

Προγραμματιστική Εργασία Πρόβλεψη κόστους ασφάλισης οχημάτων

Χαρά Τσίρκα, Πρόδρομος Αβραμίδης, Γεώργιος Γεροντίδης

{ctsirka, pavramidis, ggerontidis}@e-ce.uth.gr
8 εξάμηνο



Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών
Υπολογιστών
Πανεπιστήμιο Θεσσαλίας, Βόλος

Εξόρυξη Δεδομένων 2023-24
Διδάσκον: Μ.Βασιλακόπουλος

Μάιος 2024

1 Εισαγωγή

Στόχος της συγκεκριμένης εργασίας είναι η ανάπτυξη μίας εφαρμογής επιχειρησιακής αναλυτικής και ειδικότερα μίας εφαρμογής πρόβλεψης της τιμής ασφάλισης οχημάτων. Μία τέτοια εφαρμογή μπορεί να φανεί εξαιρετικά χρήσιμη σε εταιρείες ασφάλισης οχημάτων, καθώς τους επιτρέπει να λαμβάνουν τεκμηριωμένες επιχειρηματικές αποφάσεις και άρα να αποφεύγουν το ενδεχόμενο ζημίας. Λαμβάνοντας υπόψη τους διάφορους παράγοντες που επηρεάζουν το κόστος ασφάλισης οχημάτων, η εφαρμογή που αναπτύχθηκε παρουσιάζει στον χρήστη-υπάλληλο μία προτεινόμενη τιμή χρέωσης για κάθε πελάτη που σκοπεύει να ασφαλίσει το όχημά του.

Για την πρόβλεψη της τιμής ασφάλισης χρησιμοποιήθηκε το σύνολο δεδομένων "Motor_vehicle_insurance_data.csv" το οποίο βρίσκεται στα παραδοτέα αρχεία. Η διαδικασία της εργασίας ξεκινά με την κατάλληλη προ-επεξεργασία των δεδομένων, κατά την οποία πραγματοποιήθηκε εξερευνητική ανάλυση (exploratory analysis) για τον προσδιορισμό των κριτηρίων διαχωρισμού των δεδομένων αλλά και του βαθμού επίδρασης κάθε χαρακτηριστικού (feature) του συνόλου δεδομένων στα αποτελέσματα. Εφαρμόστηκαν τρία διαφορετικά μοντέλα πρόβλεψης (XGBoost, RandomForest, NeuralNetwork), τα αποτελέσματα των οποίων οπτικοποιήθηκαν, αξιολογήθηκαν και συγκρίθηκαν.

Στη συνέχεια αναπτύχθηκε η εφαρμογή πρόβλεψης της τιμής ασφάλισης οχημάτων ανάλογα με την είσοδο του χρήστη, στην οποία χρησιμοποιήθηκε το μοντέλο που είχε την καλύτερη απόδοση.

2 Περιγραφή dataset

Το dataset το οποίο επιλέξαμε αποτελείται από 30 μεταβλητές (columns) και 105555 εγγραφές. Στους παρακάτω πίνακες δίνεται μία σύντομη περιγραφή της κάθε μεταβλητής:

Μεταβλητή	Περιγραφή
ID	Εσωτερικός αριθμός αναγνώρισης που εκχωρείται σε κάθε ετήσια σύμβαση που επισημοποιείται από έναν ασφαλισμένο. Κάθε ασφαλισμένος μπορεί να έχει πολλές σειρές στο σύνολο δεδομένων, που αντιπροσωπεύουν διαφορετικές προσόδους του προϊόντος.
Date_start_contract	Ημερομηνία έναρξης του συμβολαίου (HH/MM/YYYY).
Date_last_renewal	Ημερομηνία τελευταίας ανανέωσης του συμβολαίου (HH/MM/YYYY).
Date_next_renewal	Ημερομηνία επόμενης ανανέωσης του συμβολαίου (HH/MM/YYYY).
Distribution_channel	Κανάλι μέσω του οποίου έγινε το ασφαλιστήριο, 0: για Πράκτορα, 1: για Ασφαλιστικοί μεσίτες.
Date_birth	Ημερομηνία γέννησης του ασφαλισμένου που δηλώνεται στο ασφαλιστήριο (HH/MM/YYYY).
Date_driving_licence	Ημερομηνία έκδοσης της άδειας οδήγησης του ασφαλισμένου (HH/MM/YYYY).

Μεταβλητή	Περιγραφή
Seniority	Συνολικός αριθμός ετών που ο ασφαλισμένος έχει συνδεθεί με την ασφαλιστική οντότητα, υποδεικνύοντας το επίπεδο αρχαιότητάς του.
Policies_in_force	Συνολικός αριθμός συμβολαίων που κατείχε ο ασφαλισμένος στην ασφαλιστική οντότητα κατά την περίοδο αναφοράς.
Max_policies	Μέγιστος αριθμός συμβολαίων που είχε ποτέ σε ισχύ ο ασφαλισμένος με τον ασφαλιστικό φορέα.
Max_products	Μέγιστος αριθμός προϊόντων που κατέχει ο ασφαλισμένος ταυτόχρονα σε οποιαδήποτε δεδομένη χρονική στιγμή.
Lapse	Αριθμός πολιτικών που ο πελάτης έχει ακυρώσει ή έχει ακυρωθεί λόγω μη πληρωμής κατά το τρέχον έτος λήξης, εξαιρουμένων αυτών που έχουν αντικατασταθεί από άλλο συμβόλαιο.
Date_Lapse	Ημερομηνία ακύρωσης της σύμβασης (HH/MM/YYYY).
Payment	Τελευταία μέθοδος πληρωμής της πολιτικής 1: εξαμηνιαία πληρωμή, 0: ετήσια πληρωμή
Premium	Καθαρό ποσό ασφαλιστρού που σχετίζεται με το ασφαλιστήριο συμβόλαιο κατά τη διάρκεια του τρέχοντος έτους.
Cost_claims_year	Συνολικό κόστος ζημιών που πραγματοποιήθηκαν για το ασφαλιστήριο συμβόλαιο κατά τη διάρκεια του τρέχοντος έτους.
N_claims_year	Συνολικός αριθμός ζημιών που πραγματοποιήθηκαν για το ασφαλιστήριο συμβόλαιο κατά τη διάρκεια του τρέχοντος έτους.
N_claims_history	Συνολικός αριθμός απαιτήσεων που υποβλήθηκαν καθ' όλη τη διάρκεια του ασφαλιστηρίου συμβολαίου.
R_Claims_history	Παρέχει μια ένδειξη του ιστορικού συχνότητας αξιώσεων του ασφαλιστηρίου.
Type_risk	Τύπος κινδύνου για κάθε όχημα 1: μοτοσικλέτα, 2: μικρά φορτηγά, 3: επιβατικά οχήματα, 4: αγροτικά οχήματα
Area	0: αγροτική περιοχή, 1: αστική περιοχή (>30.000 κάτοικοι όσον αφορά τις κυκλοφοριακές συνθήκες)
Second_driver	1: περισσότεροι από ένας δηλωμένοι οδηγοί 0: μόνο ένας δηλωμένος οδηγός
Year_matriculation	Έτος καταχώρησης οχήματος (YYYY)
Power	Ίπποι δύναμης οχήματος
Cylinder_capacity	Χωρητικότητα κυλίνδρων του οχήματος
Value_vehicle	Αξία αγοράς οχήματος στις 31/12/2019
N_doors	Αριθμός θυρών οχήματος
Type_fuel	Τύπος καυσίμου, P: πετρέλαιο, D: ντίζελ
Length	Μήκος του οχήματος σε m
Weight	Βάρος του οχήματος σε kg

3 Data preprocessing

Το πρώτο βήμα για την προεπεξεργασία των δεδομένων ήταν να κρατήσουμε μία γραμμή για κάθε 'ID'. Σε ένα 'ID' μπορεί να αντιστοιχούν περισσότερες από μία γραμμές που αντιπροσωπεύουν το ίδιο συμβόλαιο του ίδιου πελάτη για διαφορετική χρονική περίοδο. Έτσι, για κάθε 'ID' κρατάμε την τελευταία ανανέωση του συμβολαίου, δηλαδή την γραμμή με το μεγαλύτερο χρονολογικά 'last_renewal_date'. Έπειτα, στη θέση του 'premium' υπολογίζουμε και τοποθετούμε τον μέσο όρο των 'premium' όλων των γραμμών με κοινό 'ID'.

Το δεύτερο βήμα ήταν η επεξεργασία όλων των ημερομηνιών. Ειδικότερα, οι στήλες 'Date_birth', 'Date_driving_license', 'Date_start_contract', 'Date_last_renewal', 'Date_next_renewal', 'Date_lapse' δίνονται στην μορφή HH/MM/EEEE. Αρχικά, για κάθε μία από αυτές τις μεταβλητές κρατήσαμε το έτος (EEEE) και στην συνέχεια πραγματοποιώντας τις κατάλληλες αφαιρέσεις δημιουργήσαμε νέες στήλες στο dataset που πήραν την θέση αυτών που αναφέρθηκαν νωρίτερα. Έτσι, δημιουργήσαμε τις στήλες: 'Age' που προσδιορίζει την ηλικία του πελάτη, 'Years_driving' που προσδιορίζει πόσα χρόνια οδηγεί ο πελάτης, 'Year_on_road' που προσδιορίζει πόσα χρόνια κυκλοφορεί το κάθε όχημα υπό την κατοχή συγκεκριμένου πελάτη και 'Years_on_policy' που προσδιορίζει πόσα χρόνια ο πελάτης βρίσκεται στον ίδιο τύπο συμβολαίου. Πρέπει να σημειωθεί πως το dataset περιέχει δεδομένα μέχρι και το 2019. Για να έχουμε μια σωστή εικόνα των χρονολογιών σε όλες αυτές τις μεταβλητές που δημιουργήσαμε, χρησιμοποιήσαμε ως σημείο αναφοράς την χρονολογία τελευταίας ανανέωσης του συμβολαίου. Για παράδειγμα η μεταβλητή 'Age' προκύπτει από την αφαίρεση: 'Age' = 'Date_last_renewal' - 'Date_birth'.

Επιπλέον, δημιουργήσαμε μία ακόμη νέα στήλη με όνομα 'accidents' για να υπάρχει μια συσχέτιση μεταξύ των αριθμών των ατυχημάτων με τα χρόνια που ένας πελάτης είναι ασφαλισμένος στην εταιρεία. Διαχειριστήκαμε την απουσία τιμών με δύο τρόπους. Στην στήλη 'Length' αντικαταστήσαμε τα κενά πεδία με τον μέσο όρο των τιμών της στήλης. Στην στήλη 'Type_fuel' αντικαταστήσαμε τα κενά πεδία με την τιμή 'Unknown'.

Μετά από δοκιμές διαπιστώσαμε πως κάποιες μεταβλητές του dataset δεν συνεισφέραν καθόλου στην βελτίωση της απόδοσης και παραλείφθηκαν. Οι στήλες που χρησιμοποιήθηκαν τελικά είναι οι:

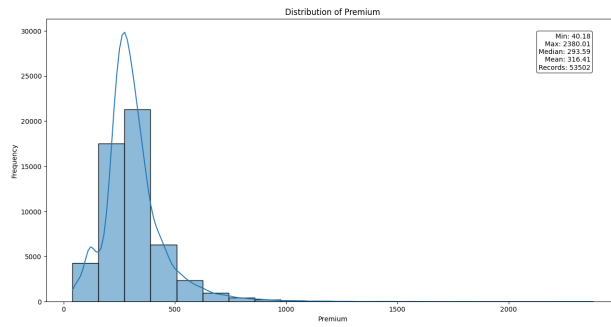
Αριθμητικές: 'Seniority', 'Power', 'Years_on_road', 'R_claims_history', 'accidents', 'Years_on_policy', 'Value_vehicle', 'Age', 'Years_driving', 'N_claims_history', 'Weight', 'Cylinder_capacity', 'Length', 'Contract_year', 'Policies_in_force'
Κατηγορικές: 'Type_risk', 'Area', 'Second_driver', 'Distribution_channel', 'Type_fuel', 'Payment', 'Lapse'
Στόχος: 'Premium'

Για τις αριθμητικές χρησιμοποιούμε το StandardScaler από την βιβλιοθήκη scikit-learn το οποίο κανονικοποιεί τα δεδομένα και κάνει την μέση τιμή (mean) κάθε στήλης να είναι 0 και την απόκλιση (variance) να είναι 1.

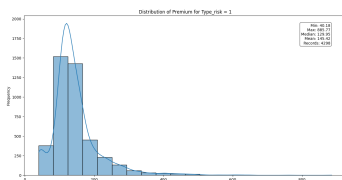
Για τις κατηγορικές χρησιμοποιούμε το OneHotEncoder από την βιβλιοθήκη scikit-learn που δημιουργεί όσες στήλες όσες και οι μοναδικές τιμές κάθε κατηγορίας και για κάθε στήλη βάζει 1 αν ανήκει και 0 αν δεν ανήκει.

4 Ανάλυση Δεδομένων

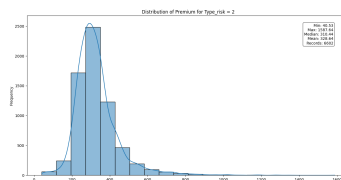
Στο πρώτο διάγραμμα [Εικ.1] παρουσιάζεται η κατανομή των ασφαλίσεων για όλες τις καταχωρίσεις. Τα επόμενα διαγράμματα [Εικ.2] δείχνουν την κατανομή των ασφαλίσεων για κάθε κατηγορία οχήματος ξεχωριστά.



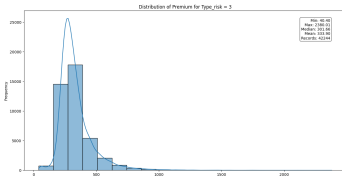
Εικ. 1: Κατανομή Premium



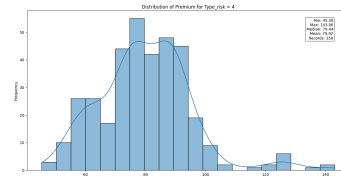
(α') Μοτοσυκλέτα



(β') Βανάκι

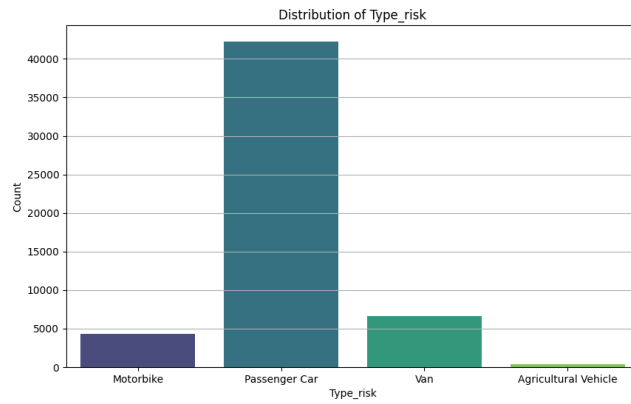


(γ') Επιβατικό Αυτοκίνητο

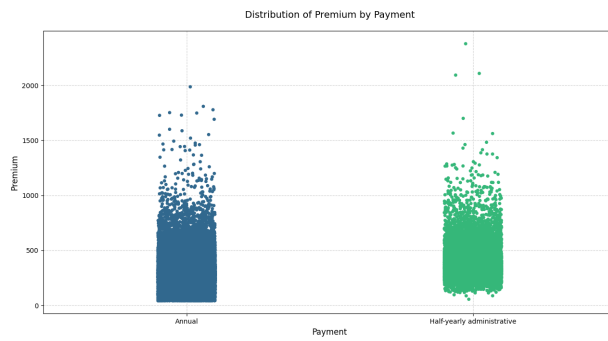


(δ') Αγροτικό Όχημα

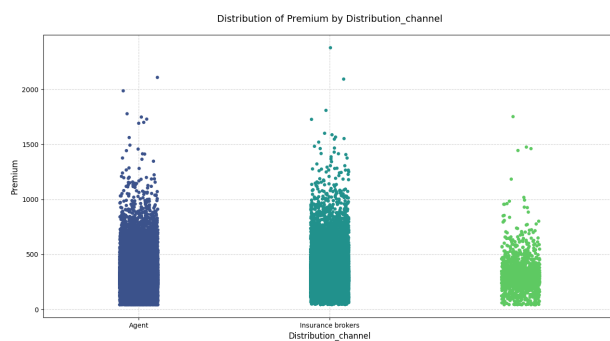
Εικ. 2: Κατανομή Premium ανα κατηγορία οχήματος



Εικ. 3: Αριθμός εγγραφών σε κάθε κατηγορία



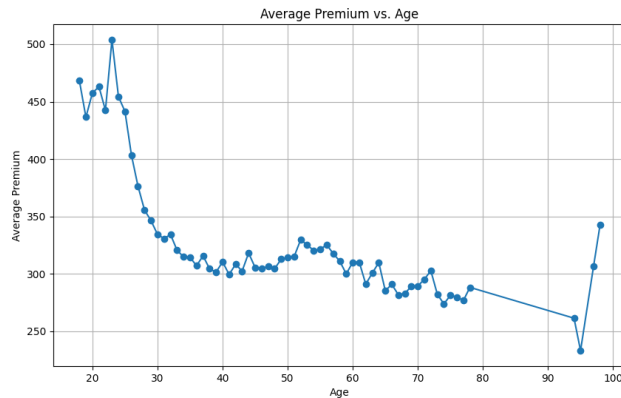
Εικ. 4: Premium και τρόπος πληρωμής



Εικ. 5: Premium και κανάλι διανομής



Εικ. 6: Premium και δεύτερος οδηγός



Εικ. 7: Premium και ηλικία

Παρατήρηση 1 Παρατηρούμε ότι η κατηγορία των επιβατικών οχημάτων έχει πολύ περισσότερες εγγραφές από τις άλλες [Εικ. 3] και ότι υπάρχει μεγάλη διαφορά των Premium σε κάθε κατηγορία. [Εικ. 2]

Παρατήρηση 2 Παρατηρούμε ότι για τα χαρακτηριστικά τρόπος πληρωμής, κανάλι διανομής και δεύτερος οδηγός μία κατηγορία έχει πιο φθηνό ασφάλιστρο (Premium) από τις υπόλοιπες. [Εικ. 4, Εικ. 5, Εικ. 6]

Παρατήρηση 3 Παρατηρούμε ότι όσο πιο νέος είναι ο οδηγός τόσο πιο ακριβό το ασφάλιστρο (premium). [Εικ. 7]

5 Εκπαίδευση μοντέλων

Δοκιμάσαμε να εκπαιδεύσουμε τα μοντέλα με δύο τρόπους. Αρχικά τα εκπαιδεύσαμε μη διαχωρίζοντας τα ανά τύπο οχήματος ("Type_risk") και στην συνέχεια διαχωρίζοντας τα. Ύστερα από παρατήρηση των αποτελεσμάτων και της [Παρατήρησης 1] φαινόταν καλύτερη η επιλογή να εκπαιδεύσουμε τα μοντέλα για κάθε τύπο οχήματος ξεχωριστά. Για την εκπαίδευση των μοντέλων έχει χρησιμοποιηθεί το 80% των δεδομένων και για την αξιολόγηση το 20 %.

5.1 XGBoost

Το πρώτο μοντέλο πρόβλεψης που χρησιμοποιήσαμε ήταν το XGBoost. Είναι μοντέλο μηχανικής μάθησης που βασίζεται στη μέθοδο gradient boosting σύμφωνα με την οποία πολλαπλά δέντρα αποφάσεων, τα οποία κατασκευάζονται διαδοχικά με σκοπό να διορθώσουν τα σφάλματα των προηγούμενων. Παρακάτω αναφέρονται οι παράμετροι

που επιλέχθηκαν για το συγκεκριμένο μοντέλο, ενώ στις παρενθέσεις βρίσκεται το συνηθισμένο εύρος τιμών, δηλαδή το εύρος που έγιναν οι δοκιμές για να πετύχουμε το βέλτιστο αποτέλεσμα.

Objective (linear): Το 'reg:squarederror' είναι το προεπιλεγμένο και ευρέως χρησιμοποιούμενο loss function για regression tasks, παρέχοντας ένα σαφές πλαίσιο για τη βελτιστοποίηση του μοντέλου XGBoost. Συγκεκριμένα, είναι κατάλληλο για την πρόβλεψη ασφάλιστρων, καθώς στοχεύει στην επίτευξη υψηλής ακρίβειας στην πρόβλεψη της συνεχούς τιμής των ασφάλιστρων.

Eval_metric ('rmse', 'mae'): Για το eval_metric, επιλέξαμε το 'rmse' (Root Mean Squared Error) επειδή είναι ένα ευρέως αποδεκτό μέτρο για regression tasks. Αναλυτικότερα, δίνει μεγαλύτερη βαρύτητα σε μεγαλύτερες αποκλίσεις (λάθη) μεταξύ πραγματικών και προβλεπόμενων τιμών, γεγονός που μπορεί να είναι χρήσιμο στην ασφαλιστική πρόβλεψη όπου τα μεγάλα λάθη μπορεί να είναι πιο σοβαρά.

Learning_rate (0.001-0.3): Ο ρυθμός μάθησης (learning rate) επηρεάζει πόσο γρήγορα προσαρμόζεται το μοντέλο κατά την εκπαίδευση. Επιλέχθηκε η τιμή 0.01 έπειτα από δοκιμές, διότι όντας σχετικά χαμηλή τιμή επιβραδύνει τη διαδικασία μάθησης, μειώνοντας την πιθανότητα υπερβολικής προσαρμογής (overfitting).

Max_depth (3-10): Το μέγιστο βάθος των δέντρων στο μοντέλο. Ένας μεγαλύτερος αριθμός επιτρέπει πιο περίπλοκα δέντρα, αλλά μπορεί να οδηγήσει σε overfitting, επομένως επιλέχθηκε η τιμή 6.

Min_child_weight (1-10): Ελάχιστο βάρος που πρέπει να έχει ένας κόμβος για να διαχωριστεί. Επηρεάζει την ευαισθησία του μοντέλου στις διακυμάνσεις στα δεδομένα. Ύστερα από δοκιμές προτιμήθηκε ο αριθμός βάρους 5.

Subsample (0.5-1): Αναφέρεται στο ποσοστό των δειγμάτων που θα χρησιμοποιηθούν σε κάθε επανάληψη της εκπαίδευσης. Ένα ποσοστό 0.8 σημαίνει ότι το 80% των δειγμάτων θα χρησιμοποιηθεί.

Colsample_bytree (0.5-1): Το ποσοστό των χαρακτηριστικών (columns) που θα επιλεγθούν τυχαία για να κατασκευαστεί κάθε δέντρο στο μοντέλο. Ένα ποσοστό 0.8 σημαίνει ότι το 80% των χαρακτηριστικών θα χρησιμοποιηθεί σε κάθε δέντρο.

N_estimators (100-2000): Ο αριθμός των δέντρων που θα κατασκευαστούν στο μοντέλο. Δοκιμάστηκαν διάφοροι αριθμοί, συμπεριλαμβανομένων μεγαλύτερων από 1000, αλλά δεν υπήρχε μεγάλη βελτίωση στην απόδοση του μοντέλου, ενώ ο χρόνος εκπαίδευσης αυξανόταν σημαντικά. Έτσι, ως βέλτιστη θεωρήθηκε η τιμή 1000.

Seed: Για να καταφέρουμε να εξασφαλίσουμε επαναληψιμότητα στα δεδομένα μας και να αφαιρεθεί η τυχαιότητα, επιλέξαμε έναν σταθερό αριθμό seed 42.

Parameter	Before Tuning	After Tuning
objective	reg:squarederror	reg:squarederror
eval_metric	rmse	rmse
learning_rate	0.05	0.01
max_depth	4	5
min_child_weight	1	5
subsample	0.7	0.8
colsample_bytree	0.7	0.8
n_estimators	500	1000

Πίνακας 1: XGBoost Parameter Settings Before and After Tuning

	Before Tuning	After Tuning
	Mean squared error	Mean squared error
Motorbikes	2298.58	2228.745
Vans	8087.68	7846.43
Passenger Cars	10776.074	10693.593
Agricultural Vehicles	356.954	311.899

Πίνακας 2: XGBoost Parameter Tuning Results

5.2 Random forest

Το δεύτερο μοντέλο πρόβλεψης που χρησιμοποιήσαμε ήταν το RandomForest. Είναι μοντέλο μηχανικής μάθησης που αποτελείται από πολλαπλά δέντρα απόφασης, συνδυάζοντας τα αποτελέσματά τους. Παρακάτω αναφέρονται οι παράμετροι που επιλέχθηκαν για το συγκεκριμένο μοντέλο, ενώ στις παρενθέσεις βρίσκεται το συνηθισμένο εύρος τιμών, δηλαδή το εύρος που έγιναν οι δοκιμές για να πετύχουμε το βέλτιστο αποτέλεσμα.

N_estimators (100-1000): Επιλέχθηκε η τιμή 700, διότι παρατηρήθηκε ότι, παρόλο που η αύξηση της τιμής βελτίωνε την απόδοση του RF, ο χρόνος εκτέλεσης αυξανόταν σημαντικά λόγω της δημιουργίας πολλών δέντρων.

Max_depth (10-20): Επιλέχθηκε η τιμή 15, καθώς η αύξηση της τιμής αυξάνει τον κίνδυνο overfitting, αν και προσφέρει σημαντική βελτίωση στην απόδοση του RF.

Min_samples_split (2-10): Επιλέχθηκε η τιμή 4, που βρίσκεται στη μέση του εύρους τιμών, για να ισορροπήσει μεταξύ της πρόληψης του overfitting και της διατήρησης της απόδοσης του μοντέλου.

Min_samples_leaf (2-10): Επιλέχθηκε η τιμή 2, διότι παρατηρήθηκε ότι η αύξηση της τιμής αυτής της παραμέτρου βελτίωνε μόνο την απόδοση οχημάτων "type_risk4", ενώ έπεφτε η συνολική απόδοση σε κάθε άλλο τύπο οχήματος.

Max_features ('sqrt', 'log2', ή float 0.1-1): Συνήθως, σε αλγορίθμους RF προτιμάται η τιμή 'sqrt', και αυτή επιλέχθηκε και από εμάς για καλύτερη απόδοση.

Bootstrap (True, False): Επιλέξαμε να έχουμε bootstrap, καθώς βελτιώνονται σημαντικά τα αποτελέσματα σε σύγκριση με την επιλογή της μη ύπαρξής τους.

N_jobs (-1, 1, αριθμός πυρήνων): Με την επιλογή της τιμής -1, χρησιμοποιούμε όλους τους διαθέσιμους πυρήνες.

Random_state: Πρόκειται για τιμή τυχαιότητας όπως το seed στο XGBoost. Έχει επιλεγεί και εδώ η τιμή 42.

Parameter	Before Tuning	After Tuning
n_estimators	100	700
max_depth	10	15
min_samples_split	2	4
min_samples_leaf	2	2
max_features	log2	sqrt
bootstrap	False	True

Πίνακας 3: RandomForest Parameter Settings Before and After Tuning

	Before Tuning	After Tuning
	Mean squared error	Mean squared error
Motorbikes	2380.952	2265.092
Vans	8345.571	8184.899
Passenger Cars	11551.75	10923.639
Agricultural Vehicles	282.682	267.008

Πίνακας 4: Random Forest Parameter Tuning Results

5.3 NeuralNetwork

Το νευρωνικό δίκτυο δέχεται δύο εισόδους: αριθμητικά χαρακτηριστικά και κατηγορικά χαρακτηριστικά. Τα αριθμητικά χαρακτηριστικά χρησιμοποιούνται ως είναι, ενώ τα κατηγορικά ενσωματώνονται μέσω ενός embedding layer πριν συνδεθούν με τα αριθμητικά χαρακτηριστικά στα πλήρως συνδεδεμένα επίπεδα (fully connected layers). Για κάθε κατηγορική στήλη, υπάρχει ένα επίπεδο ενσωμάτωσης που μετατρέπει κάθε κατηγορική τιμή σε ένα διάνυσμα μικρής διάστασης. Η διάσταση αυτού του διανύσματος καθορίζεται από τον αριθμό των μοναδικών τιμών στη στήλη. Υπάρχουν έξι πλήρως

συνδεδεμένα επίπεδα με μειώσεις των διαστάσεων (από 256 σε 16). Ανάμεσα στα πλήρως συνδεδεμένα επίπεδα υπάρχουν επίσης επίπεδα απόρριψης (dropout) που βοηθούν στη μείωση της υπερπροσαρμογής (overfitting). Τα πλήρως συνδεδεμένα επίπεδα χρησιμοποιούν τη συνάρτηση ενεργοποίησης ReLU για τη μη γραμμική μετατροπή των εξόδων. Το μοντέλο εκπαιδεύτηκε με τον αλγόριθμο βελτιστοποίησης Adam και έχει ως συνάρτηση απώλειας το Mean Squared Error. Χρησιμοποιήθηκαν batch size = 64 και epochs = 10.

6 Οπτικοποίηση και αξιολόγηση αποτελεσμάτων

6.1 Παρουσίαση των αποτελεσμάτων

Ως μέτρα για την αξιολόγηση των μοντέλων χρησιμοποιήσαμε το mean squared error, το absolute error και το ποσοστό των προβλέψεων που η τιμή τους ήταν εντός ενός ποσοστού της πραγματικής τιμής. Τα αποτελέσματα για κάθε μοντέλο όταν έχει εκπαιδευθεί στο σύνολο των δεδομένων φαίνονται στον πίνακα 5 και στα διαγράμματα 8, 9, 10. Από αυτήν την αξιολόγηση φαίνεται ότι το πιο ακριβές μοντέλο με εκπαίδευση στο σύνολο των δεδομένων είναι το XGBoost.

Επίσης παρατηρήσαμε ότι υπάρχουν στα δεδομένα μας 4 διαφορετικές κατηγορίες οχημάτων. [Παρατήρηση 1] Για τα μοντέλα του Random Forest και XGBoost που έχουν εκπαιδευθεί στο σύνολο των δεδομένων κάναμε ξεχωριστή αξιολόγηση ανα κατηγορία [Πίνακες 6, 7 και διαγράμματα 13, 14]. Ακόμη κάναμε ανάλυση για την σημασία των μεταβλητών σε αυτά τα μοντέλα [Πίνακες 11, 12].

	Mean squared error	Absolute error	Percentage within 10%
XGBoost	10370.87	66.19	34.87
Random Forest	10852.02	67.38	33.36
Neural network	11689.55	70.16	20.00

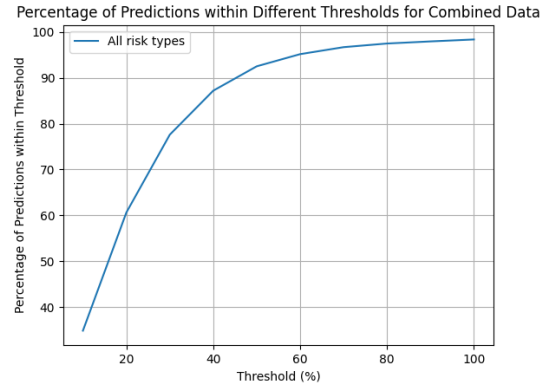
Πίνακας 5: Αξιολόγηση των μοντέλων

Παρατηρήσαμε ότι το Type risk παίζει πολύ σημαντικό ρόλο για το XGBoost και σημαντικό ρόλο στο Random Forest καθώς και ότι υπήρχε πολύ χαμηλή ακρίβεια στις μικρότερες κατηγορίες οχημάτων. Έτσι αποφασίσαμε να εκπαιδεύσουμε τα μοντέλα ξεχωριστά για κάθε κατηγορία οχήματος. Αυτό βελτίωσε αρκετά την ακρίβεια του XGBoost [πίνακας 8, διάγραμμα 15], λίγο την ακρίβεια του random forest [πίνακας 9, διάγραμμα 16] ενώ η ακρίβεια του νευρωνικού μειώθηκε [πίνακας 10, διάγραμμα 17].

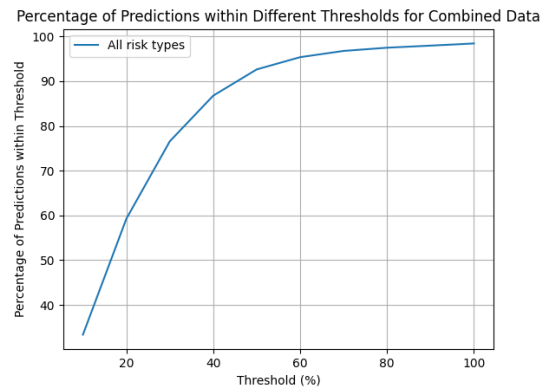
Πραγματοποιήσαμε και μελέτη της σημαντικότητας των features στο XGBoost [διάγραμμα 18] και στο Random Forest [διάγραμμα 19]

Απο την παραπάνω ανάλυση προκύπτει ότι το καλύτερο μας μοντέλο είναι το **XGBoost** όταν εκπαιδευθεί ξεχωριστά για κάθε κατηγορία οχήματος. Το 34.68 % των προβλεπόμενων τιμών Premium είναι εντός 10 % της πραγματικής τιμής τους. [διάγραμμα 20]

6.2 Σχολιασμός των αποτελεσμάτων



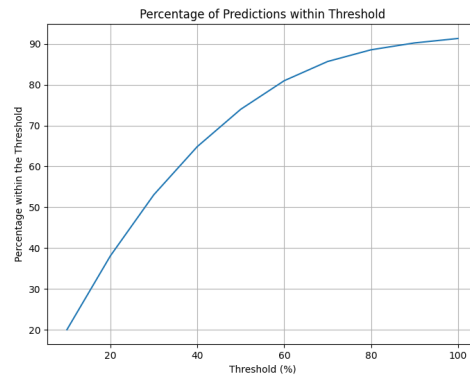
Εικ. 8: Μοντέλο XGBoost που εκπαιδεύτηκε στο σύνολο των δεδομένων



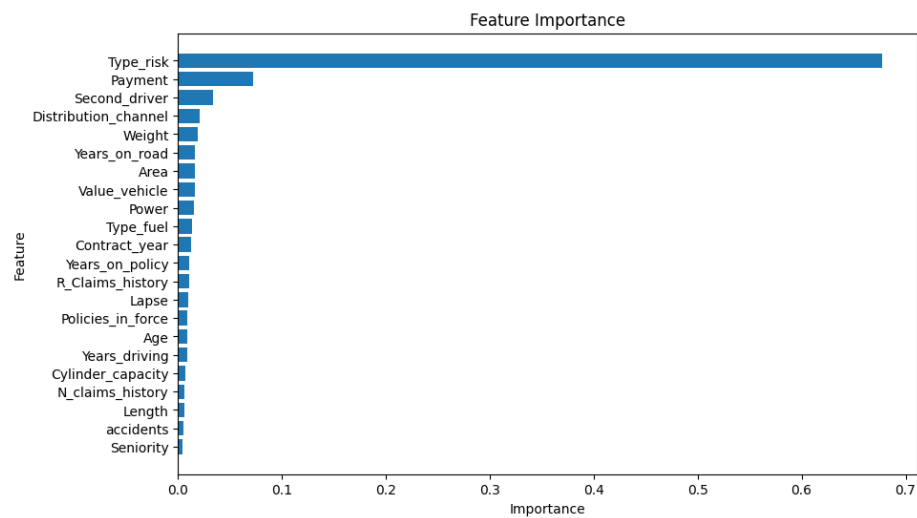
Εικ. 9: Μοντέλο Random Forest που εκπαιδεύτηκε στο σύνολο των δεδομένων

	Mean squared error	Absolute error	Percentage within 10%
Μοτοσυκλέτα	2358.08	30.92	35.25
Βανάκι	8266.77	61.80	37.50
Επιβατικό όχημα	11574.03	70.76	34.50
Αγροτικό όχημα	2348.79	27.96	26.92

Πίνακας 6: Αξιολόγηση XGBoost που εκπαιδεύτηκε στο σύνολο των δεδομένων ανα κατηγορία



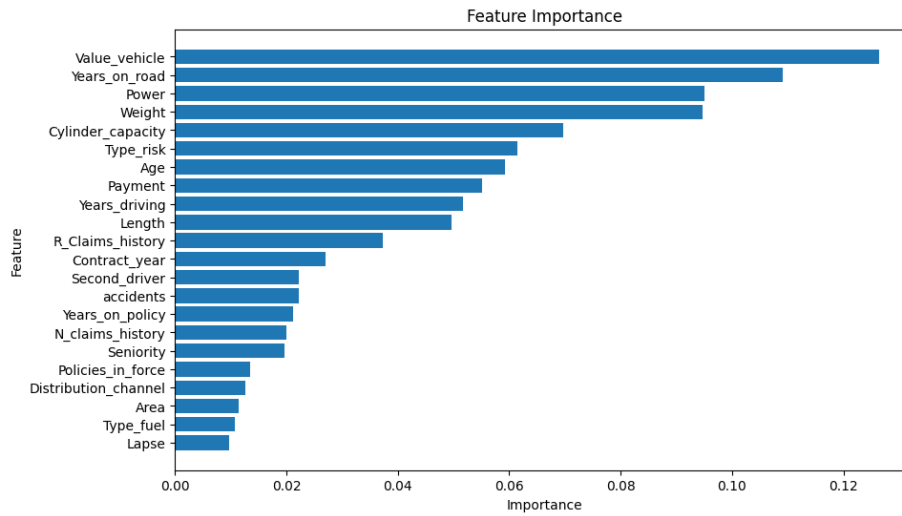
Εικ. 10: Μοντέλο Νευρωνικού δικτύου που εκπαιδεύτηκε στο σύνολο των δεδομένων



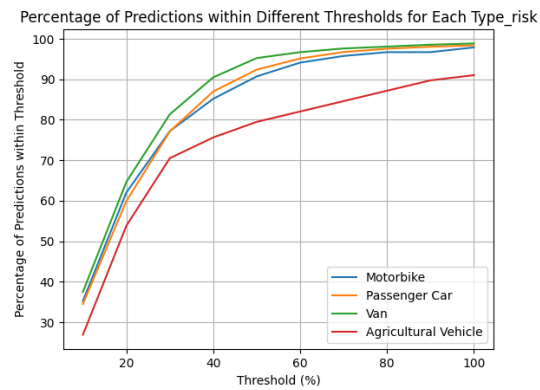
Εικ. 11: Σημασία χαρακτηριστικών σε μοντέλο XGBoost που εκπαιδεύτηκε στο σύνολο των δεδομένων

	Mean squared error	Absolute error	Percentage within 10%
Μοτοσυκλέτα	2229.48	29.05	36.54
Βανάκι	8586.59	62.96	36.57
Επιβατικό όχημα	12159.72	72.34	32.58
Αγροτικό όχημα	829.67	20.90	29.48

Πίνακας 7: Αξιολόγηση Random Forest που εκπαιδεύτηκε στο σύνολο των δεδομένων ανα κατηγορία



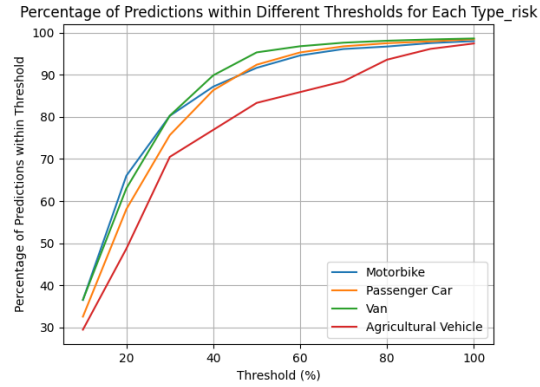
Εικ. 12: Σημασία χαρακτηριστικών σε μοντέλο Random Forest που εκπαιδεύτηκε στο σύνολο των δεδομένων



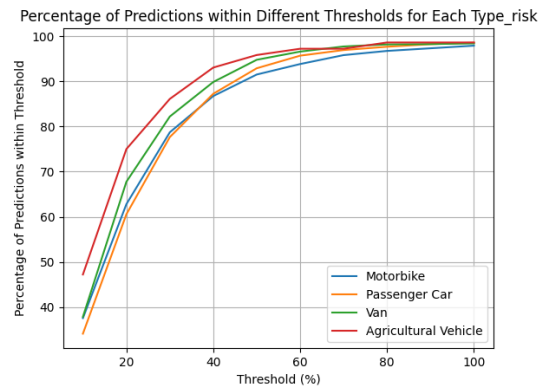
Εικ. 13: Ξεχωριστή αξιολόγηση ανα κατηγορία σε μοντέλο XGBoost που εκπαιδεύτηκε στο σύνολο των δεδομένων

	Mean squared error	Absolute error	Percentage within 10%
Μοτοσυκλέτα	2228.74	29.77	38.13
Βανάκι	7846.43	60.34	38.07
Επιβατικό όχημα	10693.59	69.07	33.66
Αγροτικό όχημα	311.89	12.15	51.38

Πίνακας 8: Αξιολόγηση XGBoost που εκπαιδεύτηκε ξεχωριστά σε κάθε κατηγορία



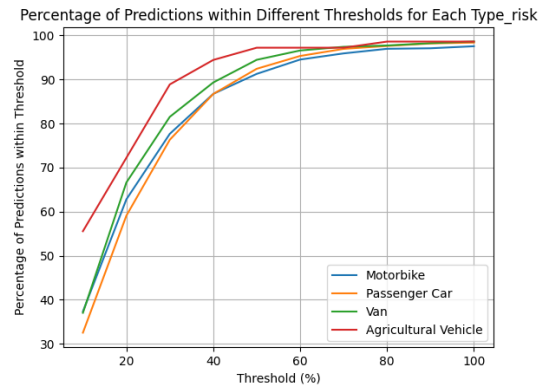
Εικ. 14: Ξεχωριστή αξιολόγηση ανα κατηγορία σε μοντέλο Random Forest που εκπαιδεύτηκε στο σύνολο των δεδομένων



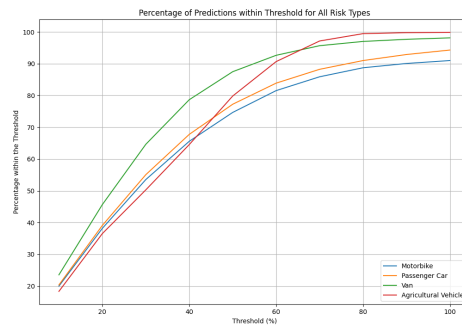
Εικ. 15: Μοντέλο XGBoost που εκπαιδεύτηκε ξεχωριστά για κάθε κατηγορία

	Mean squared error	Absolute error	Percentage within 10%
Μοτοσυκλέτα	2265.09	29.42	37.32
Βανάκι	8184.89	61.39	37.01
Επιβατικό όχημα	10923.63	70.22	32.51
Αγροτικό όχημα	267.00	11.47	55.55

Πίνακας 9: Αξιολόγηση Random Forest που εκπαιδεύτηκε ξεχωριστά σε κάθε κατηγορία



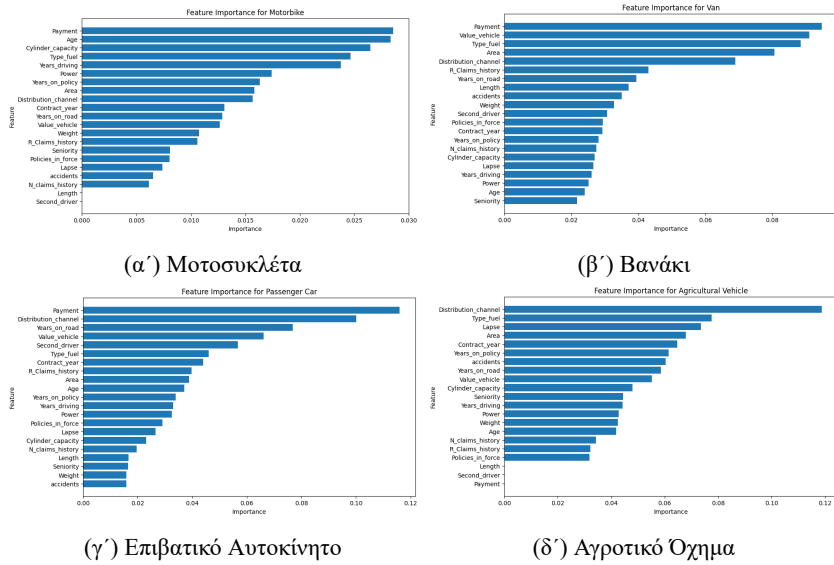
Εικ. 16: Μοντέλο Random Forest που εκπαιδεύτηκε ξεχωριστά για κάθε κατηγορία



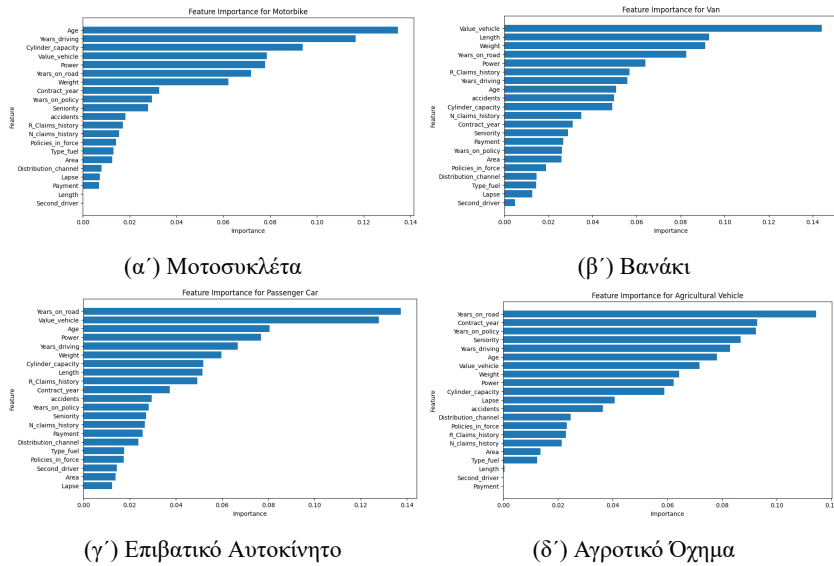
Εικ. 17: Μοντέλο Νευρωνικού δικτύου που εκπαιδεύτηκε ξεχωριστά για κάθε κατηγορία

	Mean squared error	Absolute error	Percentage within 10%
Μοτοσυκλέτα	3391.05	36.22	20.46
Βανάκι	8966.14	62.96	23.95
Επιβατικό όχημα	11465.31	71.66	22.41
Αγροτικό όχημα	1325.41	29.64	12.73

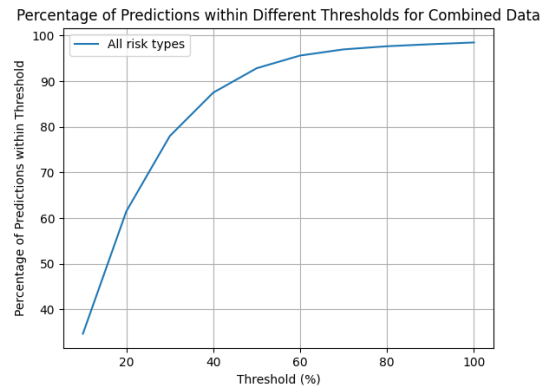
Πίνακας 10: Αξιολόγηση Νευρωνικού δικτύου ανα κατηγορία



Εικ. 18: Σημασία χαρακτηριστικών σε μοντέλο XGBoost που εκπαιδεύτηκε ξεχωριστά ανα κατηγορία



Εικ. 19: Σημασία χαρακτηριστικών σε μοντέλο Random Forest που εκπαιδεύτηκε ξεχωριστά ανα κατηγορία



Εικ. 20: Μοντέλο XGBoost που εκπαιδεύτηκε ξεχωριστά για κάθε κατηγορία συγκεκριμένα αποτελέσματα

Παρατηρούμε ότι ακόμα και το καλύτερο μοντέλο μας δεν πετυχαίνει την απολύτη ακρίβεια αν και όπως φαίνεται στο διάγραμμα σχεδόν το 90 % των προβλέψεων είναι εντός του 40 % της πραγματικής τιμής ενώ λιγότερες από 3 % των προβλέψεων έχουν ξεφύγει πάνω από 80 % της πραγματικής τιμής. Αυτή η αποκλίση μπορεί να οφείλεται σε έλλειψη πληροφορίας από το δεδομένα μας αφού ενδέχεται η ασφαλιστική εταιρεία να μην δημοσίευσε όλες της παραμέτρους που χρησιμοποιεί για να χρεώσει τους πελάτες της (π.χ αν έκανε επιπλέον εκπτώσεις για να κερδίσει μερίδιο στην αγορά)

Σχετικά με την σημαντικότητα των χαρακτηριστικών είναι λογικό που προκύπτει ότι ο τρόπος πληρωμής είναι το πιο σημαντικό χαρακτηριστικό στις 3 από τις 4 κατηγορίες αφού η εξόφληση του πόσου για όλο τον χρόνο παρέχει μεγαλύτερη σταθερότητα στα έσοδα της ασφαλιστικής οπότε η τιμή είναι καλύτερη. [Διάγραμμα 4] Παρατηρούμε επίσης ότι μεγάλη σημασία έχει και ο τρόπος που αγοράστηκε η ασφάλεια αφού όσοι αγόρασαν την ασφάλεια κατευθείαν από την εταιρία έχουν καλύτερη τιμή από όσους την αγόρασαν από τρίτους. [Διάγραμμα 5] Σημαντική είναι και η συνολική αξία του οχήματος αφού όσο πιο ακριβό είναι τόσο πιο πολύ κοστίζει η επισκευή. Παρατηρούμε ότι αυτό είναι σημαντικότερος παράγοντας απόφασης στις κατηγορίες με μεγαλύτερο εύρος αξιών. Για τις μοτοσυκλέτες είναι λογικό τα πιο σημαντικά χαρακτηριστικά να είναι ο κύβισμος τους και η ηλικία του οδηγού τους αφού οι νέοι οδηγοί και η μηχανές με υψηλό κύβισμο προκαλούν πιο πολλά ατυχήματα. Για τα επιβατικά αυτοκίνητα η ηλικία του αυτοκινήτου είναι σημαντική γιατί τα παλαιότερα αυτοκίνητα είναι πιο πιθανό να έχουν βλάβες. Αξιοσημείωτο είναι και ότι το ιστορικό ατυχημάτων και το πόσα χρόνια έχει κάποιος την ίδια ασφαλιστική πολιτική δεν επηρεάζουν πολύ την τελική τιμή σε σχέση με άλλες παραμέτρους.

7 Οδηγίες εγκατάστασης

7.1 Εγκατάσταση της python και δημιουργία virtual environment

Για να μπορέσει να λειτουργήσει η εφαρμογή χρειάζεται αρχικά εγκατάσταση της Python 3.12.2 και μετά την δημιουργία ενός virtual environment ώστε να εγκατασταθούν εκεί οι βιβλιοθήκες της python και να μην υπάρχουν συγκρούσεις που προκύπτουν από διαφορά εκδόσεων. Για την δημιουργία του virtual environment δημιουργούμε ένα φάκελο μέσα στον οποίο θα εγκατασταθεί το περιβάλλον και ότι άλλο χρειαστούμε για την εφαρμογή (π.χ. ονομάζεται app) και ανοίγουμε ένα τερματικό στα Windows και πλοηγούμαστε εκεί. Με την εντολή

```
python3 -m venv virt
```

δημιουργούμε το περιβάλλον και με την εντολή

```
virt\Scripts\activate
```

το ενεργοποιούμε. Με αυτή την εντολή αναβαθμίζουμε και το pip

```
python.exe -m pip install --upgrade pip
```

7.2 Εγκατάσταση των βιβλιοθηκών απαραίτητων για την εφαρμογή

Η εφαρμογή μας για το γραφικό περιβάλλον χρησιμοποιεί kivy το οποίο εγκαθιστούμε με τις εντολές:

```
git clone https://github.com/kivymd/KivyMD.git --depth 1
cd KivyMD
pip install .
```

Για να εγκατασταθούν οι υπόλοιπες απαραίτητες βιβλιοθήκες οι εντολές είναι:

```
pip install numpy==1.26.4
pip install pandas==2.2.2
pip install joblib==1.4.2
pip install xgboost==2.0.3
```

Για να τρέξει η εφαρμογή πλοηγούμαστε στον φάκελο με τα αρχεία ui και xgboostModels χρησιμοποιούμε την εντολή:

```
python ui\front.py
```

και ακολουθούμε τις οδηγίες χρήσης στην επόμενη ενότητα.

7.3 Εγκατάσταση των βιβλιοθηκών απαραίτητων για τα python scripts και τα γραφήματα

Σε περίπτωση που θέλετε να τρέξετε εκτός από την εφαρμογή και τα scripts με τα μοντέλα μας πρέπει να τρέξετε τις εντολές:

```
pip install torch==2.3.0
pip install scikit-learn==1.4.2
pip install matplotlib==3.8.4
```

Μετά την εγκατάσταση μπορείτε να τρέξετε οποιοδήποτε από τα script και αν δείτε τα διαγράμματα

8 User Interface και Οδηγίες Χρήσης

Το user interface της εφαρμογής αναπτύχθηκε με χρήση της Python και ειδικότερα του framework Kivy, καθώς και της συλλογής από γραφικά στοιχεία KivyMD. Στην εφαρμογή μας υπάρχουν πέντε διαφορετικές "οθόνες":

1. Η οθόνη του login
2. Η οθόνη συμπλήρωσης στοιχείων του πελάτη που πρόκειται να ασφαλίσει το όχημά του.
3. Η οθόνη συμπλήρωσης στοιχείων του οχήματος που πρόκειται να ασφαλιστεί.
4. Η οθόνη συμπλήρωσης στοιχείων παλαιότερων συμβολαίων που είχε ο πελάτης στην εταιρεία.
5. Η οθόνη παρουσίασης της προτεινόμενης ετήσιας τιμής χρέωσης του πελάτη με βάση τα στοιχεία που συμπληρώθηκαν.

8.1 Login

Η οθόνη του login είναι η αρχική οθόνη της εφαρμογής [Εικ. 21] στην οποία ο χρήστης - υπάλληλος της εταιρείας θα πρέπει να συμπληρώσει τα σωστά στοιχεία σύνδεσής του (username και password) και στην συνέχεια να πατήσει το κουμπί "LOG IN". Σε περίπτωση που ένα από τα δύο πεδία μείνει κενό, ο χρήστης θα λάβει το αντίστοιχο μήνυμα λάθους [Εικ. 22α']. Σε περίπτωση που τα στοιχεία σύνδεσης δεν είναι σωστά, ο χρήστης θα λάβει διαφορετικό μήνυμα λάθους [Εικ. 22β'].

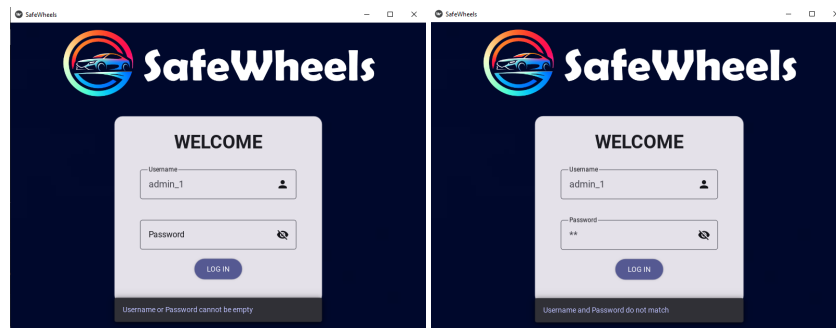
Έχει προβλεφθεί και δημιουργηθεί μόνο ένας λογαριασμός υπαλλήλου. Έτσι, για να πάμε στην επόμενη οθόνη θα πρέπει να συμπληρώσουμε στο πεδίο username: **admin_1** και στο πεδίο password: **12345**. Αφού συμπληρώσουμε αυτά τα στοιχεία σωστά και πατήσουμε το κουμπί "LOG IN" μεταφερόμαστε στην δεύτερη οθόνη [Εικ. 23].

8.2 Στοιχεία πελάτη

Στην οθόνη συμπλήρωσης στοιχείων του πελάτη [Εικ. 23], ο χρήστης-υπάλληλος της εταιρείας θα πρέπει να συμπληρώσει τα στοιχεία του ανθρώπου που ενδιαφέρεται να ασφαλίσει το όχημά του. Ειδικότερα, υπάρχουν δύο πεδία επιλογής ημερομηνίας ("Date



Εικ. 21: Οθόνη login



(α') Κενό πεδίο

(β') Λάθος στοιχεία

Εικ. 22: Μηνύματα λάθους στο login

of Birth”, ”License Issue Date”), τα οποία όταν επιλεγθούν εμφανίζεται ένα ημερολόγιο [Εικ. 24α'] προκειμένου να επιλεγθεί η κατάλληλη ημερομηνία. Πατώντας το πεδίο ”Area” ο χρήστης βλέπει ένα μενού δύο επιλογών [Εικ. 24β'] από τις οποίες θα πρέπει να επιλέξει μία. Στο πεδίο ”Seniority” ο χρήστης θα πρέπει να πληκτρολογήσει έναν αριθμό που προσδιορίζει τα χρόνια που ο συγκεκριμένος πελάτης είναι ασφαλισμένος στην εταιρεία.

Αφού συμπληρωθούν όλα τα πεδία ο χρήστης θα πρέπει να πατήσει το κουμπί ”NEXT” για να μεταφερθεί στην επόμενη οθόνη [Εικ. 25]. Σε περίπτωση που επιθυμεί να πάει στην οθόνη του login [Εικ. 21] θα πρέπει να πατήσει το κουμπί ”GO BACK”.

Εικ. 23: Οθόνη συμπλήρωσης στοιχείων πελάτη

(α') Ημερολόγιο

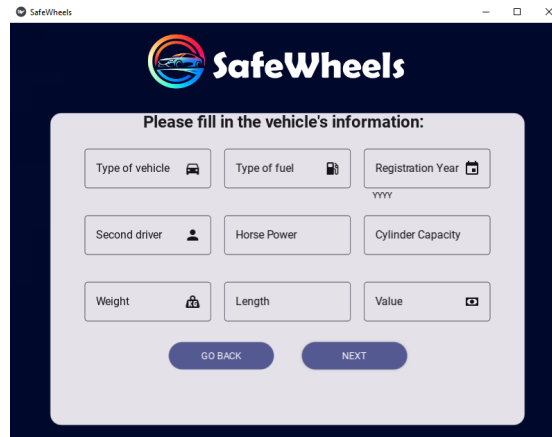
(β') Μενού επιλογών

Εικ. 24: Παραδείγματα εισόδου

8.3 Στοιχεία οχήματος

Σε αυτήν την οθόνη [Εικ. 25], ο χρήστης-υπάλληλος της εταιρείας θα πρέπει να συμπληρώσει τα στοιχεία του οχήματος το οποίο επιθυμεί να ασφαλίσει ο πελάτης. Ειδικότερα, για τα πεδία "Type of vehicle", "Type of fuel", "Second driver" ο χρήστης θα πρέπει να πατήσει καθένα από αυτά τα πεδία και να επιλέξει μία από τις επιλογές που εμφανίζονται στο κάθε μενού επιλογών (ενδεικτικά [Εικ. 26α']). Στα υπόλοιπα πεδία ο χρήστης θα πρέπει να πληκτρολογήσει έναν αριθμό σύμφωνα με τις υποδείξεις του κάθε πεδίου (ενδεικτικά [Εικ. 26β']).

Για να μεταβεί στην επόμενη οθόνη [Εικ. 27] θα πρέπει να πατήσει το κουμπί "NEXT" ενώ για να πάει στην προηγούμενη οθόνη [Εικ. 23] συμπλήρωσης των στοιχείων του πελάτη θα πρέπει να πατήσει το κουμπί "GO BACK".



The screenshot shows a web application window titled "SafeWheels". The main heading is "Please fill in the vehicle's information:". Below this, there are nine input fields arranged in a grid. The first row contains "Type of vehicle" (with a car icon), "Type of fuel" (with a fuel pump icon), and "Registration Year" (with a calendar icon and a "YYYY" placeholder). The second row contains "Second driver" (with a person icon), "Horse Power", and "Cylinder Capacity". The third row contains "Weight" (with a scale icon), "Length", and "Value" (with a currency icon). At the bottom of the form are two buttons: "GO BACK" and "NEXT".

Εικ. 25: Οθόνη συμπλήρωσης στοιχείων οχήματος προς ασφάλιση

(α') Παράδειγμα μενού επιλογών

(β') Παράδειγμα συμπλήρωσης πεδίου

Εικ. 26: Παραδείγματα εισόδου

8.4 Στοιχεία συμβολαίου

Στην οθόνη συμπλήρωσης στοιχείων συμβολαίου [Εικ. 27] ο χρήστης-υπάλληλος της εταιρείας θα πρέπει να συμπληρώσει στοιχεία που αφορούν παλαιότερα συμβόλαια που είχε ο πελάτης στην εταιρεία. Υπάρχουν τρία πεδία επιλογής ημερομηνίας ("Start Contract", "Last renewal", "Next renewal"), τα οποία όταν επιλεγθούν εμφανίζεται ημερολόγιο (ενδεικτικά [Εικ. 28α']). Τα πεδία "Distribution Channel" και "Payment" είναι πεδία που κατά την επιλογή τους εμφανίζεται μενού επιλογών (ενδεικτικά [Εικ. 28β']), ενώ τα υπόλοιπα πρέπει να συμπληρωθούν με πληκτρολόγηση σύμφωνα με τις υποδείξεις των πεδίων.

Για την μετάβαση στην προηγούμενη οθόνη [Εικ. 25] ο χρήστης θα πρέπει να πατήσει το κουμπί "GO BACK". Διαφορετικά, βρισκόμαστε στο σημείο που έχουν συμπληρωθεί όλα τα στοιχεία και ο χρήστης θα πρέπει να πατήσει το κουμπί "CALCULATE" προκειμένου να μεταβεί στην τελευταία οθόνη [Εικ. 29].

The screenshot shows the SafeWheels application window with a dark blue header containing the logo and name. Below the header, a light gray box contains the text "Please fill in the contract information:". Inside this box, there are nine input fields arranged in a 3x3 grid. Each field has a label and a calendar icon: "Start Contract", "Last renewal", "Next renewal", "Distribution Channel", "Payment", "Claims", "Policies", "Lapse", and "Ratio claims". Below the grid are two buttons: "GO BACK" and "CALCULATE".

Εικ. 27: Οθόνη συμπλήρωσης στοιχείων οχήματος προς ασφάλιση

The left screenshot (α') shows the same contract information form as in Εικ. 27, but with a calendar popup open over the "Start Contract" field. The calendar displays the month of May 2024, with the 10th selected. The right screenshot (β') shows the same form with dropdown menus open for the "Payment" and "Policies" fields. The "Payment" dropdown shows "Half-yearly" and "Annual" options, and the "Policies" dropdown shows "Annual" and "Ratio claims" options.

(α') Παράδειγμα ημερολογίου

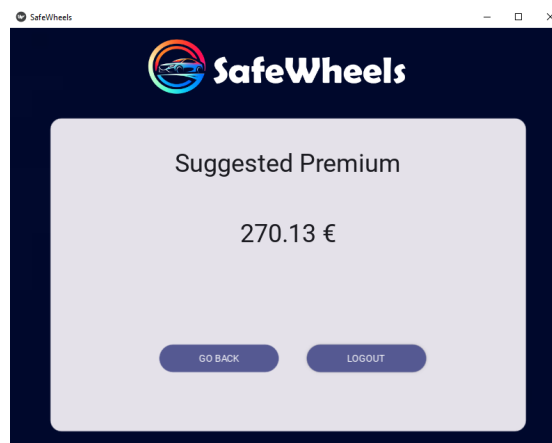
(β') Παράδειγμα μενού επιλογών

Εικ. 28: Παραδείγματα εισόδου

8.5 Οθόνη προτεινόμενης χρέωσης

Σε αυτήν την οθόνη [Εικ. 29], ο χρήστης-υπάλληλος της εταιρείας βλέπει την προτεινόμενη τιμή ασφάλισης του οχήματος που πρόκειται να ασφαλίσει ο πελάτης. Πατώντας το κουμπί "GO BACK" μπορεί να μεταβεί στην προηγούμενη οθόνη [Εικ. 27], ενώ πατώντας το κουμπί "LOGOUT" μεταβαίνει στην αρχική οθόνη της εφαρμογής [Εικ. 21].

Αξίζει να σημειωθεί πως ο χρήστης μπορεί να πηγαίνει προς τα πίσω και να αλλάζει στοιχεία ανά πάσα στιγμή. Κάθε φορά που πατά το κουμπί "CALCULATE" της προτελευταίας οθόνης [Εικ. 27], όλα τα ανανεωμένα στοιχεία δίνονται στο μοντέλο προκειμένου να βγει η προτεινόμενη τιμή χρέωσης.



Εικ. 29: Οθόνη παρουσίασης προτεινόμενης τιμής χρέωσης

8.6 Γενικές παρατηρήσεις για το ui

Ο χρήστης μπορεί να αφήσει κάποιο ή και όλα τα πεδία κενά. Σε αυτήν την περίπτωση για τα αριθμητικά πεδία θα συμπληρωθεί η μέση τιμή (median) όλων των τιμών των δεδομένων, ενώ για τα κατηγορικά πεδία θα συμπληρωθεί η πιο συχνή κατηγορία. Σε περίπτωση που η ημερομηνία έναρξης συμβολαίου ή τελευταίας ανανέωσης συμβολαίου μείνει κενή θα συμπληρωθεί η τρέχουσα ημερομηνία. Σε περίπτωση που η ημερομηνία επόμενης ανανέωσης συμβολαίου μείνει κενή θα συμπληρωθεί η ημερομηνία ένα χρόνο μετά από την ημερομηνία τελευταίας ανανέωσης.