα καθημερινά αγγλικά, καθώς χρησιμοποιεί λέξεις-κλειδιά. Αυτό μειώνει τον χρόνο που χρειάζονται οι προγραμματιστές για να μάθουν και να κατανοήσουν τη σύνταξη και να την εφαρμόσουν.
- **Διαθεσιμότητα Βιβλιοθηκών:** Υπάρχει μια πληθώρα βιβλιοθηκών και πλατφορμών που υποστηρίζουν την Python, που διευκολύνουν την ανάπτυξη και την επέκταση λογισμικού.
- **Ανεξαρτησία από την πλατφόρμα:** Ο κώδικας Python μπορεί να εκτελεστεί σε διαφορετικές πλατφόρμες όπως Windows, Mac, UNIX και Linux.
- **Κοινότητα Ανοιχτού κώδικα:** Οι χρήστες μπορούν να συνεισφέρουν στην βελτίωση της και να δημιουργήσουν δίκες τους εφαρμογές και εργαλεία.
- **Δυνατότητα επεκτασιμότητας:** Οι χρήστες μπορούν να προσθέσουν λειτουργικές μονάδες χαμηλού επιπέδου (όπως δημιουργία προσαρμοσμένων βιβλιοθηκών, επέκταση της γλώσσας με νέα χαρακτηριστικά ή ενσωμάτωση κώδικα που γράφτηκε σε άλλες γλώσσες προγραμματισμού C/C++) στον διερμηνέα Python για να προσαρμόσουν και να βελτιστοποιήσουν τα εργαλεία τους.

2.2.1 Χρήση Python για Μηχανική Μάθηση και Επιστήμη Δεδομένων

Η Python αποτελεί την προτιμώμενη γλώσσα προγραμματισμού για επιστήμονες και ερευνητές που εργάζονται με δεδομένα. Η διαχείριση και ανάλυση των δεδομένων μπορεί να είναι χρονοβόρα για αυτούς. Με την συμβολή της Python δίνεται η δυνατότητα δημιουργίας ενός ευρέως φάσματος ποικίλων οπτικοποιήσεων, όπως γραφήματα γραμμών, ράβδων, γραφήματα πίτας, ιστογράμματα και τρισδιάστατες αναπαραστάσεις δεδομένων.

Επιπλέον, η Python κατατάσσεται ανάμεσα στις κορυφαίες γλώσσες για την εκπαίδευση μοντέλων μηχανικής μάθησης (ML). Αυτά τα μοντέλα, χρησιμοποιώντας συγκεκριμένους αλγορίθμους, είναι σε θέση να αναλύσουν δεδομένα και να αναγνωρίσουν μοτίβα για να προβλέψουν αποτελέσματα ή να λάβουν αποφάσεις βάσει αυτών των δεδομένων. Επιπλέον, είναι σε θέση να προσαρμοστούν συνεχώς με βάση τα αποτελέσματα προηγούμενων δεδομένων για να ανταποκριθούν σε νέες μεταβλητές

Επομένως, επιστήμονες δεδομένων και προγραμματιστές που εκπαιδεύουν μοντέλα ML συχνά χρησιμοποιούν βιβλιοθήκες όπως το NumPy, το Pandas και το Matplotlib για την αυτοματοποίηση λειτουργιών όπως ο καθαρισμός, ο μετασχηματισμός των δεδομένων και η οπτικοποίησή τους.



Εικόνα 12: Βιβλιοθήκες Python

(Inc, S. , 2021)

2.5 Jupiter Notebook

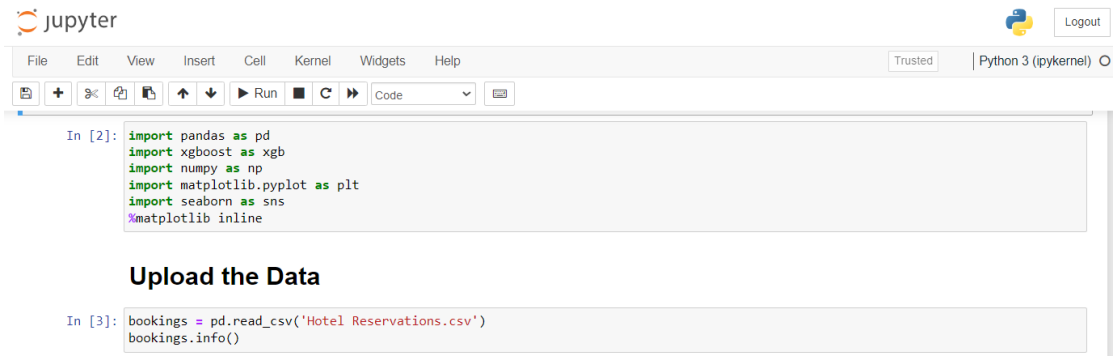
2.5.1 Ορισμός και Χαρακτηριστικά

Το Jupyter Notebook είναι η πρωτότυπη διαδικτυακή εφαρμογή για τη δημιουργία και την κοινή χρήση υπολογιστικών εγγράφων. Αποτελεί ένα διαδραστικό εργαλείο που προσφέρει στους χρήστες την δυνατότητα να εκτελούν και να αναλύουν κώδικα, παρακολουθώντας την έξοδο του και διαμορφώνοντας τον κώδικα ώστε να επιτύχουν το επιθυμητό αποτέλεσμα. Το Jupyter Notebook διαθέτει ανεξάρτητα κελιά που μπορούν να εκτελεστούν σε οποιαδήποτε σειρά, επιτρέποντας τη δημιουργία εύκολα δομημένων και αναλυτικών αναφορών. Αξίζει να σημειωθεί ότι το σημειωματάριο αποθηκεύεται σε μορφή αρχείου με κατάληξη ".ipynb", προσφέροντας τη δυνατότητα αποθήκευσης και επαναφόρτωσης του έργου για μελλοντική ανάλυση και συνεχή συνεργασία.

Μερικά χαρακτηριστικά του Jupiter Notebook αποτελούν:

- Υποστήριξη περισσότερων από 40 γλώσσες προγραμματισμού, συμπεριλαμβανομένης της Python.
- Εύκολη κοινοποίηση και διαμοιρασμός των notebooks με άλλους χρήστες αλλά και ομάδες χρηστών, χρησιμοποιώντας email, GitHub, Dropbox και Jupiter Notebook Viewer.
- Δημιουργία πλούσιων και διαδραστικών αποτελεσμάτων, καθώς υπάρχει η δυνατότητα ενσωμάτωσης HTML, LaTeX, εικόνας, Βίντεο και προσαρμοσμένους τύπους MIME.

2.5.2 Περιβάλλον εργασίας Jupiter Notebook



Εικόνα 13: Επιφάνεια Εργασίας Jupiter Notebook

Κατά τη χρήση ενός ανοιχτού σημειωματαρίου, παρατηρούμε την εμφάνιση δύο βασικών στοιχείων: τη γραμμή μενού και τη γραμμή εργαλείων. Η γραμμή μενού περιλαμβάνει διάφορες επιλογές που μπορούν να χρησιμοποιηθούν για τον έλεγχο της λειτουργίας του σημειωματαρίου. Από την άλλη, η γραμμή εργαλείων παρέχει γρήγορη πρόσβαση στις πιο συχνά χρησιμοποιούμενες λειτουργίες μέσω εικονιδίων.

Ένα σημαντικό στοιχείο που βρίσκεται τόσο στη γραμμή μενού όσο και στη γραμμή εργαλείων είναι ο πυρήνας (kernel). Ο πυρήνας είναι ουσιαστικά η "υπολογιστική μηχανή" που χρησιμοποιείται για την εκτέλεση του κώδικα που περιέχεται στο σημειωματάριο. Για παράδειγμα, ο πυρήνας ipkernel, που αναφέρεται σε αυτόν τον οδηγό, χρησιμοποιείται για την εκτέλεση κώδικα Python.

Όταν ανοίγετε ένα σημειωματάριο, ο σχετικός πυρήνας εκκινείται αυτόματα. Ωστόσο, υπάρχει και η πιθανότητα ότι θα χρειαστεί να επανεκκινήσετε τον πυρήνα κατά τη διάρκεια της χρήσης, ανάλογα με τις ανάγκες σας ή τις ενδεχόμενες δυσλειτουργίες. Αυτή η επανεκκίνηση του πυρήνα είναι μια συνήθης ενέργεια που μπορεί να απαιτηθεί για να διασφαλιστεί η ομαλή λειτουργία του σημειωματαρίου και η σωστή εκτέλεση του κώδικα.

2.6 Πλατφόρμα Kaggle

Το Kaggle είναι θυγατρική της Google που λειτούργει ως κοινότητα για επιστήμονες και προγραμματιστές. Θέματα ενδιαφέροντος είναι η μηχανική μάθηση και άλλες σύγχρονες τεχνολογίες. Η κοινότητα του Kaggle απαρτίζεται πλέον με πάνω από ένα εκατομμύρια μέλη χρήστες, οι οποίοι μπορούν να αποκτήσουν πρόσβαση σε διαθέσιμα σύνολα δεδομένων από 194 διαφορετικές χώρες σε όλο τον κόσμο και να συζητήσουν για την ανάπτυξη μοντέλων με άλλα μέλη της κοινότητας.

Η πλατφόρμα του Kaggle, έχει θεσπίσει διαγωνισμούς που παρέχουν σε όσους καταφέρουν να διακριθούν πόντους στην Γενική Παγκόσμια κατάταξη των επιστημόνων αλλά και χρηματική ανταμοιβή. Μεγάλες εταιρίες όπως η AMAZON, HP, CERN κοκ παρέχουν έναν μεγάλο όγκο ανοιχτών δεδομένων στο Kaggle, με

σκοπό την ανάλυση και επεξεργασία αυτών και την δημιουργία μοντέλων πρόβλεψης σε υπαρκτά προβλήματα που αντικρίζει η επιχείρηση.

Το περιβάλλον του Kaggle πέρα από τους διαγωνισμούς, προτιμάται και για προσωπική χρήση. Απαρτίζεται από πολλά οφέλη, μερικά από τα οποία θα αναφερθούν στην συνέχεια, που το καθιστούν μια από τις πρώτες επιλογές των χρηστών.

Η χρήση όλων των δυνατοτήτων της πλαρφόρμας Kaggle είναι δωρεάν. Δημιουργώντας ένα λογαριασμό, οι χρήστες έχουν πρόσβαση στους διακομιστες της πλατφόρμας, στα ανοιχτά σύνολα δεδομένων κ.α. χωρίς καμία επιπλέον χρέωση.

Διαθέτει Cloud Computing, δηλαδή πυρήνες του Kaggle που μπορούν να χρησιμοποιηθούν για την εκτέλεση του κώδικα από οπουδήποτε, με προϋπόθεση να υπάρχει σύνδεση στο διαδίκτυο.

Τα περισσότερα πακέτα Python είναι ήδη εγκατεστημένα στο Kaggle Notebook, επομένως επιταχύνεται η εξοικονόμηση χρόνου και κώδικα που θα απαιτούνταν για την εγκατάσταση τους.

3. Βιβλιογραφική Επισκόπηση (Related Work)

3.1 Βιβλιογραφική επισκόπηση ερευνών στον τομέα των ξενοδοχειακών κρατήσεων

Ο τουρισμός είναι ένας από τους πιο αναπτυγμένους κλάδους στον κόσμο και η σημασία του στην παγκόσμια οικονομία δεν προκαλεί αμφιβολίες. Απόδειξη της ανάπτυξής του μπορεί να φανεί αν σκεφτεί κανείς ότι το 1990 ο αριθμός των διεθνών τουριστών ξεπέρασε ελαφρώς τα 400 εκατομμύρια, ενώ το 2017 ο αριθμός αυτός αυξήθηκε σε 1300 εκατομμύρια.

Υπάρχει έντονη αλληλεξάρτηση ανάμεσα στην πρόβλεψη της ζήτησης και τη διαχείριση των εσόδων. Καθώς τα ξενοδοχεία αναγκάζονται να διαχειρίζονται την πληρότητα των δωματίων σε ένα περιβάλλον αβεβαιότητας, εκτίθενται σε ασαφείς προβλέψεις εσόδων και αναγκάζονται να αναλαμβάνουν επιχειρηματικούς κινδύνους. Επομένως η ανάγκη για αποτελεσματική διαχείριση της ζήτησης γίνεται προφανής, καθώς η προσαρμογή στην ζήτηση επηρεάζει άμεσα τα εισοδήματα των ξενοδοχείων.

Σε σχετική μελέτη το 2019, κατασκευάστηκε ένα πρότυπο μοντέλο, βασισμένο σε αυτοματοποιημένο σύστημα μηχανικής μάθησης και εφαρμόστηκε σε 2 ξενοδοχεία. (Antonio et al., 2019) .Επιχειρηματικά, το μοντέλο απέδειξε την αποτελεσματικότητά του, επιτυγχάνοντας αποτελέσματα που υπερβαίνουν το 84% στην ακρίβεια (Accuracy). Το σύστημα επέτρεπε στα ξενοδοχεία να προβλέπουν την καθαρή τους ζήτηση, επιτρέποντας τους να λαμβάνουν καλύτερες αποφάσεις σχετικά με το ποιες κρατήσεις να αποδεχτούν ή να απορρίπτουν, ποιες τιμές να ορίσουν και πόσα δωμάτια να διαθέσουν για υπερπώληση. Η δυνατότητα συστηματικής πρόβλεψης

κρατήσεων με υψηλή πιθανότητα ακύρωσης επέτρεψε στα ξενοδοχεία να μειώσουν τα ποσοστά ακυρώσεων κατά 37%, λαμβάνοντας μέτρα για την αποφυγή των ακυρώσεων.

Για την επίτευξη αυτού χρησιμοποιήθηκε ο αλγόριθμος XGBoost (machine learning gradient tree boosting algorithm) για να χτίσει το μοντέλο ταξινόμησης για την πρόβλεψη κάθε ακύρωσης κράτησης. Για την εκτίμηση των παραμέτρων του μοντέλου, εφαρμόστηκε ένας συνδυασμός δύο πολύ γνωστών τεχνικών αναζήτησης πλέγματος (Grid Search)και τυχαίας αναζήτησης (Random Search). Οι τιμές παραμέτρων επιλέχθηκαν από το μοντέλο με την καλύτερη απόδοση, το οποίο προέκυψε από συνολικά 100 επαναλήψεις της διαδικασίας Cross validations με 10 φακέλους και μέγιστο αριθμό δέντρων απόφασης 200.

Το 2020, υλοποιήθηκε μια μελέτη σύγκρισης ανάμεσα στην απόδοση τριών διαφορετικών τύπων μοντέλων μηχανικής μάθησης, με βάση το F-score, με σκοπό την πρόβλεψη ακυρώσεων στις ξενοδοχειακές κρατήσεις. (Sánchez-Medina & C-Sánchez, 2020) Τα μοντέλα που χρησιμοποιήθηκαν ήταν

- SVM (Support Vector Machine), αυτή η τεχνική προσπαθεί να καθορίσει μια οριακή επιφάνεια μεταξύ δύο τάξεων σύμφωνα με τα χαρακτηριστικά των δεδομένων.
- Δέντρα Απόφασης (Decision Trees), η προσέγγιση τους αποτελείται από εξαγωγή μοτίβων και προτύπων από ένα δεδομένο σύνολο δεδομένων. Υπάρχουν πολλές κατηγορίες δέντρων αποφάσεων. Στην συγκεκριμένη μελέτη επιλέγονται 2 από τις πιο δημοφιλείς, το τυχαίο δάσος (random forest) και ο C5.0.
- ANN (Artificial Neural Network - Τεχνητό Νευρωνικό Δίκτυο), τα οποία είναι πολύπλοκα μοντέλα μηχανικής μάθησης αποτελούμενα από ένα συγκεκριμένο αριθμό απλών υπολογιστικών κυττάρων, γνωστά ως νευρώνες συνδεδεμένα μεταξύ τους με πολύπλοκές δομικές πτυχές.

Σύμφωνα με την μελέτη αυτή, το SVM φαίνεται να έχει τη χαμηλότερη τιμή F-score σε σύγκριση με τα άλλα μοντέλα, ενώ τα ANN φαίνεται να έχουν τη υψηλότερη τιμή F-score. Επιπλέον, η έρευνα αποκαλύπτει τη δυνατότητα πρόβλεψης ακυρώσεων με υψηλή ακρίβεια, χρησιμοποιώντας έναν περιορισμένο αριθμό μεταβλητών. Συγκεκριμένα, χρησιμοποιούνται 13 ανεξάρτητες μεταβλητές, ένας αριθμός σημαντικά χαμηλότερος σε σύγκριση με προηγούμενες έρευνες. Διοικητικά, τα επιτευχθέντα αποτελέσματα υποδεικνύουν ότι τα ιστορικά αρχεία ενός πελάτη αποτελούν ουσιαστική πτυχή για τις επιχειρήσεις φιλοξενίας και πρέπει να αντιμετωπίζονται ως κρίσιμο περιουσιακό στοιχείο.

Μια άλλη προσέγγιση το 2020, δεδομένου ότι η εξαρτημένη μεταβλητή «Ακυρωση/Cancelled» είναι διάδικο χαρακτηριστικό, μελέτησε τα αποτελέσματα 8 μοντέλων δυαδικής ταξινόμησης: Λογιστική Παλινδρόμηση, Naïve Bayes, Κοντινότεροι Γείτονες (K-Nearest Neighbors), SVM, δέντρα απόφασης, τυχαίο

δάσος(Random Forest) , Gradient Boosting Machines, Extreme Gradient Boost. Τα μοντέλα αξιολογήθηκαν με βάση τη μέση ακρίβεια, το F1 και το score. Η χαμηλότερη τιμή ακρίβειας με 87,17% αντιπροσωπεύει το μοντέλο Naïve Bayes, ενώ όλα τα υπόλοιπα μοντέλα έχουν τιμή ακρίβειας που αγγίζει το 99%. Όσον αφορά την ακρίβεια ξεχώρισαν τα μοντέλο δέντρου απόφασης και το μοντέλο Gradient Boost. Είναι σημαντικό να τονιστεί ότι η λογιστική παλινδρόμηση σχεδόν έφτασε τα αποτελέσματα άλλων πολύ πιο εξελιγμένων αλγορίθμων. Επομένως εφαρμόζοντας το βέλτιστο μοντέλο μηχανικής μάθησης σε μία ξενοδοχειακή μονάδα, είναι εφικτό να προβλεφθούν οι ακυρώσεις κρατήσεων κατά 99.93% και να σωθούν με αυτόν τον τρόπο 160.254,625 ευρώ ετησίως. (Timamopoulos, 2020)

Οι ακυρώσεις που γίνονται κοντά στην ώρα της εξυπηρέτησης είναι οι πιο επιζήμιες για τα ξενοδοχεία, διότι αφήνουν τη διοίκηση χωρίς χρόνο αντίδρασης. Η χρήση των Αρχείων Προσωπικών Ονομάτων (PNR- Personal Name Records) οδήγησε σε νέες προσεγγίσεις σε αυτόν τον τομέα. Παρα τη νέα αυτή ερευνητική περιοχή, δεν υπάρχουν έρευνες που να εστιάζουν στην πρόβλεψη για μεμονωμένες ακυρώσεις ξενοδοχείων κρατήσεων που πραγματοποιούνται κοντά στην ώρα της υποτιθέμενης κράτησης. Με στόχο να καλύψει αυτό το κενό, πραγματοποιήθηκε έρευνα με σκοπό να εντοπίσει τα άτομα που είναι πιθανό να κάνουν ακυρώσεις σε σύντομο χρονικό διάστημα, δηλαδή μεταξύ 4 εως 7 ημέρών νωρίτερα,οι οποίες μπορούν να θεωρηθούν «κρίσιμες ακυρώσεις», χρησιμοποιώντας τεχνικές Τεχνητής Νοημοσύνης (AI) μέσω δεδομένων PNR. Υπό αυτή την έννοια, το προτεινόμενο μοντέλο επιχείρησε να ανιχνεύσει άτομα που ενδέχεται να ακυρώσουν κοντά στο χρόνο υπηρεσίας, σύμφωνα με τα χαρακτηριστικά τους και τους ιστορικούς λόγους που έχουν για να αλλάξουν γνώμη. Για την εκπαίδευση του μοντέλου χρησιμοποιήθηκαν 13 ανεξάρτητες μεταβλητές, οι πιο συχνά ζητούμενες από τους διαδικτυακούς ισότοπους κρατήσεων καταλυμάτων φιλοξενίας, εθνικότητα, αριθμός διανυκτερεύσεων κα. Από το αρχικό σύνολο δεδομένων εκμαιεύτηκε και μία νέα ανεξάρτητη μεταβλητή «Σαββατοκύριακο», που όπως είναι φανερό αντιπροσωπεύει τον αριθμό ημερών του Σαββ/κου εντός του διαστήματος της κράτησης. Η ανάπτυξη του μοντέλου έγινε με χρήση διάφορων τεχνικών τεχνητής νοημοσύνης C5.0, SVM και ANN. Σύμφωνα με τα αποτελέσματα της έρευνας, ο δενδροειδής αλγόριθμος C5.0 δείχνει την καλύτερη έξοδο, και στην συνέχεια ακολουθούν SVM και ANN αντίστοιχα. (Sánchez et al.,2020)

Μεταξύ του κρίσιμου διαστήματος ημερών που επιλέχθηκε στην συγκεκριμένη μελέτη, φαίνεται ότι η ακρίβεια συναντάει χαμηλότερες τιμές και για τις 3 τεχνικές AI στις 4 ημέρες πριν την ημερομηνία κράτησης. Επιπλέον, πρέπει να σημειωθεί ότι αυτή η μελέτη έχει αναπτυχθεί χρησιμοποιώντας μόνο 13 μεταβλητές, σε σύγκριση με άλλες σχετικές ερευνητικές εργασίες, οι οποίες χρησιμοποιούν 37 μεταβλητές. Αυτή η μελέτη επέτρεψε επίσης να αποδείξουμε την αποτελεσματικότητα της χρήσης αλγορίθμων συνόλου για την πρόβλεψη μεμονωμένων ακυρώσεων κοντά στο χρόνο εξυπηρέτησης, οι οποίοι βελτιώνουν την AUC έως και 14% σε σχέση με τους μεμονωμένους ταξινομητές. Από τα πρακτικά συμπεράσματα, ξεχωρίζει η επιτυχής

χρήση τεχνικών Τεχνητής Νοημοσύνης σε βάσεις δεδομένων PNR. Η δυνατότητα πρόβλεψης ατόμων που ενδέχεται να ακυρώσουν λίγες ημέρες πριν την κράτηση έχει επιτευχθεί με ικανοποιητική ακρίβεια. Αντί να παρουσιάζουμε απλές αναλογίες πιθανών ακυρώσεων, τώρα μπορούμε να προσδιορίζουμε συγκεκριμένα άτομα που ενδέχεται να προβούν σε ακύρωση, προσφέροντας πιο συγκεκριμένα και χρήσιμα εργαλεία στον τομέα των κρατήσεων. (Sánchez et al.,2020)

Οι περισσότερες έρευνες τείνουν να επικεντρώνονται στη χρήση νευρωνικών δικτύων στην τουριστική βιομηχανία, ειδικότερα στην πρόβλεψη υψηλού επιπέδου ζήτησης. Παρ' όλα αυτά, λίγες μελέτες έχουν εξετάσει τη χρήση νευρωνικών δικτύων σε μεμονωμένα ξενοδοχεία, ή πιο λεπτομερή επίπεδα ζήτησης, κυρίως λόγω της υψηλής μεταβλητότητας και της πολυκλαδικής εποχικότητας. Η μελέτη που διεξήχθη το 2020 από τον Misuk Lee και τους συνεργάτες του, σκοπεύει να καλύψει αυτό το κενό. Η υψηλή μεταβλητότητα στη ζήτηση ξενοδοχείων, σε συνδυασμό με την εποχικότητα, δυσκολεύει συχνά την ακριβή πρόβλεψη της ημερήσιας ζήτησης με υψηλή ακρίβεια. Προκειμένου να αντιμετωπίστει η υψηλή μεταβλητότητα και η πολυσυλλεκτικότητα, εξετάστηκαν διάφορα νευρωνικά μοντέλα, συμπεριλαμβανομένων δεδομένων χρονοσειρών και εκ των προτέρων κρατήσεων. Επιπλέον, απόρροια της έρευνας είναι ότι τα συνδυασμένα μοντέλα νευρωνικών δικτύων είναι τα βέλτιστα για την ακρίβεια πρόβλεψης. Χρησιμοποιήθηκαν πραγματικά δεδομένα ξενοδοχείων, παρέχοντας εμπειρικά αποτελέσματα για το προτεινόμενο μοντέλο, συγκρίνοντάς το με παραδοσιακές μεθόδους πρόβλεψης. Τέλος για την επαλήθευση της αποτελεσματικότητας, διεξήχθει μια λεπτομερή σύγκριση μεταξύ των ικανοτήτων πρόβλεψης των νευρωνικών δικτύων και των παραδοσιακών μεθόδων. (Lee et al., 2020)

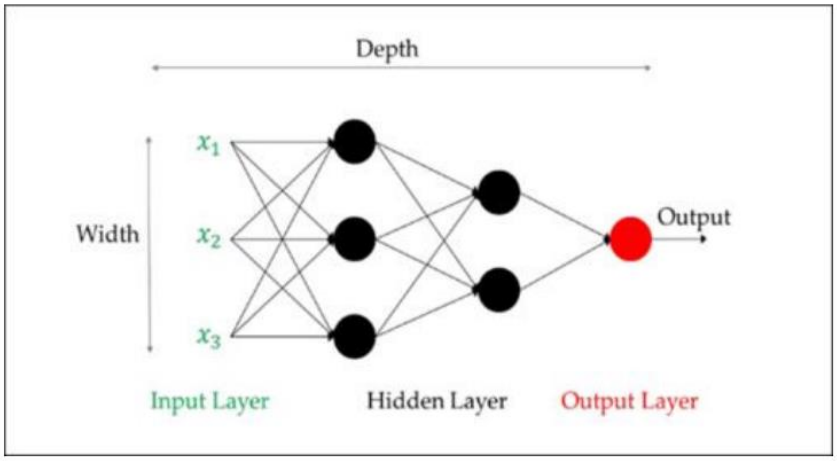
Πολλοί ερευνητές είχαν συγκεντρώσει το ενδιαφέρον τους στην εφαρμογή και διερεύνηση της τεχνητής νοημοσύνης και της μηχανικής μάθησης στον ξενοδοχειακό κλάδο την ίδια χρόνια 2020. Διαπιστώθηκε λοιπόν πως η μηχανική μάθηση είναι χρήσιμη στην πρόβλεψη της ζήτησης, την πρόβλεψη των τιμών, την πρόβλεψη ακύρωσης κρατήσεων, την οικονομική αποδοτικότητα και την αποδοτικότητα της εργασίας. Οι ερευνητές έχουν εξετάσει την εφαρμογή της μηχανικής μάθησης στον κλάδο των ξενοδοχείων, προκειμένου να επιτύχουν διάφορα αποτελέσματα. Για παράδειγμα, έχουν μελετήσει την αξιολόγηση της τοποθεσίας των ξενοδοχείων, τις ρομποτικές υπηρεσίες δωματίου, την πρόβλεψη της διατήρησης της κράτησης , τη διάκριση μεταξύ αληθινών και ψευδών online κριτικών, την πρόβλεψη της ενεργειακής κατανάλωσης, την πρόβλεψη ακυρώσεων κρατήσεων και την πρόβλεψη των τιμών δωματίων. Παράλληλα, ποιοτικές μελέτες έχουν εξετάσει την εφαρμογή της τεχνητής νοημοσύνης στον ξενοδοχειακό κλάδο, αναδεικνύοντας τα πλεονεκτήματα που προκύπτουν από τη χρήση της μηχανικής μάθησης για τα ξενοδοχεία. (ALOTAIBI, 2020)

Στην μελέτη που προηγήθηκε τονίζεται επίσης πως η μηχανική μάθηση υπερτερεί σημαντικά στην ακρίβειας πρόβλεψης έναντι των στατιστικών μοντέλων. Οι ερευνητές έχουν χρησιμοποιήσει τους αλγορίθμους μηχανικής μάθησης XGBoost,

Naive Bayes Classifier (NB), Maximum Entropy (ME), Generalized Linear Model (GLM), Multinomial Naive Bayes (MNB), Extreme learning machine (ELM), support vector regression (SVR), Boosted Regression Tree (BRT), Random Forest Regression (RFR), Natural language processing (NLP), Convolutional Neural Network-based Deep Learning (CNN-DL) και Nearest Neighbors. Ορισμένοι από τους αλγορίθμους μηχανικής μάθησης έχουν πλεονεκτήματα έναντι άλλων στον κλάδο των ξενοδοχείων στον τομέα των προβλέψεων. Για παράδειγμα, η ακρίβεια είναι 93,47% (ME), 87,82% (NB) και 86,34% (SVM). Οι μέθοδοι Gradient Boosting είναι πιο αποτελεσματικές από το SVM, ενώ γενικότερα οι αλγόριθμοι που ανήκουν στην κατηγορία μηχανικής μάθησης ξεπερνούν τα στατιστικά μοντέλα χρονοσειρών SARIMA (Seasonal Autoregressive Integrated Moving Average). (ALOTAIBI, 2020)

Οι ακυρώσεις κρατήσεων επηρεάζουν σημαντικά τις αποφάσεις του διοικητικού συμβουλίου στα ξενοδοχεία. Για να αντιμετωπίσουν αυτό το πρόβλημα, τα ξενοδοχεία χρησιμοποιούν αυστηρές πολιτικές ακύρωσης και τακτικές υπερβολικής κράτησης, που μπορεί, ωστόσο, να επηρεάσουν αρνητικά τα έσοδα και τη φήμη τους. Αντί αυτού προτείνεται η εφαρμογή μοντέλων βαθιάς μηχανικής μάθησης, ειδικότερα νευρωνικών δικτύων. Μια σχετική έρευνα το 2021 είχε στόχο να αξιολογήσει την απόδοση ενός βαθιού νευρωνικού δικτύου με δύο κατηγορίες ταξινόμησης (ακυρωμένη και μη). Το μοντέλο χρησιμοποιεί 13 ανεξάρτητες μεταβλητές, συμπεριλαμβανομένων των πιο συχνών παραγόντων που ερωτούνται οι πελάτες κατά την διαδικασία της κράτησης.

Ένα Βαθύ Νευρωνικό Δίκτυο (DNN) είναι ένα τεχνητό νευρωνικό δίκτυο που αποτελείται από πολλά επίπεδα. Γενικά, τα βαθιά νευρωνικά δίκτυα έχουν περισσότερα από 3 επίπεδα (επίπεδο εισόδου, Ν κρυφά επίπεδα, επίπεδο εξόδου). Κατά τη διάρκεια της διαδικασίας εκπαίδευσης, οι παράμετροι του DNN προσθέτουν τον βελτιστοποιητή (optimizer) και τον Ρυθμό μάθησης (Learning Rate). Ένας βελτιστοποιητής είναι ένας αλγόριθμος ή μια μέθοδος που χρησιμοποιείται για την αλλαγή των χαρακτηριστικών του νευρωνικού δικτύου, όπως το βάρος και ο ρυθμός μάθησης με σκοπό για τη μείωση των απωλειών. Η μεγαλύτερη τιμή ακρίβειας (Accuracy = 85,73%) βρέθηκε με τον συνδυασμό του Adadelta optimizer και Learning Rate 0,001.



Εικόνα 14: Βαθύ Νευρωνικό Δίκτυο (Adil et al., 2021)

Μία άλλη προσέγγιση το 2021 αντιμετώπισε το θέμα της ανισορροπίας στις προβλέψεις ακυρώσεων ξενοδοχείων, εισάγοντας μια τεχνική υπερδειγματοληψίας SMOTE-ENN για την αντιμετώπιση αυτού του ζητήματος. Η προτεινόμενη μεθοδολογία μπορεί να αντιμετωπίσει το πρόβλημα της ανισορροπίας στα σύνολα δεδομένων και να βελτιώσει τις προβλέψεις. (Adil et al., 2021)

Την ίδια χρόνια, το 2021, μελετήθηκε η ακρίβεια διαφόρων αλγορίθμων μηχανικής μάθησης με σκοπό την εύρεση αυτού που παράγει τις βέλτιστες προβλέψεις σχετικά με το πότε ένας πελάτης θα ακυρώσει την κράτηση του σε ένα ξενοδοχείο. (Novakovic, 2021) . Το σύνολο δεδομένων που χρησιμοποιήθηκε περιείχε 32 ανεξάρτητες μεταβλητές – χαρακτηριστικά. Μεταξύ των αλγορίθμων που συγκρίθηκαν ήταν : αλγόριθμοι λογιστικής παλινδρόμησης, αλγόριθμοι K-γείτονες, δέντρα απόφασης, bagging αλγόριθμοι (όπως το τυχαίο δάσος/Random Forest, SVM (Support Vector Machines)) και ταξινομητής AdaBoost (Adaptive Boosting).Το αποτέλεσμα της μελέτης, συγκρίνοντας όλους τους παραπάνω ταξινομητές στα δεδομένα δοκιμής, υποδεικνύει τον ταξινομητή bagging με την υψηλότερη ακρίβεια και τα καλύτερα αποτελέσματα στην πρόβλεψη ακυρώσεων.

Η σύγκριση αλγορίθμων μηχανικής μάθησης πάνω σε ξενοδοχειακά δεδομένα απασχόλησε ακόμη μια μελέτη εκείνη την χρόνια, εστιάζοντας στα μοντέλα RandomForest, ExtraTreeClassifier, Δέντρα αποφάσεων καθώς και στο XGBoost. (Nanang, 2021). Τα αποτελέσματα δείχνουν ότι τα RandomForest και ExtraTreeClassifier μοντέλα επιδεικνύουν υψηλότερους λόγους ακρίβειας σε σύγκριση με τους άλλους αλγορίθμους. Και τα δύο μοντέλα κατάφεραν να προβλέψουν σωστά το 88% όλων των θετικών ετικετών στο σύνολο δεδομένων που χρησιμοποιήθηκε στη μελέτη. Το συγκεκριμένο σύνολο δεδομένων αποτελούνταν από 29 ανεξάρτητες μεταβλητές, προσφέροντας ένα ευρύ φάσμα πληροφοριών για την εκπαίδευση των μοντέλων.

Το 2023 πραγματοποιήθηκε μια μελέτη σχετικά με τις ακυρώσεις κρατήσεων ξενοδοχείων υιοθετώντας το μοντέλο που ενσωματώνει BN (Bayes Network) και ένα

γραμμικό μοντέλο Μηχανικής Μάθησης, δηλαδή την παλινδρόμηση Lasso για την υποστήριξη της πρόβλεψης ακύρωσης ξενοδοχείου. (Chen et al., 2023). Τα BN ή αλλιώς τα Μαθηματικά Δίκτυα Βαθμονόμησης είναι μοντέλα πιθανοτήτων που αναπαριστούν τις σχέσεις μεταξύ των μεταβλητών. Το μοντέλο που κατασκευάστηκε συγκρίνεται με άλλους προηγμένους αλγορίθμους πρόβλεψης όπως ο XGBoost και ANN (Artificial Neural Network).Η προτεινόμενη αλληλεπίδραση Lasso- -Bayesian σημείωσε μεγαλύτερη ακρίβεια από τον XGB και την ANN-Bayesian αλληλεπίδραση.

Το 2023, δημοσιεύτηκε μια μελέτη με κύριο ερευνητικό στόχο την ανάπτυξη ενός υβριδικού μοντέλου μηχανικής εκμάθησης για την ακριβή πρόβλεψη των ακυρώσεων κρατήσεων σε ξενοδοχεία. Για την επίτευξη αυτού, χρησιμοποιήθηκαν δύο σύνολα δεδομένων από βάση δεδομένων δύο ξενοδοχείων, καθένα με 31 ανεξάρτητες μεταβλητές, και χρονική διάσταση 2 χρόνων. Λαμβάνοντας υπόψη τα δικά του πλεονεκτήματα, τρεις διαφορετικοί αλγόριθμοι ταξινόμησης – random forest (RF), XGBoost και support vector machine (SVM) – χρησιμοποιούνται στην τρέχουσα μελέτη. (Yoo et al., 2023). Τα σύνολα δεδομένων (H1 και H2) χωρίστηκαν τυχαία σε ένα σύνολο εκπαίδευσης 75% και ένα σύνολο δοκιμών 25%. Τέλος, συνδυάζονται οι μέθοδοι μηχανικής εκμάθησης RF, SVM, XGBoost και ένας αλγόριθμος ψηφοφορίας για να αναπτυχθεί ένας υβριδικός ταξινομητής, γνωστός ως μέθοδος συνόλου (ensemble method).

3.2 Ερμηνευσιμότητα στις επιχειρήσεις / Explainabilty in Business

Η τεχνητή νοημοσύνη και οι τεχνικές μοντελοποίησης αυτής όπως η μηχανική μάθηση και βαθιά μάθηση(Machine Learning, Deep learning) αποκτούν όλο και πιο καθοριστικό ρόλο στην καθημερινότητα και στην λήψη αποφάσεων. Έτσι τα συστήματα τεχνικής νοημοσύνης έχουν εδραιωθεί και στον επιχειρηματικό τομέα. Η χρήση της τεχνητής νοημοσύνης στις επιχειρήσεις αποτελεί ένα σημαντικό κεφάλαιο στην εποχή της ψηφιακής μετάβασης. Οι περισσότερες αποφάσεις πλέον λαμβάνονται με την υποστήριξη και βοήθεια της τεχνητής νοημοσύνης, και αφορούν σημαντικά ατομικά δικαιώματα, ανθρώπινη ασφάλεια, κρίσιμες επιχειρηματικές λειτουργίες, παροχή υπηρεσιών κα.

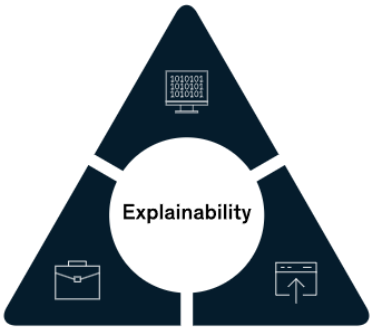
3.2.1 Σημαντικότητα ερμηνευσιμότητας στις επιχειρήσεις

Οι σύγχρονες τεχνικές μοντελοποίησης που τροφοδοτούν πολλές εφαρμογές τεχνητής νοημοσύνης, όπως η μηχανική μάθηση, βαθιά μάθηση και τα νευρωνικά δίκτυα, παρουσιάζουν συχνά προκλήσεις στον κατανοητικό επίπεδο του ανθρώπου, καθώς θεωρούνται δυσνόητα και μη προσπελάσιμα. Τα προηγμένα μοντέλα μηχανικής εκμάθησης εμφανίζονται συχνά ως "μαύρα κουτιά," καθώς η πολυπλοκότητα τους δυσκολεύει τον ανθρώπινο νου να τα κατανοήσει εύκολα. Έρευνες έχουν διαπιστώσει ότι οι εταιρείες που έχουν τις μεγαλύτερες αποδόσεις και κέρδη από την τεχνητή νοημοσύνη —δηλαδή όσες αποδίδουν τουλάχιστον το 20 τοις εκατό του EBIT (Earnings Before Interest and Taxes)/καθαρά κέρδη στη χρήση της τεχνητής νοημοσύνης— ακολουθούν βέλτιστες πρακτικές που επιτρέπουν την επεξηγησιμότητα και την ερμηνευσιμότητα των συστημάτων μοντελοποίησης

τεχνικής νοημοσύνης. Επιπλέον, οι επιχειρήσεις που εδραιώνουν ψηφιακή εμπιστοσύνη μεταξύ των καταναλωτών μέσω πρακτικών όπως η εξηγήσιμη τεχνητή νοημοσύνη (explainable AI) έχουν υψηλότερες πιθανότητες να παρατηρήσουν αύξηση των ετήσιων εσόδων και του EBIT τους, με ετήσιους ρυθμούς αύξησης 10 τοις εκατό ή ακόμη υψηλότερους (10% ή περισσότερο).

Πιθανώς να προκύπτει το ερώτημα γιατί η κατανόηση και πλήρης επεξήγηση των συστημάτων τεχνητής νοημοσύνης έχει τόσο σημαντικές επιπτώσεις στα κέρδη και στη βελτίωση του τρόπου λειτουργίας και λήψης αποφάσεων των επιχειρήσεων. Η γνώση της επεξηγηματικότητας βοηθά τους επαγγελματίες της τεχνολογίας, των επιχειρήσεων και του χώρου του κινδύνου με ποικίλους τρόπους.

Explainability creates conditions in which technical, business, and risk professionals get the most value from AI systems.



Technologists
1. More efficiently monitor, maintain, and improve AI systems

Business professionals
2. Trust AI outputs, so they increasingly adopt AI tools
3. Apply knowledge about the why of an AI prediction or recommendation to identify effective interventions
4. Assess whether AI applications meet business objectives

Legal and risk professionals
5. See whether technology and associated workflows comply with applicable regulations and are in line with customer expectations

Αύξηση της παραγωγικότητας : Η επίτευξη αυξημένης παραγωγικότητας αποτελεί ένα κρίσιμο πλεονέκτημα των τεχνικών που επιτρέπουν την επεξήγηση στα συστήματα τεχνητής νοημοσύνης. Αυτές οι τεχνικές μπορούν να αποκαλύψουν με άμεσο τρόπο πιθανά σφάλματα ή πεδία που χρήζουν βελτίωσης, επιτρέποντας στις ομάδες Μηχανικής Μάθησης (ML Operations) να εντοπίζουν και να αντιμετωπίζουν αποτελεσματικά προβλήματα στα συστήματα τεχνητής νοημοσύνης στην φάση της επίβλεψης και συντήρησης των συστημάτων. Για παράδειγμα, η κατανόηση των συγκεκριμένων χαρακτηριστικών που οδηγούν το μοντέλο στα αποτελέσματα εξόδου, μπορεί να συνεισφέρει στη διαδικασία επιβεβαίωσης εάν τα μοτίβα που προσδιορίζονται από το μοντέλο εφαρμόζονται ευρέως και είναι συναφή με μελλοντικές προβλέψεις. Αυτό βοηθά στη διασφάλιση ότι τα μοντέλα δεν αντικατοπτρίζουν αποκλειστικά μεμονωμένα ή ανωμαλίες στα ιστορικά δεδομένα. Συνεπώς, η επεξήγηση των αποφάσεων των μοντέλων συμβάλλει στη βελτίωση της απόδοσης και της αξιοπιστίας των συστημάτων τεχνητής νοημοσύνης.

Οικοδόμηση εμπιστοσύνης και υιοθεσία : Η επεξήγηση των αποφάσεων των συστημάτων τεχνητής νοημοσύνης είναι κρίσιμη για την οικοδόμηση εμπιστοσύνης και την υιοθέτησή τους. Τόσο οι πελάτες όσο και οι ρυθμιστικές αρχές, καθώς και το ευρύ κοινό, πρέπει να είναι βέβαιοι ότι τα μοντέλα τεχνητής νοημοσύνης λαμβάνουν αποφάσεις με ακρίβεια και δικαιοσύνη. Ακόμη και τα πιο προηγμένα συστήματα τεχνητής νοημοσύνης χάνουν την αξία τους εάν οι χρήστες δεν κατανοούν την λογική

πίσω από τις συστάσεις τους και τα αποτελέσματα τους. Για παράδειγμα, εάν εστιάσουμε στις ομάδες πωλήσεων είναι περισσότερο πιθανό να εμπιστεύονται μια εφαρμογή τεχνητής νοημοσύνης όταν κατανοούν τον λόγο πίσω από τις προτάσεις της, αντί να το βλέπουν ως ένα αχαρτογράφητο μαύρο κουτί. Επομένως, όταν ο λόγος που μια εφαρμογή τεχνητής νοημοσύνης προτείνει κάτι είναι γνωστός και κατανοητός, συνεισφέρει στην αύξηση της εμπιστοσύνης της επιχείρησης για την ακολούθηση της.

Εμφάνιση νέων παρεμβάσεων που παράγουν αξία : Η διευκρίνιση του τρόπου λειτουργίας ενός μοντέλου μπορεί να αποτελέσει κλειδί για τις εταιρείες, επιτρέποντάς τους να αναδείξουν επιχειρηματικές παρεμβάσεις που διαφορετικά θα παρέμεναν αόρατες. Σε ορισμένες περιπτώσεις, η βαθύτερη κατανόηση των αιτίων που οδήγησαν σε μια πρόβλεψη μπορεί να παράσχει μεγαλύτερη αξία από την ίδια την πρόβλεψη. Για παράδειγμα, μια πρόβλεψη σχετικά με την αποχώρηση πελατών από ένα συγκεκριμένο τμήμα ή την ακύρωση μιας κράτησης σε ένα ξενοδοχείο μπορεί να είναι χρήσιμη καθαυτή, αλλά μια εξήγηση γιατί συμβαίνει μπορεί να αποκαλύψει πιο αποτελεσματικούς τρόπους παρέμβασης για την επιχείρηση. Ακολουθεί ένα ρεαλιστικό παράδειγμα για να γίνει πιο αντιληπτή η σημαντικότητα εμφάνισης νέων παραγόντων. Σε μια επιχείρηση ασφάλισης αυτοκινήτων , η χρήση εργαλείων επεξήγησης όπως οι τιμές SHAP (SHAP explainabilty method) αποκάλυψε πόσο μεγαλύτερος κίνδυνος συσχετίζεται με ορισμένες αλληλεπιδράσεις μεταξύ των χαρακτηριστικών του οχήματος και του οδηγού. Εκμεταλλευόμενη αυτές τις πληροφορίες, η εταιρεία προσαρμόστηκε, βελτίωσε και εξατομίκευσε τα μοντέλα κινδύνου της, με αποτέλεσμα τη σημαντική βελτίωση της απόδοσής της.

Η διασφάλιση της τεχνητής νοημοσύνης παρέχει επιχειρηματική αξία : Όταν η τεχνική ομάδα είναι σε θέση να εξηγήσει πώς λειτουργεί ένα σύστημα τεχνητής νοημοσύνης, η ομάδα επιχειρηματικής ανάπτυξης μπορεί να επιβεβαιώσει ότι οι επιχειρηματικοί στόχοι επιτυγχάνονται και να εντοπίσει περιπτώσεις όπου κάτι ίσως χάθηκε κατά τη μετάφραση. Αυτή η διαδικασία εξασφαλίζει ότι μια εφαρμογή τεχνητής νοημοσύνης έχει προσαρμοστεί για να παρέχει την αναμενόμενη αξία.

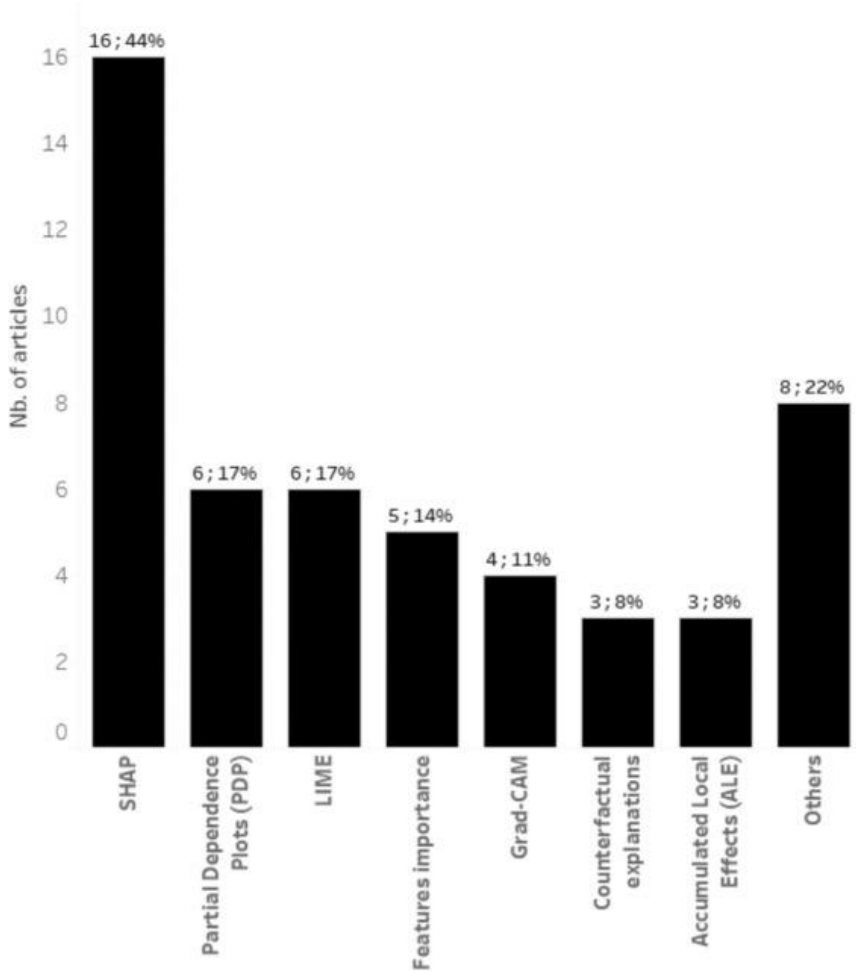
Μετριασμός ρυθμιστικών και άλλων κινδύνων : Η επεξηγησιμότητα συμβάλλει στην αντιμετώπιση των κινδύνων για τους οργανισμούς. Όταν τα συστήματα τεχνητής νοημοσύνης παραβιάζουν ηθικούς κανόνες, ακόμα και αν είναι ακούσια, μπορεί να προκαλέσουν έντονο δημόσιο, δημοσιογραφικό και ρυθμιστικό ενδιαφέρον. Οι νομικές ομάδες και οι ομάδες κινδύνου μπορούν να χρησιμοποιήσουν την εξήγηση που παρέχεται από την τεχνική ομάδα, σε συνδυασμό με την προβλεπόμενη επαγγελματική χρήση, προκειμένου να επιβεβαιώσουν τη συμμόρφωση του συστήματος με τους ισχύοντες νόμους και κανονισμούς, καθώς και τη συμφωνία με τις εσωτερικές πολιτικές και αξίες της εταιρείας.

Σε ορισμένους τομείς, η ανάγκη για επεξήγηση είναι πλέον υποχρεωτική. Για παράδειγμα, το πρόσφατο ενημερωτικό δελτίο που εκδόθηκε από το Υπουργείο Ασφάλισης της Καλιφόρνια απαιτεί από τις ασφαλιστικές εταιρείες να εξηγήσουν τις

αρνητικές ενέργειες που λαμβάνονται με βάση πολύπλοκους αλγόριθμους. Καθώς η εφαρμογή της τεχνητής νοημοσύνης αυξάνεται, οι οργανισμοί μπορεί να αντιμετωπίσουν περισσότερους κανόνες σχετικά με την ανάγκη για επεξήγηση. Πιθανόν, νέοι κανονισμοί όπως το σχέδιο κανονισμού ΕΕ για την τεχνητή νοημοσύνη, να περιλαμβάνουν συγκεκριμένα βήματα συμμόρφωσης με την επεξήγηση. Ακόμη και σε περιπτώσεις όπου δεν υπάρχει σαφής οδηγία, οι επιχειρήσεις θα πρέπει να διασφαλίζουν ότι τα εργαλεία που χρησιμοποιούνται για τη λήψη αποφάσεων συμμορφώνονται με τους ισχύοντες νόμους κατά των διακρίσεων και απαγορεύουν αθέμιτες ή παραπλανητικές πρακτικές.

Καθώς η επεξήγηση είναι μια τόσο κρίσιμη απαίτηση, είναι επιτακτική ανάγκη η εξηγήσιμη τεχνητή νοημοσύνη να περιλαμβάνεται στις αρχές της τεχνητής νοημοσύνης κάθε οργανισμού και να αποτελεί βασικό στοιχείο στη στρατηγική της. Σε πρόσφατο άρθρο, που συνοδεύεται με την αντίστοιχη μελέτη τονίζεται πως εάν τα συστήματα τεχνητής νοημοσύνης αφεθούν χωρίς παρακολούθηση, δηλαδή χωρίς την πολύτιμη συμβολή των συστημάτων ερμηνευσιμότητας - explainable AI, μπορούν να οδηγήσουν σε παράλογα αποτελέσματα, υπογραμμίζοντας έτσι τη σημασία του εξηγήσιμου AI (XAI). Για παράδειγμα, ένα chatbot που δημιουργήθηκε από τη Microsoft για το Twitter άρχισε να εκφράζει ρατσιστικές και προσβλητικές απόψεις καθώς μετά την έκθεσή του στο τοξικό περιβάλλον της πλατφόρμα, αφομοίωσε την συμπεριφορά των χρηστών και την κουλτούρα. Σε ένα άλλο παράδειγμα, ένα εργαλείο πρόσληψης της Amazon, που χρησιμοποιεί τεχνητή νοημοσύνη, ανακαλύφθηκε να εμφανίζει προκαταλήψεις κατά των γυναικών, οδηγώντας σε πρακτικές πρόσληψης που προωθούν τις διακρίσεις. Αυτά τα παραδείγματα αναδεικνύουν τις πιθανές επιπτώσεις της χρήσης συστημάτων τεχνητής νοημοσύνης χωρίς επαρκή παρακολούθηση, επισημαίνοντας την ανάγκη για εξηγήσιμη τεχνητή νοημοσύνη. ("The Role Of Explainable AI In Business Decision Making.", 2023)

Σε άλλη πρόσφατη έρευνα υπογραμμίζει τη σημασία της εστίασης στις Μεθόδους Εξηγήσιμης Τεχνητής Νοημοσύνης (XAI), τονίζοντας παράλληλα ότι αυτός ο τομέας αποτελεί ένα εξαιρετικά πρακτικό πεδίο έρευνας. Είναι ουσιώδες να διερευνηθεί πώς οι μέθοδοι XAI χρησιμοποιούνται σε πραγματικές επιχειρηματικές εφαρμογές, με στόχο την ανάδειξη βέλτιστων πρακτικών για βελτιωμένες υλοποιήσεις ή υιοθέτηση. Η έρευνα είναι βασισμένη σε 37 αυστηρά επιλεγμένα άρθρα δημοσιευμένα σε περιοδικά υψηλής ποιότητας. Αποκαλύπτει πως το πιο δημοφιλές πλαίσιο επεξήγησης είναι το SHAP Framework. Πολλα από αυτα τα άρθρα χρησιμοποιούν το SHAP ως κύριο πλαίσιο επεξήγησης. Είναι σημαντικό να επισημαίνουμε ότι ορισμένα άρθρα συγκρίνουν ή συμπληρώνουν το SHAP με άλλες μεθόδους επεξήγησης, όπως LIME, PDP, ALE, Grad-CAM. (Tchuente et al., 2024)



Εικόνα 15: Δημοτικότητα μεθόδων ερμηνευσιμότητας

(Tchuente et al., 2024)

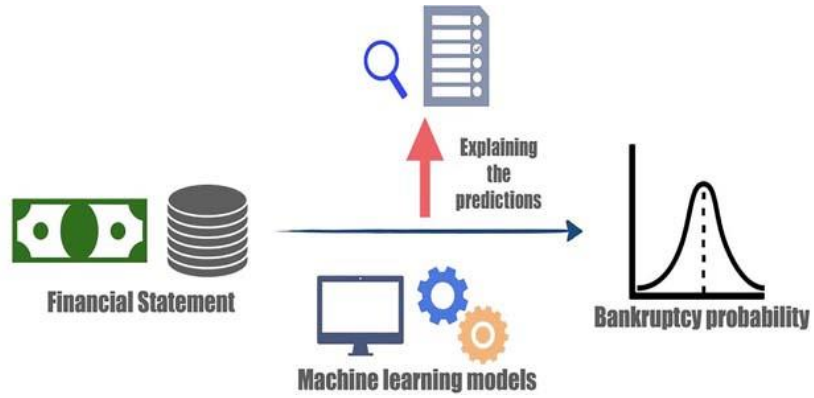
Προκειμένου να καταστεί διαφανής και επεξηγήσιμη κάθε διαδικασία λήψης αποφάσεων που βασίζεται σε μοντέλα ML, το πλαίσιο αυτό χωρίζεται σε έξι κύρια βήματα για την επεξήγηση ολόκληρης της αναλυτικής διαδικασίας: σημασία του επιχειρηματικού ερωτήματος, συλλογή δεδομένων, μηχανική χαρακτηριστικών, μοντελοποίηση και αξιολόγηση ML, έξοδοι μοντέλων (προβλέψεις), έλεγχος ευρωστίας και επικύρωση των εξηγήσεων από τους αρμόδιους φορείς. (Tchuente et al., 2024).

Η Ερμηνεύσιμη τεχνητή νοημοσύνη (XAI), έχει διαδοθεί σε διαφόρους επιχειρηματικούς τομείς, όπως ο χρηματοοικονομικός τομέας, η υγειονομική περίθαλψη και το λιανικό εμπόριο, όπου επωφελούνται από τις προτάσεις και συστάσεις των μοντέλων.

Η τεχνητή νοημοσύνη (AI) χρησιμοποιείται όλο και περισσότερο στον χρηματοπιστωτικό τομέα. Νομοθέτες και εποπτικές αρχές, συμπεριλαμβανομένης της Ευρωπαϊκής Επιτροπής και της Ευρωπαϊκής Αρχής Τραπεζών, εξετάζουν τη δυνατότητα εφαρμογής πολιτικών και κανονισμών για την τεχνητή νοημοσύνη. Στο επίκεντρο της συζήτησης βρίσκεται η ερμηνευσιμότητα, καθώς οι προηγμένες

αναλύσεις δεδομένων, όπως τα βαθιά νευρωνικά δίκτυα, δημιουργούν το πρόβλημα του "μαύρου κουτιού". Καθώς οι χρηματοπιστωτικές εταιρείες υιοθετούν προηγμένες τεχνικές, όπως η βαθιά μάθηση, η ανάγκη να κατανοήσουμε τις διαδικασίες γίνεται επιτακτική. Ρυθμιστικές αρχές και εταιρείες πρέπει να αντιμετωπίσουν τους κινδύνους και να καθορίσουν ποιο επίπεδο ερμηνευσιμότητας απαιτείται σε διάφορες καταστάσεις. Αν και δύσκολο, αυτό είναι κρίσιμο για την υπεύθυνη χρήση της τεχνητής νοημοσύνης στον χρηματοπιστωτικό τομέα. Οι σύνθετες τεχνολογίες τεχνητής νοημοσύνης είναι νέες και θα χρειαστεί χρόνος για τις εποπτικές αρχές των τραπεζών, καθώς και την ευρύτερη κοινωνία, να εμπιστευτούν την εφαρμογή τους. Η ερμηνευσιμότητα μπορεί να διασφαλίσει αυτή την σχέση εμπιστοσύνης, παρέχοντας διαφάνεια και αποσαφήνιση των μεθόδους και των διαδικασιών που ακολουθούν τα συστήματα τεχνητής νοημοσύνης. Οι τράπεζες πρέπει να διασφαλίσουν ότι αυτή η τεχνολογία εφαρμόζεται ευσυνείδητα σε όλο το τραπεζικό σύστημα, κάτι που ενδέχεται να απαιτήσει ορισμένες αλλαγές στο ρυθμιστικό πλαίσιο. (Burgt, 2020)

Στον τομέα χρηματοοικονομικών, λόγω της σημασίας που έχει για τη μέτρηση της εταιρικής φερεγγυότητας, η πρόβλεψη της πτώχευσης έχει μελετηθεί ευρέως. Το μοντέλο πρόβλεψης πτώχευσης, το οποίο προβλέπει εάν μια εταιρεία θα πτωχεύσει, πρέπει να πληροί δύο βασικές απαιτήσεις, υψηλή ακρίβεια και ερμηνευσιμότητα. Επειδή είναι σημαντικό για τους πιστωτές, τους επενδυτές και τις τράπεζες, η σαφής ερμηνεία των αποτελεσμάτων αποτελεί βασική πτυχή για της αξιολόγηση και τον καθορισμό του κατά πόσον το μοντέλο είναι χρησιμοποιήσιμο στον κλάδο. Σε σχετική μελέτη το 2021, αναφέρεται ότι τα ρυθμιστικά συστήματα πίστωσης απαιτούν την παροχή κατάλληλων πληροφοριών σχετικά με τα πρότυπα αξιολόγησης της πιστοληπτικής ικανότητας. Σε πρακτικό επίπεδο, ένα μοντέλο που είναι συνεπές στην επιλογή της σημασίας των χαρακτηριστικών μπορεί να επιλεγεί με τη χρήση της μεθόδου επεξήγησης LIME σε μοντέλα που θεωρούνται "μαύρα κουτιά", όπως τα XGB και LightGBM. (Park et al., 2021)



Εικόνα 16: Ερμηνευρισμότητα Μοντέλου Μηχανικής Μάθησης στον Χρηματοοικονομικό τομέα

Ο τομέας των πωλήσεων επωφελείται και αυτός από την χρήση επεξηγησιμότητας στα μοντέλα μηχανικής μάθησης. Τα μοντέλα που είναι επεξηγήσιμα μπορούν επίσης να παρέχουν πολύτιμες πληροφορίες σχετικά με κρίσιμες επιχειρηματικές μετρήσεις, όπως οι πωλήσεις, η απόρριψη πελατών, η φήμη του προϊόντος, ο κύκλος εργασιών

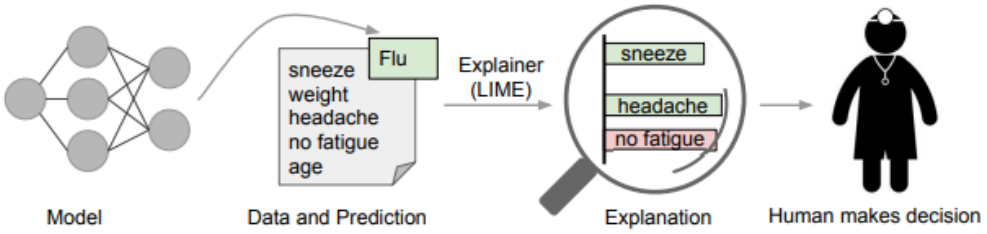
των εργαζομένων κ.λπ. Αυτές οι πληροφορίες αποτελούνται σε σημαντικά εργαλεία για την καλύτερη λήψη αποφάσεων και τον πιο αποτελεσματικό σχεδιασμό στρατηγικής. Πολλές εταιρείες χρησιμοποιούν μοντέλα μηχανικής μάθησης για να αναλύσουν το συναίσθημα των πελατών. Ενώ το συναίσθημα αυτό καθαυτό είναι σημαντικό, μια εξήγηση μοντέλου μπορεί να παράσχει επιπλέον πληροφορίες για τους παράγοντες που επηρεάζουν το συναίσθημα, όπως η τιμή, η εξυπηρέτηση πελατών, η ποιότητα των προϊόντων κλπ. Επιπλέον, μπορεί να αποκαλύψει τον τρόπο με τον οποίο αυτοί οι παράγοντες επηρεάζουν τον πελάτη, βοηθώντας τις επιχειρήσεις να αντιμετωπίσουν αποτελεσματικά τα ζητήματα.

Παρόμοια, τα μοντέλα πρόβλεψης πωλήσεων χρησιμοποιούνται ευρέως για τον υπολογισμό των πωλήσεων και τον σχεδιασμό του αποθέματος. Η εξήγηση αυτών των μοντέλων μπορεί να αποκαλύψει πώς συνεισφέρουν βασικοί παράγοντες όπως η τιμή, η προώθηση, ο ανταγωνισμός κλπ., στην πρόβλεψη των πωλήσεων. Αυτές οι πληροφορίες μπορούν να χρησιμοποιηθούν για να ενισχυθούν οι στρατηγικές πωλήσεων και η απόδοση των επιχειρήσεων, άλλα και για την άμεση ανανέωση του αποθέματος ανά προϊών ανάλογα με την δημοτικότητα και την ζήτηση που προτείνει το μοντέλο πως θα υπάρχει μελλοντικά. (Singh, 2021)

Η εξάπλωση της τεχνητής νοημοσύνης σε κρίσιμα περιβάλλοντα έχει επιφέρει επείγουσα ανάγκη για διαφανή μοντέλα, που παρέχουν μια πιο σαφή εικόνα του τρόπου που λήψης των τελικών αποφάσεων. Σε εφαρμογές που σχετίζονται με ζητήματα ζωής ή θανάτου, όπως στην υγειονομική περίθαλψη ή στα αυτόνομα οχήματα, η ερμηνευσιμότητα είναι κρίσιμη. Αυτό επιτρέπει τον άμεσο εντοπισμό ανωμαλιών στις αποφάσεις και εξασφαλίζει ότι αυτές λαμβάνονται για τους σωστούς λόγους. Για παράδειγμα, το 2019 η ομάδα του Syed και συνεργάτες, χρησιμοποίησαν βαθιά μάθηση για μια διαγνωστική έρευνα πάνω στην εντεροπάθεια και την κοιλιοκάκη (ασθένεια του εντέρου και ανοσολογική αντίδραση στην κατανάλωση γλυοτένης αντίστοιχα) σε παιδιά. Ένα μικρό νευρωνικό δίκτυο εκπαιδεύτηκε σε εικόνες βιοψίας δωδεκαδακτύλου, παρουσιάζοντας εξαιρετική ακρίβεια. Παρόλα αυτά, η ακρίβεια παρέχει μια συνολική αξιολόγηση χωρίς να εξηγεί τις επιμέρους αποφάσεις. Σε ευαίσθητα πλαίσια, όπως η υγειονομική διάγνωση, ένα σύστημα πρέπει να παρέχει διαφανή στοιχεία που υποστηρίζουν κάθε απόφασή του. Αυτό επιτρέπει σε εμπειρογνώμονες να αναλύουν τις αποφάσεις και να τις επικυρώνουν ή να τις αμφισβητήσουν. Έτσι, εξασφαλίζεται η εμπιστοσύνη και η ακρίβεια των αποτελεσμάτων σε τέτοιου είδους σύνθετα και σημαντικά πεδία. (Baralis, 2022)

Σύμφωνα με ερεύνα με θέμα την χρησιμότητα της εξηγήσιμης τεχνητής νοημοσύνης, είναι σαφές ότι ένας γιατρός είναι πολύ πιο εύκολα σε θέση να λάβει μια απόφαση και να καταλήξει σε διάγνωση με τη βοήθεια ενός μοντέλου τεχνητής νοημοσύνης εάν παρέχονται κατανοητές και επαρκής εξηγήσεις σχετικά με τις προτάσεις και τα αποτελέσματα που προέκυψαν από αυτό. Σε αυτή την περίπτωση, μια επαρκής εξήγηση περιλαμβάνει έναν κατάλογο συμπτωμάτων με σχετική βαρύτητα. Συμπτώματα που συμβάλλουν θετικά στην πρόβλεψη, χαρακτηρίζονται με πράσινο χρώμα ενώ συμπτώματα που αντιπροσώπευαν ενδείξεις αντίθετα της πρόβλεψης,

χαρακτηρίζονται από κόκκινο χρώμα. Ένας έμπειρος γιατρός συνήθως έχει προγενέστερες γνώσεις στον τομέα εφαρμογής και μπορεί να χρησιμοποιήσει αυτές τις γνώσεις για να αποδεχτεί ή να απορρίψει μια πρόβλεψη, αν κατανοήσει τη λογική πίσω από αυτήν. Έχει παρατηρηθεί, για παράδειγμα, ότι η παροχή εξηγήσεων μπορεί να αυξήσει την αποδοχή των προτεινόμενων συστάσεων και άλλων αυτοματοποιημένων συστημάτων.(Tulio Robeiro, 2016)



Εικόνα 17: Επεξήγηση μεμονωμένων προβλέψεων για διάγνωση γρίπης μέσω LIME .

(Tulio Robeiro, 2016)

Η εφαρμογή της τεχνητής νοημοσύνης στη χειρουργική έχει προχωρήσει σημαντικά τα τελευταία χρόνια, παρέχοντας πληθώρα οφελών όπως ενημερωμένες αποφάσεις, ασφάλεια, βελτιωμένη ακρίβεια και αποτελεσματικότητα. Η ανάλυση ιατρικών εικόνων, η πραγματικού χρόνου καθοδήγηση των χειρουργών, η εκτέλεση ελάχιστα επεμβατικών χειρουργικών επεμβάσεων με ρομποτικά συστήματα και η προγνωστική ανάλυση αποτελούν σημαντικούς τομείς εφαρμογής της τεχνητής νοημοσύνης στη χειρουργική.

Ωστόσο, η ενσωμάτωση της τεχνητής νοημοσύνης στον ιατρικό τομέα έχει προκαλέσει ηθικές ανησυχίες. Τα συστήματα τεχνητής νοημοσύνης μπορεί να λαμβάνουν αποφάσεις που είναι δύσκολο να εξηγήσουν ή να κατανοήσουν οι άνθρωποι. Είναι ζωτικής σημασίας να διασφαλιστεί ότι αυτά τα συστήματα είναι διαφανή και εξηγήσιμα, επιτρέποντας σε ασθενείς, παρόχους υγειονομικής περίθαλψης και ρυθμιστικές αρχές να κατανοούν τις αποφάσεις που λαμβάνονται. Η εξηγήσιμη τεχνητή νοημοσύνη διασφαλίζει ότι οι λόγοι που οδηγούν σε συγκεκριμένα συμπεράσματα είναι κατανοητοί, επιτρέποντας τη διόρθωση ανωμαλιών και την άμεση αντιμετώπιση προβλημάτων. Οι επαγγελματίες της υγείας εμπιστεύονται τα συστήματα τεχνητής νοημοσύνης μόνο εάν ο τρόπος λειτουργίας τους είναι κατανοητός και σαφής, εξασφαλίζοντας την αξιοπιστία της τεχνολογίας και την αποδοχή των αποτελεσμάτων χωρίς αμφισβήτηση των ορίων της. (Φώτη, 2023)

3.2.2 Τρόποι επίτευξης ερμηνευσιμότητας στις επιχειρήσεις

Για να φωτίσουν τα συστήματα αυτά και να ανταποκριθούν στις ανάγκες των πελατών, των εργαζομένων και των ρυθμιστικών αρχών, οι οργανισμοί πρέπει να υιοθετήσουν τις βασικές αρχές της επεξήγησης. Η λύση δεν εξαρτάται απλά από την

αναζήτηση βελτιωμένων μεθόδων μετάδοσης και κατανόησης της λειτουργίας του συστήματος. Αντίθετα, απαιτεί τη επένδυση και απόκτηση κατάλληλου συνόλου εργαλείων που επιτρέπουν ακόμη και σε έναν ειδικό να κατανοήσει την διαδικασία και το αποτέλεσμα και, στη συνέχεια, να το εξηγήσει σε άλλους.

Οι οργανισμοί που χτίζουν ένα πλαίσιο επεξήγησης και αποκτούν τα κατάλληλα εργαλεία ενεργοποίησης θα είναι σε καλύτερη θέση ώστε να αποτυπώνουν την πλήρη αξία της βαθιάς μάθησης και άλλων προόδων τεχνητής νοημοσύνης.

Δημιουργία επιτροπής διακυβέρνησης AI : Η δημιουργία μιας επιτροπής διακυβέρνησης για την καθοδήγηση των ομάδων ανάπτυξης τεχνητής νοημοσύνης αποτελεί στρατηγικό βήμα για τους οργανισμούς που επιδιώκουν να ενσωματώσουν αυτήν την τεχνολογία. Η επιτροπή, συγκροτούμενη από ηγέτες επιχειρήσεων, τεχνικούς, νομικούς και ειδικούς κινδύνου, προσφέρει τη διακυβέρνηση και τη σοφία που απαιτούνται για την αντιμετώπιση πολύπλοκων θεμάτων.

Κεντρική λειτουργία της επιτροπής είναι η καθοδήγηση στην καθιέρωση προτύπων για την επεξήγηση των συστημάτων τεχνητής νοημοσύνης. Με την ταξινόμηση κινδύνων, η επιτροπή διευκρινίζει τον τρόπο αξιολόγησης της ευαισθησίας σε διάφορες περιπτώσεις χρήσης, ενώ παράλληλα ενισχύει τη συμμόρφωση με νομικούς κανονισμούς και υιοθετεί υπεύθυνες αρχές.

Τονίζεται η ανάγκη για διαρκή αξιολόγηση και παρακολούθηση των μοντέλων τεχνητής νοημοσύνης, χρησιμοποιώντας απλούστερες στατιστικές μεθόδους όταν απαιτείται εξήγηση χωρίς να θυσιάζεται η απόδοση. Μέσα από αυτήν την διαδικασία, οι εταιρείες μπορούν να διαχειρίζονται αποτελεσματικά τους κινδύνους και να εξάγουν τη μέγιστη δυνατή αξία από την τεχνητή νοημοσύνη.

Επένδυση στις τεχνολογίες επεξήγησης (explainability methods), στην έρευνα και εκπαίδευση : Για να ανταποκριθούν στον ραγδαίο ρυθμό εξελίξεων στην τεχνολογία επεξήγησης, οι επιχειρήσεις πρέπει να επενδύσουν στο σωστό ταλέντο, εργαλεία και εκπαίδευση. Στόχος των εταιριών είναι η συνεργασία μεταξύ νομικών, ειδικών κινδύνου και τεχνολόγων για να διαχειρίζονται αποτελεσματικά τις νέες απαιτήσεις και προκλήσεις.

Η επένδυση σε τεχνολογία επεξήγησης πρέπει να στοχεύει στην απόκτηση εργαλείων που καλύπτουν τις ανάγκες των ομάδων ανάπτυξης, ενισχύοντας την εξήγηση χωρίς να θυσιάζεται η ακρίβεια. Επιλογές όπως εξατομικευμένες λύσεις ή ανοιχτός κώδικα πρέπει να είναι προσαρμοσμένες στις ανάγκες και τους περιορισμούς της εταιρείας.

Η συνεχής έρευνα είναι αναγκαία, καθώς οι νομικές και κανονιστικές προκλήσεις εξελίσσονται συνεχώς. Οι επιτροπές διακυβέρνησης πρέπει να διατηρούν ενεργή έρευνα, εξασφαλίζοντας συνεχή εκπαίδευση για τους εργαζόμενους, προκειμένου να ανταποκρίνονται στις αλλαγές και να διασφαλίζουν την ηθική χρήση της τεχνητής νοημοσύνης.

Παρόλο που η υιοθέτηση εξηγήσιμης τεχνητής νοημοσύνης είναι αναγκαία, αντιμετωπίζει προκλήσεις. Ένα από τα κύρια ζητήματα είναι η έλλειψη ενός ενιαίου ορισμού για την επεξηγηματικότητα, καθώς διάφοροι χρήστες μπορεί να έχουν διαφορετικές ανάγκες και προσδοκίες. Για παράδειγμα, ένας επιχειρηματίας ενδιαφέρεται να κατανοήσει τους λόγους πίσω από μια συγκεκριμένη σύσταση, ενώ ένας επιστήμονας δεδομένων ενδιαφέρεται περισσότερο για τεχνικές λεπτομέρειες σχετικά με τον τρόπο λειτουργίας του μοντέλου.

Ένα άλλο εμπόδιο στον δρόμο προς την εξηγήσιμη τεχνητή νοημοσύνη απορρέει από το γεγονός ότι πολλά υπάρχοντα συστήματα τεχνητής νοημοσύνης δεν έχουν σχεδιαστεί με σκοπό την εύκολη επεξήγηση. Η πολυπλοκότητα του σχεδιασμού των δικτύων καθιστά αυτά τα συστήματα δύσκολα στην ερμηνεία, δημιουργώντας πρόκληση στην κατανόηση του τρόπου λήψης αποφάσεων. Για να υπερκεραστεί αυτή η πρόκληση, είναι αναγκαίο όχι μόνο να εξελίσσονται συστήματα που επιδιώκουν την εξηγησιμότητα, αλλά και να εφαρμόζονται αποτελεσματικά σε διάφορες περιπτώσεις χρήσης, προσαρμόζοντας τον σχεδιασμό τους ώστε να προάγει τη διαφάνεια και την κατανόηση των αποφάσεων που λαμβάνουν.

Επιπλέον, τα πραγματικά δεδομένα είναι συνήθως πολύπλοκα και θορυβώδη, δυσκολεύοντας την κατανόηση των λόγων πίσω από μια απόφαση. Η δημιουργία εξηγήσεων απαιτεί σημαντικούς υπολογιστικούς και χρονικούς πόρους.

Σε επιχειρηματικούς τομείς με υψηλό κίνδυνο, όπως η υγειονομική περίθαλψη ή τα οικονομικά, η ανάγκη για εξηγήσιμη τεχνητή νοημοσύνη είναι περισσότερο κρίσιμη. Παρ' όλα αυτά, πρέπει να ληφθεί υπόψη η ισορροπία μεταξύ επεξήγησης και ακρίβειας, καθώς η δημιουργία εξηγήσεων μπορεί να επηρεάσει την ακρίβεια ή να αυξήσει την πολυπλοκότητα. Επομένως, οι επιχειρήσεις πρέπει να εξετάσουν προσεκτικά τα οφέλη και το κόστος της εξηγήσιμης τεχνητής νοημοσύνης, βρίσκοντας την κατάλληλη ισορροπία για κάθε περίπτωση χρήσης.

4. Μεθοδολογία

Στο παρόν κεφάλαιο, παρουσιάζεται η μεθοδολογία που χρησιμοποιήθηκε, η επιλογή της οποίας έχει σημαντικό ρόλο στην επίτευξη των επιθυμητών αποτελεσμάτων της εργασίας.

Το πρώτο βήμα, μετά τον καθορισμό της θεματολογίας και τον προσδιορισμό του πεδίου του προβλήματος, εστιάζει στην αναζήτηση δεδομένων. Μετά από ενδελεχή έρευνα, επιλέχθηκε το σύνολο των δεδομένων να αντληθεί από το Kaggle.com . Το Kaggle παρέχει ευρέως προσβάσιμα σύνολα δεδομένων για ανάλυση και εξαγωγή συμπερασμάτων. Επιλέχθηκε ένα σύνολο δεδομένων που πληρούσε τις απαιτήσεις που τέθηκαν. Πιο συγκεκριμένα, αποτελείται από ένα μεγάλο αριθμό εγγραφών, περιλαμβάνοντας ετικέτες (labeled data), πρόσφατη χρονολογία συγκέντρωσης δεδομένων (2017/18) και κατάλληλα χαρακτηριστικά (features).

Μετά την εύρεση του κατάλληλου συνόλου δεδομένων, απαιτείται μια πρώτη επαφή με αυτά. Είναι αναγκαίο να γνωρίζουμε τις παραμέτρους, τα χαρακτηριστικά, και το εύρος των τιμών που μπορεί να παρουσιάζουν. Τα δεδομένα χρειάζεται να περιγραφούν για να γίνουν κατανοητά. Έπειτα, πραγματοποιούμε διερευνητική ανάλυση των δεδομένων. Με την χρήση της γλώσσας Python δημιουργήθηκαν οπτικοποιήσεις και διαγράμματα που συμβάλουν στην εξερεύνηση των σχέσεων μεταξύ των χαρακτηριστικών, καθώς και τη σύνδεσή τους με την τιμή της στήλης ετικέτας (target column). Οι βιβλιοθήκες που χρησιμοποιήθηκαν είναι NumPy, Pandas, Matplotlib, Seaborn. Μέσω αυτής της διαδικασίας, επιτυγχάνεται καλύτερη κατανόηση των δεδομένων και αναδύονται σημαντικές σχέσεις μεταξύ των χαρακτηριστικών, καθώς και πρώτες ενδείξεις και παρατηρήσεις.

Η επόμενη φάση επικεντρώνεται στον τομέα της επεξεργασίας των δεδομένων. Συγκεκριμένα, υλοποιείται μια προ-επεξεργασία με σκοπό τη διατήρηση μόνο των δεδομένων που προσφέρουν προβλεπτική αξία στο μοντέλο. Επιπλέον, προβαίνουμε στην προσθήκη νέων μεταβλητών για τη βελτιστοποίηση της ανάλυσης.

Στη συνέχεια, προχωρούμε στον σχεδιασμό του προβλεπτικού μοντέλου μηχανικής μάθησης με χρήση της γλώσσας προγραμματισμού Python, επιλέγοντας τον αλγόριθμο XGBoost. Τα δεδομένα διαιρούνται σε δύο υποσύνολα, τα train και test data. Για την εκπαίδευση του μοντέλου XGBoost χρησιμοποιείται το train data. Το μοντέλο XGBoost είναι ένας αλγόριθμος μηχανικής εκμάθησης που βασίζεται στην αρχή της ενισχυμένης κλίσης (GBDT). Ξεκινάει με ένα αρχικό δέντρο απόφασης που παίρνει τη μέση τιμή των ετικετών ως αρχική πρόβλεψη. Υπολογίζει τα σφάλματα μεταξύ της αρχικής πρόβλεψης και των πραγματικών ετικετών. Δημιουργεί ένα νέο δέντρο απόφασης που προσπαθεί να διορθώσει τα σφάλματα του προηγούμενου. Αναθέτει βάρη στα λάθη, ώστε τα επόμενα δέντρα να επικεντρώνονται περισσότερο στα αδιόρθωτα σημεία. Επαναλαμβάνει τη διαδικασία, δημιουργώντας νέα δέντρα που συμβάλλουν στη μείωση των σφαλμάτων. Οι προβλέψεις από όλα τα δέντρα αθροίζονται, δημιουργώντας την τελική πρόβλεψη. Η τελική αυτή εκτίμηση συγκρίνεται με την πραγματική τιμή που θα έπρεπε να προέβλεπε το μοντέλο.

Στην συνεχεία παρουσιάζονται οι εκτιμητές αξιολόγησης του μοντέλου μαζί με κάποια συμπληρωματικά διαγράμματα.

Στο τελευταίο κεφάλαιο, πραγματοποιείται η ερμηνεία του μοντέλου μηχανικής μάθησης με χρήση της βιβλιοθήκης SHAP, εστιάζοντας στην δυνατότητα επεξήγησης του τρόπου όπου το μοντέλο καταλήγει στις προβλέψεις του. Έπειτα, παρουσιάστηκαν τα αποτελέσματα της ανάλυσης και πραγματοποιείτε η εξαγωγή συμπερασμάτων από τα προηγούμενα βήματα της μελέτης.

5. Σύνολο Δεδομένων

5.1 Περιγραφή του προβλήματος

Οι διαδικτυακές πλατφόρμες κρατήσεων για ξενοδοχεία έχουν επιφέρει σημαντικές αλλαγές στον τρόπο που οι πελάτες κάνουν κρατήσεις και στη συμπεριφορά τους. Ένας σημαντικός αριθμός κρατήσεων ξενοδοχείων ακυρώνεται είτε λόγω αλλαγής σχεδίων είτε λόγω διαφορετικών προγραμμάτων ή άλλων αιτιών. Αυτό συμβαίνει συχνά επειδή η δυνατότητα ακύρωσης είναι εύκολα προσβάσιμη και συχνά δωρεάν ή με χαμηλό κόστος για τους πελάτες. Αυτό μπορεί να είναι ευεργετικό για τους επισκέπτες των ξενοδοχείων, αλλά αποτελεί μείον για τα ίδια τα ξενοδοχεία, καθώς μπορεί να έχει αρνητική επίδραση στα εισοδήματά τους.

Το ερώτημα που τίθεται είναι το εξής: "Μπορούμε να προβλέψουμε αν ένας πελάτης θα τηρήσει την κράτησή του ή θα την ακυρώσει;"

Για να απαντήσουμε σε αυτήν την ερώτηση, πραγματοποιήθηκε μια μελέτη χρησιμοποιώντας δεδομένα που προέρχονται από την πλατφόρμα Kaggle.com *(Hotel Reservations Dataset: Can you predict if customer is going to cancel the reservation?).* Η μελέτη αυτή επικεντρώθηκε στην ανάλυση και την εύρεση απαντήσεων σχετικά με το εν λόγω ερώτημα, διερευνώντας τα διαθέσιμα δεδομένα.

5.2 Επεξήγηση των Δεδομένων

Το διαθέσιμο σύνολο δεδομένων περιλαμβάνει ένα αρχείο με την ονομασία «Hotel_Reservation.csv» το οποίο περιέχει 36,275 παρατηρήσεις. Κάθε παρατήρηση αντιπροσωπεύει μια κράτηση σε ξενοδοχείο. Τα δεδομένα καλύπτουν τη διάρκεια δύο ετών, από το 2017 έως το 2018, και περιλαμβάνουν τόσο κρατήσεις που έχουν πραγματοποιηθεί όσο και κρατήσεις που έχουν ακυρωθεί. Δεδομένου ότι πρόκειται για πραγματικά δεδομένα, έχουν διαγραφεί όλες οι πληροφορίες που αφορούν την ταυτότητα του ξενοδοχείου ή των πελατών.

Αναλυτικότερα, το «Hotel_Reservation.csv» αρχείο περιέχει 19 στήλες, καθεμία από τις οποίες περιγράφει τις παρακάτω πληροφορίες για τις κρατήσεις πελατών.

Μεταβλητή	Περιγραφή
Booking_ID	Μοναδικό αναγνωριστικό κάθε κράτησης

no_of_adults	Αριθμός Ενηλίκων
no_of_children	Αριθμός Παιδιών
no_of_weekend_nights	Αριθμός διανυκτερεύσεων Σαββατοκύριακου (Σάββατο ή Κυριακή) που ο επισκέπτης έμεινε ή έκανε κράτηση για να μείνει στο ξενοδοχείο
no_of_week_nights	Αριθμός Εβδομαδιαίων διανυκτερεύσεων (Δευτέρα έως Παρασκευή) που ο επισκέπτης έμεινε ή έκανε κράτηση για να μείνει στο ξενοδοχείο
type_of_meal_plan	Τύπος προγράμματος γευμάτων που έχει κλείσει ο πελάτης
required_car_parking_space	Απαιτούμενος χώρος στάθμευσης αυτοκινήτων (0=OXI, 1=NAI)
room_type_reserved	Τύπος δωματίου που έχει επιλεγεί στην κράτηση
lead_time	Αριθμός ημερών μεταξύ της ημερομηνίας κράτησης και της ημερομηνίας άφιξης
arrival_year	Έτος της ημερομηνίας άφιξης
arrival_month	Μήνας της ημερομηνίας άφιξης
arrival_date	Ημέρα της ημερομηνίας άφιξης

market_segment_type	Προσδιορισμός τμήματος της αγοράς
repeated_guest	Επαναλαμβανόμενος Πελάτης (0=ΟΧΙ, 1=ΝΑΙ)
no_of_previous_cancellations	Αριθμός προηγούμενων κρατήσεων που ακυρώθηκαν από τον πελάτη πριν από την τρέχουσα κράτηση
no_of_previous_bookings_not_canceled	Αριθμός προηγούμενων κρατήσεων που δεν ακυρώθηκαν από τον πελάτη πριν από την τρέχουσα κράτηση
avg_price_per_room	Μέση τιμή ανά ημέρα κράτησης- οι τιμές των δωματίων είναι δυναμικές (σε ευρώ)
no_of_special_requests	Συνολικός αριθμός ειδικών αιτημάτων του πελάτη (π.χ. υψηλός όροφος, θέα από το δωμάτιο κ.λπ.)
booking_status	Σημαία που δείχνει αν η κράτηση ακυρώθηκε ή όχι (Canceled= Ακυρώθηκε, Not_Canceled=Δεν ακυρώθηκε)

Σε αυτό το αρχείο η τελευταία στήλη, booking status είναι η λεγόμενη στήλη στόχος (Target Column), όπου στα δεδομένα δοκιμής περιέχει τις ιστορικές τιμές με τις οποίες συγκρίνονται οι προβλέψεις.

Ανάμεσα σε όλες τις μεταβλητές, διακρίνονται τρεις που περιγράφουν μη αριθμητικές τιμές, αλλά τύπους χαρακτηριστικών. Αυτές οι μεταβλητές είναι ο " "type_of_meal_plan", "room_type_reserved" και ο "market_segment_type". Κάθε μία από αυτές περιέχει πληροφορίες σχετικά με τις επιλογές τύπων γευμάτων, τύπων δωματίων και τύπο τμήματος της αγορά των κρατήσεων, αντίστοιχα.

Για την μεταβλητή "type_of_meal_plan" (τύπος προγράμματος γευμάτων):

- Meal Plan 1 (Πρόγραμμα Γευμάτων 1): 27,835 πελάτες
- Not Selected (Δεν Έχει Επιλεγεί): 5,130 πελάτες
- Meal Plan 2 (Πρόγραμμα Γευμάτων 2): 3,305 πελάτες

- Meal Plan 3 (Πρόγραμμα Γευμάτων 3): 5 πελάτες

Για την μεταβλητή " room_type_reserved" (τύπος δωματίου):

- Room_Type 1 (Τύπος δωματίου 1) 28130 πελάτες
- Room_Type 4 (Τύπος δωματίου 4) 6057 πελάτες
- Room_Type 6 (Τύπος δωματίου 6) 966 πελάτες
- Room_Type 2 (Τύπος δωματίου 2) 692 πελάτες
- Room_Type 5 (Τύπος δωματίου 5) 265 πελάτες
- Room_Type 7 (Τύπος δωματίου 7) 158 πελάτες
- Room_Type 3 (Τύπος δωματίου 3) 7 πελάτες

Για την μεταβλητή "market_segment_type" (τύπος τμήματος αγοράς):

- Online (Διαδικτυακή): 23,214 πελάτες
- Offline (Εκτός Διαδικτύου): 10,528 πελάτες
- Corporate (Εταιρική): 2,017 πελάτες
- Complementary (Συμπληρωματική): 391 πελάτες
- Aviation (Αεροπορία): 125 πελάτες

6. Διερευνητική ανάλυση δεδομένων

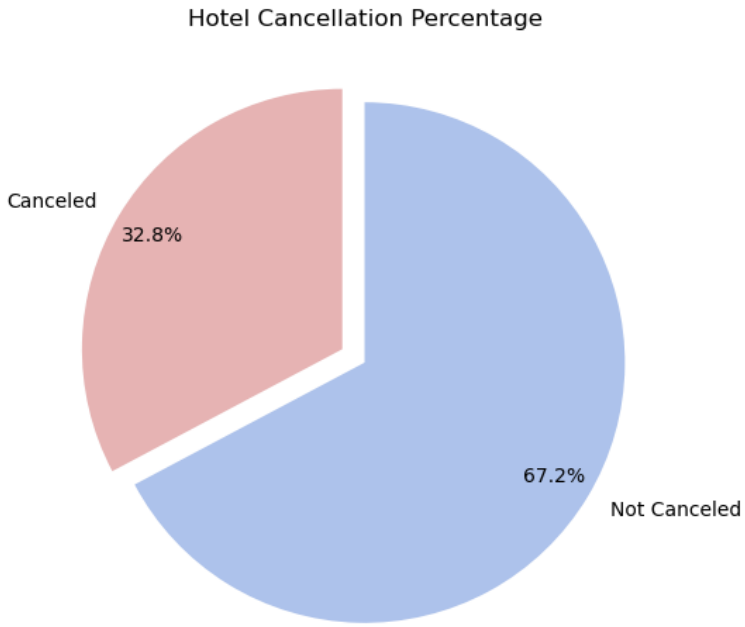
Η διερευνητική ανάλυση δεδομένων χρησιμοποιείται για την ανάλυση και τη διερεύνηση συνόλων δεδομένων και τη σύνοψη των κύριων χαρακτηριστικών τους. Βοηθά στον καθορισμό του καλύτερου τρόπου χειρισμού των πηγών δεδομένων για την λήψη των απαντήσεων που χρειάζονται, διευκολύνοντας την ανίχνευση μοτίβων, τον εντοπισμό ανωμαλιών, τον έλεγχο υποθέσεων κ.α.

Σε αυτό το κεφάλαιο, επιδιώκεται η απεικόνιση ορισμένων χαρακτηριστικών και των σχέσεων μεταξύ αυτών καθώς και με την μεταβλητή στόχου. Στην περίπτωση μας η μεταβλητή στόχου είναι η κατάσταση κρατήσεων (Bookings Status).

Η διαδικασία αυτή πραγματοποιείται μέσω της Python, χρησιμοποιώντας τις βιβλιοθήκες Matplotlib και Seaborn για την οπτικοποίηση των διαγραμμάτων.

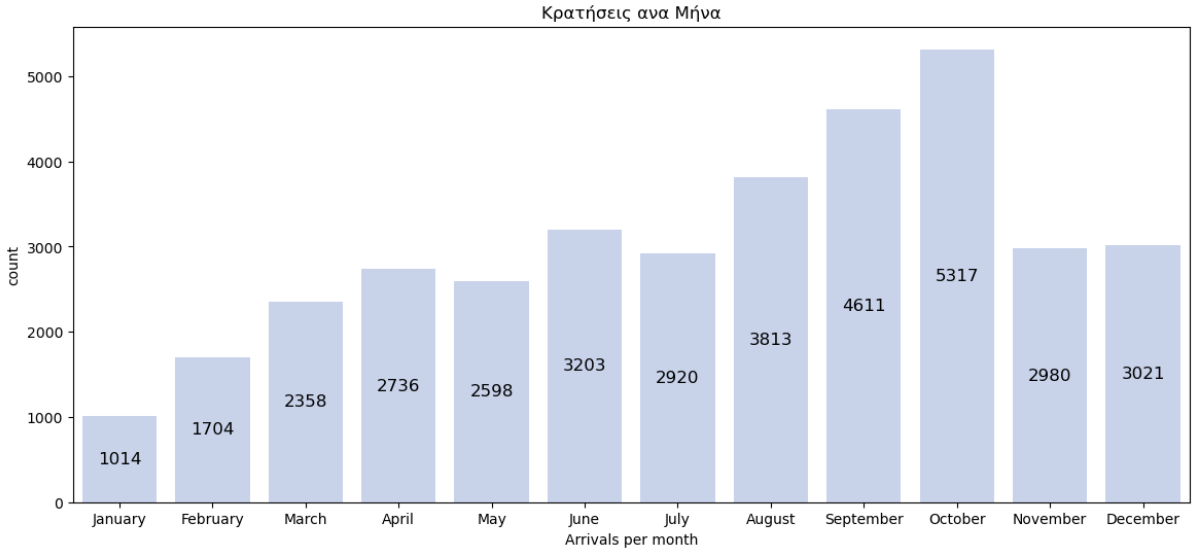
6.1 Κατανομή μεταβλητής στόχου: Ακυρωμένες και μη κρατήσεις

Το επιλεγμένο σύνολο δεδομένων περιλαμβάνει συνολικά 36,275 κρατήσεις, εκ των οποίων 11,885 ακυρώθηκαν. Στο διάγραμμα πίτας παρουσιάζονται ποσοστιαία τα μεγέθη των ακυρωμένων και μη κρατήσεων. Παρατηρείται ότι το ποσοστό των ακυρωμένων κρατήσεων σε ξενοδοχεία είναι σημαντικά υψηλό.

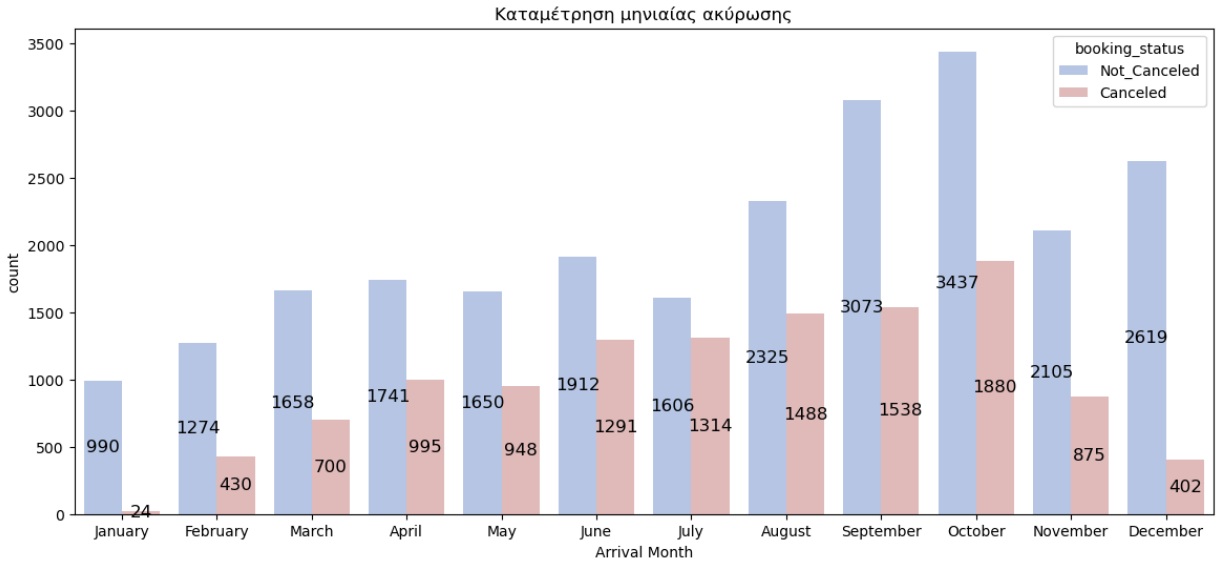


Εικόνα 18: Ποσοστό Ακυρώσεων/Κρατήσεων

Πολλοί είναι οι παράγοντες που επηρεάζουν το υψηλό ποσοστό ακυρώσεων, μεταξύ των οποίων ένας κρίσιμος παράγοντας είναι η ημερομηνία άφιξης. Στην εικόνα 12 παρουσιάζεται η κατανόηση του συνολικού αριθμού κρατήσεων, χωρισμένων σε ακυρωμένες και μη ανά μήνα. Ο μήνας Οκτώβριος ξεχωρίζει με τον υψηλότερο αριθμό κρατήσεων, φτάνοντας τις 5317, ενώ ο Σεπτέμβριος ακολουθεί με 4611 κρατήσεις.



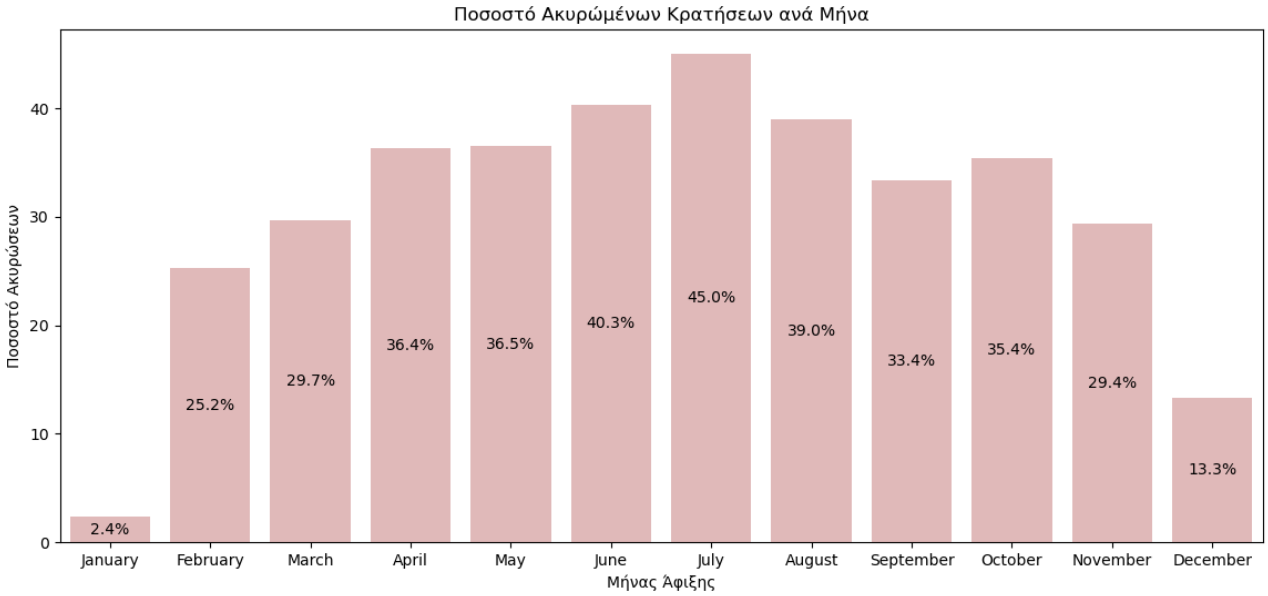
Εικόνα 19: Κατανομή συνολικών κρατήσεων ανά μήνα



Εικόνα 20: Καταμέτρηση μηνιαίας ακύρωσης

Κατά τους καλοκαιρινούς μήνες, παρατηρούμε ότι ο αριθμός των ακυρωμένων κρατήσεων είναι πολύ κοντά στον αριθμό των μη ακυρωμένων κρατήσεων, προσφέροντας ένα υψηλό ποσοστό ακυρώσεων σε σχέση με το συνολικό αριθμό κρατήσεων. Ο μέσος ορός ακυρωμένων κρατήσεων του Ιουνίου, Ιούλιου και Αύγουστου υποδηλώνει ότι περίπου το 40% των κρατήσεων της καλοκαιρινής περιόδου καταλήγουν σε ακυρώσεις. Αυτό σημαίνει ότι, ενώ οι καλοκαιρινοί μήνες είναι δημοφιλείς για κρατήσεις, ο αριθμός των ακυρώσεων παραμένει υψηλός και υποδεικνύει προβλήματα ή τάσεις που πρέπει να εξεταστούν περαιτέρω.

Αντίθετα, ην περίοδο των χειμερινών διακοπών (Δεκέμβριος - Ιανουάριος), παρατηρείται το χαμηλότερο ποσοστό ακυρώσεων, ιδίως τον Ιανουάριο με περίπου 2.4%. Αυτό υποδηλώνει ότι κατά τη διάρκεια αυτής της περιόδου, οι επισκέπτες είναι πιο δεσμευμένοι και λιγότερο πιθανό να ακυρώσουν τις κρατήσεις τους. Σχετικά με τον Δεκέμβριο, η αναλογία ακυρώσεων προς τον συνολικό αριθμό κρατήσεων παραμένει σχετικά χαμηλή με ποσοστό 13.3%, υπογραμμίζοντας τη σταθερότητα της κατάστασης και το γεγονός ότι οι περισσότεροι επισκέπτες διατηρούν τις κρατήσεις τους ίσως λόγω των διακοπών και των εορτών.



Εικόνα 21: Ποσοστό Ακυρωμένων Κρατήσεων ανά μήνα

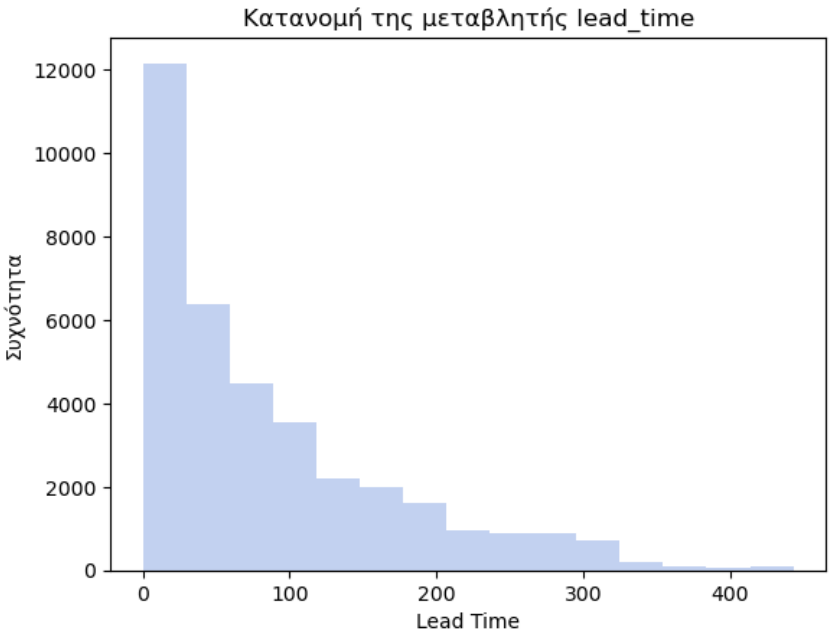
6.2 Κατανομή της μεταβλητής lead time

Η μεταβλητή lead time αναφέρεται στον αριθμό των ημερών που διανύουν από την ημερομηνία κράτησης μέχρι την ημερομηνία άφιξης.

Ακλουθεί το ιστόγραμμα για την κατανομή της μεταβλητής lead time. Η συχνότητα είναι ένας όρος που χρησιμοποιείται στη στατιστική για να περιγράψει πόσο συχνά εμφανίζεται μια τιμή σε ένα σύνολο δεδομένων. Η συχνότητα μπορεί να υπολογιστεί ως ο αριθμός των παρατηρήσεων που έχουν μια συγκεκριμένη τιμή.

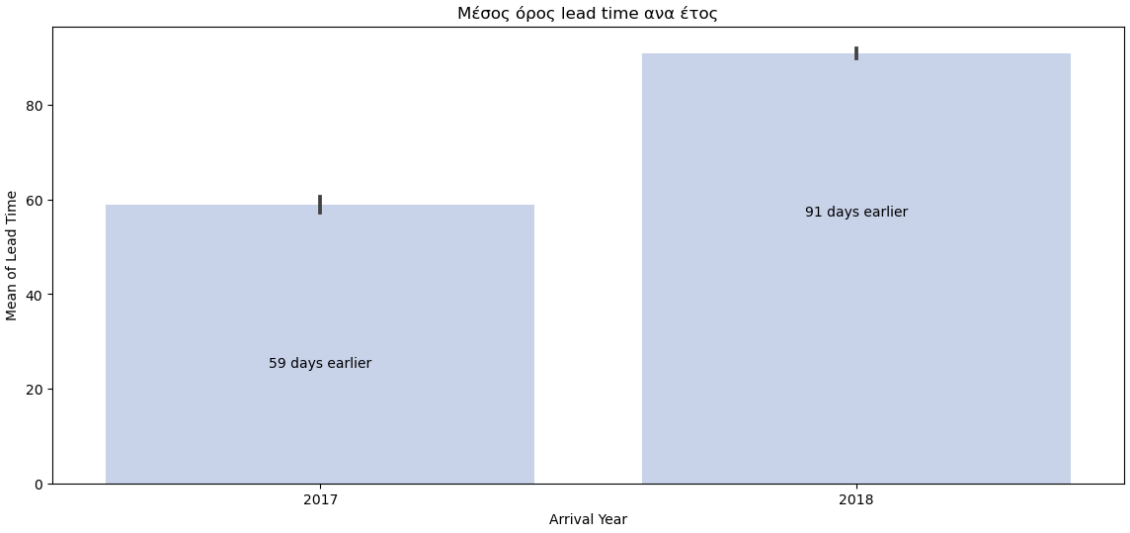
Στο διάγραμμα κατανομής lead time, η συχνότητα δείχνει τον αριθμό των κρατήσεων για κάθε lead time. Συναντάμε την μέγιστη συχνότητα της κατανομής 11885 στο διάστημα 0-30 ημερών μεταξύ ημέρας κράτησης και άφιξης, που υποδηλώνει ότι η πλειοψηφία των πελατών κάνουν κράτηση για τις διακοπές τους το πολύ 1 μήνα πριν την ημερομηνία άφιξής τους.

Υπάρχει ένα μικρότερο ποσοστό κρατήσεων με lead time μεγαλύτερο από 300 ημέρες, δηλαδή 10 μήνες νωρίτερα από την ημέρα άφιξης. Αυτό μπορεί να οφείλεται σε κρατήσεις για μακροπρόθεσμες διακοπές. Παρόλα αυτά παρατηρούμε ότι οι πελάτες δεν επιλέγουν συχνά τόσο μεγάλη απόσταση ημερών από την ημέρα άφιξης.



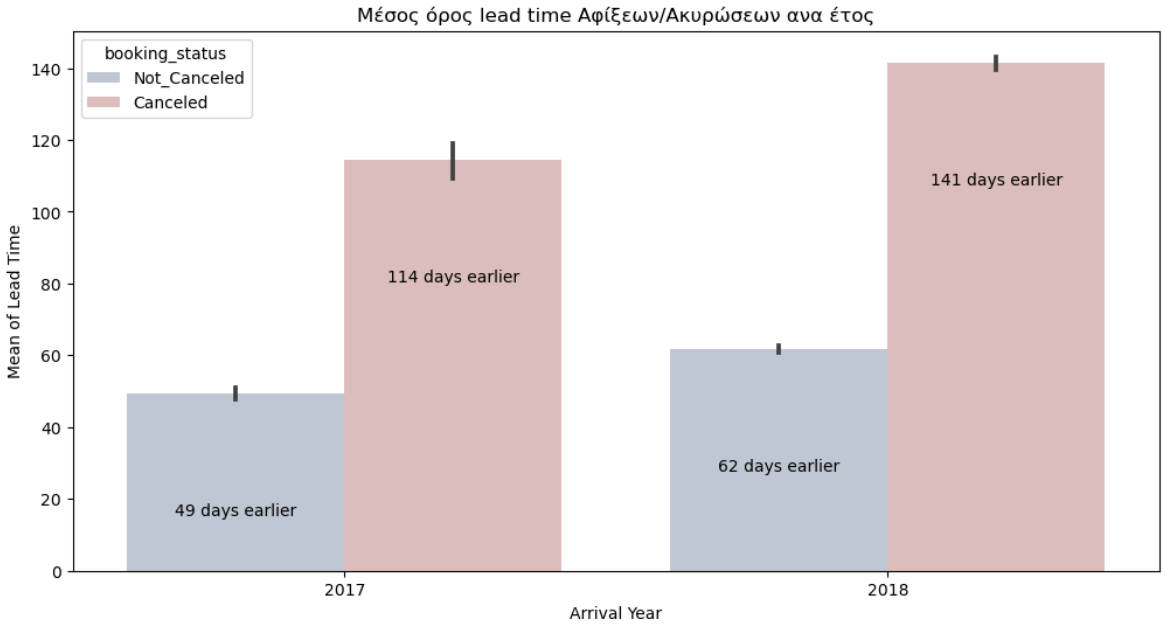
Εικόνα 22: Κατανομή Lead Time

Η διαφορά στον μέσο όρο της διάρκειας μεταξύ ημέρας κράτησης και ημερομηνίας άφιξης μεταξύ των ετών 2017 και 2018 μπορεί να υποδηλώνει μια τάση που αλλάζει με την πάροδο του χρόνου. Το 2018 παρατηρείται υψηλότερη τιμή του lead time.

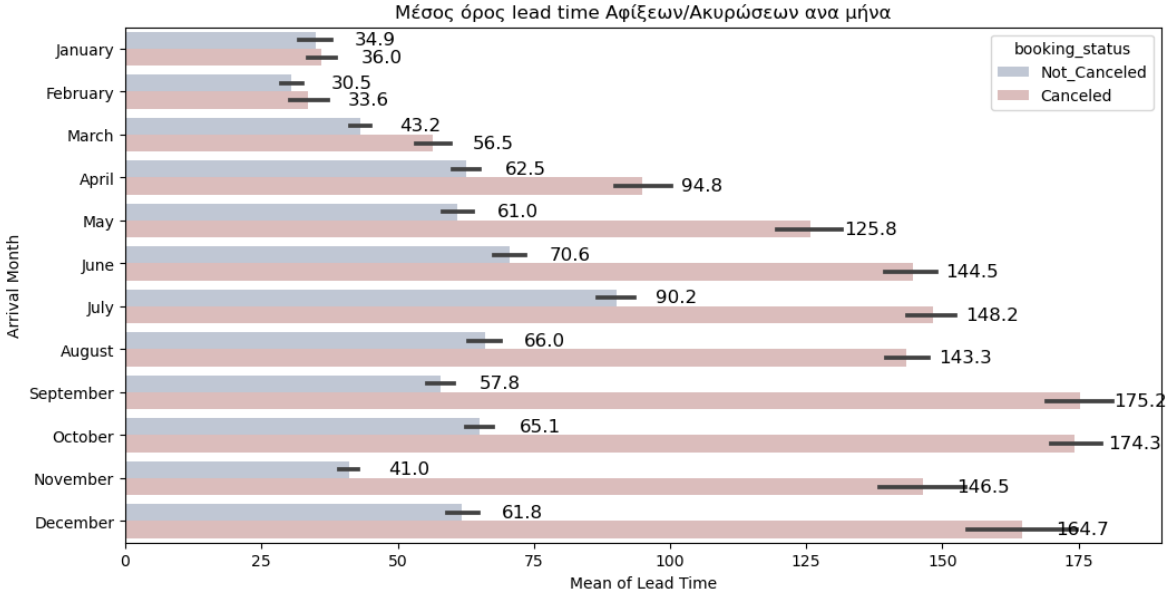


Εικόνα 23: Μέσος Όρος lead time ανά έτος

Ο μέσος όρος του lead time για τις ακυρωμένες κρατήσεις είναι σχεδόν διπλάσιος από αυτόν των μη ακυρωμένων κρατήσεων και τα δύο έτη. Αυτό υποδεικνύει ότι υψηλότερες τιμές στο lead time μπορεί να συνδέονται με ακυρώσεις. Ενδεχομένως, οι πελάτες που ακυρώνουν έχουν τάση να κάνουν κρατήσεις εκ των προτέρων, ενώ οι υπόλοιποι πελάτες μπορεί να κάνουν κρατήσεις πιο κοντά στην ημερομηνία άφιξης.



Εικόνα 24: Μέσος όρος lead time ανά έτος (Ακυρωμένες/Μη-ακυρωμένες Κρατήσεις)

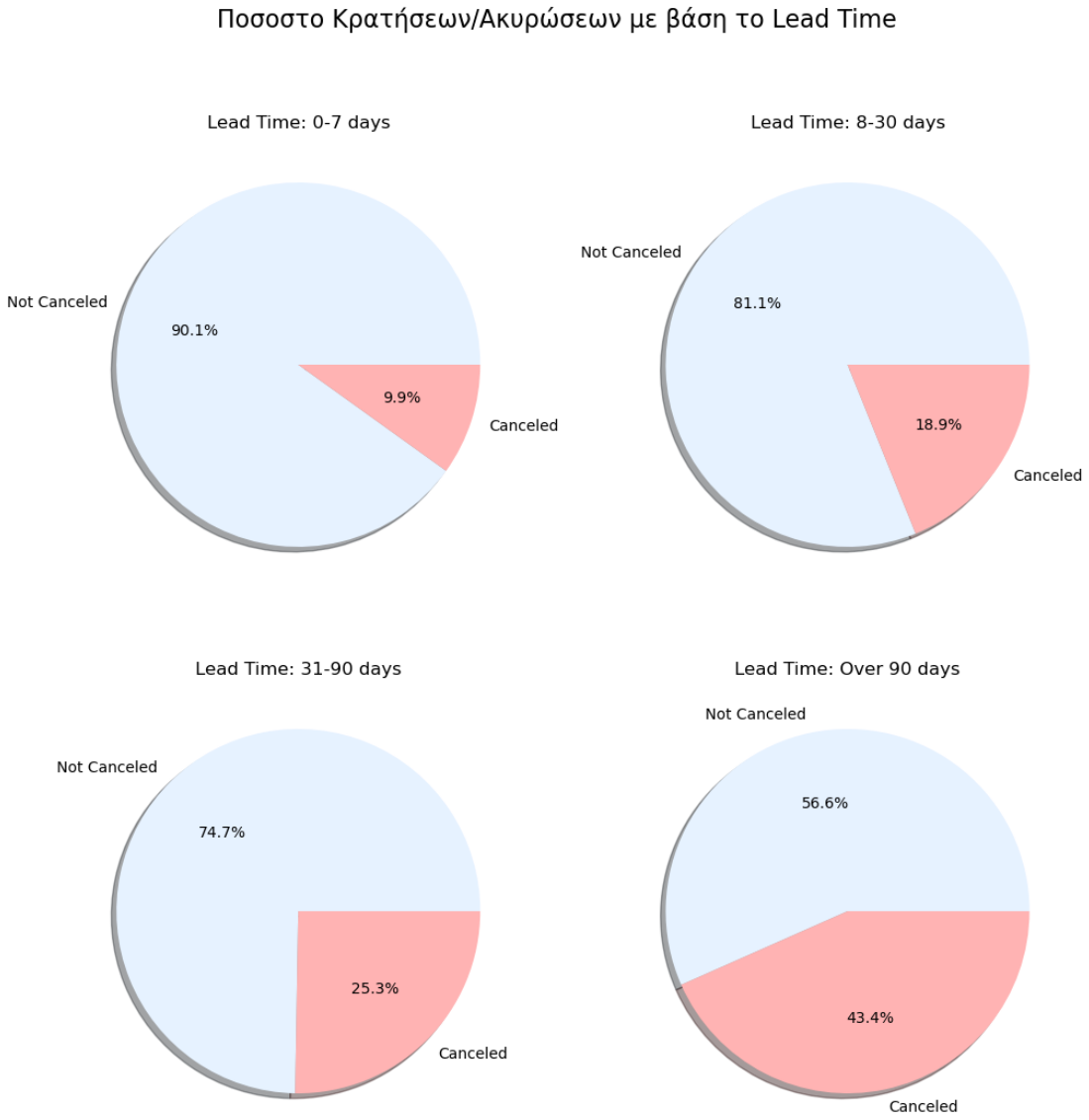


Εικόνα 25: Μέσος όρος lead time ανά μήνα (Ακυρωμένες/Μη-ακυρωμένες Κρατήσεις)

Το παραπάνω διάγραμμα δείχνει ότι η μέση τιμή του lead time στις ακυρωμένες κρατήσεις ποικίλλει σημαντικά ανάλογα με τον μήνα της άφιξης. Για παράδειγμα, η μέση τιμή του lead time είναι 33 ημέρες τον Φεβρουάριο, ενώ είναι 175 ημέρες τον Σεπτέμβριο. Οι υψηλότερες τιμές σημειώνονται τους μήνες Οκτώβριου και Νοέμβριου με μέσο ορό περίπου 175 ημέρες/ 6 μήνες.

Ο μέσος όρος του lead time, για τις κρατήσεις που δεν ακυρώθηκαν δείχνει μικρότερες διακυμάνσεις, παραμένοντας στο εύρος 30-90 ημερών (1 –3 μήνες) . Αυτό υποδηλώνει ότι οι μη ακυρωμένες κρατήσεις τείνουν να έχουν σταθερότερο

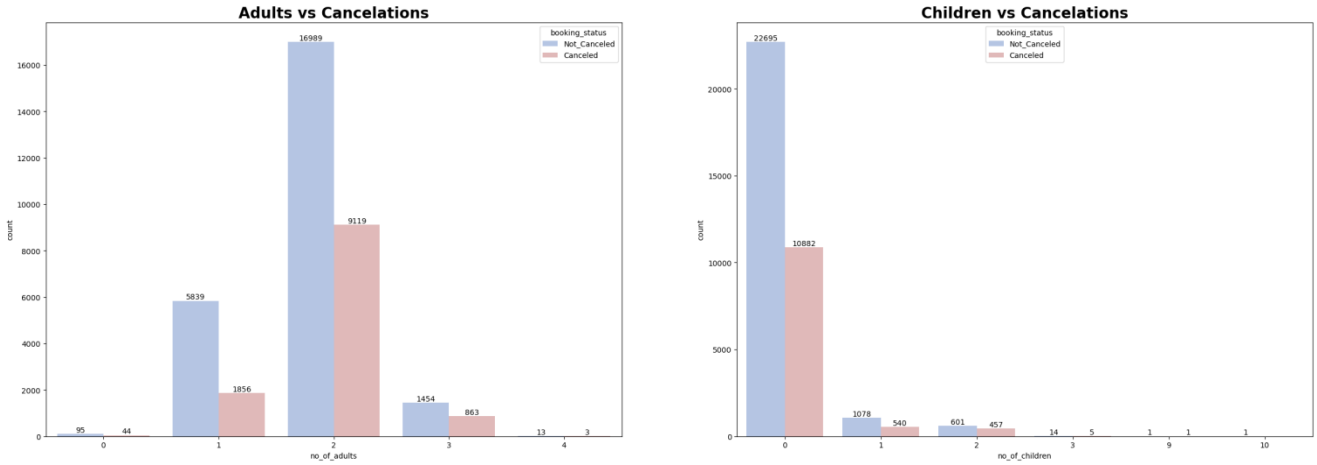
χρονικό διάστημα μεταξύ της κράτησης και της άφιξης, κυμαινόμενο σε λογικά πλαίσια.



Εικόνα 26: Ποσοστό Κρατήσεων με βάση το lead time

Στην εικόνα 19, απεικονίζονται τέσσερα διαγράμματα πίτας που κατανέμουν τις συνολικές κρατήσεις (ακυρωμένες και μη) βάσει του lead time, δηλαδή της διαφοράς μεταξύ της ημέρας κράτησης και της ημερομηνίας άφιξης. Τα τέσσερα διαστήματα είναι τα εξής: 0-7 ημέρες, 8-30 ημέρες (περίπου 1 μήνας), 31-90 ημέρες (περίπου 2-3 μήνες), και άνω των 90 ημερών (>3 μήνες). Παρατηρούμε ότι το ποσοστό των ακυρωμένων κρατήσεων αυξάνεται με την αύξηση του lead time. Άρα, όσο περισσότερες ημέρες απέχει η ημέρα κράτησης από την ημέρα άφιξης, τόσο μεγαλύτερες είναι οι πιθανότητες να ακυρώσει ο πελάτης την κράτηση του.

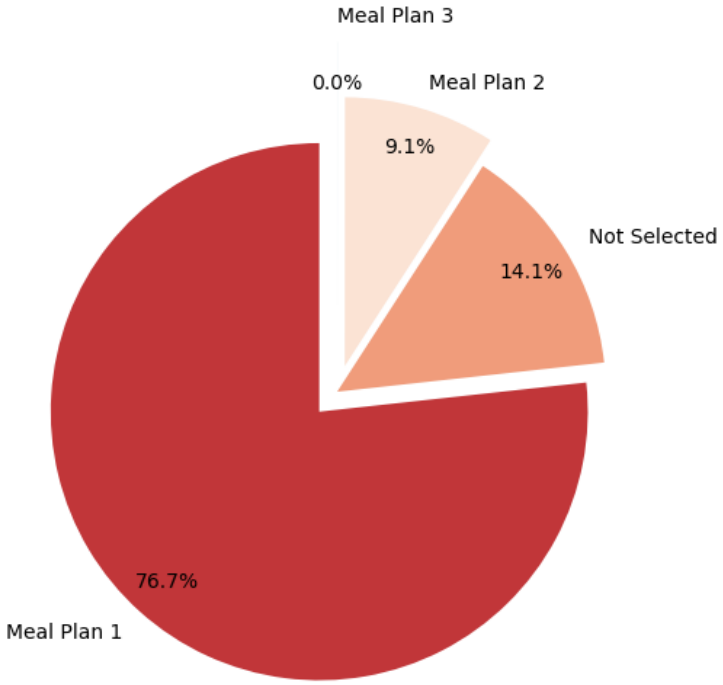
6.3 Κατανομή ατόμων της κράτησης



Εικόνα 27: Κατανομή αριθμού ενηλίκων και παιδιών στην κράτηση

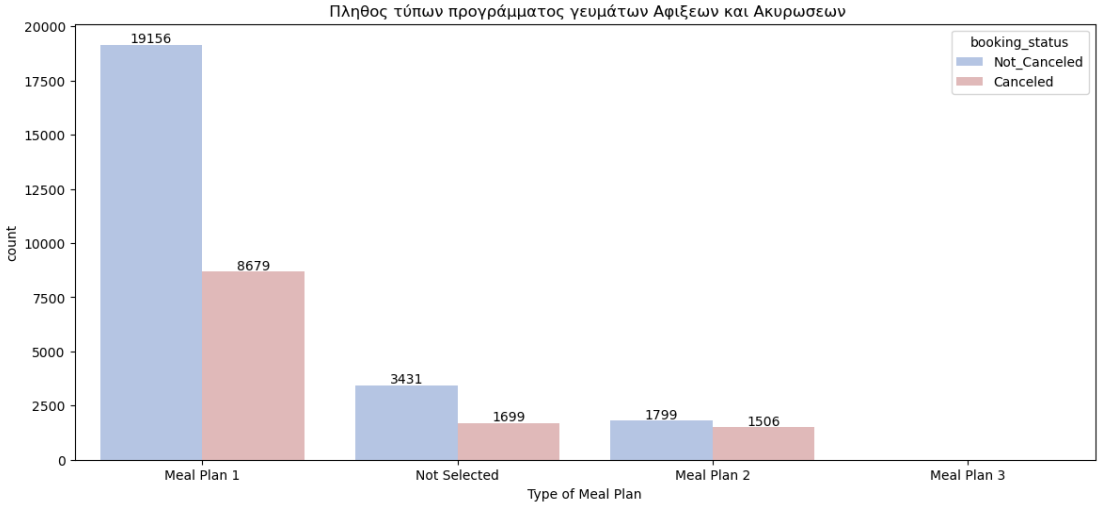
Από τα παραπάνω διαγράμματα φαίνεται ότι ο συνήθης τύπος ατόμων που πρόκειται να διαμείνει στην κράτηση είναι συνήθως ένα ζευγάρι ενηλίκων χωρίς παιδιά. Ο αριθμός των ενηλίκων που διατηρούν την κράτηση υπερτερεί εκείνων που ακυρώνουν, είτε πρόκειται για 1, 2, 3 ή περισσότερους ενήλικες. Όταν τα άτομα στην κράτηση περιλαμβάνουν και παιδί, φαίνεται ότι η πιθανότητα ακύρωσης είναι ελαφρώς χαμηλότερη από την πιθανότητα διατήρησης της κράτησης, υποδηλώνοντας ότι η παρουσία παιδιών δεν παίζει καθοριστικό ρόλο στην απόφαση ακύρωσης της κράτησης.

6.4 Κατανομή προγράμματος γεύματος



Εικόνα 28: Κατανομή προγράμματος γεύματος (Meal plan)

Κατά τη διάρκεια της διαμονής του στο ξενοδοχείο, ο πελάτης έχει τη δυνατότητα να επιλέξει ένα από τα διαθέσιμα προγράμματα γευμάτων. Ενδεικτικά, το Meal Plan 1 αποτελεί τη δημοφιλέστερη επιλογή, καθώς το επιλέγει το 76.7% των πελατών. Αντίθετα, το Meal Plan 3 δεν έχει καμία προτίμηση από τους πελάτες, καθώς το ποσοστό επιλογής του ανέρχεται στο 0%.

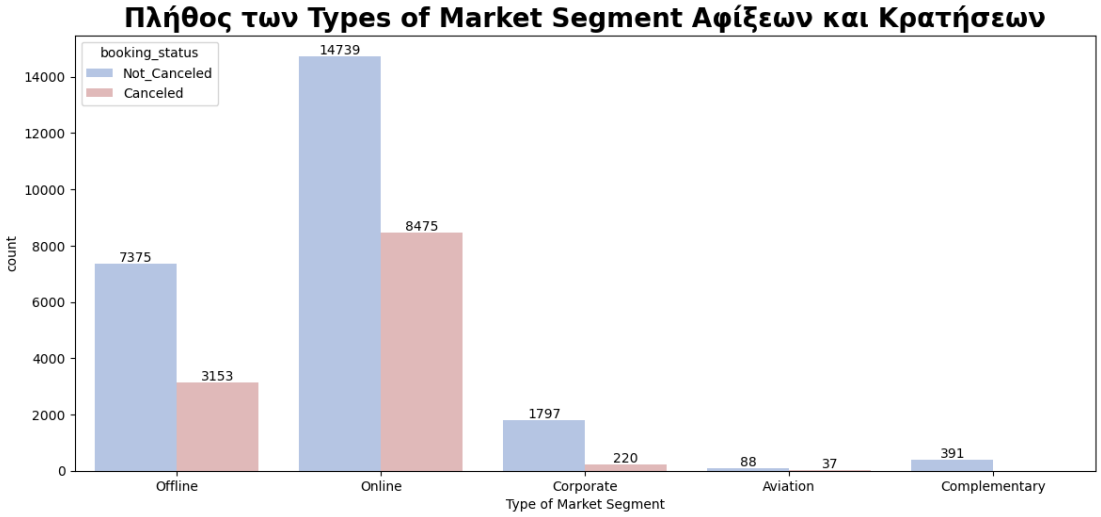


Εικόνα 29: Κατανομή των προγραμμάτων γεύματος με βάση την πιθανότητα ακύρωσης

Οι πελάτες που επιλέγουν το Meal type 1 ανέρχονται σε 27.835, από τους οποίους 8679 τελικά ακυρώνουν την κράτησή τους. Αυτό σημαίνει ότι το 31% των πελατών που επιλέγουν το Meal type 1 τελικά ακυρώνουν. Αντίστοιχα, οι πελάτες που δεν επιλέγουν κανένα πρόγραμμα γευμάτων ανέρχονται σε 5.130, με ποσοστό ακύρωσης 33%, καθώς 1.699 από αυτούς τελικά ακυρώνουν την κράτησή τους. Τέλος, οι πελάτες που έχουν επιλέξει το Meal type 2 είναι 3.305, και το ποσοστό ακύρωσής τους φτάνει το 46%.

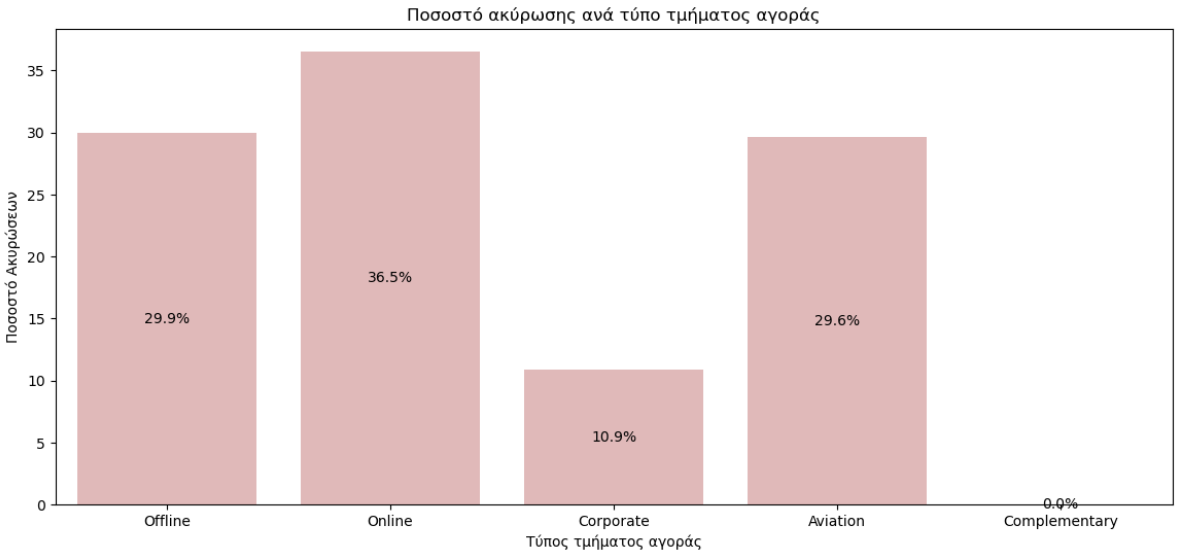
Από αυτά τα στοιχεία, προκύπτει ότι οι πελάτες που επιλέγουν το Meal type 2 έχουν υψηλότερο ποσοστό ακύρωσης σε σύγκριση με αυτούς που επιλέγουν το Meal type 1, ενώ οι πελάτες που δεν επιλέγουν κανένα πρόγραμμα γεύματος βρίσκονται κάπου στη μέση.

6.4 Κατανομή τύπου τμήματος αγοράς



Εικόνα 30: Πλήθος τύπων τμήματος αγοράς

Το τμήμα αγοράς που ξεχωρίζει περισσότερο είναι το online booking, όπου οι πελάτες πραγματοποιούν τις κρατήσεις τους μέσω του διαδικτύου. Ωστόσο, εντοπίζουμε και το υψηλότερο ποσοστό ακύρωσης σε αυτό το τμήμα, το οποίο φτάνει το 36.5%. Είναι αξιοσημείωτο να σημειώσουμε ότι στις περιπτώσεις του τύπου "complementary" (συμπληρωματική κράτηση) δεν υπάρχουν ακυρώσεις, με το ποσοστό ακύρωσης να είναι μηδενικό. Αυτό φαίνεται λογικό, καθώς η αρχική κράτηση έχει ήδη πραγματοποιηθεί, και εμφανίζεται μια επιπλέον κράτηση ως συμπληρωματική, όπως για παράδειγμα επιπλέον ημέρες διαμονής.

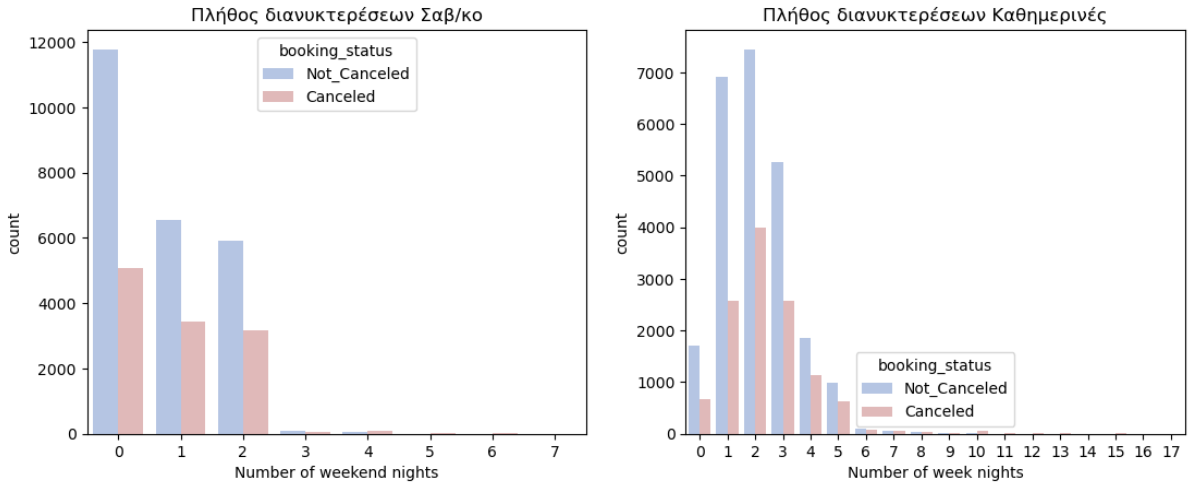


Εικόνα 31: Ποσοστό ακύρωσης ανά τύπο τμήματος αγοράς

6.5 Κατανομή ημερών κράτησης

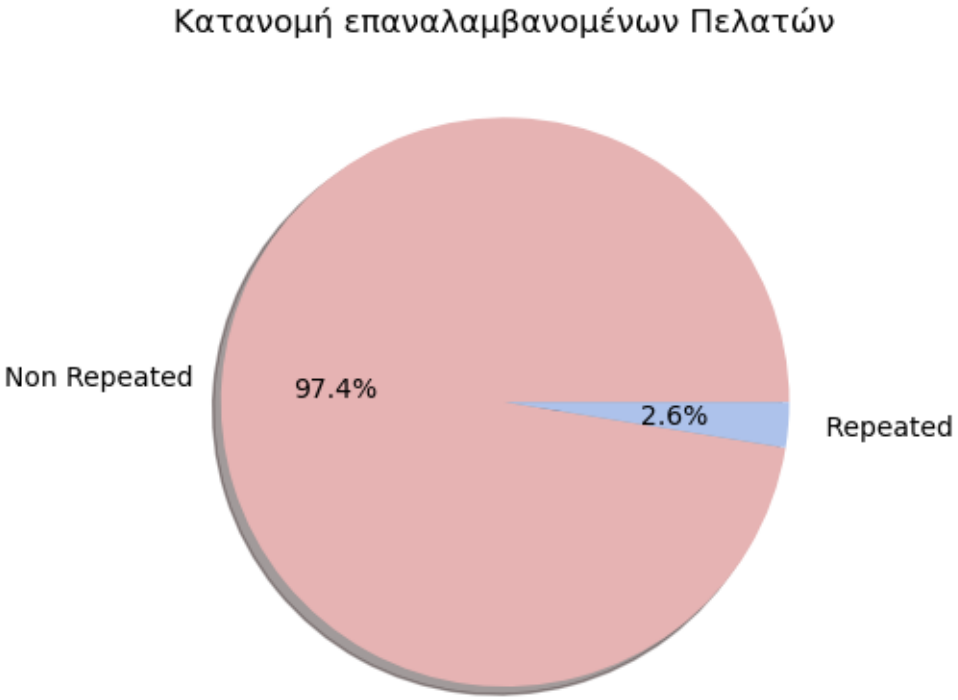
Στα διαγράμματα εντοπίζεται ότι οι κρατήσεις που περιλαμβάνουν τουλάχιστον μια ημέρα του Σαββατοκύριακου είναι λιγότερες, καθώς οι περισσότερες κρατήσεις δεν περιλαμβάνουν ούτε Σάββατο ούτε Κυριακή. Επιπλέον, η μεγαλύτερη ζήτηση συναντάται κυρίως για κρατήσεις που διαρκούν 1 έως 3 καθημερινές ημέρες. Σε αυτό το εύρος, το ποσοστό συντήρησης υπερτερεί κατά πολύ του ποσοστού ακυρώσεων.

Γενικά, παρατηρείται ότι οι κρατήσεις με μικρό αριθμό διανυκτερεύσεων και κατα τη διάρκεια της εβδομάδας έχουν υψηλότερο ποσοστό συντήρησης σε σύγκριση με το ποσοστό ακυρώσεων.

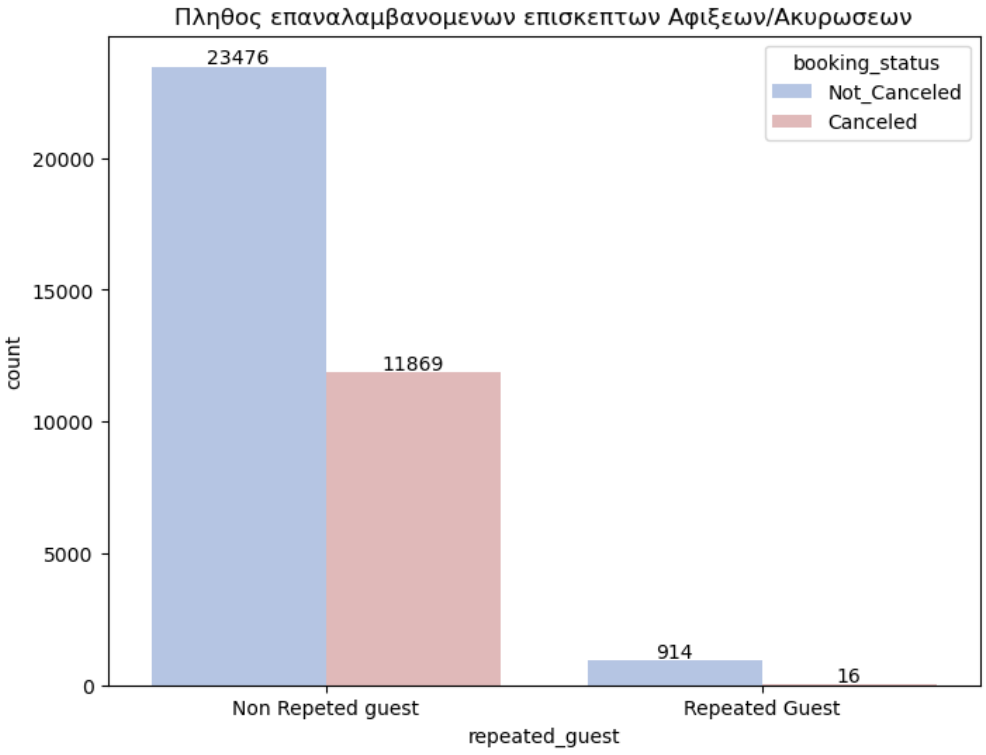


Εικόνα 32: Πλήθος διανυκτερεύσεων σαββ/κο και καθημερινές

6.6 Κατανομή επαναλαμβανόμενου πελάτη

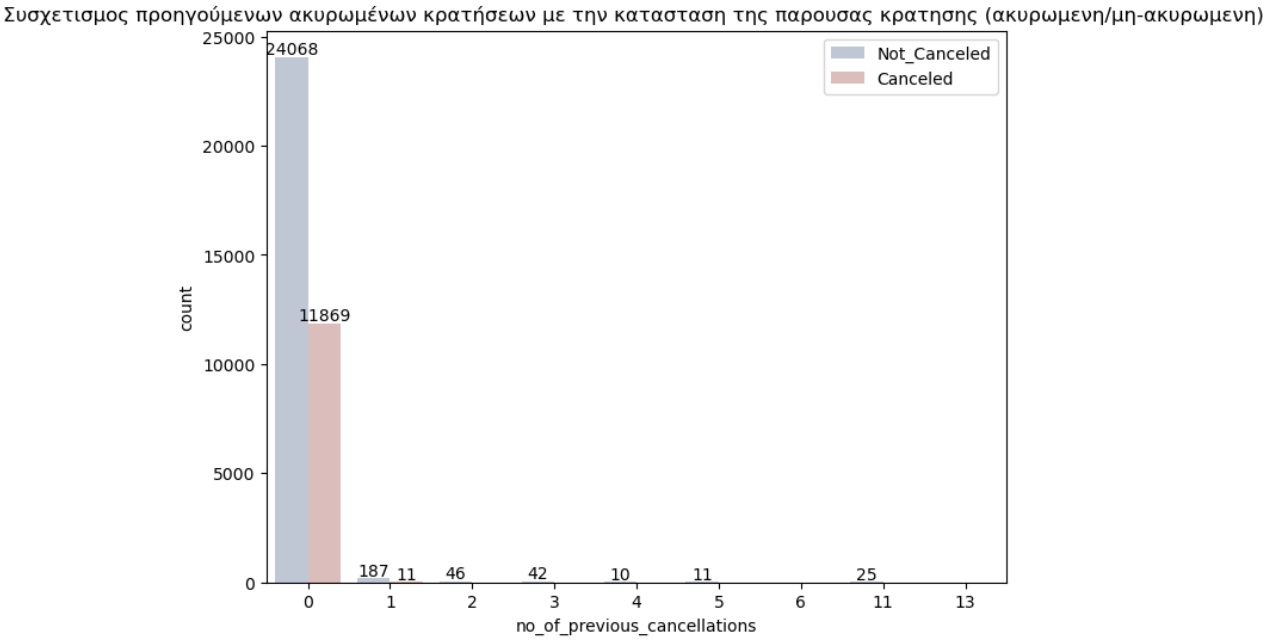


Εικόνα 33: Κατανομή επαναλαμβανομένων Πελατών



Εικόνα 34: Πλήθος επαναλαμβανομένων πελατών

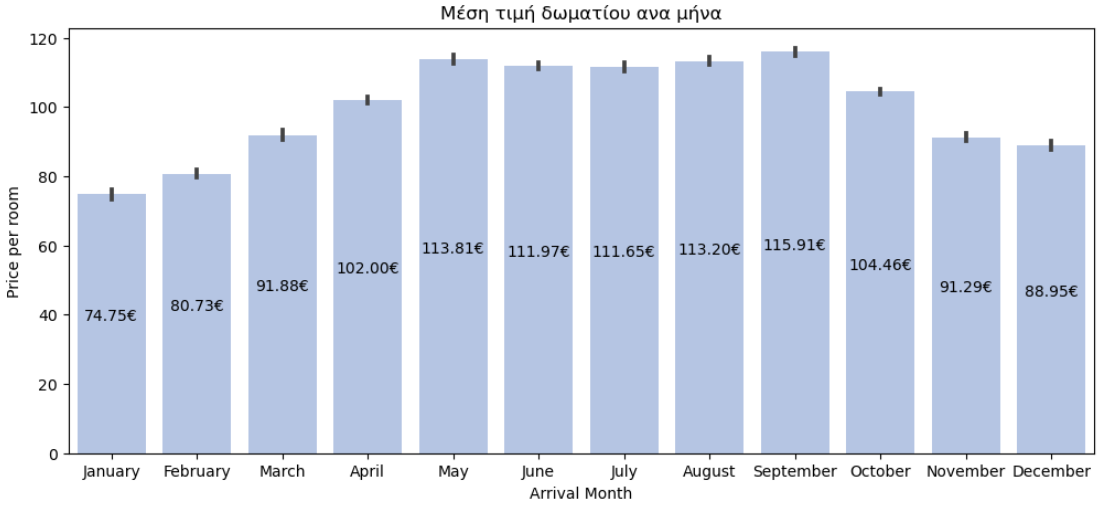
Το συγκεκριμένο ξενοδοχείο όπου αντλήθηκαν τα δεδομένα , βασίζεται κυρίως σε μη επαναλαμβανόμενους πελάτες, καθώς αυτοί αντιστοιχούν στο 97.4% του συνόλου. Ως προς το υπόλοιπο 2.6%, που αποτελεί το ποσοστό των επαναλαμβανόμενων πελατών, το ποσοστό ακύρωσης των κρατήσεών τους φτάνει το 1.7%. Αυτό σημαίνει πως η πιθανότητα ακύρωσης για έναν επαναλαμβανόμενο πελάτη είναι εξαιρετικά χαμηλή. Αυτό μπορεί να οδηγήσει σε σημαντικά κέρδη, παρά το μικρό ποσοστό επαναλαμβανόμενων πελατών.



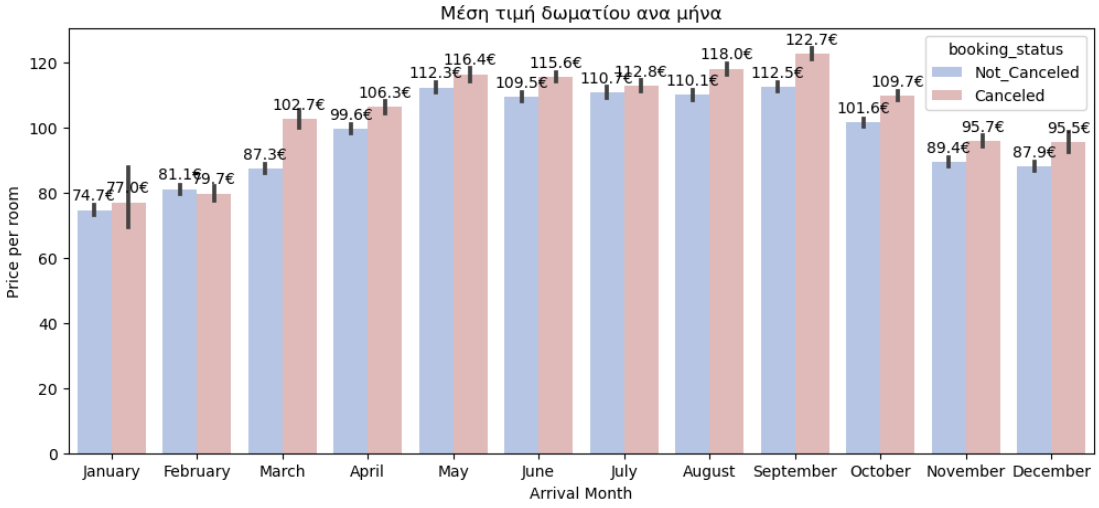
Εικόνα 35: Συσχετισμός προηγούμενων κρατήσεων με την κατάσταση της παρούσας κράτησης

Παρατηρείται ότι ο αριθμός προηγούμενων κρατήσεων είναι μηδενικός στην πλειοψηφία των καταγραφών (35937 στο σύνολο τους), καθώς οι επαναλαμβανόμενοι πελάτες αντιπροσωπεύουν μια μικρή μερίδα στο σύνολο των δεδομένων. Επομένως για να υπάρχουν επαρκής προηγούμενες κρατήσεις πρέπει να υπάρχει ο ανάλογος αριθμός πελατών που έχουν ξανά επισκεφτεί το ξενοδοχείο. Οι μεταβλητές που σχετίζονται με τους επαναλαμβανόμενους πελάτες και τις προηγούμενες κρατήσεις αποτελούν μια μειονότητα στο σύνολο των δεδομένων. Κατά την εξέταση αυτών των μεταβλητών λοιπόν, δεν μπορεί να παραχθεί κάποιο συμπέρασμα για την πιθανότητα ακύρωσης.

6.7 Μεταβλητή μέση τιμή ανά ημέρα κράτησης



Εικόνα 36: Μέση τιμή δωματίου ανά μήνα



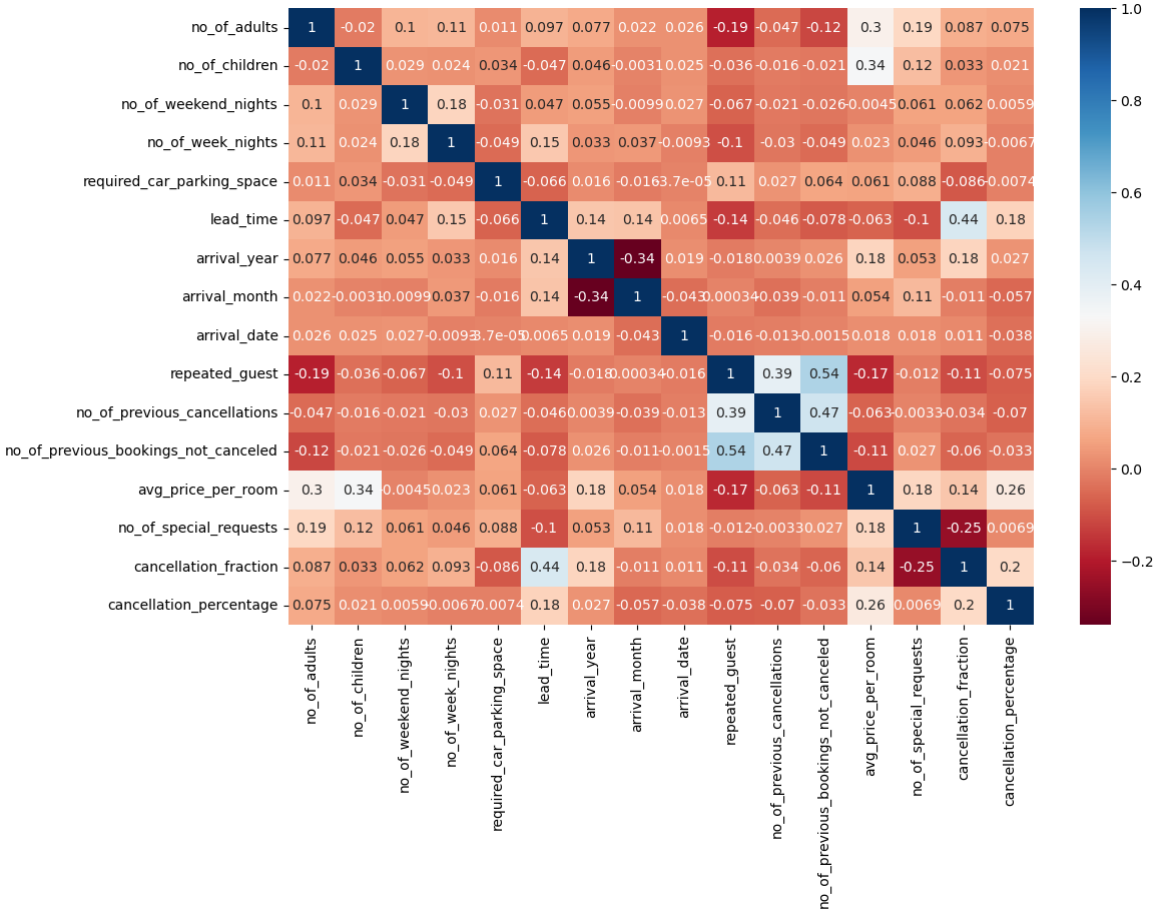
Εικόνα 37: Μέση τιμή δωματίου ανά μήνα με βάση την κατάσταση κράτησης

Η μέση τιμή του δωματίου αποτελεί κρίσιμη μεταβλητή που επηρεάζει σημαντικά την πιθανότητα ακύρωσης μιας κράτησης. Αναλύοντας τη μέση τιμή του δωματίου κατά μήνα, παρατηρούμε την εξής εξέλιξη: Η μέση τιμή ξεκινά από τα 75€ τον Ιανουάριο και αυξάνεται σταδιακά προς τους καλοκαιρινούς μήνες. Ιδιαίτερα τον Μάιο, αγγίζει τη δεύτερη υψηλότερη τιμή, ενώ ο Σεπτέμβριος κατακτά την πρώτη θέση με 116€. Από το Σεπτέμβριο και μετά, σημειώνεται μια μικρή κάθοδος.

Σημαντικό είναι το εύρημα ότι η μέση τιμή των ακυρωμένων κρατήσεων είναι πάντα υψηλότερη από των μη ακυρωμένων. Ειδικότερα, ο μέγιστος μέσος όρος της τιμής για ακυρωμένη κράτηση παρατηρείται τον Σεπτέμβριο, φτάνοντας τα 123€.

Αυτά τα δεδομένα υποδηλώνουν ότι οι πελάτες είναι πρόθυμοι να πληρώσουν υψηλότερη τιμή για το δωμάτιο τους τους καλοκαιρινούς μήνες και τον Σεπτέμβριο, ενδεχομένως λόγω αυξημένης ζήτησης. Ωστόσο, η υψηλότερη τιμή σχετίζεται και με υψηλότερο ποσοστό ακυρώσεων, πιθανότατα λόγω μεγαλύτερου κινδύνου και αβεβαιότητας από τους πελάτες.

6.8 Heat Map



7. Προ επεξεργασία Data

7.1 Δημιουργία νέων μεταβλητών

Δημιουργία νέων παραμέτρων στο σύνολο δεδομένων αποτελεί σημαντικό στάδιο στην προ επεξεργασία και βελτιστοποίηση των μοντέλων μηχανικής μάθησης, όπως το XGBoost. Αυτή η διαδικασία,επίσης γνωστή ως feature engineering (μηχανική

χαρακτηριστικών) περιλαμβάνει τη μετατροπή ακατέργαστων δεδομένων σε ουσιαστικά και ενημερωτικά χαρακτηριστικά, τα οποία μπορούν να χρησιμοποιηθούν αποτελεσματικά από αλγόριθμους μηχανικής μάθησης για την πραγματοποίηση ακριβών προβλέψεων. Το feature engineering είναι κρίσιμο για τη γεφύρωση του χάσματος μεταξύ των αρχικών δεδομένων και των πληροφοριών που αντλούνται από τα μοντέλα μηχανικής μάθησης.

Μετά από πολλές δοκιμές με διάφορες μεταβλητές και συνδυασμούς τους, αναζητώντας το βέλτιστο accuracy και best score, επιλέχθηκαν 4 μεταβλητές:

- Booking Duration (Διάρκεια κράτησης)
- Difference in price and average price (Διαφορά τιμής και μέσης τιμής του μήνα κράτησης)
- Arrival_date_format (Ημερομηνία Κράτησης)
- is Holidays (Διακοπές: εάν μέσα στις ημερομηνίες κρατήσεων περιέχονται ημέρες αργίας, γιορτών κ.λπ.)

Αυτές οι μεταβλητές επιλέχθηκαν με βάση την απόδοσή τους στο μοντέλο, προσδίδοντας του ισχυρά προγνωστικά χαρακτηριστικά και συνεισφέροντας στη βελτιστοποίηση των αποτελεσμάτων.

7.1.1 Κριτήρια αξιολόγησης νέων μεταβλητών

Η προσθήκη νέων χαρακτηριστικών σε ένα σύνολο δεδομένων αποτελεί σημαντικό βήμα για τη βελτιστοποίηση των μοντέλων μηχανικής μάθησης, όπως προαναφέρθηκε. Παρόλα αυτά, δεν είναι δεδομένο ότι κάθε νέο χαρακτηριστικό θα συμβάλει θετικά στην απόδοση του μοντέλου. Σε αυτό το πλαίσιο, χρησιμοποιούνται δύο μετρικές μεθόδοι για την αξιολόγηση των χαρακτηριστικών, με σκοπό να κριθεί εάν η προσθήκη τους έχει θετική επίδραση στο μοντέλο ή εάν δεν αξίζει να συμπεριληφθούν στο σύνολο δεδομένων.

I. Best Score από Grid Search
II. Accuracy Score στα Δεδομένα Δοκιμής (test data)

7.1.1.1 Best Score από Grid Search

Η Best_score επιστρέφει τη βαθμολογία που έχει πετύχει το βέλτιστο μοντέλο που προέκυψε από την αναζήτηση υπερπαραμέτρων. Αυτή η βαθμολογία αντιπροσωπεύει την απόδοση του μοντέλου χρησιμοποιώντας τη μετρική αξιολόγησης που έχει καθοριστεί στην παράμετρο scoring κατά την διάρκεια της αναζήτησης. Ένα υψηλότερο best_score υποδηλώνει καλύτερη απόδοση του μοντέλου σύμφωνα με την επιλεγμένη μετρική αξιολόγησης, στη συγκεκριμένη περίπτωση το ROC AUC.

Η βαθμολογία αυτή δεν δείχνει απαραίτητα την επίδραση της προσθήκης ενός νέου χαρακτηριστικού, αλλά απλώς τονίζει το πόσο καλά προσαρμόζεται το μοντέλο στα δεδομένα με βάση τις υπερπαραμέτρους που έχουν επιλεγεί.

```
rand_search = GridSearchCV(estimator=xgb, param_grid=xgb_param_grid, n_jobs=2, cv=cv_f, verbose=1, scoring='roc_auc')

rand_search.fit(X_train,y_train)
```

```
print("Best parameters found:", rand_search.best_params_)
print("Best Cross-Validated Score:", rand_search.best_score_)
```

7.1.1.2 Accuracy score σε Δεδομένα δοκιμής (test data)

Το Accuracy Score είναι μια μετρική που χρησιμοποιείται για την αξιολόγηση της απόδοσης ενός μοντέλου μηχανικής μάθησης στα δεδομένα δοκιμής. Προσδιορίζει το ποσοστό των σωστών προβλέψεων συνολικά. Όσο πιο υψηλό είναι το Accuracy Score, τόσο καλύτερη θεωρείται η απόδοση του μοντέλου.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}$$

Εικόνα 38: Λόγος Accuracy

Στο πλαίσιο της αξιολόγησης ενός νέου χαρακτηριστικού, το Accuracy Score μπορεί να χρησιμοποιηθεί για να δειχθεί εάν η προσθήκη αυτού του χαρακτηριστικού βελτιώνει ή χειροτερεύει τη συνολική ακρίβεια του μοντέλου. Ένα υψηλό Accuracy Score ύστερα από την προσθήκη του νέου χαρακτηριστικού θα υποδηλώνει ότι το μοντέλο αποτυγχάνει λιγότερο στην πρόβλεψη των κλάσεων. Από την άλλη πλευρά, μια μείωση του Accuracy Score μπορεί να υπονοεί ότι το νέο χαρακτηριστικό δεν προσφέρει ικανοποιητικά στοιχεία για την πρόβλεψη.

Συνολικά, το Accuracy Score είναι ένα χρήσιμο μέτρο για την εκτίμηση του γενικού αντίκτυπου της προσθήκης ενός νέου χαρακτηριστικού στην επίδοση του μοντέλου.

```
rand_search = GridSearchCV(estimator=xgb, param_grid=xgb_param_grid, n_jobs=2, cv=cv_f, verbose=1, scoring='roc_auc')

rand_search.fit(X_train,y_train)

preds= rand_search.predict(X_test)

print("The accuracy score is: " , accuracy_score(y_test, preds))
```

7.1.2 Διάρκεια κράτησης (Booking Duration)

Ένα σημαντικό στοιχείο που μπορεί να αντληθεί από τα δεδομένα μας είναι η συνολική διάρκεια κράτησης. Μέσα από το σύνολο των δεδομένων, έχουμε πληροφορίες για τον αριθμό των καθημερινών ημερών και των ημερών Σαββατοκύριακων κατά τη διάρκεια μιας κράτησης. Η δημιουργία μιας μεταβλητής που συνδυάζει αυτά τα δύο χαρακτηριστικά προσφέρει νέες πληροφορίες που μπορεί να βελτιώσουν το μοντέλο μας.

```
orig_data.insert(loc=orig_data.columns.get_loc('no_of_week_nights') + 1,
                 column='booking_duration',
                 value= orig_data['no_of_weekend_nights'] + orig_data['no_of_week_nights'])
```

Όπως φαίνεται στον παραπάτω κώδικα, γίνεται η εισάγωγή μιας νέας στήλης στα αρχικά δεδομένα με την ονομασία "booking duration". Το loc επιτρέπει τον καθορισμό της θέσης όπου θα εισαχθεί αυτή η νέα μεταβλητή στο σύνολο των

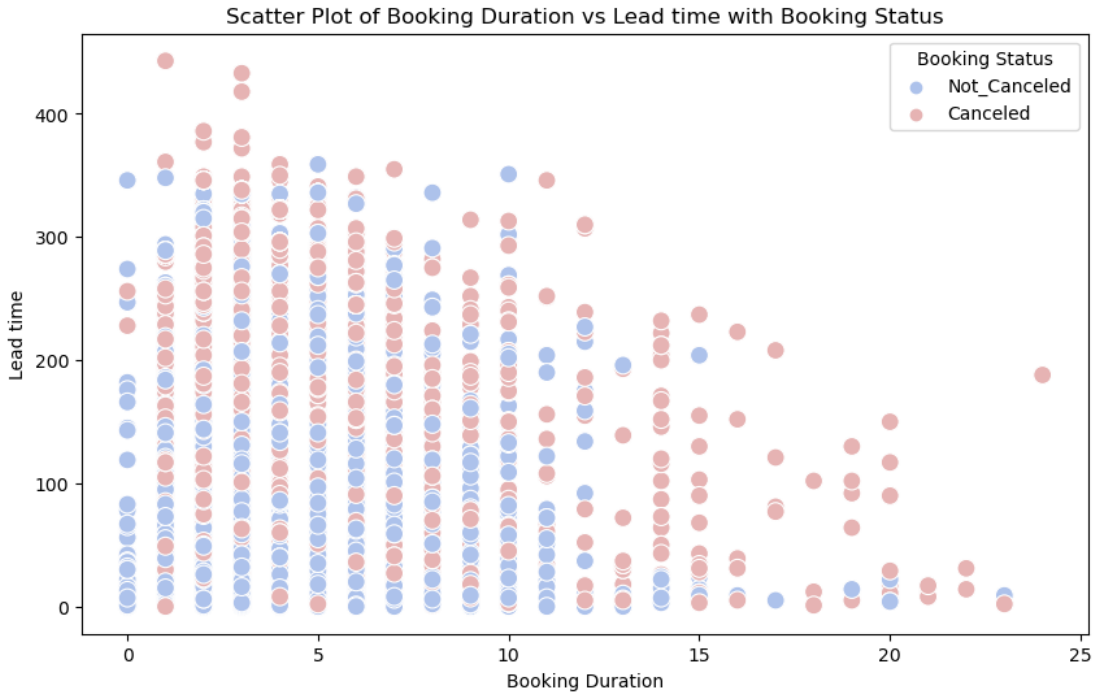
δεδομένων. Αυτή η νέα μεταβλητή αποτελεί μια συνδυαστική προσθήκη που μπορεί να προσφέρει πολύτιμες πληροφορίες για τη διάρκεια των κρατήσεων, βελτιώνοντας έτσι την απόδοση του μοντέλου μας.

no_of_weekend_nights	no_of_week_nights	booking_duration
1	2	3
2	3	5
0	2	2
1	1	2
0	2	2
...	...	...
2	6	8
1	3	4
2	6	8
0	3	3
1	2	3

Μετά την προσθήκη αυτής της μεταβλητής, αξίζει να επισημανθούν τα αποτελέσματα που προκύπτουν από το διάγραμμα διασποράς μεταξύ της νέας μεταβλητής και του lead time (διαφορά ημέρας κράτησης από την ημέρα άφιξης). Οι παρατηρήσεις με κόκκινο χρώμα αναπαριστούν ακυρωμένες κρατήσεις, ενώ με μπλε χρώμα αναπαριστούν κρατήσεις που τελικά πραγματοποιήθηκαν.

Παρατηρείται ότι όσο πιο δεξιά στο διάγραμμα βρίσκονται οι παρατηρήσεις, δηλαδή από όσο περισσότερες ημέρες αποτελείται η κράτηση , η πλειοψηφία των κρατήσεων αυτών κατέληξε σε ακύρωση.

Από την αριστερή πλευρά, στο πάνω τεταρτημόριο του πίνακα, οι κόκκινες κουκίδες πάλι πλειοψηφούν. Ενώ ο αριθμός των διανυκτερεύσεων είναι μειωμένος, το lead time είναι πολύ υψηλό. Στην αριστερά και κάτω τεταρτημόριο, κυριαρχούν πολλές μπλε κουκίδες, δηλαδή μη ακυρωμένες κρατήσεις. Φαίνεται λοιπόν ότι όταν ο πελάτης κάνει την κράτηση με μικρό lead time και περιορισμένο αριθμό διανυκτερεύσεων, είναι πιο πιθανό να μην ακυρώσει.



Εικόνα 39: Διάγραμμα Διασποράς μεταξύ της διάρκειας κράτησης και της απόστασης ημέρας κράτησης απο την ημέρα άφιξης

7.1.3 Διαφορά τιμής και μέσης τιμής του μήνα κράτησης (Difference in price and average price)

Το σύνολο δεδομένων περιλαμβάνει ένα χαρακτηριστικό που παρέχει τη μέση τιμή κράτησης ανά ημέρα. Με βάση αυτό ένα ιδιαίτερα ενδιαφέρον βήμα είναι η εξαγωγή νέας πληροφορίας, η διαφορά τιμής (μέση τιμή κράτησης ανά ημέρα) με τη μέση τιμή κράτησης τον μήνα της κράτησης.

Το πρώτο βήμα για την επίτευξη αυτού είναι ο υπολογισμός της μέσης τιμής τον μήνα κράτησης:

orig_data.groupby('arrival_month')['avg_price_per_room'].transform('mean'))

Στη συνέχεια, ακολουθεί η αφαίρεση αυτής της τιμής από την ήδη υπάρχουσα στήλη του συνόλου avg_price_per_room:

```
orig_data.insert(loc=orig_data.columns.get_loc('avg_price_per_room') + 1,
                 column='price_difference',
                 value=orig_data['avg_price_per_room'] -
                       orig_data.groupby('arrival_month')['avg_price_per_room'].transform('mean'))
```

Όταν η τιμή της νέας μεταβλητής price_difference είναι αρνητική, αυτό σημαίνει ότι η τιμή της κράτησης ήταν χαμηλότερη από τη μέση τιμή κρατήσεων τον συγκεκριμένο μήνα, και έτσι ο πελάτης βγήκε κερδισμένος. Αντίθετα, όταν έχουμε θετική τιμή, η τιμή κράτησης υπερτερεί της μέσης τιμής του μήνα, και επομένως ο πελάτης έδωσε περισσότερα χρήματα για την κράτησή του σε σχέση με τον μέσο όρο των τιμών των κρατήσεων τον συγκεκριμένο μήνα.

avg_price_per_room	price_difference
65.00	-39.460873
106.68	15.394574
100.00	-13.810677
94.50	-7.498801
115.00	-0.906916
...	...
167.80	54.596016
90.95	-13.510873
98.39	-13.259318
94.50	-7.498801
161.67	72.717815

7.1.4 Διακοπές (is Holidays)

Μια πρόσθετη πληροφορία που μπορεί να εκμαιευτεί από το σύνολο δεδομένων είναι εάν συμπεριλαμβάνονται ημέρες αργίας, εορτές και γενικότερα διακοπών στις ημέρες των κρατήσεων. Σύμφωνα με το Kaggle.com, δεδομένου ότι πρόκειται για πραγματικά δεδομένα, έχουν διαγραφεί όλες οι πληροφορίες που αφορούν την ταυτότητα του ξενοδοχείου ή των πελατών. Παρόλα αυτά, ο συγγραφέας του συγκεκριμένου dataset στο Kaggle αναφέρει ότι πρόκειται για δεδομένα ξενοδοχείων της Πορτογαλίας. Αυτή είναι μια πολύ βασική πληροφορία προκειμένου να καθορίσουμε τις διακοπές της εκάστοτε χώρας.

Αρχικά γίνεται η εγκατάσταση της βιβλιοθήκης holidays, μια γρήγορη, αποτελεσματική βιβλιοθήκη της Python για τη δημιουργία συγκεκριμένων διαστημάτων διακοπών για κάθε χώρα αλλά και μικρότερα παραρτήματα αυτής (π.χ. πολιτεία ή επαρχία).

```
pip install holidays
```

Στην συνεχεια γινεται η ενσωμάτωση της βιβλιοθήκης holidays, καθώς και των βιβλιοθηκών datetime και calendar

```
from datetime import date
import holidays
import calendar
```

Το επόμενο βήμα περιλαμβάνει την εισαγωγή μιας νέας στήλης ή χαρακτηριστικού, το "arrival_date_format", στο σύνολο δεδομένων. Αυτό το χαρακτηριστικό λειτουργεί ως βοηθητικό χαρακτηριστικό για τον βέλτιστο υπολογισμό του χαρακτηριστικού διακοπών.

Με την αποδόμηση του ορισμού της μεταβλητής value, η διαδικασία γίνεται πιο ευδιάκριτη:

Η μέθοδο .astype(str) χρησιμοποιείται για να μετατρέψει τις τιμές στις στήλες arrival_year, arrival_month, και arrival_date σε συμβολοσειρές. Αυτές οι συμβολοσειρές συνδυάζονται με το σύμβολο της παύλας για να δημιουργηθεί η συνολική ημερομηνία άφιξης.

Στην συνέχεια, για τη μετατροπή αυτής της συνολικής συμβολοσειράς σε χρονοσφραγίδα (timestamp) χρησιμοποιείται η συνάρτηση pd.to_datetime() .

```
value = pd.to_datetime(
       orig_data['arrival_year'].astype(str) + '-' +
       orig_data['arrival_month'].astype(str) + '-' +
       orig_data['arrival_date'].astype(str), errors = 'coerce')

orig_data.insert(loc=orig_data.columns.get_loc('arrival_date') + 1, column = 'arrival_date_format', value = value )
```

Προστίθεται το νέο χαρακτηριστικό "Is Holiday", το οποίο είναι λογική μεταβλητή (boolean) με αρχική τιμή False. Στη συνέχεια, κάθε γραμμή εγγραφής ελέγχεται ώστε να διαπιστωθεί αν η ημερομηνία του χαρακτηριστικού "arrival_date_format" εμπίπτει στο εύρος των ημερών διακοπών στην Πορτογαλία ("holidays_portugal"). Εάν ισχύει, τότε η λογική μεταβλητή παίρνει την τιμή True, διαφορετικά παραμένει False.

```
holidays_portugal = holidays.Portugal(years=2018)

# Create a binary indicator for holidays for each booking
orig_data.insert(loc=orig_data.columns.get_loc('arrival_date_format') + 1, column = 'is_holiday', value = False )


for idx, row in orig_data.iterrows():
    arrival_date = row['arrival_date_format']
    total_nights = row['no_of_weekend_nights'] + row['no_of_week_nights']
    departure_date = arrival_date + pd.DateOffset(days=total_nights - 1)

    # Check if arrival_date is not NaT
    if pd.notna(arrival_date) & pd.notna(departure_date):

     # Check if any day of the booking period is a holiday
        if any(date.date() in holidays_portugal for date in pd.date_range(arrival_date, departure_date, freq ='D')):
            orig_data.at[idx, 'is_holiday'] = True
```

arrival_date_format	is_holiday
2017-10-02	False
2018-11-06	False
2018-02-28	False
2018-05-20	False
2018-04-11	False
2018-09-13	False
2017-10-15	False
2018-12-26	False
2018-07-06	False
2018-10-18	False
2018-09-11	False
2018-04-30	True
2018-11-26	False
2018-11-20	False
2017-10-20	False

Συνολικά 4177 ημέρες των συνολικών ημερών κρατήσεων ανήκουν στο εύρος των ημερών διακοπών στην Πορτογαλία

```
orig_data['is_holiday'].value_counts()
```

```
False    32098
True      4177
```

8. Μηχανική Μάθηση

Λαμβάνοντας υπόψη τις απαιτήσεις και τις ανάγκες του προβλήματος που αναπτύσσετε στην εργασία, καθώς επίσης τα χαρακτηριστικά και τις δυνατότητες του αλγορίθμου XGBoost, ο συγκεκριμένος αλγόριθμος επιλέχθηκε, μελετήθηκε και εφαρμόστηκε στην παρούσα εργασία.

8.1 Ενσωμάτωση βιβλιοθηκών

Το πρώτο και ίσως το πιο σημαντικό βήμα, χωρίς το οποίο δεν είναι δυνατή η εκτέλεση του μοντέλου XGBoost, είναι η ενσωμάτωση των απαραίτητων βιβλιοθηκών.

```
import xgboost as xgb
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Λίγα λόγια για αυτές:

- Η βιβλιοθήκη xgboost περιέχει όλες τις δυνατότητες και τη δημιουργία του XGBoost μοντέλου (eXtreme Gradient Boosting) και είναι ανοικτού κώδικα.

- Η βιβλιοθήκη NumPy, είναι μία από τις πιο ευρέως χρησιμοποιούμενες βιβλιοθήκες Python ανοιχτού κώδικα, για επιστημονικούς υπολογισμούς. Επιτρέπει την υποστήριξη πολυδιάστατων data μέσω των ενσωματωμένες μαθηματικές λειτουργίες που διαθέτει.

- Το Matplotlib αποτελεί μια βιβλιοθήκη για τη δημιουργία σταθερών, δυναμικών και κινούμενων Python οπτικοποιήσεων. Το Matplotlib είναι ανοιχτού κώδικα . Ευρεία γκάμα γραφημάτων, όπως ιστόγραμμα, ραβδόγραμμα και διάγραμμα διασποράς (scatter plot) είναι διαθέσιμα μέσω της Matplotlib, με σκοπό την απεικόνιση και οπτικοποίηση δεδομένων .

- Το Pandas είναι μια βιβλιοθήκη ανοικτού κώδικα που χρησιμοποιείται εκτενώς από τους data scientists. Κυρίως εκτιμάται για τον ρόλο του στην ανάλυση και επεξεργασία των δεδομένων, καθώς και στον καθαρισμό τους. Ανάμεσα στα βασικά του χαρακτηριστικά: τα DataFrames ξεχωρίζουν, παρέχοντας αποτελεσματική δυνατότητα επεξεργασίας και διαχείρισης δεδομένων (data manipulation) και συνάμα περιλαμβάνουν ενσωματωμένη δυνατότητα αρίθμησης (data indexing). Επιπλέον, υποστηρίζει τη συγχώνευση και τη σύνδεση συνόλων δεδομένων υψηλής απόδοσης. Προσφέρει επίσης ευρεία γκάμα εργαλείων για τη διαχείριση δεδομένων μεταξύ διαφορετικών δομών, όπως αρχεία Excel, αρχεία κειμένου και CSV, καθώς και βάσεις δεδομένων SQL.

8.2 Σύνολο δεδομένων

```
orig_data = pd.read_csv("Hotel Reservations.csv")
orig_data.shape
orig_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36275 entries, 0 to 36274
Data columns (total 23 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Booking_ID                            36275 non-null  object
 1   no_of_adults                          36275 non-null  int64
 2   no_of_children                        36275 non-null  int64
 3   no_of_weekend_nights                  36275 non-null  int64
 4   no_of_week_nights                     36275 non-null  int64
 5   booking_duration                      36275 non-null  int64
 6   type_of_meal_plan                     36275 non-null  object
 7   required_car_parking_space            36275 non-null  int64
 8   room_type_reserved                    36275 non-null  object
 9   lead_time                             36275 non-null  int64
 10  arrival_year                          36275 non-null  int64
 11  arrival_month                         36275 non-null  int64
 12  arrival_date                          36275 non-null  int64
 13  arrival_date_format                   36238 non-null  datetime64[ns]
 14  is_holiday                            36275 non-null  bool
 15  market_segment_type                   36275 non-null  object
 16  repeated_guest                        36275 non-null  int64
 17  no_of_previous_cancellations          36275 non-null  int64
 18  no_of_previous_bookings_not_canceled  36275 non-null  int64
 19  avg_price_per_room                    36275 non-null  float64
 20  price_difference                      36275 non-null  float64
 21  no_of_special_requests                36275 non-null  int64
 22  booking_status                        36275 non-null  object
dtypes: bool(1), datetime64[ns](1), float64(2), int64(14), object(5)
memory usage: 6.1+ MB
```

Εικόνα 40: Σχήμα και Πληροφορίες συνόλου δεδομένων

Ακολουθεί η φόρτωση των πρωτότυπων δεδομένων μετονομασμένο διαφορετικά σε orig_data ώστε το πρωτότυπο σύνολο δεδομένων να μείνει ανέγγιχτο. Τα δεδομένα περιέχουν 36275 έγγραφες και αποτελούνται από 23 στήλες μετά την προσθήκη των νέων μεταβλητών.

Παρατηρώντας τον τύπο κάθε μεταβλητής, προκύπτει ότι υπάρχουν 5 μεταβλητές τύπου object, μια λογική μεταβλητή (bool) και μια μεταβλητη τύπου datetime. Οι υπόλοιπες στήλες είναι τύπου είτε int64 είτε float64, που υποδηλώνει αριθμητικές μεταβλητές.

Οι στήλες τύπου "object" συνήθως περιέχουν κατηγορικά δεδομένα, τα οποία αναπαρίστανται ως συμβολοσειρές. Κατά την επεξεργασία των δεδομένων, πολλά μοντέλα μηχανικής μάθησης όπως και αυτό του XGBoost απαιτούν αριθμητικές τιμές για να λειτουργήσουν αποτελεσματικά. Αριθμητικές τιμές επιτρέπουν τον υπολογισμό μαθηματικών λειτουργιών, τον υπολογισμό αποστάσεων μεταξύ σημείων, και άλλες μεθόδους που είναι απαραίτητες για την εκπαίδευση των μοντέλων. Επομένως, η μετατροπή των στηλών "object"

σε αριθμητικά δεδομένα είναι σημαντική για να εξασφαλιστεί ότι οι αλγόριθμοι μπορούν να αντιληφθούν, επεξεργαστούν και εκμεταλλευτούν αυτές τις πληροφορίες κατηγορίας κατά τη διάρκεια της εκπαίδευσης.
Η μετατροπή αυτή μπορεί να γίνει με διάφορους τρόπους, όπως η συνάρτηση LabelEncoder της βιβλιοθήκης sklearn.

```
from sklearn.preprocessing import LabelEncoder
```

```
le = LabelEncoder()
le_count = 0
for col in orig_data:
    if orig_data[col].dtype == 'object' or orig_data[col].dtype == 'datetime64[ns]' or orig_data[col].dtype == 'bool' :
        orig_data[col] = le.fit_transform(orig_data[col])
        le_count += 1
print('%d στήλες έχουν μετατραπεί.' % le_count)
```

```
7 στήλες έχουν μετατραπεί.
```

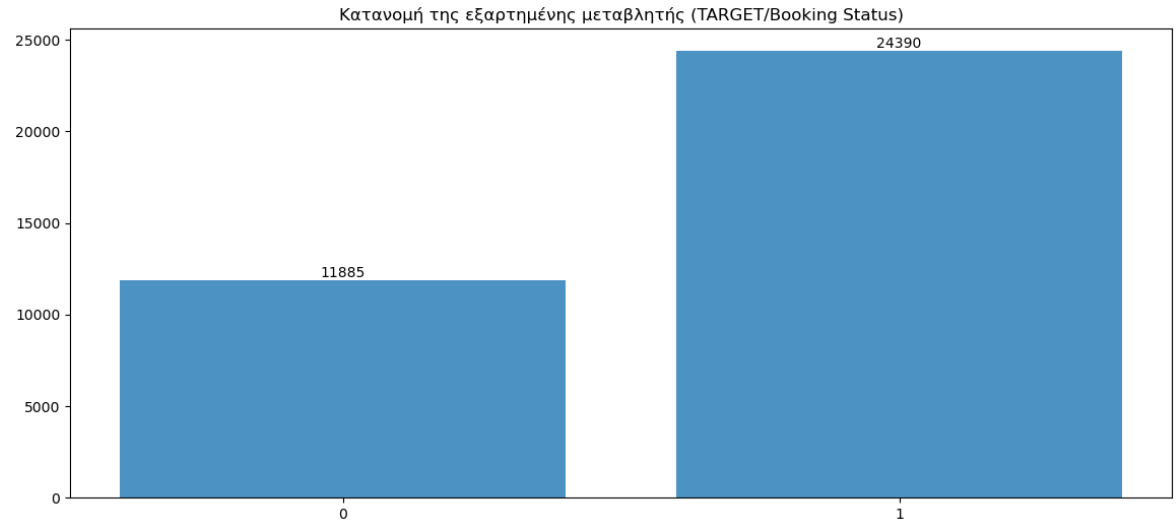
Για τον μετασχηματισμό των δεδομένων από κατηγορικά σε αριθμητικά δεδομένα, πρέπει προηγουμένως να δημιουργηθεί ένα αντικείμενο για τη συνάρτηση LabelEncoder. Επίσης, καθαρά για οπτικούς σκοπούς δημιουργείται μια μετρητική μεταβλητή le_count που αρχικοποιείται στο μηδέν και αυξάνεται κατά ένα με κάθε μετατροπή μιας στήλης τύπου "object" σε αριθμητικά δεδομένα. Έτσι όταν η συνάρτηση έχει τρέξει για κάθε χαρακτηριστικό που πληροί τα κριτήρια, δηλαδή ανήκει σε έναν από τους τύπους: object, datetime, bool, τότε εμφανίζεται ο συνολικός αριθμός μετατροπών.
Μια συνοπτική ματιά αμέσως μετά την μετατροπή.

type_of_meal_plan	required_car_parking_space	room_type_reserved	lead_time	...	arrival_date_format	is_holiday	market_segment_type	repeated_guest
36275.000000	36275.000000	36275.000000	36275.000000	...	36275.000000	36275.000000	36275.000000	36275.000000
0.515644	0.030986	0.708890	85.232557	...	327.918429	0.115148	3.552447	0.025637
1.048131	0.173281	1.399851	85.930817	...	139.854142	0.319205	0.681536	0.158053
0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	17.000000	...	242.000000	0.000000	3.000000	0.000000
0.000000	0.000000	0.000000	57.000000	...	346.000000	0.000000	4.000000	0.000000
0.000000	0.000000	0.000000	126.000000	...	445.000000	0.000000	4.000000	0.000000
3.000000	1.000000	6.000000	443.000000	...	549.000000	1.000000	4.000000	1.000000

Εικόνα 41: Σύνολο δεδομένων μετά την μετατροπή σε αριθμητικές στήλες

Παρακάτω, παρέχεται μια υπενθύμιση σχετικά με τη μεταβλητή στόχου στο σύνολο δεδομένων. Η κατανομή αυτής της μεταβλητής έχει υποβληθεί σε λεπτομερή ανάλυση, η οποία περιλαμβάνεται στο κεφάλαιο 6. Η μεταβλητή "booking status" μπορεί να λάβει μόνο δύο τιμές:

Τιμή 0: Η κράτηση έχει ακυρωθεί.
Τιμή 1: Η κράτηση πραγματοποιήθηκε.



Εικόνα 42: Κατανομή εξαρτημένης μεταβλητής

8.3 Διαχωρισμός συνόλου δεδομένων

Ένα κρίσιμο βήμα είναι ο διαχωρισμός του συνόλου δεδομένων. Αρχικά, πραγματοποιείται ο διαχωρισμός των χαρακτηριστικών και της μεταβλητής στόχου. Η μεταβλητή στόχου αντιστοιχεί στην τελευταία στήλη και δεν πρέπει να συμπεριλαμβάνεται ως χαρακτηριστικό. Εκτός από την στήλη που περιέχει την μεταβλητή στόχου, υπάρχουν και άλλες στήλες προς αφαίρεση, όπως η πρώτη (booking_id), η οποία περιέχει ένα μοναδικό αριθμό αναγνώρισης και δεν παρέχει κάποια προβλεπτική αξία στο μοντέλο. Συνοψίζοντας, αφαιρούνται η πρώτη και η τελευταία στήλη από τα χαρακτηριστικά (X), ενώ η μεταβλητή στόχου (y) καθορίζεται ως η τελευταία στήλη, δηλαδή η booking_status.

```
X, y = orig_data.iloc[:,1:-1], orig_data.iloc[:,-1]
```

Να αναφερθεί ότι το σύνολο δεδομένων χωρίστηκε σε training και test data με ποσοστά 67% και 33% αντίστοιχα, ώστε το μοντέλο να είναι αμερόληπτο. Το μοντέλο μαθαίνει-εκπαιδεύεται από τα train data και στη συνέχεια αξιολογείται από τα test data.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.33, random_state=123)
```

Η τιμή της random_state ελέγχει τον τρόπο που γίνεται ο τυχαίος διαχωρισμός, και η ίδια τιμή θα παράγει πάντα τον ίδιο διαχωρισμό, βοηθώντας στην αναπαραγωγή των αποτελεσμάτων.

Όποτε έχουν προκύψει πλέον 2 ξεχωριστά σύνολα, το σύνολο «X» που εμπεριέχει τα χαρακτηριστικά και το σύνολο «y» που εμπεριέχει μονο την μεταβλητή στόχου.

	no_of_adults	no_of_children	no_of_weekend_nights	no_of_week_nights	booking_duration	type_of_meal_plan	required_car_parking_space	room_type_reserv
0	2	0	1	2	3	0	0	
1	2	0	2	3	5	3	0	
2	1	0	2	1	3	0	0	
3	2	0	0	2	2	0	0	
4	2	0	1	1	2	3	0	

5 rows × 21 columns

Εικόνα 43: Σύνολο "X" χαρακτηριστικά

```
y.head()
```

```
0    1
1    1
2    0
3    0
4    0
Name: booking_status, dtype: int32
```

Εικόνα 44: Σύνολο "y" μεταβλητή στόχου

8.3 Δημιουργία μοντέλου XGBoost

8.3.1 Συντονισμός υπερπαραμέτρων με Αναζήτηση Πλέγματος (Grid Search)

Το GridSearchCV είναι η διαδικασία που χρησιμοποιείται για τον αυτοματοποιημένο συντονισμό των υπερπαραμέτρων ενός μοντέλου, με στόχο την εύρεση των βέλτιστων τιμών για αυτές. Όπως έχουμε αναφέρει, η απόδοση ενός μοντέλου εξαρτάται σε μεγάλο βαθμό από τις υπερπαραμέτρους του. Λόγω της αδυναμίας μας να προβλέψουμε προκαταρκτικά τις ιδανικές τιμές για αυτές, το GridSearchCV εκτελεί αυτόματη αναζήτηση μέσω δοκιμής διάφορων τιμών. Αυτό αποτελεί αποτελεσματικό τρόπο για τη βελτιστοποίηση των υπερπαραμέτρων χωρίς την ανάγκη χειροκίνητης παρέμβασης και εξοικονομεί χρόνο και πόρους. Το GridSearchCV είναι μια συνάρτηση που έρχεται στο πακέτο model_selection του Scikit-learn

```
from xgboost import XGBClassifier
from sklearn.model_selection import RandomizedSearchCV, StratifiedKFold, GridSearchCV
from sklearn.metrics import accuracy_score, f1_score, classification_report, confusion_matrix
```

Είναι πολύ σημαντικο σε αυτή την φαση η έννοια της παραμέτρου και της υπερπαραμέτρου να είναι πλήρως διακριτές. Οι παράμετροι είναι εσωτερικές του μοντέλου και καθορίζονται διάρκεια της διαδικασίας εκπαίδευσης. Οι παράμετροι

είναι γνωστές ως "βάροι" ή οι "συντελεστές" που προσαρμόζονται ώστε το μοντέλο να προσαρμόζεται στα δεδομένα εκπαίδευσης και να κάνει καλή πρόβλεψη στα νέα δεδομένα .Σε αντίθεση με τις υπερπαράμετρους όπου αποτελούν εξωτερικές μεταβλητές του μοντέλου και οριζονται πάντα πριν την έναρξη της εκπαίδευσης του από τον ερευνητή ή τον χρήστη . Οι υπερπαράμετροι επηρεάζουν τη συμπεριφορά του αλγορίθμου μάθησης και την επίδοσή του μοντέλου.

Παρακάτω ακολουθεί η δημιουργία ενός ταξινομητή XGBoost και επιθεώρηση των προκαθορισμένων παραμέτρων.

```
# Instatiate a XGBClassifier
xgb_clf=xgb.XGBClassifier(random_state=123)

# Inspect the parameters
xgb_clf.get_params()
```

```
{'objective': 'binary:logistic',
 'use_label_encoder': None,
 'base_score': None,
 'booster': None,
 'callbacks': None,
 'colsample_bylevel': None,
 'colsample_bynode': None,
 'colsample_bytree': None,
 'early_stopping_rounds': None,
 'enable_categorical': False,
 'eval_metric': None,
 'feature_types': None,
 'gamma': None,
 'gpu_id': None,
 'grow_policy': None,
 'importance_type': None,
 'interaction_constraints': None,
 'learning_rate': None,
 'max_bin': None,
 'max_cat_threshold': None,
 'max_cat_to_onehot': None,
 'max_delta_step': None,
 'max_depth': None,
 'max_leaves': None,
 'min_child_weight': None,
 'missing': nan,
 'monotone_constraints': None,
 'n_estimators': 100,
 'n_jobs': None,
 'num_parallel_tree': None,
 'predictor': None,
 'random_state': 123,
 'reg_alpha': None,
 'reg_lambda': None,
 'sampling_method': None,
 'scale_pos_weight': None,
 'subsample': None,
 'tree_method': None,
 'validate_parameters': None,
 'verbosity': None}
```

Εικόνα 45: Δημιουργία XGBoost Classifier και επιθεώρηση παραμέτρων

Όπως φαίνεται στην παραπάνω εικόνα οι προκαθορισμένες τιμές παραμέτρων του μοντέλου είναι:

n_estimators: 100

random_state': 123

max_depth': None

learning_rate': None

8.3.2 Δημιουργία μοντέλου με την αναζήτηση πλέγματος

Ο υπολογισμός των βέλτιστων τιμών των υπερπαραμέτρων με την Αναζήτηση Πλέγματος εκτελείτε σε συγκεκριμένες τιμές παραμέτρων, που όπως προαναφέρθηκε δίνονται από τον χρήστη ή τον ερευνητή. Στην έρευνα μας δόθηκαν οι παρακάτω τιμές στις παραμέτρους.

```
xgb_param_grid={"n_estimators":[100,600,1000],
                "max_depth":[3,4,5],
                "learning_rate":[0.01,0.1,0.3],
                'scale_pos_weight':[0.487228987]}
```

- n_estimators: επιλέχθηκαν για δοκιμή οι τιμές 100, 600 και 1000. Αυτή η παράμετρος καθορίζει τον αριθμό των δέντρων που θα χρησιμοποιηθούν στο μοντέλο XGBoost. Ένα μεγαλύτερο νούμερο συνήθως οδηγεί σε καλύτερη απόδοση, αλλά αυξάνει τον χρόνο εκπαίδευσης.

- max_depth: επιλέχθηκαν για δοκιμή οι τιμές 3, 4 και 5. Όπως υποδηλώνει και το όνομα της, η παράμετρος max_depth καθορίζει το μέγιστο βάθος κάθε δέντρου απόφασης. Χρησιμοποιείται για τον έλεγχο over-fitting, καθώς ενα μεγαλύτερο βάθος μπορεί να οδηγήσει σε πιο πολύπλοκα μοντέλα που μπορούν να μάθουν περισσότερες λεπτομέρειες από τα δεδομένα εκπαίδευσης με κίνδυνο όμως μια κακή επίδοση του μοντέλου σε νέα δεδομένα, που δεν έχει δει κατά τη διάρκεια της εκπαίδευσης (over-fitting).

- learning_rate: Αυτή η παράμετρος καθορίζει το μέγεθος του βήματος με το οποίο το μοντέλο προσαρμόζει τις προβλέψεις του κατά την εκπαίδευση. Είναι επίσης γνωστός ως "συντελεστής συρρίκνωσης" και συνήθως ορίζεται σε μια μικρή τιμή, όπως 0,1 ή 0,01, για να διασφαλιστεί ότι το μοντέλο συγκλίνει αργά και ομαλά.

- scale_pos_weight: Αυτή η παράμετρος χρησιμοποιείται συνήθως σε περιπτώσεις ανισορροπίας μεταξύ των κατηγοριών (imbalanced classes) και καθορίζει το βάρος της θετικής κλάσης. Ως θετική κλάση ορίζουμε το πλήθος των εγγραφών που λαμβάνουν την τιμή 1 (μη ακυρωμένες κρατήσεις στην περίπτωση της παρούσας έρευνας). Προκύπτει από το λόγο sum(negative instances / sum(positive instances), δηλαδή διαιρώντας το άθροισμα των εγγραφών της αρνητικής κλάσης δια το άθροισμα των εγγραφών της θετικής. Επομένως στην συγκεκριμένη περίπτωση 11885/24390 = 0.487228987.

```
xgb = XGBClassifier(objective="binary:logistic", eval_metric="auc", random_state=123)

cv_f=StratifiedKFold(n_splits=3,shuffle=True)

rand_search = GridSearchCV(estimator=xgb, param_grid=xgb_param_grid, n_jobs=2, cv=cv_f, verbose=1, scoring='roc_auc')

rand_search.fit(X_train,y_train)

preds= rand_search.predict(X_test)

print("The accuracy score is: " , accuracy_score(y_test, preds))
print("Below, it is presented the classification report")
print(classification_report(y_test, preds))
print("An initial presentation of the confusion matrix")
print(confusion_matrix(y_test, preds))
```

Στην συνέχεια, δημιουργείτε ένα αντικείμενο XGBClassifier, δηλαδή ένας ταξινομητής XGBoost για δυαδική ταξινόμηση. Ορίζετε ο στόχος (binary:logistic) και τη μετρική αξιολόγησης auc. Δημιουργείτε επίσης, ένα αντικείμενο StratifiedKFold για Cross-Validation με 3 φακέλους (3 splits). Το StratifiedKFold, χρησιμοποιείτε σε περιπτώσεις ανισορροπίας μεταξύ των κατηγοριών (imbalanced classes) και εξασφαλίζει ότι οι αναλογίες των κλάσεων διατηρούνται σε κάθε φάκελο, βελτιώνοντας την ακρίβεια του μοντέλου.

Κατόπιν αυτού δημιουργείτε ένα αντικείμενο GridSearchCV χρησιμοποιώντας το προαναφερθέν μοντέλο, τον ορισμό υπερπαραμέτρων (param_grid=xgb_param_grid) και τον ορισμό μετρικής αξιολόγησης (scoring='roc_auc'). Το μοντελο εκπαιδευεται χρησιμοποιώντας τα δεδομένα εκπαιδευσης (X_train, y_train). Ο Grid Search θα εξερευνήσει διάφορους συνδυασμούς υπερπαραμέτρων και θα επιλέξει αυτούς που οδηγούν στην καλύτερη επίδοση.

Χρησιμοποιείται το πλέον εκπαιδευμένο μοντέλο με δεδομένα εισαγωγής τα test data (δοκιμαστικά δεδομένα) για να παράξει τις προβλέψεις .Οι προβλέψεις αποθηκεύονται στη μεταβλητή preds.

```
Fitting 3 folds for each of 27 candidates, totalling 81 fits
The accuracy score is:  0.8887311001587169
Below, it is presented the classification report
              precision    recall  f1-score   support

           0       0.81      0.86      0.83      3869
           1       0.93      0.90      0.92      8102

    accuracy                           0.89     11971
   macro avg       0.87      0.88      0.87     11971
weighted avg       0.89      0.89      0.89     11971

An initial presentation of the confusion matrix
[[3311  558]
 [ 774 7328]]
```

```
print("Best parameters found:", rand_search.best_params_)
print("Best Cross-Validated Score:", rand_search.best_score_)
```

```
Best parameters found: {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 1000, 'scale_pos_weight': 0.487228987}
Best Cross-Validated Score: 0.9503593584275101
```

Το τελικό accuracy score είναι ενθαρρυντικά υψηλό, καθώς έχει την τιμή 0.8887. Αυτό υποδεικνύει ότι το μοντέλο έχει καλή ικανότητα να κάνει σωστές προβλέψεις σε σχέση με τα συνολικά δεδομένα δοκιμής (test data). Ωστόσο, παρά το γεγονός ότι το accuracy είναι μια σημαντική μετρική, είναι πάντα καλό να εξετάζονται και άλλες μετρικές (όπως precision, recall, και F1-score) την απόκτηση μια πιο πλήρης εικόνας της απόδοσης του μοντέλου. Οι τιμές των προαναφερόμενων μετρικών βρίσκονται στο classification report της παραπάνω εικόνας.

Από τις συνδυαστικές δοκιμές προέκυψε καλύτερο αποτέλεσμα πρόβλεψης με τις παρακάτω τιμές παραμέτρων.

n_estimator = 1000

max_depth = 5

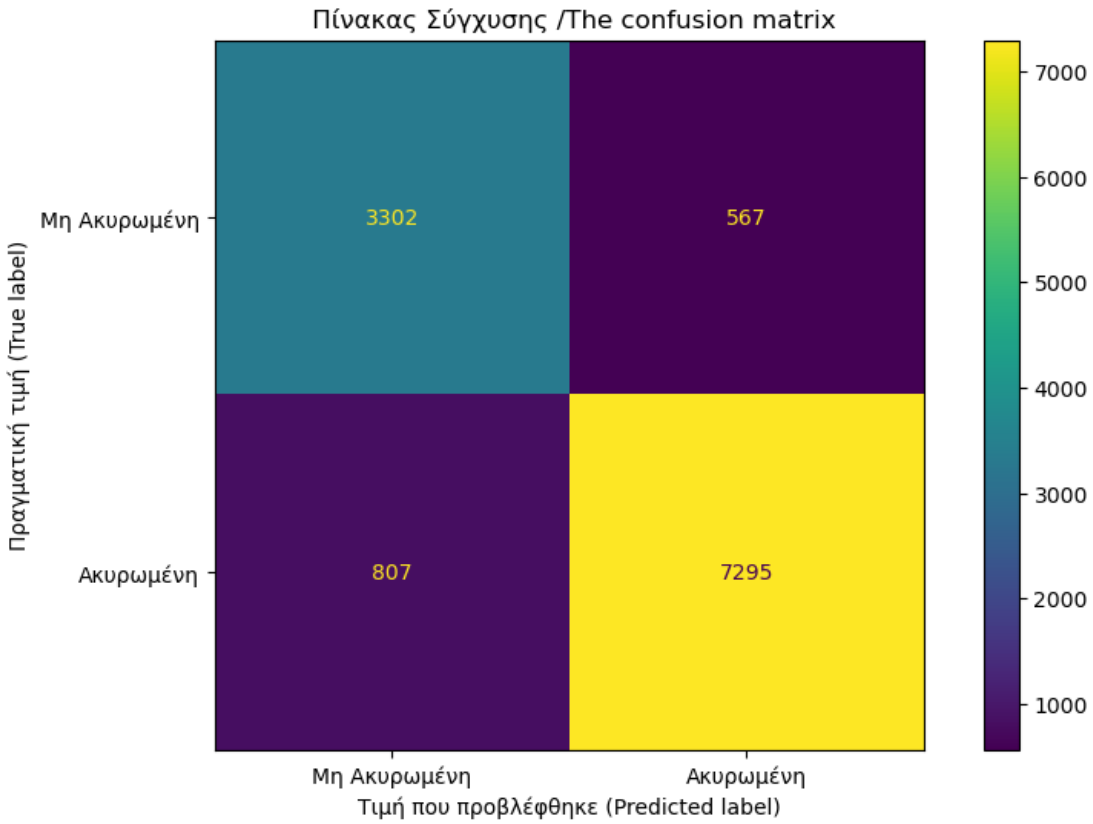
learning_rate = 0.1

scale_pos_weight = 0.487228987

8.3.3 Πίνακας Σύγχυσης Confusion Matrix

```
import matplotlib.pyplot as plt
from sklearn.datasets import make_classification
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

cm=confusion_matrix(y_test, preds, labels=rand_search.classes_)
disp=ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=['Not Canceled','Canceled'])
fig, ax = plt.subplots(figsize=(10,6))
disp.plot(ax=ax)
plt.title(" The confusion matrix")
plt.show()
```

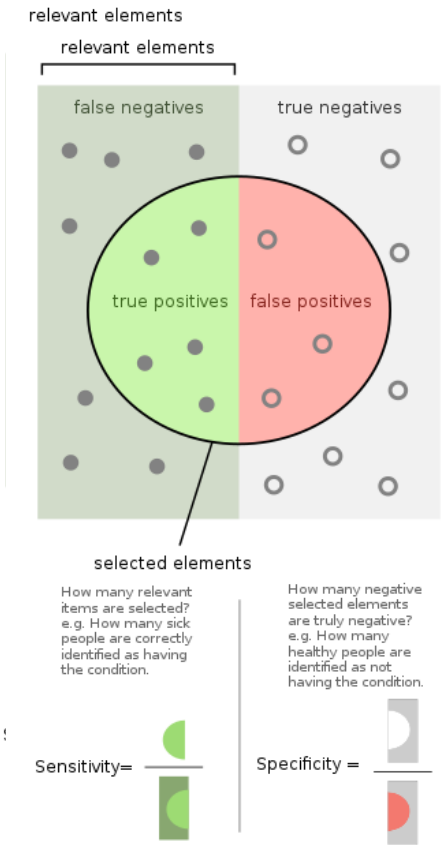


Εικόνα 46: Πίνακας Σύγχυσης (Confusion Matrix)

Μετά το διαχωρισμό των δεδομένων μου σε δεδομένα εκπαίδευσης και δεδομένα δοκι 92.254 τιμές μής, τα δεδομένα δοκιμή αποτελούν το 33% των συνολικών που μεταφράζεται σε 11970 έγγραφες. Οι 567 από αυτές ταξινομούνται λανθασμένα ως θετικές ενώ είναι αρνητικές (FP) και οι 805 από αυτές ταξινομούνται ως αρνητικές ενώ είναι θετικές (FN).

8.3.4 Καμπύλη ROC (Receiver Operating Characteristic)

Η καμπύλη ROC είναι η γραφική παράσταση του αληθώς θετικού ρυθμού (True Positive Rate TPR) έναντι του ψευδώς θετικού ρυθμού (False Positive Rate FPR) σε κάθε ρύθμιση κατωφλίου. Ο αληθώς θετικός ρυθμός (TPR) είναι επίσης γνωστός ως ευαισθησία (sensitivity), ενώ ψευδώς θετικός ρυθμός (FPR) λέγεται επίσης ειδικότητα (specificity).



Εικόνα 47: Ευαισθησία και Ειδικότητα

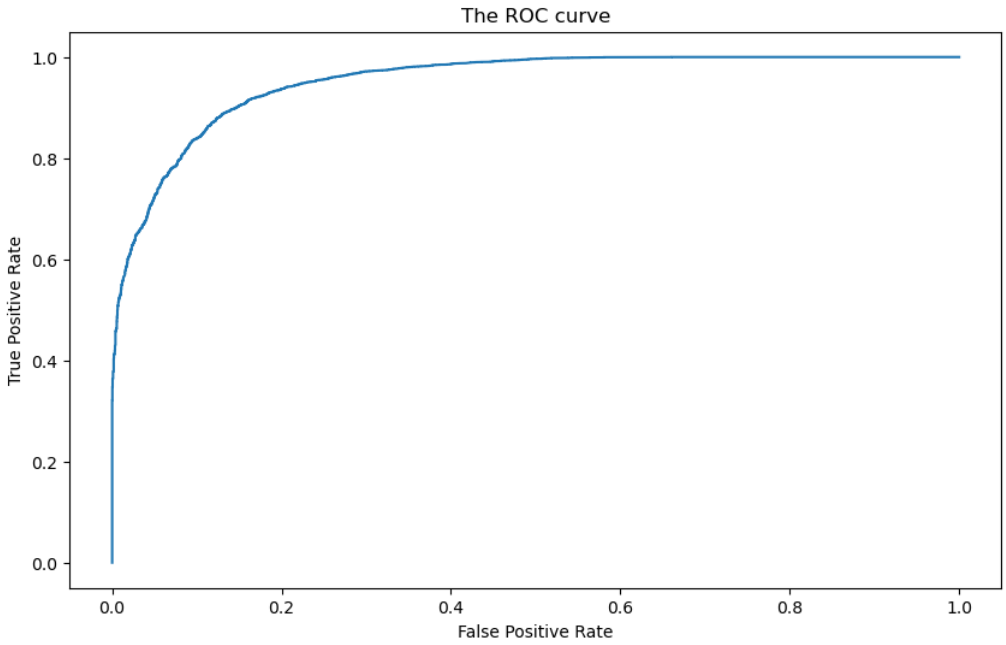
("File:Sensitivity and specificity.svg - Wikimedia Commons", n.d.)

Ο άξονας x αναπαριστά τον ψευδώς θετικό ρυθμό (specificity) και ο άξονας y τον αληθώς θετικό ρυθμό (sensitivity). Η επάνω αριστερή γωνία υποδηλώνει μηδενικό ρυθμό των ψευδών θετικών και μονάδα στον αληθώς θετικό ρυθμό, με αποτέλεσμα να θεωρείται το βέλτιστο σημείο.

Η δημιουργία και ο σχεδιασμός της καμπύλής περιγράφονται στον παρακάτω κώδικα:

```
from sklearn import metrics
preds2 = rand_search.predict_proba(X_test)[::,1]
fpr,tpr, _ = metrics.roc_curve(y_test, preds2)
```

```
plt.subplots(figsize=(10,6))
plt.plot(fpr,tpr)
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.title("The ROC curve")
plt.show()
```



Εικόνα 48: Καμπύλη ROC

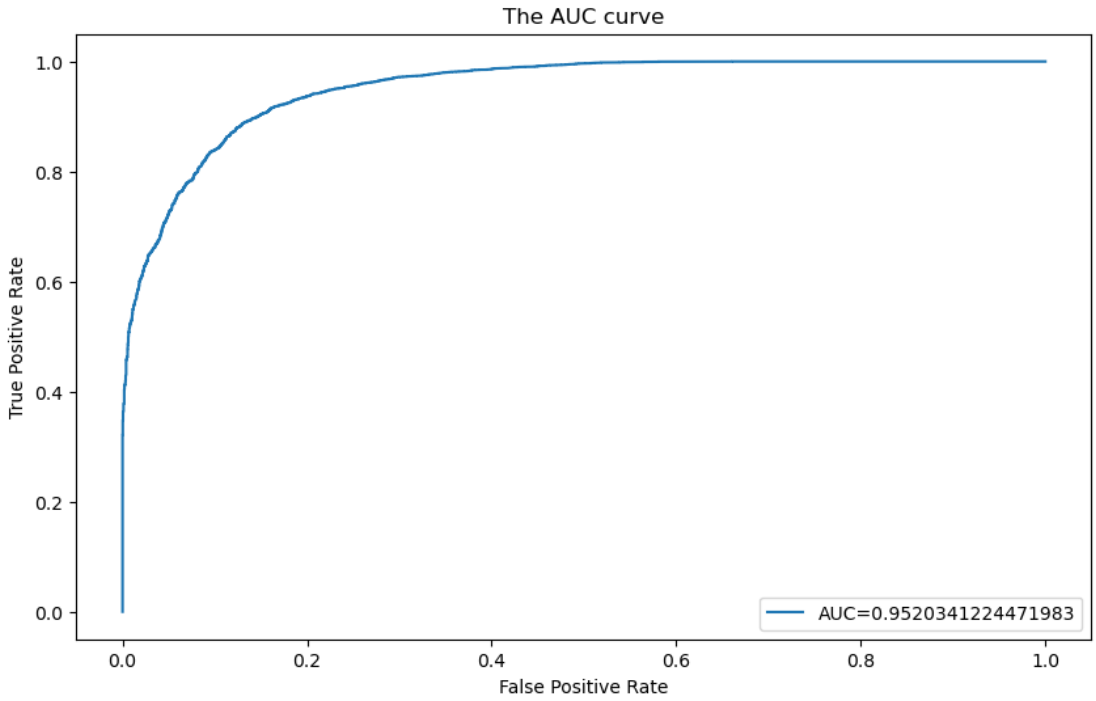
8.3.5 Καμπύλη AUC (Area Under the Curve)

Η καμπύλη AUC (Area Under the Curve) αναφέρεται στο εμβαδό κάτω από την καμπύλη ROC (Receiver Operating Characteristic). Μια τιμή AUC ίση με 1 υποδεικνύει ένα ιδανικό μοντέλο που έχει τέλεια διάκριση μεταξύ των κλάσεων.

Η δημιουργία και ο σχεδιασμός της καμπύλής περιγράφονται στον παρακάτω κώδικα:

```
auc = metrics.roc_auc_score(y_test, preds2)
```

```
plt.subplots(figsize=(10,6))
plt.plot(fpr,tpr,label="AUC="+str(auc))
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.legend(loc=4)
plt.title("The AUC curve")
plt.show()
```



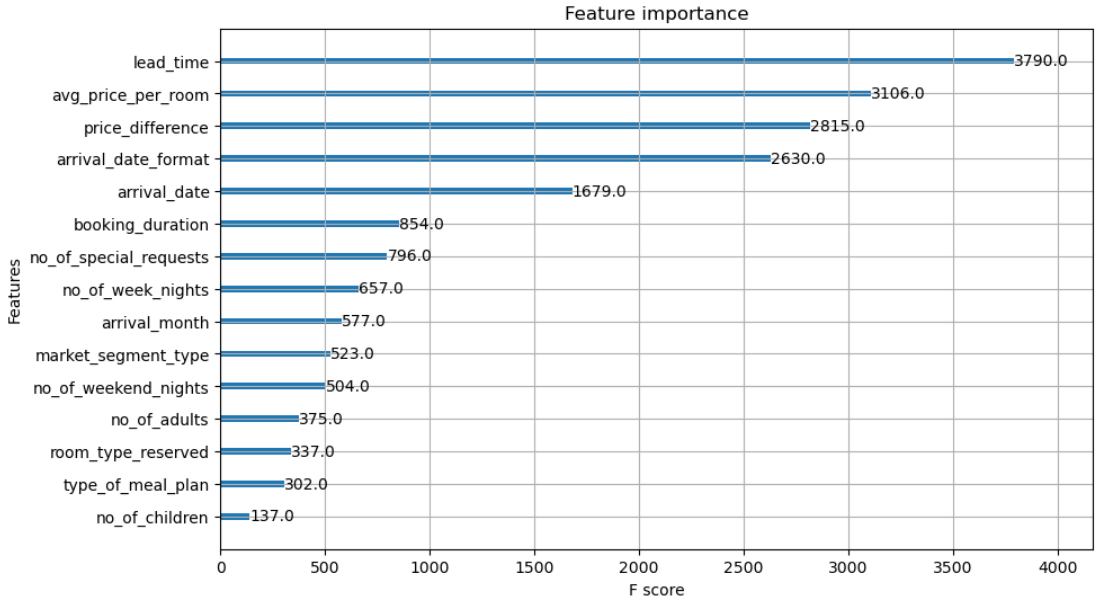
Εικόνα 49: Καμπύλη AUC

8.3.6 Διάγραμμα σπουδαιότητας βάσει βαρύτητας (weight)

Το διάγραμμα σπουδαιότητας (importance plot περιγράφει τη συνεισφορά των χαρακτηριστικών στις αποφάσεις του μοντέλου. Συγκεκριμένα, στα μοντέλα μηχανής μάθησης, όπως τα δέντρα αποφάσεων ή οι μέθοδοι όπως το Gradient Boosting, τα χαρακτηριστικά έχουν συντελεστές που δείχνουν πόσο σημαντικά είναι κατά την λήψη αποφάσεων. Ένας από αυτούς τους συντελεστές αποτελεί και το βάρος (weight) οπου αναπαριστά τον αριθμός των φορών που εμφανίζεται ένα χαρακτηριστικό σε ένα δέντρο απόφασης.

```
from xgboost import plot_importance
xgb_cl = rand_search.best_estimator_
fig, ax = plt.subplots(figsize=(10, 6))
plot_importance(xgb_cl, importance_type='weight', max_num_features=15, ax=ax)
plt.show()
```

Για τη δημιουργία του διαγράμματος σπουδαιότητας των χαρακτηριστικών, χρειάστηκε να επαναληφθεί η εκπαίδευση του μοντέλου. Αυτό έγινε επειδή η αναζήτηση πλέγματος (Grid Search) είχε δημιουργήσει μια δομή δεδομένων που δεν μπορούσε να προσπελαστεί από τη συνάρτηση plot_importance. Κατά συνέπεια, πραγματοποιήθηκε μια σύντομη εκπαίδευση του μοντέλου για τη δημιουργία του διαγράμματος σημασίας των χαρακτηριστικών.

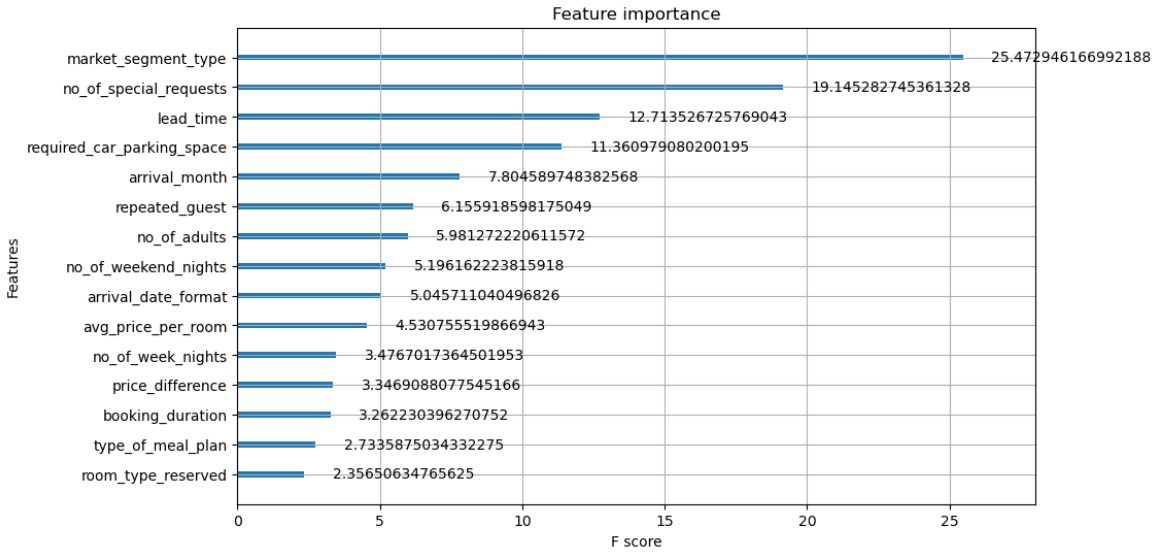


Εικόνα 50: Διάγραμμα σπουδαιότητας βάσει βαρύτητας (weight)

8.3.7 Διάγραμμα σπουδαιότητας βάσει κέρδους (gain)

Το κέδρος είναι ακόμη ένας αξιοσημείωτος συντελεστής των χαρακτηριστικών που φανερώνει πόσο σημαντικά είναι κατά την λήψη αποφάσεων. Το διάγραμμα σπουδαιότητας βάσει κέρδους (gain) είναι ένα γράφημα που απεικονίζει το κέρδος που προσφέρει κάθε χαρακτηριστικό κατά τη διαχωριστική διαδικασία σε ένα δέντρο απόφασης.

```
xgb_cl = rand_search.best_estimator_
fig, ax = plt.subplots(figsize=(10, 6))
plot_importance(xgb_cl, importance_type='gain', max_num_features=15, ax=ax)
plt.show()
```



Εικόνα 51: Διάγραμμα σπουδαιότητας βάσει κέρδους (gain)

8.4 Ερμηνευσιμότητα Μηχανικής Μάθησης

Παρά το γεγονός ότι τα προηγούμενα διαγράμματα σημαντικότητας παρείχαν μια πρώτη εικόνα του πώς τα χαρακτηριστικά επηρέασαν το τελικό αποτέλεσμα, θα επεκτείνουμε την ανάλυσή μας χρησιμοποιώντας τη βιβλιοθήκη SHAP. Στόχος μας είναι η ενίσχυση της ερμηνείας και της κατανόησης του μοντέλου μέσω αυτής της προηγμένης τεχνικής.

Αρχικά η βιβιοθήκη SHAP πρέπει να εγκατασταθεί και να ενσωματωθεί στο περιβάλλον εργασιας. Για να υπολογίσουμε τις τιμές SHAP για το μοντέλο, πρέπει να δημιουργήσουμε ένα αντικείμενο Explainer και να το χρησιμοποιήσουμε για να αξιολογήσουμε ένα δείγμα ή το πλήρες σύνολο δεδομένων:

```
pip install shap
```

```
import shap
```

```
# compute SHAP values
explainer = shap.Explainer(rand_search.best_estimator_, X_train)
shap_values = explainer(X_train)
```

```
100%|==================| 24289/24304 [05:17<00:00]
```

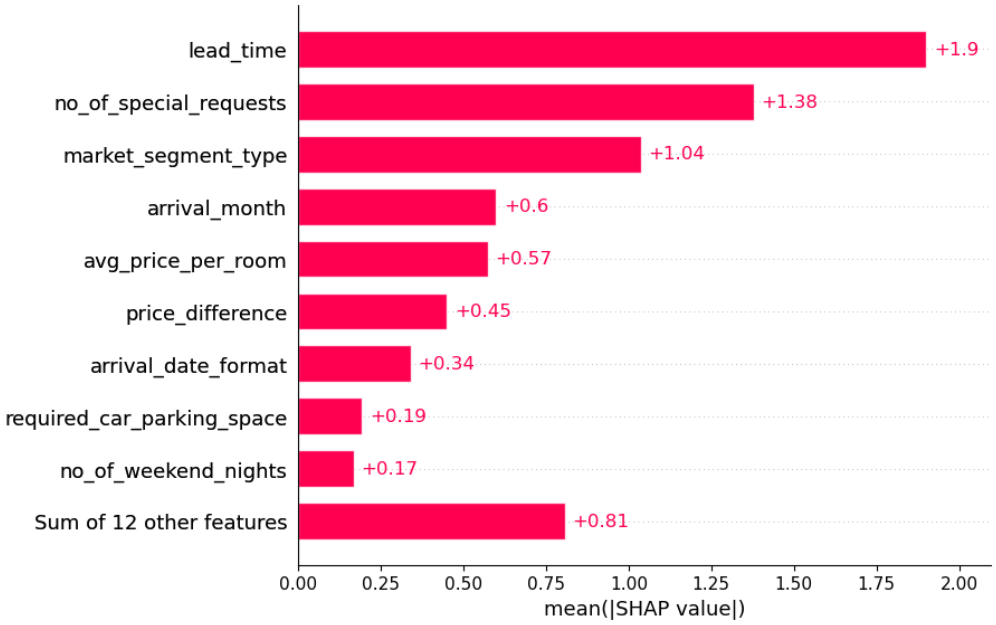
Η μεταβλητή shap_values έχει τρία χαρακτηριστικά:

- shap_values.values, οι οποίες είναι οι τιμές SHAP για κάθε παράδειγμα
- shap_values.base_values, οι οπόιες αναπαριστούν την αναμενόμενη τιμή του στόχου ή την μέση τιμή στόχου όλων των δεδομένων
- shap_values.data,όπου είναι απλώς ένα αντίγραφο των δεδομένων εισόδου

στην παρούσα μελέτη θα εστιάσουμε μόνο στις τιμές SHAP (.values), οπότε χρησιμοποείται η μέθοδος explainer.shap_values().

Το διάγραμμα που ακολουθεί τα χαρακτηριστικά ταξινομούνται από την υψηλότερη προς τη χαμηλότερη επίδραση στην πρόβλεψη. Λαμβάνει υπόψη την απόλυτη τιμή SHAP, επομένως δεν έχει σημασία αν το χαρακτηριστικό επηρεάζει την πρόβλεψη με θετικό ή αρνητικό τρόπο.

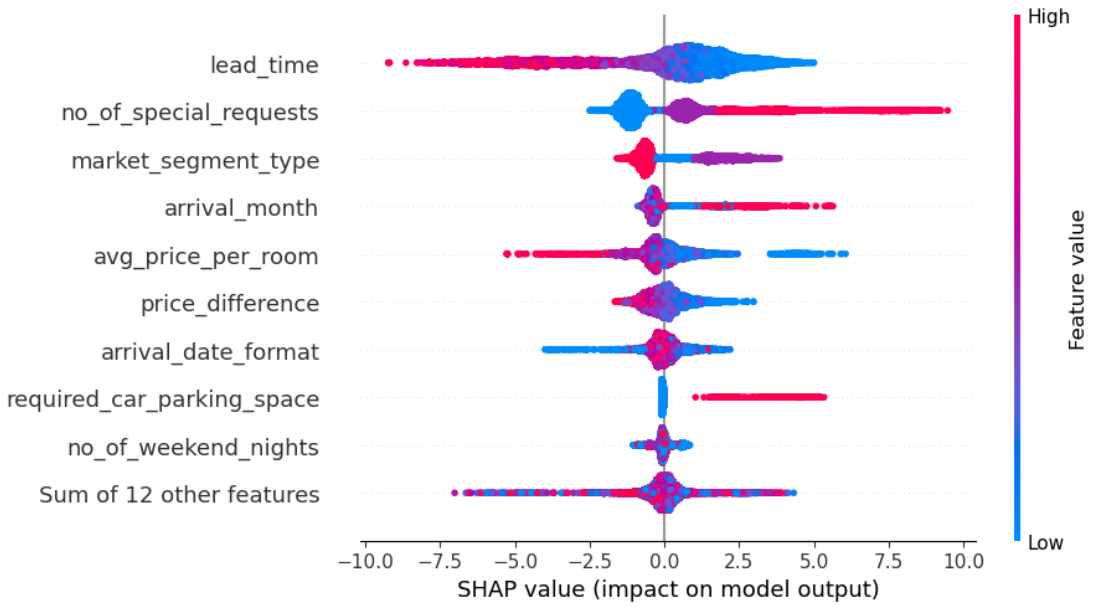
```
shap.plots.bar (shap_values)
```



Εικόνα 52: Διάγραμμα απόλυτης τιμής SHAP

Η απόσταση μεταξύ ημερομηνίας κράτησης και ημερομηνία άφιξης, ο αριθμός ειδικών αιτημάτων και ο τύπος του τμήματος της αγοράς έπαιξαν καθοριστικό ρόλο στην διεξαγωγή συμπερασμάτων του μοντέλου. Αυτή η πληροφορία ταξινόμησης παρέχεται και στο παρακάτω διάγραμμα beeswarm, αλλά με την δυνατότητα πλέον να ερμηνεύσουμε πως επηρεάζουν το αποτέλεσμα οι υψηλότερες ή χαμηλότερες τιμές του χαρακτηριστικού.

```
shap.plots.beeswarm(shap_values)
```

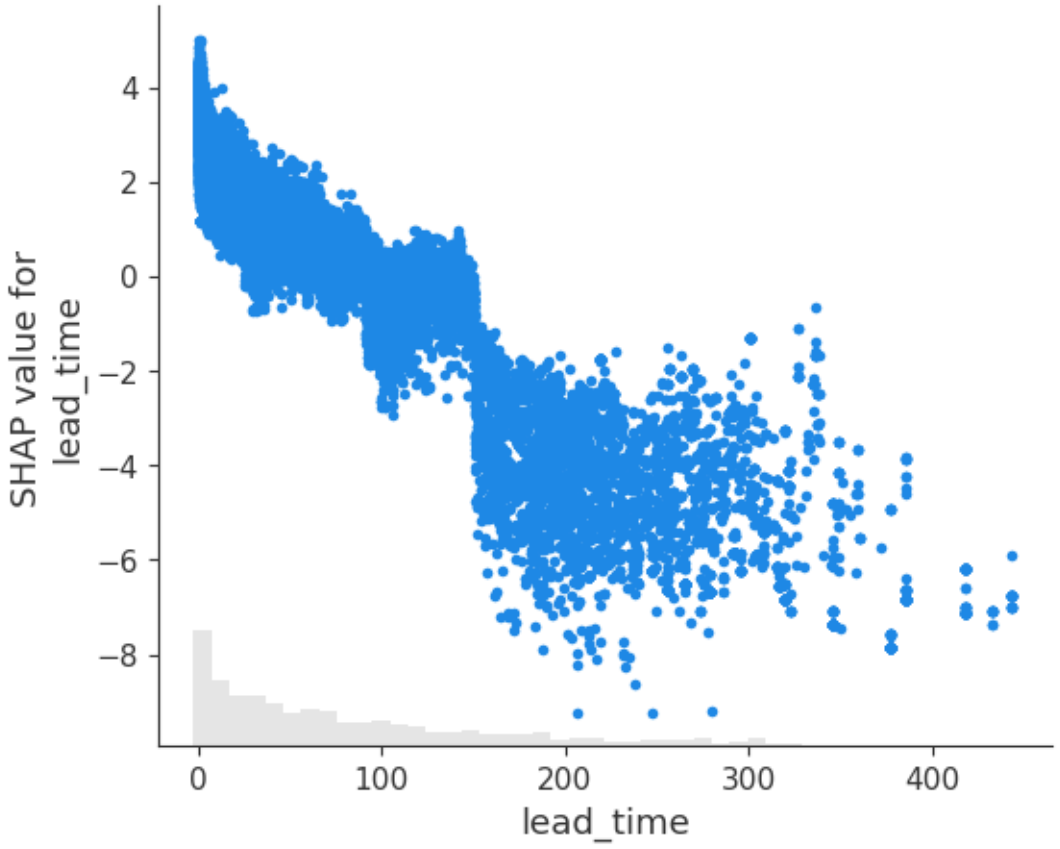


Εικόνα 53: Διάγραμμα Beeswarm

Όλες οι μικρές κουκκίδες στην πλοκή αντιπροσωπεύουν μια ενιαία παρατήρηση. Στον οριζόντιο άξονα βρίσκεται η τιμή SHAP, ενώ το χρώμα του σημείου μας δείχνει εάν η τιμή αυτής της παρατήρησης είναι υψηλότερη ή χαμηλότερη σε σχέση με άλλες παρατηρήσεις. Όταν το shap value κυμαίνεται στον θετικό άξονα, υποδεικνύει οτι το χαρακτηριστικό συμβάλει θετικά στην πρόβλεψη της θετικής κλάσης, δηλαδή όταν η κράτηση δεν καταλήγει σε ακύρωση. Αντίθετα αρνητικές τιμές του shap value υποδεικνύουν ότι το χαρακτηριστικό συμβάλει σε προβλέψεις που ανήκουν στην αρνητική κλάση, δηλαδή κρατήσεις που καταλήγουν σε ακύρωση.

Παρατηρείται λοιπόν, ότι υψηλές τιμές lead time και μέσης τιμής δωματίου έχουν αρνητικό αντίκτυπο στην πρόβλεψη, εφόσον το SHAP value είναι στον αρνητικό άξονα, ενώ οι χαμηλότερες τιμές έχουν θετικό αντίκτυπο. Αντίθετα για μεγάλο αριθμό ειδικών αιτημάτων, το χαρακτηριστικό συμβάλει θετικά στην τελική πρόβλεψη, ενώ όσο μειώνεται ο αριθμός αιτημάτων η συμβολή του γίνεται αρνητική.

Το διάγραμμα διασποράς του lead time ευθυγραμμίζεται με το αποτέλεσμα του διαγράμματος beeswarm, καθώς φαίνεται καθαρά πως όσο μεγαλώνει το lead value, μειώνεται το shap value αντιστοιχά.

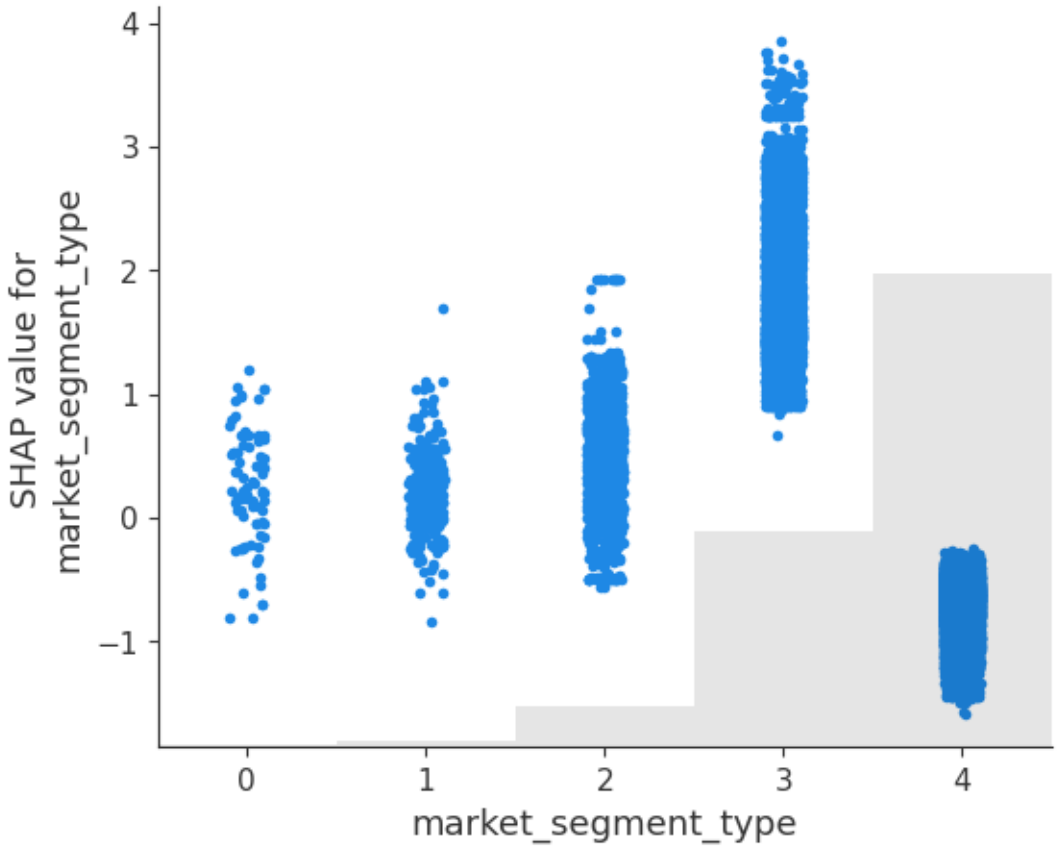


Εικόνα 54: Διάγραμμα διασποράς του Lead time

Από την ανάλυση της κατανομής του χαρακτηριστικού "market_type_segment" (τύπος τμήματος της αγοράς) στο διάγραμμα Beeswarm, προκύπτει ότι για υψηλές τιμές του "market_type_segment," η συμβολή του χαρακτηριστικού στην τελική

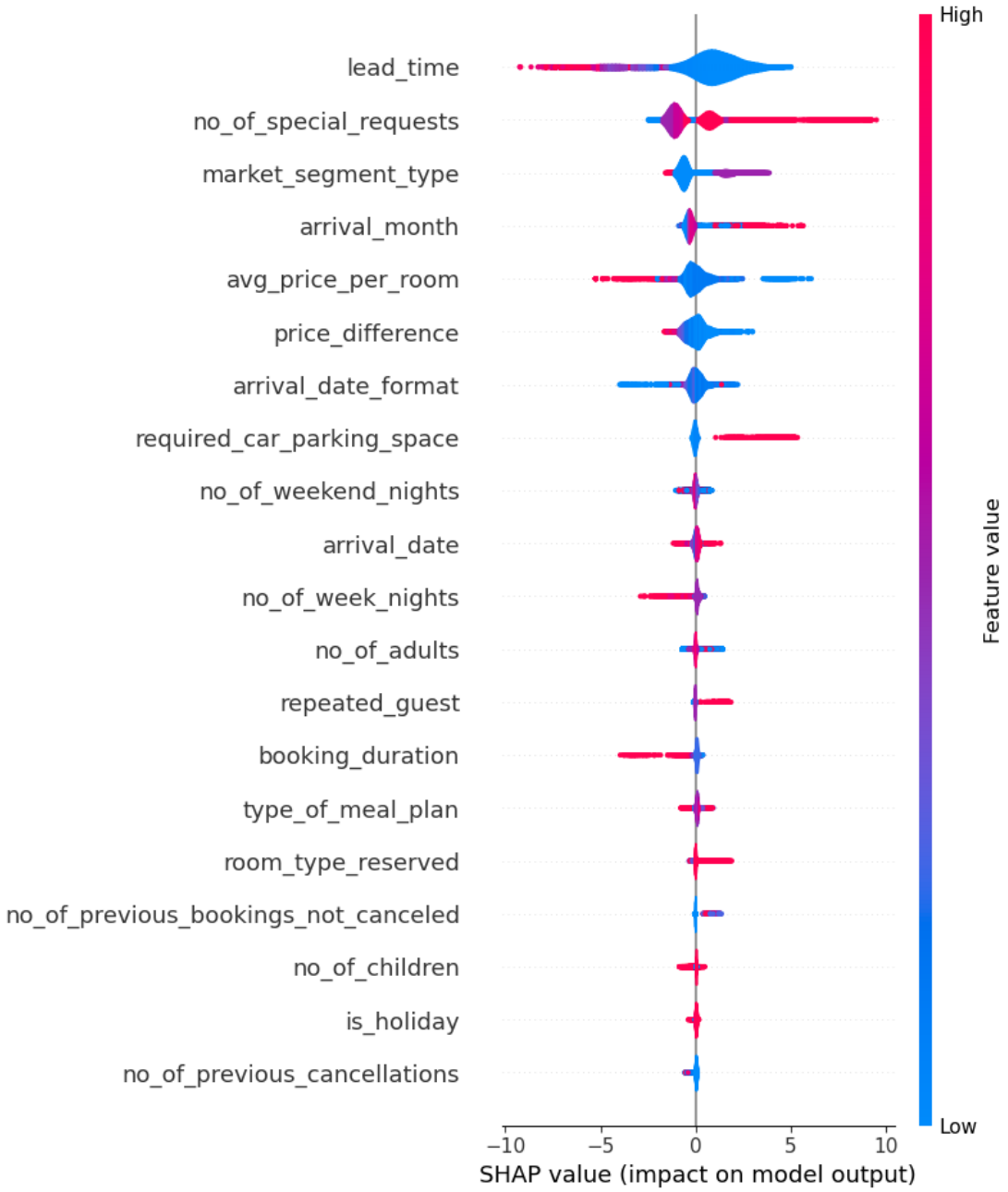
πρόβλεψη είναι χαμηλή. Για μικρές τιμές του "market_type_segment," η συμβολή αυξάνεται ελάχιστα, ενώ για τις ενδιάμεσες τιμές, παρατηρείται η περισσότερη θετική συνεισφορά στο μοντέλο.

Ένα διάγραμμα διασποράς μπορεί να βοηθήσει στην κατανόηση αυτής της συμβολής. Η μεταβλητή "market_type_segment" είναι κατηγορική και μπορεί να λάβει 5 τιμές. Όπως φαίνεται, όταν το "market_type_segment" λαμβάνει την τιμή 3, που υποδηλώνει ότι η κράτηση είναι συμπληρωματική (Complementary), έχει το μεγαλύτερο SHAP value.



Εικόνα 55: Διάγραμμα Διασποράς του τύπου τμήματος της αγοράς

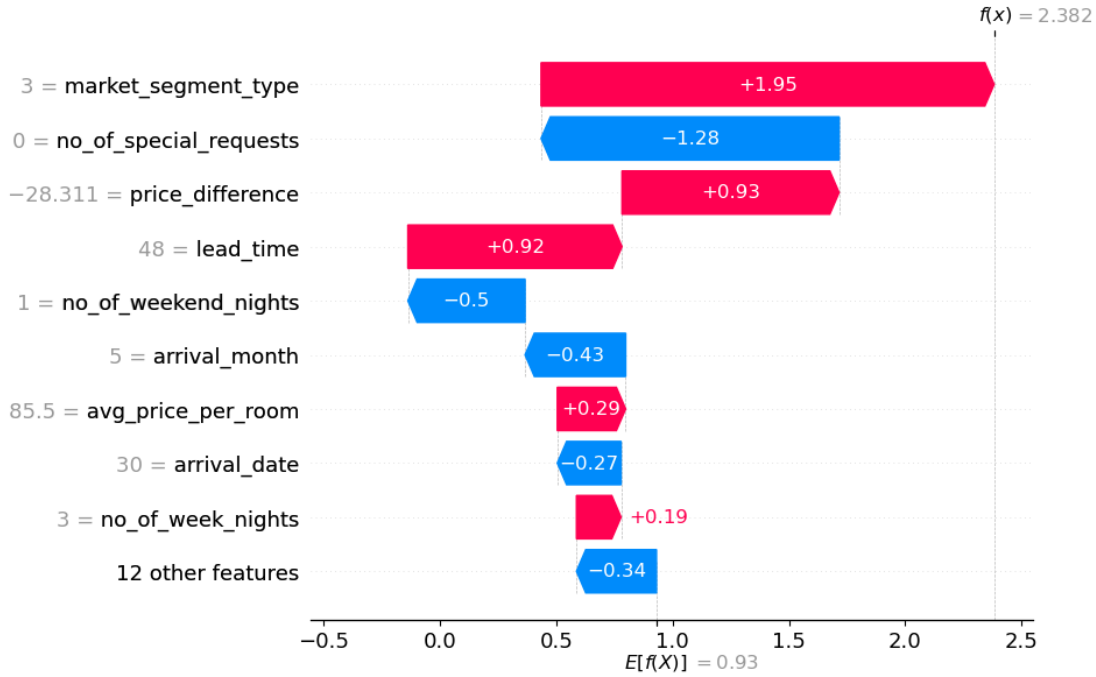
Στην συνέχεια παρατίθεται ένα διάγραμμα βιολί (Violin diagram) που παρέχει τις ίδιες πληροφορίες με το διάγραμμα Beeswarm, αλλά για όλο το εύρος των χαρακτηριστικών.



Εικόνα 56: Διάγραμμα Βιολί

Για την προσέγγισης τοπικών επιπτώσεων μίας μεμονωμένης παρατήρησης, χρησιμοποιούμε shap_values[0]. Το shap_values[0] αντιπροσωπεύει τα SHAP values για την πρώτη παρατήρηση στο σύνολο δεδομένων. Αυτό επιτρέπει την κατανόηση των συνεισφορών κάθε χαρακτηριστικού στην πρόβλεψη για αυτήν τη συγκεκριμένη παρατήρηση. Με το να επιλέγουμε ένα συγκεκριμένο παράδειγμα, μπορούμε να εστιάσουμε στην τοπική συμπεριφορά του μοντέλου γύρω από αυτήν την παρατήρηση, αναδεικνύοντας τις σημαντικές επιρροές των χαρακτηριστικών γι' αυτήν την συγκεκριμένη περίπτωση.

```
shap.plots.waterfall(shap_values[0])
```



Εικόνα 57: Διάγραμμα καταρράκτης της παρατήρησης shap_value[0]

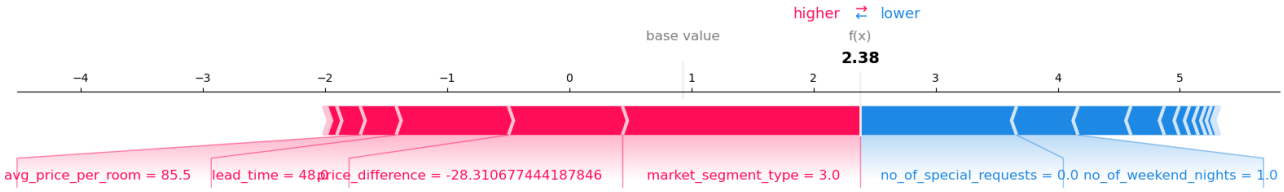
Το άθροισμα όλων των τιμών SHAP ισούται με τη διαφορά μεταξύ της πρόβλεψης f(x) =2,382 και της αναμενόμενης τιμής E[f(x)] = 0.93.

Η λεπτή γραμμή στη μέση υποδηλώνει τη μέση πρόβλεψη για το σύνολο των παρατηρήσεων E[f(x)] = 0.93. Ο κατακόρυφος άξονας στα αριστερά δείχνει τις τιμές χαρακτηριστικών της 1ης παρατήρησης και η λεπτή γραμμή στα δεξιά είναι η προβλεπόμενη τιμή για την συγκεκριμένη παρατήρηση. Οι ράβδοι αντιπροσωπεύουν πώς κάθε ιδιότητα χαρακτηριστικού μετατόπισε την τιμή από τη μέση πρόβλεψη. Οι κόκκινες γραμμές αντιπροσωπεύουν θετικές μετατοπίσεις. οι μπλε γραμμές αντιπροσωπεύουν αρνητικές μετατοπίσεις.

Ένας εναλλακτικός τρόπος να περιγραφεί η επίδραση κάθε χαρακτηριστικού στην πρόβλεψη, για μια μεμονωμένη παρατήρηση είναι το διάγραμμα δύναμης (force plot). Σε αυτό το διάγραμμα οι θετικές τιμές SHAP παρατάσσονται στην αριστερή πλευρά και οι αρνητικές στη δεξιά πλευρά, σαν να ανταγωνίζονται μεταξύ τους. Για να εκτελεστεί το διάγραμμα δύναμης, είναι απαραίτητο να γίνει φόρτωση της απαιτούμενης JavaScript βιβλιοθήκης με την εντολή shap.initjs().

```
import matplotlib
import shap

shap.initjs()
shap.force_plot(shap_values[0], matplotlib=True)
```



Εικόνα 58: Διάγραμμα δύναμης της παρατήρησης shap_value[0].

9. Συμπεράσματα

Η ραγδαία αύξηση των διαδικτυακών πλατφόρμων κρατήσεων σε συνδυασμό με τον άκρως επιχειρηματικό ανταγωνιστικό τομέα έχει δημιουργήσει την ανάγκη στις ξενοδοχειακές μονάδες και γενικότερα στις επιχειρήσεις τουρισμού που προσφέρουν διαμονή, να στραφούν στις μεθόδους μηχανικής μάθησης. Αυτή η κίνηση ωφελεί τον σκοπό τους για καλύτερη κατανόηση των προτιμήσεων των πελατών πάνω στις υπηρεσίες που προσφέρει το εκάστοτε κατάλυμα και διαφάνεια ως προς τους λόγους πιθανής ακύρωσης. Κατά συνέπεια τα αποτελέσματα που παρέχει το μοντέλο μηχανικής μάθησης μπορούν να επιφέρουν βελτίωση στις παρεχόμενες υπηρεσίες, αποφυγή απώλειας κερδών και οργάνωση ενός πλάνου στρατηγικής που να εστιάζει στην πιο επικερδή λήψη αποφάσεων.

Το πρόβλημα που διαπραγματεύεται η παρούσα εργασία ήταν η τελική κατάσταση μίας κράτησης: ακυρωμένη ή όχι. Το εξης πρόβλημα θεωρείται πρόβλημα δυαδικής ταξινόμησης (binary classification problem), καθώς στόχος του είναι να ταξινομήσουμε ένα σύνολο δεδομένων σε δύο κατηγορίες ή κλάσεις. Συγκεκριμένα, πρέπει να αποφασίσουμε σε ποια από τις δύο κατηγορίες ανήκει ένα νέο δείγμα δεδομένων, με βάση τα χαρακτηριστικά του.

Για την διεκπεραίωση της μελέτης χρησιμοποιήθηκε ο αλγόριθμος XGBoost πάνω σε ένα σύνολο ξενοδοχειακών δεδομένων. Όπως είναι γνωστό οι αλγόριθμοι μηχανικής μάθησης λειτουργούν ως μαύρα κουτιά, καθώς δεν παρέχουν διαφάνεια των αποτελεσμάτων τους. Τα διαγράμματα σπουδαιότητας, βοήθησαν στην ερμηνεία του μοντέλου δίνοντας μια πρώτη εικόνα των παραμέτρων που συνέβαλαν περισσότερο στο τελικό αποτέλεσμα. Ωστόσο, η κυριότερη συμβολή στην ερμηνεία του μοντέλου προήλθε από τη βιβλιοθήκη SHAP και τα παρεχόμενα διαγράμματα. Αυτή η βιβλιοθήκη διευκολύνει την κατανόηση και εξήγηση του πώς τα χαρακτηριστικά συνεισφέρουν στις προβλέψεις του μοντέλου.

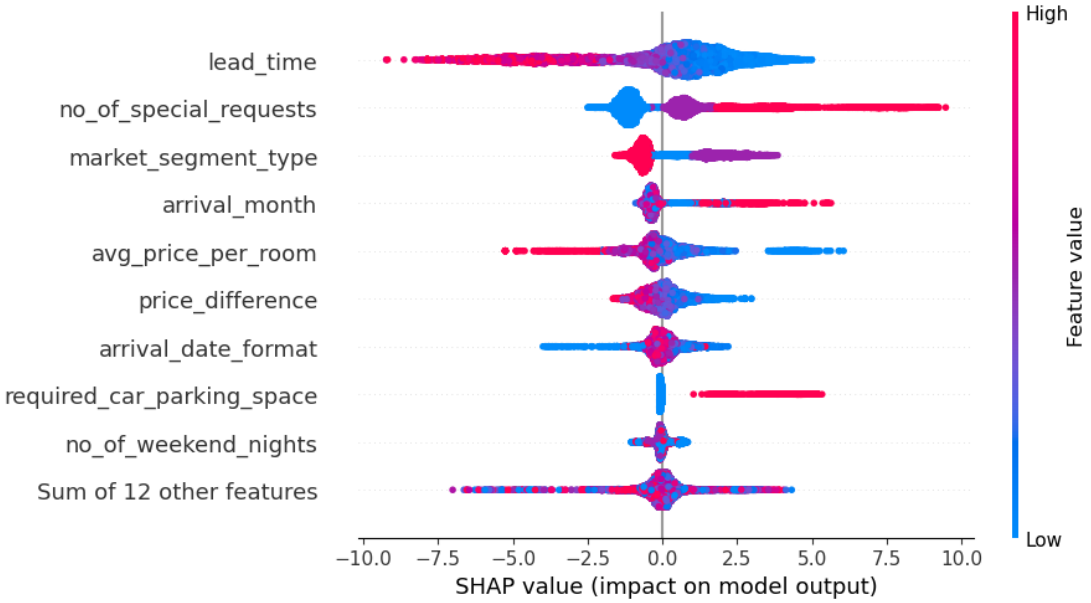
Αυτή η διευκρίνιση και η βαθύτερη κατανόηση των παραγόντων που επηρεάζουν τις προβλέψεις ενισχύουν την ικανότητά ερμηνείας των αποτελεσμάτων του μοντέλου και την διεξαγωγή συμπερασμάτων.

Εξετάζοντας τα διαγράμματα σπουδαιότητας χαρακτηριστικών με βάση την βαρύτητα, δηλαδή με βάση πόσες φορές εμφανίζεται ένα χαρακτηριστικό σε ένα δέντρο απόφασης του XGBoost, είναι ξεκάθαρο ότι αυτά που ξεχωρίζουν είναι η απόσταση από την μέρα κράτησης έως την ημέρα άφιξης, η μέση τιμή του δωματίου και η διαφορά της μέσης τιμής του δωματίου με την μέση τιμή των δωματίων τον μήνα που θα πραγματοποιηθεί η κράτηση. Επιπρόσθετα, μέσα στα 15 χαρακτηριστικά που εμφανίζονται ποιο πολλές φορές στα δέντρα απόφασης εμφανίζονται και 3 μεταβλητές που δεν υπήρχαν εξαρχής στο σύνολο δεδομένων, αλλα προστέθηκαν στην φάση του data engineering. Αυτές είναι:

- price_difference: η διαφορά της μέσης τιμής του δωματίου με την μέση τιμή των δωματίων τον μήνα που θα πραγματοποιηθεί η κράτηση. Μάλιστα είναι το 3^{ο} πιο συχνά επαναλαμβανόμενο χαρακτηριστικό στα δέντρα αποφάσεων,
- το arrival_date_format: ημερομηνία κράτησης ημέρα/μηνάς/έτος και
- το booking_duration: συνολικές ημέρες που διαρκεί η κράτηση.

Από την άλλη πλευρά, το διαγράμματα σπουδαιότητας χαρακτηριστικών με βάση το κέρδος φανερώνει πόσο σημαντικό είναι το χαρακτηριστικό κατά την λήψη αποφάσεων. Τα χαρακτηριστικά market_segments_type, no_of_special_requests, lead time και required_parcing_space φέρνουν τη μεγαλύτερη βελτίωση στην ακρίβεια των αποτελεσμάτων. Όπως είναι εμφανές τα 3 χαρακτηριστικά που προαναφέρθηκαν και δεν άνηκαν εξαρχής στο σύνολο δεδομένων: price_difference, arrival_date_format, booking_duration ανήκουν παράλληλα και στα 15 χαρακτηριστικά που παίζουν το σημαντικότερο ρόλο στην βελτίωση του μοντέλου.

Εξετάζοντας το διάγραμμα Beeswarm της βιβλιοθήκης SHAP, είναι αντιληπτά όχι μόνο η κατάταξη των χαρακτηριστικών με βάση την συνεισφορά τους αλλά και την εναλλαγή αυτής της συνεισφοράς σε θετική η αρνητική ανάλογα το την τιμή του χαρακτηριστικού.



Παρατηρείται λοιπόν, ότι υψηλές τιμές lead time έχουν αρνητικό αντίκτυπο στην πρόβλεψη, εφόσον το SHAP value είναι στον αρνητικό άξονα, ενώ οι χαμηλότερες τιμές έχουν θετικό αντίκτυπο. Αντίθετα για μεγάλο αριθμό ειδικών αιτημάτων, το χαρακτηριστικό συμβάλει θετικά στην τελική πρόβλεψη, ενώ όσο μειώνεται ο αριθμός αιτημάτων η συμβολή του γίνεται αρνητική.

Η μέση τιμή δωματίου ανά ημέρα και η διαφορά μέσης τιμής δωματίου με την διαφορά μέσης τιμής των δωματίων τον μήνα που θα πραγματοποιηθεί η κράτηση, ενεργούν με παρόμοια συμπεριφορά. Φαίνεται ότι για υψηλές τιμές της μέσης τιμής δωματίου, η συνεισφορά του χαρακτηριστικού στο μοντέλο όλο και μειώνεται. Ενώ όσο πιο οικονομική είναι η μέση τιμή ενός δωματίου ανά ημέρα, η συνεισφορά αυξάνεται και το μοντέλο βελτιστοποιείται. Αντίστοιχες είναι και οι επιπτώσεις της διαφοράς μέσης τιμής δωματίου με την μέση τιμή των δωματίων τον μήνα που θα πραγματοποιηθεί η κράτηση. Για μικρή διαφορά, το SHAP value του χαρακτηριστικού αυξάνεται.

Γενικότερα, τα παραπάνω συμπεράσματα εξαρτώνται από τα συγκριμένα χαρακτηριστικά και είναι βασισμένα στις ανάγκες της μελέτης περίπτωσης της παρούσας εργασίας. Η επίτευξη αξιόπιστων αποτελεσμάτων σε ένα μοντέλο μηχανικής μάθησης είναι συνονθύλευμα πολλών παραγόντων.

Καταρχάς, τα δεδομένα αποτελούν θεμέλιο για την επιτυχή εκπαίδευση του μοντέλου. Το μέγεθος του συνόλου δεδομένων έχει ουσιαστική σημασία, καθώς η πληθώρα δεδομένων επιτρέπει πιο ακριβή και αποτελεσματική εκπαίδευση. Η έννοια των "Big Data" αφορά τη χρήση μεγάλων και πολύπλοκων συνόλων δεδομένων.

Στη συνέχεια, η ανάλυση των δεδομένων παίζει έναν κρίσιμο ρόλο. Ο καθαρισμός των δεδομένων, η ανίχνευση ακραίων τιμών, και η δημιουργία νέων χαρακτηριστικών αποτελούν σημαντικά βήματα στην ανάπτυξη ενός ακριβούς

μοντέλου. Η επαρκής προσοχή στην επεξεργασία και ανάλυση των δεδομένων είναι καθοριστική για την τελική ακρίβεια.

Ταυτόχρονα, οι υπερπαράμετροι του μοντέλου παίζουν σημαντικό ρόλο. Η χρήση τεχνικών όπως το cross-validation και η αναζήτηση πλέγματος (grid search) βοηθούν στη βελτιστοποίηση των υπερπαραμέτρων για καλύτερα αποτελέσματα.

Ωστόσο, όλα αυτά δεν είναι αρκετά χωρίς τους κατάλληλους υπολογιστικούς πόρους. Η επεξεργαστική ισχύς, η μνήμη (RAM), ο αποθηκευτικός χώρος και η πιθανή χρήση μονάδων επεξεργασίας γραφικών (GPU) συμβάλλουν στην ολοκληρωμένη εκπαίδευση και αξιολόγηση του μοντέλου.

Τέλος, παρά τους πόρους, η ανθρώπινη αφοσίωση αποτελεί τον τελικό παράγοντα. Ένα ακριβές μοντέλο μηχανικής μάθησης συνήθως δεν δημιουργείται με την πρώτη προσπάθεια. Χρειάζεται επανάληψη της διαδικασίας αλλάζοντας τις παραμέτρους, ρυθμίζοντας διαφορετικά τις υπερπαραμετρους με σκοπό εύρεση καλύτερων τιμών ,αλλά και αναλύοντας το σύνολο δεδομένων εκτενεστέρα. Η επαναληπτική διαδικασία εκπαίδευσης απαιτούν μεθοδικότητα και αφοσίωση για την επίτευξη βέλτιστης απόδοσης.

10. Παράρτημα Κώδικα Α

Data Visualization

```
pip install xgboost
import pandas as pd
import xgboost as xgb
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
bookings = pd.read_csv('Hotel Reservations.csv')
bookings.info()

cols = ['gold', 'lightcoral']
labels = ['Not Canceled',' Canceled']
bookings['booking_status'].value_counts().plot.pie(labels = labels,
autopct='%1.1f%%',shadow=True, colors=cols)

plt.title('Hotel Cancellation Percentage')
plt.show()

custom_palette = [ "#adc2eb", "#e6b3b3", "#ecc6c6", "#d98c8c"]
sns.set_palette(custom_palette)
plt.figure(figsize=(14,6))
countplot = sns.countplot( data = bookings, x = 'arrival_month', col
or = "#c2d1f0")

new_labels = ['January','February','March','April','May','June','Jul
```

```
y','August','September','October','November','December']
countplot.set_xticklabels(new_labels)

for p in countplot.patches:
    height = p.get_height()
    print(height)
    width = p.get_width()
    x, y = p.get_x(), p.get_y()
    percentage = f'{height}'
    countplot.annotate(percentage, (x + width / 2, y + height / 2 ),
ha='center', va='center', fontsize=12)

plt.title('Κρατήσεις ανα Μήνα')
plt.xlabel('Arrivals per month')
plt.show()

plt.figure(figsize=(14,6))
countplot = sns.countplot( data= bookings, x = 'arrival_month', hue
= 'booking_status')

new_labels = ['January','February','March','April','May','June','Jul
y','August','September','October','November','December']
countplot.set_xticklabels(new_labels)

for p in countplot.patches:
    height = p.get_height()
    print(height)
    width = p.get_width()
    x, y = p.get_x(), p.get_y()
    percentage = f'{height}'
    countplot.annotate(percentage, (x + width / 2, y + height / 2 ),
ha='center', va='center', fontsize=12)


plt.title('Καταμέτρηση μηνιαίας ακύρωσης')
plt.xlabel('Arrival Month')
plt.show()

ax = pd.crosstab(bookings['arrival_month'], bookings['booking_status
']).plot(kind='bar', stacked=True, color = ["#e6b3b3" , "#adc2eb"] )

new_labels = ['January','February','March','April','May','June','Jul
y','August','September','October','November','December']
ax.set_xticklabels(['January','February','March','April','May','June
','July','August','September','October','November','December'])

plt.title('Καταμέτρηση μηνιαίας ακύρωσης')
plt.xlabel('Arrival Month')
plt.show()

bookings['cancellation_fraction'] = bookings['booking_status'].eq('C
anceled').astype(int)
plt.figure(figsize=(14, 6))
barplot = sns.barplot(data=bookings, x='arrival_month', y='cancellat
```

```
ion_fraction', color = '#e6b3b3')

for p in barplot.patches:
    height = p.get_height()
    width = p.get_width()
    x, y = p.get_x(), p.get_y()
    percentage = f'{height}'
    formatted_number = "{:.3f}".format(height)
    barplot.annotate( formatted_number, (x + width / 2, y + height /
2 ), ha='center', va='center', fontsize=10)

new_labels = ['January','February','March','April','May','June','Jul
y','August','September','October','November','December']
barplot.set_xticklabels(['January','February','March','April','May',
'June','July','August','September','October','November','December'])

plt.title('Κλάσμα Ακυρωμένων Κρατήσεων ανά Μήνα')
plt.xlabel('Μήνας Άφιξης')
plt.ylabel('Κλάσμα Ακυρώσεων')
plt.xticks(rotation=0)
plt.show()

bookings['cancellation_fraction'] = bookings['booking_status'].eq('C
anceled').astype(int)
total_bookings_per_month = bookings['arrival_month'].value_counts()

# Υπολογισμός του ποσοστού ακυρώσεων ανά μήνα
bookings['cancellation_percentage'] = bookings.groupby('arrival_mont
h')['cancellation_fraction'].transform('mean') * 100

plt.figure(figsize=(14, 6))
barplot = sns.barplot(data=bookings, x='arrival_month', y='cancellat
ion_percentage', color='#e6b3b3')

for p in barplot.patches:
    height = p.get_height()
    width = p.get_width()
    x, y = p.get_x(), p.get_y()
    percentage = f'{height:.1f}'
    barplot.annotate(percentage + '%', (x + width / 2, y + height /
2), ha='center', va='center', fontsize=10)

new_labels = ['January','February','March','April','May','June','Jul
y','August','September','October','November','December']
barplot.set_xticklabels(['January','February','March','April','May',
'June','July','August','September','October','November','December'])

plt.title('Ποσοστό Ακυρώμένων Κρατήσεων ανά Μήνα')
plt.xlabel('Μήνας Άφιξης')
plt.ylabel('Ποσοστό Ακυρώσεων')
plt.xticks(rotation=0)
plt.show()

plt.figure(figsize=(12,6))
```

```
countplot = sns.countplot( x = bookings['booking_status'] , palette
= ["#adc2eb", "#e6b3b3"])

plt.title(' Πλήθος ακυρώσεων και μη ακυρώσεων ')
plt.xlabel('Booking Status')

for p in countplot.patches:
    height = p.get_height()
    width = p.get_width()
    x, y = p.get_x(), p.get_y()
    count = f'{height}'
    countplot.annotate(count, (x + width / 2, y + height / 2 ), ha='
center', va='center', fontsize=12)

plt.show()

plt.hist(bookings['lead_time'], bins=15, color = '#c2d1f0')
plt.xlabel('Lead Time')
plt.ylabel('Συχνότητα')
plt.title('Κατανομή της μεταβλητής lead_time')

plt.show()
print('Συναντάμε την μέγιστη συχνότητα της κατανομής', height, ' στο
διάστημα 0-30 ημερών μεταξυ ημερας κράτησης και άφιξης')

plt.figure(figsize=(14,6))
barplot = sns.barplot( x = 'arrival_year', y = 'lead_time', data= bo
okings, color = '#c2d1f0' )

plt.title('Μέσος όρος lead time ανα έτος')
plt.xlabel('Arrival Year')
plt.ylabel('Mean of Lead Time')

for p in barplot.patches:
    height = p.get_height()
    width = p.get_width()
    x, y = p.get_x(), p.get_y()
    barplot.annotate(f'{height:.0f} days earlier', (x + width / 2, y
+ height - 35), ha='center', va='bottom', fontsize=10)

plt.show()

plt.figure(figsize=(12,6))
barplot = sns.barplot( x = 'arrival_year', y = 'lead_time', hue = 'b
ooking_status', data= bookings, palette='vlag' )

plt.title('Μέσος όρος lead time Αφίξεων/Ακυρώσεων ανα έτος')
plt.xlabel('Arrival Year')
plt.ylabel('Mean of Lead Time')

for p in barplot.patches:
    height = p.get_height()
    width = p.get_width()
    x, y = p.get_x(), p.get_y()
```

```
    barplot.annotate(f'{height:.0f} days earlier', (x + width / 2, y
+ height - 35), ha='center', va='bottom', fontsize=10)

plt.show()

plt.figure(figsize=(12,6))
countplot = sns.barplot( y = 'arrival_month', x = 'lead_time', hue =
'booking_status', data= bookings, palette='vlag', orient='h')

for p in countplot.patches:
    height = p.get_height()
    width = p.get_width()
    x, y = p.get_x(), p.get_y()
    count = f'{width:.1f}'
    countplot.annotate(count, (x + width + 11 , y + height / 2 ), ha
='center', va='center', fontsize=12)

plt.title('Μέσος όρος lead time Αφίξεων/Ακυρώσεων ανα μήνα')
plt.ylabel('Arrival Month')
plt.xlabel('Mean of Lead Time')

countplot.set_yticklabels(['January','February','March','April','May
','June','July','August','September','October','November','December'
])

plt.show()

import matplotlib.pyplot as plt
import pandas as pd

# Assuming 'lead_time' is the column representing lead time
bins = [0, 7, 30, 90, float('inf')]
labels = ['0-7 days', '8-30 days', '31-90 days', 'Over 90 days']

# Create a new column for lead time categories
bookings['lead_time_category'] = pd.cut(bookings['lead_time'], bins=
bins, labels=labels)

# Specify colors for the pie charts
cols = ['#e6f2ff', '#ffb3b3']
#labels = ['Not Canceled','Canceled']

# Set up subplots
fig, axes = plt.subplots(2, 2, figsize=(12, 12))
fig.suptitle('Ποσοστο Κρατήσεων/Ακυρώσεων με βάση το Lead Time', fon
tsize=16)

# Iterate through lead time categories
for i, lead_time_category in enumerate(labels):
    ax = axes[i // 2, i % 2]

    # Filter data for the specific lead time category and 'Canceled'
bookings
    data_subset = bookings[(bookings['lead_time_category'] == lead_t
```

```
ime_category) & (bookings['booking_status'])]

    # Plot the pie chart
    data_subset['booking_status'].value_counts().plot.pie(
        labels=['Not Canceled','Canceled'], autopct='%1.1f%%', shado
w=True, colors=cols, ax=ax
    )

    ax.set_title(f'Lead Time: {lead_time_category}')
    ax.set_ylabel('')
    #ax.legend().set_visible(False)

plt.show()

plt.figure(figsize=(12,8))
sns.heatmap( bookings.corr(), annot=True, cmap='RdBu')
plt.show()

plt.figure(figsize=(15, 10))

plt.subplot(1, 2, 1)
countplot = sns.countplot( data=bookings , x='no_of_adults', hue='bo
oking_status', palette = ["#adc2eb", "#e6b3b3"])
plt.title('Adults vs Cancelations',fontweight="bold", size=20)
for label in countplot.containers:
    countplot.bar_label(label)

plt.subplot(1, 2, 2)
countplot2 = sns.countplot(data = bookings, x = 'no_of_children', hu
e='booking_status', palette= ["#adc2eb", "#e6b3b3"])
plt.title('Children vs Cancelations',fontweight="bold", size=20)
for label in countplot2.containers:
    countplot2.bar_label(label)
plt.subplots_adjust(right=1.7)

plt.show()

print(bookings.columns.get_loc('no_of_children') + 1)
bookings.insert(loc=bookings.columns.get_loc('no_of_children') + 1,
column='total_people', value= bookings['no_of_adults']
                + bookings['no_of_children'])
bookings.head()
countplot = sns.countplot(data = bookings, x = 'total_people', hue='
booking_status', palette= ["#adc2eb", "#e6b3b3"])
plt.title('Total people vs Cancelations',fontweight="bold", size=20)
for label in countplot.containers:
    countplot.bar_label(label)
plt.subplots_adjust(right=1.7)

print(bookings.columns.get_loc('no_of_week_nights') + 1)
bookings.insert(loc=bookings.columns.get_loc('no_of_week_nights') +
1, column='total_nights',
                value= bookings['no_of_weekend_nights'] + bookings['
no_of_week_nights'])
```

```
bookings.head()
countplot = sns.countplot(data = bookings, x = 'total_nights', hue='
booking_status', palette= ["#adc2eb", "#e6b3b3"])
plt.title('Total nights vs Cancelations',fontweight="bold", size=20)
for label in countplot.containers:
    countplot.bar_label(label)
plt.subplots_adjust(right=1.7)
ax = pd.crosstab(bookings['total_nights'], bookings['booking_status'
]).plot(kind='bar', stacked=True,

color = ["#e6b3b3" , "#adc2eb"] )


# Create a scatter plot with Seaborn
plt.figure(figsize=(10, 6))
sns.scatterplot(data=bookings, x='total_nights', y= 'lead_time', hue
='booking_status', s=100)

# Set labels and title
plt.xlabel('Booking Duration')
plt.ylabel('Lead time')
plt.title('Scatter Plot of Booking Duration vs Lead time with Bookin
g Status')

# Display the legend
plt.legend(title='Booking Status')

# Show the plot
plt.show()

bins = [0, 3, 7, float('inf')]
labels = ['Short Stay', 'Medium Stay', 'Long Stay']
bookings.insert(loc=bookings.columns.get_loc('total_nights') + 1, co
lumn='stay_duration_category',
                value=pd.cut(bookings['total_nights'], bins=bins, la
bels=labels))

cols = ['#ffdd99','#ffc34d','#e69900']
labels = ['Short Stay', 'Medium Stay', 'Long Stay']
ax = bookings['stay_duration_category'].value_counts().plot.pie(labe
ls = labels, autopct='%1.1f%%',shadow=True, colors=cols)

plt.title('Stay_duration_category')
ax.set_ylabel('')
plt.show()

plt.figure(figsize=(14,6))
palette= ["#adc2eb", "#e6b3b3"]
countplot = sns.countplot( data= bookings, x = 'stay_duration_catego
ry', hue = 'booking_status', palette = palette)

for label in countplot.containers:
    countplot.bar_label(label)
```

```
plt.title('Καταμέτρηση κατηγοριων διαρκειας διαμονης')
plt.xlabel('Stay_duration_category')
plt.show()

plt.figure(figsize=(15, 8))
plt.subplot(1, 2, 1)
sns.countplot(x= 'market_segment_type' , data= bookings, palette='Rd
Bu')
plt.title('Types of market segment', fontweight="bold", size=20)

plt.subplot(1, 2, 2)
sns.countplot(x= 'type_of_meal_plan' , data= bookings, palette='RdBu
')
plt.title('Types of Meal Plan', fontweight="bold", size=20)
plt.subplots_adjust(right=1.7)
plt.show()

meal_plan_counts = bookings['type_of_meal_plan'].value_counts()
explode = (0.1, 0.0, 0.1, 0.3)
colors = sns.color_palette('RdBu'[0:4])

plt.figure(figsize=(6,6))
plt.pie( meal_plan_counts, autopct='%1.1f%%', labels=meal_plan_count
s.index, startangle=90, explode = explode,
        pctdistance=0.85, colors = colors)
plt.show()

plt.figure(figsize=(14, 6))
palette= ["#adc2eb", "#e6b3b3"]
countplot = sns.countplot(x= 'type_of_meal_plan' , data= bookings, h
ue = 'booking_status', palette= palette)
plt.title(' Πληθος τύπων προγράμματος γευμάτων Αφιξεων και Ακυρωσεων
')
plt.xlabel('Type of Meal Plan')
for p in countplot.patches:
    height = p.get_height()
    width = p.get_width()
    x, y = p.get_x(), p.get_y()
    countplot.annotate(f'{height:.0f}', (x + width / 2, y + height -
35), ha='center', va='bottom', fontsize=10)
plt.show()

plt.figure(figsize=(14, 6))
palette= ["#adc2eb", "#e6b3b3"]
countplot = sns.countplot(x= 'market_segment_type' , data= bookings,
hue = 'booking_status', palette=palette)
plt.title('Πλήθος των Types of Market Segment Αφίξεων και Κρατήσεων'
, fontweight="bold", size=20)
plt.xlabel('Type of Market Segment')

for p in countplot.patches:
    height = p.get_height()
    width = p.get_width()
    x, y = p.get_x(), p.get_y()
```

```
    countplot.annotate(f'{height:.0f}', (x + width / 2, y + height -
35), ha='center', va='bottom', fontsize=10)
plt.show()

print(bookings['cancellation_fraction'].value_counts())
bookings['cancellation_fraction'] = bookings['booking_status'].eq('C
anceled').astype(int)
total_bookings = bookings['market_segment_type'].value_counts()

bookings['cancellation_percentage'] = bookings.groupby('market_segme
nt_type')['cancellation_fraction'].transform('mean') * 100
# Σχεδίαση του γράφου
plt.figure(figsize=(14, 6))

barplot = sns.barplot(data=bookings, x='market_segment_type', y='can
cellation_percentage', color='#e6b3b3')

plt.title('Ποσοστό ακύρωσης ανά τύπο τμήματος αγοράς')
plt.xlabel('Τύπος τμήματος αγοράς')
plt.ylabel('Ποσοστό Ακυρώσεων')

# Προσθήκη ποσοστού ακύρωσης πάνω στο γράφημα
for p in barplot.patches:
    height = p.get_height()
    width = p.get_width()
    x, y = p.get_x(), p.get_y()
    percentage = f'{height:.1f}'
    barplot.annotate(percentage + '%', (x + width / 2, y + height /
2), ha='center', va='center', fontsize=10)
plt.show()

plt.subplot(1, 2, 1)
palette= ["#adc2eb", "#e6b3b3"]
sns.countplot(x= 'no_of_weekend_nights' , data= bookings, hue = 'boo
king_status', palette= palette)
plt.title('Πλήθος διανυκτερέσεων Σαβ/κο')
plt.xlabel('Number of weekend nights')
plt.subplot(1, 2, 2)
sns.countplot(x= 'no_of_week_nights' , data= bookings, hue = 'bookin
g_status', palette=palette)
plt.title('Πλήθος διανυκτερέσεων Καθημερινές')
plt.xlabel('Number of week nights')
plt.subplots_adjust(right=1.7)
plt.show()

plt.figure(figsize=(8,6))
countplot = sns.countplot(data = bookings , x = 'repeated_guest', hu
e = 'booking_status', palette = ["#adc2eb", "#e6b3b3"] )
countplot.set_title('Πληθος επαναλαμβανομενων επισκεπτων Αφιξεων/Ακυ
ρωσεων')
new_labels = [ 'Non Repeted guest' , 'Repeated Guest']
countplot.set_xticklabels(new_labels)
for p in countplot.patches:
    height = p.get_height()
```

```
    width = p.get_width()
    x, y = p.get_x(), p.get_y()
    percentage = height
    countplot.annotate(f'{percentage}', (x + width / 2, y + height -
35), ha='center', va='bottom', fontsize=10)
plt.show()

plt.figure(figsize=(8,6))
countplot = sns.countplot(data = bookings , x = 'no_of_previous_canc
ellations', palette = 'vlag' )
countplot.set_title('Προηγούμενες ακυρωμένες κρατήσεις')
for p in countplot.patches:
    height = p.get_height()
    width = p.get_width()
    x, y = p.get_x(), p.get_y()
    countplot.annotate(f'{height:.0f}', (x + width / 2, y + height -
35), ha='center', va='bottom', fontsize=10)
plt.show()

plt.figure(figsize=(8,6))
countplot = sns.countplot(data = bookings , x = 'no_of_previous_canc
ellations', hue = 'booking_status', palette = 'vlag' )
countplot.set_title('Συσχετισμος προηγούμενων ακυρωμένων κρατήσεων μ
ε την κατασταση της παρουσας κρατησης (ακυρωμενη/μη-ακυρωμενη) ')
countplot.legend(loc='upper right')
for p in countplot.patches:
    height = p.get_height()
    width = p.get_width()
    x, y = p.get_x(), p.get_y()
    countplot.annotate(f'{height:.0f}', (x + width / 2, y + height -
35), ha='center', va='bottom', fontsize=10)
plt.show()

plt.figure(figsize=(14,6))
sns.countplot( data= bookings, x = 'no_of_previous_bookings_not_canc
eled', palette='vlag' )
plt.title('Non Cancelations', fontsize = 20)
plt.show()

plt.figure(figsize=(12,5))
barplot = sns.barplot(data = bookings, x = 'arrival_month', y = 'avg
_price_per_room', color = '#adc2eb')
barplot.set_xticklabels(['January','February','March','April','May',
'June','July','August','September','October','November',
                         'December'])
for p in barplot.patches:
    height = p.get_height()
    value = f'{height:.2f}€'
    barplot.annotate(value , (p.get_x() + p.get_width() / 2, height
/ 2), ha='center', va='bottom', fontsize=10, color='black')
plt.xlabel('Arrival Month')
plt.ylabel('Price per room')
plt.title('Μέση τιμή δωματίου ανα μήνα')
plt.show()
```

```
plt.figure(figsize=(12,5))
barplot = sns.barplot(data = bookings, x = 'arrival_month', y = 'avg
_price_per_room', hue = 'booking_status')
barplot.set_xticklabels(['January','February','March','April','May',
'June','July','August','September','October','November',
                         'December'])
for p in barplot.patches:
    height = p.get_height()
    barplot.annotate(f'{height:.1f}€', (p.get_x() + p.get_width() /
2, p.get_y() + height + 2), ha='center', va='bottom',
                     fontsize=10, color='black')
plt.xlabel('Arrival Month')
plt.ylabel('Price per room')
plt.title('Μέση τιμή δωματίου ανα μήνα')
plt.show()

avg_by_month = bookings.groupby('arrival_month')['avg_price_per_room
'].mean()
print(avg_by_month)
plt.figure(figsize=(12,5))
sns.lineplot(data = bookings, x = 'arrival_month', y = 'avg_price_pe
r_room')
plt.xlabel('Month')
plt.ylabel('Avg Price')
plt.xticks(bookings['arrival_month'].unique())
plt.show()

bookings.insert(loc=bookings.columns.get_loc('avg_price_per_room') +
1, column='price_difference',
                value=bookings['avg_price_per_room'] - bookings.grou
pby('arrival_month')['avg_price_per_room'].transform('mean'))
```

11. Παράρτημα Κώδικα Β

ML Model

```
import xgboost as xgb
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
orig_data = pd.read_csv("Hotel Reservations.csv")
orig_data.shape
orig_data.info()
orig_data.describe()
#Feature 1--> Bookings Duration
orig_data.insert(loc=orig_data.columns.get_loc('no_of_week_nights')
+ 1,
                 column='booking_duration', value= orig_data['no_of_
weekend_nights'] + orig_data['no_of_week_nights'])

#Feature 2--> Difference in price and avg price
orig_data.insert(loc=orig_data.columns.get_loc('avg_price_per_room')
+ 1,
                 column='price_difference', value=orig_data['avg_pri
ce_per_room']
                 - orig_data.groupby('arrival_month')['avg_price_per
_room'].transform('mean'))

pip install holidays
from datetime import date
import holidays
import calendar
for ptr in holidays.Portugal(years = 2017).items():
    print(ptr)
print('-----------------------------------------')
for ptr in holidays.Portugal(years = 2018).items():
    print(ptr)

print('-----------------------------------------')
print('Ιδιες ημερομηνιες αργιων/εορτων τα ετη 2017/2018')
value = pd.to_datetime(orig_data['arrival_year'].astype(str) + '-' +
orig_data['arrival_month'].astype(str) + '-' +
                       orig_data['arrival_date'].astype(str), errors
= 'coerce')
orig_data.insert(loc=orig_data.columns.get_loc('arrival_date') + 1,
column = 'arrival_date_format', value = value )

holidays_portugal = holidays.Portugal(years=2018)

# Create a binary indicator for holidays for each booking
orig_data.insert(loc=orig_data.columns.get_loc('arrival_date_format'
) + 1, column = 'is_holiday', value = False )
```

```
for idx, row in orig_data.iterrows():
    arrival_date = row['arrival_date_format']
    total_nights = row['no_of_weekend_nights'] + row['no_of_week_nig
hts']
    departure_date = arrival_date + pd.DateOffset(days=total_nights
- 1)

    # Check if arrival_date is not NaT
    if pd.notna(arrival_date) & pd.notna(departure_date):

     # Check if any day of the booking period is a holiday
        if any(date.date() in holidays_portugal for date in pd.date_
range(arrival_date, departure_date, freq ='D')):
            orig_data.at[idx, 'is_holiday'] = True
orig_data['is_holiday'].value_counts()

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
le_count = 0
for col in orig_data:
    if orig_data[col].dtype == 'object' or orig_data[col].dtype == '
datetime64[ns]' or orig_data[col].dtype == 'bool' :
        orig_data[col] = le.fit_transform(orig_data[col])
        le_count += 1
print('%d στήλες έχουν μετατραπεί.' % le_count)
orig_data.describe()
class_sep = orig_data['booking_status'].value_counts().reset_index()

# Bar plot with matplotlib
plt.figure(figsize=(14, 6))
p1 = plt.bar(class_sep['index'], class_sep['booking_status'], alpha=
0.8)
plt.xticks(class_sep['index'])
plt.bar_label(p1)
plt.title("Κατανομή της εξαρτημένης μεταβλητής (TARGET/Booking Statu
s)")
plt.show()
# Get an exact percentage of not cancelled and cancelled
orig_data['booking_status'].value_counts()/orig_data['booking_status
'].count()
X, y = orig_data.iloc[:,1:-1], orig_data.iloc[:,-1]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=
.33, random_state=123)
y.head()
X.head()
# Instatiate a XGBClassifier
xgb_clf=xgb.XGBClassifier(random_state=123)
# Inspect the parameters
xgb_clf.get_params()
from xgboost import XGBClassifier
from sklearn.model_selection import RandomizedSearchCV, StratifiedKF
old, GridSearchCV
from sklearn.metrics import accuracy_score, f1_score, classification
```

```
_report, confusion_matrix
xgb_param_grid={"n_estimators":[100,600,1000],
                "max_depth":[3,4,5],
                "learning_rate":[0.01,0.1,0.3],
                'scale_pos_weight':[0.487228987]}

xgb = XGBClassifier(objective="binary:logistic", eval_metric="auc",
random_state=123)

cv_f=StratifiedKFold(n_splits=3,shuffle=True)

rand_search = GridSearchCV(estimator=xgb, param_grid=xgb_param_grid,
n_jobs=2, cv=cv_f, verbose=1, scoring='roc_auc')

rand_search.fit(X_train,y_train)

preds= rand_search.predict(X_test)

print("The accuracy score is: " , accuracy_score(y_test, preds))
print("Below, it is presented the classification report")
print(classification_report(y_test, preds))
print("An initial presentation of the confusion matrix")
print(confusion_matrix(y_test, preds))
print("Best parameters found:", rand_search.best_params_)
print("Best score found:", rand_search.best_score_)
import matplotlib.pyplot as plt
from sklearn.datasets import make_classification
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
cm=confusion_matrix(y_test, preds, labels=rand_search.classes_)
disp=ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=['Μη
Ακυρωμένη','Ακυρωμένη'])
fig, ax = plt.subplots(figsize=(10,6))
disp.plot(ax=ax)
plt.title(" Πίνακας Σύγχυσης /The confusion matrix")
plt.ylabel("Πραγματική τιμή (True label)")
plt.xlabel("Τιμή που προβλέφθηκε (Predicted label)")
plt.show()

from sklearn import metrics
preds2 = rand_search.predict_proba(X_test)[::,1]
fpr,tpr, _ = metrics.roc_curve(y_test, preds2)
plt.subplots(figsize=(10,6))
plt.plot(fpr,tpr)
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.title("The ROC curve")
plt.show()
auc = metrics.roc_auc_score(y_test, preds2)
plt.subplots(figsize=(10,6))
plt.plot(fpr,tpr,label="AUC="+str(auc))
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.legend(loc=4)
plt.title("The AUC curve")
```

```
plt.show()

from xgboost import plot_importance
xgb_cl = rand_search.best_estimator_
fig, ax = plt.subplots(figsize=(10, 6))
plot_importance(xgb_cl, importance_type='weight', max_num_features=1
5, ax=ax)
plt.show()
xgb_cl = rand_search.best_estimator_
fig, ax = plt.subplots(figsize=(10, 6))
plot_importance(xgb_cl, importance_type='gain', max_num_features=15,
ax=ax)
plt.show()


import shap
# compute SHAP values
explainer = shap.Explainer(rand_search.best_estimator_, X_train)
shap_values = explainer(X_train)
shap_values.base_values
shap_values.shape()
shap.plots.bar (shap_values)
shap.plots.beeswarm(shap_values)
shap.summary_plot(shap_values, plot_type='violin')
shap.plots.scatter(shap_values[:, "lead_time"])
shap.plots.scatter(shap_values[:, "market_segment_type"])
shap.plots.bar(shap_values[0])
shap.plots.waterfall(shap_values[0])
import matplotlib
import shap
shap.initjs()
shap.force_plot(shap_values[0], matplotlib=True)
```

Βιβλιογραφία

Adil, M., Ansari, M. F., Alahmadi, A., Wu, J.-Z., & Chakrabortty, R. K. (2021**). Solving the Problem of Class Imbalance in the Prediction of Hotel Cancelations: A Hybridized Machine Learning Approach. Processes**, 9(10), 1713.

https://doi.org/10.3390/pr9101713

ALOTAIBI, E. (2020). **Application of Machine Learning in the Hotel Industry: A Critical Review**. Journal of Association of Arab Universities for Tourism and Hospitality, 0.

https://doi.org/10.21608/jaauth.2020.38784.1060

Antonio, N., de Almeida, A., & Nunes, L. (2019). **An Automated Machine Learning Based Decision Support System to Predict Hotel Booking Cancellations**. Data Science Journal, 18.

https://doi.org/10.5334/dsj-2019-032

Baralis, E. M. (2022). **Explainable AI for business decision-making.** [Master Thesis, Politecnico di Torino].

https://webthesis.biblio.polito.it/secure/19854/1/tesi.pdf

Biswal, A. (2021, April 27). **The Complete Guide on Overfitting and Underfitting in Machine Learning. Simplilearn.com.**

https://www.simplilearn.com/tutorials/machine-learning-tutorial/overfitting-and-underfitting

Burgt, J. v. d. (2020). **Explainable AI in banking.** Journal of Digital Banking, 4(4).

Chen, S., Ngai, E. W. T., Ku, Y., Xu, Z., Gou, X., & Zhang, C. (2023). **Prediction of hotel booking cancellations: Integration of machine learning and probability model based on interpretable feature interaction**. *Decision Support Systems*, 113959.

https://doi.org/10.1016/j.dss.2023.113959

Contributors to Wikimedia projects. (2003, September 15). Receiver operating characteristic - Wikipedia. Wikipedia, the free encyclopedia.

https://en.wikipedia.org/wiki/Receiver_operating_characteristic

Exploring Explainable AI with LIME Technology. (n.d.). Where data becomes art | Steadforce.

https://www.steadforce.com/blog/explainable-ai-with-lime

File:Sensitivity and specificity.svg - Wikimedia Commons. (n.d.). Wikimedia Commons.

https://commons.wikimedia.org/wiki/File:Sensitivity_and_specificity.svg

Grennan, L., Kremer, A., Singla, A., & Zipparo, P. (2022, September 29). **Why businesses need explainable AI and how to deliver it**. McKinsey & Company.

https://www.mckinsey.com/capabilities/quantumblack/our-insights/why-businesses-need-explainable-ai-and-how-to-deliver-it

Hotel Reservations Dataset. (n.d.). Kaggle: Your Machine Learning and Data Science Community.

https://www.kaggle.com/datasets/ahsan81/hotel-reservations-classification-dataset?select=Hotel+Reservations.csv

How to Check the Accuracy of Your Machine Learning Model. (n.d.). Deepchecks.

https://deepchecks.com/how-to-check-the-accuracy-of-your-machine-learning-model/

Hyperparameter Tuning with GridSearchCV. (n.d.). Great Learning Blog: Free Resources what Matters to shape your Career!

https://www.mygreatlearning.com/blog/gridsearchcv/

Inc, S. (2021, May 1). **TOP 10 MACHINE LEARNING LIBRARIES**. Medium.

https://swapincvec.medium.com/top-10-machine-learning-libraries-e049c4dc644

info8425872. (2020, March 19). **Η ιστορία για τα πρώτα ξενοδοχεία στην Ελλάδα, την Ευρώπη και την Αμερική**. Morpho Hotel Design.

https://www.morphohoteldesign.com/post/i-istoria-gia-ta-prota-xenodoxeia-stin-ellada-tin-europi-kai-tin-americi

Kozan, M. (2021, September 1). **Supervised and Unsupervised Learning (an Intuitive Approach).** Medium.

https://medium.com/@metehankozan/supervised-and-unsupervised-learning-an-intuitive-approach-cd8f8f64b644

Lee, M., Mu, X., & Zhang, Y. (2020). **A MACHINE LEARNING APPROACH TO IMPROVING FORECASTING ACCURACY OF HOTEL DEMAND: A COMPARATIVE ANALYSIS OF NEURAL NETWORKS AND TRADITIONAL MODELS**. Issues In Information Systems.

https://doi.org/10.48009/1_iis_2020_12-21

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "**Why should I trust you?" Explaining the predictions of any classifier.** In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pages 1135–1144, 2016.

Nanang, H. (2021). **Exploratory Data Analysis & Booking Cancelation Prediction on Hotel Booking Demands Datasets.** Journal of Applied Data Sciences, 2(1).

https://doi.org/10.47738/jads.v2i1.20

Novakovic, J. (2021). **Hotel reservation cancellations: analysis and prediction using machine learning algorithms.** International Academic Journal, 2(1).

Sánchez-Medina, A. J., & C-Sánchez, E. (2020). **Using machine learning and big data for efficient forecasting of hotel booking cancellations.** International Journal of Hospitality Management, 89, 102546.

https://doi.org/10.1016/j.ijhm.2020.102546

Sana Syed, Mohammad Al-Boni, Marium N. Khan, Kamran Sadiq, Najeeha T Iqbal, Christopher A Moskaluk, Paul Kelly, Beatrice Amadi, S Asad Ali, Sean R Moore, et al. **Assessment of machine learning detection of environmental enteropathy and celiac disease in children.** JAMA network open, 2(6):e195822–e195822, 2019.

Sánchez, E. C., Sánchez-Medina, A. J., & Pellejero, M. (2020). **Identifying critical hotel cancellations using artificial intelligence.** Tourism Management Perspectives, 35, 100718.

https://doi.org/10.1016/j.tmp.2020.100718

Sharma, A. (2022, December 6). Industry 4.0 – **Intelligent & Smart Factories: Technologies Shaping the future of Oil, Gas & Energy (OGE) Industry.** All Blog Posts | SAP Community.

https://blogs.sap.com/2022/12/06/industry-4.0-intelligent-smart-factories-technologies-shaping-the-future-of-oil-gas-energy-oge-industry/

Singh, C. (2021, October 21). **Explainable AI: Why Should Business Leaders Care?** Medium.

https://towardsdatascience.com/explainable-ai-why-should-business-leaders-care-5e5078c609b5

Tchuente, D., Lonlac, J., & Kamsu-Foguem, B. (2024). **A methodological and theoretical framework for implementing explainable artificial intelligence (XAI) in business applications.** Computers in Industry, 155, 104044.

https://doi.org/10.1016/j.compind.2023.104044

The Jupyter Notebook — Jupyter Notebook 7.0.3 documentation. (n.d.). Jupyter Notebook Documentation — Jupyter Notebook 7.0.3 documentation.

https://jupyter-notebook.readthedocs.io/en/stable/notebook.html

The Role Of Explainable AI In Business Decision Making. (2023). Pangeatech.

https://pangeatech.net/the-role-of-explainable-ai-in-business-decision-making/

Timamopoulos, C. (2020). **Anomaly Detection: Predicting hotel booking cancellations** [Master Thesis, International Hellenic University].

https://repository.ihu.edu.gr//xmlui/handle/11544/29631

Trevisan, V. (2022, January 17). **Using SHAP Values to Explain How Your Machine Learning Model Works.** *Medium.*

https://towardsdatascience.com/using-shap-values-to-explain-how-your-machine-learning-model-works-732b3f40e137

Park, M. S., Son, H., Hyun, C., & Hwang, H. J. (2021). **Explainability of Machine Learning Models for Bankruptcy Prediction**. IEEE Access, 9, 124887–124899.

https://doi.org/10.1109/access.2021.3110270

Types of Machine Learning - Javatpoint. (2021). www.javatpoint.com. *https://www.javatpoint.com/types-of-machine-learning*

Welcome to the SHAP documentation — SHAP latest documentation. (n.d.). Welcome to the SHAP documentation — SHAP latest documentation.

https://shap.readthedocs.io/en/latest/index.html

What is Jupyter Notebook? | Domino Data Lab. (n.d.). Domino Data Lab | Unleash Data Science at Scale. *https://domino.ai/data-science-dictionary/jupyter-notebook*

What is Machine Learning? | IBM. (n.d.). IBM - Deutschland | IBM. *https://www.ibm.com/topics/machine-learning*

What Is Python Used For? A Beginnerâ□□s Guide. (n.d.). Coursera.

https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python

What is XGBoost? (n.d.). NVIDIA Data Science Glossary.

https://www.nvidia.com/en-us/glossary/data-science/xgboost/

Why Python keeps growing, explained. (n.d.). The GitHub Blog.

https://github.blog/2023-03-02-why-python-keeps-growing-explained/

Wohlwend, B. (2023, July 23). **Decision Tree, Random Forest και XGBoost: An Exploration into the Heart of Machine Learning.***https://medium.com/*.

https://medium.com/@brandon93.w/decision-tree-random-forest-and-xgboost-an-exploration-into-the-heart-of-machine-learning-90dc212f4948

Ένας ολοκληρωμένος οδηγός για τη Feature Engineering το 2023 - Backlnktools. (n.d.). Backlnktools.

https://backlinktools.xyz/a-comprehensive-guide-to-feature-engineering-in-2023/

Φώτη, Μ. (2023). **Η τεχνητή νοημοσύνη στον χειρουργικό τομέα** [Master Thesis, Δημοκρίτειο Πανεπιστήμιο Θράκης, τμήμα Ιατρικής].

https://repo.lib.duth.gr/jspui/handle/123456789/15944