

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΣΧΟΛΗ ΕΠΙΣΤΗΜΗΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ**



Τίτλος:
**Μοντέλα μηχανικής μάθησης και εφαρμογές για την αξιολόγηση
δανειοληπτικού ρίσκου**

Διπλωματική Εργασία

Λαβδός Ιωάννης

Επιβλέπον Διδάσκον: Έξαρχος Θέμης

Επιτροπή: Κερμανίδου Κάτια Λήδα, Κουρουθανάσης Παναγιώτης

ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη	3
Εισαγωγή	3
Ερευνητικός χώρος	4
Περιγραφή Δεδομένων	30
Αναλυτική περιγραφή δεδομένων	31
Προ-επεξεργασία δεδομένων	33
Πειράματα μάθησης	36
Αποτελέσματα αλγορίθμων	39
Logistic Regression	39
Decision Tree	39
Random Forest	40
K- Nearest Neighbor	40
Naive Bayes	40
Αναλυτική περιγραφή μοντέλου	41
Εφαρμογή	43
Συμπεράσματα	49

Περίληψη

Τα τελευταία χρόνια, με την ανάπτυξη της τεχνολογίας, η χρήση της τεχνητής νοημοσύνης και κατά προέκταση της μηχανικής μάθησης έχει αυξηθεί και εισέλθει στην καθημερινότητα όλων. Ως εκ τούτου, είναι αδύνατον οι τράπεζες να μη χρησιμοποιούν τις συγκεκριμένες τεχνολογίες. Αυτό συμβαίνει διότι, σε μία τράπεζα, η χρήση αλγορίθμων μηχανικής μάθησης μπορεί να χρησιμοποιηθεί σε πολλούς τομείς, από την καλύτερη εξυπηρέτηση πελατών μέχρι και τη λήψη αποφάσεων από ηγετικά στελέχη. Ένας εξίσου όμως σημαντικός κλάδος είναι και αυτός των δανείων, ο οποίος αποτελεί το οξυγόνο των εμπορικών τραπεζών. Η μηχανική μάθηση δίνει πλέον τη δυνατότητα στις τράπεζες να αυτοματοποιήσουν τη διαδικασία έγκρισης ή μη ενός δανείου σε έναν πελάτη. Ομοίως και στη συγκεκριμένη έρευνα χρησιμοποιήθηκαν διάφοροι αλγόριθμοι με σκοπό την έγκριση ή μη ενός δανείου. Επικρατέστερος ήταν ο “Random Forest”, ο οποίος εφαρμόστηκε με τη χρήση της “Python” και των κατάλληλων βιβλιοθηκών της.

Εισαγωγή

Το Credit Scoring¹ (Πιστοληπτική Ικανότητα) χρησιμοποιείται πολύ συχνά ως κύριο εργαλείο για την διαχείριση των κινδύνων χρηματοδοτικών ιδρυμάτων. Πιο συγκεκριμένα, αντιπροσωπεύει ένα σύνολο από μοντέλα αποφάσεων που βασίζονται σε τεχνικές εξόρυξης δεδομένων καθώς και αλγορίθμων που περιλαμβάνουν την εξαγωγή προτύπων και πληροφοριών που χρησιμοποιούνται για την πρόβλεψη έγκρισης ή μη μιας αίτησης δανείου. Η ανάπτυξη αυτού του εργαλείου προέκυψε λόγω κακών αποφάσεων που βασίζονταν σε λάθη ή στην έλλειψη εμπειρίας του προσωπικού.

Ένα μοντέλο Credit Scoring εξαρτάται από ένα σύνολο πιστωτικών παραγόντων, οι πιο συνηθισμένοι από τους οποίους είναι η ηλικία του πελάτη, το ιστορικό πληρωμής, η χρήση της κάρτας και ο αριθμός λογαριασμών. Οι πιστωτικοί παράγοντες για ένα ενυπόθηκο δάνειο, μπορεί να περιλαμβάνουν το ιστορικό

¹ Nalic, J., & Svraka, A. (2018). Using data mining approaches to build credit scoring model: Case study — Implementation of credit scoring model in microfinance institution. 2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH).

απασχόλησης, την προκαταβολή και το μέγεθος του δανείου. Επομένως, μετά τη χρήση του μοντέλου εξάγονται αποτελέσματα διάκρισης μεταξύ “καλών” και “κακών” δανείων ως προς την εκάστοτε τράπεζα.

Αυτού του είδους τα μοντέλα αναπτύσσονται από τις τράπεζες ώστε να ενισχύσουν τη διαδικασία αξιολόγησης της πιστοληπτικής τους ικανότητας και εκχωρούν πιστωτικούς κινδύνους. Για την ακρίβεια, πρόκειται για μια εκτίμηση πιθανής ανεπιθύμητης συμπεριφοράς ενός πελάτη στο μέλλον.

Ερευνητικός χώρος

Τα τελευταία χρόνια, υπάρχει τεράστια ανάγκη στην αγορά για αξιόπιστα μοντέλα πιστοληπτικής ικανότητας, καθώς μειώνουν τόσο το κόστος όσο και το λειτουργικό κίνδυνο. Αυτό έχει ως αποτέλεσμα την ανάπτυξη καλύτερων τεχνικών και μοντέλων, πολλές φορές βασισμένων σε ήδη υπάρχοντες αλγόριθμους, από τους επιστήμονες των δεδομένων που δραστηριοποιούνται στον ερευνητικό χώρο. Υπάρχουν αρκετές εργασίες που έχουν αναπτυχθεί πάνω στο θέμα του Credit Scoring, οι οποίες είναι άξιες προσοχής.

Πιο αναλυτικά, χρησιμοποιούνται πολλές τεχνικές μηχανικής μάθησης για την ανάπτυξη ενός μοντέλου πιστοληπτικής ικανότητας. Σε μία έρευνα² χρησιμοποιήθηκε μοντέλο ταξινόμησης με βάση ιστορικά πιστωτικά δεδομένα. Το σύνολο των δεδομένων περιείχε 74 χαρακτηριστικά, από τα οποία ένα ήταν εξαρτημένο ενώ τα άλλα 73 ήταν ανεξάρτητα. Ο συνολικός αριθμός των εγγραφών που χρησιμοποιήθηκαν ήταν 10.000. Για την ομαλοποίηση των δεδομένων χρησιμοποιήθηκε η τεχνική “Min-Max” και ο αλγόριθμος που είχε τα καλύτερα αποτελέσματα ήταν ο K-NN (K-Nearest Neighbor). Δοκιμάστηκαν αρκετές αναλογίες δεδομένων όσον αφορά την εκπαίδευση και την αξιολόγηση. Σε αυτά, υπολογίστηκε το Accuracy, RMSE, Correlation και το Error Rate. Τα αποτελέσματα τους φαίνονται στο πίνακα που ακολουθεί:

² Arutjothi, G., & Senthamarai, C. (2017). Prediction of loan status in commercial bank using machine learning classifier. 2017 International Conference on Intelligent Sustainable Systems (ICISS).

Πίνακας 1: Αποτελέσματα K-nearest Neighbor

K-nearest Neighbor model (k=30)				
Percentage Split	Accuracy	RMSE	Correlation	Error Rate
80% :20%	74.5	2.04	0.99	0.256
70% :30%	73.4	1.91	0.98	0.259
60% :40%	74.9	2.29	0.92	0.25
50% :50%	75.08	2.41	-0.26	0.24

Όπως φαίνεται και από τα αποτελέσματα, επιλέχθηκε το μοντέλο με αναλογία 50% με Accuracy 75.08%. Το μοντέλο αυτό υλοποιήθηκε με τη γλώσσα R και απευθύνεται σε εμπορικές τράπεζες. Επίσης, δοκιμάστηκαν αρκετοί άλλοι αλγόριθμοι, όπως οι Decision Tree, Support Vector Machine και Neural Network.

Σε άλλη εργασία³, εξετάζεται ένα πείραμα ώστε να αξιολογηθεί ο αλγόριθμος ταξινόμησης Naïve Bayes. Συγκεκριμένα, υπογραμμίζεται πως δεν εμφανίζεται συχνά σε προβλήματα σχετιζόμενα με πιστοληπτικά δεδομένα. Τα χαρακτηριστικά του συνόλου δεδομένων φαίνονται στο παρακάτω πίνακα:

Πίνακας 2: Χαρακτηριστικά του συνόλου δεδομένων

Name	Description	Role
Gender	Gender (Male, Female)	Input
Age	Age (in years)	Input
Tribe	Tribe/Sector	Input
Marrystat	Marital status	Input
Consistuecy	Consistuecy/Address	Input
YearsAtRes	Years at resident	Input
Status	Credit status (Good, Bad)	Target
ValueOfCars	Value of Cars	Input
ValueOfHouse	Value of House	Input
ValueOfLand	Value of Land	Input
EduLevel	Educational level	Input

³ Okesola, O. J., Okokpujie, K. O., Adewale, A. A., John, S. N., & Omoruyi, O. (2017). An Improved Bank Credit Scoring Model: A Naïve Bayesian Approach. 2017 International Conference on Computational Science and Computational Intelligence (CSCI).

Ο αλγόριθμος αναπτύχθηκε πάνω σε ιστορικά δεδομένα πελατών. Όπως έχει αναφερθεί, οι ερευνητές προσπάθησαν να προσεγγίσουν το πρόβλημα με τον αλγόριθμο του Naive Bayes, που το τελικό αποτέλεσμα υπολογίζεται από τον τύπο

$$P(X/H) = \frac{P(X/H)P(H)}{P(X)} . \text{ Τα δεδομένα χωρίστηκαν σε δυο κατηγορίες: σε}$$

δεδομένα εκπαίδευσης και σε δεδομένα αξιολόγησης όπως φαίνεται και στον επισυνημμένο πίνακα.

Πίνακας 3: Αριθμός δεδομένων

Data	Good	Bad	Total
Training	187	266	453 (65.7%)
Test	121	116	237 (34.3%)
Total	308 (44.6%)	382 (55.4%)	690 (100%)

Τέλος, τα αποτελέσματα δείχνουν πως το μοντέλο του Naive Bayes είχε πολύ θετικά αποτελέσματα με ακρίβεια 83.3%, η οποία είναι υψηλότερη από το δέντρο ταξινόμησης και του K-NN, τα οποία είχαν 82% και 82.8% αντίστοιχα.

Ακόμη μία ενδιαφέρον έρευνα⁴ που εξετάζει τραπεζικά δεδομένα προσπάθησε να εφαρμόσει ένα νευρωνικό δίκτυο για την σωστή ταξινόμηση των δανείων. Για το πείραμα, δοκιμάστηκαν δύο σύνολα δεδομένων· το πρώτο περιλαμβάνει δεδομένα ιστορικά από πελάτες τράπεζας. Το σύνολο των δεδομένων σε αριθμό ήταν 1000, από τα οποία τα 700 χρησιμοποιήθηκαν για εκπαίδευση και τα 300 για αξιολόγηση. Το δεύτερο περιλαμβάνει πιστωτικά δεδομένα και ισχύουν τα ίδια όπως και στο πρώτο. Αναπτύχθηκε, από τους ερευνητές, ένα τεχνητό νευρωνικό δίκτυο και εφαρμόστηκε πάνω στα δύο σύνολα. Στο πρώτο, το δίκτυο είχε ακρίβεια 72%, με RMSE (Root Mean Square Error) ίσο με 0.04, το οποίο δείχνει πόσο κοντά ήταν οι τιμές των προβλεπόμενων τιμών σε σχέση με τις πραγματικές. Στο δεύτερο η ακρίβεια ήταν 98%, όμως το RMSE ήταν στο 0.10. Φυσικά και στα δύο ήταν κοντά στο 0. Αυτό σημαίνει πως οι τιμές και στα δύο ήταν πολύ κοντά με τις πραγματικές.

⁴ Patil, P. S., & Dharwadkar, N. V. (2017). Analysis of banking data using machine learning. 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)

Οι ερευνητές επισήμαναν ότι το μοντέλο δουλεύει αποτελεσματικά και για τα δύο σύνολα δεδομένων.

Οι Destile, Celik και Potsane ανέπτυξαν μία έρευνα⁵ πάνω στην οποία πραγματοποίησαν μία βιβλιογραφική ανάλυση για στην εφαρμογή πιο σύνθετων μοντέλων ταξινόμησης. Επισημαίνουν ότι τα σύνθετα προβλήματα έχουν δυσκολία στην εύκολη εξήγηση των προβλέψεών τους. Συνολικά, ερευνήθηκαν 74 έγγραφα, τα οποία δημοσιεύτηκαν μεταξύ του 2010 και του 2018. Οι στατιστικές τεχνικές που εξετάστηκαν περιλαμβάνουν τους αλγόριθμους Linear Discriminant Analysis, Logistic regression, Naive Bayes, k-Nearest Neighbor (k-NN), Decision Tree, Support Vector Machine (SVM), Random forest, XGBoost, καθώς επίσης και μία πληθώρα από τεχνητά νευρωνικά δίκτυα. Στην εργασία ελέγχθηκαν τα δεδομένα που παρουσιάζονται στον πίνακα που ακολουθεί:

Πίνακας 4: Σύνολο Δεδομένων που χρησιμοποιήθηκε.

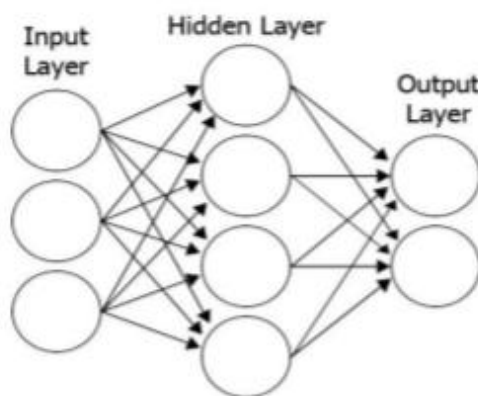
Dataset	Sample size	#Goods	#Bads	# of features
European Credit Bureau	186.574	179.544	7.030	324
UK	30.000	28.800	1.200	14
Barbados	21.117	20.614	503	20
Indonesia	14.700	14.290	4.410	30
Benelux 2 (Belgium,Netherlands and Luxembourg)	7.190	5.033	2.157	28
Brazilian	4.504	4.144	360	5
Benelux 1 (Belgium,Netherlands and Luxembourg)	3.123	1.040	2.083	27
University of California, San Diego	2.435	1.836	599	38
China	1.057	552	505	10
German	1.000	700	300	20
Iranian	1.000	950	50	27

⁵ Destile, X., Celik, T., & Potsane, M. (2020). Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied Soft Computing*

Australian	690	307	383	14
Japanese	653	296	357	15
Polish	240	128	112	30
Texas Banks	162	81	81	19

Η συγγραφείς έπειτα από εκτενή έρευνα σε όλες τις τεχνικές καταλήγουν στο συμπέρασμα πως οι σύνθετοι ταξινομητές είχαν καλύτερα αποτελέσματα από τους απλούς. Ωστόσο, παρόλο που στην αναλυμένη βιβλιογραφία δεν υπήρχε συχνή χρήση, τη διαφορά την έκαναν τα νευρωνικά δίκτυα.

Σε μία ακόμη ενδιαφέρον έρευνα⁶ πραγματοποιήθηκαν πειράματα για την εφαρμογή ενός νευρωνικού δικτύου που αποσκοπούσε στην ανάλυση ρίσκου. Ειδικότερα, οι ερευνητές προσπάθησαν να εφαρμόσουν δύο διαφορετικά νευρωνικά δίκτυα, το “Artificial Neural Network Multilayer Perceptron” και το “Artificial Neural Network with the Ensemble Method”. Όσον αφορά το πρώτο μοντέλο, όπως αναφέρεται στην εργασία, η αρχιτεκτονική του δικτύου αποτελείται από τρία επίπεδα. Το πρώτο επίπεδο είναι αυτό στο οποίο εισάγονται τα δεδομένα. Στο δεύτερο, που ονομάζεται Hidden layer, γίνεται η επεξεργασία των προαναφερόμενων δεδομένων. Τέλος, στο τρίτο, που ονομάζεται Επίπεδο Εξόδου (Output layer), εξάγονται τα τελικά αποτελέσματα. Η συγκεκριμένη μέθοδος παρουσιάζεται στην εικόνα που ακολουθεί.

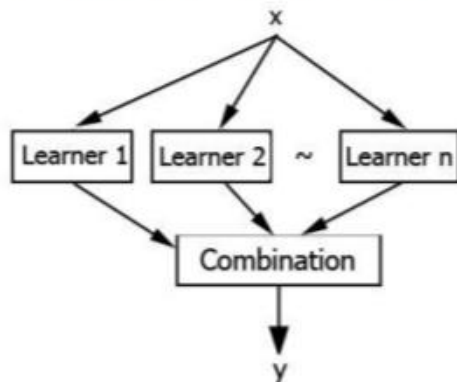


Εικόνα 1: Βασικό πολυεπίπεδο νευρωνικό δίκτυο με ένα κρυφό επίπεδο.

⁶ Edmond C., Girsang Abba S, (2020), Classification Performance for Credit Scoring using Neural Network. International Journal of Emerging Trends in Engineering Research

Το δεύτερο μοντέλο που αναπτύχθηκε βασίζεται στην αρχιτεκτονική που εμφανίζεται στην εικόνα που ακολουθεί.

Ensemble Method: Bootstrap Aggregation



Εικόνα 2: Μέθοδος Ensemble: Bootstrap Aggregation

Σύμφωνα με τους συγγραφείς, η διαφορά με την προηγούμενη αρχιτεκτονική που αναπτύχθηκε, είναι πως σε αυτή τη μέθοδο το μοντέλο εκπαιδεύεται με σκοπό να έχει πιο ακριβή αποτελέσματα. Αυτό επιτυγχάνεται εφόσον υπάρχει συνδυασμός από ταξινομητές στο κομμάτι της εκπαίδευσης. Το πιο πειστικό στοιχείο για τον παραπάνω ισχυρισμό είναι το γεγονός ότι η ακρίβεια του πρώτου νευρωνικού δικτύου ήταν στο 71%, ενώ του δεύτερου μοντέλου 74%. Στους πίνακες που ακολουθούν εμφανίζονται τα αποτελέσματα με λεπτομέρειες.

Πίνακας 5: Confusion Matrix νευρωνικού δικτύου.

	Positive (Actual)	Negative (Actual)
Positive (Predicted)	41.510	6.324
Negative (Predicted)	12.278	5.152
Accuracy	Precision	Recall
71%	87%	77%

Πίνακας 6: Confusion Matrix Ensembled νευρωνικού δικτύου.

	Positive (Actual)	Negative (Actual)
Positive (Predicted)	41.084	3.350
Negative (Predicted)	13.776	5.154
Accuracy	Precision	Recall
74%	93%	76%

Μία ομάδα επιστημόνων ανέπτυξε μία εργασία⁷ που αφορά την εκτίμηση ενός θετικού ή αρνητικού δανειολήπτη μέσα από ιστορικά δεδομένα. Η έρευνα βασίζεται πάνω στην ανάπτυξη ενός μοντέλου με βάση τον WKNN (Weighted K-Nearest Neighbor), ο οποίος είναι μία προέκταση του κλασικού KNN, με τη διαφορά ότι αυτός δίνει μεγαλύτερο βάρος στους κοντινούς γείτονες από ότι στους πιο απομακρυσμένους. Αυτή η διαδικασία επιτυγχάνεται σε δύο στάδια. Το πρώτο στάδιο αφοσιώνεται στην καταμέτρηση της απόστασης ενώ το δεύτερο στάδιο μετατρέπει αυτές τις αποστάσεις σε βάρη μέσω μίας kernel συνάρτησης. Στο κομμάτι των δεδομένων, οι ερευνητές χρησιμοποίησαν ένα σύνολο δεδομένων με τα χαρακτηριστικά όπως φαίνονται στο πίνακα 7.

Πίνακας 7: Χαρακτηριστικά του μοντέλου πιστοληπτικού ρίσκου.

Variable	Definition
Age	Age of the applicant in Years
Working Experience	WOeking experience of the applicant in years
Total Income	Monthly income in Rupiahs
Other Loan	Other loan amount in Rupiahs
Net Income	Monthly net income of the applicants in Rupiahs
Interaction To Bank	Long interaction to bank in years
Saving	Amount of he applicants savings in Rupiahs
Debt Ratio	% per month

Επιπρόσθετα, οι συγγραφείς παραθέτουν πίνακες οι οποίοι απεικονίζουν τα στατιστικά στοιχεία των καλών και κακών δανειοληπτών, καθώς και τα αποτελέσματα με βάση των kernel συναρτήσεων.

⁷Mukid, M. A., Widiharhi, T., Rusgiyono, A., & Prahutama, A. (2018). *Credit scoring analysis using weighted k nearest neighbor. Journal of Physics: Conference Series*

Πίνακας 8: Στατιστική περίληψη για τους καλούς δανειολήπτες

Variables	Mean	Standard Deviation	Minimum	Maximum
Age	37.726	7.606	23.000	64.000
Working Experience	7.665	6.781	1.000	39.000
Total Income	10278655.530	12089028.731	2253967.200	144000000.000
Other Loan	358474.500	1194111.250	0.000	17392868.000
Net income	5295639.944	5980694.323	0.000	63786168.000
Interaction to Bank	3.881	2.884	0.000	19.000
Saving	15065382.662	124231063.340	18599.400	2880181083.600
Debt Ratio	30.608	11.054	7.485	77.069

Πίνακας 9: Στατιστική περίληψη για τους κακούς δανειολήπτες

Variables	Mean	Standard Deviation	Minimum	Maximum
Age	37.038	8.331	22.000	57.000
Working Experience	6.190	6.608	1.000	34.000
Total Income	8855448.065	9809392.959	2275227.000	88471800.000
Other Loan	237887.793	1020949.380	0.000	9512016.000
Net income	4668156.853	4966941.559	0.000	49151000.000
Interaction to Bank	2.549	2.244	0.000	10.000
Saving	7318661.557	21114280.029	40266.000	176765400.000
Debt Ratio	32.496	14.979	8.282	130.724

Πίνακας 10: Αποτελέσματα του WKNN χρησιμοποιώντας μερικούς Kernel

Kernel	Accuration Measure	1	3	5	7	9	11	13
Rectangular	Sensitivity	0.8618	0.8947	0.9539	0.9605	0.9737	0.9934	1.0000
	Specificity	0.3056	0.2500	0.1667	0.1667	0.0833	0.1111	0.0556
	Accuration	0.7553	0.7713	0.8032	0.8085	0.8032	0.8245	0.8191
Triangular	Sensitivity	0.8618	0.8750	0.9013	0.9408	0.9539	0.9671	0.9671
	Specificity	0.3056	0.3333	0.2222	0.1944	0.1667	0.1389	0.1389
	Accuration	0.7553	0.7713	0.7713	0.7979	0.8032	0.8085	0.8085
Epanechnikov	Sensitivity	0.8618	0.8816	0.9079	0.9474	0.9539	0.9671	0.9671
	Specificity	0.3056	0.3333	0.1944	0.1944	0.1944	0.1667	0.1667
	Accuration	0.7553	0.7766	0.7713	0.8032	0.8085	0.8138	0.8138
Triweight	Sensitivity	0.8618	0.8816	0.8750	0.8947	0.9079	0.9145	0.9342
	Specificity	0.3056	0.3056	0.3056	0.2778	0.2222	0.1944	0.1944
	Accuration	0.7553	0.7713	0.7660	0.7766	0.7766	0.7766	0.7926
Gaussian	Sensitivity	0.8618	0.8947	0.9539	0.9605	0.9671	0.9934	1.000
	Specificity	0.3056	0.2500	0.1667	0.1667	0.0833	0.1111	0.0556
	Accuration	0.7553	0.7713	0.8032	0.8085	0.7979	0.8245	0.8191
Inversion	Sensitivity	0.8618	0.8882	0.9539	0.9539	0.9605	0.9803	1.0000
	Specificity	0.3056	0.2500	0.1667	0.1667	0.1111	0.1111	0.0556
	Accuration	0.7553	0.7660	0.8032	0.8032	0.7979	0.8138	0.8191

Όπως φαίνεται και από το τελευταίο πίνακα, οι kernel “Rectangular” και “Gaussian” είχαν τα αποδοτικότερα αποτελέσματα.

Ο Pradeep Singh, συγγραφέας της έρευνας⁸ προσπάθησε να πραγματοποιήσει μία συγκριτική μελέτη ανάμεσα σε δύο μεθόδους με σκοπό την ταξινόμηση τιμών. Όσον αφορά τη πρώτη μέθοδο (individual) ταξινόμησης, ο συγγραφέας εξηγεί πως όλες οι κατηγορίες αλγορίθμων χρησιμοποιούνται για αξιολόγηση. Αυτοί οι αλγόριθμοι μπορεί να είναι ταξινομητές που βασίζονται σε κανόνες, MLP, logistics κ.α. Αντίθετα στη δεύτερη μέθοδο (ensemble) χρησιμοποιούνται μια πληθώρα από αλγορίθμους ώστε να σημειωθούν καλύτερα αποτελέσματα. Τέτοιου είδους ταξινομητές βασίζονται κυρίως σε δέντρα και ένα πολύ καλό παράδειγμα είναι ο Random Forest.

⁸ Singh, P. (2017). *Comparative study of individual and ensemble methods of classification for credit scoring. 2017 International Conference on Inventive Computing and Informatics (ICICI).*

Τα δεδομένα τα οποία χρησιμοποιήθηκαν σε αυτή την έρευνα προέρχονται από το κλάδο του λιανικού εμπορίου της Ταϊβάν, αλλά είναι εξατομικευμένα για ανάπτυξη μοντέλου πιστοληπτικής ικανότητας. Στον πίνακα που ακολουθεί ο συγγραφέας παρουσιάζει με νούμερα τα δεδομένα που χρησιμοποίησε. Πάνω σε αυτά τα δεδομένα δοκιμάστηκαν αρκετοί αλγόριθμοι, οι οποίοι επίσης παρουσιάζονται σε έναν πίνακα.

Πίνακας 11: Περίληψη των πιστοληπτικών δεδομένων

Name	Independent Variable	Instances	Good/bad (%)	K- Cross validation
TC	24	30.000	78/22	10

Πίνακας 12: Αποτελέσματα του πειράματος ανά ταξινόμητή.

Classifier	Classified instances	ACC	TP	FP	P	Recall	F-M	ROC
ADB	24385	81.28	0.278	0.035	0.691	0.278	0.396	0.73
ASTREE	24554	81.84	0.357	0.051	0.668	0.357	0.465	0.765
ATSELECT	24576	81.92	0.325	0.04	0.695	0.325	0.443	0.707
BAGGING	24480	81.6	0.371	0.058	0.646	0.371	0.472	0.764
BAYSNET	23570	78.56	0.509	0.136	0.516	0.509	0.512	0.756
CBAL	24221	80.7	0.349	0.063	0.613	0.349	0.445	0.669
CVC	16921	56.40	0.36	0.378	0.213	0.36	0.268	0.491
CVR	24621	82.07	0.354	0.047	0.682	0.354	0.467	0.771
CONRULE	24402	81.34	0.273	0.033	0.701	0.273	0.393	0.618
IB1	21936	73.12	0.393	0.173	0.393	0.393	0.393	0.61
J48	24221	80.73	0.349	0.063	0.613	0.349	0.445	0.669
JRIP	24528	81.76	0.373	0.056	0.653	0.373	0.475	0.661
LOGI	24599	81.99	0.38	0.055	0.662	0.38	0.483	0.767
MLP	24666	82.22	0.538	0.097	0.612	0.538	0.572	0.792
MBOOSTAB	24385	81.28	0.278	0.035	0.691	0.278	0.396	0.716

NAVEBAYS	19671	65.57	0.715	0.361	0.36	0.715	0.479	0.745
NB TREE	23996	79.98	0.451	0.101	0.559	0.451	0.499	0.747
ONER	24576	81.92	0.325	0.04	0.695	0.325	0.443	0.642
RACEI	24309	81.03	0.32	0.05	0.643	0.32	0.427	0.742
RC	24228	80.76	0.347	0.061	0.616	0.347	0.444	0.732
RFOREST	24539	81.79	0.371	0.055	0.657	0.371	0.474	0.768
SPEGASOS	24581	81.93	0.326	0.041	0.695	0.326	0.444	0.643
TS	23491	78.30	0.558	0.153	0.509	0.558	0.532	0.766
VFI	23878	79.59	0.516	0.124	0.541	0.516	0.528	0.748
VPTR	23346	77.82	0.03	0.009	0.478	0.03	0.056	0.511

Σε ακόμη μία εργασία⁹ χρησιμοποιήθηκαν αλγόριθμοι μηχανικής μάθησης πάνω σε τραπεζικά δεδομένα. Η συγκεκριμένη έρευνα όμως είναι λίγο διαφορετική από τις υπόλοιπες καθώς δεν εξετάζει αν κάποιος πελάτης θα είναι αξιόπιστος ή όχι ώστε να χρηματοδοτηθεί με κάποιο δάνειο. Οι ερευνητές προσπάθησαν να εξετάσουν αν συμφέρει να χορηγήσει σε πελάτες πιστωτική κάρτα. Τα δεδομένα που εφαρμόστηκαν προέρχονται από πελάτες και αποτελούνται από 30000 εγγραφές και 24 χαρακτηριστικά. Στον πίνακα που ακολουθεί εμφανίζονται τα χαρακτηριστικά τα οποία χρησιμοποιήθηκαν από το σύνολο δεδομένων.

Πίνακας 13: Περιγραφή δεδομένων

Attribute Name	Description
ID	User ID
Limit_Bal	Amount of the given credit (NT dollar)
Sex	Gender(1=male, 2=female)
Education	Education(1=graduateschool,2=University, 3=others)
Marriage	Marital Status(1=married, 2=unmarried)
Age	Age(year)

⁹ Sayjadah, Y., Hashem, I. A. T., Alotaibi, F., & Kasmiran, K. A. (2018). Credit Card Default Prediction using Machine Learning Techniques. 2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA).

Pay_1 - Pay_5	The repayment status from April to September 2005
Bill_Amt1 to Bill_Amt 6	Ammount of bill statement from April to September, 2005
Pay_Amt1 to Pay_Amt6	Amoynt paid ins September, 2005
Default_Payment_Next_Month	Amount to be paid next month

Στη συνέχεια, οι ερευνητές πραγματοποίησαν την προ-επεξεργασία των δεδομένων και χώρισαν το σύνολο δεδομένων δύο μέρη. Το πρώτο για την εκπαίδευση τους, το οποίο αποτελείται από το 70% του συνόλου και το δεύτερο για την αξιολόγηση, δηλαδή το 30%. Οι αλγόριθμοι που δοκιμάστηκαν ήταν τρεις στον αριθμό, Logistic Regression, Decision Tree και Random Forest. Έπειτα εφαρμόστηκαν κάποιες μετρήσεις ώστε να επιλεγεί το πιο αποδοτικό μοντέλο. Οι μετρήσεις αυτές έχουν να κάνουν με το Accuracy (ακρίβεια), τα True Positive (τα σωστά ταξινομημένα δείγματα ως σωστά) και το AUC (Area Under Curve). Τα αποτελέσματα εμφανίζονται στον πίνακα.

Πίνακας 14: Αποτελέσματα αλγορίθμων

Algorithm	Accuracy	True Positive	AUC
Logistic Regression	0.82	6659	0.75
Rpart Decision Tree	0.8206	6744	0.64
Random Forest	0.8181	6639	0.77

Όπως φαίνεται και από τα αποτελέσματα και οι τρεις αλγόριθμοι είχαν αρκετά αισιόδοξα νούμερα με πάνω από 80% ακρίβεια. Ωστόσο επιλέχθηκε ο Random Forest.

Μία ομάδα ερευνητών, σε μία εργασία¹⁰, προσπάθησαν να εφαρμόσουν αλγορίθμους μηχανικής μάθησης με σκοπό την πρόωρη διάγνωση πιστοληπτικών λαθών. Στη προσπάθεια αυτή εφαρμόστηκαν αλγόριθμοι όπως Support Vector

¹⁰ Shoumo, S. Z. H., Dhruba, M. I. M., Hossain, S., Ghani, N. H., Arif, H., & Islam, S. (2019). Application of Machine Learning in Credit Risk Assessment: A Prelude to Smart Banking. TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON).

Machine, Random Forest, Extreme Gradient Boosting και Logistic Regression. Αυτοί με τη σειρά τους αξιολογήθηκαν με τη βοήθεια μερικών τεχνικών αξιολόγησης. Πρώτο βήμα των ερευνητών ήταν η προετοιμασία του συνόλου δεδομένων. Σε αυτή τη διαδικασία καθαρίστηκε ο θόρυβος των δεδομένων και αφαιρέθηκαν τυχόν μη χρήσιμα χαρακτηριστικά. Επομένως, το σύνολο των δειγμάτων σε αριθμό ήταν 38661 και αποτελούνται από 32 χαρακτηριστικά. Έπειτα εφαρμόστηκε η τεχνική της κανονικοποίησης “Min-Max” και στη συνέχεια πραγματοποιήθηκαν διάφορα πειράματα. Η αξιολόγηση αυτών των πειραμάτων έγινε σε δύο στάδια. Στο πρώτο ελέγχονται οι αλγόριθμοι πάνω στα “test” δεδομένα, δηλαδή αυτά που είναι υποψήφια για αξιολόγηση και στο δεύτερο στάδιο εφαρμόζεται η τεχνική του “k-fold cross validation” με k=5 και “GridSearchCV”. Επίσης, αξίζει να τονιστεί πως οι αναλυτές προσπάθησαν να συγκρίνουν τα αποτελέσματα εφαρμόζοντας τη μία φορά τη μέθοδο “RFECV” για την επιλογή χαρακτηριστικού και την άλλη ο αλγόριθμος εξαγωγής χαρακτηριστικού PCA. Τα αποτελέσματα τους εμφανίζονται στις εικόνες που ακολουθούν.

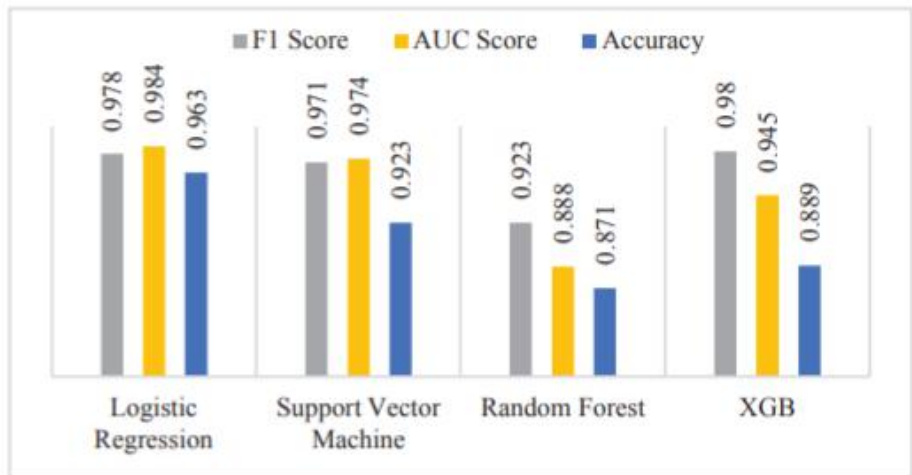
Πίνακας 15: PCA

PCA				
	Logistic Regression	Support Vector Machine	Random Forest	XGB
Precision	0.987	0.982	0.945	0.971
Recall	0.969	0.961	0.902	0.898
F1 Score	0.978	0.971	0.923	0.980
AUC Score	0.984	0.974	0.888	0.945
Accuracy	0.963	0.923	0.871	0.889

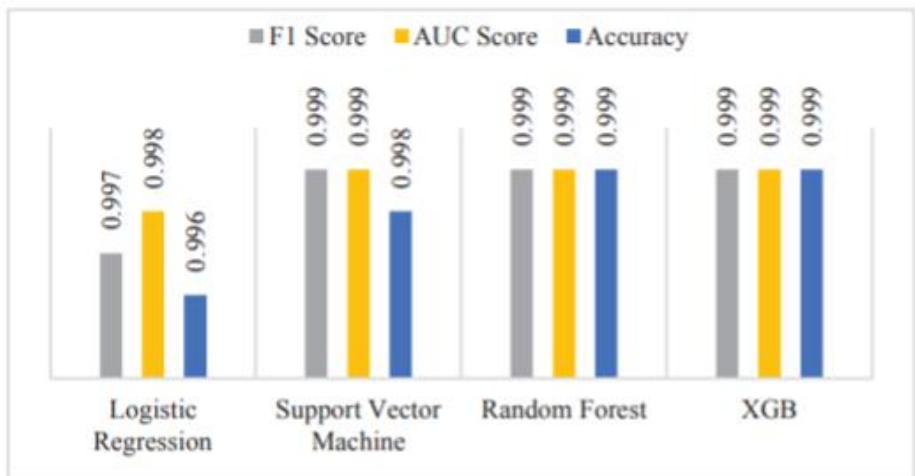
Πίνακας 16: RFECV

RFECV				
	Logistic Regression	Support Vector Machine	Random Forest	XGB
Precision	0.995	0.998	0.998	0.998
Recall	1.0	1.0	1.0	1.0
F1 Score	0.997	0.999	0.999	0.999

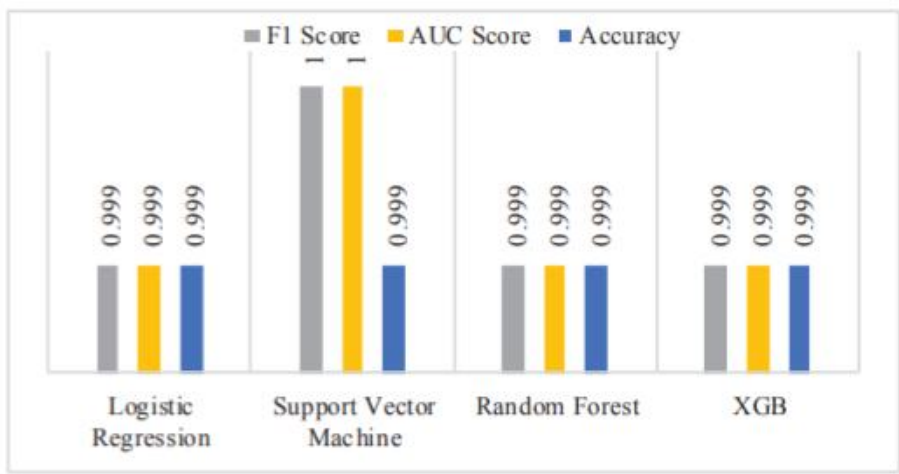
AUC Score	0.998	0.999	0.999	0.999
Accuracy	0.996	0.998	0.999	0.999



Εικόνα 3: Σύγκριση μοντέλων με PCA



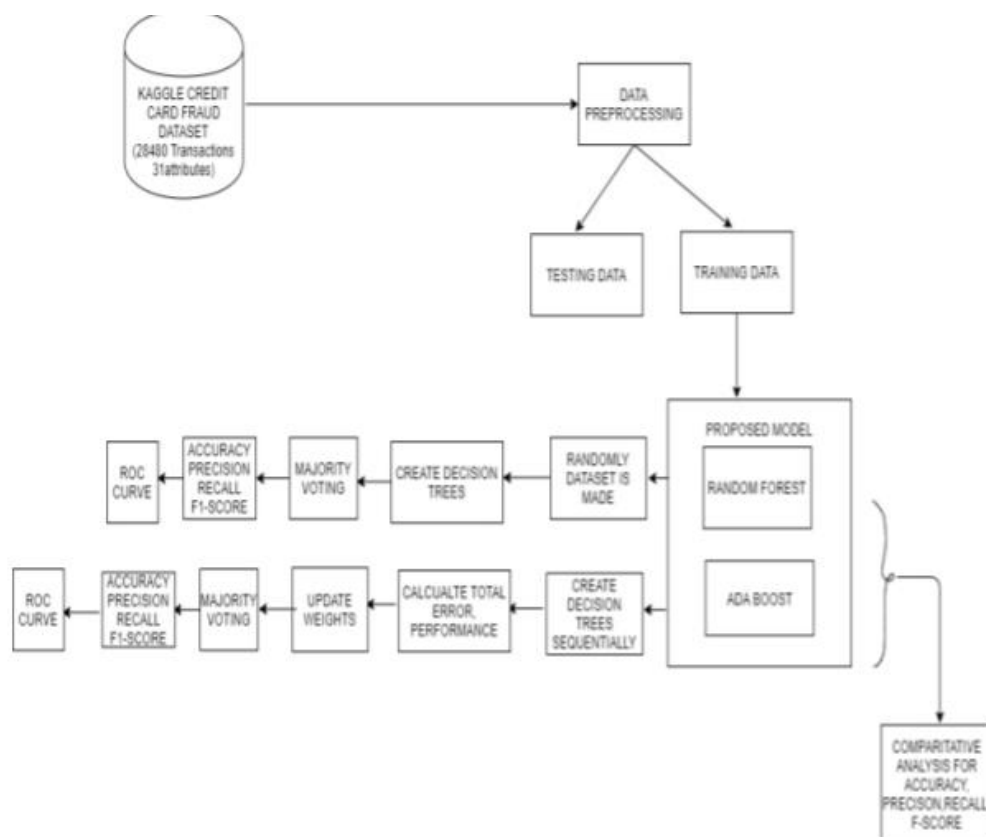
Εικόνα 4: Σύγκριση μοντέλων RFECV



Εικόνα 5: Σύγκριση μοντέλων με RFECV και Cross-Validation

Ένα πρόβλημα το οποίο έχει απασχολήσει τον ερευνητικό χώρο, το οποίο είναι σχετικό με το credit scoring είναι η ανίχνευση απάτης (Fraud Detection) πάνω στις πιστωτικές κάρτες. Με άλλα λόγια ο όρος αυτός εκφράζει το γεγονός κλοπής ή χρήσης της πιστωτικής κάρτας από έναν μη εξουσιοδοτημένο χρήστη. Έτσι λοιπόν, μία ομάδα επιστημόνων ανέπτυξε μία έρευνα πάνω στο συγκεκριμένο πρόβλημα.

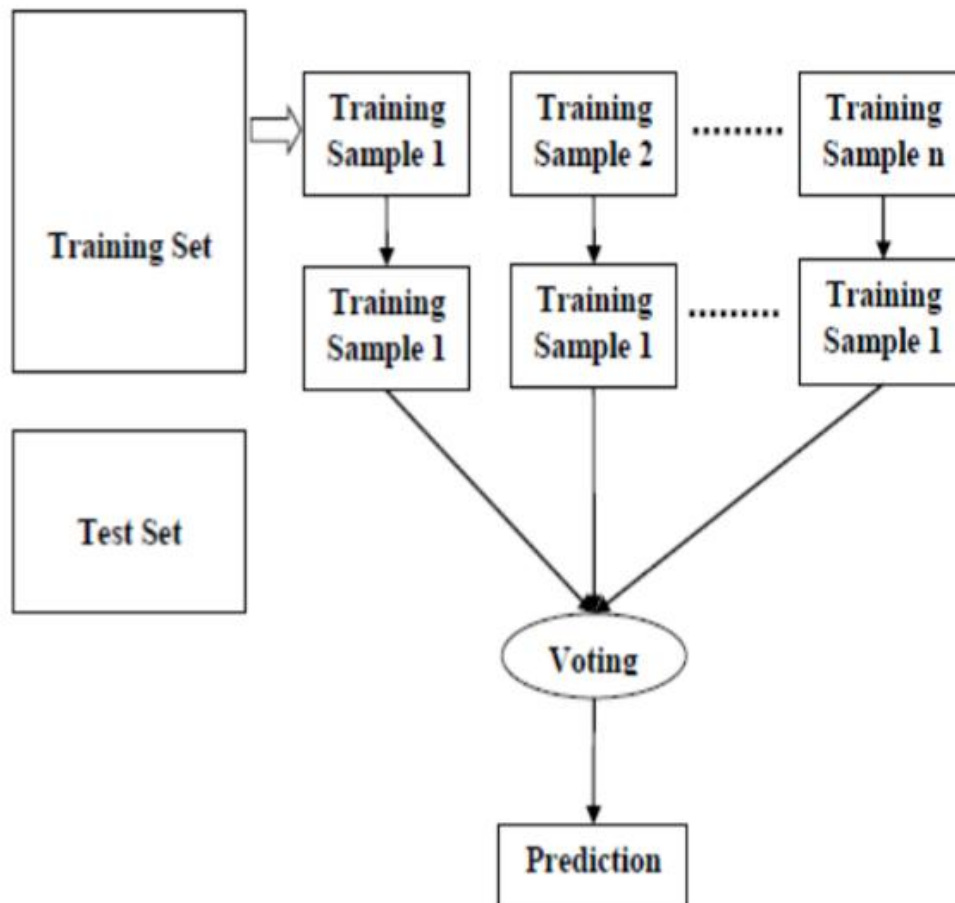
Οι συγγραφείς αυτής της έρευνας¹¹ προσπάθησαν να ανιχνεύσουν τέτοια περιστατικά με τη βοήθεια της μηχανικής μάθησης. Πιο συγκεκριμένα, επικεντρώθηκαν σε δύο αλγόριθμους, Random Forest και Adaboost. Όπως αναφέρεται στην εργασία τα δεδομένα προέρχονται από τον ιστότοπο “Kaggle”. Προτού εφαρμοστούν οι αλγόριθμοι χωρίστηκαν τα δεδομένα σε δύο μέρη για αξιολόγηση και εκπαίδευση. Οι μετρήσεις που υπολογίστηκαν για την αξιολόγηση του μοντέλου ήταν οι accuracy, precision, recall και F1-score. Αναλυτικότερα, στο πίνακα στο σχήμα που ακολουθεί οι συγγραφείς παρουσιάζουν τα βήματα που ακολούθησαν.



Εικόνα 6: Διάγραμμα αρχιτεκτονικής

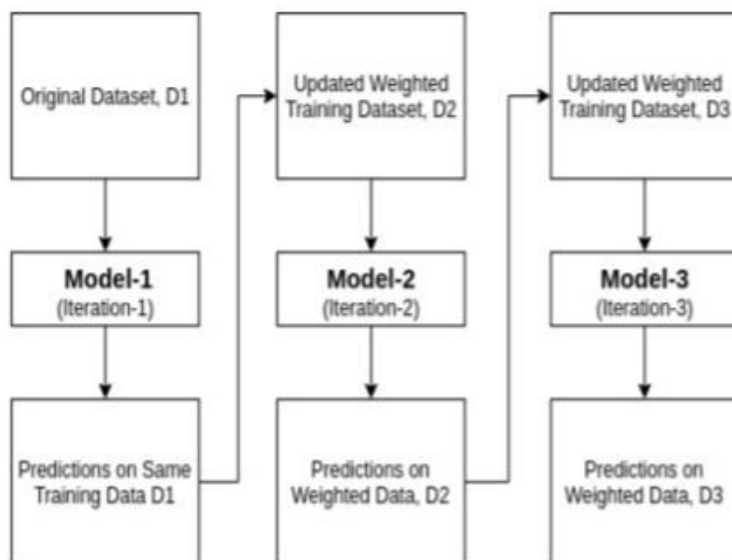
¹¹ Sailusha, R., Gnaneshwar, V., Ramesh, R., & Rao, G. R. (2020). Credit Card Fraud Detection Using Machine Learning. 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS).

Όπως έχει αναφερθεί και προηγουμένως δοκιμάστηκαν δύο αλγόριθμοι στην εργασία. Ο πρώτος Random Forest ο οποίος κατατάσσεται στην κατηγορία των δέντρων. Στο σχήμα που ακολουθεί περιγράφεται η λειτουργία του όπως την παρουσιάζουν οι αναλυτές.



Εικόνα 7: Αλγόριθμος Random Forest

Ο δεύτερος αλγόριθμος, Adaboost, είναι ένας αλγόριθμος που η λογική του βασίζεται στη βελτιώση των “αδύναμων” ταξινομητών. Πιο συγκεκριμένα, δημιουργούνται κάποια μοντέλα. Το πρώτο προκύπτει από τα δεδομένα εκπαίδευσης, το δεύτερο διορθώνοντας τις ατέλειες του πρώτου και αυτό επαναλαμβάνεται μέχρι να δημιουργηθεί ένας ανώτατος αριθμός από μοντέλα ή μέχρι να ταξινομηθούν σωστά τα δείγματα. Αυτή η διαδικασία παρουσιάζεται στην εργασία με το εξής σχήμα.



Εικόνα 8: Αλγόριθμος Adaboost

Τέλος, γίνεται αναφορά στα αποτελέσματα των αλγορίθμων μέσα από κάποιες μεθόδους αξιολόγησης. Τα αποτελέσματα του Random Forest απεικονίζονται στους πίνακες 17, 18 και 19.

Πίνακας 17: Αποτελέσματα εξόδου του αλγόριθμου Random Forest

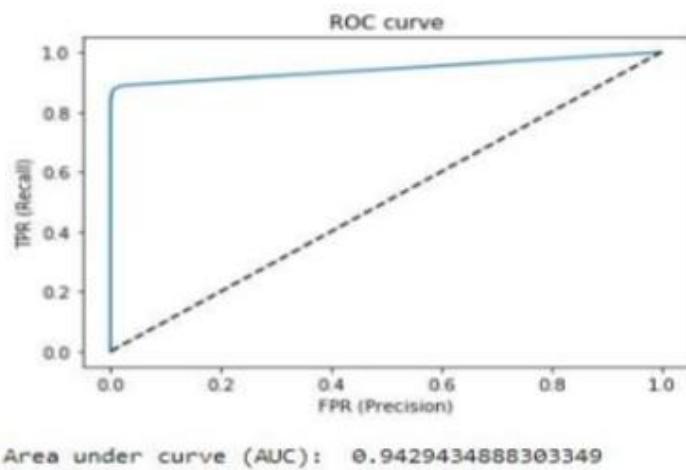
	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	93825
1	0.95	0.77	0.85	162
Accuracy			1.00	93987
Macro avg	0.97	0.89	0.93	93987
Weighted avg	1.00	1.00	1.00	93987

Πίνακας 18: Confusion Matrix για τα δεδομένα εκπαίδευσης, Random Forest

190490	0
0	330

Πίνακας 19: Confusion Matrix για τα δεδομένα αξιολόγησης, Random Forest

93818	37
7	125



Εικόνα 9: Διάγραμμα ROC, Random Forest

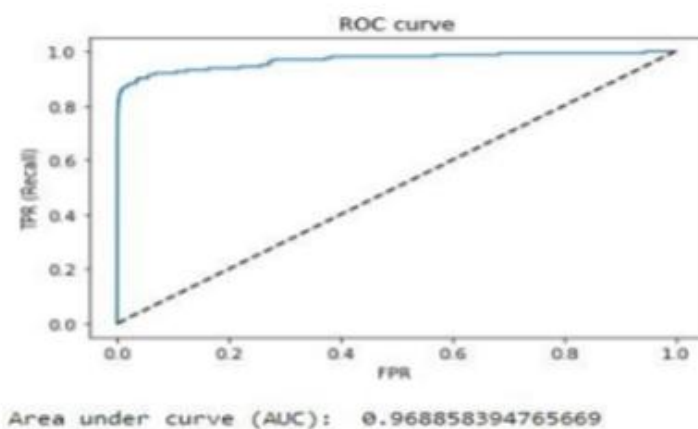
Τα αποτελέσματα του Adaboost αναγράφονται στους πίνακες που ακολουθούν. Όπως επίσης οι συγγραφείς πραγματοποίησαν και ένα συγκριτικό διάγραμμα μεταξύ των δύο αλγορίθμων, το οποίο και αυτό εμφανίζεται πιο κάτω.

Πίνακας 20: Confusion Matrix για τα δεδομένα εκπαίδευσης, Adaboost

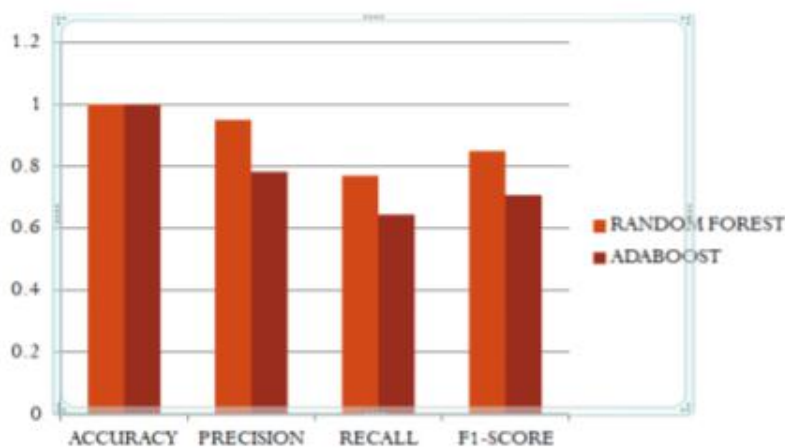
190464	120
26	210

Πίνακας 21: Confusion Matrix για τα δεδομένα αξιολόγησης, Adaboost

93811	65
14	97



Εικόνα 10: Διάγραμμα ROC, Adaboost



Εικόνα 11: Σύγκριση αλγορίθμων

Ο Yu li σε μία έρευνα¹² του πειραματίστηκε πάνω στην ιδέα της υλοποίησης ενός μοντέλου με σκοπό την πρόβλεψη ενός καλού ή κακού δανειολήπτη. Πιο αναλυτικά, τα δεδομένα τα οποία χρησιμοποιήθηκαν για την εργασία περιλαμβάνουν 45.000 δείγματα τα οποία είχαν ως αποτέλεσμα καλούς δανειολήπτες και 5.000 που είχαν κακούς, δηλαδή η αναλογία μεταξύ καλών και κακών πελατών είναι 9:1. Στο πείραμα δοκιμάστηκαν δύο αλγόριθμοι, Logistic Regression και XGBoost. Ο Logistic Regression βασίζεται στην σχέση μεταξύ εξαρτημένων μεταβλητών (με δυαδικές ή μη μεταβλητές) και ανεξάρτητων μεταβλητών, δηλαδή το χαρακτηριστικό που προορίζεται για πρόβλεψη. Με πιο απλά λόγια ο αλγόριθμος υπολογίζει τη πιθανότητα εμφάνισης συμβάντων προς τη πιθανότητα μη εμφάνισης. Από την άλλη πλευρά ο XGBoost, είναι μια βελτιωμένη έκδοση του “gradient boost”, που ανήκει στην κατηγορία των δέντρων. Η λειτουργία του είναι απλή, ξεκινά από έναν κόμβο και συνεχώς αναπτύσσεται δημιουργώντας νέους κόμβους. Σε αυτή τη προσπάθεια χρησιμοποιείται το ευρετήριο Gini με σκοπό την επιλογή των καλύτερων χαρακτηριστικών και σημείων διαχωρισμού, έτσι ώστε να δημιουργηθεί η κατάλληλη δομή δέντρου. Τα αποτελέσματα που προέκυψαν βασίζονται σε δύο μετρήσεις, AUC και KS. Τα νούμερα που προκύπτουν από τον Logistic Regression και τον XGBoost αναγράφονται στους πίνακες που ακολουθούν.

Πίνακας 22: Αποτελέσματα Logistic Regression

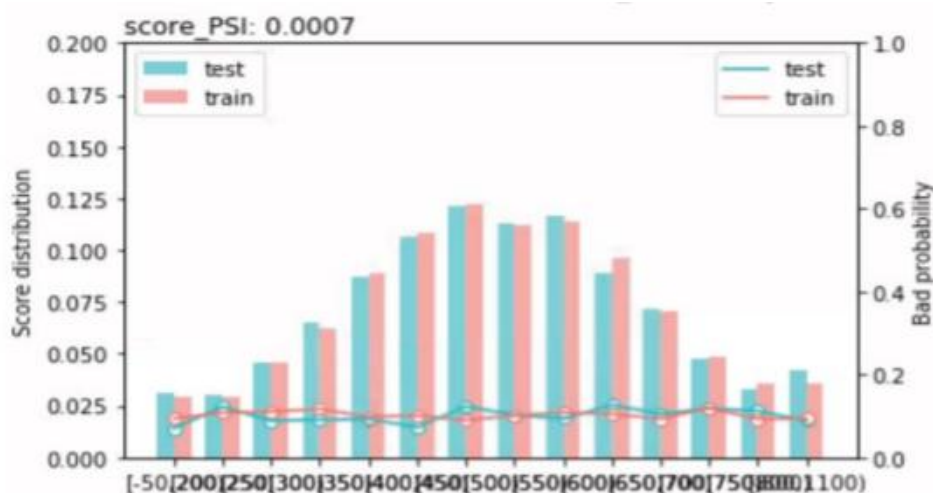
¹² Li, Y. (2019). Credit Risk Prediction Based on Machine Learning Methods. 2019 14th International Conference on Computer Science & Education (ICCSE).

Metrics	Training	Test	Validation
AUC	0.8619	0.8585	0.835
KS	0.5749	0.571	0.52449

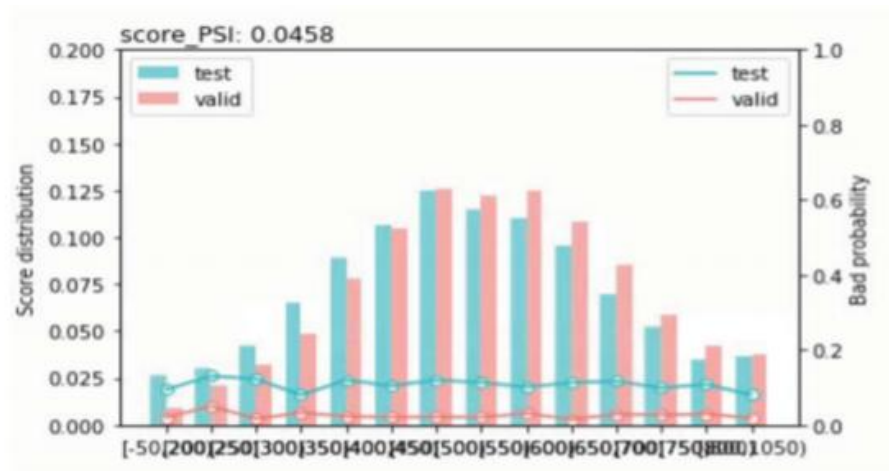
Πίνακας 23: Αποτελέσματα XGBoost

Metrics	Training	Test	Validation
AUC	0.9135	0.9079	0.8806
KS	0.6779	0.6617	0.6089

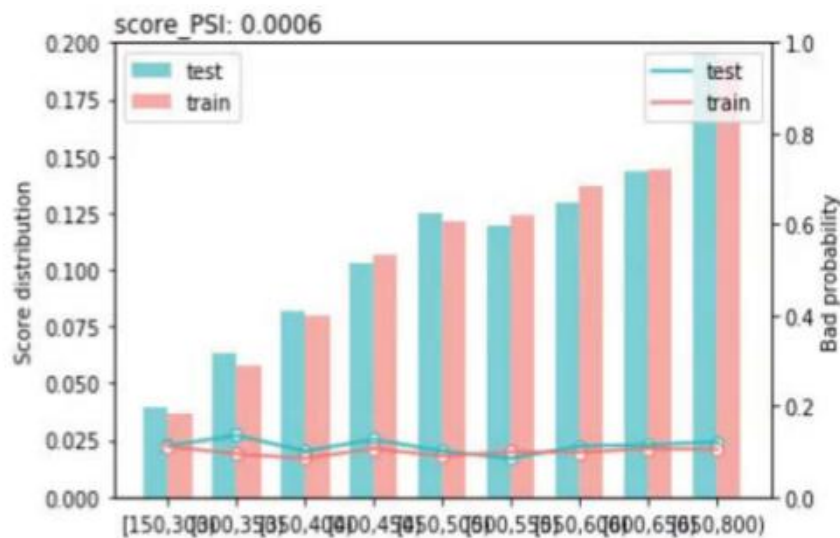
Επιπλέον, έγινε μία προσπάθεια να υπολογιστεί η σταθερότητα του κάθε αλγόριθμου ξεχωριστά. Αυτό επιτυγχάνεται με τη μέτρηση PSI. Η τιμή του PSI δείχνει αν η κατανομή του πληθυσμού έχει αλλάξει για διαφορετικά δείγματα ή δείγματα σε διαφορετικούς χρόνους μετά την ταξινόμηση ανάλογα με την πιθανότητα ή τη βαθμολογία. Με πιο απλά λόγια, δείχνει αν ο αριθμός των ατόμων για κάθε βαθμολογία έχει αλλάξει αισθητά. Αν αυτή η τιμή είναι κάτω από 0.1 τότε η σταθερότητα θεωρείται πολύ υψηλή, ενώ αν είναι μεγαλύτερη από 0.2 χαμηλή. Μια φυσιολογική τιμή είναι μεταξύ 0.1 και 0.2. Στα γραφήματα που ακολουθούν φαίνεται η σταθερότητα και με τους δύο αλγορίθμους αντίστοιχα. Παρουσιάζονται, οι τιμές PSI εφαρμοσμένες στα δεδομένα (train,test) και (test,valid) για κάθε αλγόριθμο αντίστοιχα.



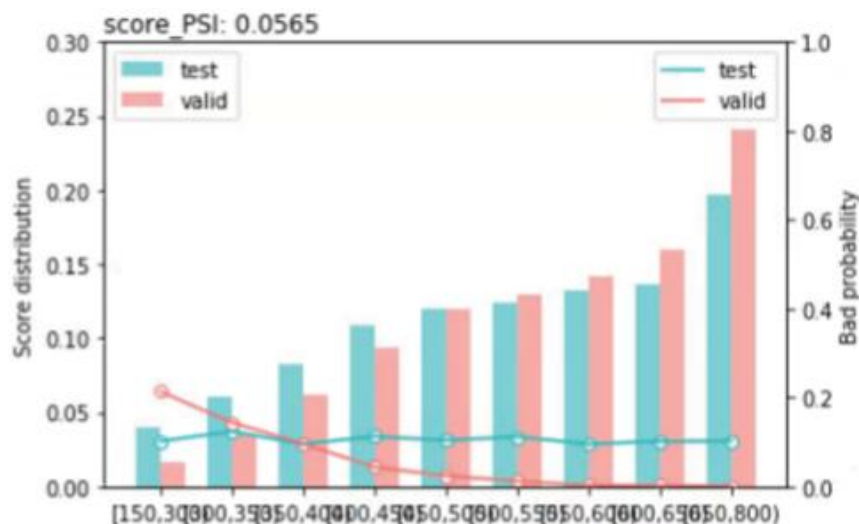
Εικόνα 12: Διάγραμμα PSI (test, train), για τον αλγόριθμο Logistic Regression



Εικόνα 13: Διάγραμμα PSI (test, valid), για τον αλγόριθμο Logistic Regression



Εικόνα 14: Διάγραμμα PSI (test, train), για τον αλγόριθμο XGBoost



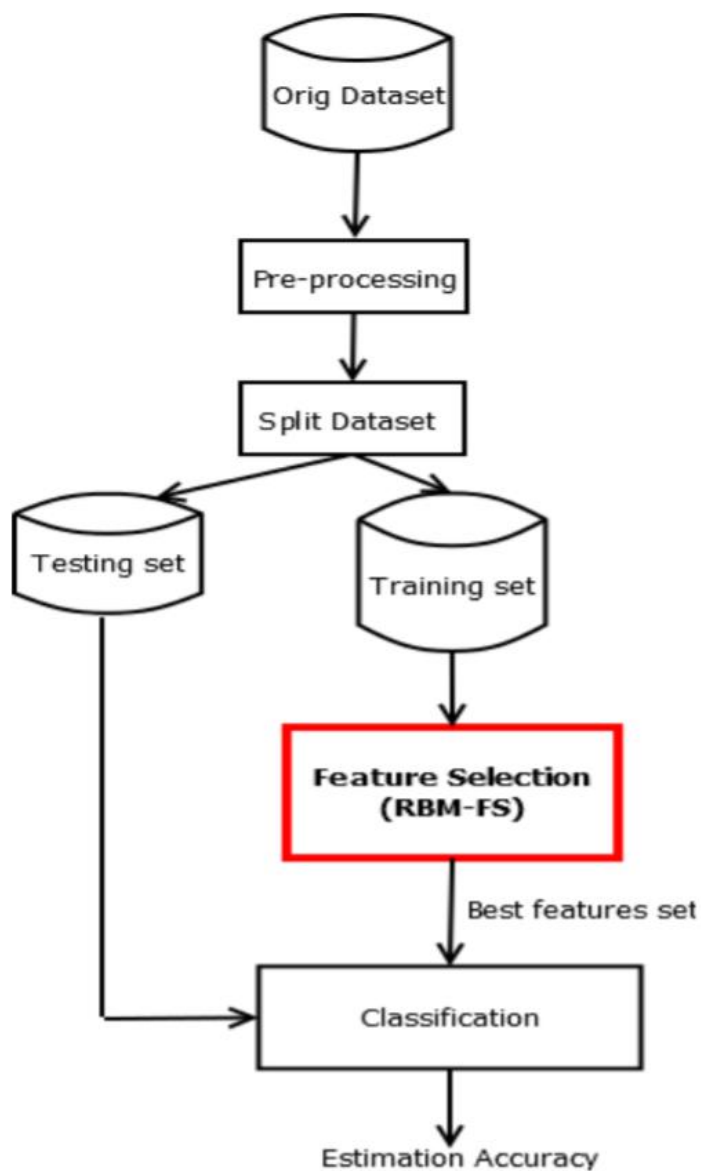
Εικόνα 15: Διάγραμμα PSI (test, valid), για τον αλγόριθμο XGBoost

Όπως φαίνεται και από τα αποτελέσματα, η απόδοση του αλγορίθμου XGBoost είναι αρκετά πιο αισιόδοξη από αυτή του Logistic Regression, όπως επισημαίνει και ο συγγραφέας.

Σε μία έρευνα¹³, λίγο διαφορετική σε σχέση με όλες τις προαναφερόμενες, βασίστηκε στην ανάπτυξη ενός μοντέλου πάνω στην αγορά των P2P (Peer to Peer) δανείων. Ο δανεισμός Peer to Peer είναι μία πρακτική δανεισμού σε ιδιώτες ή επιχειρήσεις μέσω διαδικτυακών υπηρεσιών. Το μοντέλο πιστωτικού κινδύνου που αναπτύχθηκε βασίζεται σε ήδη υπάρχον τεχνικές μηχανικής μάθησης. Τα πειράματα εφαρμόστηκαν σε πραγματικά δεδομένα από αγορές δανεισμού P2P. Αυτά διαιρούνται σε τρία διαφορετικά σύνολα (Australian Credit, German credit, Lending club dataset). Το πρώτο (Australian credit) αποτελείται από 690 εγγραφές και 17 χαρακτηριστικά, το δεύτερο (German credit) από 1000 εγγραφές με 24 αριθμητικά χαρακτηριστικά και το τρίτο (Lending club dataset) από 60.000 εγγραφές και 123 χαρακτηριστικά.

Τα βήματα για την υλοποίηση του μοντέλου ήταν τα εξής, Αρχικά, πραγματοποιήθηκαν λειτουργίες για την προ-επεξεργασία των δεδομένων (Scaling, Normalization, Transformation). Έπειτα, ακολούθησε η επιλογή χαρακτηριστικού (feature selection) μέσω RBM (Restricted Boltzmann Machines). Η επιλογή χαρακτηριστικού πραγματοποιήθηκε σε δύο μέρη. Το πρώτο, για να εντοπιστούν τα καλύτερα χαρακτηριστικά το σύνολο εκπαιδεύτηκε μέσω RBM, μετά την εκπαίδευση υπολογίστηκε η αξία κάθε χαρακτηριστικού με τη χρήση του RMSE, που εντοπίζει τη πιο σημαντική πληροφορία στο RBM. Στο δεύτερο μέρος, χρησιμοποιούνται τα χαρακτηριστικά που βρέθηκαν στο προηγούμενο βήμα ώστε να χρησιμοποιηθούν για αξιολόγηση. Το επόμενο βήμα ήταν η εφαρμογή αλγορίθμων, εφαρμόστηκαν αλγόριθμοι όπως Logistic Regression, Linear discriminant analysis (LDA), Support Vector Machine (SVM), Artificial Neural Network (ANN), k Nearest Neighbor (k-NN) και Random Forest (RF). Η διαδικασία ανάπτυξης των μοντέλων και τα αποτελέσματα που προέκυψαν από αυτά τα πειράματα απεικονίζονται στις εικόνες που ακολουθούν, όπως εμφανίζονται από τους συγγραφείς.

¹³ Ha, V.-S., Lu, D.-N., Choi, G. S., Nguyen, H.-N., & Yoon, B. (2019). *Improving Credit Risk Prediction in Online Peer-to-Peer (P2P) Lending Using Feature selection with Deep learning*. 2019 21st International Conference on Advanced Communication Technology.

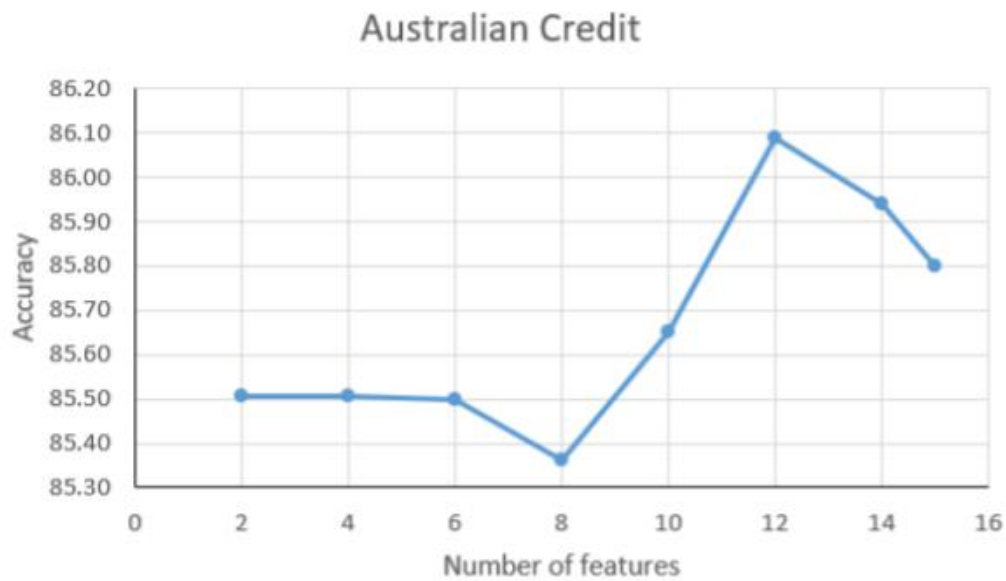


Εικόνα 16: Προτεινόμενη μέθοδος

Πίνακας 24: Επίδοση μοντέλου

Models	Predictive Performance (%)		
	German	Australia	Lending Club
LDA	76.50	85.80	81.20
Logistic Regression	76.40	85.51	81.05
ANN	75.8	71.45	66.08
K-NN	67.10	65.94	72.55
Linear SVM	74.80	84.35	76.60
Random Forest	67.72	67.72	67.72

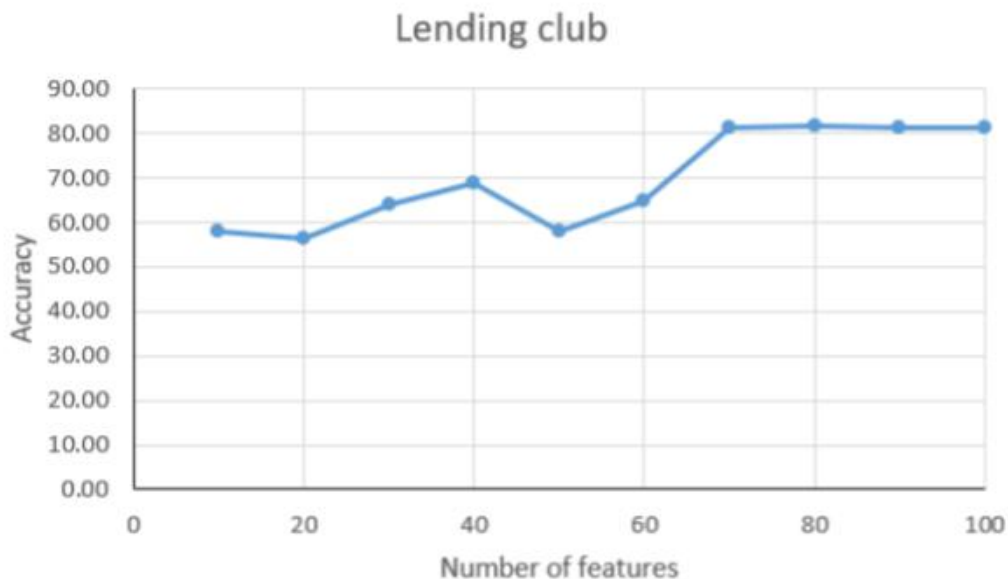
Επίσης, στην εργασία πραγματοποιήθηκε διαδικασία που αποτυπώνει την ακρίβεια του κάθε μοντέλου σε σχέση με τον αριθμό των χαρακτηριστικών, όπως φαίνεται και στα γραφήματα.



Εικόνα17: Ακρίβεια σε σχέση με τον αριθμό των χαρακτηριστικών, στο σύνολο δεδομένων Australian Credit



Εικόνα 18: Ακρίβεια σε σχέση με τον αριθμό των χαρακτηριστικών, στο σύνολο δεδομένων German Credit



Εικόνα 19: Ακρίβεια σε σχέση με τον αριθμό των χαρακτηριστικών, στο σύνολο δεδομένων Lending club

Οι συγγραφείς τονίζουν πως τα καλύτερα αποτελέσματα, όσο αφορά την ακρίβεια και την επιλογή χαρακτηριστικού, τα είχε ο αλγόριθμος Linear discriminant analysis (LDA) και για τρία σύνολα δεδομένων. Με 76.50, 85.80, και 81.20 ακρίβεια για το κάθε σύνολο.

Ο Zaron Ereiz σε εργασία¹⁴ του προσπάθησε με τη βοήθεια του “OptiML” να εφαρμόσει τεχνικές και αλγόριθμους μηχανικής μάθησης πάνω σε τραπεζικά δεδομένα πελατών. Ο συνολικός τους αριθμός ήταν 40.812 δείγματα και 29 χαρακτηριστικά. Στην πορεία της εργασίας τα δεδομένα χωρίστηκαν σε δεδομένα εκπαίδευσης και αξιολόγησης. Το 75% από αυτά προορίστηκαν για εκπαίδευση και τα υπόλοιπα για αξιολόγηση. Επόμενο βήμα ήταν η προετοιμασία των δεδομένων μέσω του OptiML με σκοπό να εφαρμοστούν οι αλγόριθμοι. Στη συγκεκριμένη έρευνα εφαρμόστηκαν αλγόριθμοι όπως οι Decision Forest, Neural Network και Logistic Regression. Όσο αφορά την αξιολόγηση αυτών των μοντέλων χρησιμοποιήθηκαν κάποιες από τις σημαντικότερες μετρήσεις όπως, Accuracy, Precision και Recall. Τα αποτελέσματα εμφανίζονται σε εικόνες που έχουν παρουσιαστεί από τον συγγραφέα.

¹⁴ Ereiz, Z. (2019). Predicting Default Loans Using Machine Learning (OptiML). 2019 27th Telecommunications Forum (TELFOR).

ACTUAL VS. PREDICTED			
		0	1
0	22,718	18	
1	1,275	41	
PREDICTED	23,993	59	
PRECISION	94.69%	69.49%	
ACTUAL	RECALL	F	Phi
22,736	99.92%	0.97	0.14
1,316	3.12%	0.06	0.14
24,052	51.52% AVG. RECALL	0.52 AVG. F	0.14 AVG. Phi
82.09% AVG. PRECISION	94.62% ACCURACY		
94.6% Accuracy	0.0596 F-measure		
69.5% Precision	3.1% Recall	0.1396 Phi coefficient	

Εικόνα 20: Αξιολόγηση του Decision forest

ACTUAL VS. PREDICTED					
		0		1	
0		19,063		3,697	
1		613		679	
PREDICTED		19,676		4,376	
PRECISION		96.89%		15.52%	
ACTUAL	RECALL	F	Phi		
22,760	83.76%	0.90	0.21		
1,292	52.55%	0.24	0.21		
24,052	68.16% AVG. RECALL	0.57 AVG. F	0.21 AVG. Phi		
56.20% AVG. PRECISION		82.08% ACCURACY			
82.1% Accuracy				0.2396 F-measure	
15.5% Precision		52.6% Recall		0.2122 Phi coefficient	

Εικόνα 21: Αξιολόγηση του νευρωνικού δικτύου

ACTUAL VS. PREDICTED		0		1	
		0		1	
0		22,744		16	
1		1,261		31	
PREDICTED		24,005		47	
PRECISION		94.75%		65.96%	
ACTUAL	RECALL	F	Phi		
22,760	99.93%	0.97	0.12		
1,292	2.40%	0.05	0.12		
24,052	51.17% AVG. RECALL	0.51 AVG. F	0.12 AVG. Phi		
80.35% AVG. PRECISION	94.69% ACCURACY				
94.7% Accuracy	0.0463 F-measure				
66.0% Precision	2.4% Recall	0.1189 Phi coefficient			

Εικόνα 22: Αξιολόγηση του Logistic Regression

Όπως φαίνεται και από τα αποτελέσματα οι Logistic Regression και Decision Forest αλγόριθμοι είχαν αποτελέσματα παρόμοια. Από τη πλευρά του Logistic Regression το Accuracy ήταν 94.7% και το Precision 94.75%, από την άλλη πλευρά του Decision Forest σημειώθηκε 94.6% Accuracy και Precision 94.69%. Νούμερα αρκετά αισιόδοξα συγκριτικά με του Neural Network που είχε Accuracy 82.1%.

Περιγραφή Δεδομένων

Στη συγκεκριμένη ενότητα αναλύονται τα δεδομένα τα οποία χρησιμοποιήθηκαν για την έρευνα. Αρχικά, αξίζει να σημειωθεί πως τα προαναφερόμενα δεδομένα συλλέχθηκαν από τον ιστότοπο “Kaggle”. Το συγκεκριμένο σύνολο δεδομένων περιέχει 30 στήλες, το περιεχόμενο των οποίων έχει να κάνει με τραπεζικά δεδομένα πελατών σχετιζόμενα με την έγκριση ή όχι ενός δανείου. Οι εγγραφές δεν ανταποκρίνονται σε φυσικά πρόσωπα και αποτελούν εικονικά δείγματα. Ο συνολικός αριθμός των δειγμάτων είναι κοντά στις 900.000 και ακριβέστερα στις 887.379 εγγραφές.

Αναλυτική περιγραφή δεδομένων

Όπως έχει αναφερθεί προηγουμένως, τα δεδομένα αποτελούνται από 30 στήλες, οι οποίες περιγράφονται στον παρακάτω πίνακα. Είναι σημαντικό να σημειωθεί πως έπειτα από επεξεργασία των δεδομένων ορισμένες στήλες έχουν αφαιρεθεί.

Πίνακας 25: Περιγραφή δεδομένων

Αριθμός	Στήλη	Περιγραφή
1	id	Καθορίζει μοναδικά έναν πελάτη
2	year	Εκφράζει τη χρονιά την οποία ένας πελάτης απέκτησε δάνειο (2007-2015)
3	issue_d	Εκφράζει την ημέρα που εκδόθηκε το δάνειο (6/01/07-12/01/15)
4	final_d	Δεν αναφέρεται
5	emp_length_int	(0,5-10)
6	home_ownership	Δηλώνει αν ο πελάτης είναι κάτοχος σπιτιού (MORTGAGE,RENT,Other)
7	home_ownership_cat	Δηλώνει αν ο πελάτης είναι κάτοχος σπιτιού αλλά εκφράζεται αριθμητικά (1-6)
8	income_category	Εκφράζει σε ποια κατηγορία εισοδήματος ανήκει ο κάτοχος του δανείου (Low,Medium,Other)
9	annual_inc	Εκφράζει το εισόδημα του πελάτη το χρόνο (0-9.5εκ.)
10	income_cat	Εκφράζει σε ποια κατηγορία εισοδήματος ανήκει ο κάτοχος του δανείου αλλά σε αριθμητική κλίμακα (1-3)
11	loan_amount	Εκφράζει το ποσό του δανείου (500-35.000)
12	term	Δηλώνει τη διάρκεια αποπληρωμής του δανείου (36 months,60 months)
13	term_cat	Δηλώνει τη διάρκεια αποπληρωμής του δανείου αλλά σε αριθμητική κλίμακα (1,2)
14	application_type	Εκφράζει αν ο δανειολήπτης είναι ένα άτομο ή περισσότερα (INDIVIDUAL,JOINT)
15	application_type_cat	Εκφράζει αν ο δανειολήπτης είναι ένα άτομο ή περισσότερα

		αλλά σε αριθμητική κλίμακα (1,2)
16	purpose	Δηλώνει το σκοπό του δανείου (debt_consolidation, Credit_card,Other)
17	purpose_cat	Δηλώνει το σκοπό του δανείου αλλά σε αριθμητική κλίμακα (1-14)
18	interest_payments	Εκφράζει το ποσό πληρωμής των τόκων σε μορφή δυαδικής κλίμακας (Low, High)
19	interest_payment_cat	Εκφράζει το ποσό πληρωμής των τόκων σε αριθμητική κλίμακα (1,2)
20	loan_condition	Δηλώνει αν το δάνειο έχει θετικό ή αρνητικό αντίκτυπο στην τράπεζα (Good Loan, Bad Loan)
21	loan_condition_cat	Δηλώνει αν το δάνειο έχει θετικό ή αρνητικό αντίκτυπο στην τράπεζα αλλά σε αριθμητική κλίμακα (0,1)
22	interest_rate	Εκφράζει το επιτόκιο (5.32-29%)
23	grade	Δηλώνει το βαθμό του δανείου (B,C,Other)
24	grade_cat	Δηλώνει το βαθμό του δανείου σε αριθμητική κλίμακα (1-7)
25	dti	Δηλώνει την αναλογία μεταξύ συνολικών μηνιαίων πληρωμών χρέους του οφειλέτη επί των συνολικών υποχρεώσεων χρέους
26	total_pymnt	Είναι το συνολικό ποσό που έχει πληρώσει ο οφειλέτης (0-57.8χιλ.)
27	total_rec_prncp	(0-35χιλ)
28	recoveries	Εκφράζει το ποσό της αποπληρωμής (0-33,5χιλ)
29	installment	Είναι το ποσό της δόσης που πληρώνει ο κάθε οφειλέτης (15.7-1.45χιλ)
30	region	Εκφράζει σε ποια περιοχή εκδόθηκε το εκάστοτε δάνειο (leinster,ulster,Other)

Προ-επεξεργασία δεδομένων

Ένα από τα σημαντικότερα βήματα στην ανάπτυξη του μοντέλου είναι η προ-επεξεργασία, δηλαδή ο καθαρισμός των δεδομένων. Αρχικά αξίζει να τονιστεί πως κάποιες στήλες του συνόλου δεδομένων αφαιρέθηκαν καθώς δεν θα ήταν χρήσιμες για την ανάπτυξη του μοντέλου. Όπως έχει παρατηρηθεί στον πιο πάνω πίνακα με την αναλυτική περιγραφή των δεδομένων, υπάρχει ένα σύνολο από χαρακτηριστικά που εκφράζονται και με αριθμητική μορφή και όχι κειμένου. Χαρακτηριστικά που έχουν μορφή κειμένου δεν μπορούν να είναι χρήσιμα για την εφαρμογή κάποιου αλγορίθμου, οπότε κρατάμε μόνο τις στήλες οι οποίες είναι εκφρασμένες με αριθμητική μορφή, όπως και έγινε. Σε περίπτωση που δεν υπάρχουν έτοιμες αυτές οι στήλες στο σύνολο δεδομένων, τις δημιουργούμε. Στο σύνολο δεδομένων δεν υπήρχαν κενές τιμές οπότε δεν χρειάστηκε να πραγματοποιηθεί κάποια ενέργεια για την αντικατάστασή τους. Στις εικόνες που ακολουθούν φαίνεται πως ήταν τα δεδομένα στην αρχική τους μορφή και πως είναι έπειτα από τον αποκλεισμό συγκεκριμένων χαρακτηριστικών.

	emp_length_int	home_ownership	home_ownership_cat	income_category	annual_inc	income_cat	loan_amount	term	term_cat	application_type
0	10.0	RENT	1	Low	24000	1	5000	36 months	1	INDIVIDUAL
1	0.5	RENT	1	Low	30000	1	2500	60 months	2	INDIVIDUAL
2	10.0	RENT	1	Low	12252	1	2400	36 months	1	INDIVIDUAL
3	10.0	RENT	1	Low	49200	1	10000	36 months	1	INDIVIDUAL
4	1.0	RENT	1	Low	80000	1	3000	60 months	2	INDIVIDUAL
5	3.0	RENT	1	Low	36000	1	5000	36 months	1	INDIVIDUAL
6	8.0	RENT	1	Low	47004	1	7000	60 months	2	INDIVIDUAL
7	9.0	RENT	1	Low	48000	1	3000	36 months	1	INDIVIDUAL
8	4.0	OWN	2	Low	40000	1	5600	60 months	2	INDIVIDUAL
9	0.5	RENT	1	Low	15000	1	5375	60 months	2	INDIVIDUAL

Εικόνα 23α: Σύνολο δεδομένων

interest_rate	grade	grade_cat	dti	total_pymnt	total_rec_prncp	recoveries	installment	loan_condition
10.65	B	2	27.65	5861.071414	5000.00	0.00	162.87	0
15.27	C	3	1.00	1008.710000	456.46	117.08	59.83	1
15.96	C	3	8.72	3003.653644	2400.00	0.00	84.33	0
13.49	C	3	20.00	12226.302210	10000.00	0.00	339.31	0
12.69	B	2	17.94	3242.170000	2233.10	0.00	67.79	0
7.90	A	1	11.20	5631.377753	5000.00	0.00	156.46	0
15.96	C	3	23.51	8136.840000	5110.85	0.00	170.08	0
18.64	E	5	5.35	3938.144334	3000.00	0.00	109.43	0
21.28	F	6	5.55	646.020000	162.02	189.06	152.39	1
12.69	B	2	18.08	1476.190000	673.48	269.29	121.45	1

Εικόνα 23β: Σύνολο δεδομένων

emp_length_int	home_ownership_cat	annual_inc	income_cat	term_cat	application_type_cat	purpose_cat	interest_payment_cat	interest_rate
0	10.0	1	24000	1	1	1	1	10.65
1	0.5	1	30000	1	2	1	2	15.27
2	10.0	1	12252	1	1	1	3	15.96
3	10.0	1	49200	1	1	1	4	13.49
4	1.0	1	80000	1	2	1	4	12.69
...	...	...	...	...	...	...	...	...
887374	8.0	1	31000	1	1	1	6	11.99
887375	10.0	3	79000	1	1	1	7	11.99
887376	5.0	1	35000	1	2	1	6	15.99
887377	1.0	1	64400	1	2	1	6	19.99
887378	10.0	1	100000	1	1	1	1	11.99

887379 rows x 16 columns

Εικόνα 24α: Σύνολο δεδομένων έπειτα από την αφαίρεση χαρακτηριστικών

grade_cat	dti	total_pymnt	total_rec_prncp	recoveries	installment	loan_condition
2	27.65	5861.071414	5000.00	0.00	162.87	0
3	1.00	1008.710000	456.46	117.08	59.83	1
3	8.72	3003.653644	2400.00	0.00	84.33	0
3	20.00	12226.302210	10000.00	0.00	339.31	0
2	17.94	3242.170000	2233.10	0.00	67.79	0
...	...	...	...	...	...	...
2	28.69	3971.880000	2944.53	0.00	332.10	0
2	3.90	9532.390000	7066.74	0.00	797.03	0
4	30.90	3769.740000	1844.50	0.00	316.07	0
5	27.19	3787.670000	1552.69	0.00	317.86	0
2	10.83	7943.760000	5889.06	0.00	664.20	0

Εικόνα 24β: Σύνολο δεδομένων έπειτα από την αφαίρεση χαρακτηριστικών

Επόμενο βήμα ήταν η κανονικοποίηση των δεδομένων. Ο τρόπος που επιλέχθηκε να γίνει αυτή η ενέργεια ήταν με τη τεχνική Min-Max. Αυτή η τεχνική είναι η πιο γνωστή, παρόλα αυτά υπάρχουν και άλλες. Πιο αναλυτικά, η ελάχιστη τιμή που μπορούν να πάρουν τα δεδομένα είναι 0 και η μέγιστη 1. Αυτό συμβαίνει καθώς τα δεδομένα πρέπει να είναι όλα σε ίδια κλίμακα ώστε να μην επηρεάζουν, στη συνέχεια, το τελικό αποτέλεσμα πιθανές υψηλές τιμές. Με αυτό το τρόπο διαφυλάσσεται η αξιοπιστία του μοντέλου. Με άλλα λόγια, οι τιμές θα είναι πιο “δίκαιες”. Αυτή η τεχνική επιτυγχάνεται με τον τύπο
$$X = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$
, όπου X είναι η κάθε τιμή που επρόκειτο να κανονικοποιηθεί.

Παρόλο αυτά, κατά τη διάρκεια της ανάλυσης των δεδομένων παρατηρήθηκε πως στη στήλη “loan_condition” υπήρχε μεγάλη διαφορά στο σύνολο των καλών(0) δανείων με αυτά των κακών(1). Το σύνολο των καλών στα ιστορικά δεδομένα ήταν 819.950 και των κακών 67.429. Αυτό σημαίνει πως στο μέλλον, το μοντέλο θα υπερεκπαιδευτεί και η ακρίβεια του θα είναι πλασματική. Για την επίλυση αυτού του προβλήματος επιλέγουμε τυχαία τον ίδιο αριθμό δειγμάτων και από τις δύο κατηγορίες, ώστε στη συνέχεια το μοντέλο να εκπαιδευτεί σωστά. Στην εικόνα που ακολουθεί εμφανίζονται οι εντολές που χρησιμοποιούνται για την ολοκλήρωση αυτού του βήματος.

```
#Preprocess the data
data= dataset.drop(columns=['home_ownership','purpose','income_category','loan_amount','term','application',
data.columns

#We normalize the data with Min-Max technique. We can do that also with sklearn library with MinMaxScaler.
#The mathematical formulation is: x-min(x) / (max(x) - min(x))
scaled_data = (data-data.min(axis=0)) / (data.max(axis=0)-data.min(axis=0))
data
```

Εικόνα 25: Κανονικοποίηση δεδομένων

	emp_length_int	home_ownership_cat	annual_inc	income_cat	term_cat	application_type_cat	purpose_cat	interest_payment_cat	interest_rate
0	10.0	1	24000	1	1	1	1	1	10.65
1	0.5	1	30000	1	2	1	2	2	15.27
2	10.0	1	12252	1	1	1	3	2	15.96
3	10.0	1	49200	1	1	1	4	2	13.49
4	1.0	1	80000	1	2	1	4	1	12.69
...	...	...	...	...	...	...	...	...	...
887374	8.0	1	31000	1	1	1	6	1	11.99
887375	10.0	3	79000	1	1	1	7	1	11.99
887376	5.0	1	35000	1	2	1	6	2	15.99
887377	1.0	1	64400	1	2	1	6	2	19.99
887378	10.0	1	100000	1	1	1	1	1	11.99

887379 rows × 10 columns

Εικόνα 26α: Κανονικοποιημένα δεδομένα

grade_cat	dti	total_pymnt	total_rec_prncp	recoveries	installment	loan_condition
2	27.65	5861.071414	5000.00	0.00	162.87	0
3	1.00	1008.710000	456.46	117.08	59.83	1
3	8.72	3003.653644	2400.00	0.00	84.33	0
3	20.00	12226.302210	10000.00	0.00	339.31	0
2	17.94	3242.170000	2233.10	0.00	67.79	0
...	...	...	...	...	...	...
2	28.69	3971.880000	2944.53	0.00	332.10	0
2	3.90	9532.390000	7066.74	0.00	797.03	0
4	30.90	3769.740000	1844.50	0.00	316.07	0
5	27.19	3787.670000	1552.69	0.00	317.86	0
2	10.83	7943.760000	5889.06	0.00	664.20	0

Εικόνα 26β: Κανονικοποιημένα δεδομένα

Τελευταίο βήμα πριν από τις δοκιμές των αλγορίθμων είναι να χωριστούν τα δεδομένα σε για εκπαίδευση (train) και αξιολόγηση (test). Αυτή η διαδικασία, επιτεύχθηκε μέσω της βιβλιοθήκης scikit-learn. Το 70% των δεδομένων χρησιμοποιήθηκαν για εκπαίδευση και φυσικά το 30% για αξιολόγηση.

```
#We randomly take the same number of good and bad loan, due to the imbalanced number of records
#between the good and bad loan class

good_loan= scaled_data[scaled_data['loan_condition']==0]
bad_loan= scaled_data[scaled_data['loan_condition']==1]

random_good_loan = good_loan.sample(n=67429)
random_bad_loan = bad_loan.sample(n=67429)

new_data = small_dataset = pd.concat([random_good_loan,random_bad_loan])

#Split the data
X = new_data.drop(columns=['loan_condition'])
y = new_data.loan_condition
#Train(70%) and test(30%)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 100)
```

Εικόνα 27: Διαχωρισμός δεδομένων

Από τη στιγμή που όλα τα δεδομένα έχουν αριθμητική τιμή και έχουν έρθει στη κατάλληλη μορφή μπορούν να ξεκινήσουν τα πειράματα μάθησης.

Πειράματα μάθησης

Υπάρχουν αρκετοί αλγόριθμοι που μπορούν να χρησιμοποιηθούν για τη σωστή ταξινόμηση. Μερικοί από αυτούς είναι οι Decision Tree, k-NN, Naive Bayes, Support Vector Machine(SVM). Στην εργασία αναπτύχθηκαν μερικοί από αυτούς. Αρχικά αξίζει να τονιστεί πως τα πειράματα πραγματοποιήθηκαν με τη γλώσσα προγραμματισμού Python και τις κατάλληλες βιβλιοθήκες. Επίσης, για κάθε μοντέλο αναγράφονται οι μετρήσεις Accuracy, Precision , Recall καθώς και ο πίνακας

confusion matrix. Επιπλέον, για κάθε αλγόριθμο εφαρμόστηκε η τεχνική kfold cross validation με k=10. Οι αλγόριθμοι δοκιμάστηκαν με τις εξής γραμμές κώδικα:

```
#We have to import the relevant libraries and run the algorithms into scoring  
#function  
from sklearn.naive_bayes import GaussianNB, BernoulliNB  
gaussian = GaussianNB()  
bernoulli = BernoulliNB()  
  
from sklearn.tree import DecisionTreeClassifier  
tree = DecisionTreeClassifier()  
  
from sklearn.linear_model import LogisticRegression  
log_reg = LogisticRegression()  
  
from sklearn.neighbors import KNeighborsClassifier  
KNN= KNeighborsClassifier(n_neighbors = 3)  
  
from sklearn.ensemble import RandomForestClassifier  
forest = RandomForestClassifier(n_estimators=100)  
  
scoring(bernoulli)
```

Εικόνα 28: Δοκιμή αλγορίθμων μηχανικής μάθησης

```
#We are ready now to try some algorithms  
#We create a function scoring with the purpose of running the algorithms  
def scoring(model):  
    model.fit(X_train, y_train)  
    model_pred = model.predict(X_test)  
    kfold = cross_val_score(model,X,y,cv=10)  
    cnf_matrix = metrics.confusion_matrix(y_test, model_pred)  
    print(cnf_matrix)  
    print('Accuracy score:',accuracy_score(y_test, model_pred))  
    print('Precision:',metrics.precision_score(y_test, model_pred))  
    print('Recall:',metrics.recall_score(y_test, model_pred))  
    print('10-fold: \t\n',kfold)
```

Εικόνα 29: Συνάρτηση για την υλοποίηση των αλγορίθμων

Πιο αναλυτικά, με τον όρο Confusion Matrix εννοούμε τη μονάδα μέτρησης που δείχνει πως έχουν ταξινομηθεί τα δείγματα, από τον εκάστοτε αλγόριθμο. Αυτό επιτυγχάνεται μέσα από τα TP(True Positive), TN(True Negative), FP(False Positive), FN(False Negative). True Positive είναι τα δείγματα τα οποία έχουν ταξινομηθεί ορθά ως σωστά ενώ False Positive το αντίστροφο, δηλαδή τα δείγματα που έπρεπε να ταξινομηθούν ως λάθος αλλά ταξινομήθηκαν ως σωστά. Το ίδιο ισχύει για τα True Negative και False Negative. True Negative είναι τα δείγματα που ταξινομήθηκαν ορθά ως λάθος ενώ False Negative αυτά που ταξινομήθηκαν ως σωστά αλλά έπρεπε να ταξινομηθούν ως λάθος. Με βάση αυτόν τον πίνακα προκύπτουν κάποιες μετρήσεις που βοηθάν τον αναλυτή να πάρει σωστές

αποφάσεις για το μοντέλο. Οι πιο συνηθισμένες είναι οι Accuracy, Precision και Recall. Ωστόσο υπάρχουν και άλλες μετρήσεις που είναι πιο εξειδικευμένες.

Πίνακας 26: Τεχνική Confusion Matrix

	Πραγματικά δείγματα (1)	Πραγματικά δείγματα (0)
Ταξινομημένα δείγματα ως (1)	True Positives (TPs)	False Positives (FPs)
Ταξινομημένα δείγματα ως (0)	False Negatives (FNs)	True Negatives (TNs)

- **Accuracy:** Με τον όρο accuracy εννοούμε πόσο κοντά είναι η προβλεπόμενες τιμές με τις αντίστοιχες πραγματικές (αληθινές) τιμές. Όταν το accuracy είναι υψηλό τότε σημαίνει πως το μοντέλο έχει προβλέψει τιμές πολύ κοντά στις αληθινές. Αυτό το ποσοστό προκύπτει από τη σχέση:
$$\frac{TP + TN}{TP + TN + FP + FN}$$
- **Precision:** Με αυτόν τον όρο εννοούμε πόσο κοντά είναι οι τιμές μεταξύ τους. Με πιο απλά λόγια, δείχνει πόσο συχνά ταξινομούνται σωστά τα TP. Το precision προκύπτει από την εξής σχέση:
$$\frac{TP}{TP + FP}$$
- **Recall ή Sensitivity:** Δηλαδή ευαισθησία, είναι στην ουσία το ποσοστό των πραγματικών σωστών δειγμάτων που ταξινομήθηκαν ως σωστά. Πιο απλά, πόσο συχνά έχει προβληθεί ένα δείγμα ως σωστό όταν πραγματικά είναι σωστό. Αυτό προκύπτει από τον τύπο:
$$\frac{TP}{TP + FN}$$

K-fold cross validation

Στο K-fold cross validation τα δεδομένα κατανέμονται σε K, στην προκειμένη περίπτωση 10, εξίσου ίσα τμήματα. Σε αυτά τα τμήματα, η εκπαίδευση και ο έλεγχος (train και test) γίνονται σε επαναλήψεις, έτσι ώστε σε κάθε μία επανάληψη, το K τμήμα να ελέγχεται (test) και στα K-1 τμήματα που έχουν απομείνει να εκπαιδεύεται (train).

Πίνακας 27: Τεχνική k-Fold Cross Validation

DATA					
1	VALIDATE	TRAIN	TRAIN	TRAIN	TRAIN
2	TRAIN	VALIDATE	TRAIN	TRAIN	TRAIN
3	TRAIN	TRAIN	VALIDATE	TRAIN	TRAIN
...	TRAIN	TRAIN	TRAIN	VALIDATE	TRAIN
K	TRAIN	TRAIN	TRAIN	TRAIN	VALIDATE

Αποτελέσματα αλγορίθμων

Logistic Regression

Πίνακας 28: Confusion Matrix του αλγορίθμου Logistic Regression

Logistic Regression - Confusion Matrix		
Good loan	14233	5882
Bad loan	6569	13774
	Good loan	Bad loan
Αποτελέσματα		
Accuracy	Precision	Recall
73%	74.6%	70.1%

Decision Tree

Πίνακας 29: Confusion Matrix του αλγορίθμου Decision Tree

Decision Tree - Confusion Matrix		
Good loan	14229	5886
Bad loan	7249	13094
	Good loan	Bad loan
Αποτελέσματα		
Accuracy	Precision	Recall
67.25%	68.9%	64.3%

Random Forest

Πίνακας 30: Confusion Matrix του αλγορίθμου Random Forest

Random Forest - Confusion Matrix		
Good loan	14018	6097
Bad loan	6160	14183
	Good loan	Bad loan
Αποτελέσματα		
Accuracy	Precision	Recall
78.77%	82.43%	73.44%

K-Nearest Neighbor

Πίνακας 31: Confusion Matrix του αλγορίθμου k-Nearest Neighbor

KNN - Confusion Matrix		
Good loan	13590	6525
Bad loan	6983	13360
	Good loan	Bad loan
Αποτελέσματα		
Accuracy	Precision	Recall
66.6%	67.1%	65.6%

Naive Bayes

Πίνακας 32: Confusion Matrix του αλγορίθμου Naive Bayes (Bernoulli)

Naive Bayes - Confusion Matrix Bernoulli		
Good loan	13214	6901
Bad loan	5623	14720
	Good loan	Bad loan
Αποτελέσματα		
Accuracy	Precision	Recall
69%	68%	72.3%

Πίνακας 33: Confusion Matrix του αλγορίθμου Naive Bayes (Gaussian)

Naive Bayes - Confusion Matrix Gaussian		
Good loan	20115	0
Bad loan	12820	7523
	Good loan	Bad loan
Αποτελέσματα		
Accuracy	Precision	Recall
68.3%	100%	36.9%

Αναλυτική περιγραφή μοντέλου

Στην προηγούμενη ενότητα παρουσιάστηκαν τα αποτελέσματα από τα πειράματα μάθησης. Κάποιοι αλγόριθμοι από αυτούς ήταν ενδιαφέρον και άλλα δεν είχαν τα επιθυμητά αποτελέσματα. Κατά τη διάρκεια της διαδικασίας των πειραμάτων τα πιο ικανοποιητικά αποτελέσματα τα είχε ο “Random forest”. Ο Random Forest είναι ένας αλγόριθμος ο οποίος κατατάσσεται στη κατηγορία των δέντρων. Όμως έχει μία ιδιαιτερότητα που τον κάνει να ξεχωρίζει και αυτή είναι ο τρόπος που λειτουργεί. Πολλά δέντρα απόφασης επεξεργάζονται τα δεδομένα και προτείνουν μία πρόβλεψη, στη προκειμένη περίπτωση “καλό” ή “κακό” δάνειο. Τα δέντρα στα οποία η απόφαση τους θα συμφωνεί πλειοψηφικά, δίνουν και τη τελική πρόβλεψη του αλγορίθμου. Για αυτό το λόγο ονομάζεται τυχαίος δάσος. Πιο συγκεκριμένα, τα βήματα για να κατασκευαστεί ένα τυχαίο δάσος είναι τα εξής:

1. Επιλέγουμε έναν τυχαίο αριθμό k χαρακτηριστικών από τα m συνολικά, όπου το k ανήκει στο m .
2. Στη συνέχεια, δημιουργούμε δέντρο απόφασης. Υπολογίζουμε το κόμβο $D(\text{root node})$ χρησιμοποιώντας το καλύτερο σημείο διαχωρισμού.
3. Βρίσκουμε τα παιδιά του κόμβου(D) στο σωστό σημείο διαχωρισμού.
4. Επαναλαμβάνουμε τα βήματα 2 και 3 μέχρι να φτάσουμε στα φύλλα του των κόμβων.
5. Τέλος, επαναλαμβάνονται τα βήματα από το 1 έως το 4 για n φορές ώστε να δημιουργηθεί ένας αριθμός n από δέντρα.

Όπως έχει αναφερθεί, ο αλγόριθμος Random Forest δημιουργεί έναν αριθμό από δέντρα απόφασης. Σύμφωνα με επιστήμονες¹⁵, Σε κάθε δέντρο που δημιουργείται, η “Scikit-learn” υπολογίζει τη σημασία κάθε κόμβου χρησιμοποιώντας το Gini. Το Gini είναι ένα στατιστικό μέτρο του βαθμού διακύμανσης ενός συνόλου τιμών. Για ένα δυαδικό δέντρο ισχύει ο εξής τύπος:

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)}$$

- ni_j = Η σημασία του κόμβου j
- w_j = Ο σταθμισμένος αριθμός των δειγμάτων που φτάνουν το κόμβο j
- C_j = Η μη καθαρές τιμές του κόμβου j
- $left(j)$ = Τα παιδιά από τα αριστερά του κόμβου j
- $right(j)$ = Τα παιδιά από τα δεξιά του κόμβου j

Στη συνέχεια υπολογίζεται η σημασία κάθε χαρακτηριστικού σε κάθε δέντρο απόφασης. Αυτό επιτυγχάνεται με τον τύπο που ακολουθεί.

$$fi_i = \frac{\sum ni_j}{\sum ni_k}$$

- fi_i = Η σημασία του χαρακτηριστικού i
- ni_j = Η σημασία του κόμβου j
- j = Κόμβος j σε κάθε χαρακτηριστικό i
- k = Αριθμός όλων των κόμβων

Έπειτα πραγματοποιείται κανονικοποίηση στα αποτελέσματα σημασίας για κάθε δέντρο, με τον εξής μαθηματικό τύπο.

$$normfi_i = \frac{fi_i}{\sum_j fi_j}$$

- $normfi_i$ = Κανονικοποίηση της σημασίας του χαρακτηριστικού i

¹⁵ Stracey Ronaghan (2018), The Mathematics of Decision Trees, Random Forest and Feature Importance in Scikit-learn and Spark, Towards data science

- $f\hat{i}_i$ = Η σημασία του χαρακτηριστικού i

Η καθοριστική σημασία χαρακτηριστικού, είναι η μέση τιμή (όλων των σημασιών) από όλα τα δέντρα. Υπολογίζεται το άθροισμα σημασίας των χαρακτηριστικών από κάθε δέντρο και στη συνέχεια διαιρείται από το συνολικό αριθμό των δέντρων, Αυτό φαίνεται στο τύπο που αναγράφεται από κάτω.

$$RFf\hat{i}_i = \frac{\sum_j normf\hat{i}_{ij}}{T}$$

- $RFf\hat{i}_i$ = Η σημασία του χαρακτηριστικού i από το σύνολο των δέντρων στο μοντέλο του Random Forest
- $normf\hat{i}_{ij}$ = Η κανονικοποιημένη σημασία του χαρακτηριστικού για i στο δέντρο j
- T = Ο συνολικός αριθμός των δέντρων

Με αυτό το τρόπο πραγματοποιείται η υλοποίηση των δέντρων απόφασης και υπολογίζεται η σημασία των χαρακτηριστικών από τη βιβλιοθήκη “Scikit-learn”.

Εφαρμογή

Η ανάπτυξη της εφαρμογής πραγματοποιήθηκε με ένα νέο framework της python με τίτλο “Streamlit”. Το Streamlit επιτρέπει τους προγραμματιστές να δημιουργήσουν μία διαδικτυακή εφαρμογή εύκολα μέσω της python.

Το πρώτο βήμα που είναι αναγκαίο να γίνει είναι η δημιουργία μίας φόρμας εισαγωγής δεδομένων από το χρήστη. Στη φόρμα ζητείται από το χρήστη να συμπληρώσει μερικά δανειοληπτικά στοιχεία, τα οποία εκφράζουν τα χαρακτηριστικά του συνόλου δεδομένων. Η διαδικασία αυτή πραγματοποιήθηκε με τον τρόπο που φαίνεται στην εικόνα που ακολουθεί.

```
def user_input_features():
    emp_length_int = st.number_input('Employment length in years', X.emp_length_int.min(), X.emp_length_int.max())

    home_ownership = st.radio('Home Ownership', ('Rent', 'Own', 'Mortgage', 'Other', 'None'))
    if home_ownership == 'Rent':
        home_ownership_cat = 1
    elif home_ownership == 'Own':
        home_ownership_cat = 2
    elif home_ownership == 'Mortgage':
        home_ownership_cat = 3
    elif home_ownership == 'Other':
        home_ownership_cat = 4
    else:
        home_ownership_cat = 5

    annual_inc = st.radio('Annual Income', ('<=10000', '10000-20000', '>20000'))
    if annual_inc == '<=10000':
        income_cat = 1
    elif annual_inc == '10000-20000':
        income_cat = 2
    else:
        income_cat = 3

    term = st.radio('Term', ('36 months', '60 months'))
    if term == '36 months':
        term_cat = 1
    else:
        term_cat = 2

    application_type = st.radio('Application Type', ('Individual', 'Joint'))
    if application_type == 'Individual':
        application_type_cat = 1
    else:
        application_type_cat = 2
```

Εικόνα 30α: Δημιουργία φόρμας εισαγωγής δεδομένων

```
purpose = st.radio('Purpose', ('Credit card', 'Car', 'Small business', 'Debt consolidation', 'Wedding', 'Other'))
if purpose == 'Credit card':
    purpose_cat = 1
elif purpose == 'Car':
    purpose_cat = 2
elif purpose == 'Small business':
    purpose_cat = 2
elif purpose == 'Other':
    purpose_cat = 3
elif purpose == 'Wedding':
    purpose_cat = 4
else:
    purpose_cat = 5

interest_payments = st.radio('Interest paymentnts', ('Low', 'High'))
if interest_payments == 'Low':
    interest_payment_cat = 1
else:
    interest_payment_cat = 2

interest_rate = st.number_input('Interest Rate', X.interest_rate.min(), X.interest_rate.max())
grade = st.radio('Grade', ('A', 'B', 'C', 'D', 'E', 'F', 'G'))
if grade == 'A':
    grade_cat = 1
elif grade == 'B':
    grade_cat = 2
elif grade == 'C':
    grade_cat = 3
elif grade == 'D':
    grade_cat = 4
elif grade == 'E':
    grade_cat = 5
elif grade == 'F':
    grade_cat = 6
else:
    grade_cat = 7

dti = st.slider('Debt in Income', X.dti.min(), 10000.00)
total_pymnt = st.slider('Total Payment', X.total_pymnt.min(), 50000.00)
recoveries = st.slider('Recoveries', X.recoveries.min(), 100000.00)
installment = st.slider('Installment', X.installment.min(), 500000.00)
```

Εικόνα 30β: Δημιουργία φόρμας εισαγωγής δεδομένων

Επόμενο βήμα μετά τη δημιουργία της φόρμας είναι να περαστούν τα δεδομένα του σε μία λίστα ώστε να μπορέσουν στη συνέχεια να χρησιμοποιηθούν στον αλγόριθμο πρόβλεψης. Αυτό επιτυγχάνεται με τον εξής τρόπο:

```

data = {'emp_length_int': emp_length_int,
        'home_ownership_cat': home_ownership_cat,
        'income_cat': income_cat,
        'term_cat': term_cat,
        'application_type_cat': application_type_cat,
        'purpose_cat': purpose_cat,
        'interest_payment_cat': interest_payment_cat,
        'interest_rate': interest_rate,
        'grade_cat': grade_cat,
        'dti': dti,
        'total_pymnt': total_pymnt,
        'recoveries': recoveries,
        'installment': installment
        }
features = pd.DataFrame(data, index=[0])
return features

```

Εικόνα 31: Εισαγωγή της εγγραφής σε DataFrame

Όπως φαίνεται στην εικόνα τα δεδομένα περάστηκαν σωστά σε μία λίστα και είναι έτοιμα ώστε να εφαρμοστούν στον αλγόριθμο. Το μοντέλο έχει αποθηκευτεί στο δίσκο ώστε να μπορεί να χρησιμοποιηθεί στην διαδικτυακή εφαρμογή. Με αυτό τον τρόπο στην εφαρμογή επιτυγχάνονται πιο γρήγοροι χρόνοι, καθώς δεν χρειάζεται να πραγματοποιήσει τους υπολογισμούς του μοντέλου ξανά. Επομένως, ορίζουμε σε μια μεταβλητή τη νέα εγγραφή ώστε να είναι πιο απλός ο κώδικας και κανονικοποιούμε τα δεδομένα ώστε να είναι στην ίδια μορφή με αυτά του μοντέλου. Επίσης, εμφανίζουμε στο χρήστη την εγγραφή που πραγματοποίησε

```

df = user_input_features()
scaled_data = (df.min(axis=0)) / (df.max(axis=0)-df.min(axis=0))

st.header('Specified Input parameters')
st.dataframe(df)
st.write('---')

```

Εικόνα 32: Εμφάνιση της εγγραφής στο χρήστη

Το μοντέλο αποθηκεύτηκε με την βοήθεια της βιβλιοθήκης “pickle” και η εντολή που πραγματοποιεί αυτή την ενέργεια είναι η εξής:

```

import pickle
pickle.dump(forest, open('final_model.sav', 'wb'))

```

Εικόνα 33: Αποθήκευση μοντέλου

Όπως φαίνεται με αυτό τρόπο το μοντέλο του Random Forest έχει αποθηκευτεί, οπότε το επόμενο βήμα είναι να εφαρμοστεί το νέο δείγμα στο μοντέλο ώστε να επιτευχθεί η πρόβλεψη για το νέο δείγμα. Αυτή ενέργεια πραγματοποιήθηκε με την ίδια βιβλιοθήκη “pickle”. Επιπλέον, εμφανίζονται τα αποτελέσματα του μοντέλου που υπολογίστηκαν στο πειραματικό στάδιο.

```
import pickle
loaded_model = pickle.load(open('final_model.sav', 'rb'))

user_prediction = loaded_model.predict(df)

st.header('Prediction')

if user_prediction == 0:
    st.success('That will be a good loan')
elif user_prediction == 1:
    st.error('That will be a bad loan')

st.write('Accuracy: 78.77%')
st.write('Precision: 82.43%')
st.write('Recall: 73.44%')
```

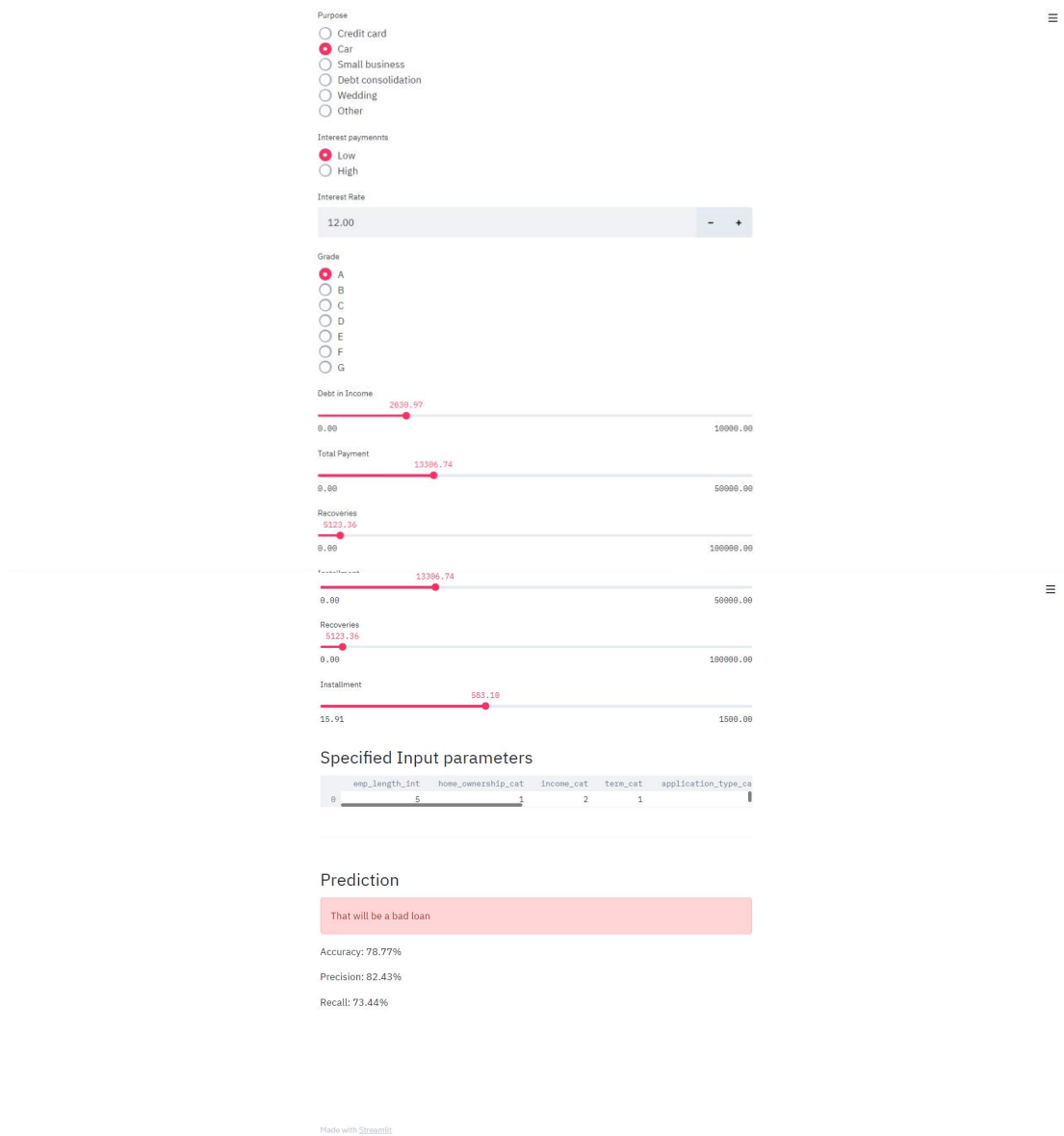
Εικόνα 34: Φόρτωση μοντέλου στην εφαρμογή για πρόβλεψη και εμφάνιση αποτελεσμάτων

Τέλος, ακολουθούν εικόνες με την απεικόνιση όλων των προαναφερθέντων όπως φαίνεται στην εφαρμογή που χρησιμοποιεί ο χρήστης. Αρχικά δοκιμάστηκαν τιμές οι οποίες είχαν ως αποτέλεσμα “Bad loan”. Έπειτα, εφαρμόστηκαν τιμές που είχαν σαν αποτέλεσμα “Good loan”.

The screenshot displays the 'Credit Scoring' application interface. At the top, the title 'Credit Scoring' is followed by a subtitle: 'This app predicts the acceptance or the rejection of a petition for a loan'. Below this, the 'Input Parameters' section contains several form elements:

- Employment length in years:** A numeric input field with the value '5.00' and minus/plus buttons.
- Home Ownership:** A group of radio buttons with 'Rent' selected.
- Annual Income:** A group of radio buttons with '10000-20000' selected.
- Term:** A group of radio buttons with '36 months' selected.
- Application Type:** A group of radio buttons with 'Individual' selected.

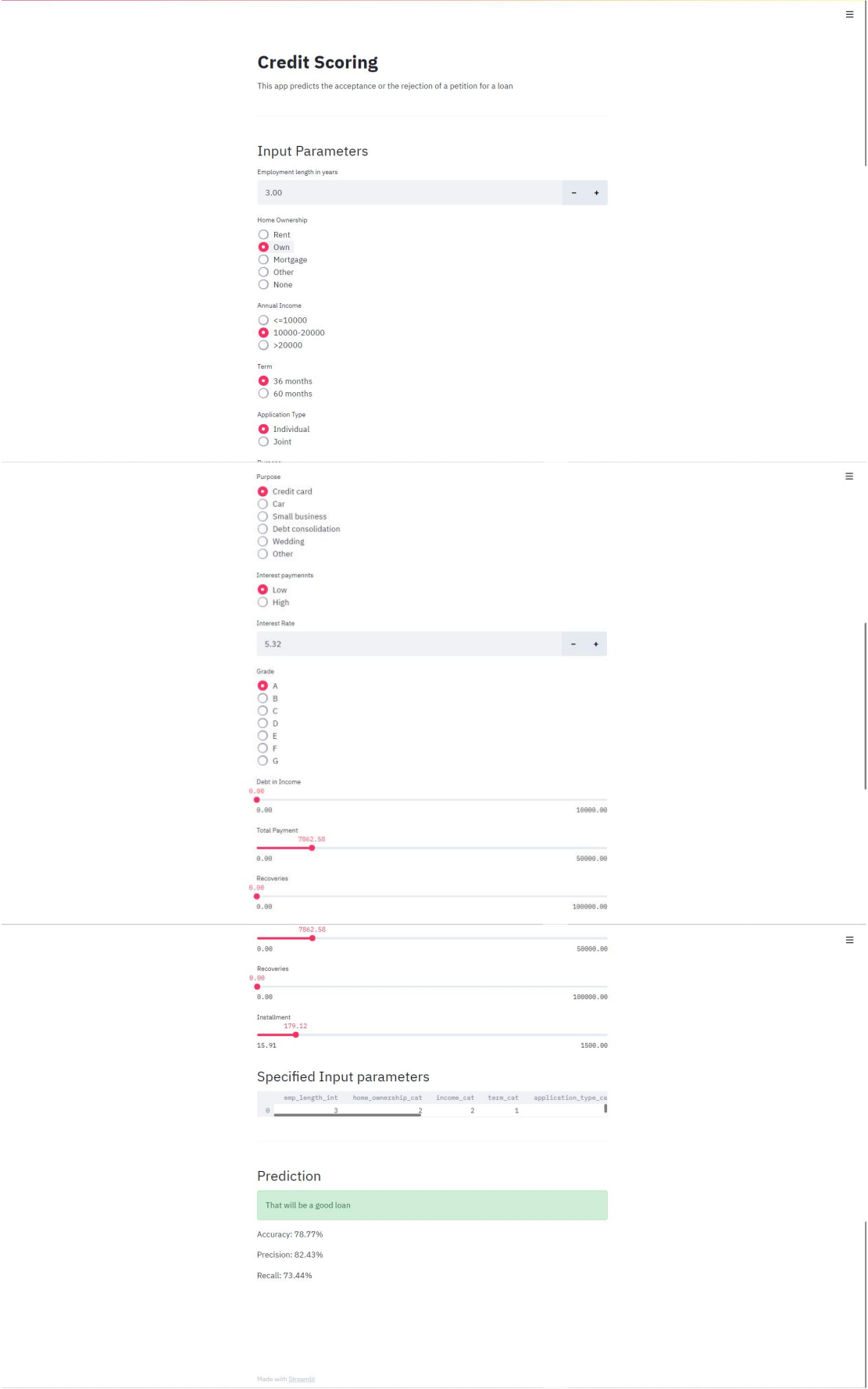
At the bottom of the form, there is a 'Predict' button.



Εικόνα 35: Παράδειγμα πρόβλεψης κακού δανειολήπτη

	emp_length_int	home_ownership_cat	income_cat	term_cat	application_type_cat	purpose_cat	interest_payment_cat	interest_rate	grade_cat	dti	total_pymnt	recoveries	installment
0	5	1	2	1	1	2	1	12	1	2,839.9799	13,396.7499	5,123.3699	883.1899

Εικόνα 36: Εμφάνιση εγγραφής χρήστη



Εικόνα 37: Παράδειγμα πρόβλεψης καλού δανειολήπτη

	emp_length_int	home_ownership_cat	income_cat	term_cat	application_type_cat	purpose_cat	interest_payment_cat	interest_rate	grade_cat	dti	total_pymnt	recoveries	installment
0	3	2	2	1	1	1	1	5.3200	1	0	7,862.5800	0	179.1200

Εικόνα 38: Εμφάνιση εγγραφής χρήστη

Συμπεράσματα

Η επιστήμη των δεδομένων όπως φαίνεται και από τη συγκεκριμένη έρευνα μπορεί να είναι αρκετά χρήσιμη για την μείωση του ρίσκου για τον αν κάποιος υποψήφιος δανειολήπτης θα είναι αξιόπιστος ή όχι. Μέσω των αλγορίθμων μηχανικής μάθησης μπορούν να παρθούν αποφάσεις πολύ πιο γρήγορα συγκριτικά με παραδοσιακές τεχνικές. Υπάρχουν αρκετές έρευνες οι οποίες έχουν δοκιμάσει διάφορους αλγορίθμους και τεχνικές οι οποίες είχαν ενδιαφέρον αποτελέσματα.

Η συγκεκριμένη εργασία είχε σκοπό την ανάπτυξη ενός μοντέλου το οποίο θα πραγματοποιεί πρόβλεψη της κατάστασης ενός δανείου για έναν πελάτη σε μία εμπορική τράπεζα. Εφαρμόστηκαν συνολικά έξι αλγόριθμοι με διαφορετική λογική ο καθένας. Αυτός που είχε τα πιο εντυπωσιακά αποτελέσματα ήταν ο Random Forest με Accuracy: 78.77%, Precision: 82.43% και Recall: 73.44%. Νούμερα αρκετά ικανοποιητικά συγκριτικά με τους υπόλοιπους αλγόριθμους που συμμετείχαν στο πείραμα.