ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ
ΤΜΗΜΑ ΟΡΓΑΝΩΣΗΣ & ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ

ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΣΤΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ ΜΕ ΤΙΤΛΟ:

«ΣΥΣΤΗΜΑΤΑ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ ΣΤΑ ΧΡΗΜΑΤΟΠΙΣΤΩΤΙΚΑ ΙΔΡΥΜΑΤΑ»

Αθανάσιος Οικονόμου

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

Επιβλέπων καθηγητής
Ευάγγελος Καλαμπόκης

ΘΕΣΣΑΛΟΝΙΚΗ
2022

Περίληψη

Στην παρούσα διπλωματική εργασία με τίτλο «Συστήματα μηχανικής μάθησης στα χρηματοπιστωτικά ιδρύματα», γίνεται εφαρμογή της τεχνητής νοημοσύνης στο χρηματοπιστωτικό τομέα.

Μία από τις βασικές υπηρεσίες των χρηματοπιστωτικών ιδρυμάτων, είναι η χορήγηση δανείων. Ο αριθμός των αιτήσεων που λαμβάνονται για δάνεια είναι μεγάλος και ο προβληματισμός είναι αν ένα δάνειο θα αποπληρωθεί εφόσον η αίτησή του εγκριθεί. Με κάθε νέα αίτηση, συγκεντρώνεται μία πληθώρα στοιχείων σχετικά με τον αιτούντα, συνδυασμός των οποίων ενδέχεται να είναι αποτρεπτικός ή όχι στο να εγκριθεί μία αίτηση.

Στα πλαίσια της διπλωματικής εργασίας, αξιοποιήθηκαν τα σύνολα δεδομένων που διέθεσε η Home Credit στην πλατφόρμα της Kaggle, με σκοπό τη διενέργεια διαγωνισμού μηχανικής μάθησης με θέμα «Home Credit Default Risk. Can you predict how capable each applicant is of repaying a loan?».

Με την κατανόηση των δεδομένων, ακολούθησε η διερευνητική ανάλυση αυτών, συνοδευόμενη από γραφικές απεικονίσεις με τη χρήση της εφαρμογής Tableau.

Στη φάση της προεργασίας, δημιουργήθηκαν νέες παράμετροι οι οποίες ενσωματώθηκαν με τις υφιστάμενες σε ένα τελικό σύνολο δεδομένων. Μερικά από τα δεδομένα ήταν αναγκαίο να μετατραπούν ώστε να είναι δυνατή η περαιτέρω επεξεργασία τους.

Για το μοντέλο που δημιουργήθηκε, αξιοποιήθηκε η ενισχυτική διαβάθμιση δέντρων αποφάσεων (GBDT) με τη χρήση της βιβλιοθήκης XGBoost (Extreme Gradient Boosting). Ο συντονισμός των υπερπαραμέτρων επιτεύχθηκε με την Αναζήτηση Πλέγματος (Grid Search) της βιβλιοθήκης Scikit-learn. Η διαχείριση των πινάκων έγινε με συναρτήσεις της βιβλιοθήκης NumPy η οποία διαθέτει μια μεγάλη συλλογή μαθηματικών συναρτήσεων. Για τη δημιουργία γραφημάτων, χρησιμοποιήθηκε η συλλογή συναρτήσεων pyplot της βιβλιοθήκης matplotlib.

Το μοντέλο αξιολογήθηκε με το ποσοστό του AUC ROC score (Area Under the ROC (Receiver Operating Characteristic) Curve). Επιπλέον, έγινε η ανεύρεση των σημαντικότερων από τις παραμέτρους του συνόλου δεδομένων.

Η μηχανική μάθηση, αξιοποιώντας σωστά τα ακριβή και μεγάλα σύνολα δεδομένων, μπορεί να συμβάλλει σημαντικά στη πρόγνωση των τομέων της οικονομίας.

Λέξεις κλειδιά: σύνολο δεδομένων, μηχανική μάθηση, τεχνητή νοημοσύνη, επιστήμη δεδομένων, αλγόριθμος μηχανικής μάθησης, λήψη αποφάσεων, tableau, μεγάλα δεδομένα, επιχειρηματική ευφυία, python, jupyter notebook, kaggle, csv, home credit, grid search, xgboost

Abstract

In present dissertation "Machine learning systems in financial institutions", artificial intelligence is applied in the financial sector.

One of the basic services of financial institutions is the granting of loans. The number of loan applications received is high and the concern is whether a loan will be repaid if its application is approved. With each new application, many information about the applicant is collected, the combination of which may or may not prevent an application from being approved.

In the context of this diploma work, the datasets used, were available on the Kaggle platform where Home Credit offered them, in order to conduct a machine learning competition on the topic of "Home Credit Default Risk. Can you predict how capable each applicant is of repaying a loan?".

With the understanding of the data, the exploratory analysis followed, accompanied by graphical representations using the Tableau application.

In the preprocessing phase, new parameters were created and integrated with the existing ones into a final data set. Some of the data needed to be converted to enable further processing.

For the generated model, gradient boosting decision trees (GBDT) was leveraged using the XGBoost (Extreme Gradient Boosting) library. Tuning of the hyperparameters was achieved with the Grid Search of the Scikit-learn library. The tables were managed with functions from the NumPy library which has a large collection of mathematical functions. The pyplot function collection of the matplotlib library was used to generate plots.

The model was evaluated with the percentage of the AUC ROC score (Area Under the ROC (Receiver Operating Characteristic) Curve). In addition, the most important parameters of the data set were found.

By making proper use of accurate and large data sets, machine learning can significantly contribute to the forecasting of sectors of the economy.

Keywords: data set, machine learning, Artificial Intelligence, data science, machine learning algorithm, decision making, tableau, big data, business intelligence, python, jupyter notebook, kaggle, csv, home credit, grid search, xgboost

Ευχαριστίες

Ένα μεγάλο ευχαριστώ

στους γονείς μου Πέτρο και Μαρία για αυτό που έγινα,
στη σύζυγό μου Μαρία που με δέχτηκε όπως ήμουν,
στην κόρη μου Ελένη που με έκανε καλύτερο άνθρωπο,
σε όλους τους καθηγητές μου ανεξαιρέτως που με βοήθησαν να εξελιχθώ
και ιδιαιτέρως στον κο Ευάγγελο Καλαμπόκη ως επιβλέποντα της παρούσης εργασίας.

Αθανάσιος Οικονόμου

Περιεχόμενα

1. Εισαγωγή 1
1.1. Το πρόβλημα 1
1.2. Ο στόχος 2
1.3. Το περιεχόμενο της μελέτης 3
2. Γνωστικό υπόβαθρο 5
2.1. Μηχανική μάθηση 5
2.1.1. Αναδρομή 5
2.1.2. Ορισμός μηχανικής μάθησης 5
2.1.3. Στοιχεία μηχανικής μάθησης 5
2.1.4. Τύποι μηχανικής μάθησης 6
2.2. Χρήσιμα βοηθήματα 9
2.2.1. Η εφαρμογή Tableau 9
2.2.2. Η γλώσσα προγραμματισμού (πthon) Python 10
2.2.3. Το Jupyter Notebook 12
2.2.4. Η κοινότητα Kaggle 13
3. Βιβλιογραφική επισκόπηση του προβλήματος 15
4. Μεθοδολογία 19
5. Αναζήτηση δεδομένων του πεδίου προβλήματος 21
6. Τα σύνολα δεδομένων 23
6.1. Το χρηματοπιστωτικό ίδρυμα Home Credit 23
6.2. Περιγραφή του προβλήματος 23
6.3. Επεξήγηση των δεδομένων 24
6.3.1. Αρχεία application_test.csv και application_train.csv 25
6.3.2. Αρχείο bureau.csv 28
6.3.3. Αρχείο bureau_balance.csv 29
6.3.4. Αρχείο POS_CASH_balance.csv 29
6.3.5. Αρχείο credit_card_balance.csv 30
6.3.6. Αρχείο previous_application.csv 31
6.3.7. Αρχείο installments_payments.csv 32
7. Διερευνητική ανάλυση δεδομένων 35
7.1. Κατανομή ποσών των δανείων 35
7.2. Κατανομή εισοδήματος πελατών 36

7.3. Κόστος αγαθών για τα οποία δόθηκαν δάνεια 37
7.4. Συνοδοί πελατών κατά την αίτηση δανείου 37
7.5. Ισορροπία της μεταβλητής απόκρισης 38
7.6. Τύποι δανείων 38
7.7. Προέλευση εισοδήματος αιτούντων 39
7.8. Οικογενειακή κατάσταση αιτούντων 39
7.9. Επαγγελματική απασχόληση αιτούντων 40
7.10. Εκπαίδευση αιτούντων 40
7.11. Στεγαστική κατάσταση αιτούντων 41
7.12. Είδη επιχειρήσεων όπου εργάζονται οι αιτούντες 41
7.13. Προβληματικά δάνεια ανά κατηγορία 42
7.14. Τύποι προηγούμενων αιτήσεων δανείων 46
7.15. Κατάσταση προηγούμενων δανείων 47
7.16. Μέθοδοι πληρωμής προηγούμενων αιτήσεων δανείου 47
7.17. Αιτίες απόρριψης προηγούμενων αιτήσεων 48
7.18. Συνοδοί πελατών κατά τις προηγούμενες αιτήσεις 48
7.19. Ήδη πελάτες ή νέοι κατά τις προηγούμενες αιτήσεις; 49
7.20. Είδος αγαθών προς αγορά προηγούμενων αιτήσεων 49
7.21. Ασφάλεια δανείων στις προηγούμενες αιτήσεις 50
7.22. Τιμές «XNA», «XAP» και «NaN» 50
8. Προ-επεξεργασία δεδομένων 51
8.1. Δημιουργία νέων παραμέτρων 51
8.1.1. Πλήθος προηγούμενων δανείων 51
8.1.2. Πλήθος τύπων προηγούμενων δανείων 52
8.1.3. Λόγος ενεργών προς συνολικά προηγούμενα δάνεια 52
8.1.4. Μέσος όρος ημερών που λήγουν τα προηγούμενα δάνεια στο μέλλον 55
8.1.5. Λόγος χρέους προς πίστωση 58
8.1.6. Λόγος ληξιπρόθεσμων οφειλών προς πίστωση 61
8.1.7. Μέσος όρος ημερών καθυστέρησης πληρωμής δόσεων δανείων σε μετρητά 64
8.1.8. Λόγος προηγούμενων απορριπτέων αιτήσεων προς συνολικές προηγούμενες αιτήσεις 66
8.1.9. Λόγος ποσού αιτούμενου δανείου προς κόστος αγαθών 68
8.1.10. Λόγος ετήσιου ποσού καταβολής δανείου προς ετήσιο εισόδημα πελάτη .. 69
8.2. Συγχώνευση συνόλων δεδομένων 70

9. Μηχανική μάθηση 73
9.1. Το σύνολο δεδομένων 73
9.2. Διαχωρισμός δεδομένων 76
9.3. Δημιουργία του μοντέλου XGBoost 77
9.3.1. Συντονισμός υπερπαραμέτρων με την Αναζήτηση Πλέγματος (Grid Search). 77
9.3.2. Δημιουργία του μοντέλου με την Αναζήτηση Πλέγματος (Grid Search) 78
9.3.3. Πίνακας σύγχυσης (Confusion matrix) 80
9.3.4. Η καμπύλη ROC (Receiver operating characteristic) 81
9.3.5. Η καμπύλη AUC (Area under the ROC Curve) 82
9.3.6. Διάγραμμα σπουδαιότητας βάσει βαρύτητας (weight) 83
9.3.7. Διάγραμμα σπουδαιότητας βάσει κέρδους (gain) 84
10. Συμπεράσματα 85
Παράρτημα Α’ - Κώδικας 89
Βιβλιογραφία 95

Λίστα πινάκων

Πίνακας 1. Γενικές / αταξινόμητες μεταβλητές αίτησης δανείου 25
Πίνακας 2. Δημογραφικά στοιχεία πελάτη 26
Πίνακας 3. Οικογενειακή κατάσταση πελάτη 26
Πίνακας 4. Οικονομική και επαγγελματική κατάσταση πελάτη 26
Πίνακας 5. Περιουσιακή κατάσταση πελάτη 26
Πίνακας 6. Στοιχεία επικοινωνίας πελάτη / Υποβαλλόμενα έγγραφα 27
Πίνακας 7. Έλεγχος παρεχόμενων στοιχείων πελάτη 27
Πίνακας 8. Στοιχεία αιτούντος δανείου 28
Πίνακας 9. Στοιχεία προηγούμενων δανείων πελάτη 28
Πίνακας 10. Ιστορικό δόσεων προηγούμενων δανείων 29
Πίνακας 11. Μηνιαίο ιστορικό προηγούμενων καταναλωτικών δανείων στη Home Credit 29
Πίνακας 12. Ποσά κινήσεων δανείων πιστωτικών καρτών ανά μήνα 30
Πίνακας 13. Σύνολα κινήσεων δανείων πιστωτικών καρτών ανά μήνα 31
Πίνακας 14. Ημέρες καθυστέρησης εξόφλησης δόσεων δανείων πιστωτικών καρτών ανά μήνα 31
Πίνακας 15. Γενικές μεταβλητές προηγούμενης αίτησης για δάνειο 31
Πίνακας 16. Στοιχεία δανείων προηγούμενων αιτήσεων 32
Πίνακας 17. Ποσά δόσεων δανείων 33

Λίστα εικόνων

Εικόνα 1. Ποσοστά μη εξυπηρετούμενων δανείων στην Ελλάδα 2
Εικόνα 2. Τρόπος λειτουργίας αλγόριθμου εποπτευόμενης μηχανικής μάθησης 6
Εικόνα 3. Εξαγωγή χαρακτηριστικών και εύρεση μοτίβων στα δεδομένα 7
Εικόνα 4. Πεδίο εφαρμογής ημι-εποπτευόμενων μοντέλων μηχανικής μάθησης 8
Εικόνα 5. Κορυφαίες εταιρίες διάθεσης εφαρμογών ανάλυσης και επιχειρηματικής ευφυίας 9
Εικόνα 6. Γράφημα με σημάδια πυκνότητας (heatmap) στο Tableau 10
Εικόνα 7. Το Zen της Python 11
Εικόνα 8. Οι πιο δημοφιλείς γλώσσες προγραμματισμού στο Stack Overflow 12
Εικόνα 9. Σημειώσεις, κώδικας και εισαγωγή δεδομένων στο Jupyter Notebook 13
Εικόνα 10. Εγγεγραμμένα μέλη Kaggle 13
Εικόνα 11. Χορηγούμενα δάνεια Home Credit 23
Εικόνα 12. Η σχέση ανάμεσα στα αρχεία δεδομένων 24
Εικόνα 13. Κατανομή ποσών των δανείων 35
Εικόνα 14. Ομαδοποιημένη κατανομή ποσών των δανείων 36
Εικόνα 15. Κατανομή εισοδήματος πελατών 36
Εικόνα 16. Κόστος αγαθών για τα οποία δόθηκαν δάνεια 37
Εικόνα 17. Συνοδοί πελατών κατά την αίτηση δανείου 37
Εικόνα 18. Ισορροπία της μεταβλητής απόκρισης 38
Εικόνα 19. Τύποι δανείων 38
Εικόνα 20. Προέλευση εισοδήματος αιτούντων 39
Εικόνα 21. Οικογενειακή κατάσταση αιτούντων 39
Εικόνα 22. Επαγγελματική απασχόληση αιτούντων 40
Εικόνα 23. Εκπαίδευση αιτούντων 40
Εικόνα 24. Στεγαστική κατάσταση αιτούντων 41
Εικόνα 25. Είδη επιχειρήσεων όπου εργάζονται οι αιτούντες 41
Εικόνα 26. Ποσοστά προβληματικών δανείων ανά προέλευση εισοδήματος αιτούντων 42
Εικόνα 27. Ποσοστά προβληματικών δανείων ανά οικογενειακή κατάσταση αιτούντων 43
Εικόνα 28. Ποσοστά προβληματικών δανείων ανά επαγγελματική απασχόληση αιτούντων 43
Εικόνα 29. Ποσοστά προβληματικών δανείων ανά τύπο εκπαίδευσης αιτούντων 44
Εικόνα 30. Ποσοστά προβληματικών δανείων ανά στεγαστική κατάσταση αιτούντων .. 45
Εικόνα 31. Ποσοστά προβληματικών δανείων ανά είδος επιχείρησης όπου εργάζονται οι αιτούντες 45
Εικόνα 32. Ποσοστά προβληματικών δανείων ανά τύπο συνοδείας που είχαν κατά την αίτηση για δάνειο 46
Εικόνα 33. Τύποι προηγούμενων αιτήσεων 46
Εικόνα 34. Ποσοστά κατάστασης προηγούμενων αιτήσεων δανείου 47
Εικόνα 35. Επιθυμητός τρόπος πληρωμής δόσης προηγούμενων δανείων 47
Εικόνα 36. Ποσοστά λόγων απόρριψης προηγούμενων αιτήσεων 48
Εικόνα 37. Ποσοστά συνοδών προηγούμενων αιτούντων 48
Εικόνα 38. Ποσοστά νέων ή υφιστάμενων πελατών προηγούμενων αιτήσεων 49

Εικόνα 39. Για τι αγαθά έγιναν οι προηγούμενες αιτήσεις δανείου 49
Εικόνα 40. Ποσοστά ασφάλισης δανείων στις προηγούμενες αιτήσεις 50
Εικόνα 41. Πλήθος προηγούμενων δανείων ανά πελάτη 51
Εικόνα 42. Πλήθος τύπων προηγούμενων δανείων ανά πελάτη 52
Εικόνα 43. Κατάσταση προηγούμενων δανείων πελατών 53
Εικόνα 44. Προηγούμενα ενεργά δάνεια πελατών 53
Εικόνα 45. Πλήθος προηγούμενων ενεργών δανείων ανά πελάτη 54
Εικόνα 46. Σύνολο δεδομένων πλήθους προηγούμενων δανείων ανά πελάτη 54
Εικόνα 47. Λόγος ενεργών προς συνολικά προηγούμενα δάνεια 55
Εικόνα 48. Πλήθος ημερών που λήγουν τα προηγούμενα δάνεια 55
Εικόνα 49. Δημιουργία παραμέτρου για τα προηγούμενα δάνεια 56
Εικόνα 50. Παράμετρος λογικής μορφής όπου δηλώνει παρελθόν (0) ή μέλλον (1) 57
Εικόνα 51. Προηγούμενα δάνεια που είναι ακόμη σε ισχύ ή άγνωστης κατάστασης 57
Εικόνα 52. Προηγούμενα δάνεια που είναι ακόμη σε ισχύ 58
Εικόνα 53. Μέσος όρος ημερών που λήγουν τα προηγούμενα δάνεια του κάθε πελάτη στο μέλλον 58
Εικόνα 54. Ποσά πίστωσης και χρέους προηγούμενων δανείων 59
Εικόνα 55. Ποσά πίστωσης και χρέους προηγούμενων δανείων με μηδενισμό NaN τιμών 59
Εικόνα 56. Συνολικό ποσό πίστωσης κάθε πελάτη 60
Εικόνα 57. Συνολικό ποσό χρέους κάθε πελάτη 60
Εικόνα 58. Σύνολο δεδομένων ποσών πίστωσης και χρεών προηγούμενων δανείων 61
Εικόνα 59. Λόγος χρέους προς πίστωση ανά πελάτη για προηγούμενα δάνεια 61
Εικόνα 60. Ποσά συνολικού χρέους και ληξιπρόθεσμων οφειλών προηγούμενων δανείων 62
Εικόνα 61. Ποσά συνολικού χρέους και ληξιπρόθεσμων οφειλών προηγούμενων δανείων με των τιμών NaN 62
Εικόνα 62. Συνολικό ληξιπρόθεσμο ποσό προηγούμενων δανείων ανά πελάτη 63
Εικόνα 63. Σύνολο δεδομένων ληξιπρόθεσμων οφειλών πελατών για προηγούμενα δάνεια 64
Εικόνα 64. Λόγος ληξιπρόθεσμων οφειλών προς πίστωση για προηγούμενα δάνεια 64
Εικόνα 65. Ημέρες καθυστέρησης πληρωμής δόσεων για τον πελάτη με ID 100008 65
Εικόνα 66. Μέσος όρος ημερών καθυστέρησης πληρωμής δόσεων δανείων σε μετρητά 65
Εικόνα 67. Προηγούμενες αιτήσεις και προηγούμενες απορριφθείσες αιτήσεις στη Home Credit 67
Εικόνα 68. Λόγος απορριπτέων προηγούμενων αιτήσεων δανείων προς συνολικές προηγούμενες αιτήσεις στη Home Credit 68
Εικόνα 69. Λόγος ποσού αιτούμενου δανείου προς κόστος αγαθών 69
Εικόνα 70. Λόγος ετήσιου ποσού καταβολής δανείου προς ετήσιο εισόδημα πελάτη 70
Εικόνα 71. Το τελικό σύνολο δεδομένων (εμφάνιση νέων παραμέτρων) 71
Εικόνα 72. Ζητούμενη μορφή αρχείου υποβολής 73
Εικόνα 73. Το σχήμα των δεδομένων 74
Εικόνα 74. Μετατροπή στηλών τύπου «object» σε αριθμητικά δεδομένα 74
Εικόνα 75. Κατανομή των δεδομένων 75

Εικόνα 76. Μετατροπή τιμών inf και -inf σε τιμές NaN 75
Εικόνα 77. Κατανομή της εξαρτημένης μεταβλητής (TARGET) 76
Εικόνα 78. Τα ακριβή ποσοστά των δανείων με δυσκολία πληρωμής ή όχι 76
Εικόνα 79. το σύνολο δεδομένων με τις μεταβλητές πρόβλεψης μετά το διαχωρισμό ... 77
Εικόνα 80. το σύνολο δεδομένων «X» με τη μεταβλητή στόχου μετά το διαχωρισμό 77
Εικόνα 81. Λήψη στιγμιότυπου με τον XGBClassifier και επιθεώρηση των παραμέτρων 78
Εικόνα 82. Έκθεση ταξινόμησης (Classification report) 80
Εικόνα 83. Πίνακας σύγχυσης (Confusion matrix)..... 81
Εικόνα 84. Η καμπύλη ROC (Receiver operating characteristic) 82
Εικόνα 85. Η καμπύλη AUC (Area under the ROC Curve) 83
Εικόνα 86. Σπουδαιότητα χαρακτηριστικών βάσει βαρύτητας (weight) 84
Εικόνα 87. Σπουδαιότητα χαρακτηριστικών βάσει κέρδους (gain) 84

1. Εισαγωγή

Πριν την εμφάνιση των πρώτων νομισμάτων ανά τον κόσμο, οι άνθρωποι αντάλλασσαν αντικείμενα. Ο αντιπραγματισμός ήταν ο τρόπος με τον οποίο γίνονταν οι συναλλαγές ανάμεσα στους ανθρώπους. Αν ήθελε κάποιος να αποκτήσει αχλάδια για παράδειγμα, έπρεπε να δώσει μήλα ή ό,τι άλλο είχε στην κατοχή του.

Αυτό βέβαια δεν παρείχε σχεδόν καθόλου ευελιξία. Μπορεί να υπήρχε ανάγκη να αποκτήσει κάποιος ένα αντικείμενο ή αγαθό αλλά αυτό που ο ίδιος προσέφερε ως αντάλλαγμα να μην ήταν κάτι που εξυπηρετούσε τις ανάγκες του ετέρου συναλλασσόμενου. Έτσι, η συναλλαγή ενδεχομένως να μην πραγματοποιούνταν.

Άλλο πρόβλημα που υπήρχε με τον αντιπραγματισμό, ήταν ότι στις περισσότερες περιπτώσεις, τα ανταλλάξιμα αντικείμενα ή αγαθά καταλάμβαναν και αρκετό όγκο αλλά και το βάρος τους ήταν μεγάλο. Σήμερα, μπορεί να κυκλοφορούμε στις πόλεις έχοντας σε ένα πορτοφόλι μερικές δεκάδες ευρώ αλλά τότε ήταν δύσκολο να κυκλοφορεί ο κόσμος κουβαλώντας μερικά τεμάχια μήλα.

Συνεπώς, έπρεπε να βρεθούν εναλλακτικοί τρόποι συναλλαγών και ένας από αυτούς ήταν η χρήση του μετάλλου ως αυτό που γνωρίζουμε όλοι σήμερα ως χρήμα. Το μέταλλο έδωσε αρκετές λύσεις στα προβλήματα του αντιπραγματισμού. Είχε σχετικά σταθερή ανταλλακτική αξία, ήταν ανθεκτικό σε φθορές και ήταν ιδανικό για μεταφορά και αποθήκευση.

Το μέταλλο όμως είχε και μειονεκτήματα. Κάθε φορά που λάμβανε μέρος σε κάποια συναλλαγή έπρεπε να ζυγιστεί και επίσης, ήταν δύσκολη η διαπίστωση της καθαρότητάς του ως μέταλλο.

Εν τέλει, τα νομίσματα ήταν αυτά που επικράτησαν εδώ και χρόνια, με την πρώτη τους εμφάνιση κάποιους αιώνες πριν από τη γέννηση του Χριστού. Τα χαρακτηριστικά τους είναι το συγκεκριμένο σχήμα, μέγεθος και βάρος τους. Μπορεί να είναι είτε κέρματα είτε χαρτονομίσματα.

Το 1185 μ.Χ., οι Ναΐτες Ιππότες απέκτησαν τη βρετανική τους έδρα στο εκκλησάκι Temple Church στο Λονδίνο. Το εκκλησάκι όμως δεν ήταν απλά μόνο μία έδρα. Φιλοξενούσε επίσης χρηματοπιστωτικές υπηρεσίες όπως μεταφορά χρημάτων και δανεισμό. Οι Ναΐτες Ιππότες εφηύραν πρώτοι τον τόκο.

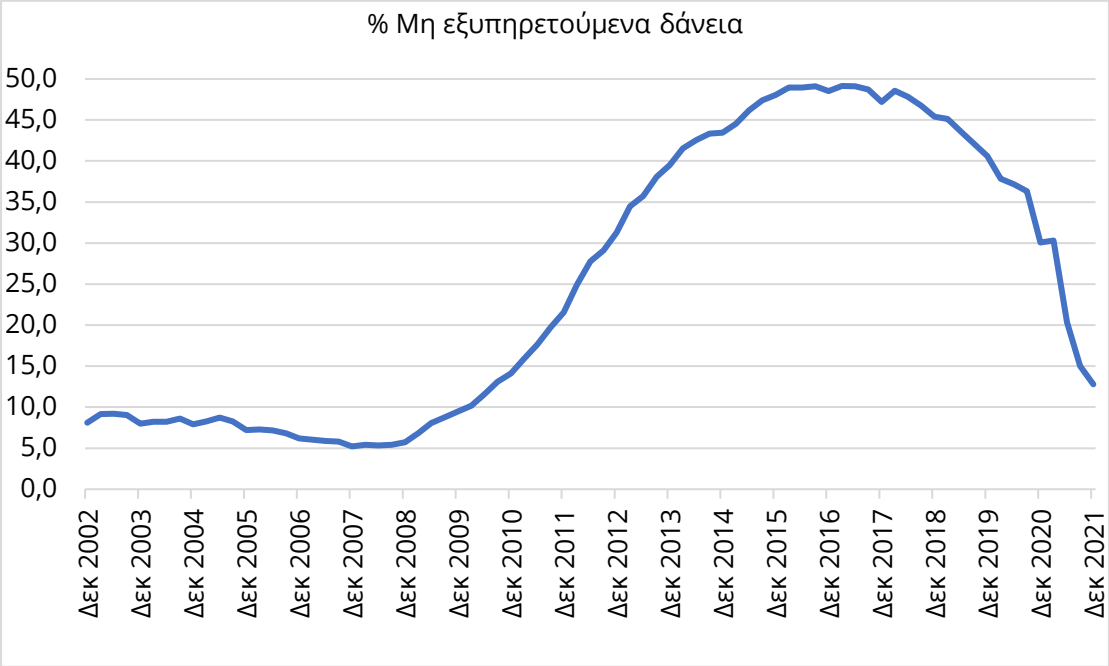
Οι τράπεζες, όπως τις γνωρίζουμε σήμερα, έκαναν την εμφάνισή τους τον 14ο αιώνα μ.Χ.. Το 1472 ιδρύθηκε η Banca Monte dei Paschi di Siena στην πόλη Σιένα της περιφέρειας Τοσκάνης της κεντρικής Ιταλίας. Στην Ελλάδα, η πρώτη τράπεζα ιδρύθηκε το 1839. Το πρώτο κατάστημα της Ιονικής Τράπεζας και συνάμα το κεντρικό γραφείο της, λειτούργησε για πρώτη φορά στις 2 Μαρτίου 1840 στην Κέρκυρα.

1.1. Το πρόβλημα

Μία από τις βασικές υπηρεσίες των τραπεζών, είναι η χορήγηση δανείων. Η τράπεζα κερδίζει χρήματα από τον τόκο των δανείων που εκδίδει. Η σταθερότητα στην

οικονομική κατάσταση του πελάτη είναι σημαντική ώστε να μπορέσει να καταβάλλει τις προβλεπόμενες δόσεις και τον τόκο. Αν το δάνειο αποπληρώνεται όπως έχει προγραμματιστεί, το δάνειο χαρακτηρίζεται ως εξυπηρετούμενο.

Υπάρχουν περιπτώσεις όπου κάποιος δανειολήπτης να μην είναι σε θέση να αποπληρώσει το δάνειο στο χρονοδιάγραμμα που έχει προ-συμφωνηθεί. Αυτό μπορεί να οφείλεται σε οικονομικές δυσκολίες, όπως για παράδειγμα απώλεια θέσεως εργασίας αν αφορά εργαζόμενο, πράγμα που κλονίζει την οικονομική του κατάσταση. Αν συμβεί κάτι τέτοιο ή φαίνεται πιθανόν να συμβεί, η τράπεζα χαρακτηρίζει το δάνειο ως «μη εξυπηρετούμενο».



Εικόνα 1. Ποσοστά μη εξυπηρετούμενων δανείων στην Ελλάδα

Σύμφωνα με στοιχεία από την Τράπεζα της Ελλάδος, στην Ελλάδα μετά από την οικονομική κρίση το 2009, το ποσοστό των μη εξυπηρετούμενων δανείων αυξήθηκε δραματικά. Τείνει όμως να επιστρέψει στα προ κρίσης επίπεδα.

1.2. Ο στόχος

Ένα χρηματοπιστωτικό ίδρυμα, όταν λαμβάνει μία νέα αίτηση για δάνειο, σαφώς είτε θα την εγκρίνει είτε θα την απορρίψει. Από την άλλη, αν εκδοθεί εν τέλει το δάνειο, ο δανειολήπτης είτε θα το αποπληρώσει είτε όχι. Βάσει λογικής, προκύπτουν τρεις περιπτώσεις σχετικά με τα δάνεια:

1. Το δάνειο θα εκδοθεί και θα αποπληρωθεί.
2. Το δάνειο θα εκδοθεί αλλά δε θα αποπληρωθεί.
3. Το δάνειο δε θα εκδοθεί.

Το εμφανές πρόβλημα βρίσκεται στην περίπτωση 2, όπου ένα δάνειο εκδίδεται αλλά δεν αποπληρώνεται. Πρόβλημα όμως για το χρηματοπιστωτικό ίδρυμα είναι και η

περίπτωση που δεν εκδόσει κάποιο δάνειο ενώ αυτό θα αποπληρωνόταν, χάνοντας έτσι το ίδρυμα έσοδα.

Συνεπώς, οι τρεις προηγούμενες περιπτώσεις γίνονται τέσσερις, αυξάνοντας τον προβληματισμό για το αν ένα δάνειο εν τέλει πρέπει ή όχι να εκδοθεί:

1. Το δάνειο θα εκδοθεί και θα αποπληρωθεί.
2. Το δάνειο θα εκδοθεί αλλά δε θα αποπληρωθεί.
3. Το δάνειο δε θα εκδοθεί αλλά αν εκδιδόταν θα αποπληρωνόταν.
4. Το δάνειο δε θα εκδοθεί και αν εκδιδόταν δε θα αποπληρωνόταν.

Στόχος της συγκεκριμένης διπλωματικής εργασίας είναι η κατασκευή και βελτιστοποίηση ενός μοντέλου μηχανικής μάθησης. Το μοντέλο θα αξιοποιεί σύνολα δεδομένων χρηματοπιστωτικών ιδρυμάτων. Τα δεδομένα θα επεξεργαστούν και θα προκύψουν νέες παράμετροι. Το μοντέλο μηχανικής μάθησης εν τέλει, θα προβλέπει για κάθε νέα αίτηση δανείου αν το δάνειο θα αποπληρωθεί ή όχι, ώστε να ληφθεί η απόφαση αν θα πρέπει να εκδοθεί ή όχι αντίστοιχα.

1.3. Το περιεχόμενο της μελέτης

Στο κεφάλαιο 2 γίνεται αναφορά στο γνωστικό υπόβαθρο που απαιτείται για τη μελέτη. Γίνεται εισαγωγή στη μηχανική μάθηση, τα στοιχεία που περιλαμβάνει και τους τύπους της. Αναφέρονται τα χρήσιμα εργαλεία που θα βοηθήσουν στην παρούσα μελέτη.

Στο κεφάλαιο 3 γίνεται μία βιβλιογραφική επισκόπηση με σκοπό την πληροφόρηση για τη χρονική διάρκεια που απασχολεί το πεδίο του προβλήματος την ερευνητική κοινότητα καθώς και τον τρόπο αντιμετώπισής του.

Στο κεφάλαιο 4 παρουσιάζεται η μεθοδολογία που θα ακολουθηθεί, κάθε στάδιο της οποίας αποτελεί και ένα επιπλέον κεφάλαιο.

Έτσι λοιπόν, στο κεφάλαιο 5 γίνεται η αναζήτηση δεδομένων για το πεδίο του προβλήματος.

Στο κεφάλαιο 6 παρουσιάζεται κάποιο παρόμοιο πρόβλημα ενώ στη συνέχεια γίνεται η επεξήγηση των δεδομένων που λήφθηκαν ώστε να χρησιμοποιηθούν στη μελέτη.

Το κεφάλαιο 7 περιλαμβάνει την διερευνητική ανάλυση των δεδομένων μέσα από την κατανόησή τους, την εξερεύνησή τους και την οπτικοποίησή τους με τη χρήση της εφαρμογής Tableau.

Στο κεφάλαιο 8 γίνεται προ-επεξεργασία στα δεδομένα, παράλληλα με τη δημιουργία νέων παραμέτρων. Ακολουθεί η συγχώνευση των συνόλων δεδομένων με σκοπό την τροφοδότηση του αλγορίθμου μηχανικής μάθησης.

Στο κεφάλαιο 9 υλοποιείται η μηχανική μάθηση. Αρχικά, γίνεται ο συντονισμός υπερπαραμέτρων με την Αναζήτηση Πλέγματος (Grid Search) και η δημιουργία του μοντέλου XGBoost μετά από αυτήν. Ακολουθεί ο πίνακας σύγχυσης (Confusion matrix) και οι οπτικοποιήσεις των καμπυλών ROC και AUC. Τέλος, δημιουργούνται τα

διαγράμματα σπουδαιότητας χαρακτηριστικών με βάση τη βαρύτητα (weight) και το κέρδος (gain).

Στο κεφάλαιο 10 αναφέρονται τα συμπεράσματα.

2. Γνωστικό υπόβαθρο

2.1. Μηχανική μάθηση

Η μηχανική μάθηση είναι ένα σημαντικό εργαλείο για τον στόχο της μόχλευσης των τεχνολογιών γύρω από την τεχνητή νοημοσύνη. Λόγω των ικανοτήτων μάθησης και λήψης αποφάσεων, η μηχανική μάθηση αναφέρεται συχνά ως τεχνητή νοημοσύνη, αν και, στην πραγματικότητα, είναι ένας κλάδος της τεχνητής νοημοσύνης. Μέχρι τα τέλη της δεκαετίας του 1970, η μηχανική μάθηση ήταν μέρος της εξέλιξης της τεχνητής νοημοσύνης. Στη συνέχεια, άρχισε να εξελίσσεται από μόνη της.

2.1.1. Αναδρομή

Η μηχανική μάθηση βασίζεται εν μέρει σε ένα μοντέλο αλληλεπίδρασης των εγκεφαλικών κυττάρων που δημιουργήθηκε από τον Donald Hebb το 1949 σε ένα βιβλίο με τίτλο «The Organization of Behavior». Στο βιβλίο παρουσιάζονται οι θεωρίες του Hebb σχετικά με τη διέγερση των νευρώνων και την επικοινωνία μεταξύ των νευρώνων.

Ο Arthur Samuel, τη δεκαετία του 1950 ανέπτυξε ένα πρόγραμμα υπολογιστή για να παίζει ντάμα. Το πρόγραμμά του βαθμολογούσε τις θέσεις των κομματιών στη σκακιέρα, εκτιμώντας ταυτόχρονα πόσο πιθανό είναι η κάθε πλευρά να κερδίσει το παιχνίδι.

Ο Άρθουρ Σάμιουελ ανακάλυψε για πρώτη φορά τη φράση «μηχανική μάθηση» το 1952.

2.1.2. Ορισμός μηχανικής μάθησης

Για τη μηχανική μάθηση έχουν δοθεί κατά καιρούς πολλοί ορισμοί, άλλοι περισσότερο κατανοητοί και άλλοι πιο ασαφείς. Όμως τί είναι, με απλούστερα λόγια, η μηχανική μάθηση;

Σύμφωνα με την IBM, η μηχανική μάθηση εστιάζει στη χρήση δεδομένων και αλγορίθμων για τη μίμηση του τρόπου με τον οποίο μαθαίνουν οι άνθρωποι, βελτιώνοντας σταδιακά την ακρίβειά της. (What is Machine Learning?, 2020)

Η βασική ιδέα της μηχανικής μάθησης στην επιστήμη δεδομένων περιλαμβάνει τη χρήση στατιστικής μάθησης και μεθόδων βελτιστοποίησης που επιτρέπουν στους υπολογιστές να αναλύουν σύνολα δεδομένων και να αναγνωρίζουν μοτίβα στα δεδομένα αυτά.

2.1.3. Στοιχεία μηχανικής μάθησης

Ένας τυπικός αλγόριθμος μηχανικής μάθησης αποτελείται από τρία (3) βασικά στοιχεία:

1. **Μια διαδικασία λήψης αποφάσεων**: Είναι μία διαδικασία που λαμβάνει τα δεδομένα και «μαντεύει» το είδος του μοτίβου που ψάχνει να βρει ο αλγόριθμος.
2. **Μια συνάρτηση σφάλματος**: Αφορά μία μέθοδο μέτρησης για το πόσο καλά μάντεψε ο αλγόριθμος. Αυτή η μέτρηση σαφώς γίνεται σε σύγκριση με γνωστά παραδείγματα εφόσον είναι διαθέσιμα. Εδώ τίθεται το ερώτημα αν λειτούργησε σωστά η διαδικασία λήψης αποφάσεων και αν όχι, πόσο άσχημα λειτούργησε.
3. **Διαδικασία ενημέρωσης ή βελτιστοποίησης**: Αυτή είναι μία μέθοδος κατά την οποία ο αλγόριθμος εξετάζει τα λάθη του, ενημερώνοντας τον τρόπο λειτουργίας της διαδικασίας λήψης αποφάσεων ώστε την επόμενη φορά τα λάθη να είναι λιγότερα.

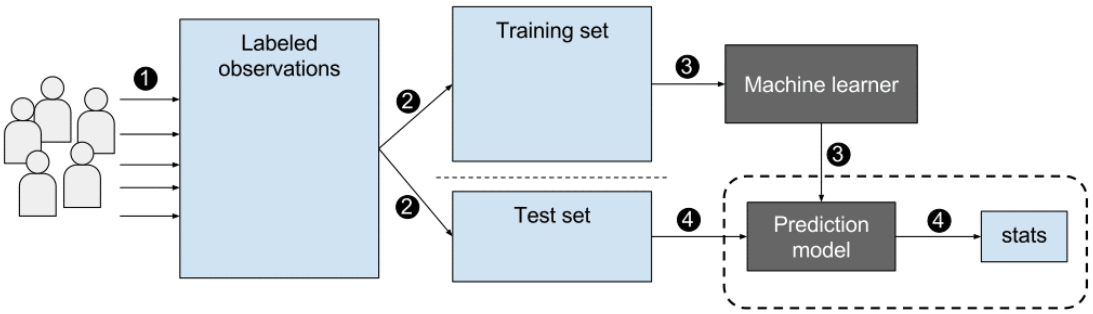
2.1.4. Τύποι μηχανικής μάθησης

Σύμφωνα με την εταιρία NVIDIA, ανάλογα με τα είδη των συνόλων δεδομένων και των προβλημάτων για τα οποία αναζητείται λύση, υπάρχουν τέσσερις (4) τύποι μηχανικής μάθησης:

1. η εποπτευόμενη μηχανική μάθηση,
2. η μη εποπτευόμενη μηχανική μάθηση,
3. η ημι-εποπτευόμενη μηχανική μάθηση,
4. η ενισχυτική μηχανική μάθηση.

2.1.4.1. Εποπτευόμενη μηχανική μάθηση

Όπως γίνεται στον πραγματικό κόσμο, σε μία εργασία ή εξέταση υπό επίβλεψη ή υπό εποπτεία, υπάρχει κάποιος που κρίνει αν η απάντηση είναι σωστή ή λανθασμένη. Ομοίως και στην εποπτευόμενη μηχανική μάθηση, υπάρχει ένα σύνολο δεδομένων όπου όλα τα δεδομένα έχουν προ-επισημανθεί για την εκπαίδευση ενός αλγορίθμου. Έτσι, κάθε παράδειγμα στο σύνολο δεδομένων προς εκπαίδευση του αλγορίθμου επισημαίνεται με την απάντηση που θα κληθεί να βρει ο αλγόριθμος από μόνος του.



Εικόνα 2. Τρόπος λειτουργίας αλγόριθμου εποπτευόμενης μηχανικής μάθησης

Για παράδειγμα, ένα επισημασμένο σύνολο δεδομένων εικόνων λουλουδιών θα έλεγε στο μοντέλο ποιες φωτογραφίες ήταν από τριαντάφυλλα και ποιες από μαργαρίτες. Κατά την εμφάνιση μίας νέας φωτογραφίας, το μοντέλο τη συγκρίνει με τα παραδείγματα

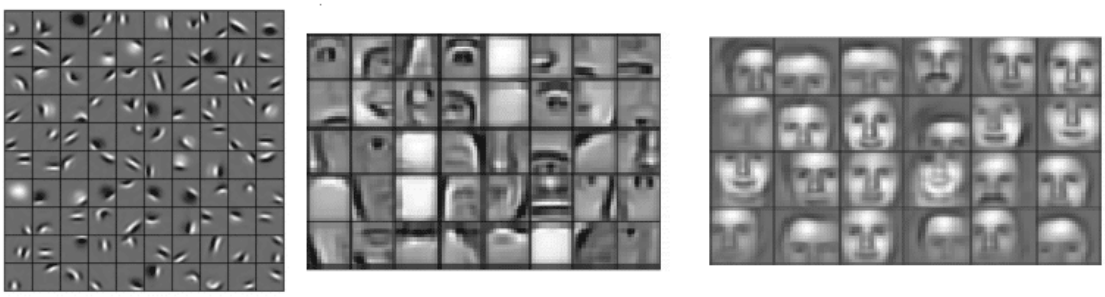
στα οποία έχει εκπαιδευτεί ώστε να προβλέψει τη σωστή επισήμανση, δηλαδή αν η φωτογραφία απεικονίζει ένα τριαντάφυλλο ή μία μαργαρίτα.

Με την εποπτευόμενη μηχανική μάθηση λοιπόν, ο αλγόριθμος μαθαίνει από δεδομένα στα οποία έχει δοθεί εκ των προτέρων μία ονομασία, μία τιμή πρόβλεψης.

2.1.4.2. Μη εποπτευόμενη μηχανική μάθηση

Τα επισημασμένα σύνολα δεδομένων στις περισσότερες περιπτώσεις δεν υπάρχουν. Πολλές φορές, οι αλγόριθμοι τίθενται να απαντήσουν σε ερωτήματα που οι ερευνητές δε γνωρίζουν την απάντηση. Εφόσον τα δεδομένα δεν είναι εφικτό να προ-επισημανθούν, η μηχανική μάθηση γίνεται μη εποπτευόμενα.

Στη μη εποπτευόμενη μηχανική μάθηση, το μοντέλο λαμβάνει ένα σύνολο δεδομένων χωρίς να γνωρίζει τί πρέπει να κάνει με αυτό και χωρίς να υπάρχει κάποια συγκεκριμένη σωστή απάντηση. Έτσι, το μοντέλο επιχειρεί να βρει κάποια δομή ή μοτίβα στα σύνολα δεδομένων, εξάγοντας χρήσιμα χαρακτηριστικά.



Εικόνα 3. Εξαγωγή χαρακτηριστικών και εύρεση μοτίβων στα δεδομένα

Ανάλογα με το πρόβλημα που τίθεται, το μοντέλο μη εποπτευόμενης μάθησης μπορεί να οργανώσει τα δεδομένα με διαφορετικούς τρόπους, κάνοντας:

- ομαδοποίηση (αναζήτηση δεδομένων που είναι παρόμοια μεταξύ τους),
- ανίχνευση ανωμαλιών (επισήμανση ακραίων στοιχείων σε ένα σύνολο δεδομένων),
- συσχέτιση (πρόβλεψη συσχέτισης χαρακτηριστικών),
- κ.ά.

2.1.4.3. Ημι-εποπτευόμενη μηχανική μάθηση

Η ημι-εποπτευόμενη μηχανική μάθηση, είναι συνδυασμός των δύο προηγουμένως αναφερθέντων τύπων. Στην περίπτωση αυτή, το σύνολο δεδομένων περιέχει δεδομένα τόσο προ-επισημασμένα όσο και μη. Εφαρμογή έχει σε περιπτώσεις όπου η προ-επισήμανση των δεδομένων είναι μία χρονοβόρα διαδικασία.

Συνηθισμένο πεδίο εφαρμογής για αυτόν τον τύπο μηχανικής μάθησης είναι η ιατρική, όπου γίνεται χρήση εικόνων όπως η αξονική ή η μαγνητική τομογραφία.

Ένας ειδικά εκπαιδευμένος ακτινολόγος είναι ικανός να εξετάσει και να επισημάνει όγκους σε μία αξονική ή μαγνητική τομογραφία. Είναι όμως πολύ χρονοβόρο

να επαναλάβει το ίδιο για ένα μεγάλο αριθμό εξετάσεων ώστε να επισημάνει όλες τις σαρώσεις.

Εικόνα 4. Πεδίο εφαρμογής ημι-εποπτευόμενων μοντέλων μηχανικής μάθησης

Για ένα μοντέλο μηχανικής μάθησης, έστω και ένα μικρό δείγμα προ-επισημασμένων δεδομένων, είναι αρκετό ώστε να μπορέσει να βελτιώσει την ακρίβειά του σε σύγκριση με ένα μη εποπτευόμενο μοντέλο.

2.1.4.4. Ενισχυτική μηχανική μάθηση

Στα παιχνίδια, οι παίκτες προσπαθούν να βελτιωθούν μαθαίνοντας από τις σωστές και λανθασμένες του κινήσεις. Όπως συμβαίνει στα ηλεκτρονικά παιχνίδια, όταν ένα επίπεδο ολοκληρωθεί, ο παίκτης κερδίζει πόντους μεταβαίνοντας στο επόμενο επίπεδο. Αν κάνει μία λάθος κίνηση, ο παίκτης ξεκινάει το επίπεδο από την αρχή φροντίζοντας να μην επαναλάβει το ίδιο λάθος.

Το ίδιο ισχύει και για την ενισχυτική μηχανική μάθηση. Χρησιμοποιεί ένα σύστημα επιβράβευσης ή ποινής, προσφέροντας ανατροφοδότηση στον αλγόριθμο ώστε να αποκτήσει την εμπειρία για να ολοκληρώσει το στόχο του.

Αυτή είναι μία επαναληπτική διαδικασία: όσο περισσότεροι είναι οι γύροι ανατροφοδότησης – προσπαθειών, τόσο περισσότερο εκπαιδεύεται ο αλγόριθμος.

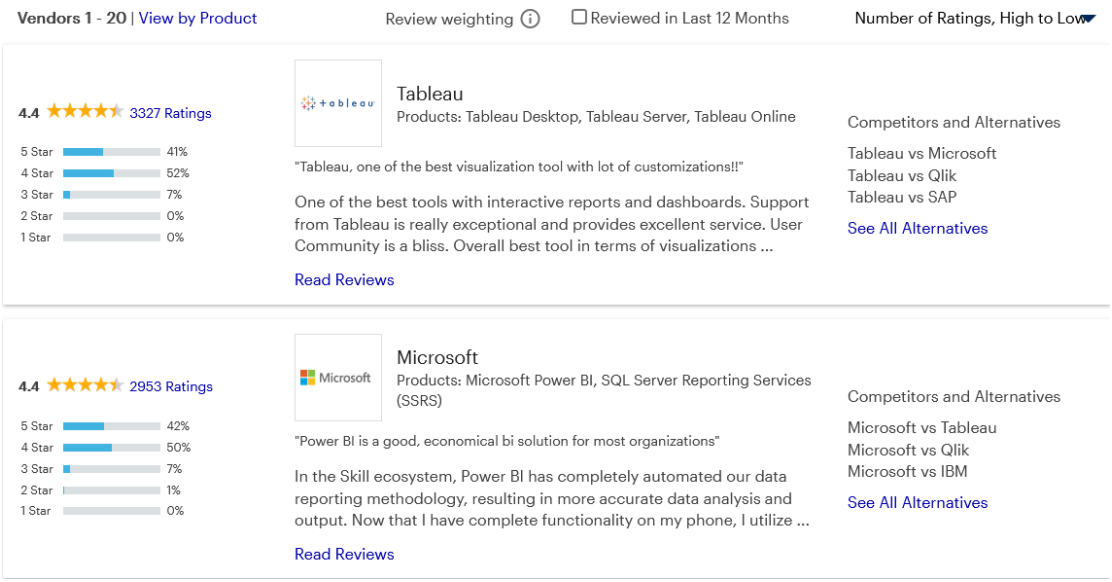
Μία οπτική εισαγωγή στη μηχανική μάθηση με κινούμενα γραφικά που έχει ενδιαφέρον να δει κανείς, υπάρχει στον διαδικτυακό τόπο r2d3.us. Όπως αναφέρεται στο συγκεκριμένο ιστότοπο «*Το R2D3 είναι ένα πείραμα έκφρασης στατιστικής σκέψης με διαδραστικό σχεδιασμό*». (R2D3: Statistics and Data Visualization, 2015)

2.2. Χρήσιμα βοηθήματα

2.2.1. Η εφαρμογή Tableau

Το Tableau είναι μία εφαρμογή η οποία μπορεί να εκτελεστεί είτε τοπικά σε ηλεκτρονικό υπολογιστή είτε διαδικτυακά χρησιμοποιώντας υπολογιστική ισχύ του νέφους. Η ύπαρξή του οφείλεται στη διαρκή παραγωγή δεδομένων, πόσο μάλλον μεγάλων δεδομένων.

Ανήκει στην κατηγορία των εργαλείων επιχειρηματικής ευφυίας και συναγωνίζεται εφαρμογές όπως οι Microsoft Power BI, SAP BusinessObjects BI Suite, IBM Cognos Analytics κ.ά..



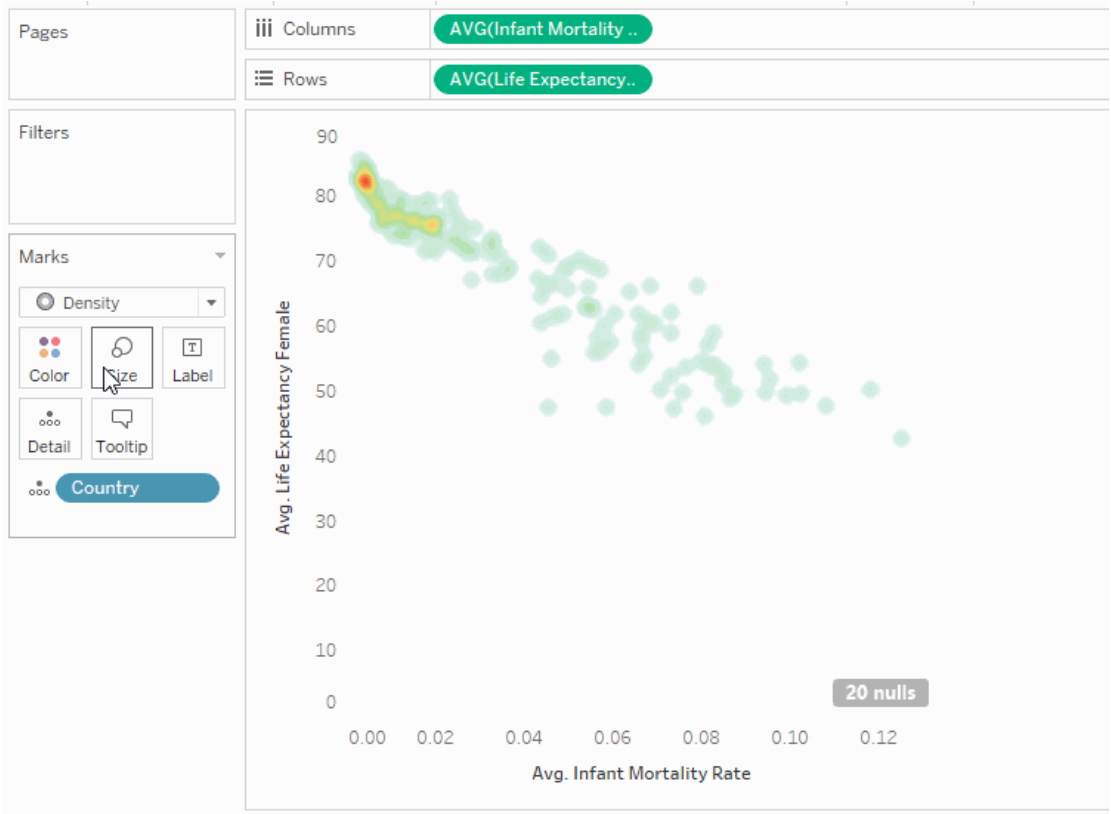
Εικόνα 5. Κορυφαίες εταιρίες διάθεσης εφαρμογών ανάλυσης και επιχειρηματικής ευφυίας

Το Tableau, εξυπηρετεί στη διαχείριση και την εξερεύνηση των δεδομένων καθώς και στην ανακάλυψη σημαντικών πληροφοριών που μπορούν να βελτιώσουν τις επιχειρήσεις. Αυτό είναι εφικτό μέσα από ένα περιβάλλον μεταφοράς και απόθεσης δεδομένων σε ερωτήματα δεδομένων.

Αποτέλεσμα της χρήσης του Tableau, είναι η οπτικοποίηση των δεδομένων με διαισθητικά, αναλυτικά στοιχεία. Προσφέρονται αρκετού είδους γραφήματα, όπως:

- περιοχής,
- ράβδων,
- κουκκίδων,
- με σημάδια πυκνότητας (heatmap),
- ιστόγραμμα,
- γραμμικό,
- πίτας
- κ.ά.

Μέσα από την εξερεύνηση, την ανάλυση και την ομαδοποίηση δεδομένων, αυτά τα δεδομένα όχι μόνο μπορούν να οπτικοποιηθούν, αλλά στην ίδια γραφική παράσταση μπορούν να απεικονιστούν περισσότερα του ενός γραφήματα. Αυτό βοηθά στην εύρεση κοινών συμπεριφορών στις διάφορες παραμέτρους των δεδομένων.



Εικόνα 6. Γράφημα με σημάδια πυκνότητας (heatmap) στο Tableau

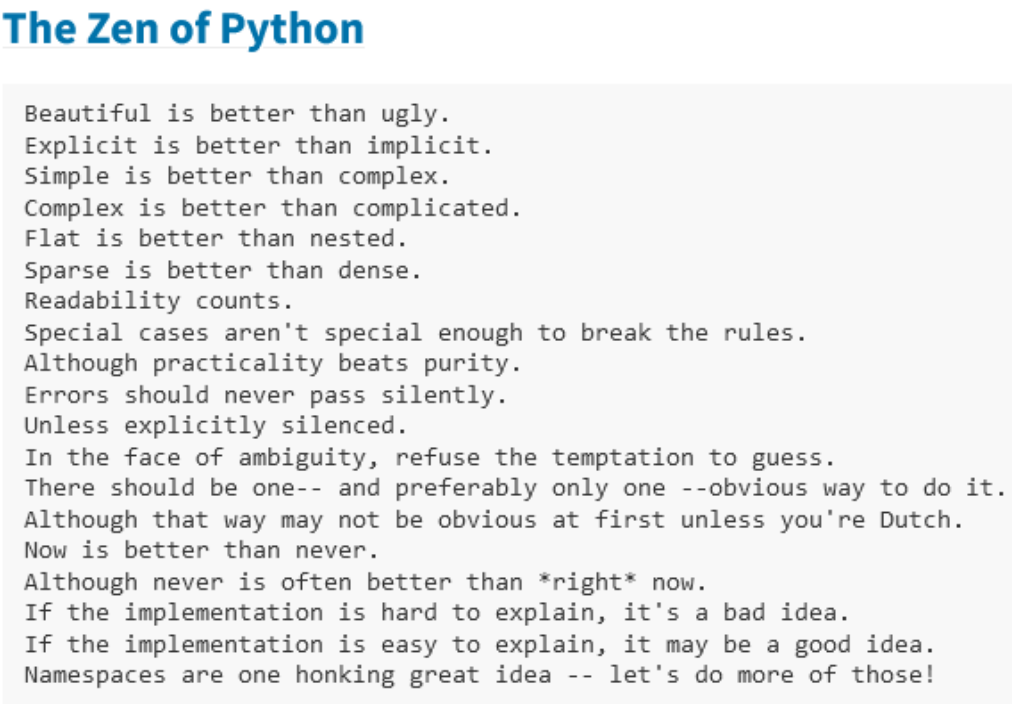
Επίσης σημαντική, είναι η λειτουργία δημιουργίας πίνακα ελέγχου (dashboard) στο Tableau. Ο πίνακας ελέγχου, εκτός από την οπτική αναπαράσταση των δεδομένων, προσφέρει τη δυνατότητα επιλογής πληροφοριών προς εμφάνιση, με τη χρήση φίλτρων.

2.2.2. Η γλώσσα προγραμματισμού (πthon) Python

Όπως αναφέρεται στον επίσημο ιστότοπο της Python, *«Η Python είναι μια γλώσσα προγραμματισμού που σας επιτρέπει να εργάζεστε γρήγορα και να ενσωματώνετε συστήματα πιο αποτελεσματικά.»* (Welcome to Python.org, χ.χ.)

Είναι μια υψηλού επιπέδου αντικειμενοστραφής γλώσσα προγραμματισμού. Είναι σχετικά εύκολη και απλή. Μπορεί να χρησιμοποιηθεί για τη γρήγορη ανάπτυξη ολοκληρωμένων εφαρμογών σε αρκετές πλατφόρμες λειτουργικών συστημάτων, όπως Windows, Linux, Mac OS, κ.ά.. Διαθέτει πολλές και έτοιμες για χρήση βιβλιοθήκες.

Οι εφαρμογές φτιαγμένες σε Python έχουν ευανάγνωστο κώδικα, ενώ η συντήρησή τους γίνεται πιο γρήγορα σε σχέση με άλλες δημοφιλείς γλώσσες προγραμματισμού όπως η C++ και η Java.



Εικόνα 7. Το Zen της Python

Γιατί όμως η Python χρησιμοποιείται τόσο πολύ στη μηχανική μάθηση;

Είναι εξαιρετικά ευανάγνωστη. Είναι μία γλώσσα ιδιαίτερα φιλική για τους αρχάριους. Έχει συνεπή σύνταξη ώστε είναι εύκολη τόσο η ανάγνωση του κώδικα όσο και η σύνταξη νέου.

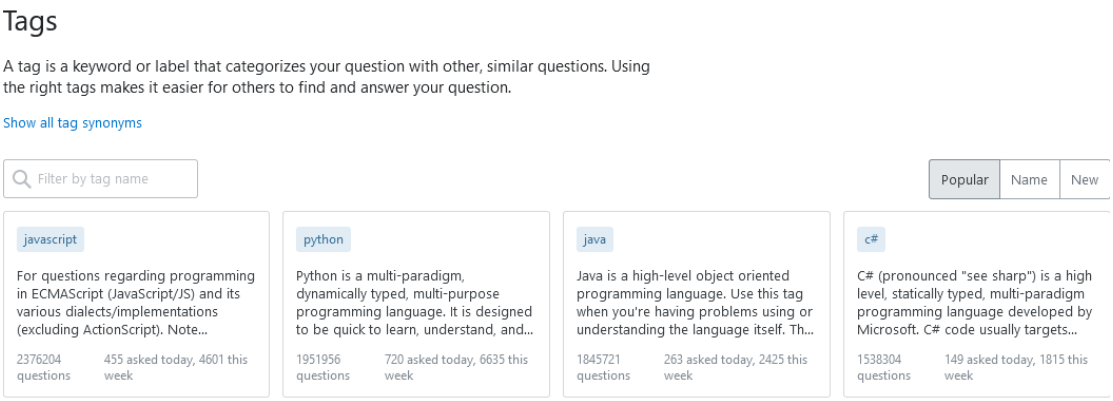
Είναι εύκολη στην εκμάθηση. Έτσι, οι προγραμματιστές μπορούν να δοκιμάσουν γρηγορότερα τον κώδικά τους και να δουν τα αποτελέσματά του εφόσον απαιτείται λιγότερος χρόνος για να την κατανοήσουν.

Έχει μαθηματική σημασιολογία. Παρά την απλότητά της, είναι παρόμοια με πολλές μαθηματικές έννοιες, κάτι που προσφέρεται για τα μαθηματικά που είναι απαραίτητα στη μηχανική μάθηση.

Είναι ανοικτού κώδικα. Ως επακόλουθο, οι προγραμματιστές μπορούν να την προσαρμόσουν στις δικές τους ανάγκες. Συνεπώς, έχει ταχύτερη ανάπτυξη ενώ αξιολογείται συνεχώς από τα μέλη της ανοικτής της κοινότητας ώστε τα σφάλματα να μπορούν να διορθωθούν γρηγορότερα.

Είναι πολύ δημοφιλής. Οι προγραμματιστές αναζητούν στις μηχανές αναζήτησης λύσεις για να ολοκληρώσουν τον κώδικά τους. Μία, κορυφαία σε επισκεψιμότητα, διαδικτυακή κοινότητα όπου προγραμματιστές θέτουν ερωτήματα και αναζητούν απαντήσεις, είναι η Stack Overflow. Η Python, είναι η δεύτερη πιο συζητούμενη γλώσσα, με βάση τις ετικέτες, στην κοινότητα αυτή.

Διαθέτει πολλές βιβλιοθήκες. Ενσωματώνονται εύκολα στην Python, πολλές μάλιστα, όπως οι NumPy, SciPy, PyTorch και scikit-learn, έχουν σχεδιαστεί για χρήση στη μηχανική μάθηση.



Εικόνα 8. Οι πιο δημοφιλείς γλώσσες προγραμματισμού στο Stack Overflow

2.2.3. Το Jupyter Notebook

Το Jupyter υποστηρίζει τη δια-δραστική επιστήμη δεδομένων και τον επιστημονικό υπολογισμό σε όλες τις γλώσσες προγραμματισμού. (About Us: Project Jupyter's origins and governance, χ.χ.)

Πιο απλά, είναι μία διαδικτυακή εφαρμογή. Μία εφαρμογή που μπορεί να εκτελεστεί από έναν περιηγητή διαδικτύου. Υποστηριζόμενοι περιηγητές είναι οι τελευταίες εκδόσεις των

- Chrome,
- Safari και
- Firefox

καθιστώντας ικανή την εκτέλεσή του από σχεδόν κάθε ηλεκτρονικό υπολογιστή ανά τον κόσμο.

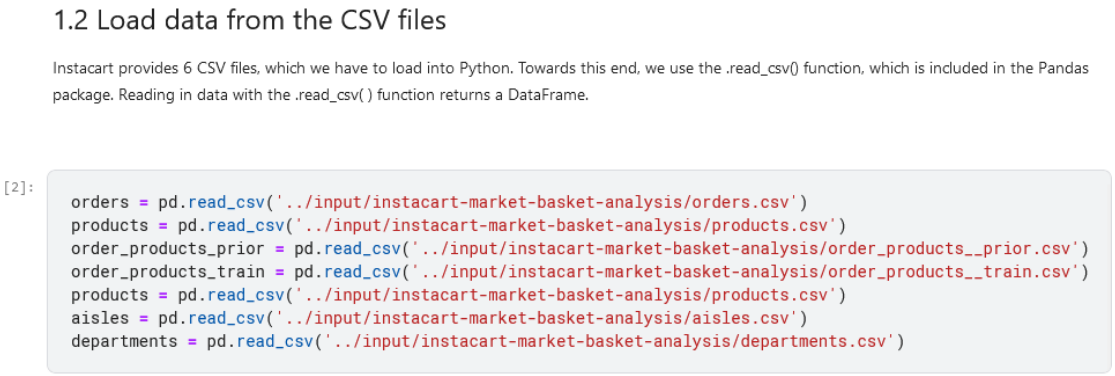
Η χρήση του αφορά τρία (3) βασικά χαρακτηριστικά:

1. τον κώδικα,
2. το σημειωματάριο και
3. τα δεδομένα.

Είναι ένα δια-δραστικό περιβάλλον ανάπτυξης (IDE: Interactive Development Environment) προγραμματιστικού κώδικα. Επιπλέον, δίνει τη δυνατότητα εκτέλεσης του κώδικα είτε τμηματικά είτε ολόκληρου, με ταυτόχρονη προβολή τού αποτελέσματος είτε αφορά μαθηματικές πράξεις είτε γραφικές παραστάσεις ή άλλες απεικονίσεις. Υποστηρίζει πάνω από σαράντα γλώσσες προγραμματισμού, συμπεριλαμβανομένων των Python, R, Julia, και Scala.

Το σημειωματάριο που βρίσκεται ενσωματωμένο σε αυτό, δίνει τη δυνατότητα επεξήγησης του προγραμματιστικού κώδικα. Διαθέτει τη λειτουργία των σχολίων που προσφέρουν άλλες πολύ δημοφιλείς γλώσσες, όπως η HTML και η PHP. Στην περίπτωση του Jupyter όμως, τα σχόλια είναι αποκομμένα από τον κώδικα ανακαλύπτοντας μία νέα δομή στο σύνολο ενός έργου προς υλοποίηση.

Καθώς το εν λόγω notebook χρησιμοποιείται ειδικότερα στην επιστήμη των δεδομένων και τη μηχανική μάθηση, προσφέρει τη δυνατότητα εισαγωγής μεγάλων συνόλων δεδομένων. Δεδομένα, τα οποία θα αναλυθούν και στα οποία θα ανακαλυφθούν μοτίβα ή θα δημιουργηθούν νέες παράμετροι για να ακολουθήσει η περεταίρω επεξεργασία τους.



1.2 Load data from the CSV files

Instacart provides 6 CSV files, which we have to load into Python. Towards this end, we use the .read_csv() function, which is included in the Pandas package. Reading in data with the .read_csv( ) function returns a DataFrame.

[2]:

```
orders = pd.read_csv('../input/instacart-market-basket-analysis/orders.csv')
products = pd.read_csv('../input/instacart-market-basket-analysis/products.csv')
order_products_prior = pd.read_csv('../input/instacart-market-basket-analysis/order_products__prior.csv')
order_products_train = pd.read_csv('../input/instacart-market-basket-analysis/order_products__train.csv')
products = pd.read_csv('../input/instacart-market-basket-analysis/products.csv')
aisles = pd.read_csv('../input/instacart-market-basket-analysis/aisles.csv')
departments = pd.read_csv('../input/instacart-market-basket-analysis/departments.csv')
```

Εικόνα 9. Σημειώσεις, κώδικας και εισαγωγή δεδομένων στο Jupyter Notebook

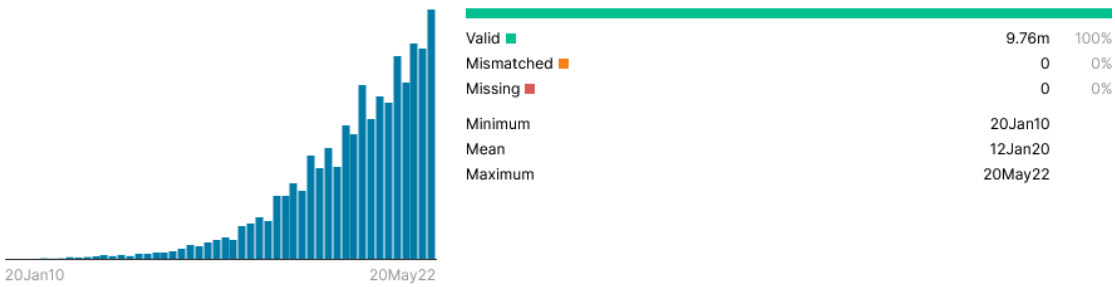
Το περιβάλλον του Jupyter επιτρέπει την εισαγωγή επεκτάσεων ώστε να επεκτείνουν και να εμπλουτίσουν τη λειτουργικότητά του.

Τέλος, τα σημειωματάρια μπορούν να αποθηκευτούν σε κάποιο μέσο αποθήκευσης ή να μοιραστούν με άλλους με τη χρήση ηλεκτρονικού ταχυδρομείου, πλατφόρμες διαμοιρασμού αρχείων όπως το Dropbox κ.ά..

2.2.4. Η κοινότητα Kaggle

Η Kaggle είναι μια διαδικτυακή κοινότητα. Ιδρύθηκε το έτος 2010. Μέλη της είναι επιστήμονες δεδομένων. Από τις 20 Ιανουαρίου του 2010 έως σήμερα αριθμεί σχεδόν δέκα εκατομμύρια (10.000.000) μέλη, με αυξανόμενο ρυθμό εγγραφής νέων μελών. Μάλιστα, μόνο το τελευταίο τρίμηνο, από 18 Φεβρουαρίου έως 20 Μαΐου του 2022, οι νέες εγγραφές έχουν ξεπεράσει τις οκτακόσιες χιλιάδες (800.000) μέλη.

Ο αριθμός των ακολούθων στα κοινωνικά δίκτυα είναι περί των τριακοσίων τριάντα χιλιάδων (330.000) στο Linkedin, διακοσίων σαράντα (240.000) στο Twitter και εβδομήντα χιλιάδων (70.000) στο Facebook.



Εικόνα 10. Εγγεγραμμένα μέλη Kaggle

Επιπλέον, η Kaggle προσφέρει ένα διαδικτυακό περιβάλλον συγγραφής προγραμματιστικού κώδικα, το Jupyter Notebook, παρέχοντας ταυτόχρονα και δωρεάν υπολογιστική ισχύ για την εκτέλεσή του.

Μέσα από το Jupyter Notebook της Kaggle, υπάρχει η δυνατότητα να εκτελεστεί κώδικας Python ώστε να καθαριστούν τα δεδομένα και να ομαδοποιηθούν ή να ανακαλυφθούν νέες παράμετροι ώστε στη συνέχεια να τροφοδοτήσουν και να εκπαιδεύσουν έναν αλγόριθμο μηχανικής μάθησης.

Συνεπώς, τρία (3) είναι τα βασικά στοιχεία που κάνουν ιδανική τη χρήση της πλατφόρμας της Kaggle, και είναι τα εξής:

1. Παροχή χρήσης του Jupyter Notebook
2. Δυνατότητα εκτέλεσης κώδικα Python
3. Παροχή δωρεάν υπολογιστικής ισχύος

Η δημοφιλής, έπειτα από επτά χρόνια παρουσίας, Kaggle, εξαγοράστηκε από την Google το 2017. (Moyer, 2017)

3. Βιβλιογραφική επισκόπηση του προβλήματος

Η ικανότητα αποπληρωμής των δανείων είναι ένα πρόβλημα που απασχολεί εδώ και δεκαετίες. Από τα μέσα της δεκαετίας του 1960, εφαρμόζονται με αυξανόμενη συχνότητα η αξιολόγηση πιστοληπτικής ικανότητας και οι σχετικές διαδικασίες αναθεώρησης δανείων. Δεδομένης όμως της φύσης των συστημάτων αξιολόγησης καθώς είναι ιδιόκτητα, το περιεχόμενο των μοντέλων είναι σχεδόν άγνωστο. Στην πραγματικότητα, τα έως τότε μοντέλα αξιολόγησης έχουν κυρίως εστιάσει στην ελαχιστοποίηση των ποσοστών αθέτησης υποχρεώσεων (Eisenbeis, 1978). Αυτό βέβαια είναι η μία μόνο όψη του νομίσματος. Από τη στιγμή όμως που στόχος των χρηματοπιστωτικών ιδρυμάτων είναι η μεγιστοποίηση του κέρδους ή η ελαχιστοποίηση του κόστους, τότε είναι εμφανές πως όλα αυτά τα μοντέλα φαίνονται ημιτελή.

Το 1996, εξετάστηκε παρόμοιο πρόβλημα με σκοπό τη διάκριση καλών και κακών πιστωτικών κινδύνων μεταξύ ενός πληθυσμού. Προτάθηκε η εφαρμογή της μεθόδου k-NN (k-Nearest-Neighbour) και διαπιστώθηκε πως η μέθοδος απέδωσε καλά, επιτυγχάνοντας το χαμηλότερο αναμενόμενο ποσοστό πιστωτικού κινδύνου (Henley & Hand, 1996). Με τη συγκεκριμένη μέθοδο, επιβεβαιώθηκε η εγκυρότητα των αποτελεσμάτων τους για πληθυσμούς με χαμηλότερα ποσοστά πιστωτικών κινδύνων σε σχέση με τον πλήρη πληθυσμό ενώ υποστηρίχθηκε ότι θα πρέπει να χρησιμοποιείται ένα σύνολο σχεδιασμού με ίσες αναλογίες καλών και κακών κινδύνων για την ταξινόμηση των μελλοντικών αιτούντων ανεξάρτητα από το ποσοστό πιστωτικού κινδύνου του πληθυσμού.

Το 2012, υλοποιήθηκε μία σύγκριση των τεχνικών στατιστικής μοντελοποίησης της απλής λογιστικής παλινδρόμησης (naive logistic regression) και λογιστικής παλινδρόμησης με επιλογή δείγματος εξαρτώμενη από την κατάσταση (logistic regression with state-dependent sample selection). Η σύγκριση πραγματοποιήθηκε με πραγματική μελέτη περίπτωσης (Louzada, Ferreira-Silva & Diniz, 2012) σε ένα πραγματικό σύνολο δεδομένων τράπεζας της Βραζιλίας. Η μελέτη έδειξε ότι ανεξάρτητα από το ποια από αυτές τις δύο τεχνικές χρησιμοποιείται, υπάρχει ανάγκη για εργασία με ισορροπημένα δείγματα, τα οποία διασφαλίζουν μοντέλα με καλές μετρήσεις ευαισθησίας και ειδικότητας και υψηλό ποσοστό ακρίβειας. Οι δύο τεχνικές, έχουν παρόμοια ικανότητα πρόβλεψης. Ιδανική τεχνική κρίνεται η λογιστική παλινδρόμηση με επιλογή δείγματος που εξαρτάται από την κατάσταση (logistic regression with state-dependent sample selection) που οδηγεί στην πραγματική πιθανότητα αθέτησης. Το απλό μοντέλο λογιστικής παλινδρόμησης (naive logistic regression) υποεκτιμά την πιθανότητα αθέτησης.

Οι προβληματισμοί για τη σωστή διαχείριση του πιστωτικού κινδύνου συνεχίστηκαν καθώς είναι απαραίτητη για τα πιστωτικά ιδρύματα, εφόσον μπορεί να προκύψουν σημαντικές ζημίες όταν οι δανειολήπτες χρεοκοπούν. Σε άλλη μελέτη τους χρησιμοποιούνται και αναλύονται 11 μέθοδοι μηχανικής μάθησης για να προβλέψουν τον πιστωτικό κίνδυνο με βάση τα χαρακτηριστικά των πελατών και να συγκρίνουν την ακρίβεια πρόβλεψής τους (Hamori, Kawai, Kume, Murakami & Watanabe, 2018).

Συγκεκριμένα, χρησιμοποιήθηκαν τρεις μέθοδοι εκμάθησης συνόλου (bagging, random forest και boosting) και οκτώ μέθοδοι νευρωνικών δικτύων με διαφορετικές λειτουργίες ενεργοποίησης. Η απόδοση κάθε μεθόδου συγκρίθηκε ως προς την ικανότητά τους να προβλέπουν τον πιστωτικό κίνδυνο χρησιμοποιώντας πολλαπλούς δείκτες:

- ακρίβεια (accuracy),
- ρυθμός πρόβλεψης (rate of prediction),
- αποτελέσματα (results),
- καμπύλη ROC (receiver operating characteristic (ROC) curve),
- περιοχή κάτω από την καμπύλη ROC (AUC) και
- βαθμολογία F (F-score)

Τα αποτελέσματα που ελήφθησαν δείχνουν ότι η ικανότητα ταξινόμησης της ενίσχυσης (boosting) είναι ανώτερη από άλλες μεθόδους μηχανικής μάθησης, συμπεριλαμβανομένων των νευρωνικών δικτύων. Διαπιστώθηκε επίσης ότι η απόδοση των μοντέλων νευρωνικών δικτύων εξαρτάται από την επιλογή της συνάρτησης ενεργοποίησης και τον αριθμό των μεσαίων στρωμάτων.

Λόγω της προηγμένης τεχνολογίας που σχετίζεται με τα Big Data, τη διαθεσιμότητα δεδομένων και την υπολογιστική ισχύ, οι περισσότερες τράπεζες ή πιστωτικά ιδρύματα ανανεώνουν τα επιχειρηματικά τους μοντέλα. Οι προβλέψεις πιστωτικού κινδύνου και η αποτελεσματική διεκπεραίωση δανείων είναι βασικά για τη λήψη αποφάσεων του κλάδου αυτού.

Σε σχετική μελέτη (Addo, Guegan & Hassani, 2018), κατασκευάστηκαν δυαδικοί ταξινομητές βασισμένοι σε μοντέλα μηχανικής και βαθιάς μάθησης με σκοπό την πρόβλεψη της πιθανότητας αθέτησης δανείων. Η μελέτη βασίστηκε σε πραγματικά δεδομένα των ετών 2016 και 2017 και αφορούσαν εταιρίες. Το σύνολο δεδομένων περιέχει 117.019 γραμμές, καθεμία από τις οποίες αντιπροσωπεύει είτε αθέτηση υποχρεώσεων είτε όχι (δυαδική τιμή) μιας επιχείρησης για ένα δάνειο από μια τράπεζα. Οι 115.288 γραμμές αντιπροσώπευαν εταιρείες με καλή υγεία και 1.731 αντιπροσώπευαν εταιρείες σε αδυναμία πληρωμής. Οι 235 μεταβλητές λήφθηκαν απευθείας από τις εταιρείες και αφορούν οικονομικές καταστάσεις, ισολογισμούς και ταμειακές ροές. Παρατηρήθηκε ότι τα μοντέλα που βασίζονται σε δέντρα είναι πιο σταθερά από τα μοντέλα που βασίζονται σε πολυστρωματικά τεχνητά νευρωνικά δίκτυα. Αυτό ανοίγει πολλά ερωτήματα σχετικά με την εντατική χρήση συστημάτων βαθιάς μάθησης στις επιχειρήσεις.

Μία άλλη προσέγγιση ήταν η διαμόρφωση ενός μοντέλου εκμάθησης επαναδειγματοληψίας συνόλου, με μηχανές υποστήριξης διανυσμάτων SVM (support vector machines) με βάση ένα δίκτυο βαθιάς πεποίθησης DBN (deep belief network) (Len Yu, Zhou, Tang & Chen, 2018). Σκοπός, η ταξινόμηση πιστώσεων με μη ισορροπημένα δεδομένα με παράλληλη προσπάθεια του μοντέλου να βελτιώσει την απόδοσή του εύλογα, ειδικά όταν αντιμετωπίζεται το πρόβλημα της ανισορροπίας δεδομένων. [*] Τα σύνολα δεδομένων που χρησιμοποιήθηκαν ήταν πραγματικά πιστωτικά δεδομένα από τη

Γερμανία (Statlog (German Credit Data) Data Set) και την Ιαπωνία (Credit Approval Data Set) και βρίσκονται διαθέσιμα στο UCI Machine Learning Repository (https://archive.ics.uci.edu/). Το DBN-SVM δείχνει να είναι ένα αποτελεσματικό εργαλείο για την επίλυση του προβλημάτων μη ισορροπημένων δεδομένων στην ταξινόμηση πιστώσεων.

4. Μεθοδολογία

Η απόφαση για τη μεθοδολογία που θα ακολουθηθεί είναι σημαντική. Η μεθοδολογία θα έχει αντίκτυπο τόσο στην ποιότητα της μελέτης όσο και στο συνολικό χρόνο που θα δαπανηθεί για αυτήν.

Σε πρώτη φάση, αφού το πεδίο του προβλήματος είναι γνωστό, πρέπει να γίνει αναζήτηση για σύνολα δεδομένων. Τα δεδομένα θα πρέπει άμεσα να αφορούν δανειολήπτες. Για να αποκτηθούν δεδομένα, δύο είναι οι λύσεις:

1. Συλλογή δεδομένων από τη πλευρά των δανειοληπτών, ή
2. συλλογή δεδομένων από τη πλευρά των δανειοδοτών.

Η πρώτη περίπτωση είναι εφικτή με άντληση πληροφοριών μετά από συνάντηση με δανειολήπτες. Έπειτα, πρέπει να γίνει εισαγωγή των πληροφοριών σε κάποιο πληροφοριακό σύστημα ή βάση δεδομένων. Είναι μία χρονοβόρα διαδικασία και απαιτεί πολύ κόπο.

Η δεύτερη περίπτωση, είναι να γίνει προσέγγιση σε κάποια τράπεζα που εκδίδει δάνεια. Ο χρόνος και ο κόπος είναι μηδαμινοί. Οι τράπεζες διατηρούν καταχωρημένα στοιχεία για τους πελάτες τους. Άλλωστε, είναι δουλειά λίγων ωρών να δώσει μία τράπεζα στοιχεία για ένα εκατομμύριο πελάτες της από το να γίνει συνάντηση με ένα εκατομμύριο ανθρώπους για να αντληθούν πληροφορίες. Αν υπολογιστεί δε και ένας αριθμός ανθρώπων που δε θέλουν να μοιραστούν τα στοιχεία τους ή δεν έχουν πάρει δάνειο, τότε τα πράγματα δυσκολεύουν περισσότερο.

Μία ενδιάμεση λύση είναι η αναζήτηση σε ανοικτά δεδομένα (open data) είτε από ιδιώτες είτε από κυβερνήσεις (gov data).

Παράλληλα με τη διαδικασία αναζήτησης δεδομένων, θα αναζητηθεί μήπως το πεδίο του προβλήματος είναι ήδη γνωστό και υπάρχουν άνθρωποι να μοιραστούν δεδομένα, ιδέες ή λύσεις.

Εφόσον αποκτηθούν τα δεδομένα, το επόμενο στάδιο είναι η γνωριμία με αυτά. Πρέπει να είναι γνωστές οι παράμετροι και τα χαρακτηριστικά αλλά και το εύρος των τιμών που μπορεί να έχουν. Τα δεδομένα πρέπει να επεξηγηθούν για να γίνουν κατανοητά.

Επόμενη φάση, είναι μία διερευνητική ανάλυση των δεδομένων. Με ανάλογες οπτικοποιήσεις από την εφαρμογή Tableau, τα δεδομένα μπορούν να δώσουν αρκετές πληροφορίες ώστε να ληφθούν αποφάσεις για το ποια είναι λιγότερο ή περισσότερο σημαντικά.

Στη συνέχεια, ανάλογα με τις αποφάσεις που λήφθηκαν προηγουμένως, θα γίνει μία προ-επεξεργασία στα δεδομένα. Κάποια ενδέχεται θα απορριφθούν και άλλα να ομαδοποιηθούν. Πάντα όμως, υπάρχει η πιθανότητα να δημιουργηθούν νέα δεδομένα.

Σκοπός είναι η δημιουργία ενός συνόλου δεδομένων, που θα τροφοδοτήσει τον αλγόριθμο μηχανικής μάθησης. Αυτό είναι και το τελευταίο στάδιο της μεθοδολογίας.

5. Αναζήτηση δεδομένων του πεδίου προβλήματος

Έπειτα από ενδελεχή αναζήτηση για διαθέσιμα δεδομένα χρηματοπιστωτικών ιδρυμάτων, προέκυψε ένα συμπέρασμα. Το κατάλληλο μέρος για να βρει κανείς δωρεάν διαθέσιμα μεγάλα δεδομένα, είναι η κοινότητα του Kaggle. Η Kaggle επίσης, είναι το κατάλληλο μέρος για να θέσει κανείς προβλήματα που για τη λύση τους απαιτείται μηχανική μάθηση

Όπως χαρακτηριστικά αναφέρεται στην επίσημη σελίδα της στο Facebook, η Kaggle παρέχει επιστήμη δεδομένων αιχμής, περισσότερο γρήγορα και καλύτερα από ό,τι θα μπορούσαν να φανταστούν οι περισσότεροι άνθρωποι. Έχει ένα αποδεδειγμένο ιστορικό επίλυσης προβλημάτων του πραγματικού κόσμου σε πολλούς τομείς, συμπεριλαμβανομένων των φαρμακοβιομηχανιών, των χρηματοοικονομικών, του λιανικού εμπορίου κ.ά.. Η Kaggle, γενικότερα προσφέρει διαγωνισμούς επιστήμης δεδομένων και ειδικότερα συμβουλευτικές υπηρεσίες κατ' απαίτηση. (Kaggle: Overview, 2022)

Η Kaggle επιτρέπει στα μέλη της να δημοσιεύουν σύνολα δεδομένων καθώς και να έχουν πρόσβαση σε δημοσιευμένα σύνολα δεδομένων άλλων μελών. Με τη διάθεση αυτού του όγκου των δεδομένων, επιστήμονες δεδομένων έχουν τη δυνατότητα να τα εξερευνούν και να τα αναλύουν δημιουργώντας διάφορα μοντέλα. Τα μοντέλα βοηθούν στην κατανόηση της συμπεριφοράς των δεδομένων και μετέπειτα στη δυνατότητα πρόβλεψης καταστάσεων.

Κατά τη διαδικασία αναζήτησης για δεδομένα χρηματοπιστωτικών ιδρυμάτων, ανακαλύφθηκε η Home Credit. Η εταιρία αυτή, αποφάσισε να διενεργήσει ένα διαγωνισμό στο Kaggle. Μάλιστα, έχει διαθέσει ένα μεγάλο σύνολο δεδομένων σχετικά με πελάτες της δανειολήπτες.

Τα δεδομένα που έχει διαθέσει δημόσια η Home Credit, βρίσκονται εντός του πεδίου του προβλήματος της παρούσας διπλωματικής εργασίας. Ως εκ τούτου, δεν μπορεί να μείνουν ανεκμετάλλευτα και θα είναι αυτά που θα χρησιμοποιηθούν.

6. Τα σύνολα δεδομένων

6.1. Το χρηματοπιστωτικό ίδρυμα Home Credit

Η εταιρία Home Credit είναι ένα διεθνές μη τραπεζικό χρηματοπιστωτικό ίδρυμα με δραστηριότητα σε εννέα (9) χώρες. Έχει έδρα στην Ολλανδία, ενώ ιδρύθηκε στην Τσεχία το 1977. Κύριος σκοπός της είναι η παροχή δανείων κυρίως σε άτομα με ελάχιστο ή καθόλου πιστωτικό ιστορικό.

Οι χώρες στις οποίες δραστηριοποιείται αριθμούν περίπου τρεισήμισι δισεκατομμύρια (3.500.000.000) κατοίκους. Από την ίδρυσή της μέχρι σήμερα, έχει χορηγήσει 267.707.088 δάνεια. Μόνο σήμερα, μέχρι τη συγγραφή αυτής της πρότασης, έχει χορηγήσει 28.840 δάνεια.

Η Home Credit, προσπαθεί να παρέχει μία θετική και ασφαλή εμπειρία δανεισμού, τόσο για την ίδια όσο και για το δανειολήπτη.

Εικόνα 11. Χορηγούμενα δάνεια Home Credit

6.2. Περιγραφή του προβλήματος

Πολλοί άνθρωποι επιθυμούν να πάρουν δάνειο, χωρίς να υπάρχει για αυτούς κάποιο ιστορικό πίστωσης. Μέρος αυτού του πληθυσμού ανήκει στην ομάδα των αναξιόπιστων δανειοληπτών.

Η Home Credit, προσπαθώντας να παρέχει μία θετική και ασφαλή εμπειρία δανεισμού, προκάλεσε την κοινότητα των ανθρώπων που ασχολούνται με τη μηχανική μάθηση να ξεκλειδώσουν πλήρως τις δυνατότητες των δεδομένων της. Προκειμένου να το εξασφαλίσει αυτό, χρησιμοποιεί μια ποικιλία δεδομένων για να προβλέψει τις ικανότητες αποπληρωμής των πελατών της. Για να κάνει αυτές τις προβλέψεις, χρησιμοποιεί διάφορες μεθόδους στατιστικής και μηχανικής μάθησης.

Στόχος της Home Credit είναι να γίνονται προβλέψεις με μεγαλύτερη ακρίβεια, εξασφαλίζοντας ταυτόχρονα ότι:

- δε θα απορριφθούν πελάτες με δυνατότητα αποπληρωμής του δανείου, και
- δε θα δοθούν δάνεια σε πελάτες που δε θα τα αποπληρώσουν.

Έτσι, έθεσε στο διαγωνισμό το ερώτημα «Μπορείτε να προβλέψετε πόσο ικανός είναι κάθε αιτών να αποπληρώσει ένα δάνειο;». (Kaggle: Your Home for Data Science, 2018)

Τα χρήματα που επένδυσε το συγκεκριμένο χρηματοπιστωτικό ίδρυμα ήταν 70.000$ και έδωσε τα εξής χρηματικά βραβεία:

- 35.000$ για την πρώτη θέση,
- 25.000$ για τη δεύτερη θέση και
- 10.000$ για την τρίτη θέση.

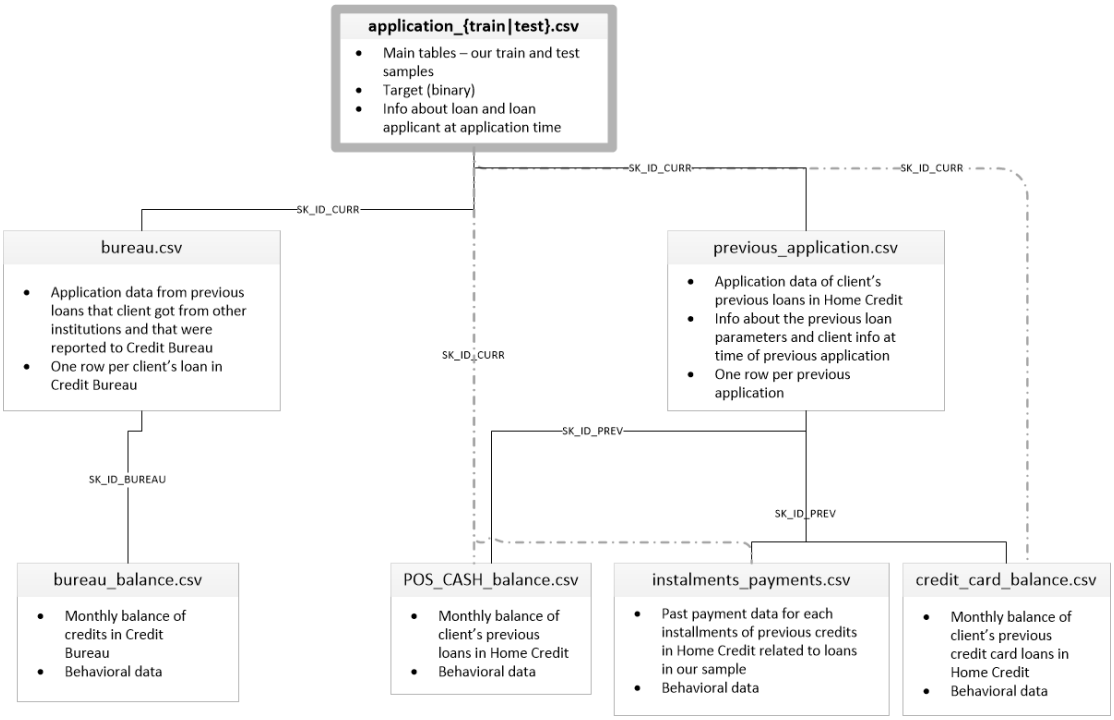
Σαφώς, έδωσε και όλα τα δεδομένα που έπρεπε στους διαγωνιζόμενους, ώστε να μπορέσουν να ανακαλύψουν νέα δεδομένα και να προβλέψουν με ακρίβεια το αποτέλεσμα.

6.3. Επεξήγηση των δεδομένων

Τα σύνολα δεδομένων που διατίθενται για τη μελέτη, περιλαμβάνουν εννέα (9) αρχεία, με τιμές διαχωρισμένες με κόμμα (csv: comma separated values). Αυτά είναι τα:

- application_test.csv
- application_train.csv
- bureau.csv
- bureau_balance.csv
- POS_CASH_balance.csv
- credit_card_balance.csv
- previous_application.csv
- installments_payments.csv

Οι πληροφορίες που δίνονται στα παραπάνω σύνολα δεδομένων, μπορεί να είναι άλλες λιγότερο και άλλες περισσότερο σημαντικές στο να προβλεφθεί αν είναι ικανός κάποιος που αιτείται δάνειο να το αποπληρώσει ή όχι.



Εικόνα 12. Η σχέση ανάμεσα στα αρχεία δεδομένων

Όλες οι πληροφορίες αξιολογούνται. Δεν είναι γνωστό εξ αρχής ποιες είναι σημαντικότερες από τις υπόλοιπες. Εξ αιτίας όμως αυτών των δεδομένων πληροφοριών απαιτείται να γίνει μία πρόβλεψη. Για το λόγο αυτό, όλες ονομάζονται μεταβλητές πρόβλεψης.

Συχνά, από τις μεταβλητές που δίνονται για τη δημιουργία μίας πρόβλεψης, δημιουργούνται νέες μεταβλητές, όπως για παράδειγμα σύνολα ίδιων τιμών, μέσος όρων τιμών κ.ά..

Στην εικόνα 12, φαίνονται οι σχέσεις που έχουν τα σύνολα δεδομένων, όπως είναι αναρτημένες στη σελίδα του διαγωνισμού στην κοινότητα Kaggle από όπου και λήφθηκαν.

Στη συνέχεια θα γίνει μία σύντομη αναφορά στο κάθε αρχείο καθώς και στις μεταβλητές πρόβλεψης που έχουν αποθηκευμένες.

6.3.1. Αρχεία application_test.csv και application_train.csv

Τα αρχεία application_test.csv και application_train.csv περιλαμβάνουν τις ίδιες μεταβλητές. Κάθε σειρά, αντιπροσωπεύει ένα ξεχωριστό δάνειο.

Η διαφορά στα δύο αρχεία είναι ότι στο αρχείο application_train.csv υπάρχει η πληροφορία για το αν το κάθε δάνειο είχε κάποιο πρόβλημα στην πληρωμή του. Στο αρχείο application_test.csv, η πληροφορία αυτή δεν υπάρχει.

Στη συνέχεια, γίνεται μία περιγραφική αναφορά σε μεταβλητές πρόβλεψης που παρέχουν τα σύνολα δεδομένων, ξεκινώντας από τις γενικές και αταξινόμητες.

Πίνακας 1. Γενικές / αταξινόμητες μεταβλητές αίτησης δανείου

Μεταβλητή	Περιγραφή
SK_ID_CURR	Μοναδικός αριθμός του κάθε δανείου.
TARGET	Μεταβλητή απόκρισης. Όπου 1: πελάτης με δυσκολίες πληρωμής. Όπου 0: όλες οι άλλες περιπτώσεις.
NAME_TYPE_SUITE	Ποιος συνόδευε τον πελάτη όταν έκανε αίτηση για το δάνειο (Οικογένεια, τέκνο, συνεργάτης κ.ά.).
NAME_EDUCATION_TYPE	Επίπεδο ανώτερης εκπαίδευσης που έχει ο πελάτης (απόφοιτος λυκείου κ.ά.).
WEEKDAY_APPR_PROCESS_START	Ποια ημέρα της εβδομάδας έκανε ο πελάτης αίτηση για το δάνειο;
HOUR_APPR_PROCESS_START	Περίπου ποια ώρα ο πελάτης έκανε αίτηση για το δάνειο;

Η μεταβλητή TARGET, είναι σαφώς η πρόβλεψη που πρέπει να γίνει. Είναι απαραίτητο να είναι γνωστή για να γίνει εκπαίδευση του αλγορίθμου μηχανικής μάθησης. Ονομάζεται μεταβλητή απόκρισης ή μεταβλητή στόχου. Μόλις ο αλγόριθμος εκπαιδευτεί, θα κληθεί να προβλέψει νέα, άγνωστα προς αυτόν δεδομένα, στα οποία η μεταβλητή απόκρισης δε θα είναι διαθέσιμη.

Για κάθε νέα αίτηση, η Home Credit, διατηρεί δημογραφικά στοιχεία για τους υποψήφιους πελάτες της, όπως δείχνει ο επόμενος πίνακας.

Πίνακας 2. Δημογραφικά στοιχεία πελάτη

Μεταβλητή	Περιγραφή
CODE_GENDER	Το φύλο του πελάτη.
DAYS_BIRTH	Η ηλικία του πελάτη σε ημέρες κατά τη στιγμή της αίτησης του δανείου.

Επιπλέον, καταχωρούνται στοιχεία της οικογενειακής κατάστασης του πελάτη, ώστε να είναι γνωστό αν είναι συζευγμένος ή όχι και πόσα παιδιά έχει.

Πίνακας 3. Οικογενειακή κατάσταση πελάτη

Μεταβλητή	Περιγραφή
CNT_CHILDREN	Το πλήθος των παιδιών που έχει ο πελάτης.
NAME_FAMILY_STATUS	Η οικογενειακή κατάσταση του πελάτη.
CNT_FAM_MEMBERS	Αριθμός μελών οικογένειας που έχει ο πελάτης.

Το χρηματοπιστωτικό ίδρυμα, επιθυμεί να γνωρίζει πληροφορίες για την οικονομική και επαγγελματική κατάσταση των πελατών του. Ζητά από τους αιτούντες πληροφορίες σχετικά με το εισόδημά τους, θέλοντας παράλληλα να γνωρίζει και κάποια στοιχεία σχετικά με την επαγγελματική τους ιδιότητα.

Πίνακας 4. Οικονομική και επαγγελματική κατάσταση πελάτη

Μεταβλητή	Περιγραφή
AMT_INCOME_TOTAL	Το εισόδημα του πελάτη.
NAME_INCOME_TYPE	Ο τύπος εισοδήματος (επιχειρηματίας, εργαζόμενος, άνεργος κ.ά.).
DAYS_EMPLOYED	Πόσες ημέρες πριν από την αίτηση ο αιτών ξεκίνησε την τρέχουσα απασχόληση.
OCCUPATION_TYPE	Τι είδους επάγγελμα κάνει ο αιτών (πωλητής, οδηγός κ.ά.).
ORGANIZATION_TYPE	Τύπος οργανισμού όπου εργάζεται ο αιτών (σχολείο, δημόσιο κ.ά.).

Επιπλέον στοιχεία που συλλέγει η Home Credit, είναι αν ο πελάτης διαθέτει αυτοκίνητο ή σπίτι.

Πίνακας 5. Περιουσιακή κατάσταση πελάτη

Μεταβλητή	Περιγραφή
FLAG_OWN_CAR	Επισήμανση εάν ο πελάτης έχει αυτοκίνητο.
OWN_CAR_AGE	Η ηλικία του αυτοκινήτου του πελάτη.
FLAG_OWN_REALTY	Επισήμανση εάν ο πελάτης έχει σπίτι ή διαμέρισμα.
NAME_HOUSING_TYPE	Ποια είναι η στεγαστική κατάσταση του πελάτη (νοικιάζει, συμβιώνει με γονείς κ.ά.).

Πέραν των άλλων, το ίδρυμα ζητά από τον πελάτη στοιχεία ώστε να είναι σε θέση να έρθει σε επαφή μαζί του, σε περίπτωση που χρειαστεί. Για κάποια από αυτά, ζητούνται και σχετικά έγγραφα.

Πίνακας 6. Στοιχεία επικοινωνίας πελάτη / Υποβαλλόμενα έγγραφα

Μεταβλητή	Περιγραφή
FLAG_MOBIL	Παρείχε ο πελάτης κινητό τηλέφωνο; (1=ΝΑΙ, 0=ΟΧΙ).
FLAG_WORK_PHONE	Παρείχε ο πελάτης τηλέφωνο εργασίας; (1=ΝΑΙ, 0=ΟΧΙ).
FLAG_PHONE	Παρείχε ο πελάτης τηλέφωνο σπιτιού; (1=ΝΑΙ, 0=ΟΧΙ).
FLAG_EMAIL	Παρείχε ο πελάτης email; (1=ΝΑΙ, 0=ΟΧΙ).
FLAG_DOCUMENT_2 έως FLAG_DOCUMENT_21	Παρείχε ο πελάτης το έγγραφο 2 έως το έγγραφο 21;

Για τις μεταβλητές FLAG_DOCUMENT_2 έως FLAG_DOCUMENT_21, δεν είναι γνωστό τί ακριβώς αφορά η κάθε μία. Αυτό είναι κάτι που γνωρίζει αποκλειστικά το χρηματοπιστωτικό ίδρυμα και προφανώς δεν ήθελε να μοιραστεί αυτήν την πληροφορία κατά την ανάρτηση των συνόλων δεδομένων. Παρ' όλα αυτά, οι μεταβλητές αυτές, ενδέχεται να παίξουν ρόλο στη ζητούμενη πρόβλεψη.

Πίνακας 7. Έλεγχος παρεχόμενων στοιχείων πελάτη

Μεταβλητή	Περιγραφή
DAYS_ID_PUBLISH	Πόσες ημέρες πριν από την αίτηση άλλαξε ο πελάτης ταυτότητα με την οποία έκανε αίτηση για το δάνειο;
FLAG_CONT_MOBILE	Ήταν προσβάσιμο το κινητό τηλέφωνο; (1=ΝΑΙ, 0=ΟΧΙ).
REG_REGION_NOT_LIVE_REGION	Η μόνιμη διεύθυνση του πελάτη ταιριάζει με τη διεύθυνση επικοινωνίας; (1: διαφορετική, 0: ίδια, σε επίπεδο περιοχής).
REG_REGION_NOT_WORK_REGION	Η μόνιμη διεύθυνση του πελάτη ταιριάζει με τη διεύθυνση εργασίας; (1: διαφορετική, 0: ίδια, σε επίπεδο περιοχής).
LIVE_REGION_NOT_WORK_REGION	Η διεύθυνση επικοινωνίας του πελάτη ταιριάζει με τη διεύθυνση εργασίας; (1: διαφορετική, 0: ίδια, σε επίπεδο περιοχής).
REG_CITY_NOT_LIVE_CITY	Η μόνιμη διεύθυνση του πελάτη ταιριάζει με τη διεύθυνση επικοινωνίας; (1: διαφορετική, 0: ίδια, σε επίπεδο πόλης).
REG_CITY_NOT_WORK_CITY	Η μόνιμη διεύθυνση του πελάτη ταιριάζει με τη διεύθυνση εργασίας; (1: διαφορετική, 0: ίδια, σε επίπεδο πόλης).
LIVE_CITY_NOT_WORK_CITY	Η διεύθυνση επικοινωνίας του πελάτη ταιριάζει με τη διεύθυνση εργασίας; (1: διαφορετική, 0: ίδια, σε επίπεδο πόλης).
DAYS_LAST_PHONE_CHANGE	Πόσες ημέρες πριν από την αίτηση άλλαξε τηλέφωνο ο πελάτης;

Σημαντικές πληροφορίες είναι επίσης αυτές που έχουν σχέση με καθαυτό το αιτούμενο δάνειο. Για παράδειγμα, δεν είναι λογικό, το ετήσιο ποσό καταβολής ενός αιτούμενου δανείου να αγγίζει το ετήσιο εισόδημα του πελάτη.

Πίνακας 8. Στοιχεία αιτούντος δανείου

Μεταβλητή	Περιγραφή
AMT_CREDIT	Πιστωτικό ποσό του δανείου.
AMT_ANNUITY	Ετήσιο ποσό καταβολής δόσεων δανείου.
AMT_GOODS_PRICE	Για τα δάνεια, είναι η τιμή των αγαθών για τα οποία δίνεται το δάνειο.

6.3.2. Αρχείο bureau.csv

Οι πελάτες, ενδέχεται να έχουν προηγούμενα δάνεια που παρέχονται από άλλα χρηματοπιστωτικά ιδρύματα. Σε περίπτωση που αναφέρθηκαν προηγούμενα δάνεια στο Γραφείο Πιστώσεων για πελάτες που έχουν ήδη δάνειο στα παρόντα σύνολα δεδομένων, αυτές περιλαμβάνονται στο αρχείο bureau.csv.

Πίνακας 9. Στοιχεία προηγούμενων δανείων πελάτη

Μεταβλητή	Περιγραφή
SK_ID_CURR	Μοναδικός αριθμός του κάθε δανείου.
SK_BUREAU_ID	Μοναδικός αριθμός προηγούμενου δανείου.
CREDIT_ACTIVE	Κατάσταση προηγούμενου δανείου (ενεργό, ανενεργό κ.ά.).
DAYS_CREDIT	Πόσες ημέρες πριν από την τρέχουσα αίτηση έκανε ο πελάτης αίτηση για δάνειο;
CREDIT_DAY_OVERDUE	Ημέρες καθυστέρησης πληρωμής δόσεων κατά τη στιγμή της αίτησης.
DAYS_CREDIT_ENDDATE	Υπολειπόμενη διάρκεια δανείου (σε ημέρες) κατά τη στιγμή της αίτησης.
DAYS_ENDDATE_FACT	Ημέρες από τη λήξη του δανείου κατά τη στιγμή της αίτησης (μόνο για αποπληρωμένα δάνεια).
AMT_CREDIT_MAX_OVERDUE	Μέγιστο ληξιπρόθεσμο ποσό δανείου έως την ημερομηνία αίτησης δανείου.
CNT_CREDIT_PROLONG	Πόσες φορές παρατάθηκε το δάνειο;
AMT_CREDIT_SUM	Τρέχον ποσό πίστωσης δανείου.
AMT_CREDIT_SUM_DEBT	Τρέχον χρέος.
AMT_CREDIT_SUM_LIMIT	Το τρέχον πιστωτικό όριο της πιστωτικής κάρτας.
AMT_CREDIT_SUM_OVERDUE	Τρέχον ληξιπρόθεσμο ποσό.
CREDIT_TYPE	Είδος δανείου (αυτοκίνητο, μετρητά κ.ά.).
DAYS_CREDIT_UPDATE	Πόσες ημέρες πριν από την αίτηση δανείου ήρθαν οι τελευταίες πληροφορίες του Γραφείου Πιστώσεων;
AMT_ANNUITY	Ετήσιο ποσό δόσεων προηγούμενου δανείου.

Για κάθε δάνειο στα τρέχοντα σύνολα δεδομένων, υπάρχουν τόσες σειρές όσα και τα δάνεια που είχε ο πελάτης στο Γραφείο Πιστώσεων πριν από την ημερομηνία αίτησης

νέου δανείου. Ένα δάνειο στο δείγμα δεδομένων μπορεί να σχετίζεται με κανένα ή περισσότερα δάνεια στο Γραφείο Πίστωσης.

Οι άνωθεν πληροφορίες για τα προηγούμενα δάνεια, δίνονται από το Γραφείο Πιστώσεων.

6.3.3. Αρχείο bureau_balance.csv

Στο αρχείο bureau_balance.csv δίνονται πληροφορίες για μηνιαία υπόλοιπα προηγούμενων δανείων που αναφέρθηκαν στο Γραφείο Πιστώσεων. Για κάθε δάνειο στα τρέχοντα σύνολα δεδομένων, υπάρχουν τόσες σειρές όσοι και οι μήνες για τους οποίους υπάρχει ιστορικό.

Πίνακας 10. Ιστορικό δόσεων προηγούμενων δανείων

Μεταβλητή	Περιγραφή
SK_BUREAU_ID	Μοναδικός αριθμός προηγούμενου δανείου.
MONTHS_BALANCE	Μήνας υπολοίπου σε σχέση με την ημερομηνία αίτησης (-1 δηλώνει την πιο πρόσφατη ημερομηνία υπολοίπου).
STATUS	Κατάσταση δανείου από το Γραφείο Πιστώσεων κατά τη διάρκεια του μήνα (ενεργό, κλειστό κ.ά.) C: Κλειστό, X: Άγνωστη κατάσταση, 0: Δεν υπάρχουν ημέρες καθυστέρησης, 1: Υπάρχουν 1 έως 30 ημέρες καθυστέρησης, 2: Υπάρχουν 31 έως 60 ημέρες καθυστέρησης, 3: Υπάρχουν 61 έως 90 ημέρες καθυστέρησης, 4: Υπάρχουν 91 έως 120 ημέρες καθυστέρησης, 5: Υπάρχουν πάνω από 120 ημέρες καθυστέρησης ή πουλήθηκε ή διαγράφτηκε.

6.3.4. Αρχείο POS_CASH_balance.csv

Πίνακας 11. Μηνιαίο ιστορικό προηγούμενων καταναλωτικών δανείων στη Home Credit

Μεταβλητή	Περιγραφή
SK_ID_CURR	Μοναδικός αριθμός του κάθε δανείου.
SK_ID_PREV	Μοναδικός αριθμός προηγούμενου δανείου στη Home Credit.
MONTHS_BALANCE	Μήνας υπολοίπου σε σχέση με την ημερομηνία αίτησης (-1 δηλώνει την πιο πρόσφατη ημερομηνία υπολοίπου).
CNT_INSTALMENT	Διάρκεια προηγούμενου δανείου.
CNT_INSTALMENT_FUTURE	Δόσεις που έμειναν για την αποπληρωμή του προηγούμενου δανείου.
NAME_CONTRACT_STATUS	Κατάσταση συμβολαίου κατά τη διάρκεια του μήνα
SK_DPD	Ημέρες καθυστέρησης κατά τη διάρκεια του μήνα.
SK_DPD_DEF	Ημέρες καθυστέρησης κατά τη διάρκεια του μήνα με ανοχή (αγνοούνται χρέη με χαμηλά ποσά δανείου) της προηγούμενης πίστωσης.

Στο αρχείο POS_CASH_balance.csv, περιλαμβάνονται μηνιαίες πληροφορίες των προηγούμενων καταναλωτικών δανείων που είχε ο πελάτης στην Home Credit.

Για κάθε δάνειο στα τρέχοντα σύνολα δεδομένων, υπάρχουν τόσες σειρές όσοι και οι μήνες για τους οποίους υπάρχει ιστορικό.

6.3.5. Αρχείο credit_card_balance.csv

Μηνιαία στιγμιότυπα προηγούμενων δανείων πιστωτικών καρτών που έχει ο πελάτης στη Home Credit, εμφανίζονται στο αρχείο credit_card_balance.csv. Για κάθε προηγούμενο δάνειο στα παρόντα σύνολα δεδομένων, υπάρχουν τόσες σειρές όσοι και οι μήνες για τους οποίους υπάρχει ιστορικό.

Για κάθε προηγούμενο μήνα δίνονται στοιχεία για τα ποσά που κινήθηκαν είτε αυτά αφορούν αγορές είτε πληρωμές δόσεων.

Πίνακας 12. Ποσά κινήσεων δανείων πιστωτικών καρτών ανά μήνα

Μεταβλητή	Περιγραφή
SK_ID_PREV	Μοναδικός αριθμός προηγούμενου δανείου στη Home Credit.
SK_ID_CURR	Μοναδικός αριθμός του κάθε δανείου.
MONTHS_BALANCE	Μήνας υπολοίπου σε σχέση με την ημερομηνία αίτησης (-1 δηλώνει την πιο πρόσφατη ημερομηνία υπολοίπου).
AMT_BALANCE	Υπόλοιπο κατά τη διάρκεια του μήνα.
AMT_CREDIT_LIMIT_ACTUAL	Όριο πιστωτικής κάρτας κατά τη διάρκεια του μήνα.
AMT_DRAWINGS_ATM_CURRENT	Ποσό αναλήψεων σε ATM κατά τη διάρκεια του μήνα.
AMT_DRAWINGS_CURRENT	Ποσό αναλήψεων κατά τη διάρκεια του μήνα.
AMT_DRAWINGS_OTHER_CURRENT	Ποσό λοιπών αναλήψεων κατά τη διάρκεια του μήνα.
AMT_DRAWINGS_POS_CURRENT	Ποσό αναλήψεων ή αγορά αγαθών κατά τη διάρκεια του μήνα.
AMT_PAYMENT_CURRENT	Πόσα πλήρωσε ο πελάτης κατά τη διάρκεια του μήνα για προηγούμενο δάνειο.
AMT_PAYMENT_TOTAL_CURRENT	Πόσα πλήρωσε ο πελάτης κατά τη διάρκεια του μήνα για το σύνολο των προηγούμενων δανείων.
AMT_RECEIVABLE_PRINCIPAL	Απαιτούμενο ποσό για το αρχικό κεφάλαιο.
AMT_RECIVABLE	Απαίτηση για το προηγούμενο δάνειο.
AMT_TOTAL_RECEIVABLE	Συνολικό ποσό απαίτησης για όλα τα προηγούμενα δάνεια.

Εκτός από τα ποσά που κινήθηκαν κατά τους προηγούμενους μήνες, ως πληροφορία αποθηκεύονται και τα σύνολα των κινήσεων. Πόσες φορές δηλαδή έγινε αγορά και πόσες φορές έγινε η καταβολή κάποιου ποσού.

Πίνακας 13. Σύνολα κινήσεων δανείων πιστωτικών καρτών ανά μήνα

Μεταβλητή	Περιγραφή
CNT_DRAWINGS_ATM_CURRENT	Αριθμός αναλήψεων σε ATM κατά τη διάρκεια του μήνα.
CNT_DRAWINGS_CURRENT	Αριθμός αναλήψεων κατά τη διάρκεια του μήνα.
CNT_DRAWINGS_OTHER_CURRENT	Αριθμός λοιπών αναλήψεων κατά τη διάρκεια του μήνα.
CNT_DRAWINGS_POS_CURRENT	Αριθμός αναλήψεων για αγορά αγαθών κατά τη διάρκεια του μήνα.
CNT_INSTALMENT_MATURE_CUM	Αριθμός καταβληθέντων δόσεων.

Πλέον, αποθηκεύονται και πληροφορίες για τις ημέρες καθυστέρησης εξόφλησης των δόσεων των προηγούμενων δανείων πιστωτικών καρτών.

Πίνακας 14. Ημέρες καθυστέρησης εξόφλησης δόσεων δανείων πιστωτικών καρτών ανά μήνα

Μεταβλητή	Περιγραφή
SK_DPD	Ημέρες καθυστέρησης κατά τη διάρκεια του μήνα.
SK_DPD_DEF	Ημέρες καθυστέρησης κατά τη διάρκεια του μήνα με ανοχή (αγνοούνται χρέη με χαμηλά ποσά δανείου).

6.3.6. Αρχείο previous_application.csv

Όπως προαναφέρθηκε, οι πελάτες της Home Credit που έχουν ήδη δάνειο σε αυτή αλλά στο παρελθόν ενδέχεται να είχαν κάνει και άλλες αιτήσεις για δάνειο στην ίδια. Προηγουμένως, δόθηκε περιγραφή για τις προηγούμενες αιτήσεις καταναλωτικών δανείων και δανείων πιστωτικών καρτών.

Το αρχείο previous_application.csv έχει καταχωρημένες πληροφορίες για κάθε προηγούμενη αίτηση όπου κάθε σειρά αφορά μία προηγούμενη αίτηση.

Πίνακας 15. Γενικές μεταβλητές προηγούμενης αίτησης για δάνειο

Μεταβλητή	Περιγραφή
SK_ID_PREV	Μοναδικός αριθμός προηγούμενου δανείου.
SK_ID_CURR	Μοναδικός αριθμός του κάθε δανείου στη Home Credit.
NAME_TYPE_SUITE	Ποιος συνόδευε τον πελάτη κατά την υποβολή αίτησης για το προηγούμενο δάνειο;
WEEKDAY_APPR_PROCESS_START	Ποια ημέρα της εβδομάδας έκανε ο πελάτης την προηγούμενη αίτηση;
HOUR_APPR_PROCESS_START	Περίπου ποια ώρα της ημέρας υπέβαλε ο πελάτης την προηγούμενη αίτηση;

Εκτός από κάποιες πολύ γενικές πληροφορίες, δίνονται και άλλες περισσότερο συναφείς με τις προηγούμενες αιτήσεις που έχουν γίνει για δάνειο. Είτε το δάνειο εκταμιεύτηκε είτε όχι.

Πίνακας 16. Στοιχεία δανείων προηγούμενων αιτήσεων

Μεταβλητή	Περιγραφή
NAME_CONTRACT_TYPE	Τύπος προηγούμενου δανείου (καταναλωτικό, πιστωτικής κάρτας κ.ά.).
AMT_APPLICATION	Πόσο δάνειο ζήτησε ο πελάτης στην προηγούμενη αίτηση;
AMT_CREDIT	Τελικό ποσό δανείου στην προηγούμενη αίτηση.
AMT_GOODS_PRICE	Αξία αγαθών για την οποία ζήτησε ο πελάτης δάνειο στην προηγούμενη αίτηση (εφόσον είναι διαθέσιμη).
NAME_CASH_LOAN_PURPOSE	Σκοπός του καταναλωτικού δανείου.
NAME_CONTRACT_STATUS	Κατάσταση προηγούμενης αίτησης (εγκρίθηκε, ακυρώθηκε κ.ά.).
NAME_PAYMENT_TYPE	Τρόπος πληρωμής που επέλεξε ο πελάτης στην προηγούμενη αίτηση.
CODE_REJECT_REASON	Γιατί απορρίφθηκε η προηγούμενη αίτηση;
NAME_GOODS_CATEGORY	Για τι είδους προϊόντα έκανε την προηγούμενη αίτηση ο πελάτης;
NAME_PORTFOLIO	Το προηγούμενο δάνειο ήταν για μετρητά, αυτοκίνητο κ.λπ.;
NAME_CLIENT_TYPE	Ο πελάτης ήταν παλιός ή νέος όταν έκανε την προηγούμενη αίτηση;
NFLAG_INSURED_ON_APPROVAL	Ζήτησε ο πελάτης ασφάλιση κατά την προηγούμενη αίτηση;
NFLAG_MICRO_CASH	Επισήμανση αν είναι δάνειο μικροχρηματοδότησης.

Η μεταβλητή AMT_CREDIT διαφέρει από το AMT_APPLICATION. Η AMT_APPLICATION δηλώνει το ποσό για το οποίο ο πελάτης έκανε αίτηση. Κατά τη διαδικασία έγκρισης όμως του δανείου, θα μπορούσε να είχε λάβει διαφορετικό ποσό. Το ποσό που πραγματικά πήρε ο πελάτης είναι η μεταβλητή AMT_CREDIT.

6.3.7. Αρχείο installments_payments.csv

Στο αρχείο installments_payments.csv παρέχονται πληροφορίες για το ιστορικό αποπληρωμής προηγούμενων εκταμιευμένων δανείων από τη Home Credit, για πελάτες που περιέχονται στα τρέχοντα σύνολα δεδομένων.

Κάθε σειρά, ισοδυναμεί με μία πληρωμή μίας δόσης είτε τρέχοντος είτε προηγούμενου δανείου στη Home Credit.

Σε αυτό το σύνολο δεδομένων, είναι εμφανές πως για κάθε δόση που πληρώνει ο πελάτης, μπορεί να υπολογιστούν δύο ποσά. Το ποσό που αφορά τα προηγούμενα δάνεια και το ποσό που αφορά το τρέχον δάνειο.

Πίνακας 17. Ποσά δόσεων δανείων

Μεταβλητή	Περιγραφή
SK_ID_PREV	Μοναδικός αριθμός προηγούμενου δανείου.
SK_ID_CURR	Μοναδικός αριθμός του κάθε δανείου στη Home Credit.
DAYS_INSTALMENT	Πότε έπρεπε να πληρωθεί η δόση του προηγούμενου δανείου (σε σχέση με την ημερομηνία αίτησης του τρέχοντος δανείου).
DAYS_ENTRY_PAYMENT	Πότε καταβλήθηκαν πραγματικά οι δόσεις του προηγούμενου δανείου (σε σχέση με την ημερομηνία αίτησης του τρέχοντος δανείου).
AMT_INSTALMENT	Ποιο ήταν το προβλεπόμενο ποσό δόσης του προηγούμενου δανείου σε αυτήν τη δόση.
AMT_PAYMENT	Τι πραγματικά πλήρωσε ο πελάτης για το προηγούμενο δάνειο σε αυτήν τη δόση.

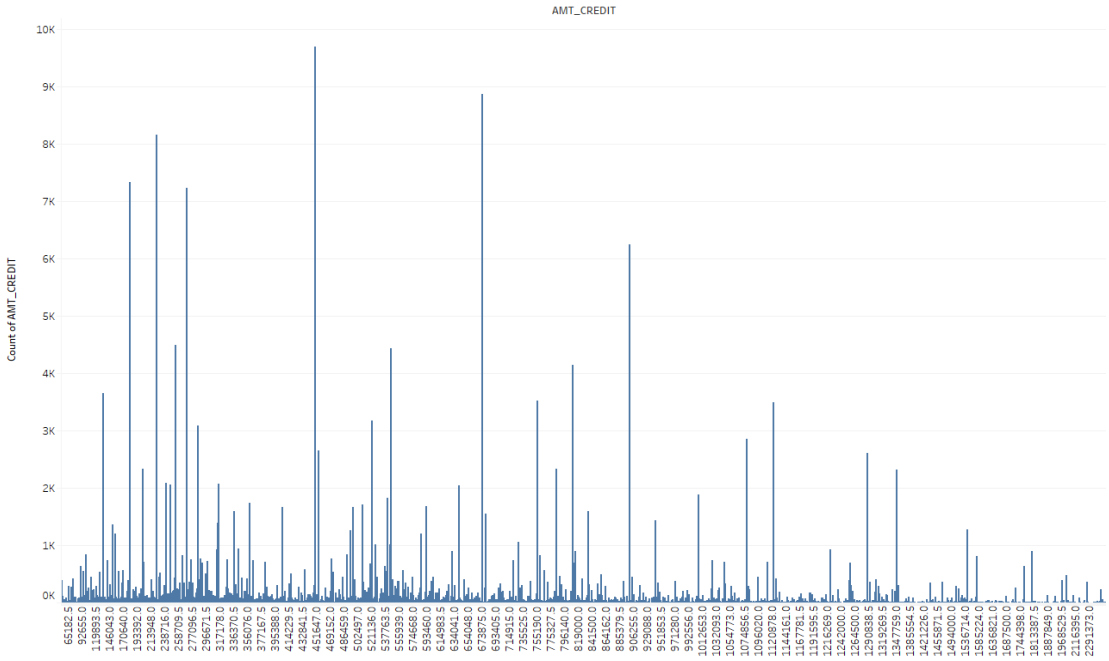
7. Διερευνητική ανάλυση δεδομένων

Στο κεφάλαιο αυτό γίνεται η διερευνητική ανάλυση των δεδομένων. Βασικό σκοπό έχει την εξερεύνηση των βασικών μεταβλητών μέσα από οπτικοποιήσεις, βοηθώντας στην καλύτερη κατανόηση των μεταβλητών του συνόλου δεδομένων αλλά και των σχέσεων μεταξύ τους.

Η διαδικασία αυτή γίνεται με τη χρήση της εφαρμογής Tableau, όπου αρχικά φορτώνονται όλα τα δεδομένα και στη συνέχεια, μέσα από ερωτήματα δημιουργούνται οι ανάλογες οπτικοποιήσεις.

Κάθε υπο-ενότητα του παρόντος κεφαλαίου, απαντάει και σε ένα ερώτημα και η απάντηση εμφανίζεται με τη μορφή γραφήματος.

7.1. Κατανομή ποσών των δανείων

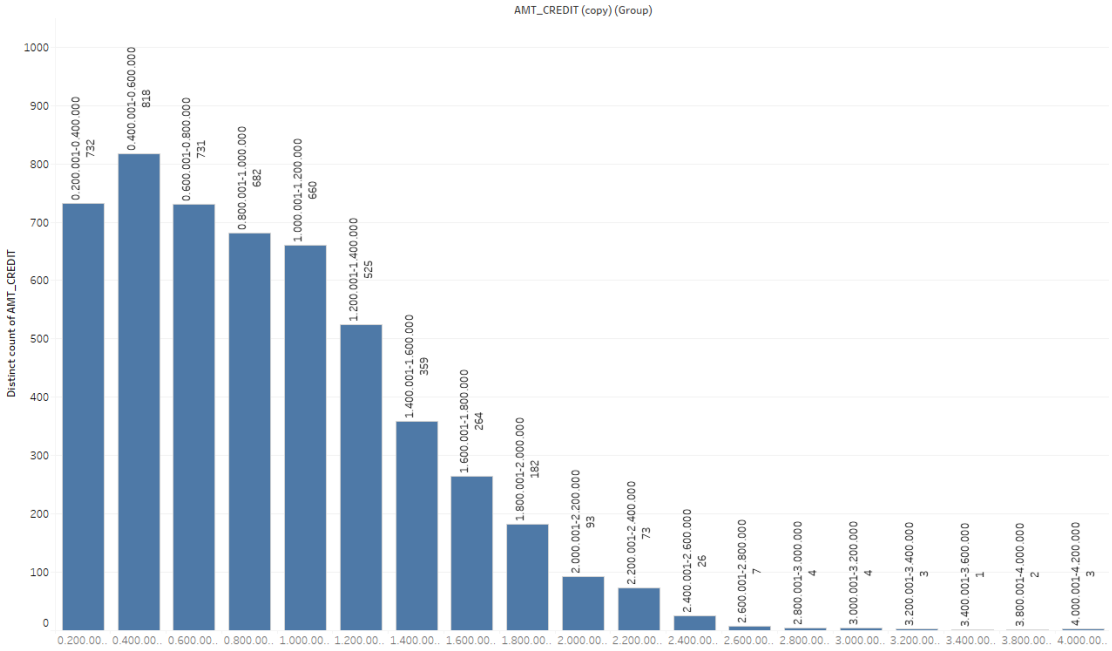


Εικόνα 13. Κατανομή ποσών των δανείων

Στην πιο πάνω εικόνα εμφανίζεται η κατανομή των ποσών των δανείων που έχουν δοθεί.

Εξαιτίας του μεγάλου πλήθους των δανείων, η γραφική παράσταση της κατανομής δεν μπορεί να δώσει κάποιες οπτικές ενδείξεις. Για να επιτευχθεί μία καλύτερη εμφάνιση της κατανομής των ποσών, τα ποσά είναι δυνατό να ομαδοποιηθούν.

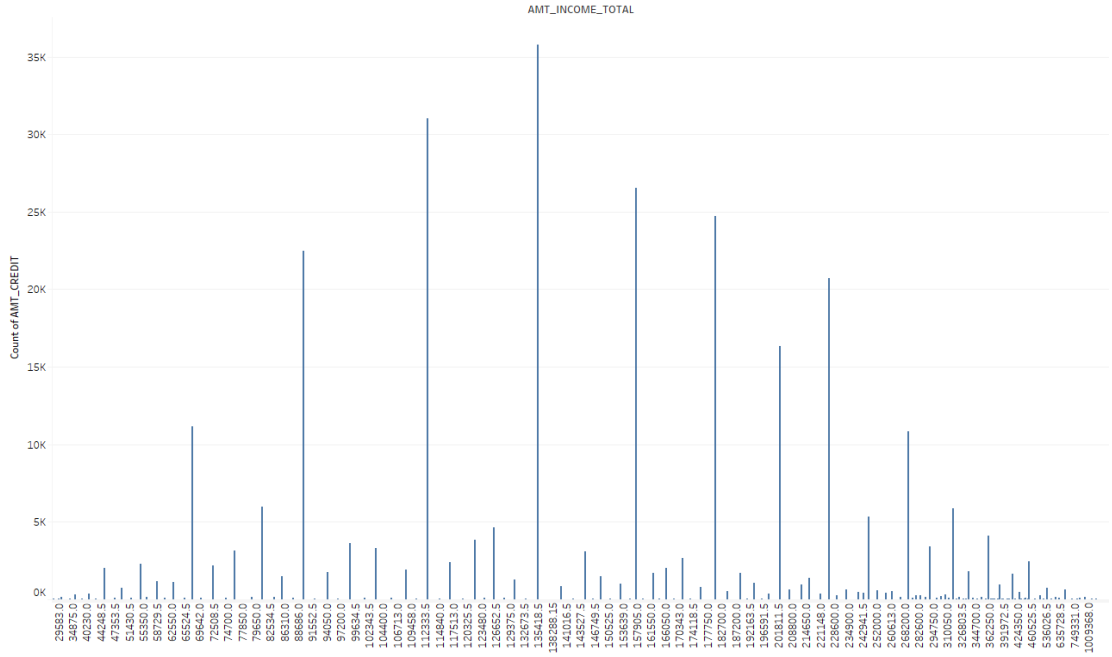
Έπειτα από ομαδοποίηση των ποσών που έχουν δοθεί σε δάνεια, προκύπτει η ακόλουθη γραφική του παράσταση. Η ομαδοποίηση έχει γίνει ανά 200.000€. με τον τρόπο αυτόν, είναι περισσότερο εμφανής η κατανομή των ποσών, όπως εμφανίζει η εικόνα 14.



Εικόνα 14. Ομαδοποιημένη κατανομή ποσών των δανείων

Το ελάχιστο ποσό δανείου που έχει δοθεί είναι 45.000€ και το μέγιστο είναι 4.050.000€. Είναι εμφανές ότι όσο το ποσό αυξάνεται, τόσο λιγότεροι είναι οι δανειολήπτες.

7.2. Κατανομή εισοδήματος πελατών

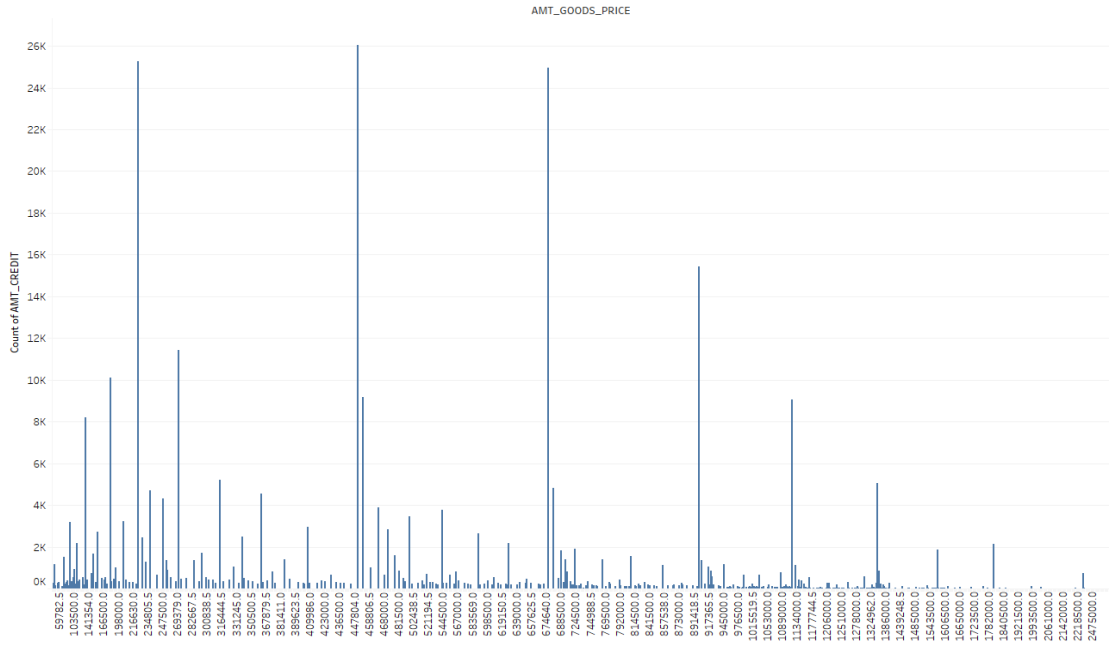


Εικόνα 15. Κατανομή εισοδήματος πελατών

Αντίστοιχα, εμφανίζεται η κατανομή εισοδήματος των δανειοληπτών. Ελάχιστο εισόδημα είναι το ποσό των 25.650€ και αντιστοιχεί σε 2 άτομα. Μέγιστο εισόδημα είναι τα 117.000.000€ και αφορά έναν πελάτη.

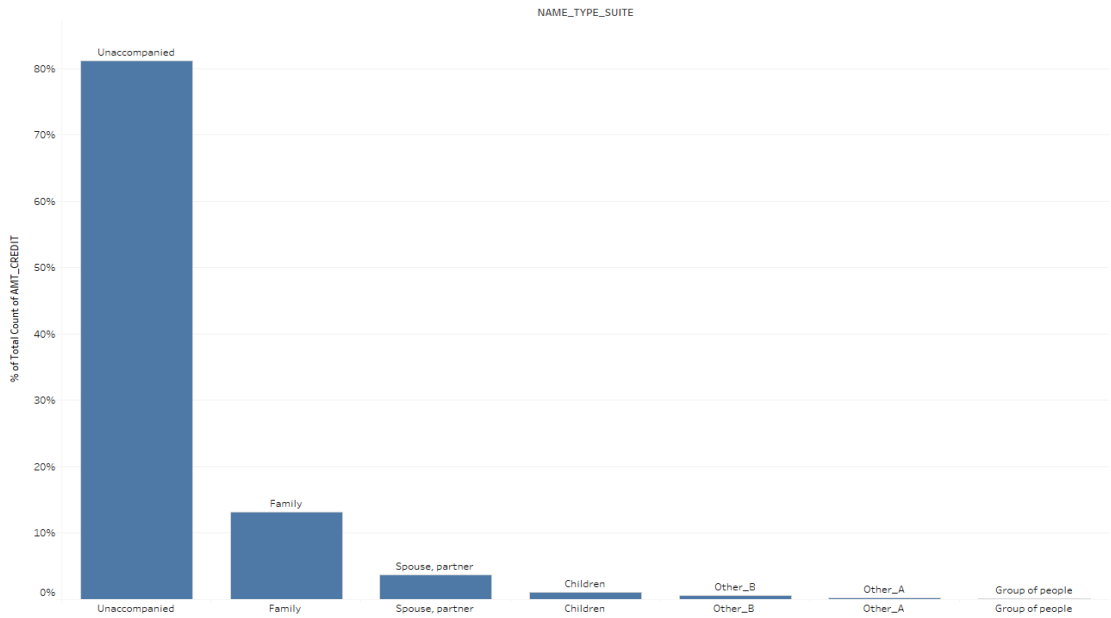
7.3. Κόστος αγαθών για τα οποία δόθηκαν δάνεια

Τα δάνεια δόθηκαν για να εξυπηρετήσουν στην αγορά κάποιων αγαθών. Η κατανομή του κόστους των αγαθών αυτών εμφανίζεται στην επόμενη εικόνα. Το ελάχιστο κόστος είναι τα 40.500€ και αφορά δάνειο ενός πελάτη. Μέγιστο κόστος είναι τα 4.050.000€ και αφορά δάνεια που έχουν λάβει οκτώ πελάτες.



Εικόνα 16. Κόστος αγαθών για τα οποία δόθηκαν δάνεια

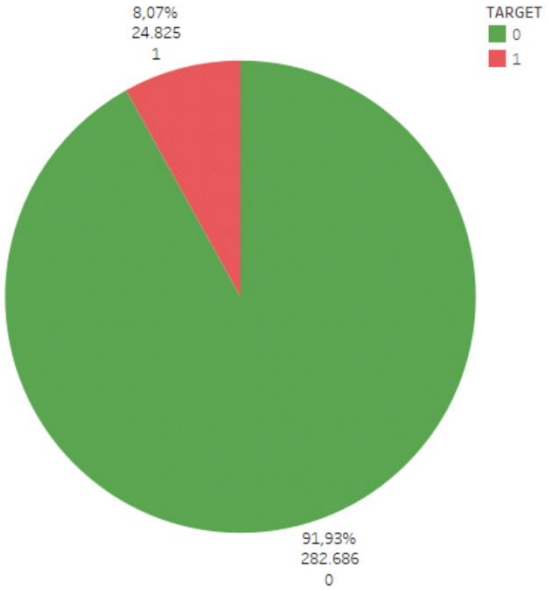
7.4. Συνοδοί πελατών κατά την αίτηση δανείου



Εικόνα 17. Συνοδοί πελατών κατά την αίτηση δανείου

Κατά τις αιτήσεις για δάνειο, το 81,16% των πελατών ήταν ασυνόδευτοι, ενώ το 13,11% συνοδευόταν από κάποιο μέλος της οικογένειάς του.

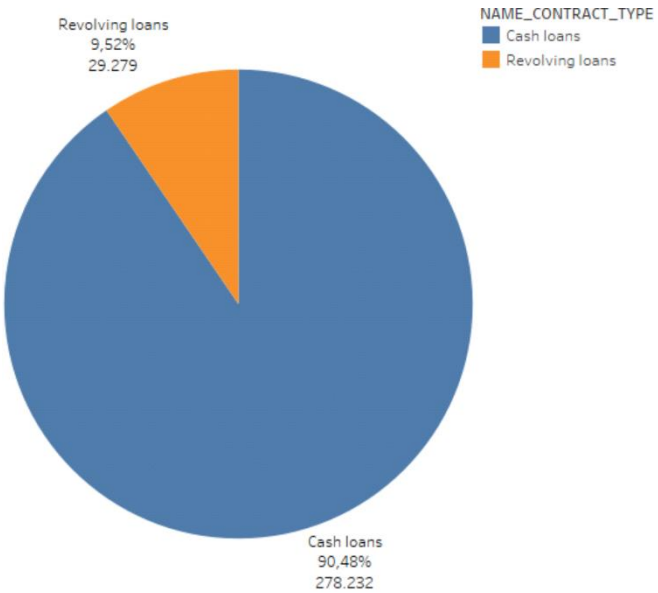
7.5. Ισορροπία της μεταβλητής απόκρισης



Εικόνα 18. Ισορροπία της μεταβλητής απόκρισης

Όπως διακρίνεται στην παραπάνω εικόνα, δεν υπάρχει ισορροπία στα δεδομένα της μεταβλητής απόκρισης. Είναι ξεκάθαρο ότι το μικρότερο ποσοστό (8,07%) είχε κάποια δυσκολία πληρωμής, ενώ το υπόλοιπο 91,93%, όχι.

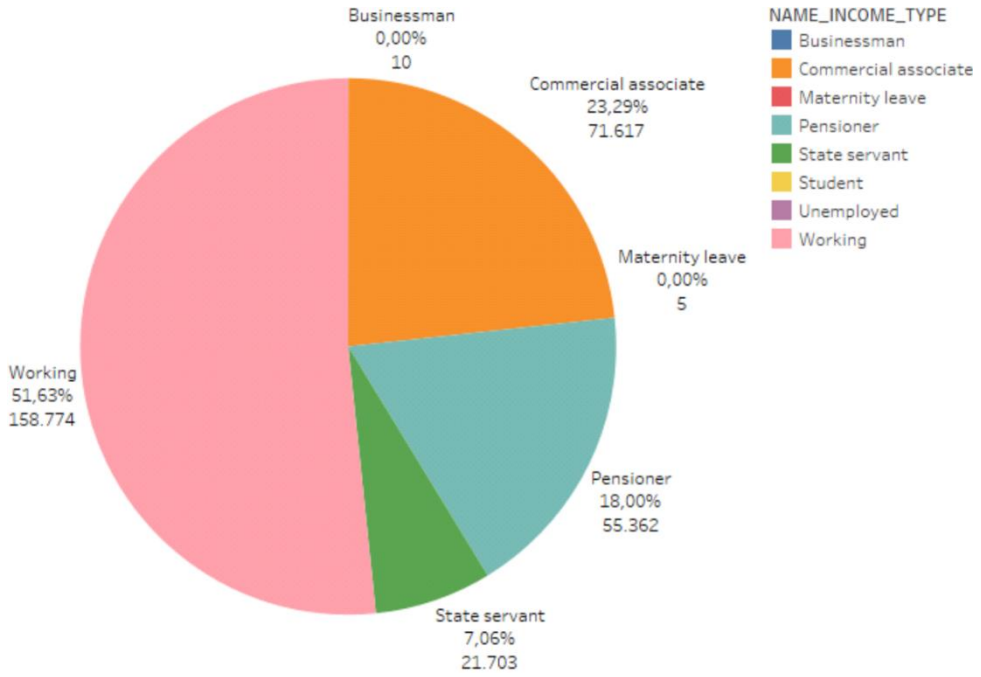
7.6. Τύποι δανείων



Εικόνα 19. Τύποι δανείων

Τα δάνεια, κατά το μεγαλύτερο ποσοστό (90,48%) είναι δάνεια με εμτρητά, ενώ το ποσοστό των ανακυκλούμενων δανείων είναι 9,62%.

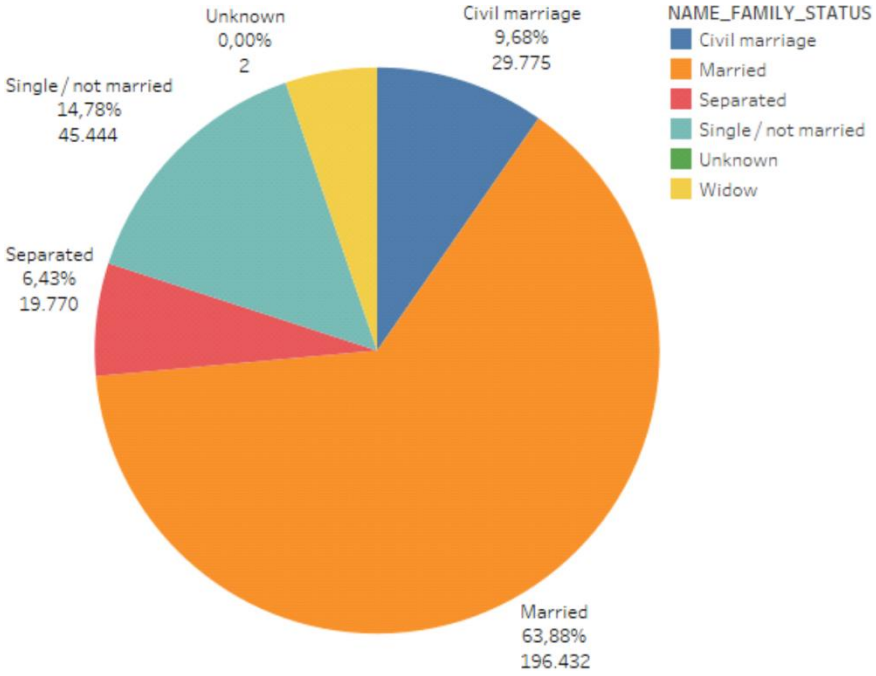
7.7. Προέλευση εισοδήματος αιτούντων



Εικόνα 20. Προέλευση εισοδήματος αιτούντων

Οι μισοί σχεδόν από τους αιτούντες (51,63%) δηλώνουν εργαζόμενοι και ελάχιστοι (μόνο 10 στο σύνολο) έχουν δηλώσει επιχειρηματίες.

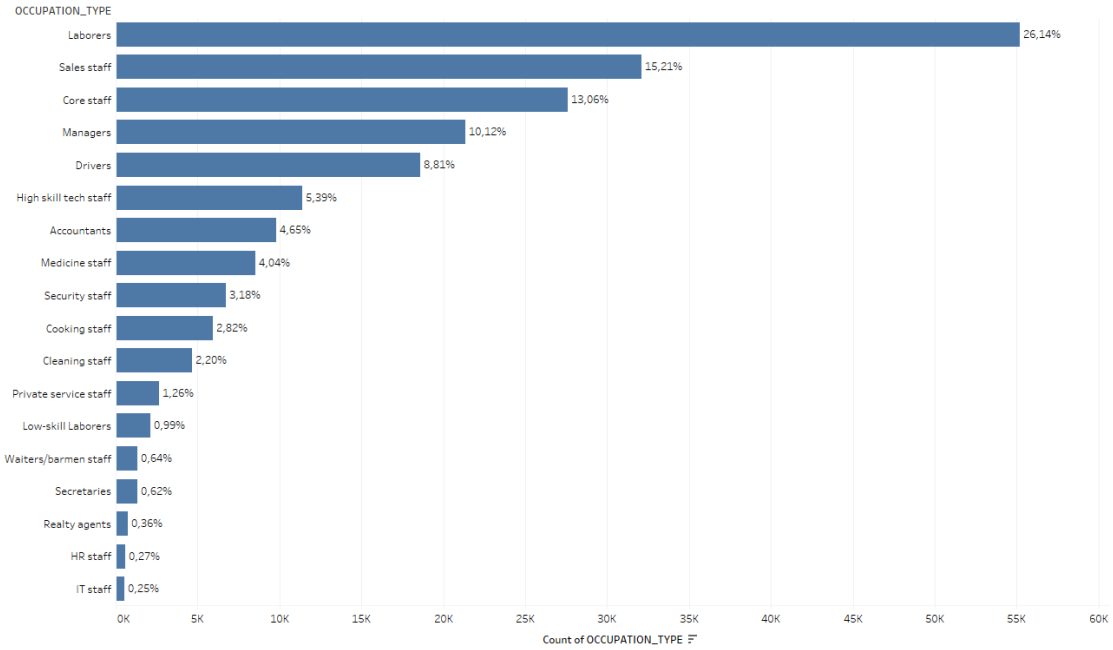
7.8. Οικογενειακή κατάσταση αιτούντων



Εικόνα 21. Οικογενειακή κατάσταση αιτούντων

Η πλειοψηφία των αιτούντων είναι συζευγμένοι, με ποσοστό 63,88% επί του συνόλου των αιτούντων.

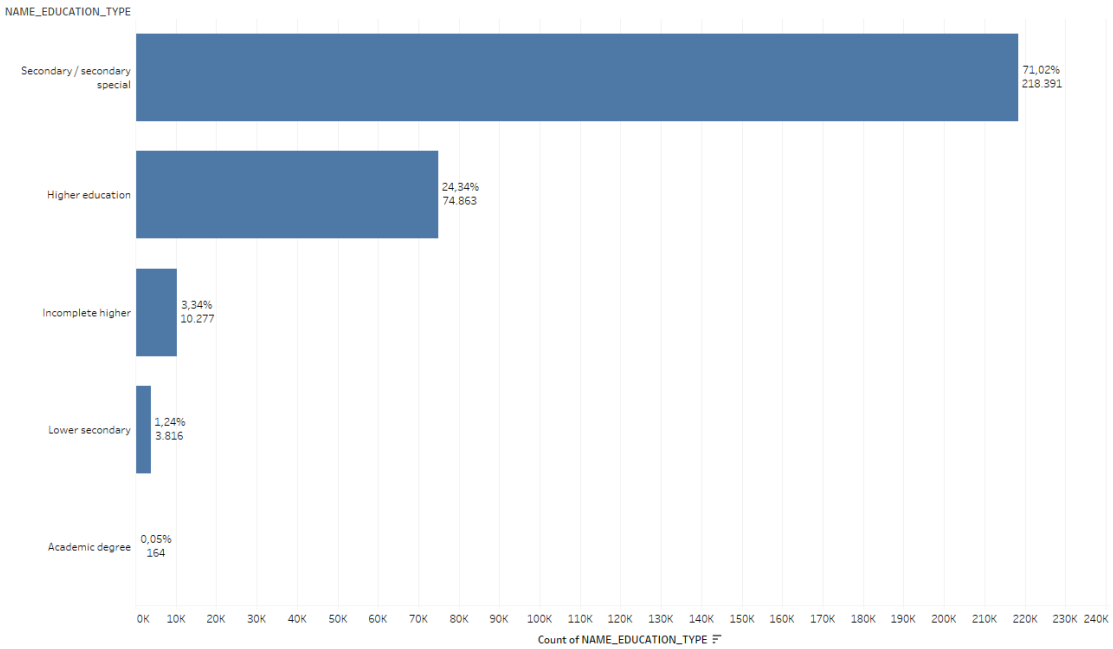
7.9. Επαγγελματική απασχόληση αιτούντων



Εικόνα 22. Επαγγελματική απασχόληση αιτούντων

Πιο πολλοί είναι οι εργάτες που έχουν κάνει αίτηση για δάνειο, με ποσοστό 26,14%. Λιγότεροι είναι αυτοί που έχουν ως αντικείμενο τις πληροφορικές τεχνολογίες.

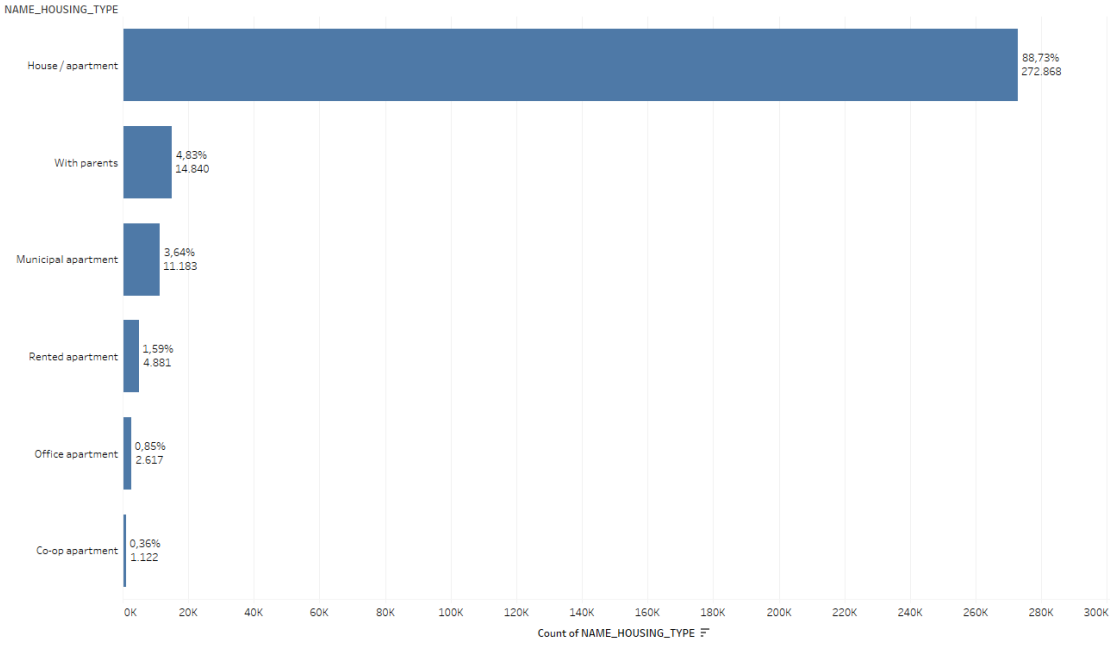
7.10. Εκπαίδευση αιτούντων



Εικόνα 23. Εκπαίδευση αιτούντων

Οι απόφοιτοι δευτεροβάθμιας εκπαίδευσης είναι αυτοί που έχουν κάνει περισσότερες αιτήσεις για δάνειο, με ποσοστό που ξεπερνάει το 71%.

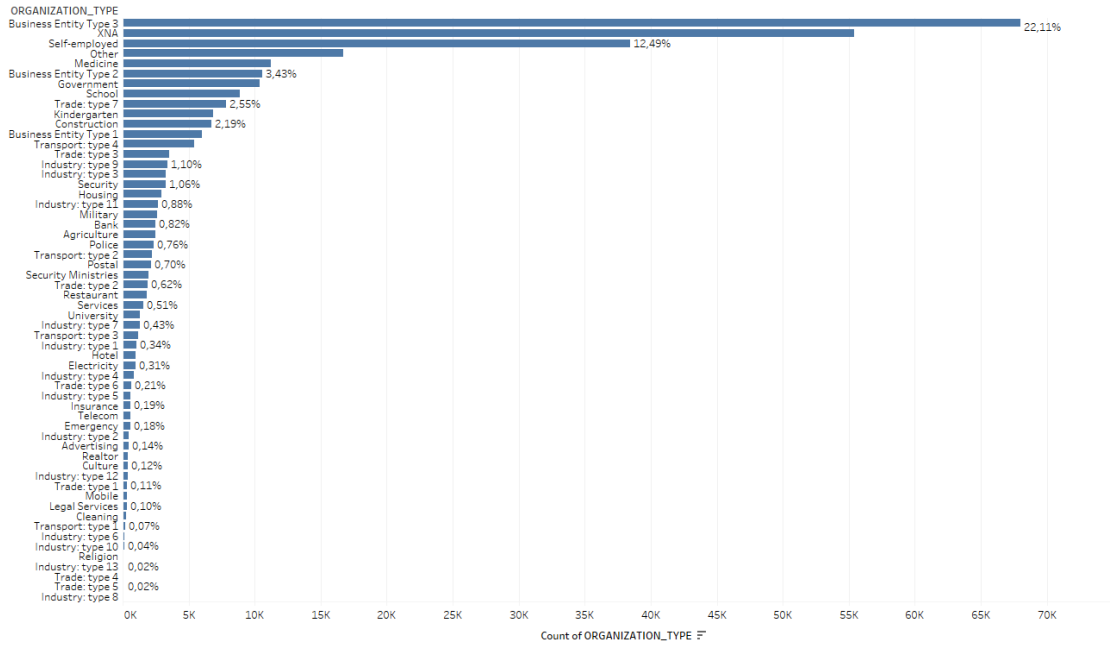
7.11. Στεγαστική κατάσταση αιτούντων



Εικόνα 24. Στεγαστική κατάσταση αιτούντων

Από όσους έχουν αιτηθεί να λάβουν κάποιο δάνειο, οι περισσότεροι διαμένουν σε ιδιόκτητο σπίτι ή διαμέρισμα. Το ποσοστό τους αγγίζει το 89%.

7.12. Είδη επιχειρήσεων όπου εργάζονται οι αιτούντες



Εικόνα 25. Είδη επιχειρήσεων όπου εργάζονται οι αιτούντες

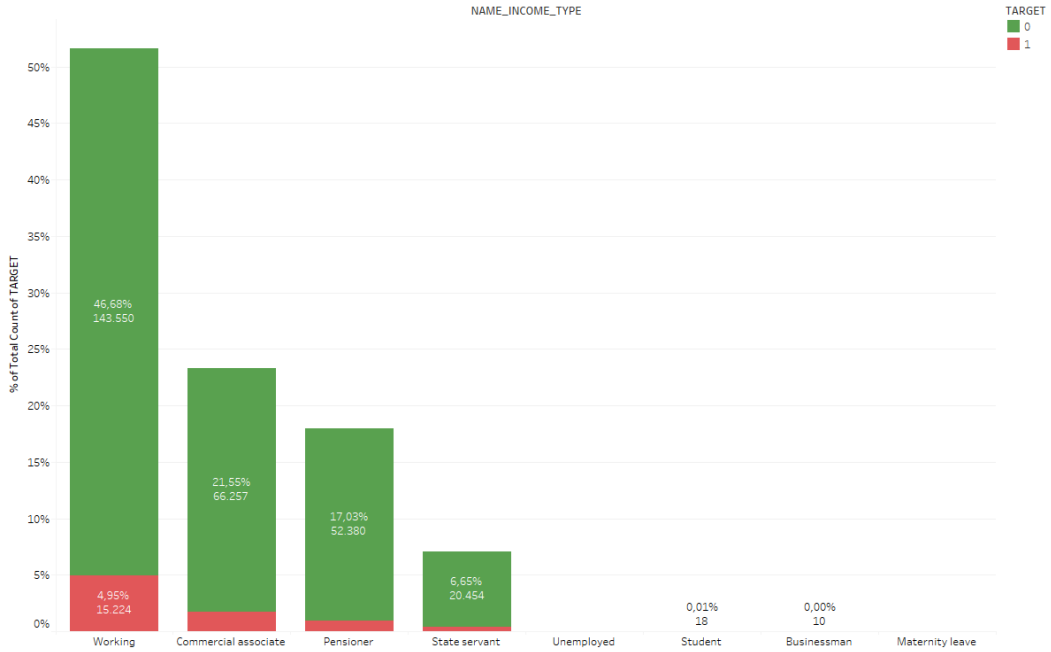
Με την κωδικοποίηση της Home Credit, δεν είναι απόλυτα ξεκάθαρο από ποια είδη επιχειρήσεων προέρχονται οι αιτούντες. Σχεδόν το ένα πέμπτο προέρχεται από επιχειρήσεις του είδους «Business Entity Type 3».

7.13. Προβληματικά δάνεια ανά κατηγορία

Σημαντικό είναι, για τις μεταβλητές που παρουσιάστηκαν προηγουμένως μέσα από τα διάφορα γραφήματα, να εμφανιστούν επίσης τα ποσοστά των δανείων στα οποία υπήρξε κάποιο πρόβλημα στην αποπληρωμή τους ή όχι.

Κατ' αυτόν τον τρόπο, στις οπτικοποιήσεις που θα ακολουθήσουν, τα ποσοστά είναι διηρημένα σε υπο-ποσοστά.

Με κόκκινο χρώμα εμφανίζονται τα δάνεια στα οποία υπήρξε πρόβλημα στην αποπληρωμή τους. Με πράσινο χρώμα, απεικονίζονται όλες οι άλλες περιπτώσεις.

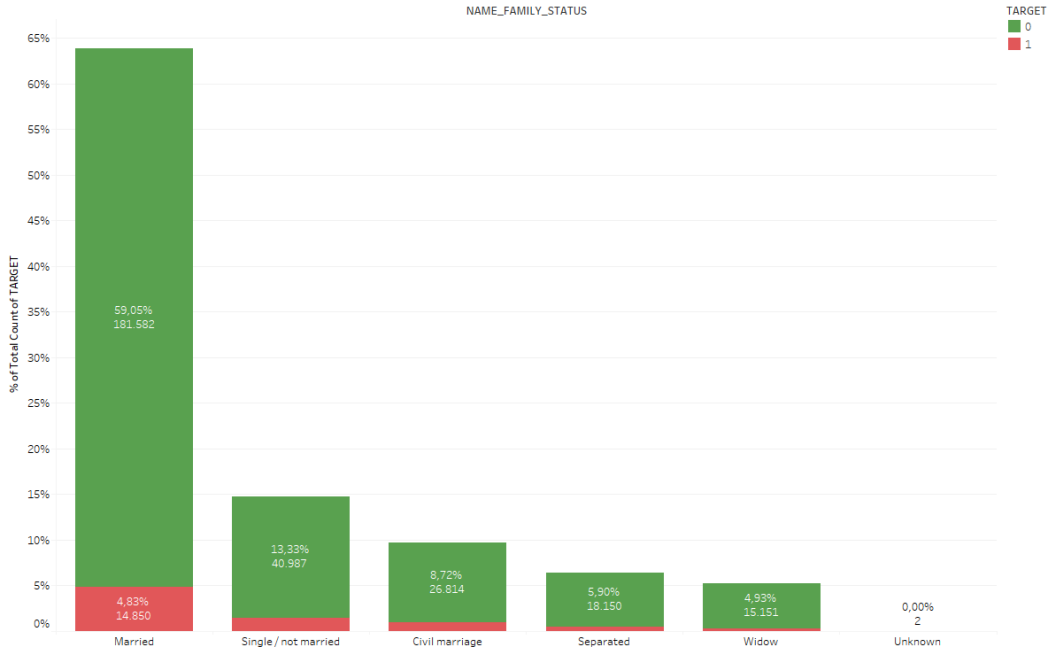


Εικόνα 26. Ποσοστά προβληματικών δανείων ανά προέλευση εισοδήματος αιτούντων

Στο παραπάνω σχήμα φαίνεται πως για τους εργαζόμενους υπάλληλους δημιουργείται σε ποσοστό 9,59% κάποιο πρόβλημα στην αποπληρωμή του δανείου, για εταίρους επιχειρήσεων σε ποσοστό 7,48%, για τους δημόσιους υπάλληλους σε ποσοστό 5,75% ενώ για τους συνταξιούχους σε ποσοστό μόνο 5,39%.

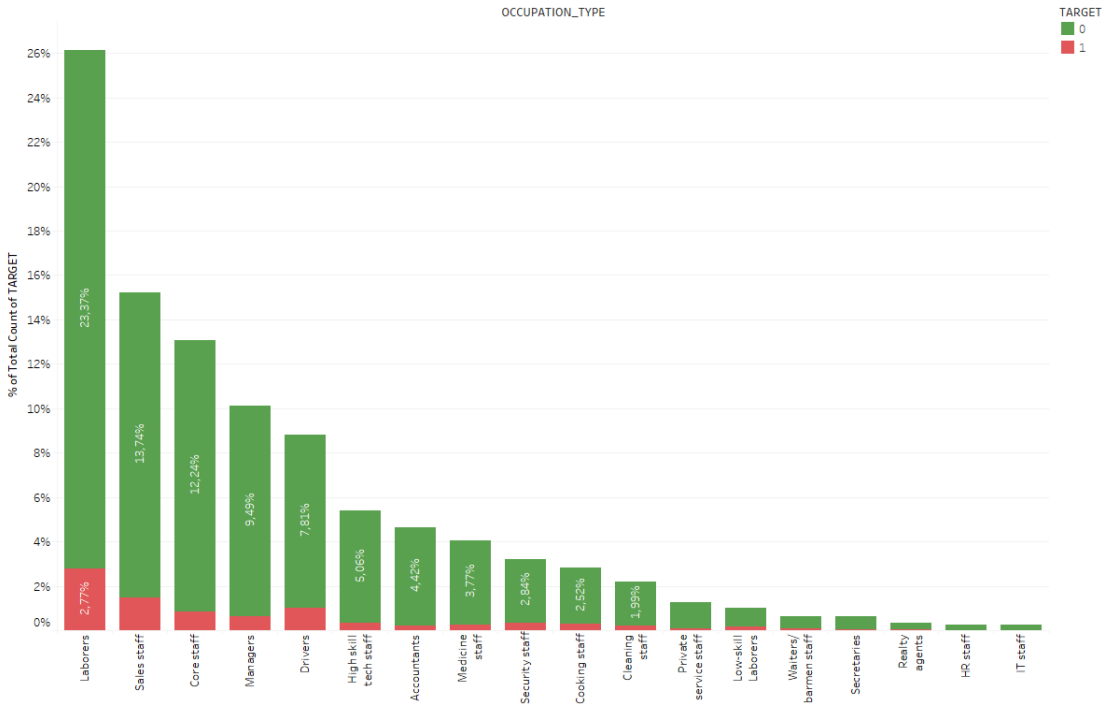
Μπορεί λοιπόν να βγει το συμπέρασμα πως οι συνταξιούχοι είναι οι πιο αξιόπιστοι δανειολήπτες σε αντίθεση με τους υπάλληλους που αποκαλύπτεται πως είναι περισσότερο αναξιόπιστοι.

Αν ληφθεί μία όψη των προβληματικών δανείων ανά οικογενειακή κατάσταση αιτούντων, φαίνεται πως πρόβλημα δημιουργείται στο 9,94% όσων έχουν τελέσει πολιτικό γάμο, στο 9,81% των μη συζευγμένων, στο 8,19% των διαζευγμένων ενώ οι παντρεμένοι παρουσιάζουν προβλήματα αποπληρωμής σε ποσοστό 7,56%.



Εικόνα 27. Ποσοστά προβληματικών δανείων ανά οικογενειακή κατάσταση αιτούντων

Πιο συνεπείς στις δανειακές υποχρεώσεις τους, φαίνεται να είναι όσοι βρίσκονται σε κατάσταση χηρείας όπου παρουσιάζουν προβλήματα στην αποπληρωμή σε ποσοστό μόνο 5,82%.

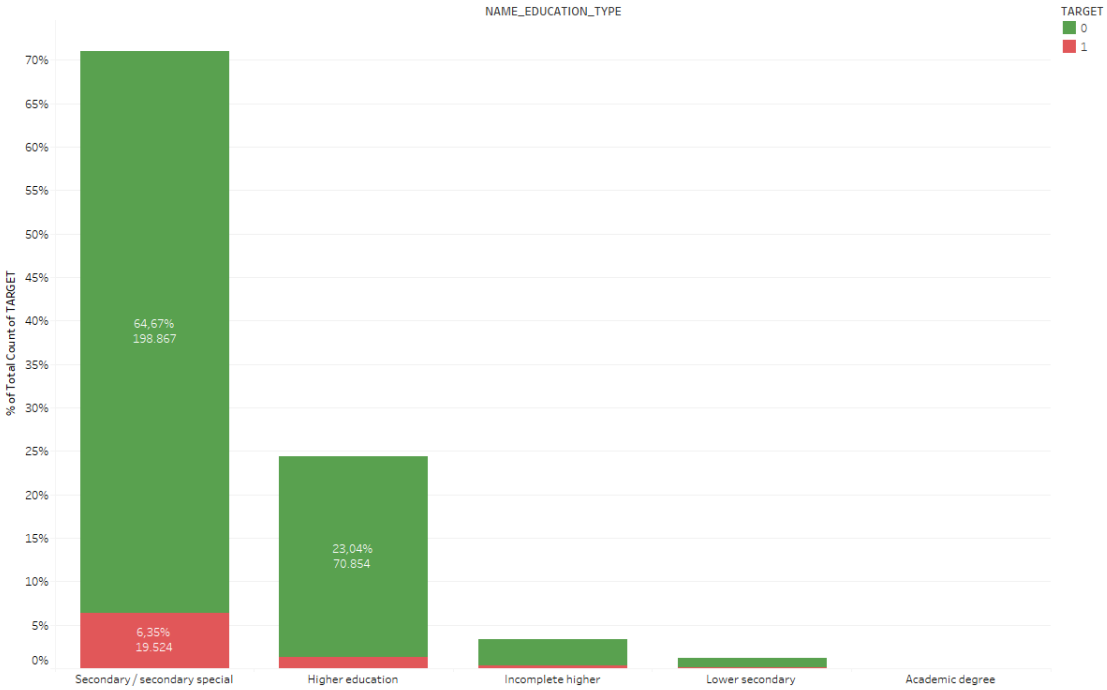


Εικόνα 28. Ποσοστά προβληματικών δανείων ανά επαγγελματική απασχόληση αιτούντων

Μελετώντας τα δεδομένα στο σχήμα της εικόνας 28, προκύπτει πως περισσότερο συνεπείς στην αποπληρωμή των δανείων είναι οι λογιστές οι οποίοι σε ποσοστό 95,17% αποπληρώνουν χωρίς προβλήματα το δάνειό τους. Το μεγαλύτερο πρόβλημα

δημιουργείται με τους εργάτες χαμηλής εξειδίκευσης, από τους οποίους μόνο το ποσοστό του 82,85% καταφέρνει να είναι εντάξει στην αποπληρωμή.

Με τα δεδομένα που απεικονίζονται στο σχήμα της εικόνας 29, προκύπτει ένα πολύ σημαντικό συμπέρασμα. Όσο μεγαλύτερη είναι η βαθμίδα της εκπαίδευσης, τόσο λιγότερα είναι τα προβληματικά δάνεια.



Εικόνα 29. Ποσοστά προβληματικών δανείων ανά τύπο εκπαίδευσης αιτούντων

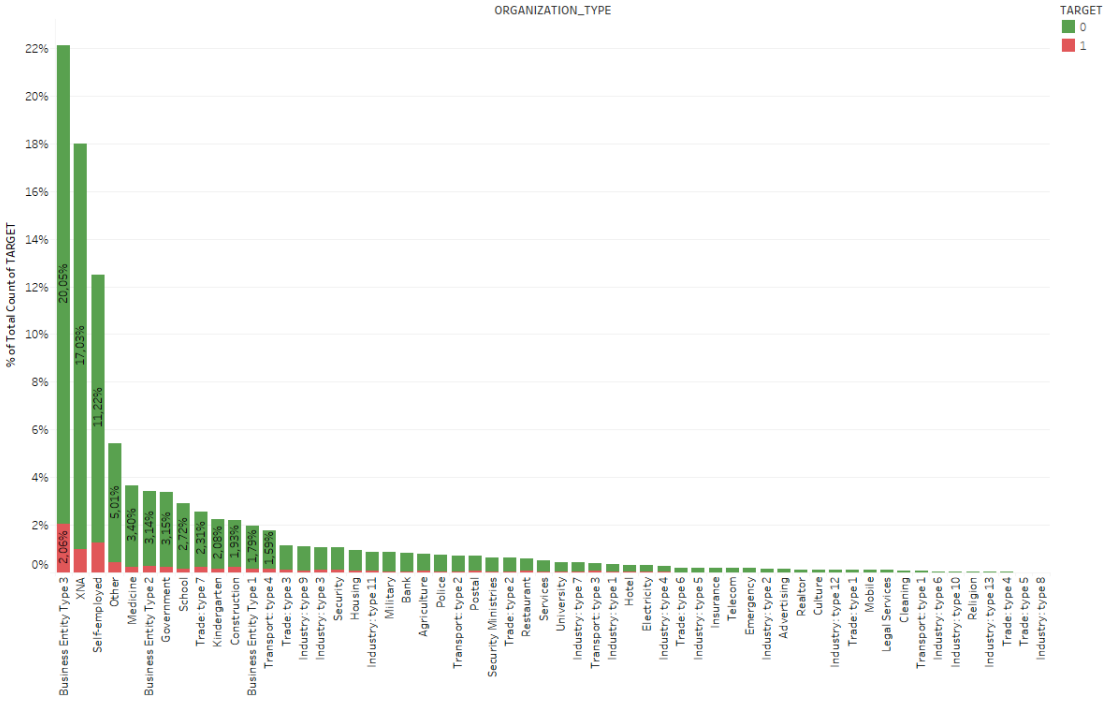
Τα ποσοστά των προβληματικών δανείων είναι 8,94% για απόφοιτους δευτεροβάθμιας εκπαίδευσης, 5,36% για απόφοιτους τριτοβάθμιας εκπαίδευσης, 8,48% για όσους δεν ολοκλήρωσαν την τριτοβάθμια εκπαίδευση, 10,93% για αυτούς που έχουν κατώτερη δευτεροβάθμια εκπαίδευση ενώ μόνο 1,83% για κατόχους πτυχίου ακαδημαϊκής εκπαίδευσης.

Η στεγαστική κατάσταση των αιτούντων, παίζει επίσης το ρόλο του για ένα δάνειο. Όσοι κατοικούν σε διαμέρισμα γραφείου σχετίζονται μόνο σε ποσοστό 5,57% με προβληματικά δάνεια.

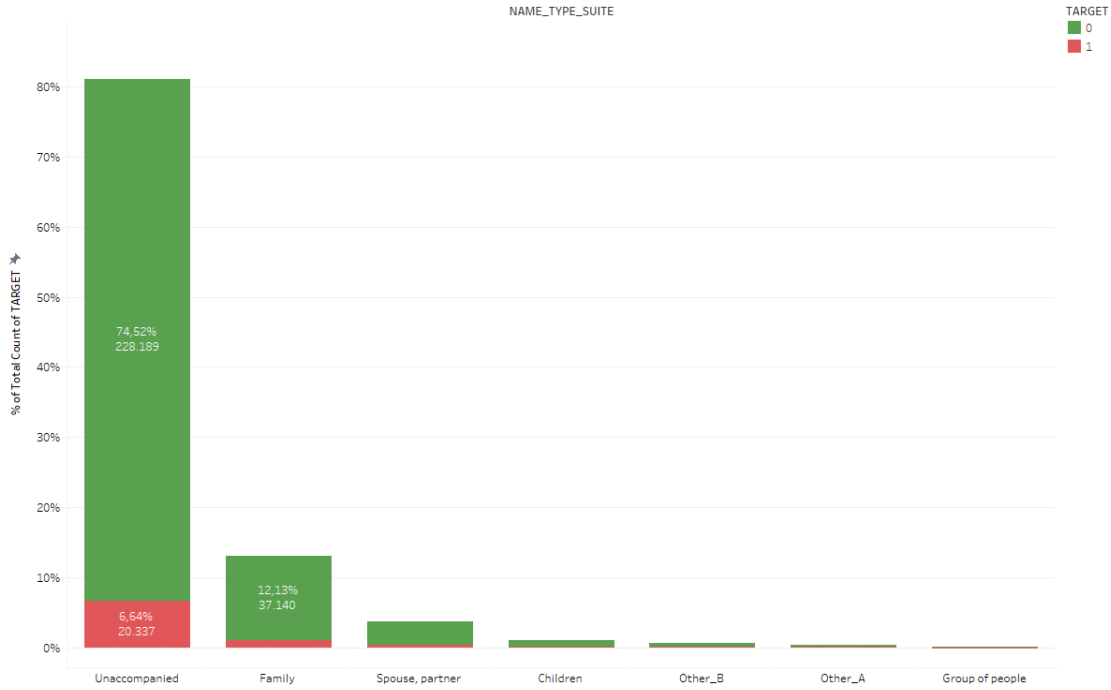
Τα προβληματικά δάνεια είναι 7,80% όσων κατοικούν σε ιδιόκτητο σπίτι ή διαμέρισμα ενώ και 7,93% όσων συγκατοικούν. Από αυτούς που διαμένουν σε δημοτικό διαμέρισμα, το 8,54% έχει προβληματικό δάνειο. Τα υψηλότερα ποσοστά δανείων με πρόβλημα τα έχουν όσοι κατοικούν με τους γονείς τους σε ποσοστό 11,70%, με πιο ασυνεπείς όσους μένουν σε ενοικιασμένο διαμέρισμα όπου το 12,31% έχει να κάνει με προβληματικό δάνειο.

Εικόνα 30. Ποσοστά προβληματικών δανείων ανά στεγαστική κατάσταση αιτούντων

Τα ποσοστά των προβληματικών δανείων σε σχέση με το είδος επιχείρησης όπου εργάζονται οι αιτούντες, δε διαφοροποιούνται πολύ. Αυτό όμως που αξίζει να παρατηρηθεί, είναι πως οι εργαζόμενοι σε επιχειρήσεις μεταφορών δημιουργούν προβλήματα με τα δάνεια σε μεγάλο ποσοστό που αγγίζει το 15,75%. Βέβαια, δεν είναι ακριβές το είδος της επιχείρησης που εργάζονται εφόσον καταχωρείται στα δεδομένα ως «Transport: type 3». Παρ' όλα αυτά, έχει τη σημασία του.



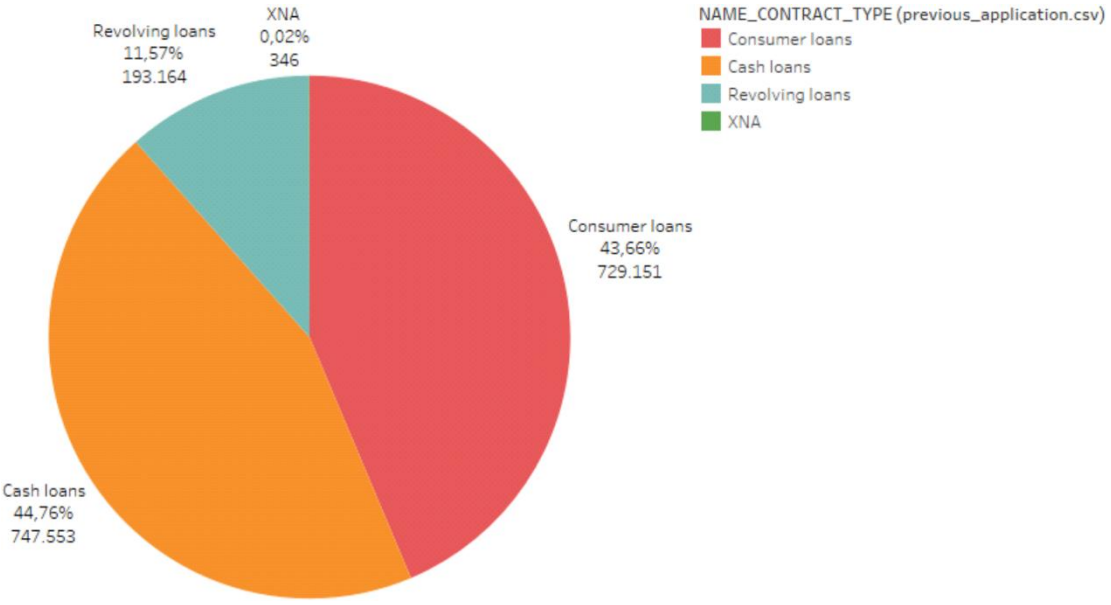
Εικόνα 31. Ποσοστά προβληματικών δανείων ανά είδος επιχείρησης όπου εργάζονται οι αιτούντες



Εικόνα 32. Ποσοστά προβληματικών δανείων ανά τύπο συνοδείας που είχαν κατά την αίτηση για δάνειο

Η συνοδεία των αιτούμενων για δάνειο, δε δείχνει να παίζει τόσο σημαντικό ρόλο στο αν ένα δάνειο είναι προβληματικό ή όχι. Τα ποσοστά κυμαίνονται στα ίδια επίπεδα. Λίγο μεγαλύτερο ποσοστό σε προβληματικά δάνεια, έχουν όσοι συνοδεύτηκαν από κάποιο πρόσωπο που καταχωρείται ως «Άλλο Α» και ως «Άλλο Β» αλλά αυτές οι καταχωρίσεις αφορούν ένα ελάχιστο αριθμό αιτούντων σε σχέση με το σύνολό τους.

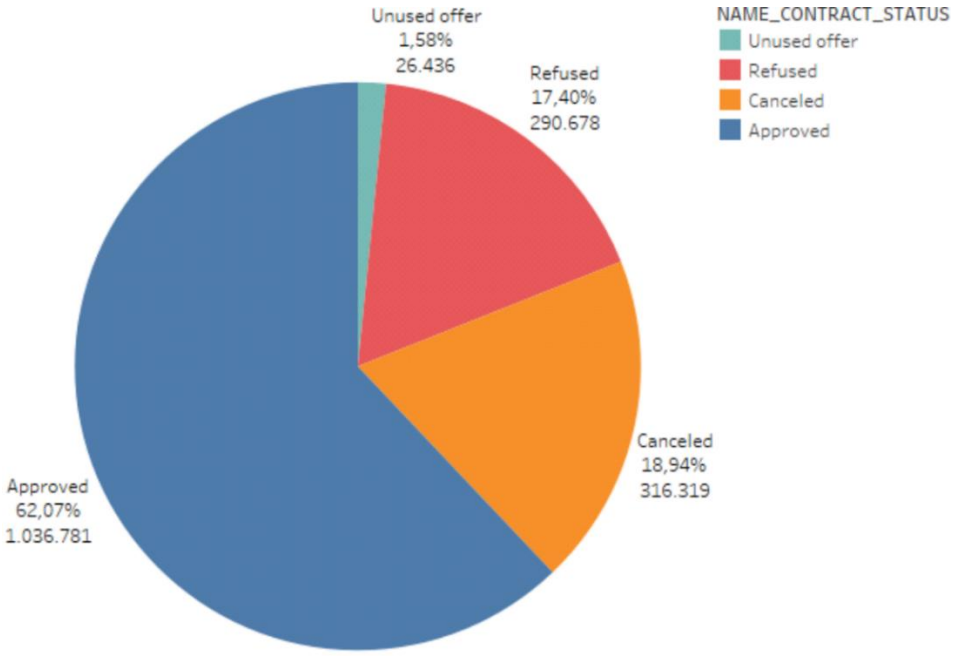
7.14. Τύποι προηγούμενων αιτήσεων δανείων



Εικόνα 33. Τύποι προηγούμενων αιτήσεων

Στο σύνολο των προηγούμενων αιτήσεων για δάνειο, είναι εμφανές ότι ένα μικρό ποσοστό αφορούσε ανακυκλούμενα δάνεια, σε ποσοστό 11,57%. Η πλειονότητά τους, αφορούσε δάνεια μετρητών ή καταναλωτικά δάνεια, με συνολικό ποσοστό 88,42%.

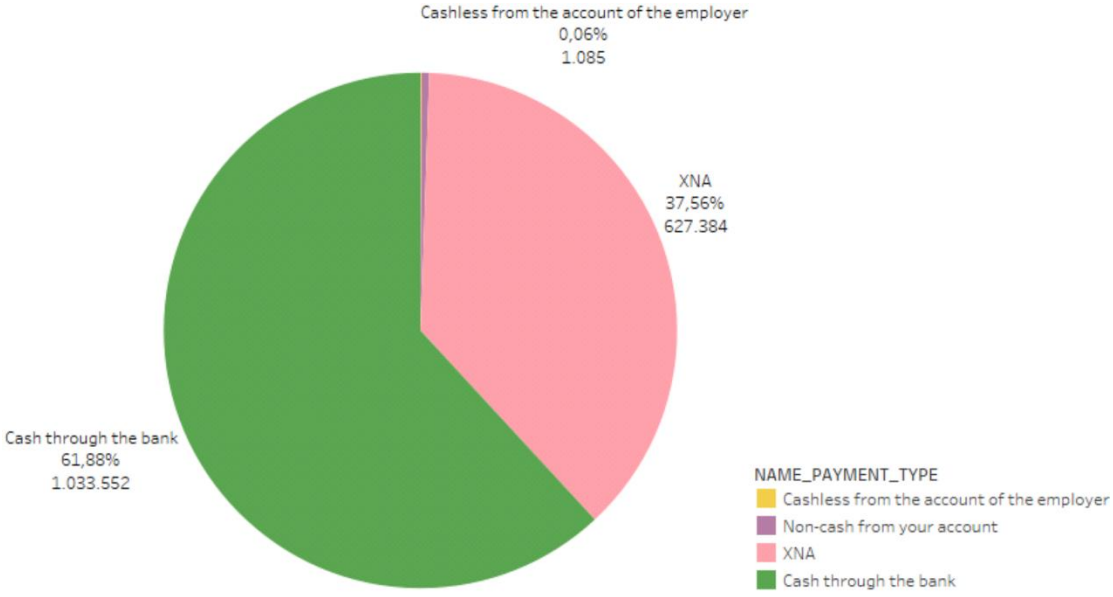
7.15. Κατάσταση προηγούμενων δανείων



Εικόνα 34. Ποσοστά κατάστασης προηγούμενων αιτήσεων δανείου

Φανερό είναι ότι οι προηγούμενες αιτήσεις για δάνειο εγκρίθηκαν κατά ποσοστό 62,07%. Το ποσοστό αυτό, πιθανόν να ήταν μεγαλύτερο αν το 18,94% των αιτήσεων δεν είχαν ακυρωθεί από τους πελάτες.

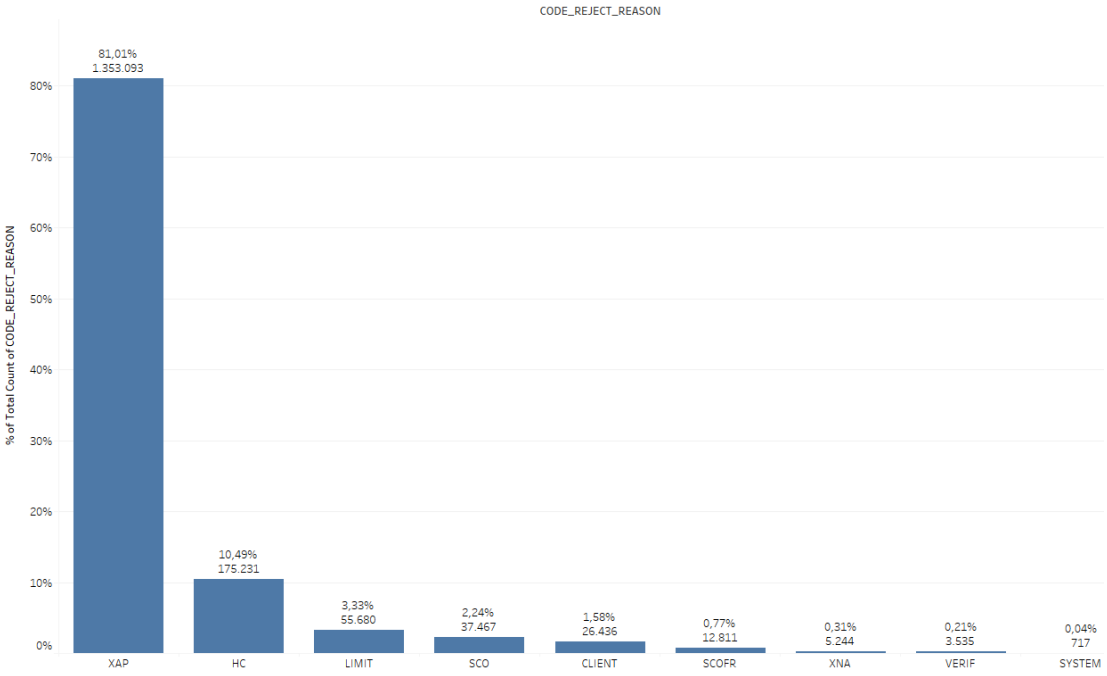
7.16. Μέθοδοι πληρωμής προηγούμενων αιτήσεων δανείου



Εικόνα 35. Επιθυμητός τρόπος πληρωμής δόσης προηγούμενων δανείων

Στις προηγούμενες αιτήσεις για δάνειο, δημοφιλέστερος τρόπος πληρωμής που επιλέχθηκε ήταν η πληρωμή της δόσης μέσω μετρητών σε τράπεζα.

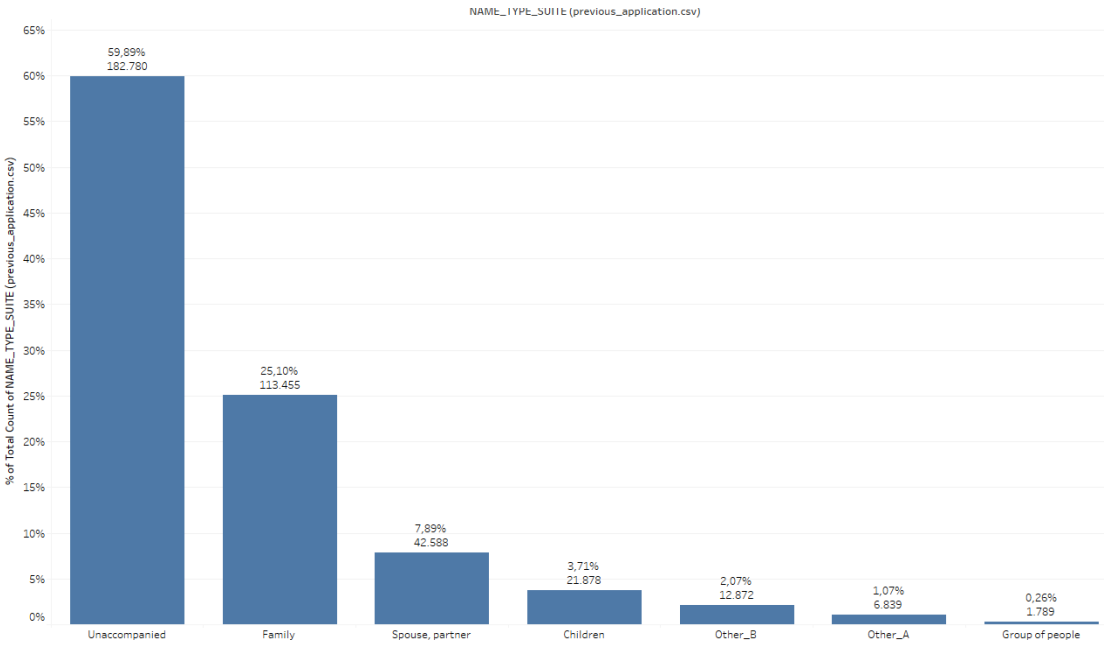
7.17. Αιτίες απόρριψης προηγούμενων αιτήσεων



Εικόνα 36. Ποσοστά λόγων απόρριψης προηγούμενων αιτήσεων

Οι σημαντικότεροι λόγοι απόρριψης των προηγούμενων αιτήσεων, έχουν να κάνουν με το πιστωτικό όριο. Είτε αυτό είναι υψηλό είτε έχει επέλθει.

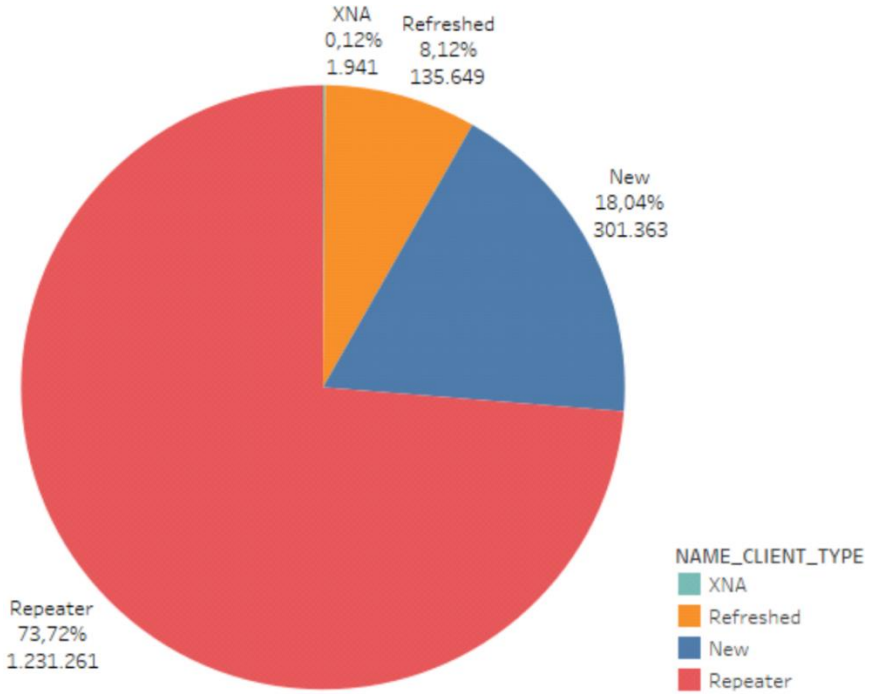
7.18. Συνοδοί πελατών κατά τις προηγούμενες αιτήσεις



Εικόνα 37. Ποσοστά συνοδών προηγούμενων αιτούντων

Όπως και στα τρέχοντα δάνεια, έτσι και κατά τις προηγούμενες αιτήσεις, το μεγαλύτερο ποσοστό των αιτούντων ήταν ασυνόδευτο κατά τη διαδικασία της αίτησης.

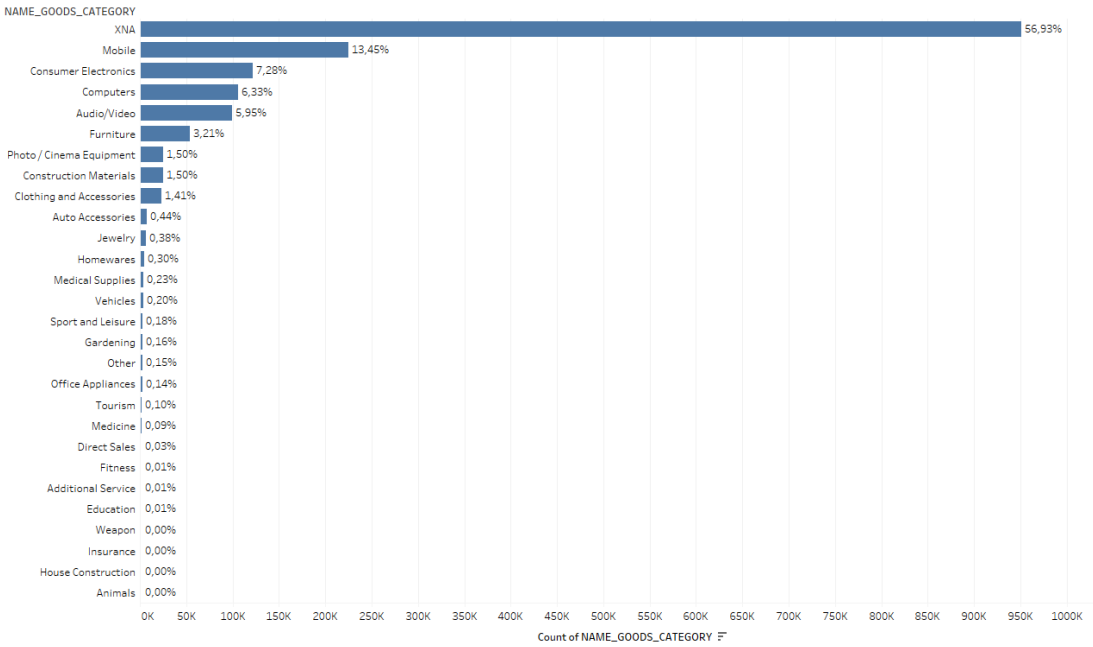
7.19. Ήδη πελάτες ή νέοι κατά τις προηγούμενες αιτήσεις;



Εικόνα 38. Ποσοστά νέων ή υφιστάμενων πελατών προηγούμενων αιτήσεων

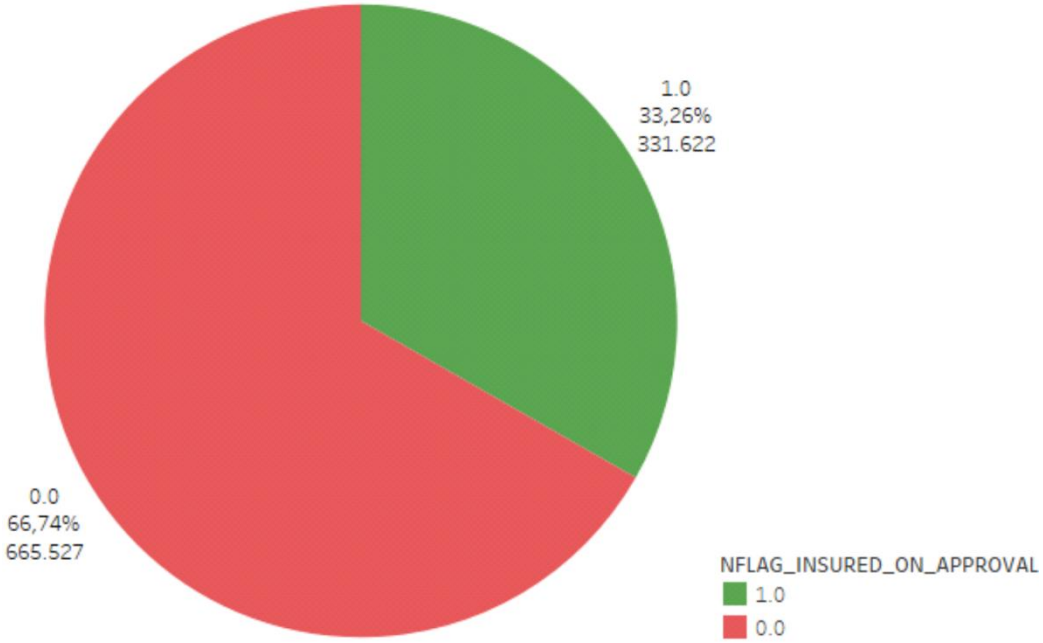
Από το σύνολο των προηγούμενων αιτήσεων για δάνειο, μόνο το 18,04% ήταν νέοι πελάτες. Το υπόλοιπο, αφορούσε ήδη πελάτες της Home Credit.

7.20. Είδος αγαθών προς αγορά προηγούμενων αιτήσεων



Εικόνα 39. Για τι αγαθά έγιναν οι προηγούμενες αιτήσεις δανείου

7.21. Ασφάλεια δανείων στις προηγούμενες αιτήσεις



Εικόνα 40. Ποσοστά ασφάλισης δανείων στις προηγούμενες αιτήσεις

Ένας στους τρεις πελάτες, κατά την προηγούμενη αίτηση για δάνειο, ζήτησε ασφάλεια. Οι δύο στους τρεις, όχι.

7.22. Τιμές «XNA», «XAP» και «NaN»

Στις οπτικοποιήσεις που παρουσιάστηκαν προηγουμένως, παρατηρείται ότι όλες οι τιμές για τα δεδομένα δεν ήταν σαφείς. Συχνά, τα γραφήματα περιείχαν μη κατανοητές τιμές.

Οι τιμές «XNA» και «XAP» για κάποια δεδομένα, δηλώνουν πως είτε κάποιος πελάτης είτε η Home Credit δεν παρείχε τη συγκεκριμένη πληροφορία. Συνεπώς, οι τιμές για τα δεδομένα αυτά μπορεί να ληφθεί ως «NaN». Η λέξη NaN προέρχεται από τα αρχικά των λέξεων «Not a Number».

Άρα, η εμφάνιση «XNA», «XAP» και «NaN» στα σύνολα δεδομένων, λαμβάνεται υπόψη ως άγνωστη τιμή ή έλλειψη τιμής.

8. Προ-επεξεργασία δεδομένων

8.1. Δημιουργία νέων παραμέτρων

Μελετώντας τα δεδομένα, είναι εύκολα διακριτό πως από τα σύνολο δεδομένων (bureau.csv, POS_CASH_balance.csv, previous_application.csv, application_train.csv) είναι εφικτό να προκύψουν νέες παράμετροι. Οι παράμετροι αυτές επεξηγούνται στη συνέχεια.

Η δημιουργία των παραμέτρων θα γίνει στο περιβάλλον Jupyter Notebook της Kaggle. Σαν πρώτο βήμα, πρέπει να φορτωθούν οι απαραίτητες βιβλιοθήκες και σύνολα δεδομένων.

Πρώτα, γίνεται ενσωμάτωση της βιβλιοθήκης pandas με το ψευδώνυμο pd. Η βιβλιοθήκη αυτή είναι απαραίτητη για τη διαχείριση των συνόλων δεδομένων.

Στη συνέχεια, γίνεται η φόρτωση στο Notebook του αρχείου bureau.csv προς επεξεργασία.

```
import pandas as pd
bureau = pd.read_csv('../input/home-credit-default-risk/bureau.csv')
```

8.1.1. Πλήθος προηγούμενων δανείων

Μία πληροφορία που μπορεί να εκρεύσει από τα δεδομένα, είναι η εύρεση του πλήθους των προηγούμενων δανείων που έχει ο κάθε τρέχων πελάτης στο Γραφείο Πίστωσης, από το σύνολο δεδομένων bureau.csv.

Εδώ, γίνεται ομαδοποίηση των εγγραφών ανά τρέχων πελάτη (SK_ID_CURR) και η μέτρηση του πλήθους των εγγραφών DAYS_CREDIT ανά ομάδα εγγραφών. Το πλήθος των εγγραφών δηλώνει το πλήθος των προηγούμενων δανείων για κάθε ομάδα, δηλαδή για κάθε πελάτη. Το πλήθος των προηγούμενων δανείων, καταχωρείται στην παράμετρο bureau_loan_count.

```
bureau_loan_count =
bureau.groupby('SK_ID_CURR')['DAYS_CREDIT'].count().to_frame('bureau_lo
an_count').reset_index()
bureau_loan_count.head()
```

	SK_ID_CURR	bureau_loan_count
0	100001	7
1	100002	8
2	100003	4
3	100004	2
4	100005	3

Εικόνα 41. Πλήθος προηγούμενων δανείων ανά πελάτη

8.1.2. Πλήθος τύπων προηγούμενων δανείων

Από το σύνολο δεδομένων bureau.csv, ενδιαφέρον έχει ο υπολογισμός πόσων τύπων είναι τα προηγούμενα δάνεια για κάθε πελάτη. Λαμβάνει διαφόρων τύπων δάνεια ή επικεντρώνεται σε συγκεκριμένο τύπο;

Εδώ, αρχικά γίνεται μία ομαδοποίηση εγγραφών ανά τρέχων πελάτη και η μέτρηση των μοναδικών τιμών CREDIT_TYPE ανά ομάδα εγγραφών. Το πλήθος των μοναδικών τιμών δηλώνει το πλήθος των διαφορετικών τύπων δανείου για κάθε πελάτη. Το πλήθος των τύπων των προηγούμενων δανείων, καταχωρείται στην παράμετρο bureau_loan_types.

```
bureau_loan_types =
bureau.groupby('SK_ID_CURR')['CREDIT_TYPE'].nunique().to_frame('bureau_
loan_types').reset_index()
bureau_loan_types.head()
```

	SK_ID_CURR	bureau_loan_types
0	100001	1
1	100002	2
2	100003	2
3	100004	1
4	100005	2

Εικόνα 42. Πλήθος τύπων προηγούμενων δανείων ανά πελάτη

8.1.3. Λόγος ενεργών προς συνολικά προηγούμενα δάνεια

Το σύνολο δεδομένων bureau.csv, μέσα από τις πληροφορίες που παρέχει, μπορεί να δώσει μία επιπλέον σημαντική παράμετρο. Η παράμετρος αυτή, συσχετίζει τα ενεργά δάνεια σε σχέση με το σύνολο των δανείων που έχει ο κάθε τρέχων πελάτης στο Γραφείο Πίστωσης.

Στην πραγματικότητα, γίνεται ο υπολογισμός του λόγου των ενεργών δανείων του κάθε πελάτη προς τα συνολικά δάνεια που έχει ο κάθε τρέχων πελάτης στο Γραφείο Πίστωσης.

Αρχικά, γίνεται η επιλογή των επιθυμητών παραμέτρων από το σύνολο δεδομένων.

```
bureau_active_loan_ratio = bureau[['SK_ID_CURR', 'CREDIT_ACTIVE']]
bureau_active_loan_ratio.head()
```

	SK_ID_CURR	CREDIT_ACTIVE
0	215354	Closed
1	215354	Active
2	215354	Active
3	215354	Active
4	215354	Active

Εικόνα 43. Κατάσταση προηγούμενων δανείων πελατών

Στη συνέχεια, γίνεται η επιλογή μόνο των εγγραφών όπου η παράμετρος CREDIT_ACTIVE έχει την τιμή «Active». Στην ουσία, γίνεται με τον τρόπο αυτό η επιλογή μόνο των ενεργών δανείων.

```
bureau_active_loan_ratio =
bureau_active_loan_ratio[bureau_active_loan_ratio.CREDIT_ACTIVE ==
'Active']
bureau_active_loan_ratio.head()
```

	SK_ID_CURR	CREDIT_ACTIVE
1	215354	Active
2	215354	Active
3	215354	Active
4	215354	Active
5	215354	Active

Εικόνα 44. Προηγούμενα ενεργά δάνεια πελατών

Ακολουθεί ο υπολογισμός του πλήθους προηγούμενων ενεργών δανείων ανά τρέχων πελάτη. Για να γίνει αυτό, εφαρμόζεται ομαδοποίηση των εγγραφών ανά τρέχων πελάτη (SK_ID_CURR) και γίνεται η μέτρηση του πλήθους των εγγραφών CREDIT_ACTIVE ανά ομάδα εγγραφών.

Το πλήθος των εγγραφών, δηλώνει το πλήθος των προηγούμενων ενεργών δανείων για κάθε ομάδα, δηλαδή για κάθε πελάτη. Το πλήθος των προηγούμενων δανείων, καταχωρείται στην παράμετρο bureau_active_loan_count.

```
bureau_active_loan_ratio =
bureau_active_loan_ratio.groupby('SK_ID_CURR')['CREDIT_ACTIVE'].count()
.to_frame('bureau_active_loan_count').reset_index()
bureau_active_loan_ratio.head()
```

	SK_ID_CURR	bureau_active_loan_count
0	100001	3
1	100002	2
2	100003	1
3	100005	2
4	100008	1

Εικόνα 45. Πλήθος προηγούμενων ενεργών δανείων ανά πελάτη

Έπειτα, υπολογίζεται το πλήθος των προηγούμενων δανείων ανά τρέχων πελάτη.

```
bureau_loan_count =
bureau.groupby('SK_ID_CURR')['DAYS_CREDIT'].count().to_frame('bureau_lo
an_count').reset_index()
bureau_loan_count.head()
```

Ακολουθεί η συγχώνευση των bureau_active_loan_ratio και bureau_loan_count.

```
bureau_active_loan_ratio =
bureau_active_loan_ratio.merge(bureau_loan_count, on='SK_ID_CURR',
how='left')
bureau_active_loan_ ratio.head()
```

Κατόπιν, υπολογίζεται ο λόγος των ενεργών προηγούμενων δανείων προς το σύνολο των προηγούμενων δανείων ανά τρέχων πελάτη. Ο λόγος αυτός καταχωρείται στην παράμετρο bureau_active_loan_ratio, η οποία μπορεί να λαμβάνει τιμές από μηδέν (0) έως και ένα (1). Όσο μεγαλύτερος αριθμός είναι η τιμή του bureau_active_loan_ratio, τόσα περισσότερα από τα προηγούμενα δάνεια του πελάτη είναι ενεργά σε σχέση με το συνολικό αριθμό των δανείων που έχει ο ίδιος πελάτης.

```
bureau_active_loan_ratio['bureau_active_loan_percentage'] =
bureau_active_loan_ratio['bureau_active_loan_count']/bureau_active_loan
_ratio['bureau_loan_count']
bureau_active_loan_ratio.head()
```

	SK_ID_CURR	bureau_active_loan_count	bureau_loan_count	bureau_active_loan_ratio
0	100001	3	7	0.428571
1	100002	2	8	0.250000
2	100003	1	4	0.250000
3	100005	2	3	0.666667
4	100008	1	3	0.333333

Εικόνα 46. Σύνολο δεδομένων πλήθους προηγούμενων δανείων ανά πελάτη

Τέλος, γίνεται η διαγραφή των παραμέτρων bureau_loan_count και bureau_loan_types που είναι πλέον περιττές, ώστε να παραμείνει μόνο ο επιθυμητός υπολογισμένος λόγος.

```
del bureau_active_loan_ratio['bureau_active_loan_count'],
bureau_active_loan_ratio['bureau_loan_count']
bureau_active_loan_ratio.head()
```

	SK_ID_CURR	bureau_active_loan_ratio
0	100001	0.428571
1	100002	0.250000
2	100003	0.250000
3	100005	0.666667
4	100008	0.333333

Εικόνα 47. Λόγος ενεργών προς συνολικά προηγούμενα δάνεια

8.1.4. Μέσος όρος ημερών που λήγουν τα προηγούμενα δάνεια στο μέλλον

Τα προηγούμενα δάνεια κάθε πελάτη, είτε τρέχουν ακόμη είτε έχουν λήξει. Για την παράμετρο που θα δημιουργηθεί, δε λαμβάνονται υπόψη δάνεια με παρελθοντική ημερομηνία λήξης αλλά μόνο αυτά που λήγουν στο μέλλον.

Η συγκεκριμένη παράμετρος λοιπόν, προκύπτει από τον υπολογισμό του μέσου όρου του πλήθους των ημερών που απομένουν ώστε να λήξουν τα προηγούμενα δάνεια κάθε πελάτη στο μέλλον. Οι πληροφορίες υπάρχουν στο σύνολο δεδομένων bureau.csv.

Στην αρχή, γίνεται η επιλογή των παραμέτρων βάσει των οποίων θα γίνουν οι υπολογισμοί, δηλαδή των SK_ID_CURR και DAYS_CREDIT_ENDDATE.

```
Bureau_avg_enddate_future = bureau[[‘SK_ID_CURR’,
‘DAYS_CREDIT_ENDDATE’]].reset_index(drop = True)
bureau_avg_enddate_future.head()
```

	SK_ID_CURR	DAYS_CREDIT_ENDDATE
0	215354	-153.0
1	215354	1075.0
2	215354	528.0
3	215354	NaN
4	215354	1197.0

Εικόνα 48. Πλήθος ημερών που λήγουν τα προηγούμενα δάνεια

Στη συνέχεια, δημιουργείται η παράμετρος days_credit_enddate_binary, η οποία είναι προσωρινή και σκοπό έχει να ξεχωριστούν οι παρελθοντικές από τις μελλοντικές ημερομηνίες.

```
bureau_avg_enddate_future['days_credit_enddate_binary'] =
bureau_avg_enddate_future['DAYS_CREDIT_ENDDATE']
bureau_avg_enddate_future.head()
```

	SK_ID_CURR	DAYS_CREDIT_ENDDATE	days_credit_enddate_binary
0	215354	-153.0	-153.0
1	215354	1075.0	1075.0
2	215354	528.0	528.0
3	215354	NaN	NaN
4	215354	1197.0	1197.0

Εικόνα 49. Δημιουργία παραμέτρου για τα προηγούμενα δάνεια

Κατόπιν, δημιουργείται μια συνάρτηση σύμφωνα με την οποία οι αρνητικές τιμές γίνονται 0 ενώ όλες οι άλλες τιμές (0, NaN, θετικές τιμές) γίνονται 1.

```
def f(x):
    if x<0:
        y = 0
    else:
        y = 1
    return y
```

Ακολουθεί η εφαρμογή της συνάρτησης στα δεδομένα της παραμέτρου DAYS_CREDIT_ENDDATE. Οι τιμές των αποτελεσμάτων της συνάρτησης, καταχωρούνται στην προσωρινή παράμετρο days_credit_enddate_binary.

Συνεπώς, για την παράμετρο days_credit_enddate_binary, οι τιμές μηδέν (0) δηλώνουν ημερομηνία λήξης που ανήκει στο παρελθόν. Οι τιμές ένα (1) δηλώνουν ημερομηνία λήξης που ανήκει στο μέλλον.

```
bureau_avg_enddate_future['days_credit_enddate_binary'] =
bureau_avg_enddate_future.apply(lambda x: f(x.DAYS_CREDIT_ENDDATE),
axis = 1)
bureau_avg_enddate_future.head()
```

	SK_ID_CURR	DAYS_CREDIT_ENDDATE	days_credit_enddate_binary
0	215354	-153.0	0
1	215354	1075.0	1
2	215354	528.0	1
3	215354	NaN	1
4	215354	1197.0	1

Εικόνα 50. Παράμετρος λογικής μορφής όπου δηλώνει παρελθόν (0) ή μέλλον (1)

Από το σύνολο δεδομένων που προέκυψε, γίνεται λήψη δεδομένων μόνο για τα δάνεια με μελλοντικές ημερομηνίες λήξης, δηλαδή μόνο για τις εγγραφές όπου η παράμετρος bureau_avg_enddate_future έχει την τιμή 1.

```
bureau_avg_enddate_future =
bureau_avg_enddate_future[bureau_avg_enddate_future['days_credit_enddat
e_binary'] == 1]
bureau_avg_enddate_future.head()
```

	SK_ID_CURR	DAYS_CREDIT_ENDDATE	days_credit_enddate_binary
1	215354	1075.0	1
2	215354	528.0	1
3	215354	NaN	1
4	215354	1197.0	1
5	215354	27460.0	1

Εικόνα 51. Προηγούμενα δάνεια που είναι ακόμη σε ισχύ ή άγνωστης κατάστασης

Επιπλέον, επιθυμητό είναι να υπάρχουν δεδομένα μόνο για όσα δάνεια που λήγουν μελλοντικά και υπάρχει καταχωρημένος ο χρόνος λήξης τους. Άρα, πρέπει να γίνει λήψη μόνο των δεδομένων όπου η παράμετρος DAYS_CREDIT_ENDDATE δεν έχει την τιμή NaN.

```
bureau_avg_enddate_future =
bureau_avg_enddate_future[bureau_avg_enddate_future['DAYS_CREDIT_ENDDAT
E'].notnull()]
bureau_avg_enddate_future.head()
```

	SK_ID_CURR	DAYS_CREDIT_ENDDATE	days_credit_enddate_binary
1	215354	1075.0	1
2	215354	528.0	1
4	215354	1197.0	1
5	215354	27460.0	1
6	215354	79.0	1

Εικόνα 52. Προηγούμενα δάνεια που είναι ακόμη σε ισχύ

Τέλος, γίνεται ομαδοποίηση των εγγραφών κατά πελάτη (SK_ID_CURR) καθώς και ο υπολογισμός του μέσου όρου των ημερών που απομένουν ώστε να λήξουν τα προηγούμενα δάνεια στο μέλλον. Ο μέσος όρος, καταχωρείται στην παράμετρο avg_days_enddate.

```
bureau_avg_enddate_future = bureau_avg_enddate_future.groupby(by =
['SK_ID_CURR'])['DAYS_CREDIT_ENDDATE'].mean().to_frame('avg_days_enddat
e').reset_index()
bureau_avg_enddate_future.head()
```

	SK_ID_CURR	avg_days_enddate
0	100001	1030.333333
1	100002	309.000000
2	100003	1216.000000
3	100005	723.000000
4	100008	471.000000

Εικόνα 53. Μέσος όρος ημερών που λήγουν τα προηγούμενα δάνεια του κάθε πελάτη στο μέλλον

8.1.5. Λόγος χρέους προς πίστωση

Η πίστωση που έχουν λάβει οι πελάτες καθώς και το χρέος, είναι δύο ποσά που μεμονωμένα δεν παρέχουν κάποια εξαιρετικά σημαντική πληροφορία. Ο λόγος όμως του χρέους προς την πίστωση είναι μία πληροφορία που σχετίζει τα δύο αυτά ποσά.

Συνεπώς, ο υπολογισμός του λόγου ενδέχεται να είναι μία καλή παράμετρος. Μια υψηλή τιμή για το λόγο αυτόν, μπορεί να σημαίνει πιθανό πρόβλημα στην αποπληρωμή.

Πρωτίστως, πρέπει να γίνει επιλογή των στηλών SK_ID_CURR, AMT_CREDIT_SUM και AMT_CREDIT_SUM_DEBT από το σύνολο δεδομένων bureau.csv.

```
dept_credit_ratio = bureau[['SK_ID_CURR', 'AMT_CREDIT_SUM',
'AMT_CREDIT_SUM_DEBT']].reset_index(drop = True)
dept_credit_ratio.head()
```

	SK_ID_CURR	AMT_CREDIT_SUM	AMT_CREDIT_SUM_DEBT
0	215354	91323.0	0.0
1	215354	225000.0	171342.0
2	215354	464323.5	NaN
3	215354	90000.0	NaN
4	215354	2700000.0	NaN

Εικόνα 54. Ποσά πίστωσης και χρέους προηγούμενων δανείων

Ακολουθεί η αντικατάσταση των τιμών NaN με το μηδέν (0).

```
dept_credit_ratio['AMT_CREDIT_SUM'] =
dept_credit_ratio['AMT_CREDIT_SUM'].fillna(0)
dept_credit_ratio['AMT_CREDIT_SUM_DEBT'] =
dept_credit_ratio['AMT_CREDIT_SUM_DEBT'].fillna(0)
dept_credit_ratio.head()
```

	SK_ID_CURR	AMT_CREDIT_SUM	AMT_CREDIT_SUM_DEBT
0	215354	91323.0	0.0
1	215354	225000.0	171342.0
2	215354	464323.5	0.0
3	215354	90000.0	0.0
4	215354	2700000.0	0.0

Εικόνα 55. Ποσά πίστωσης και χρέους προηγούμενων δανείων με μηδενισμό NaN τιμών

Έπειτα, γίνεται η ομαδοποίηση των εγγραφών ανά τρέχων πελάτη (SK_ID_CURR) και η άθροιση των τιμών της παραμέτρου AMT_CREDIT_SUM ανά ομάδα εγγραφών. Το άθροισμα των τιμών δηλώνει το συνολικό ποσό πίστωσης των προηγούμενων δανείων για κάθε ομάδα, δηλαδή για κάθε πελάτη.

Το αποτέλεσμα του κώδικα, είναι ένα προσωρινό σύνολο δεδομένων με ονομασία dept_credit_ratio_A ενώ το συνολικό ποσό πίστωσης καταχωρείται στην νέα παράμετρο total_credit.

```
dept_credit_ratio_A = dept_credit_ratio.groupby(by =
['SK_ID_CURR'])['AMT_CREDIT_SUM'].sum().to_frame('total_credit').reset_
index()
dept_credit_ratio_A.head()
```

	SK_ID_CURR	total_credit
0	100001	1453365.000
1	100002	865055.565
2	100003	1017400.500
3	100004	189037.800
4	100005	657126.000

Εικόνα 56. Συνολικό ποσό πίστωσης κάθε πελάτη

Ομοίως, έπεται η ομαδοποίηση των εγγραφών ανά τρέχων πελάτη (SK_ID_CURR) και η άθροιση των τιμών της παραμέτρου AMT_CREDIT_SUM_DEBT ανά ομάδα εγγραφών. Το άθροισμα των εγγραφών, δηλώνει το συνολικό χρέος προς τα προηγούμενα δάνεια για κάθε ομάδα, δηλαδή για κάθε πελάτη.

Το αποτέλεσμα του κώδικα είναι το προσωρινό σύνολο δεδομένων dept_credit_ratio_B ενώ το συνολικό χρέος καταχωρείται στη νέα παράμετρο total_debt.

```
dept_credit_ratio_B = dept_credit_ratio.groupby(by =
['SK_ID_CURR'])['AMT_CREDIT_SUM_DEBT'].sum().to_frame('total_debt').res
et_index()
dept_credit_ratio_B.head()
```

	SK_ID_CURR	total_debt
0	100001	596686.5
1	100002	245781.0
2	100003	0.0
3	100004	0.0
4	100005	568408.5

Εικόνα 57. Συνολικό ποσό χρέους κάθε πελάτη

Στη συνέχεια, πρέπει να γίνει η συγχώνευση των συνόλων δεδομένων dept_credit_ratio, dept_credit_ratio_A και dept_credit_ratio_B. Κατόπιν τούτου, υπολογίζεται το αποτέλεσμα της διαίρεσης του συνολικού χρέους προς τη συνολική πίστωση.

Έτσι, προκύπτει η νέα παράμετρος dept_credit_ratio, η οποία είναι ο ζητούμενος λόγος χρέους προς πίστωση. Ο λόγος αυτός, καταχωρείται στη νέα παράμετρο dept_credit_ratio.

```
dept_credit_ratio = dept_credit_ratio.merge(dept_credit_ratio_A, on =
['SK_ID_CURR'], how = 'left')
dept_credit_ratio = dept_credit_ratio.merge(dept_credit_ratio_B, on =
['SK_ID_CURR'], how = 'left')
```

```
dept_credit_ratio['debt_credit_ratio'] =
dept_credit_ratio['total_debt']/dept_credit_ratio['total_credit']
dept_credit_ratio.head()
```

	SK_ID_CURR	AMT_CREDIT_SUM	AMT_CREDIT_SUM_DEBT	total_credit	total_debt	debt_credit_ratio
0	215354	91323.0	0.0	5973945.3	284463.18	0.047617
1	215354	225000.0	171342.0	5973945.3	284463.18	0.047617
2	215354	464323.5	0.0	5973945.3	284463.18	0.047617
3	215354	90000.0	0.0	5973945.3	284463.18	0.047617
4	215354	2700000.0	0.0	5973945.3	284463.18	0.047617

Εικόνα 58. Σύνολο δεδομένων ποσών πίστωσης και χρεών προηγούμενων δανείων

Τέλος, γίνεται η διαγραφή των προσωρινών συνόλων δεδομένων dept_credit_ratio_A και dept_credit_ratio_B. Διαγράφονται επιπλέον και οι μη απαραίτητες πλέον παράμετροι AMT_CREDIT_SUM, AMT_CREDIT_SUM_DEBT, total_credit και total_debt.

Παρατηρείται όμως πως υπάρχει μία επανάληψη στις εγγραφές. Πολλές εγγραφές δείχνουν τον ίδιο πελάτη και τον ίδιο λόγο. Για το λόγο αυτό, το σύνολο δεδομένων θα ομαδοποιηθεί κατά πελάτη και ως τιμή για την παράμετρο dept_credit_ratio θα υπολογιστεί ο μέσος όρος τους, ο οποίος θα είναι η ίδια τιμή.

```
del dept_credit_ratio_A, dept_credit_ratio_B
del dept_credit_ratio['AMT_CREDIT_SUM'],
dept_credit_ratio['AMT_CREDIT_SUM_DEBT'],
dept_credit_ratio['total_credit'], dept_credit_ratio['total_debt']
dept_credit_ratio = dept_credit_ratio.groupby(by =
['SK_ID_CURR'])['debt_credit_ratio'].mean().to_frame('debt_credit_ratio
').reset_index()
dept_credit_ratio.head()
```

	SK_ID_CURR	debt_credit_ratio
0	100001	0.410555
1	100002	0.284122
2	100003	0.000000
3	100004	0.000000
4	100005	0.864992

Εικόνα 59. Λόγος χρέους προς πίστωση ανά πελάτη για προηγούμενα δάνεια

8.1.6. Λόγος ληξιπρόθεσμων οφειλών προς πίστωση

Μία παράμετρος που θα ήταν ικανή να δώσει μία εικόνα της υγιούς κατάστασης αποπληρωμής των προηγούμενων δανείων, είναι ο λόγος των ληξιπρόθεσμων οφειλών

προς τη συνολική πίστωση για τον κάθε πελάτη. Μια υψηλή τιμή για το λόγο αυτόν, μπορεί να είναι σημαίνει πιθανό πρόβλημα στην αποπληρωμή.

Για να υπολογιστεί ο λόγος αυτός, πρώτα πρέπει να γίνει η επιλογή των στηλών SK_ID_CURR, AMT_CREDIT_SUM_DEBT και AMT_CREDIT_SUM_OVERDUE από το σύνολο δεδομένων bureau.csv.

```
overdue_dept_ratio = bureau[['SK_ID_CURR', 'AMT_CREDIT_SUM_DEBT',
'AMT_CREDIT_SUM_OVERDUE']].reset_index(drop = True)
overdue_dept_ratio.head()
```

	SK_ID_CURR	AMT_CREDIT_SUM_DEBT	AMT_CREDIT_SUM_OVERDUE
0	215354	0.0	0.0
1	215354	171342.0	0.0
2	215354	NaN	0.0
3	215354	NaN	0.0
4	215354	NaN	0.0

Εικόνα 60. Ποσά συνολικού χρέους και ληξιπρόθεσμων οφειλών προηγούμενων δανείων

Έπειτα, γίνεται η αντικατάσταση των τιμών NaN που υπάρχουν στο σύνολο δεδομένων, με το μηδέν (0).

```
overdue_dept_ratio['AMT_CREDIT_SUM_DEBT'] =
overdue_dept_ratio['AMT_CREDIT_SUM_DEBT'].fillna(0)
overdue_dept_ratio['AMT_CREDIT_SUM_OVERDUE'] =
overdue_dept_ratio['AMT_CREDIT_SUM_OVERDUE'].fillna(0)
overdue_dept_ratio.head()
```

	SK_ID_CURR	AMT_CREDIT_SUM_DEBT	AMT_CREDIT_SUM_OVERDUE
0	215354	0.0	0.0
1	215354	171342.0	0.0
2	215354	0.0	0.0
3	215354	0.0	0.0
4	215354	0.0	0.0

Εικόνα 61. Ποσά συνολικού χρέους και ληξιπρόθεσμων οφειλών προηγούμενων δανείων με των τιμών NaN

Στη συνέχεια, γίνεται ομαδοποίηση των εγγραφών ανά τρέχων πελάτη (SK_ID_CURR) και η άθροιση των τιμών AMT_CREDIT_SUM_DEBT ανά ομάδα εγγραφών, που δηλώνει το συνολικό χρέος για κάθε πελάτη.

Το συνολικό χρέος, καταχωρείται στη νέα παράμετρο total_debt του προσωρινού συνόλου δεδομένων overdue_dept_ratio_A.

```
overdue_dept_ratio_A = overdue_dept_ratio.groupby(by =
['SK_ID_CURR'])['AMT_CREDIT_SUM_DEBT'].sum().to_frame('total_debt').res
et_index()
overdue_dept_ratio_A.head()
```

Το αποτέλεσμα του κώδικα είναι ακριβώς το ίδιο με αυτό της Εικόνας 56.

Ομοίως, γίνεται ομαδοποίηση των εγγραφών ανά τρέχων πελάτη (SK_ID_CURR) και η άθροιση των τιμών AMT_CREDIT_SUM_OVERDUE ανά ομάδα εγγραφών. Το άθροισμα αυτό δηλώνει το συνολικό ληξιπρόθεσμο ποσό των προηγούμενων δανείων για κάθε πελάτη.

Το συνολικό ποσό πίστωσης, καταχωρείται στην νέα παράμετρο total_overdue του προσωρινού συνόλου δεδομένων overdue_dept_ratio_B.

```
overdue_dept_ratio_B = overdue_dept_ratio.groupby(by =
['SK_ID_CURR'])['AMT_CREDIT_SUM_OVERDUE'].sum().to_frame('total_overdue
').reset_index()
overdue_dept_ratio_B.head()
```

	SK_ID_CURR	total_overdue
0	100001	0.0
1	100002	0.0
2	100003	0.0
3	100004	0.0
4	100005	0.0

Εικόνα 62. Συνολικό ληξιπρόθεσμο ποσό προηγούμενων δανείων ανά πελάτη

Ακολουθεί η συγχώνευση των συνόλων δεδομένων dept_credit_ratio, dept_credit_ratio_A και dept_credit_ratio_B καθώς και η δημιουργία μίας νέας παραμέτρου με το όνομα overdue_dept_ratio.

Η νέα αυτή παράμετρος είναι ο ζητούμενος λόγος των ληξιπρόθεσμων οφειλών προς τη συνολική πίστωση για τον κάθε πελάτη.

```
overdue_dept_ratio = overdue_dept_ratio.merge(overdue_dept_ratio_A, on
= ['SK_ID_CURR'], how = 'left')
overdue_dept_ratio = overdue_dept_ratio.merge(overdue_dept_ratio_B, on
= ['SK_ID_CURR'], how = 'left')
overdue_dept_ratio['overdue_dept_ratio'] =
overdue_dept_ratio['total_overdue']/overdue_dept_ratio['total_debt']
overdue_dept_ratio.head()
```

	SK_ID_CURR	AMT_CREDIT_SUM_DEBT	AMT_CREDIT_SUM_OVERDUE	total_debt	total_overdue	overdue_dept_ratio
0	215354	0.0	0.0	284463.18	0.0	0.0
1	215354	171342.0	0.0	284463.18	0.0	0.0
2	215354	0.0	0.0	284463.18	0.0	0.0
3	215354	0.0	0.0	284463.18	0.0	0.0
4	215354	0.0	0.0	284463.18	0.0	0.0

Εικόνα 63. Σύνολο δεδομένων ληξιπρόθεσμων οφειλών πελατών για προηγούμενα δάνεια

Τέλος, γίνεται η διαγραφή των προσωρινών συνόλων δεδομένων overdue_dept_ratio_A και overdue_dept_ratio_B. Επίσης, γίνεται η διαγραφή των μη απαραίτητων πλέον παραμέτρων AMT_CREDIT_SUM_DEBT, AMT_CREDIT_SUM_OVERDUE, total_debt και total_overdue.

```
del overdue_dept_ratio_A, overdue_dept_ratio_B
del overdue_dept_ratio['AMT_CREDIT_SUM_DEBT'],
overdue_dept_ratio['AMT_CREDIT_SUM_OVERDUE'],
overdue_dept_ratio['total_debt'], overdue_dept_ratio['total_overdue']
overdue_dept_ratio.head()
```

Όπως και προηγουμένως, πολλές εγγραφές δείχνουν τον ίδιο πελάτη και τον ίδιο λόγο. Έτσι, το σύνολο δεδομένων θα ομαδοποιηθεί κατά πελάτη και ως τιμή για την παράμετρο overdue_dept_ratio θα υπολογιστεί ο μέσος όρος τους, ο οποίος θα είναι η ίδια τιμή.

	SK_ID_CURR	overdue_dept_ratio
0	215354	0.0
1	215354	0.0
2	215354	0.0
3	215354	0.0
4	215354	0.0

Εικόνα 64. Λόγος ληξιπρόθεσμων οφειλών προς πίστωση για προηγούμενα δάνεια

8.1.7. Μέσος όρος ημερών καθυστέρησης πληρωμής δόσεων δανείων σε μετρητά

Το σύνολο δεδομένων POS_CASH_balance.csv διαθέτει πληροφορίες για το μηνιαίο ιστορικό των προηγούμενων δανείων σε μετρητά στη Home Credit. Παρέχει πληροφορίες σχετικά με τις ημέρες καθυστέρησης πληρωμής των δόσεων των δανείων σε μετρητά από το Γραφείο Πιστώσεων κατά τη στιγμή της αίτησης για δάνειο στο δείγμα μας.

Αρχικά, γίνεται η φόρτωση του συνόλου δεδομένων POS_CASH_balance.csv.

```
poscash = pd.read_csv('../input/home-credit-default-
risk/POS_CASH_balance.csv')
poscash[poscash['SK_ID_CURR'] == 100008]
```

	SK_ID_PREV	SK_ID_CURR	MONTHS_BALANCE	CNT_INSTALMENT	CNT_INSTALMENT_FUTURE	NAME_CONTRACT_STATUS	SK_DPD	SK_DPD_DEF
252978	2613879	100008	-19	30.0	28.0	Active	0	0
671303	2613879	100008	-13	8.0	0.0	Completed	0	0
938913	2613879	100008	-17	30.0	26.0	Active	0	0
1366622	1907290	100008	-72	10.0	5.0	Active	0	0
1647683	2218188	100008	-8	10.0	5.0	Active	0	0
...	...	...	...	...	...	...	...	...
9494319	1907290	100008	-64	10.0	0.0	Active	106	0
9585119	1907290	100008	-58	10.0	0.0	Active	290	0
9586973	1907290	100008	-61	10.0	0.0	Active	198	0
9865449	1907290	100008	-57	10.0	0.0	Active	320	0
9924522	2218188	100008	-3	10.0	0.0	Active	0	0

Εικόνα 65. Ημέρες καθυστέρησης πληρωμής δόσεων για τον πελάτη με ID 100008

Στη συνέχεια, γίνεται η επιλογή των στηλών SK_ID_CURR και SK_DPD από το σύνολο δεδομένων. Η παράμετρος SK_DPD δηλώνει τις ημέρες καθυστέρησης πληρωμής της δόσης για κάποιο μήνα.

```
avg_pos_cash_dpd = poscash[['SK_ID_CURR', 'SK_DPD']]
avg_pos_cash_dpd.head()
```

Κατόπιν, γίνεται ομαδοποίηση των εγγραφών κατά πελάτη (SK_ID_CURR) και υπολογισμός του μέσου όρου των ημερών καθυστέρησης πληρωμής δόσεων. Ο μέσος όρος, καταχωρείται στην παράμετρο avg_pos_cash_dpd.

```
avg_pos_cash_dpd = avg_pos_cash_dpd.groupby(by =
['SK_ID_CURR'])['SK_DPD'].mean().to_frame('avg_pos_cash_dpd').reset_ind
ex()
avg_pos_cash_dpd.head()
```

	SK_ID_CURR	avg_pos_cash_dpd
0	100001	0.777778
1	100002	0.000000
2	100003	0.000000
3	100004	0.000000
4	100005	0.000000

Εικόνα 66. Μέσος όρος ημερών καθυστέρησης πληρωμής δόσεων δανείων σε μετρητά

Μία τιμή ίση με μηδέν (0) στη νέα παράμετρο, σημαίνει ότι ο πελάτης είναι συνεπής στις πληρωμές του. Όσο πιο υψηλή είναι η τιμή της παραμέτρου, σημαίνει τόσο πιο ασυνεπή πελάτη στις υποχρεώσεις του πληρωμής των δόσεων.

8.1.8. Λόγος προηγούμενων απορριπτέων αιτήσεων προς συνολικές προηγούμενες αιτήσεις

Το σύνολο δεδομένων previous_application.csv παρέχει πληροφορίες για προηγούμενες αιτήσεις δανείων που έγιναν στη Home Credit. Κάποιες από τις προηγούμενες αιτήσεις εγκρίθηκαν ενώ κάποιες όχι.

Περισσότερο σημαντική είναι η σχέση μεταξύ του συνόλου των προηγούμενων αιτήσεων και των απορριπτέων προηγούμενων αιτήσεων. Περισσότερο σημαντικός λοιπόν είναι ο υπολογισμός του λόγου των προηγούμενων δανείων που απορρίφθηκαν προς το συνολικό αριθμό των προηγούμενων αιτήσεων για κάθε πελάτη στο δείγμα μας.

Μια υψηλή τιμή για το λόγο αυτόν, σημαίνει πως κάτι συμβαίνει με τον πελάτη ώστε να απορρίπτονται οι αιτήσεις του και συνεπώς μπορεί να σημαίνει πιθανό πρόβλημα στην αποπληρωμή ενός δανείου.

Αρχικά, γίνεται η φόρτωση συνόλου δεδομένων previous_application.csv ενώ στη συνέχεια γίνεται ομαδοποίηση εγγραφών ανά τρέχων πελάτη (SK_ID_CURR) και μέτρηση του πλήθους των εγγραφών SK_ID_PREV ανά ομάδα εγγραφών. Το πλήθος των εγγραφών δηλώνει το πλήθος των προηγούμενων αιτήσεων για κάθε ομάδα, δηλαδή για κάθε πελάτη. Το πλήθος των προηγούμενων αιτήσεων δανείων στη Home Credit, καταχωρείται στην παράμετρο prev_apps.

```
previous_application = pd.read_csv('../input/home-credit-default-
risk/previous_application.csv')
prev_apps =
previous_application['SK_ID_PREV'].groupby(previous_application['SK_ID_
CURR']).count().to_frame('prev_apps').reset_index()
prev_apps.head()
```

Στη συνέχεια, πρέπει να γίνει υπολογισμός του πλήθους των προηγούμενων αιτήσεων δανείων στη Home Credit που απορρίφθηκαν. Για να επιτευχθεί αυτό, θα γίνει φιλτράρισμα εγγραφών ώστε να εμφανίζονται μόνο τα δάνεια που απορρίφθηκαν (δηλαδή όπου NAME_CONTRACT_STATUS = Refused).

```
prev_refused_apps =
previous_application[previous_application['NAME_CONTRACT_STATUS'] ==
'Refused']
```

Έπειτα, ακολουθεί η ομαδοποίηση των εγγραφών ανά τρέχων πελάτη (SK_ID_CURR) και η μέτρηση του πλήθους των εγγραφών SK_ID_PREV ανά ομάδα εγγραφών. Το πλήθος των εγγραφών SK_ID_PREV δηλώνει το πλήθος των προηγούμενων αιτήσεων που απορρίφθηκαν για κάθε ομάδα, δηλαδή για κάθε πελάτη. Το πλήθος των προηγούμενων αιτήσεων δανείων στη Home Credit που απορρίφθηκαν, καταχωρείται στην παράμετρο prev_refused_apps.

```
prev_refused_apps =
prev_refused_apps['SK_ID_PREV'].groupby(prev_refused_apps['SK_ID_CURR']
).count().to_frame('prev_refused_apps').reset_index()
prev_refused_apps.head()
```

Ακολουθεί η συγχώνευση πινάκων prev_apps και prev_refused_apps στο νέο πίνακα prev_refused_apps_ratio.

```
prev_refused_apps_ratio = prev_apps.merge(prev_refused_apps, on =
['SK_ID_CURR'], how = 'left')
prev_refused_apps_ratio.head()
```

Όπου prev_refused_apps = NaN, σημαίνει ότι για το συγκεκριμένο SK_ID_CURR δεν απορρίφθηκε κανένα δάνειο άρα μπορούν να αντικατασταθούν οι τιμές NaN από το μηδέν (0).

```
prev_refused_apps_ratio['prev_refused_apps'] =
prev_refused_apps_ratio['prev_refused_apps'].fillna(0)
prev_refused_apps_ratio.head()
```

	SK_ID_CURR	prev_apps	prev_refused_apps
0	100001	1	0.0
1	100002	1	0.0
2	100003	3	0.0
3	100004	1	0.0
4	100005	2	0.0

Εικόνα 67. Προηγούμενες αιτήσεις και προηγούμενες απορριφθείσες αιτήσεις στη Home Credit

Τέλος, ακολουθεί ο υπολογισμός του λόγου των προηγούμενων απορριπτέων αιτήσεων δανείων προς συνολικές προηγούμενες αιτήσεις. Ο λόγος αυτός, καταχωρείται στη νέα παράμετρο prev_refused_apps_ratio.

```
prev_refused_apps_ratio['prev_refused_apps_ratio'] =
prev_refused_apps_ratio['prev_refused_apps']/prev_refused_apps_ratio['p
rev_apps']
```

Καθώς δε χρειάζονται πλέον οι στήλες prev_apps και prev_refused_apps, αυτές θα διαγραφούν.

```
del prev_refused_apps_ratio['prev_apps'],
prev_refused_apps_ratio['prev_refused_apps']
prev_refused_apps_ratio.head()
```

	SK_ID_CURR	prev_refused_apps_ratio
0	100001	0.0
1	100002	0.0
2	100003	0.0
3	100004	0.0
4	100005	0.0

Εικόνα 68. Λόγος απορριπτέων προηγούμενων αιτήσεων δανείων προς συνολικές προηγούμενες αιτήσεις στη Home Credit

8.1.9. Λόγος ποσού αιτούμενου δανείου προς κόστος αγαθών

Κάποιο ενδιαφέρον έχει και ο υπολογισμός του λόγου του ποσού του δανείου προς το ποσό που κοστίζουν τα αγαθά για τα οποία προορίζεται η τρέχουσα αίτηση δανείου. Διαθέτει ο πελάτης κάποιο χρηματικό ποσό για τα αγαθά που επιθυμεί ή τα περιμένει εξ ολοκλήρου από τη Home Credit;

Μια χαμηλή τιμή για το λόγο αυτόν, σημαίνει πως το δάνειο ζητείται για να καλύψει ένα μικρό μέρος του ποσού των αγαθών.

Μια πολύ υψηλή τιμή, σημαίνει πως το δάνειο ζητείται για να καλύψει ένα πολύ μεγάλο μέρος του ποσού των αγαθών. Αυτό σημαίνει πως ο πελάτης στην ουσία δε χρειάζεται απλά μία οικονομική υποστήριξη για την αγορά του αλλά χρηματοδότη. Αυτό πιθανώς σημαίνει πως δεν υπάρχει πραγματική ανάγκη του πελάτη για τα αγαθά αυτά, συνεπώς ενδέχεται να υπάρξει πρόβλημα στην αποπληρωμή.

Πρωτίστως, γίνεται η φόρτωση του συνόλου δεδομένων application_train.csv.

```
application = pd.read_csv('../input/home-credit-default-
risk/application_train.csv')
application.head()
```

Ακολουθεί η επιλογή τριών στηλών που δίνουν τις ζητούμενες πληροφορίες, των SK_ID_CURR, AMT_CREDIT και AMT_GOODS_PRICE.

```
currCredit_currGoodsPrice_ratio = application[['SK_ID_CURR',
'AMT_CREDIT', 'AMT_GOODS_PRICE']].reset_index(drop = True)
currCredit_currGoodsPrice_ratio.head()
```

Έπειτα, γίνεται ο υπολογισμός του λόγου του ποσού του αιτούμενου δανείου προς το κόστος αγαθών. Ο λόγος αυτός καταχωρείται στη νέα παράμετρο με όνομα currCredit_currGoodsPrice_ratio.

```
currCredit_currGoodsPrice_ratio['currCredit_currGoodsPrice_ratio'] =
currCredit_currGoodsPrice_ratio['AMT_CREDIT']/currCredit_currGoodsPrice
_ratio['AMT_GOODS_PRICE']
currCredit_currGoodsPrice_ratio.head()
```

Οι στήλες AMT_CREDIT και AMT_GOODS_PRICE πλέον δεν είναι απαραίτητες, οπότε και διαγράφονται.

```
del currCredit_currGoodsPrice_ratio['AMT_CREDIT'],
currCredit_currGoodsPrice_ratio['AMT_GOODS_PRICE']
currCredit_currGoodsPrice_ratio.head()
```

	SK_ID_CURR	currCredit_currGoodsPrice_ratio
0	100002	1.158397
1	100003	1.145199
2	100004	1.000000
3	100006	1.052803
4	100007	1.000000

Εικόνα 69. Λόγος ποσού αιτούμενου δανείου προς κόστος αγαθών

8.1.10. Λόγος ετήσιου ποσού καταβολής δανείου προς ετήσιο εισόδημα πελάτη

Από το ίδιο σύνολο δεδομένων application_train.csv, μπορούν να αντληθούν δύο πληροφορίες. Το ετήσιο εισόδημα του πελάτη αλλά και το ποσό που θα πρέπει να καταβάλει ο πελάτης κάθε έτος για την αποπληρωμή του δανείου. Είναι απόλυτα κατανοητό πως αυτά τα ποσά πρέπει να απέχουν μεταξύ τους εφόσον ο πελάτης έχει και άλλες οικονομικές υποχρεώσεις πέραν του δανείου.

Ο υπολογισμός του λόγου του ετήσιου ποσού καταβολής του δανείου προς το ετήσιο εισόδημα πελάτη είναι μία ένδειξη αν τα ποσά αυτά πλησιάζουν το ένα το άλλο.

Μια υψηλή τιμή για το λόγο αυτόν, σημαίνει πως ένα πολύ μεγάλο μέρος του εισοδήματος του πελάτη θα πρέπει να σπαταληθεί για την αποπληρωμή του δανείου, χωρίς να υπολείπεται κάποιο επαρκές ποσό για να καλυφθούν οι υπόλοιπες καθημερινές του ανάγκες, συνεπώς ενδέχεται να υπάρξει πρόβλημα στην αποπληρωμή.

Για τον υπολογισμό, θα πρέπει να επιλεχθούν οι στήλες SK_ID_CURR, AMT_ANNUITY και AMT_INCOME_TOTAL από το σύνολο δεδομένων application_train.csv.

```
annuityamt_income_ratio = application[['SK_ID_CURR', 'AMT_ANNUITY',
'AMT_INCOME_TOTAL']].reset_index(drop = True)
annuityamt_income_ratio.head()
```

Ακολουθεί ο υπολογισμός του λόγου ετήσιου ποσού καταβολής δανείου προς ετήσιο εισόδημα πελάτη και καταχωρείται στην παράμετρο annuityamt_income_ratio.

```
annuityamt_income_ratio['annuityamt_income_ratio'] =
annuityamt_income_ratio['AMT_ANNUITY']/annuityamt_income_ratio['AMT_INC
OME_TOTAL']
annuityamt_income_ratio.head()
```

Τέλος, γίνεται η διαγραφή στηλών AMT_ANNUITY και AMT_INCOME_TOTAL.

```
del annuityamt_income_ratio['AMT_ANNUITY'],
annuityamt_income_ratio['AMT_INCOME_TOTAL']
annuityamt_income_ratio.head()
```

	SK_ID_CURR	annuityamt_income_ratio
0	100002	0.121978
1	100003	0.132217
2	100004	0.100000
3	100006	0.219900
4	100007	0.179963

Εικόνα 70. Λόγος ετήσιου ποσού καταβολής δανείου προς ετήσιο εισόδημα πελάτη

8.2. Συγχώνευση συνόλων δεδομένων

Το σύνολο δεδομένων που περιέχει πληροφορίες σχετικά με τις αιτήσεις, είναι το αρχείο application_train.csv. Κατά συνέπεια, κάθε νέα παράμετρος που δημιουργήθηκε θα πρέπει να συγχωνευθεί σε αυτό.

```
application = pd.read_csv('../input/home-credit-default-
risk/application_train.csv')
print(application.shape)
application_new = application.merge(bureau_loan_count, on =
['SK_ID_CURR'], how = 'left')
application_new = application_new.merge(bureau_loan_types, on =
['SK_ID_CURR'], how = 'left')
application_new = application_new.merge(bureau_active_loan_percentage,
on = ['SK_ID_CURR'], how = 'left')
application_new = application_new.merge(bureau_avg_enddate_future, on =
['SK_ID_CURR'], how = 'left')
application_new = application_new.merge(dept_credit_ratio, on =
['SK_ID_CURR'], how = 'left')
application_new = application_new.merge(overdue_dept_ratio, on =
['SK_ID_CURR'], how = 'left')
application_new = application_new.merge(avg_pos_cash_dpd, on =
['SK_ID_CURR'], how = 'left')
application_new = application_new.merge(prev_refused_apps_ratio, on =
['SK_ID_CURR'], how = 'left')
application_new =
application_new.merge(currCredit_currGoodsPrice_ratio, on =
['SK_ID_CURR'], how = 'left')
application_new = application_new.merge(annuityamt_income_ratio, on =
['SK_ID_CURR'], how = 'left')
print(application_new.shape)
application_new.head()
```

Στο νέο σύνολο δεδομένων, μετά τις συγχωνεύσεις, δίδεται η ονομασία application_new.

Στον πιο πάνω κώδικα, με τις εντολές print(application.shape) και print(application_new.shape) εμφανίζονται τα σχήματα του αρχικού και του τελικού συνόλων δεδομένων. Είναι εμφανές, πως ενώ οι γραμμές των δύο συνόλων είναι οι ίδιες σε πλήθος, οι στήλες διαφέρουν κατά δέκα (10), όσες είναι και οι νέες παράμετροι που δημιουργήθηκαν.

```
(307511, 122)
(307511, 132)
```

_CREDIT	AMT_ANNUITY	...	bureau_loan_count	bureau_loan_types	bureau_active_loan_ratio	avg_days_enddate	debt_credit_ratio	overdue_dept_ratio	avg_pos_cash_dpd	prev_refused_apps_ratio	currCredit_currGoodsPrice_ratio	annuityamt_income_ratio
406597.5	24700.5	...	8.0	2.0	0.25	309.0	0.284122	0.0	0.0	0.000000	1.158397	0.121978
293502.5	35698.5	...	4.0	2.0	0.25	1216.0	0.000000	NaN	0.0	0.000000	1.145199	0.132217
135000.0	6750.0	...	2.0	1.0	NaN	NaN	0.000000	NaN	0.0	0.000000	1.000000	0.100000
312682.5	29686.5	...	NaN	NaN	NaN	NaN	NaN	NaN	0.0	0.111111	1.052803	0.219900
513000.0	21865.5	...	1.0	1.0	NaN	NaN	0.000000	NaN	0.0	0.000000	1.000000	0.179963

Εικόνα 71. Το τελικό σύνολο δεδομένων (εμφάνιση νέων παραμέτρων)

9. Μηχανική μάθηση

Όπως έχει αναφερθεί, στόχος είναι να προβλεφθεί για κάθε αίτηση δανείου, αν αυτό θα αποπληρωθεί ή όχι.

Πιο συγκεκριμένα, αυτό που ζητείται είναι για κάθε SK_ID_CURR στο σύνολο δοκιμής (train set), να προβλεφθεί μία πιθανότητα για τη μεταβλητή TARGET. Το αρχείο πρέπει να περιέχει μία κεφαλίδα και να έχει την ακόλουθη μορφή:

```
SK_ID_CURR,TARGET
100001,0.1
100005,0.9
100013,0.2
etc.
```

Εικόνα 72. Ζητούμενη μορφή αρχείου υποβολής

Καθώς η παρούσα μελέτη δε θα λάβει μέρος στο διαγωνισμό, το προς υποβολή αρχείο δε θα δημιουργηθεί.

9.1. Το σύνολο δεδομένων

Το σύνολο δεδομένων που θα χρησιμοποιηθεί στη μηχανική μάθηση, είναι το συγχωνευμένο σύνολο δεδομένων application_new που δημιουργήθηκε προηγουμένως και περιέχει 132 στήλες.

Για να επιτευχθεί η μηχανική μάθηση, είναι απαραίτητο να γίνει η φόρτωση κάποιων απαραίτητων βιβλιοθηκών.

Μία εξ αυτών είναι η XGBoost (Extreme Gradient Boosting). Πρόκειται για μία βιβλιοθήκη μηχανικής μάθησης με ενισχυτική διαβάθμιση δέντρων αποφάσεων (GBDT). Είναι η κορυφαία βιβλιοθήκη μηχανικής μάθησης για προβλήματα παλινδρόμησης, ταξινόμησης και κατάταξης. Η ενισχυτική διαβάθμιση εκπαιδεύει διαδοχικά αδύναμους μαθητές όπου κάθε αδύναμος μαθητής προσπαθεί να διορθώσει τα λάθη του προκατόχου του. Για τη βιβλιοθήκη XGBoost συνήθως χρησιμοποιείται το ψευδώνυμο xgb.

Η NumPy είναι μια βιβλιοθήκη για τη υποστήριξη μεγάλων πολυδιάστατων πινάκων, διαθέτοντας μεγάλη συλλογή μαθηματικών συναρτήσεων για τη διαχείριση των πινάκων. Συνήθως, χρησιμοποιείται με το ψευδώνυμο np.

Η matplotlib.pyplot είναι μια συλλογή συναρτήσεων της βιβλιοθήκης matplotlib. Χρησιμοποιείται για τη δημιουργία γραφημάτων.

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

Στη συνέχεια, γίνεται η φόρτωση των πρωτότυπων δεδομένων με διαφορετικό όνομα ώστε το πρωτότυπο σύνολο δεδομένων να μείνει ανέπαφο. Ακολουθεί μία προβολή του σχήματος των δεδομένων.

```
orig_data = application_new
orig_data.shape
orig_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 307511 entries, 0 to 307510
Columns: 132 entries, SK_ID_CURR to annuityamt_income_ratio
dtypes: float64(75), int64(41), object(16)
memory usage: 312.0+ MB
```

Εικόνα 73. Το σχήμα των δεδομένων

Παρατηρώντας το σχήμα των δεδομένων, διαπιστώνεται πως υπάρχουν 307.511 εγγραφές και 132 στήλες. Οι περισσότερες στήλες είναι τύπου «float64» ενώ 16 στήλες είναι τύπου «object». Οι στήλες τύπου «object» θα πρέπει να μετατραπούν σε αριθμητικά δεδομένα ώστε να είναι δυνατή η επεξεργασία τους.

Για τη μετατροπή στηλών από μη αριθμητικά δεδομένα σε αριθμητικά, μία δεδομένη συνάρτηση που χρησιμοποιείται είναι η LabelEncoder της βιβλιοθήκης scikit-learn, οπότε και θα πρέπει να γίνει εισαγωγή της.

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

Για τη μετατροπή του τύπου των δεδομένων, θα πρέπει πρώτα να γίνει η δημιουργία ενός αντικειμένου για τη συνάρτηση LabelEncoder. Παράλληλα, θα πρέπει να δημιουργηθεί μία μηδενική αριθμητική μεταβλητή η οποία θα αυξάνεται κατά μία μονάδα κάθε φορά που μετατρέπεται μία στήλη τύπου «object» σε αριθμητικά δεδομένα.

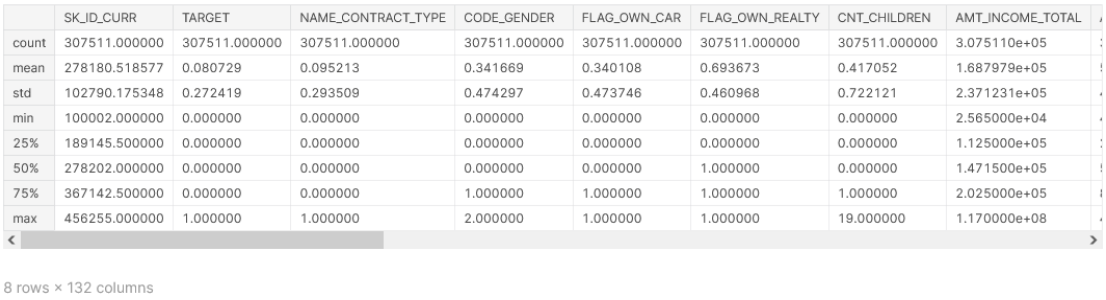
```
le = LabelEncoder()
le_count = 0
for col in orig_data:
    if orig_data[col].dtype == 'object':
        # Εφαρμογή στην τρέχουσα στήλη
        le.fit(orig_data[col])
        # Μετατροπή δεδομένων
        orig_data[col] = le.transform(orig_data[col])
        # Αύξηση της μεταβλητής κατά 1 με σκοπό τη μέτρηση των στηλών
που μετατράπηκαν
        le_count += 1
print('%d στήλες έχουν μετατραπεί.' % le_count)
```

```
16 στήλες έχουν μετατραπεί.
```

Εικόνα 74. Μετατροπή στηλών τύπου «object» σε αριθμητικά δεδομένα

Ακολουθεί μία πιο προσεκτική ματιά στην κατανομή των δεδομένων.

```
orig_data.describe()
```



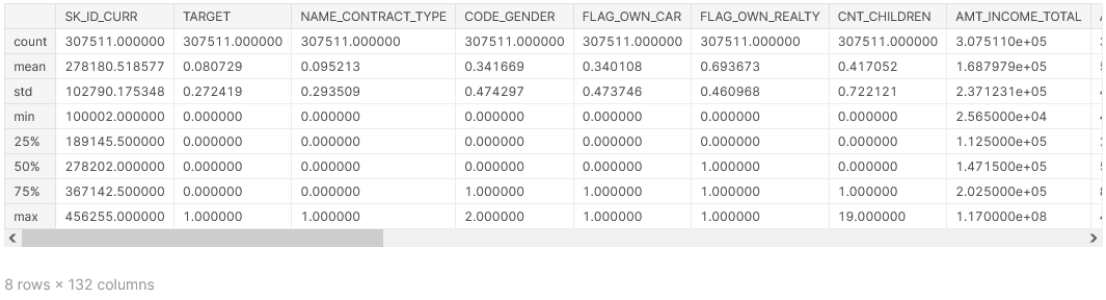
	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL
count	307511.000000	307511.000000	307511.000000	307511.000000	307511.000000	307511.000000	307511.000000	3.075110e+05
mean	278180.518577	0.080729	0.095213	0.341669	0.340108	0.693673	0.417052	1.687979e+05
std	102790.175348	0.272419	0.293509	0.474297	0.473746	0.460968	0.722121	2.371231e+05
min	100002.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	2.565000e+04
25%	189145.500000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.125000e+05
50%	278202.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	1.471500e+05
75%	367142.500000	0.000000	0.000000	1.000000	1.000000	1.000000	1.000000	2.025000e+05
max	456255.000000	1.000000	1.000000	2.000000	1.000000	1.000000	19.000000	1.170000e+08

8 rows × 132 columns

Εικόνα 75. Κατανομή των δεδομένων

Για να μπορέσει να η βιβλιοθήκη XGBoost να λειτουργήσει σωστά, θα πρέπει όπου υπάρχουν τιμές inf και -inf να μετατραπούν σε τιμές NaN.

```
orig_data.replace([np.inf, -np.inf], np.nan, inplace=True)
orig_data.describe()
```



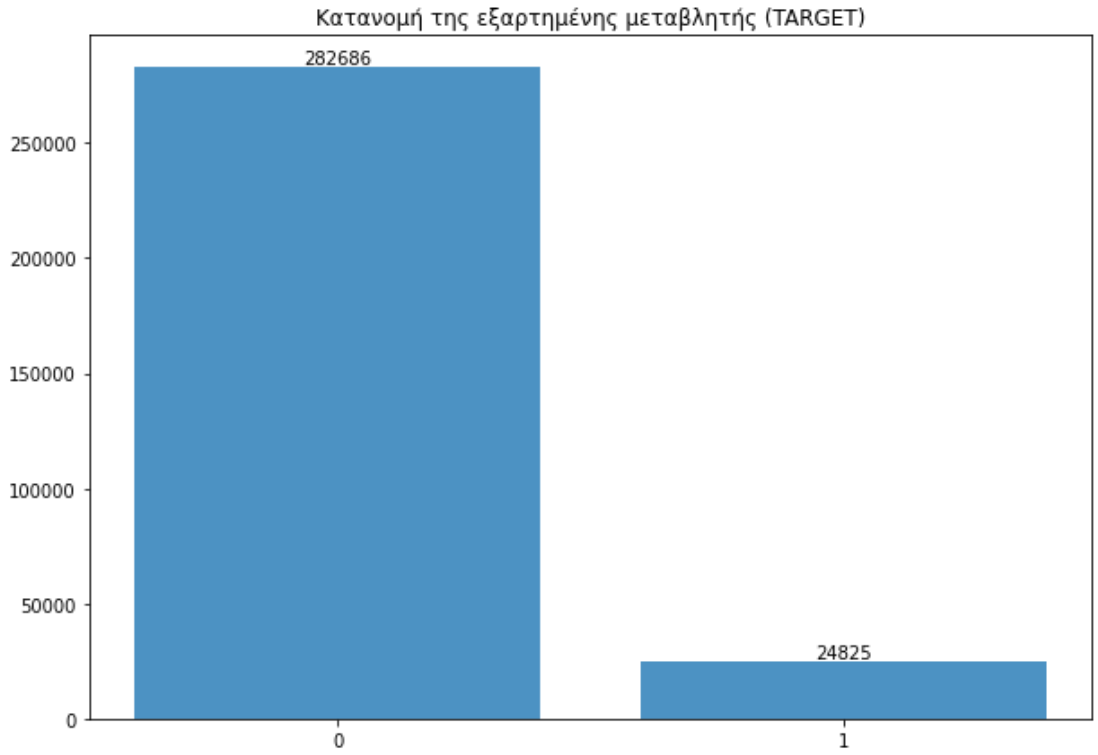
	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL
count	307511.000000	307511.000000	307511.000000	307511.000000	307511.000000	307511.000000	307511.000000	3.075110e+05
mean	278180.518577	0.080729	0.095213	0.341669	0.340108	0.693673	0.417052	1.687979e+05
std	102790.175348	0.272419	0.293509	0.474297	0.473746	0.460968	0.722121	2.371231e+05
min	100002.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	2.565000e+04
25%	189145.500000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.125000e+05
50%	278202.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	1.471500e+05
75%	367142.500000	0.000000	0.000000	1.000000	1.000000	1.000000	1.000000	2.025000e+05
max	456255.000000	1.000000	1.000000	2.000000	1.000000	1.000000	19.000000	1.170000e+08

8 rows × 132 columns

Εικόνα 76. Μετατροπή τιμών inf και -inf σε τιμές NaN

Παρακάτω, ακολουθεί μία οπτικοποίηση της κατανομής της μεταβλητής εξόδου TARGET σε ένα διάγραμμα ράβδων, με τη χρήση της βιβλιοθήκης matplotlib.

```
class_sep=orig_data['TARGET'].value_counts().reset_index()
fig, ax = plt.subplots()
fig.set_size_inches([10,7])
p1=ax.bar(class_sep['index'],class_sep['TARGET'],alpha=0.8)
ax.set_xticks(class_sep['index'])
ax.bar_label(p1)
plt.title("Κατανομή της εξαρτημένης μεταβλητής (TARGET)")
plt.show()
```



Εικόνα 77. Κατανομή της εξαρτημένης μεταβλητής (TARGET)

Για τη δυαδική μεταβλητή TARGET:

- όπου 1: Πελάτης με δυσκολία πληρωμής και
- όπου 0: Όλες οι άλλες περιπτώσεις.

Τα ακριβή ποσοστά των δανείων με δυσκολία πληρωμής ή όχι υπολογίζονται με τον ακόλουθο κώδικα.

```
print("Τα ακριβή ποσοστά δανείων με δυσκολία πληρωμής ή όχι:")
orig_data['TARGET'].value_counts(normalize=True)
```

```
Τα ακριβή ποσοστά δανείων με δυσκολία πληρωμής ή όχι:


0    0.919271
1    0.080729
Name: TARGET, dtype: float64
```

Εικόνα 78. Τα ακριβή ποσοστά των δανείων με δυσκολία πληρωμής ή όχι

9.2. Διαχωρισμός δεδομένων

Θα πρέπει να διαχωριστούν οι παράμετροι πρόβλεψης από τη μεταβλητή στόχου, ώστε να φτάσουμε στα μοντέλα κατασκευής. Η δεύτερη στήλη του συνόλου δεδομένων είναι η μεταβλητή στόχου TARGET ενώ οι υπόλοιπες οι παράμετροι πρόβλεψης. Ωστόσο,

θα αφαιρεθεί η πρώτη στήλη SK_ID_CURR από τις παραμέτρους καθώς δεν έχει προγνωστική ισχύ.

```
X, y = orig_data.iloc[:,2:], orig_data.iloc[:,1]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.3,
random_state=123)
```

Οπότε, προκύπτει το σύνολο δεδομένων «X» με τις μεταβλητές πρόβλεψης μετά το διαχωρισμό και περιλαμβάνει 130 στήλες.

```
X.head()
```

	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	AMT_GOODS_
0	0	1	0	1	0	202500.0	406597.5	24700.5	351000.0
1	0	0	0	0	0	270000.0	1293502.5	35698.5	1129500.0
2	1	1	1	1	0	67500.0	135000.0	6750.0	135000.0
3	0	0	0	1	0	135000.0	312682.5	29686.5	297000.0
4	0	1	0	1	0	121500.0	513000.0	21865.5	513000.0

5 rows × 130 columns

Εικόνα 79. το σύνολο δεδομένων με τις μεταβλητές πρόβλεψης μετά το διαχωρισμό

Ξεχωριστά, μετά το διαχωρισμό, υπάρχει το σύνολο δεδομένων «Y», μόνο με τη μεταβλητή στόχου.

```
y.head()
```

```
0    1
1    0
2    0
3    0
4    0
Name: TARGET, dtype: int64
```

Εικόνα 80. το σύνολο δεδομένων «X» με τη μεταβλητή στόχου μετά το διαχωρισμό

9.3. Δημιουργία του μοντέλου XGBoost

9.3.1. Συντονισμός υπερπαραμέτρων με την Αναζήτηση Πλέγματος (Grid Search)

Στη μηχανική μάθηση, η βελτιστοποίηση ή η ρύθμιση υπερπαραμέτρων είναι το πρόβλημα της επιλογής ενός συνόλου βέλτιστων υπερπαραμέτρων για έναν αλγόριθμο εκμάθησης. Μια υπερπαράμετρος είναι μία παράμετρος της οποίας η τιμή χρησιμοποιείται για τον έλεγχο της μαθησιακής διαδικασίας.

Μια μεθοδολογία για την επιλογή των σωστών υπερπαραμέτρων είναι η Αναζήτηση Πλέγματος (Grid Search). Πρόκειται για μία τεχνική συντονισμού που

επιχειρεί να υπολογίσει τις βέλτιστες τιμές των υπερπαραμέτρων. Είναι μια εξαντλητική αναζήτηση που εκτελείται σε συγκεκριμένες τιμές των παραμέτρων ενός μοντέλου. Το μοντέλο είναι επίσης γνωστό ως εκτιμητής.

Μία λήψη ενός στιγμιότυπου με τον XGBClassifier και η επιθεώρηση των παραμέτρων, εμφανίζεται αμέσως παρακάτω.

```
xgb_clf = xgb.XGBClassifier(random_state=123)
xgb_clf.get_params()
```

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

Εικόνα 81. Λήψη στιγμιότυπου με τον XGBClassifier και επιθεώρηση των παραμέτρων

9.3.2. Δημιουργία του μοντέλου με την Αναζήτηση Πλέγματος (Grid Search)

Οι προεπιλεγμένες τιμές παραμέτρων της Αναζήτησης Πλέγματος είναι:

- "n_estimators":[100]

- "max_depth":[6]
- "learning_rate":[0.30]
- "scale_pos_weight" = 11.38715005035247 που προκύπτει από το λόγο sum(negative instances) / sum(positive instances) δηλαδή "πλήθος των 0 / πλήθος των 1").

Καθώς ο υπολογισμός των βέλτιστων τιμών των παραμέτρων με την Αναζήτηση Πλέγματος είναι, όπως προαναφέρθηκε, μια εξαντλητική αναζήτηση, αυτός δεν είναι δυνατό να γίνει μόνο με τη χρήση κεντρικής μονάδας επεξεργασίας (CPU) της πλατφόρμας της Kaggle.

Η μέγιστη διάρκεια συνεδρίας που προσφέρει η πλατφόρμα της Kaggle είναι 12 ώρες, ενώ στην πραγματικότητα η Αναζήτηση Πλέγματος απαιτεί περισσότερο χρόνο. Συνεπώς, ο υπολογισμός των βέλτιστων τιμών των παραμέτρων μπορεί να γίνει χειροκίνητα θέτοντας συγκεκριμένες μεμονωμένες τιμές για αυτές.

Η Kaggle όμως δίνει τη δυνατότητα χρήσης μονάδας επεξεργασίας γραφικών (GPU) παράλληλα με τη CPU. Έτσι, οι ταχύτητες εκτέλεσης του κώδικα με παράλληλη χρήση CPU και GPU, αυξάνονται εκπληκτικά. Στην περίπτωση αυτή, είναι δυνατό να δοκιμαστούν ταυτόχρονα συνδυασμοί πολλών τιμών για τις υπερπαραμέτρους, ακόμη και μέσα στο χρονικό όριο συνεδρίας που επιτρέπει η Kaggle.

Επιλέχθηκαν κάποιες τιμές για μερικές από τις παραμέτρους, και για συνδυασμούς των τιμών αυτών υπολογίστηκαν τα αποτελέσματα πρόβλεψης.

Εύρεση καλύτερης τιμής για "n_estimators"

Για την παράμετρο "n_estimators" επιλέχθηκαν για δοκιμή οι τιμές 100, 600 και 1000.

Εύρεση καλύτερης τιμής για "max_depth"

Για την παράμετρο "max_depth", επιλέχθηκαν για δοκιμή οι τιμές 3, 4 και 5.

Εύρεση καλύτερης τιμής για "learning_rate"

Για την παράμετρο "learning_rate", επιλέχθηκαν για δοκιμή οι τιμές 0.01, 0.1, και 0.3.

```
import xgboost as xgb
from xgboost import XGBClassifier
from sklearn.model_selection import RandomizedSearchCV,
StratifiedKFold, GridSearchCV
from sklearn.metrics import accuracy_score, f1_score,
classification_report, confusion_matrix
xgb_param_grid={"n_estimators":[100,600,1000],
                "max_depth":[3,4,5],
                "learning_rate":[0.01,0.1,0.3],
                "scale_pos_weight":[11.38715005035247]}
```

```
xgb = XGBClassifier(objective="binary:logistic", eval_metric="auc",
use_label_encoder=False,random_state=123,tree_method='gpu_hist',
gpu_id=0)
cv_f=StratifiedKFold(n_splits=4,shuffle=True)
rand_search = GridSearchCV(estimator=xgb, param_grid=xgb_param_grid,
n_jobs=2, cv=cv_f, verbose=1, scoring='roc_auc')
rand_search.fit(X_train,y_train)
preds=rand_search.predict(X_test)
print("Παρακάτω παρουσιάζεται η έκθεση ταξινόμησης (classification
report)")
print(classification_report(y_test, preds))
print("Μια αρχική παρουσίαση του πίνακα σύγχυσης (confusion matrix)")
print(confusion_matrix(y_test, preds))
print("Best parameters found:", rand_search.best_params_)
print("Best score found:", rand_search.best_score_)
```

```
Fitting 4 folds for each of 27 candidates, totalling 108 fits
Παρακάτω παρουσιάζεται η έκθεση ταξινόμησης (classification report)
              precision    recall  f1-score   support

           0       0.96      0.72      0.82     84717
           1       0.18      0.68      0.28      7537

    accuracy                           0.72     92254
   macro avg       0.57      0.70      0.55     92254
weighted avg       0.90      0.72      0.78     92254

Μια αρχική παρουσίαση του πίνακα σύγχυσης (confusion matrix)
[[61098 23619]
 [ 2445  5092]]
Best parameters found: {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 600, 'scale_po
s_weight': 11.38715005035247}
Best score found: 0.7628310122864883
```

Εικόνα 82. Έκθεση ταξινόμησης (Classification report)

Μετά από συνδυαστικές δοκιμές, προέκυψε καλύτερο αποτέλεσμα πρόβλεψης με σκορ 0.763, με τις εξής καλύτερες τιμές:

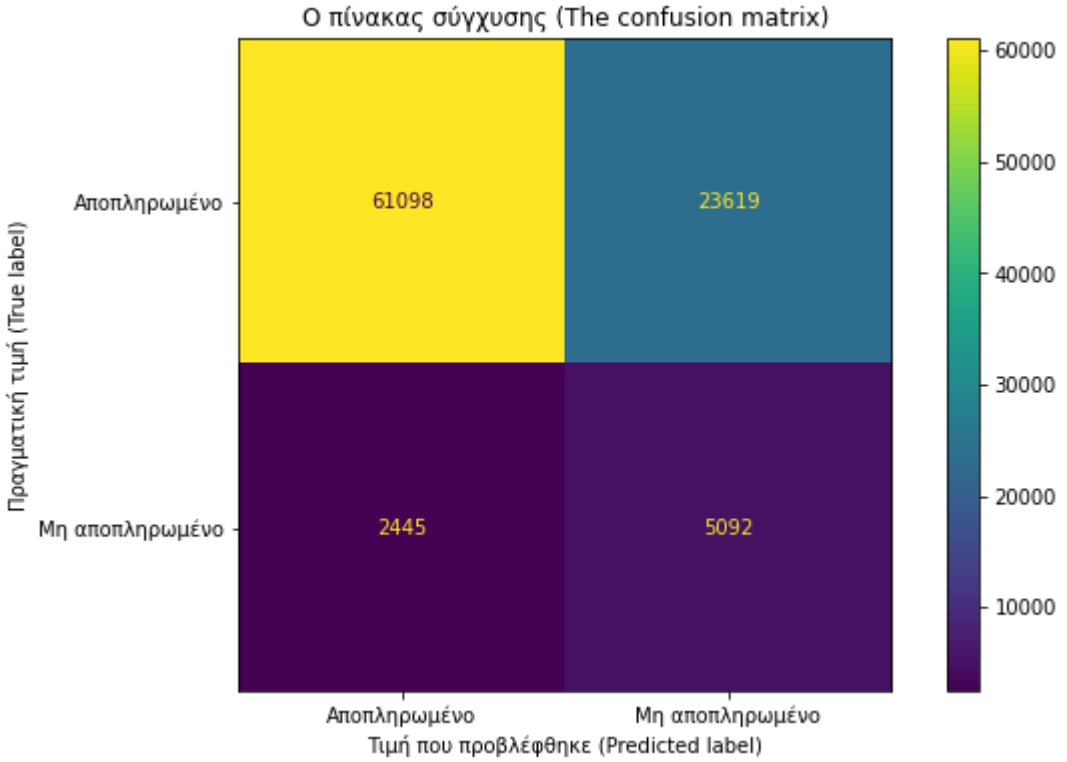
- "n_estimators":[600]
- "max_depth":[3]
- "learning_rate":[0.1]

9.3.3. Πίνακας σύγχυσης (Confusion matrix)

Ο πίνακας σύγχυσης (Confusion matrix) είναι μία οπτική παράσταση των προβλέψεων. Πιο συγκεκριμένα, απεικονίζει τις τιμές που:

- Ορθώς προβλέφθηκαν ως θετικές (True positive)
- Ορθώς προβλέφθηκαν ως αρνητικές (True negative)
- Εσφαλμένα προβλέφθηκαν ως θετικές (False positive)
- Εσφαλμένα προβλέφθηκαν ως αρνητικές (False negative)

```
import matplotlib.pyplot as plt
from sklearn.datasets import make_classification
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
cm=confusion_matrix(y_test, preds, labels=rand_search.classes_)
disp=ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=['Αποπληρωμένο','Μη αποπληρωμένο'])
fig, ax = plt.subplots(figsize=(10,6))
disp.plot(ax=ax)
plt.title("Ο πίνακας σύγχυσης (The confusion matrix)")
plt.ylabel("Πραγματική τιμή (True label)")
plt.xlabel("Τιμή που προβλέφθηκε (Predicted label)")
plt.show()
```



Εικόνα 83. Πίνακας σύγχυσης (Confusion matrix)

Το σύνολο δεδομένων που χρησιμοποιείται για την εκπαίδευση (train dataset) έχει 92.254 τιμές. Οι 23.619 από αυτές ταξινομούνται λανθασμένα ως θετικές ενώ είναι αρνητικές (FP) και οι 2.445 από αυτές ταξινομούνται ως αρνητικές ενώ είναι θετικές (FN).

9.3.4. Η καμπύλη ROC (Receiver operating characteristic)

Στα προβλήματα ταξινόμησης, υπάρχουν πολλές διαφορετικές μετρήσεις αξιολόγησης. Οι καμπύλες ROC τυπικά παρουσιάζουν το ρυθμό των ορθώς θετικών (True positives) στον άξονα Y και το ρυθμό των ψευδώς θετικών (False positives) στον άξονα X.

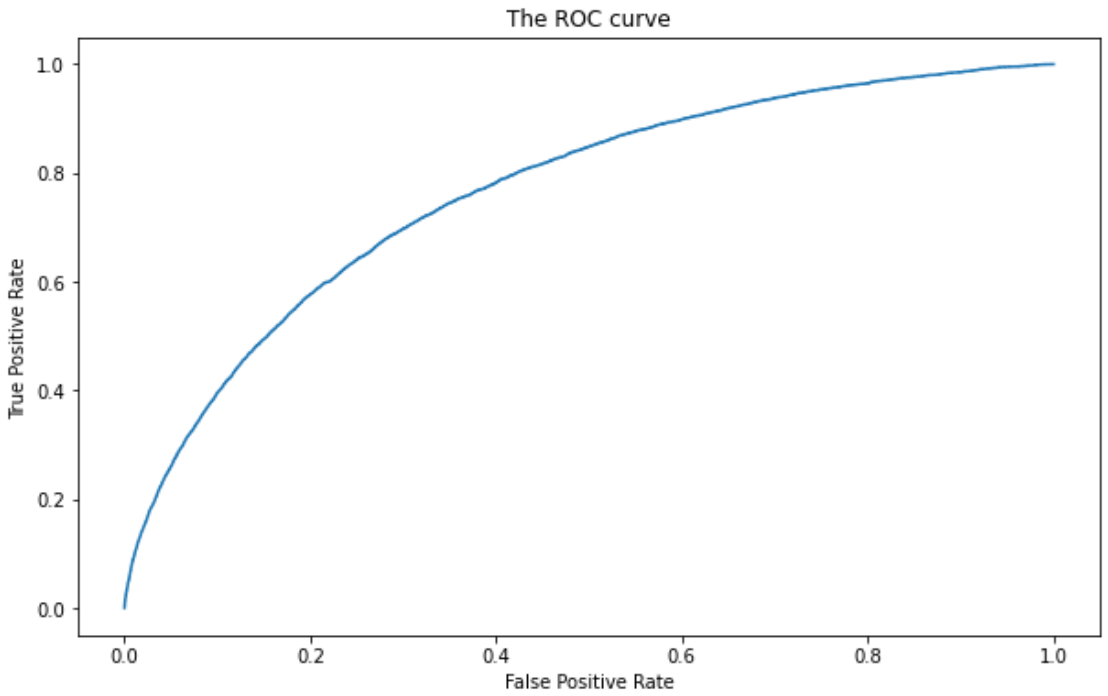
Συνεπώς, η επάνω αριστερή γωνία του γραφήματος είναι το «ιδανικό» σημείο αφού δηλώνει μηδενικό ρυθμό ψευδώς θετικών (False positives) και ένα ρυθμό ορθώς θετικών (True positives).

Για να σχεδιαστεί η καμπύλη ROC, θα πρέπει να εισαχθούν οι απαραίτητες συναρτήσεις βιβλιοθηκών και να οριστούν οι μετρικές προς εμφάνιση.

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

Έπειτα, είναι δυνατό να δημιουργηθεί και να σχεδιαστεί η καμπύλη ROC (Receiver operating characteristic).

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



Εικόνα 84. Η καμπύλη ROC (Receiver operating characteristic)

Συνεπώς, η επάνω αριστερή γωνία του γραφήματος είναι το «ιδανικό» σημείο αφού δηλώνει μηδενικό ρυθμό ψευδώς θετικών (False positives) και ένα ρυθμό ορθώς θετικών (True positives).

9.3.5. Η καμπύλη AUC (Area under the ROC Curve)

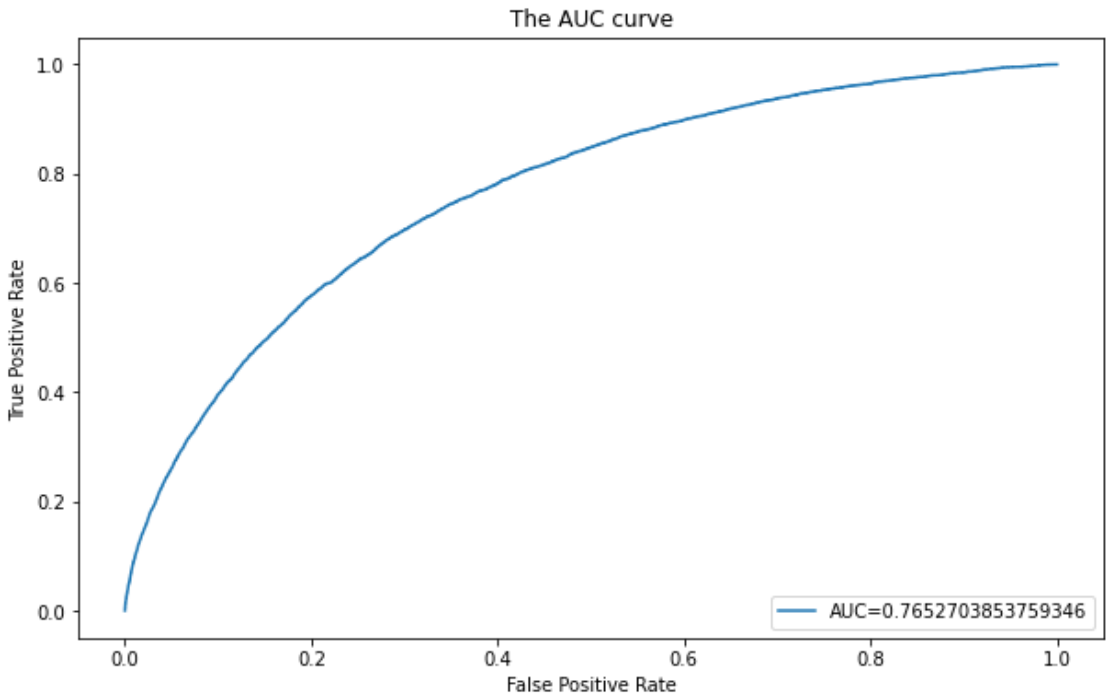
Μια άλλη συνηθισμένη μέτρηση στα προβλήματα ταξινόμησης, είναι η καμπύλη AUC. Στην πραγματικότητα, είναι η περιοχή κάτω από την καμπύλη ROC. Όσο μεγαλύτερη είναι η περιοχή κάτω από την καμπύλη (AUC), τόσο πιο καλό είναι το αποτέλεσμα..

Για να σχεδιαστεί η καμπύλη AUC, θα πρέπει να οριστούν πρώτα οι μετρικές προς εμφάνιση.

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

Έπειτα, είναι δυνατό να δημιουργηθεί και να σχεδιαστεί η καμπύλη AUC.

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



Εικόνα 85. Η καμπύλη AUC (Area under the ROC Curve)

9.3.6. Διάγραμμα σπουδαιότητας βάσει βαρύτητας (weight)

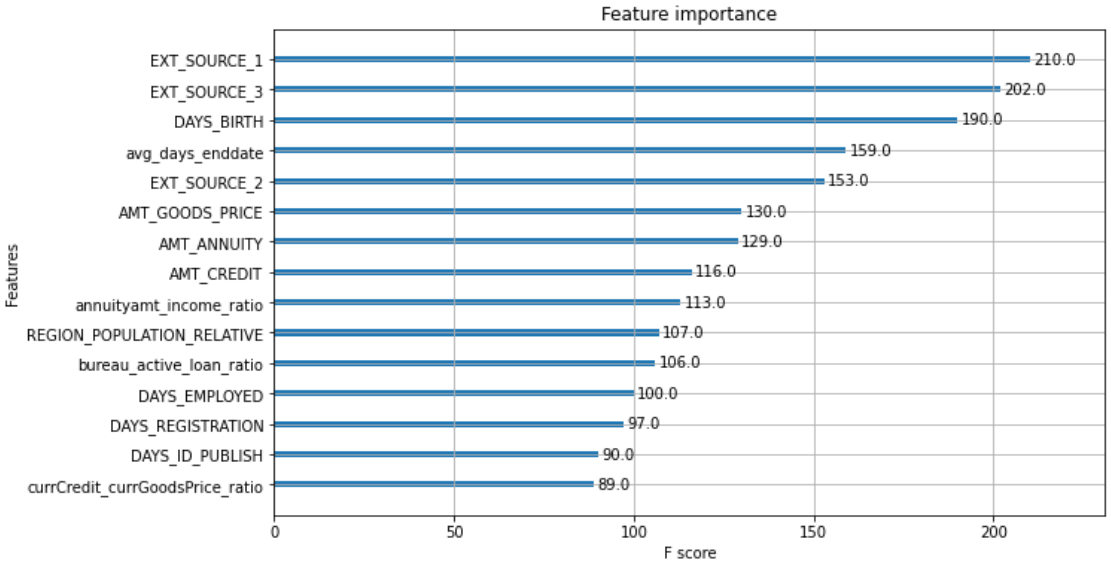
Η βιβλιοθήκη XGBoost, διαθέτει μία συνάρτηση ικανή να προβάλει σε διάγραμμα τα πιο σημαντικά χαρακτηριστικά κατά τη δημιουργία του μοντέλο μηχανικής μάθησης.

Για να σχεδιαστεί το διάγραμμα σπουδαιότητας των χαρακτηριστικών, έπρεπε να επαναληφθεί το μοντέλο καθώς η αναζήτηση πλέγματος δημιούργησε μια δομή δεδομένων που δεν ήταν προσβάσιμη από τη συνάρτηση plot_importance. Για το λόγο αυτό, δημιουργήθηκε μια σύντομη απεικόνιση του μοντέλου.

Η σπουδαιότητα των χαρακτηριστικών είναι δυνατό να υπολογιστεί χρησιμοποιώντας τη βαρύτητα που είναι ο αριθμός των φορών που εμφανίζεται ένα χαρακτηριστικό σε ένα δέντρο απόφασης.

```
from xgboost import plot_importance
xgb_cl = rand_search.best_estimator_
fig, ax = plt.subplots(figsize=(10, 6))
```

```
plot_importance(xgb_cl, importance_type='weight', max_num_features=15,
ax=ax)
plt.show()
```

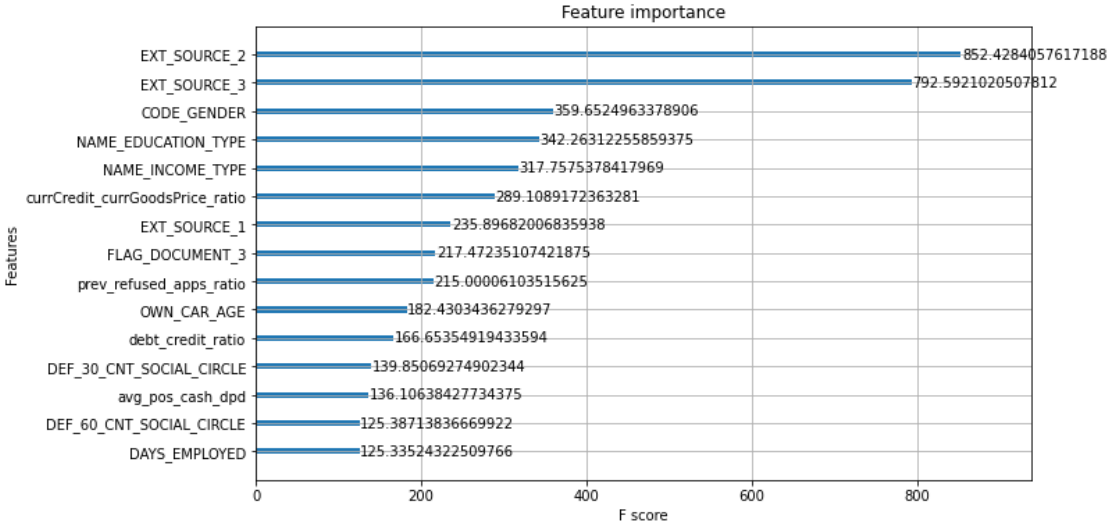


Εικόνα 86. Σπουδαιότητα χαρακτηριστικών βάσει βαρύτητας (weight)

9.3.7. Διάγραμμα σπουδαιότητας βάσει κέρδους (gain)

Η σπουδαιότητα όμως των χαρακτηριστικών δύναται να υπολογιστεί βάσει του κέρδους (το μέσο κέρδος των διαχωρισμών που χρησιμοποιούν το χαρακτηριστικό).

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



Εικόνα 87. Σπουδαιότητα χαρακτηριστικών βάσει κέρδους (gain)

10. Συμπεράσματα

Το πρόβλημα ήταν να προβλεφθεί τα δάνεια ποιων αιτήσεων θα αποπληρωνόντουσαν ή όχι. Ήταν ένα **πρόβλημα δυαδικής ταξινόμησης** (classification problem).

Δεδομένου ότι τα **δεδομένα** ήταν αρκετά **μη ισορροπημένα** (imbalanced data), επιλέχθηκε να εφαρμοστεί ο αλγόριθμος XGBoost. Ο XGBoost είναι επί του παρόντος ένας από τους πιο ευρέως χρησιμοποιούμενους αλγόριθμους μηχανικής μάθησης για προβλήματα δυαδικής ταξινόμησης και παλινδρόμησης. Επιπλέον, ο αλγόριθμος αυτός έχει την υπερπαράμετρο scale_pos_weight που είναι εξαιρετικά χρήσιμη όταν τα δεδομένα είναι μη ισορροπημένα.

Εξετάζοντας το διάγραμμα σπουδαιότητας χαρακτηριστικών με βάση τη βαρύτητα, γίνεται ξεκάθαρα κατανοητό ότι οι μεταβλητές EXT_SOURCE_3, EXT_SOURCE_2 και DAYS_BIRTH είναι οι τρεις πρώτες από αυτές που εμφανίζονται πολλές φορές στα δέντρα αποφάσεων. Από το ίδιο διάγραμμα, διακρίνεται πως στα 15 καλύτερα χαρακτηριστικά υπάρχουν και 4 που δεν υπήρχαν στα αρχικά σύνολα δεδομένων. Αυτά τα χαρακτηριστικά, δημιουργήθηκαν κατά την προ-επεξεργασία των δεδομένων. Αυτά τα χαρακτηριστικά είναι τα:

- **avg_days_enddate**: Μέσος όρος ημερών που λήγουν τα προηγούμενα δάνεια του κάθε πελάτη στο μέλλον
- **annuityamt_income_ratio**: Λόγος ετήσιου ποσού καταβολής δανείου προς ετήσιο εισόδημα πελάτη
- **bureau_active_loan_ratio**: Λόγος ενεργών προς συνολικά προηγούμενα δάνεια
- **currCredit_currGoodsPrice_ratio**: Λόγος ποσού αιτούμενου δανείου προς κόστος αγαθών

Από την άλλη πλευρά, το διάγραμμα σπουδαιότητας χαρακτηριστικών με βάση το κέρδος, δίνει μια άλλη σημαντική πληροφορία. Απεικονίζει ότι οι μεταβλητές EXT_SOURCE_2, EXT_SOURCE_3 και CODE_GENDER είναι τα τρία κύρια χαρακτηριστικά που φέρνουν τη μεγαλύτερη βελτίωση στην ακρίβεια των αποτελεσμάτων. Και σε αυτό το διάγραμμα, είναι εμφανές (καθώς ονοματίζονται με πεζούς χαρακτήρες) πως στα 15 καλύτερα χαρακτηριστικά, υπάρχουν 4 που δημιουργήθηκαν στα πλαίσια της παρούσας διπλωματικής εργασίας. Αυτά είναι τα:

- **currCredit_currGoodsPrice_ratio**: Λόγος ποσού αιτούμενου δανείου προς κόστος αγαθών
- **prev_refused_apps_ratio**: Λόγος απορριπτέων προηγούμενων αιτήσεων δανείων προς συνολικές προηγούμενες αιτήσεις στη Home Credit
- **debt_credit_ratio**: Λόγος χρέους προς πίστωση ανά πελάτη για προηγούμενα δάνεια
- **avg_pos_cash_dpd**: Μέσος όρος ημερών καθυστέρησης πληρωμής δόσεων δανείων σε μετρητά

Στο πρόβλημα αυτό, επιλέχθηκε να χρησιμοποιηθεί το AUC ROC score για την αξιολόγηση του μοντέλου μηχανικής μάθησης που δημιουργήθηκε. Αν και είναι αρκετά διαδεδομένη μετρική για προβλήματα δυαδικής ταξινόμησης, έδωσε ένα score 0,765. Το score αν και εκ πρώτης όψεως δεν φαίνεται και τόσο καλό, αξίζει να σημειωθεί πως στο διαγωνισμό που διενεργήθηκε στο Kaggle, ο πρώτος νικητής είχε ένα score 0.8057. Αυτό δηλώνει πως στην παρούσα διπλωματική εργασία επιτεύχθηκε μεν ένα καλό score, έχει δε περιθώρια βελτίωσης.

Στην παρούσα μελέτη περίπτωσης, πιθανόν θα ήταν καλύτερο να μετρηθεί το Precision-Recall curve AUC. Το Precision-Recall, είναι ένα χρήσιμο μέτρο για την επιτυχία της πρόβλεψης όταν τα δεδομένα είναι πολύ μη ισορροπημένα. Στην ανάκτηση πληροφοριών, η ακρίβεια (Precision) είναι ένα μέτρο της συνάφειας των αποτελεσμάτων, ενώ η ανάκληση (Recall) είναι ένα μέτρο για το πόσα πραγματικά σχετικά αποτελέσματα επιστρέφονται.

Η καμπύλη ακριβείας - ανάκλησης (Precision-Recall curve) δείχνει την αντιστάθμιση μεταξύ ακρίβειας και ανάκλησης για διαφορετικά όρια. Μια υψηλή περιοχή κάτω από την καμπύλη, αντιπροσωπεύει τόσο υψηλή ανάκληση όσο και υψηλή ακρίβεια, όπου η υψηλή ακρίβεια σχετίζεται με ποσοστό χαμηλών ψευδώς θετικών και η υψηλή ανάκληση σχετίζεται με το ποσοστό χαμηλών ψευδώς αρνητικών. Οι υψηλές βαθμολογίες και για τα δύο, δείχνουν ότι ο ταξινομητής επιστρέφει ακριβή αποτελέσματα (υψηλή ακρίβεια), καθώς και την πλειονότητα όλων των θετικών αποτελεσμάτων (υψηλή ανάκληση).

Ένα σύστημα με υψηλή ανάκληση αλλά χαμηλή ακρίβεια επιστρέφει πολλά αποτελέσματα, αλλά οι περισσότερες από τις προβλεπόμενες ετικέτες του είναι λανθασμένες σε σύγκριση με τις ετικέτες εκπαίδευσης. Ένα σύστημα με υψηλή ακρίβεια αλλά χαμηλή ανάκληση είναι ακριβώς το αντίθετο, επιστρέφοντας πολύ λίγα αποτελέσματα, αλλά οι περισσότερες από τις προβλεπόμενες ετικέτες του είναι σωστές σε σύγκριση με τις ετικέτες εκπαίδευσης. Ένα ιδανικό σύστημα με υψηλή ακρίβεια και υψηλή ανάκληση θα επιστρέψει πολλά αποτελέσματα, με όλα σχεδόν τα αποτελέσματα να επισημαίνονται σωστά.

Η ακρίβεια (P), ορίζεται ως ο αριθμός των ορθώς θετικών (TP) έναντι του αριθμού των ορθώς θετικών συν τον αριθμό των ψευδώς θετικών (FP).

$$P = \frac{T_p}{T_p + F_p}$$

Η ανάκληση (R), ορίζεται ως ο αριθμός των ορθώς θετικών (TP) έναντι του αριθμού των ορθώς θετικών συν τον αριθμό των ψευδώς αρνητικών (FN).

$$R = \frac{T_p}{T_p + F_n}$$

Αυτές οι ποσότητες σχετίζονται επίσης με τη βαθμολογία (F1), η οποία ορίζεται ως ο αρμονικός μέσος όρος ακρίβειας και ανάκλησης.

$$F1 = 2\frac{P \times R}{P + R}$$

Γενικότερα, τα συμπεράσματα που προκύπτουν σχετικά με τα προβλήματα για των οποίων τη λύση απαιτείται η μηχανική μάθηση, έχουν σαφέστατα εκρεύσει από την τριβή με τη μελέτη περίπτωσης της συγκεκριμένης εργασίας.

Πρωτίστως, χρειάζονται δεδομένα. Όσο πιο πολλά, τόσο καλύτερα. **Μεγάλα δεδομένα**. Οι αλγόριθμοι μηχανικής μάθησης τα χρειάζονται για να μάθουν με μεγαλύτερη ακρίβεια να προβλέψουν αυτό που τους ζητείται. Όσες περισσότερες παπαρούνες δει ένα μικρό παιδί, τόσο πιο εύκολα θα την ξεχωρίζει ανάμεσα σε μία ομάδα λουλουδιών. Στη σημερινή εποχή της άνθισης της τεχνολογίας και των επικοινωνιών, τα πάντα δημιουργούν δεδομένα. Και με τη θεμελίωση του 5G, η μεταφορά και αποθήκευση των δεδομένων θα είναι ευκολότερη. Οπότε, κάθε επόμενη εποχή θα είναι ακόμη πιο κοντά στα μεγάλα δεδομένα.

Έπειτα, αυτό που παίζει πολύ σημαντικό ρόλο, είναι η **γνωριμία με τα δεδομένα**. Πρέπει να τα αναγνωστούν, να μελετηθούν και να κατανοηθούν. Μόνον έτσι είναι εφικτό να αναλυθούν σωστά και να βρεθούν πιθανά μοτίβα σε αυτά.

Επιπρόσθετα, **τα δεδομένα πρέπει να επεξεργαστούν** κατάλληλα. Να καθαριστούν από μη χρήσιμα που πιθανόν δεν έχουν προβλεπτική ισχύ. Ακόμη, να μετατραπούν, όπου καταστεί δυνατό, ώστε να είναι εύκολα αναγνώσιμα από τους αλγόριθμους της μηχανικής μάθησης.

Σε κάθε περίπτωση, σημαντική φαίνεται πως είναι η **ομαδοποίηση των δεδομένων**, η οποία συχνά οδηγεί στη **δημιουργία νέων παραμέτρων**. Παράμετροι που δεν υπήρχαν στα αρχικά μεγάλα δεδομένα αλλά μπορεί να παίξουν μεγάλο ρόλο για ένα ακριβές μοντέλο μηχανικής μάθησης.

Η μηχανική μάθηση απαιτεί **προσοχή**, **αφοσίωση** και **μεθοδικότητα**. Είναι μία διαδικασία επίπονη, χρονοβόρα και επαναληπτική. Μέσα από την επανάληψη θα μπορέσει να εξαχθεί ένα καλύτερο τελικό σύνολο δεδομένων. Μέσα από την επανάληψη θα βρεθεί ο καλύτερος αλγόριθμος που πρέπει να χρησιμοποιηθεί. Μέσα από την επανάληψη θα ανακαλυφθούν οι καλύτερες τιμές για τις υπερπαραμέτρους των αλγορίθμων μηχανικής μάθησης. Έτσι, ο κάθε αλγόριθμος θα μπορέσει με ακρίβεια να προβλέψει αυτό το οποίο θα του ζητηθεί.

Η μηχανική μάθηση δεν απαιτεί μόνο δεδομένα και ανθρώπινο μόχθο. Απαιτεί **ισχυρή υπολογιστική ισχύ** για να μπορέσει να ανταπεξέλθει ταχέως στους υπολογισμούς που καλείται να κάνει. Περισσότερο σημαντικό ρόλο ανακαλύφθηκε πως παίζουν οι επεξεργαστικές μονάδες (CPU, GPU) παρά η υπολογιστική μνήμη RAM.

Όσα ένας άνθρωπος δεν μπορεί να καταφέρει μέσα σε έτη, με τη βοήθεια της μηχανικής μάθησης πιθανόν να μπορέσει να το καταφέρει μέσα σε μήνες.

Παράρτημα Α' - Κώδικας

```
import pandas as pd
bureau = pd.read_csv('../input/home-credit-default-risk/bureau.csv')
bureau_loan_count =
bureau.groupby('SK_ID_CURR')['DAYS_CREDIT'].count().to_frame('bureau_lo
an_count').reset_index()
bureau_loan_types =
bureau.groupby('SK_ID_CURR')['CREDIT_TYPE'].nunique().to_frame('bureau_
loan_types').reset_index()
bureau_active_loan_ratio = bureau[['SK_ID_CURR', 'CREDIT_ACTIVE']]
bureau_active_loan_ratio =
bureau_active_loan_ratio[bureau_active_loan_ratio.CREDIT_ACTIVE ==
'Active']
bureau_active_loan_ratio =
bureau_active_loan_ratio.groupby('SK_ID_CURR')['CREDIT_ACTIVE'].count()
.to_frame('bureau_active_loan_count').reset_index()
bureau_loan_count =
bureau.groupby('SK_ID_CURR')['DAYS_CREDIT'].count().to_frame('bureau_lo
an_count').reset_index()
bureau_active_loan_ratio =
bureau_active_loan_ratio.merge(bureau_loan_count, on='SK_ID_CURR',
how='left')
bureau_active_loan_ratio['bureau_active_loan_ratio'] =
bureau_active_loan_ratio['bureau_active_loan_count']/bureau_active_loan
_ratio['bureau_loan_count']
del bureau_active_loan_ratio['bureau_active_loan_count'],
bureau_active_loan_ratio['bureau_loan_count']
bureau_avg_enddate_future = bureau[['SK_ID_CURR',
'DAYS_CREDIT_ENDDATE']].reset_index(drop = True)
bureau_avg_enddate_future['days_credit_enddate_binary'] =
bureau_avg_enddate_future['DAYS_CREDIT_ENDDATE']
def f(x):
    if x<0:
        y = 0
    else:
        y = 1
    return y
bureau_avg_enddate_future['days_credit_enddate_binary'] =
bureau_avg_enddate_future.apply(lambda x: f(x.DAYS_CREDIT_ENDDATE),
axis = 1)
bureau_avg_enddate_future =
bureau_avg_enddate_future[bureau_avg_enddate_future['days_credit_enddat
e_binary'] == 1]
bureau_avg_enddate_future =
bureau_avg_enddate_future[bureau_avg_enddate_future['DAYS_CREDIT_ENDDAT
E'].notnull()]
```

```
bureau_avg_enddate_future = bureau_avg_enddate_future.groupby(by =
['SK_ID_CURR'])['DAYS_CREDIT_ENDDATE'].mean().to_frame('avg_days_enddat
e').reset_index()
dept_credit_ratio = bureau[['SK_ID_CURR', 'AMT_CREDIT_SUM',
'AMT_CREDIT_SUM_DEBT']].reset_index(drop = True)
dept_credit_ratio['AMT_CREDIT_SUM'] =
dept_credit_ratio['AMT_CREDIT_SUM'].fillna(0)
dept_credit_ratio['AMT_CREDIT_SUM_DEBT'] =
dept_credit_ratio['AMT_CREDIT_SUM_DEBT'].fillna(0)
dept_credit_ratio_A = dept_credit_ratio.groupby(by =
['SK_ID_CURR'])['AMT_CREDIT_SUM'].sum().to_frame('total_credit').reset_
index()
dept_credit_ratio_B = dept_credit_ratio.groupby(by =
['SK_ID_CURR'])['AMT_CREDIT_SUM_DEBT'].sum().to_frame('total_debt').res
et_index()
dept_credit_ratio = dept_credit_ratio.merge(dept_credit_ratio_A, on =
['SK_ID_CURR'], how = 'left')
dept_credit_ratio = dept_credit_ratio.merge(dept_credit_ratio_B, on =
['SK_ID_CURR'], how = 'left')
dept_credit_ratio['debt_credit_ratio'] =
dept_credit_ratio['total_debt']/dept_credit_ratio['total_credit']
del dept_credit_ratio_A, dept_credit_ratio_B
del dept_credit_ratio['AMT_CREDIT_SUM'],
dept_credit_ratio['AMT_CREDIT_SUM_DEBT'],
dept_credit_ratio['total_credit'], dept_credit_ratio['total_debt']
dept_credit_ratio = dept_credit_ratio.groupby(by =
['SK_ID_CURR'])['debt_credit_ratio'].mean().to_frame('debt_credit_ratio
').reset_index()
overdue_dept_ratio = bureau[['SK_ID_CURR', 'AMT_CREDIT_SUM_DEBT',
'AMT_CREDIT_SUM_OVERDUE']].reset_index(drop = True)
overdue_dept_ratio['AMT_CREDIT_SUM_DEBT'] =
overdue_dept_ratio['AMT_CREDIT_SUM_DEBT'].fillna(0)
overdue_dept_ratio['AMT_CREDIT_SUM_OVERDUE'] =
overdue_dept_ratio['AMT_CREDIT_SUM_OVERDUE'].fillna(0)
overdue_dept_ratio_A = overdue_dept_ratio.groupby(by =
['SK_ID_CURR'])['AMT_CREDIT_SUM_DEBT'].sum().to_frame('total_debt').res
et_index()
overdue_dept_ratio_B = overdue_dept_ratio.groupby(by =
['SK_ID_CURR'])['AMT_CREDIT_SUM_OVERDUE'].sum().to_frame('total_overdue
').reset_index()
overdue_dept_ratio = overdue_dept_ratio.merge(overdue_dept_ratio_A, on
= ['SK_ID_CURR'], how = 'left')
overdue_dept_ratio = overdue_dept_ratio.merge(overdue_dept_ratio_B, on
= ['SK_ID_CURR'], how = 'left')
overdue_dept_ratio['overdue_dept_ratio'] =
overdue_dept_ratio['total_overdue']/overdue_dept_ratio['total_debt']
del overdue_dept_ratio_A, overdue_dept_ratio_B
```

```
del overdue_dept_ratio['AMT_CREDIT_SUM_DEBT'],
overdue_dept_ratio['AMT_CREDIT_SUM_OVERDUE'],
overdue_dept_ratio['total_debt'], overdue_dept_ratio['total_overdue']
overdue_dept_ratio = overdue_dept_ratio.groupby(by =
['SK_ID_CURR'])['overdue_dept_ratio'].mean().to_frame('overdue_dept_rat
io').reset_index()
poscash = pd.read_csv('../input/home-credit-default-
risk/POS_CASH_balance.csv')
avg_pos_cash_dpd = poscash[['SK_ID_CURR', 'SK_DPD']]
avg_pos_cash_dpd = avg_pos_cash_dpd.groupby(by =
['SK_ID_CURR'])['SK_DPD'].mean().to_frame('avg_pos_cash_dpd').reset_ind
ex()
previous_application = pd.read_csv('../input/home-credit-default-
risk/previous_application.csv')
prev_apps =
previous_application['SK_ID_PREV'].groupby(previous_application['SK_ID_
CURR']).count().to_frame('prev_apps').reset_index()
prev_refused_apps =
previous_application[previous_application['NAME_CONTRACT_STATUS'] ==
'Refused']
prev_refused_apps =
prev_refused_apps['SK_ID_PREV'].groupby(prev_refused_apps['SK_ID_CURR']
).count().to_frame('prev_refused_apps').reset_index()
prev_refused_apps_ratio = prev_apps.merge(prev_refused_apps, on =
['SK_ID_CURR'], how = 'left')
prev_refused_apps_ratio['prev_refused_apps'] =
prev_refused_apps_ratio['prev_refused_apps'].fillna(0)
prev_refused_apps_ratio['prev_refused_apps_ratio'] =
prev_refused_apps_ratio['prev_refused_apps']/prev_refused_apps_ratio['p
rev_apps']
del prev_refused_apps_ratio['prev_apps'],
prev_refused_apps_ratio['prev_refused_apps']
application = pd.read_csv('../input/home-credit-default-
risk/application_train.csv')
currCredit_currGoodsPrice_ratio = application[['SK_ID_CURR',
'AMT_CREDIT', 'AMT_GOODS_PRICE']].reset_index(drop = True)
currCredit_currGoodsPrice_ratio['currCredit_currGoodsPrice_ratio'] =
currCredit_currGoodsPrice_ratio['AMT_CREDIT']/currCredit_currGoodsPrice
_ratio['AMT_GOODS_PRICE']
del currCredit_currGoodsPrice_ratio['AMT_CREDIT'],
currCredit_currGoodsPrice_ratio['AMT_GOODS_PRICE']
annuityamt_income_ratio = application[['SK_ID_CURR', 'AMT_ANNUITY',
'AMT_INCOME_TOTAL']].reset_index(drop = True)
annuityamt_income_ratio['annuityamt_income_ratio'] =
annuityamt_income_ratio['AMT_ANNUITY']/annuityamt_income_ratio['AMT_INC
OME_TOTAL']
del annuityamt_income_ratio['AMT_ANNUITY'],
annuityamt_income_ratio['AMT_INCOME_TOTAL']
```

```
application_new = application.merge(bureau_loan_count, on =
['SK_ID_CURR'], how = 'left')
application_new = application_new.merge(bureau_loan_types, on =
['SK_ID_CURR'], how = 'left')
application_new = application_new.merge(bureau_active_loan_ratio, on =
['SK_ID_CURR'], how = 'left')
application_new = application_new.merge(bureau_avg_enddate_future, on =
['SK_ID_CURR'], how = 'left')
application_new = application_new.merge(dept_credit_ratio, on =
['SK_ID_CURR'], how = 'left')
application_new = application_new.merge(overdue_dept_ratio, on =
['SK_ID_CURR'], how = 'left')
application_new = application_new.merge(avg_pos_cash_dpd, on =
['SK_ID_CURR'], how = 'left')
application_new = application_new.merge(prev_refused_apps_ratio, on =
['SK_ID_CURR'], how = 'left')
application_new =
application_new.merge(currCredit_currGoodsPrice_ratio, on =
['SK_ID_CURR'], how = 'left')
application_new = application_new.merge(annuityamt_income_ratio, on =
['SK_ID_CURR'], how = 'left')
import xgboost as xgb
import numpy as np
import matplotlib.pyplot as plt
orig_data = application_new
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
le_count = 0
for col in orig_data:
    if orig_data[col].dtype == 'object':
        le.fit(orig_data[col])
        orig_data[col] = le.transform(orig_data[col])
        le_count += 1
print('%d στήλες έχουν μετατραπεί.' % le_count)
orig_data.replace([np.inf, -np.inf], np.nan, inplace=True)
class_sep=orig_data['TARGET'].value_counts().reset_index()
fig, ax = plt.subplots()
fig.set_size_inches([10,7])
p1=ax.bar(class_sep['index'],class_sep['TARGET'],alpha=0.8)
ax.set_xticks(class_sep['index'])
ax.bar_label(p1)
plt.title("Κατανομή της εξαρτημένης μεταβλητής (TARGET)")
plt.show()
print("Τα ακριβή ποσοστά δανείων με δυσκολία πληρωμής ή όχι:")
orig_data['TARGET'].value_counts(normalize=True)
X, y = orig_data.iloc[:,2:], orig_data.iloc[:,1]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.3,
random_state=123)
```

```
xgb_clf = xgb.XGBClassifier(random_state=123)
xgb_clf.get_params()
import xgboost as xgb
from xgboost import XGBClassifier
from sklearn.model_selection import RandomizedSearchCV,
StratifiedKFold, GridSearchCV
from sklearn.metrics import accuracy_score, f1_score,
classification_report, confusion_matrix
xgb_param_grid={"n_estimators":[100,600,1000],
                "max_depth":[3,4,5],
                "learning_rate":[0.01,0.1,0.3],
                "scale_pos_weight":[11.38715005035247]}
xgb = XGBClassifier(objective="binary:logistic", eval_metric="auc",
use_label_encoder=False,random_state=123,tree_method='gpu_hist',
gpu_id=0)
cv_f=StratifiedKFold(n_splits=4,shuffle=True)
rand_search = GridSearchCV(estimator=xgb, param_grid=xgb_param_grid,
n_jobs=2, cv=cv_f, verbose=1, scoring='roc_auc')
rand_search.fit(X_train,y_train)
preds=rand_search.predict(X_test)
print("Παρακάτω παρουσιάζεται η έκθεση ταξινόμησης (classification
report)")
print(classification_report(y_test, preds))
print("Μια αρχική παρουσίαση του πίνακα σύγχυσης (confusion matrix)")
print(confusion_matrix(y_test, preds))
print("Best parameters found:", rand_search.best_params_)
print("Best score found:", rand_search.best_score_)
import matplotlib.pyplot as plt
from sklearn.datasets import make_classification
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
cm=confusion_matrix(y_test, preds, labels=rand_search.classes_)
disp=ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=['Αποπληρωμένο','Μη αποπληρωμένο'])
fig, ax = plt.subplots(figsize=(10,6))
disp.plot(ax=ax)
plt.title("Ο πίνακας σύγχυσης (The confusion matrix)")
plt.ylabel("Πραγματική τιμή (True label)")
plt.xlabel("Τιμή που προβλέφθηκε (Predicted label)")
plt.show()
preds2 = rand_search.predict_proba(X_test)[::,1]
fpr, tpr, _ = metrics.roc_curve(y_test,  preds2)
plt.subplots(figsize=(10,6))
plt.plot(fpr,tpr)
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.title("The ROC curve")
plt.show()
auc = metrics.roc_auc_score(y_test, preds2)
plt.subplots(figsize=(10,6))
```

```
plt.plot(fpr,tpr,label="AUC="+str(auc))
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.legend(loc=4)
plt.title("The AUC curve")
plt.show()
from xgboost import plot_importance
xgb_cl = rand_search.best_estimator_
fig, ax = plt.subplots(figsize=(10, 6))
plot_importance(xgb_cl, importance_type='weight', max_num_features=15,
ax=ax)
plt.show()
xgb_cl = rand_search.best_estimator_
fig, ax = plt.subplots(figsize=(10, 6))
plot_importance(xgb_cl, importance_type='gain', max_num_features=15,
ax=ax)
plt.show()
```

Βιβλιογραφία

(1978). Στο R. A. Eisenbeis, *Problems in applying discriminant analysis in credit scoring models* (σσ. 205-219). Journal of Banking & Finance. doi:10.1016/0378-4266(78)90012-2

(1996). Στο W. E. Henley, & D. J. Hand, *A k-Nearest-Neighbour Classifier for Assessing Consumer Credit Risk* (σσ. 77-95). Royal Statistical Society. doi:10.2307/2348414

(2012). Στο F. Louzada, P. H. Ferreira-Silva, & A. C. Diniz, *On the impact of disproportional samples in credit scoring models: An application to a Brazilian bank data* (σσ. 8071-8078). Expert Systems with Applications. doi:10.1016/j.eswa.2012.01.134

(2018). Στο L. Yu, R. Zhou, L. Tang, & R. Chen, *A DBN-based resampling SVM ensemble learning paradigm for credit classification with imbalanced data* (σσ. 192-202). Applied Soft Computing. doi:10.1016/j.asoc.2018.04.049

About Us: Project Jupyter's origins and governance. (χ.χ.). Ανάκτηση από Project Jupyter: https://jupyter.org/about

Addo, P., Guegan, D., & Hassani, B. (2018). *Credit Risk Analysis Using Machine and Deep Learning Models.* Risks. doi:10.3390/risks6020038

Bennett, S. (2019, Ιούνιος 11). *IBM Developer*. Ανάκτηση από Why machine learning is primarily written in Python: https://developer.ibm.com/blogs/why-machine-learning-is-primarily-written-in-python/

Business Intelligence and Analytics Software. (χ.χ.). Ανάκτηση από What is Tableau?: https://www.tableau.com/why-tableau/what-is-tableau

Gartner | Delivering Actionable, Objective Insight to Executives and Their Teams. (2022, Μάιος 22). Ανάκτηση από Analytics and Business Intelligence Platforms Reviews and Ratings: https://www.gartner.com/reviews/market/analytics-business-intelligence-platforms

Hamori, S., Kawai, M., Kume, T., Murakami, Y., & Watanabe, C. (2018). *Ensemble Learning or Deep Learning? Application to Default Risk Analysis.* J. Risk Financial Manag. doi:10.3390/jrfm11010012

Hebb, D. (1949). *The organization of behavior. A neuropsychological theory.* Ανάκτηση από MPG.PuRe: http://hdl.handle.net/11858/00-001M-0000-002B-9E2B-A

Home Credit. (2022, Μάιος 23). Ανάκτηση από About Us: https://www.homecredit.net/about-us.aspx

Home Credit. (2022, Μάιος 23). Ανάκτηση από Loans Tool: https://www.homecredit.net/about-us/loans-tool.aspx

Kaggle: Overview. (2022, Μάιος 20). Ανάκτηση από Linkedin: https://www.linkedin.com/company/kaggle/

Kaggle: Your Home for Data Science. (2018, Μάιος 18). Ανάκτηση από Home Credit Default Risk: Can you predict how capable each applicant is of repaying a loan?: https://www.kaggle.com/competitions/home-credit-default-risk/overview/description

Meta Kaggle. (2022, Μάιος 20). Ανάκτηση από Kaggle: Your Machine Learning and Data Science Community: https://www.kaggle.com/datasets/kaggle/meta-kaggle?select=Users.csv

Moyer, E. (2017, Μάρτιος 8). *Google buys Kaggle and its gaggle of AI geeks*. Ανάκτηση από CNET: https://www.cnet.com/science/google-buys-kaggle-and-its-gaggle-of-ai-geeks/

Peters, T. (2004, Αύγουστος 19). *Welcome to Python.org*. Ανάκτηση από Python Enhancement Proposals - PEP 20 - The Zen of Python: https://peps.python.org/pep-0020/

R2D3: Statistics and Data Visualization. (2015, Ιούλιος 27). Ανάκτηση από Μία οπτική εισαγωγή στη μηχανική μάθηση (machine learning): http://www.r2d3.us/οπτική-εισαγωγή-στη-μηχανική-μάθηση-μέρος-1/

Salian, I. (2018, Αύγουστος 2). *SuperVize Me: What's the Difference Between Supervised, Unsupervised, Semi-Supervised and Reinforcement Learning?* Ανάκτηση από NVIDIA Blog: https://blogs.nvidia.com/blog/2018/08/02/supervised-unsupervised-learning/

Samuel, A. L. (1959, Ιούλιος 3). *Some Studies in Machine Learning Using the Game of Checkers*. Ανάκτηση από IEEE Xplore: https://ieeexplore.ieee.org/document/5392560

Stack Overflow - Where Developers Learn, Share, & Build Careers. (2022, Μάιος 22). Ανάκτηση από Tags: https://stackoverflow.com/tags

Welcome to Python.org. (χ.χ.). Ανάκτηση από https://www.python.org/

What Is Machine Learning (ML)? (2020, Ιούνιος 26). Ανάκτηση από UC Berkeley School of Information: https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/

What is Machine Learning? (2020, Ιούλιος 15). Ανάκτηση από IBM Cloud Learn Hub: https://www.ibm.com/cloud/learn/machine-learning

Αγγελιδάκης, Ν. (2015). *Νίκος Αγγελιδάκης.* Ανάκτηση από Εισαγωγή στον προγραμματισμό με την Python: http://aggelid.mysch.gr/pythonbook/

Βικιπαίδεια. (2022, Μάιος 20). Ανάκτηση από Κατάλογος χωρών ανά πληθυσμό: https://el.wikipedia.org/wiki/Κατάλογος_χωρών_ανά_πληθυσμό

Ιστορία της Ιονικής Τράπεζας. (χ.χ.). Ανάκτηση από Ιονική Ενότητα: https://www.ionikienotita.gr/?page_id=1174

Κράλογλου, Σ. (2007, Μάΐος 03). *Η αρχαιότερη τράπεζα στο κόσμο*. Ανάκτηση από Capital.gr: https://www.capital.gr/epixeiriseis/287438/i-arxaioteri-trapeza-sto-kosmo

Νόμισμα (κέρμα). (2022, Μάΐος 12). Ανάκτηση από Βικιπαίδεια: https://el.wikipedia.org/wiki/Νόμισμα_(κέρμα)

Σερβετάς, Ν. (2019, Ιανουαρίου 26). *Ναΐτες: Οι ιππότες που εφηύραν τις σύγχρονες τράπεζες*. Ανάκτηση από Documento: https://www.documentonews.gr/article/naites-oi-ippotes-poy-efhyran-tis-sygxrones-trapezes/

Τι είναι τα μη εξυπηρετούμενα δάνεια (ΜΕΔ);. (2021, Ιανουάριος 14). Ανάκτηση από Ευρωπαϊκή Κεντρική Τράπεζα: https://www.bankingsupervision.europa.eu/about/ssmexplained/html/npl.el.html

Τράπεζα της Ελλάδος. (2002, Μάιος 23). Ανάκτηση από Εξέλιξη δανείων και καθυστερήσεων: https://www.bankofgreece.gr/statistika/ekseliksh-daneiwn-kai-kathysterhsewn