ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πτυχιακή Εργασία

Τίτλος Πτυχιακής Εργασίας	Εφαρμογή των δικτύων Kolmogorov-Arnold (KAN) στην πρόβλεψη πιστωτικού κινδύνου Application of Kolmogorov-Arnold networks (KAN) in credit risk forecasting
Ονοματεπώνυμο Φοιτητή	Ιωακείμ Ελ-Χαττάμπ-Μπριστογιάννης
Πατρώνυμο	Αχμέντ
Αριθμός Μητρώου	Π19048
Επιβλέπων	Διονύσιος Σωτηρόπουλος, Επίκουρος Καθηγητής

Ημερομηνία Παράδοσης Σεπτέμβριος 2024

Copyright ©

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου Σωτηρόπουλο Διονύσιο, που δίνοντας μου τα κατάλληλα εφόδια και συμβουλές με βοήθησε στην υλοποίηση της παρούσας διπλωματικής εργασίας.
Έπειτα, θα ήθελα να ευχαριστήσω και τους δικούς μου ανθρώπους, για τη βοήθεια και τη στήριξη που μου έχουν δώσει.

Περίληψη

Η πρόβλεψη του πιστωτικού κινδύνου αποτελεί έναν από τους πιο κρίσιμους παράγοντες στη λειτουργία των χρηματοπιστωτικών ιδρυμάτων, καθώς συνδέεται άμεσα με την ικανότητα των οργανισμών να αξιολογούν τον κίνδυνο αθέτησης πληρωμών από πελάτες και να λαμβάνουν στρατηγικές αποφάσεις σχετικά με την έγκριση δανείων και τη διαχείριση επενδυτικών κεφαλαίων. Η ακριβής πρόβλεψη του πιστωτικού κινδύνου επιτρέπει την αποτελεσματική διαχείριση των κεφαλαίων, τον περιορισμό των ζημιών και την ενίσχυση της σταθερότητας του χρηματοοικονομικού συστήματος.

Στην παρούσα πτυχιακή εργασία, διερευνάται η εφαρμογή της τεχνικής μηχανικής Kolmogorov-Arnold Networks (KAN) στην πρόβλεψη του πιστωτικού κινδύνου, συγκρίνοντας την αποδοτικότητά της με δύο ευρέως χρησιμοποιούμενες τεχνικές στον τομέα αυτό, το gradient boosting και τη logistic regression. Για τους σκοπούς της μελέτης χρησιμοποιήθηκε το dataset Home Credit Default Risk από το Kaggle, προκειμένου να αναλυθούν οι επιδόσεις των τριών τεχνικών στην πρόβλεψη της πιθανότητας αθέτησης πληρωμών.

Τα αποτελέσματα της ανάλυσης έδειξαν ότι η μέθοδος KAN είχε καλύτερες επιδόσεις από τη logistic regression, αλλά υστερούσε σε σχέση με το gradient boosting. Παρά τα ικανοποιητικά αποτελέσματα, η τεχνική KAN απαιτεί περαιτέρω βελτιστοποίηση για να ανταγωνιστεί αποτελεσματικά πιο προηγμένες τεχνικές, όπως το gradient boosting, που αποδείχθηκε ανώτερο στην πρόβλεψη του πιστωτικού κινδύνου. Τα ευρήματα υποδεικνύουν ότι η KAN μπορεί να αποτελέσει μια υποσχόμενη μέθοδο στον τομέα αυτόν, με σημαντικά περιθώρια βελτίωσης.

Λέξεις Κλειδιά:

Πρόβλεψη Πιστωτικού Κινδύνου, Δίκτυα Kolmogorov-Arnold(KAN), Ενίσχυση κλίσης, Λογιστική παλινδρόμησης, Πιθανότητα Αθέτησης

Abstract

Credit risk prediction is one of the most critical factors in the operation of financial institutions, as it is directly linked to their ability to assess the risk of default by clients and make strategic decisions regarding loan approvals and capital management. Accurate credit risk prediction enables effective capital allocation, minimizes losses, and enhances the stability of the financial system.

This thesis explores the application of Kolmogorov-Arnold Networks (KAN) in credit risk prediction, comparing its effectiveness with two widely used techniques in this field: gradient boosting and logistic regression. The study utilizes the Home Credit Default Risk dataset from Kaggle to analyze the performance of these three techniques in predicting default probability.

The results indicate that the KAN method outperforms logistic regression but falls short compared to gradient boosting. While KAN achieved satisfactory results, it requires further optimization to compete effectively with more advanced techniques like gradient boosting, which proved superior in credit risk prediction. The findings suggest that KAN has potential as a promising method in this field, with significant room for improvement.

Key words:

Credit Risk Prediction, Kolmogorov-Arnold Networks (KAN), Gradient Boosting, Logistic Regression, Default Probability

Πίνακας Περιεχομένων

Κατάλογος Εικόνων

Κατάλογος Πινάκων

Κατάλογος Διαγραμμάτων

Εισαγωγή

Η αξιολόγηση του πιστωτικού κινδύνου αποτελεί κρίσιμη διαδικασία για τα χρηματοπιστωτικά ιδρύματα, όπως οι τράπεζες και οι πιστωτικοί οργανισμοί, που πρέπει να λαμβάνουν αποφάσεις για τον δανεισμό με όσο το δυνατόν μεγαλύτερη ακρίβεια. Η σημασία της ακριβούς αξιολόγησης του πιστωτικού κινδύνου είναι προφανής, καθώς οι αθετήσεις δανείων μπορεί να προκαλέσουν σημαντικές οικονομικές απώλειες και αυξημένο κίνδυνο για τα ιδρύματα αυτά. Παραδοσιακά, τα συστήματα αξιολόγησης πιστοληπτικής ικανότητας, όπως τα μοντέλα λογιστικής παλινδρόμησης και το FICO, χρησιμοποιούνται ευρέως για την εκτίμηση της πιθανότητας αθέτησης από έναν δανειολήπτη. Ωστόσο, αυτά τα μοντέλα παρουσιάζουν περιορισμούς, ιδίως όσον αφορά τη δυνατότητα τους να κατανοούν την πολυπλοκότητα της συμπεριφοράς των δανειοληπτών, ειδικά στην περίπτωση ατόμων χωρίς αξιόπιστο πιστωτικό ιστορικό. Με την άνοδο νέων μορφών δεδομένων και τεχνικών ανάλυσης, όπως η μηχανική μάθηση, υπάρχει αυξανόμενη ανάγκη για καινοτόμα μοντέλα που θα μπορέσουν να δώσουν πιο ακριβείς προβλέψεις και να μειώσουν τον κίνδυνο αθέτησης.

Ένα από τα σημαντικότερα σύνολα δεδομένων για την πρόβλεψη πιστωτικού κινδύνου είναι το **Home Credit Default Risk**, το οποίο έχει δημιουργηθεί από την Home Credit Group [1]. Το συγκεκριμένο σύνολο δεδομένων περιλαμβάνει εκτενή αρχεία αιτήσεων δανείων, καθώς και δημογραφικά στοιχεία, ιστορικά δεδομένα οικονομικών συμπεριφορών, χαρακτηριστικά απασχόλησης και στοιχεία σχετικά με τα δάνεια. Το πλούσιο αυτό σύνολο δεδομένων δίνει τη δυνατότητα εφαρμογής μοντέλων μηχανικής μάθησης με στόχο την πρόβλεψη του κινδύνου αθέτησης δανείου. Το ενδιαφέρον με αυτά τα δεδομένα είναι ότι καλύπτουν τόσο πελάτες με σταθερό πιστωτικό ιστορικό όσο και εκείνους που είναι νέοι στο πιστωτικό σύστημα, γεγονός που καθιστά το σύνολο κατάλληλο για την ανάπτυξη ευέλικτων μοντέλων πρόβλεψης.

Τα τελευταία χρόνια, η μηχανική μάθηση έχει δείξει τη δυναμική της σε πολλές περιοχές πρόβλεψης, συμπεριλαμβανομένου του τομέα της πιστοληπτικής αξιολόγησης. Η ικανότητά της να επεξεργάζεται μεγάλα και πολυδιάστατα δεδομένα και να εντοπίζει περίπλοκες σχέσεις μεταξύ μεταβλητών έχει φέρει μεγάλες αλλαγές στον τομέα. Τα παραδοσιακά στατιστικά μοντέλα, όπως η λογιστική παλινδρόμηση, εξακολουθούν να χρησιμοποιούνται, αλλά η χρήση πιο σύνθετων μοντέλων μηχανικής μάθησης, όπως τα δέντρα αποφάσεων (Decision trees) , οι μηχανές διανυσμάτων υποστήριξης(Support vector machines,SVM) και τα νευρωνικά δίκτυα (neural networks), έχουν αποδείξει μεγαλύτερη ακρίβεια σε πολλά προβλήματα πρόβλεψης. Παρ' όλα αυτά, η συνεχής αναζήτηση για μοντέλα που όχι μόνο είναι ακριβή, αλλά και ερμηνεύσιμα, αποτελεί πρόκληση.

Μια καινοτόμα προσέγγιση που εξετάζεται στην παρούσα έρευνα είναι η χρήση των **Kolmogorov-Arnold Networks (KANs**), τα οποία βασίζονται στο θεώρημα αναπαράστασης των Kolmogorov-Arnold. Το θεωρητικό υπόβαθρο αυτών των δικτύων τους επιτρέπει να προσεγγίζουν οποιαδήποτε συνεχή συνάρτηση, γεγονός που τα καθιστά μια ιδιαίτερα ελκυστική επιλογή για τη μοντελοποίηση περίπλοκων προβλημάτων πρόβλεψης. Με τη χρήση αυτών των δικτύων στο σύνολο δεδομένων της Home Credit, η έρευνα στοχεύει να εξετάσει εάν μπορούν να προσφέρουν πιο ακριβείς προβλέψεις από τα παραδοσιακά μοντέλα. Επίσης, η σύγκριση της απόδοσης των KANs με άλλα καθιερωμένα μοντέλα μηχανικής μάθησης, όπως τα δέντρα αποφάσεων και οι μηχανές ενίσχυσης κλίσης, θα δώσει μια ξεκάθαρη εικόνα της πρακτικής τους αξίας.

Η δυνατότητα βελτίωσης της ακρίβειας στις προβλέψεις αθέτησης δανείων θα επιτρέψει στα χρηματοπιστωτικά ιδρύματα να λαμβάνουν πιο ενημερωμένες αποφάσεις σχετικά με την παροχή δανείων, μειώνοντας ταυτόχρονα τον οικονομικό κίνδυνο. Η παρούσα έρευνα συμβάλλει στη διαρκώς αναπτυσσόμενη βιβλιογραφία γύρω από τις εφαρμογές της μηχανικής μάθησης στον χρηματοπιστωτικό τομέα, ενώ παράλληλα εισάγει μια νέα μέθοδο με τα δίκτυα KAN, η οποία θα μπορούσε να αποτελέσει λύση σε ένα χρόνιο πρόβλημα της πρόβλεψης πιστωτικού κινδύνου.

Η πρόβλεψη της αθέτησης δανείων είναι εξαιρετικά κρίσιμη για τα χρηματοπιστωτικά ιδρύματα, καθώς μια τέτοια αθέτηση μπορεί να προκαλέσει σημαντικές απώλειες. Παραδοσιακά μοντέλα, όπως η λογιστική παλινδρόμηση, παρά την ευρεία τους χρήση, συχνά αδυνατούν να συλλάβουν τη σύνθετη αλληλεπίδραση μεταξύ πολλών μεταβλητών που σχετίζονται με την αποπληρωμή. Οι μη παραδοσιακοί δανειολήπτες,

όπως τα άτομα με περιορισμένο πιστωτικό ιστορικό, αποτελούν ειδική περίπτωση, όπου τα παραδοσιακά εργαλεία αποτυγχάνουν να δώσουν ακριβείς προβλέψεις. Το σύνολο δεδομένων Home Credit Default Risk παρέχει μια πληθώρα χαρακτηριστικών που αφορούν τη δημογραφία, την οικονομική κατάσταση και τα χαρακτηριστικά δανείων, τα οποία μπορούν να χρησιμοποιηθούν για τη δημιουργία προγνωστικών μοντέλων.

Η έρευνα αυτή επικεντρώνεται στη χρήση του **KAN,** ενός μοντέλου που έχει παραμεληθεί στον τομέα της πιστοληπτικής αξιολόγησης. Τα KANs έχουν τη θεωρητική ικανότητα να καταγράφουν μη γραμμικές σχέσεις μεταξύ μεταβλητών, κάτι που τα καθιστά ιδιαίτερα υποσχόμενα για την πρόβλεψη της αθέτησης δανείων. Η έρευνα αυτή θα εξετάσει την απόδοσή τους σε σύγκριση με πιο καθιερωμένα μοντέλα, προκειμένου να διαπιστώσει εάν μπορούν να προσφέρουν καλύτερη ακρίβεια στην πρόβλεψη πιστωτικού κινδύνου.

Τα ερευνητικά ερωτήματα που καθοδηγούν την παρούσα μελέτη περιλαμβάνουν την αξιολόγηση της ακρίβειας των KANs στην πρόβλεψη αθετήσεων δανείων, τη σύγκριση της απόδοσής τους με άλλα μοντέλα μηχανικής μάθησης και τον προσδιορισμό των πιο σημαντικών χαρακτηριστικών που επηρεάζουν την πρόβλεψη. Στόχος είναι η διερεύνηση της εφαρμοσιμότητας των δικτύων Kolmogorov-Arnold για την πρόβλεψη πιστωτικού κινδύνου και η συμβολή στη βελτίωση των πρακτικών διαχείρισης πιστωτικού κινδύνου.

1. Τεχνικές χρηματοοικονομικής πρόβλεψης

1.1 Επισκόπηση της πρόβλεψης πιστωτικού κινδύνου

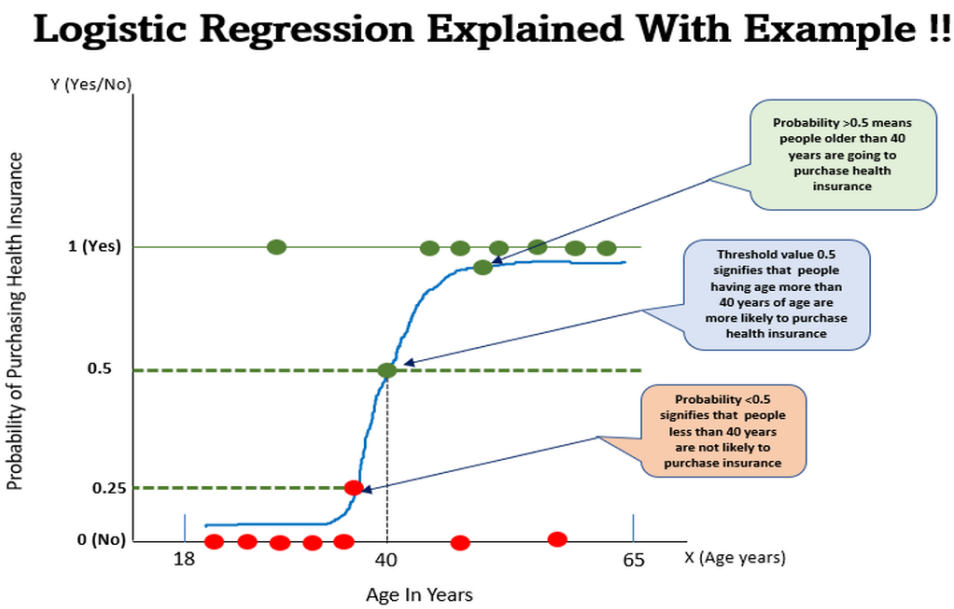
Η πρόβλεψη πιστωτικού κινδύνου είναι απαραίτητη προκειμένου τα χρηματοπιστωτικά ιδρύματα να αξιολογήσουν την πιθανότητα αθέτησης ενός δανείου από έναν δανειολήπτη. Οι ακριβείς προβλέψεις συμβάλλουν στη μείωση των ζημιών και στην ενημέρωση των αποφάσεων δανεισμού. Ιστορικά, οι στατιστικές μέθοδοι έχουν κυριαρχήσει στη μοντελοποίηση του πιστωτικού κινδύνου, αλλά οι πρόσφατες εξελίξεις στη μηχανική μάθηση έχουν παράσχει νέα εργαλεία για την ενίσχυση της προγνωστικής ακρίβειας. Αυτή η ενότητα εξετάζει την εξέλιξη των μεθόδων πρόβλεψης πιστωτικού κινδύνου από τις παραδοσιακές στατιστικές τεχνικές στις σύγχρονες προσεγγίσεις μηχανικής μάθησης.

1.1.1 Παραδοσιακές μέθοδοι πρόβλεψης πιστωτικού κινδύνου

Στο παρελθόν, τα χρηματοπιστωτικά ιδρύματα χρησιμοποιούσαν κυρίως στατιστικά μοντέλα, όπως η **λογιστική παλινδρόμηση** και η **γραμμική διακριτική ανάλυση (Linear Discriminant Analysis, LDA)** για την πρόβλεψη του πιστωτικού κινδύνου. Η λογιστική παλινδρόμηση, η οποία μοντελοποιεί την πιθανότητα ενός δυαδικού αποτελέσματος (αθέτηση ή μη αθέτηση), έχει εφαρμοστεί ευρέως λόγω της απλότητας και της ευκολίας ερμηνείας της. Υποθέτει μια γραμμική σχέση μεταξύ των ανεξάρτητων μεταβλητών (π.χ. εισόδημα, καθεστώς απασχόλησης, επίπεδα χρέους) και της μεταβλητής-στόχου (αθέτηση δανείου). Αυτή η μέθοδος ήταν ιδιαίτερα δημοφιλής λόγω της ικανότητάς της να παρέχει σαφείς εξηγήσεις για τον αντίκτυπο των μεμονωμένων μεταβλητών στην πιστοληπτική ικανότητα, η οποία είναι ζωτικής σημασίας για την κανονιστική συμμόρφωση .

Ωστόσο, η λογιστική παλινδρόμηση έχει σημαντικούς περιορισμούς, ιδιαίτερα στην αδυναμία της να συλλάβει μη γραμμικές σχέσεις μεταξύ μεταβλητών. Αυτά τα παραδοσιακά μοντέλα συχνά αποτυγχάνουν όταν το σύνολο δεδομένων είναι μεγάλο ή περιέχει πολύπλοκες αλληλεπιδράσεις μεταξύ χαρακτηριστικών [2] . Για παράδειγμα, τα δεδομένα συμπεριφοράς δανειοληπτών, όπως το ιστορικό πληρωμών ή η χρήση πίστωσης, συχνά παρουσιάζουν πολύπλοκα μοτίβα που τα απλά γραμμικά μοντέλα δεν μπορούν να αναπαραστήσουν επαρκώς.

Η γραμμική διακριτική ανάλυση έχει επίσης χρησιμοποιηθεί στη βαθμολόγηση πιστοληπτικής ικανότητας, ιδίως για τη διάκριση μεταξύ δανειοληπτών υψηλού και χαμηλού κινδύνου. Η LDA στοχεύει στην εξεύρεση ενός γραμμικού συνδυασμού μεταβλητών που διαχωρίζει καλύτερα τους δανειολήπτες σε διαφορετικές κατηγορίες κινδύνου. Ενώ αυτή η μέθοδος αποδίδει καλά όταν η υποκείμενη κατανομή δεδομένων είναι φυσιολογική, δυσκολεύεται όταν τα δεδομένα περιέχουν λοξές ή μη ισορροπημένες κατανομές, κάτι που είναι τυπικό σε σύνολα πιστωτικών δεδομένων όπου οι περιπτώσεις αθέτησης είναι σπάνιες .



ΕΙΚΟΝΑ 1.1: Λογιστική παλινδρόμηση

1.1.2 Σύγχρονες προσεγγίσεις μηχανικής μάθησης

Οι περιορισμοί των παραδοσιακών μοντέλων έχουν οδηγήσει στην υιοθέτηση τεχνικών μηχανικής μάθησης για την πρόβλεψη πιστωτικού κινδύνου. Μέθοδοι όπως τα **τυχαία δάση (random forests)**, οι **μηχανές ενίσχυσης κλίσης (Gradient Boosting Machine , GBM)** και οι **μηχανές διανυσμάτων υποστήριξης (Support vector machine , SVM)** χρησιμοποιούνται πλέον ευρέως λόγω της ικανότητάς τους να μοντελοποιούν πολύπλοκες, μη γραμμικές σχέσεις και να επεξεργάζονται αποτελεσματικά μεγάλα σύνολα δεδομένων. Αυτά τα μοντέλα είναι επίσης πιο ευέλικτα στο χειρισμό αλληλεπιδράσεων χαρακτηριστικών και προσφέρουν καλύτερη προγνωστική απόδοση σε μη ισορροπημένα σύνολα δεδομένων.

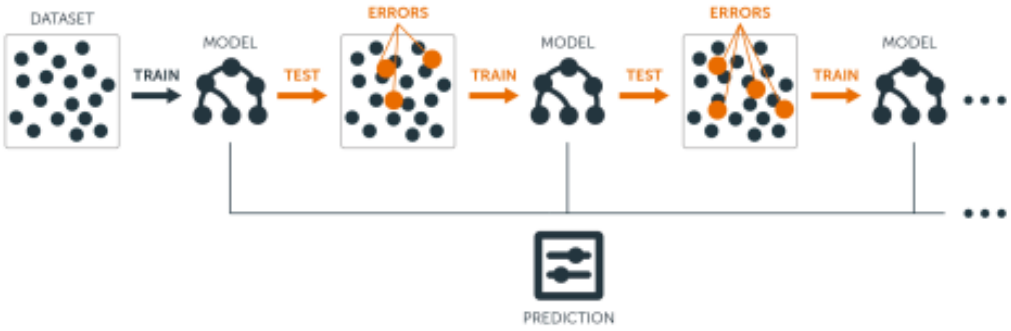
Για παράδειγμα, τα τυχαία δάση, μια μέθοδος συνόλου που δημιουργεί πολλαπλά δέντρα αποφάσεων και συγκεντρώνει τις προβλέψεις τους, έχει αποδειχθεί ότι ξεπερνά την λογιστική παλινδρόμηση υπολογίζοντας μεταβλητές αλληλεπιδράσεις και μη γραμμικές σχέσεις. Ομοίως, το Gradient Boosting δημιουργεί επαναλαμβανόμενα μοντέλα διορθώνοντας σφάλματα σε προηγούμενες προβλέψεις, γεγονός που επιτρέπει υψηλή ακρίβεια στην πρόβλεψη σπάνιων γεγονότων όπως αθετήσεις δανείων.

Τα μοντέλα μηχανικής μάθησης επωφελούνται επίσης από την αυτοματοποιημένη επιλογή χαρακτηριστικών, επιτρέποντάς τους να επεξεργάζονται περισσότερες μεταβλητές και να εντοπίζουν εκείνες που προβλέπουν περισσότερο την αθέτηση. Οι μέθοδοι συνόλου όπως το Gradient Boosting και το Random Forests ξεπερνούν σταθερά τα παραδοσιακά μοντέλα στην πρόβλεψη αθετήσεων δανείων, ειδικά όταν ασχολούνται με μεγάλα, υψηλής διάστασης σύνολα δεδομένων [3].

Ενώ τα μοντέλα μηχανικής μάθησης παρέχουν ανώτερη προγνωστική ισχύ, έρχονται με συμβιβασμούς. Μία από τις βασικές προκλήσεις είναι η ερμηνευσιμότητά τους, ειδικά σε ρυθμιζόμενα περιβάλλοντα όπως η τραπεζική, όπου η κατανόηση του τρόπου λήψης αποφάσεων είναι κρίσιμη . Ενώ μοντέλα όπως τα τυχαία δάση και τα SVM μπορούν να προβλέψουν με μεγάλη ακρίβεια, η έλλειψη διαφάνειας μπορεί να τα καταστήσει δύσκολη στην εφαρμογή στην πράξη.

Με τη μετάβαση από τα παραδοσιακά στατιστικά μοντέλα σε πιο εξελιγμένες τεχνικές μηχανικής μάθησης, τα χρηματοπιστωτικά ιδρύματα μπόρεσαν να βελτιώσουν την ακρίβεια των προβλέψεων πιστωτικού κινδύνου. Ωστόσο, οι συμβιβασμοί μεταξύ ακρίβειας και ερμηνευσιμότητας εξακολουθούν να αποτελούν σημαντική πρόκληση, ιδίως σε κλάδους με υψηλό ρυθμιστικό πλαίσιο, όπως ο τραπεζικός τομέας. Οι επόμενες ενότητες θα διερευνήσουν πιο προηγμένες τεχνικές

μηχανικής μάθησης, συμπεριλαμβανομένων των **δικτύων Kolmogorov-Arnold (KANs)**, τα οποία υπόσχονται να βελτιώσουν περαιτέρω την προγνωστική ισχύ σε σύνθετα σύνολα δεδομένων.



ΕΙΚΟΝΑ 1.2: Εκπαίδευση μηχανής ενίσχυσης κλίσης

1.2 Δίκτυα Kolmogorov-Arnold στη μηχανική μάθηση

Τα δίκτυα Kolmogorov-Arnold (KANs) προσφέρουν μια νέα προσέγγιση στη μηχανική μάθηση αξιοποιώντας το **θεώρημα υπέρθεσης Kolmogorov** (1.1), το οποίο ισχυρίζεται ότι οποιαδήποτε πολυμεταβλητή συνεχής συνάρτηση μπορεί να αναπαρασταθεί ως υπέρθεση συνεχών μονομεταβλητών συναρτήσεων. Αυτή η ιδέα, που εισήχθη για πρώτη φορά από τον σοβιετικό μαθηματικό **Andrey Kolmogorov**, έθεσε τα θεμέλια για μια κατηγορία νευρωνικών δικτύων που μπορούν να μοντελοποιήσουν εξαιρετικά μη γραμμικές σχέσεις με λιγότερους νευρώνες από τα παραδοσιακά δίκτυα .
Τα KAN μελετώνται όλο και περισσότερο λόγω της δυνατότητάς τους να απλοποιούν την αρχιτεκτονική των μοντέλων βαθιάς μάθησης, διατηρώντας παράλληλα υψηλή προγνωστική ισχύ. Σε αντίθεση με τα παραδοσιακά βαθιά νευρωνικά δίκτυα, τα οποία απαιτούν μεγάλο αριθμό κρυφών στρωμάτων και νευρώνων για την προσέγγιση πολύπλοκων λειτουργιών, τα KANs στοχεύουν στην προσέγγιση αυτών των λειτουργιών με λιγότερους υπολογιστικούς πόρους. Αυτό καθιστά τα KAN εξαιρετικά αποτελεσματικά για εργασίες που απαιτούν γρήγορο υπολογισμό.

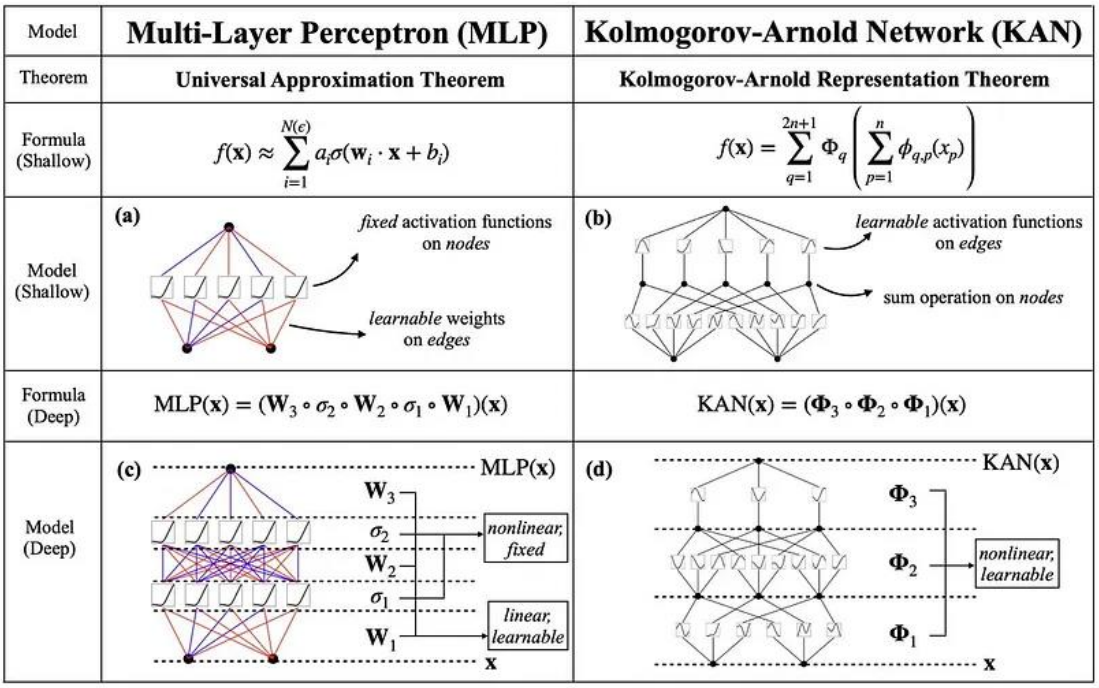
$$f(x) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \Phi_{q,p}(x_p) \right) \quad (1.1)$$

1.2.1 Δομή και πλεονεκτήματα των KAN

Η δομή ενός δικτύου Kolmogorov-Arnold αποτελείται από δύο κύρια στρώματα: ένα σύνολο μονομεταβλητών συναρτήσεων και ένα σύνολο συναρτήσεων άθροισης. Οι μονομεταβλητές συναρτήσεις μετασχηματίζουν κάθε μεταβλητή εισόδου ξεχωριστά, ενώ το επίπεδο άθροισης συνδυάζει αυτές τις μετασχηματισμένες εισόδους για να μοντελοποιήσει τις πολύπλοκες σχέσεις μεταξύ των χαρακτηριστικών. Αυτή η αρχιτεκτονική επιτρέπει στα KANs να προσεγγίζουν οποιαδήποτε συνεχή συνάρτηση, συμπεριλαμβανομένων εκείνων με εισόδους υψηλών διαστάσεων και μη γραμμικές εξαρτήσεις.

Ένα βασικό πλεονέκτημα των KANs είναι η ικανότητά τους να μειώνουν την πολυπλοκότητα των αρχιτεκτονικών βαθιάς μάθησης. Μελέτες των Maiorov και Pinkus (1999) [5] έδειξαν ότι τα KANs θα μπορούσαν να επιτύχουν το ίδιο επίπεδο ακρίβειας προσέγγισης με τα παραδοσιακά νευρωνικά δίκτυα, αλλά με λιγότερους νευρώνες. Αυτή η μείωση της πολυπλοκότητας όχι μόνο επιταχύνει τη διαδικασία κατάρτισης, αλλά μειώνει επίσης τον κίνδυνο υπερβολικής τοποθέτησης (overfitting), κάτι που αποτελεί ζήτημα σε σύνολα δεδομένων υψηλών διαστάσεων, όπως αυτά που χρησιμοποιούνται στην πρόβλεψη πιστωτικού κινδύνου.

Επιπλέον, τα KAN έχουν δείξει ισχυρές δυνατότητες σε τομείς όπου οι σχέσεις μεταξύ μεταβλητών είναι εξαιρετικά μη γραμμικές, όπως η μοντελοποίηση του κλίματος και η φυσική. Στο πλαίσιο του πιστωτικού κινδύνου, όπου η συμπεριφορά των δανειοληπτών και οι οικονομικοί δείκτες παρουσιάζουν μη γραμμικά μοτίβα, τα KAN προσφέρουν μια αποτελεσματική εναλλακτική λύση στα παραδοσιακά μοντέλα μηχανικής μάθησης όπως τα τυχαία δάση και οι μηχανές ενίσχυσης κλίσης.



ΕΙΚΟΝΑ 1.3: Σύγκριση δικτύου KAN με Multilayer perceptron

1.2.2 Εφαρμογές των KAN στην πρόβλεψη πιστωτικού κινδύνου

Η εφαρμογή των KAN στην πρόβλεψη πιστωτικού κινδύνου είναι ελλιπής. Τα χρηματοπιστωτικά ιδρύματα ασχολούνται με τεράστια σύνολα δεδομένων που περιλαμβάνουν χαρακτηριστικά δανειοληπτών, ιστορικά πληρωμών και οικονομικούς δείκτες, τα οποία παρουσιάζουν περίπλοκες σχέσεις που είναι δύσκολο να αποτυπωθούν με απλούστερα μοντέλα. Τα KAN, με την ικανότητά τους να μοντελοποιούν πολύπλοκες, μη γραμμικές συναρτήσεις, παρέχουν ένα ισχυρό εργαλείο για την πρόβλεψη αθετήσεων δανείων με μεγαλύτερη ακρίβεια από τις παραδοσιακές στατιστικές μεθόδους όπως η λογιστική παλινδρόμηση.

Επιπλέον, η ικανότητα των KAN να χειρίζονται δεδομένα υψηλών διαστάσεων τα καθιστά κατάλληλα για σύνολα δεδομένων που περιέχουν πολλά χαρακτηριστικά δανειολήπτη. Καθώς τα σύνολα οικονομικών δεδομένων συνεχίζουν να αυξάνονται σε μέγεθος και πολυπλοκότητα, η επεκτασιμότητα και η αποτελεσματικότητα των KAN θα γίνονται όλο και πιο σημαντικές. Το έργο των Hornik et al. (1991) [6] υποδηλώνει ότι καθώς αυξάνεται η διάσταση των δεδομένων εισόδου, τα KAN μπορούν να διατηρήσουν την προγνωστική τους ισχύ με λιγότερους πόρους, καθιστώντας τα ένα πολλά υποσχόμενο εργαλείο για την αξιολόγηση πιστωτικού κινδύνου μεγάλης κλίμακας.

1.3 Τεχνικές Προεπεξεργασίας Δεδομένων στον Πιστωτικό Κίνδυνο

Η εξαγωγή χαρακτηριστικών (feature engineering) και η προεπεξεργασία δεδομένων είναι θεμελιώδη βήματα για τη δημιουργία ισχυρών μοντέλων μηχανικής μάθησης για την πρόβλεψη πιστωτικού κινδύνου. Η ποιότητα των χαρακτηριστικών και το πόσο καλά αντιπροσωπεύουν τα υποκείμενα μοτίβα στα δεδομένα επηρεάζουν σημαντικά την ακρίβεια οποιουδήποτε μοντέλου πρόβλεψης. Στη μοντελοποίηση πιστωτικού κινδύνου, τα ανεπεξέργαστα δεδομένα από αιτήσεις δανείων περιλαμβάνουν συχνά διάφορα αριθμητικά, κατηγορικά και μερικές φορές βασισμένα σε κείμενο χαρακτηριστικά. Έτσι, οι κατάλληλες τεχνικές μηχανικής χαρακτηριστικών και προεπεξεργασίας είναι απαραίτητες για τη μετατροπή αυτών των ακατέργαστων δεδομένων σε σημαντικές εισόδους που μπορούν να επεξεργαστούν τα μοντέλα μηχανικής μάθησης.

1.3.1 Χειρισμός ελλειπόντων δεδομένων και μη ισορροπημένων συνόλων δεδομένων

Μία από τις κρίσιμες προκλήσεις στα σύνολα δεδομένων πιστωτικού κινδύνου είναι η έλλειψη δεδομένων, η οποία μπορεί να προκύψει λόγω ελλιπών αιτήσεων δανείων ή σφαλμάτων στη συλλογή δεδομένων. Μέθοδοι όπως **ο μέσος καταλογισμός** ή ο **καταλογισμός k-πλησιέστερων γειτόνων (KNN)** χρησιμοποιούνται συνήθως για τη συμπλήρωση τιμών που λείπουν . Ωστόσο, αυτές οι τεχνικές μπορούν να εισάγουν μεροληψία στα δεδομένα εάν δεν χρησιμοποιηθούν προσεκτικά.

Ένα άλλο σημαντικό ζήτημα στα σύνολα δεδομένων πιστωτικού κινδύνου είναι το **πρόβλημα της ανισορροπίας των τάξεων**, όπου ο αριθμός των πελατών που αθετούν τις υποχρεώσεις τους είναι πολύ μικρότερος σε σύγκριση με τους πελάτες που δεν αθετούν τις υποχρεώσεις τους. Για την αντιμετώπιση αυτού του προβλήματος έχουν αναπτυχθεί διάφορες τεχνικές, συμπεριλαμβανομένων **μεθόδων υπερδειγματοληψίας** όπως η SMOTE (Synthetic Minority Over-sampling Technique) και η **υποδειγματοληψία** της πλειοψηφικής κατηγορίας. Αυτές οι μέθοδοι διασφαλίζουν ότι το μοντέλο δεν γίνεται προκατειλημμένο προς την πλειοψηφική τάξη, βελτιώνοντας έτσι την ικανότητά του να προβλέπει αθετήσεις δανείων.

1.3.2 Μετασχηματισμός και κωδικοποίηση χαρακτηριστικών

Οι κατηγορικές μεταβλητές, όπως οι τύποι δανείων ή τα δημογραφικά στοιχεία των πελατών, συχνά πρέπει να μετατραπούν σε αριθμητικές μορφές για τους αλγόριθμους μηχανικής μάθησης για την αποτελεσματική επεξεργασία τους. Η **κωδικοποίηση One-hot (OHE)** και **η κωδικοποίηση ετικετών (label encoding)** χρησιμοποιούνται συχνά για το χειρισμό τέτοιων μεταβλητών. Η κωδικοποίηση one-hot, αν και χρήσιμη, μπορεί να αυξήσει τη διάσταση του συνόλου δεδομένων όταν υπάρχουν πολλές κατηγορίες.

Τα αριθμητικά χαρακτηριστικά απαιτούν επίσης προεπεξεργασία, ειδικά όταν δεν κατανέμονται κανονικά. Τεχνικές όπως **log transformation**, η **κλιμάκωση min-max (min-man scaling)** και η **κανονικοποίηση z-score (z-score normalization)** εφαρμόζονται για να διασφαλιστεί ότι όλα τα χαρακτηριστικά είναι σε συγκρίσιμη κλίμακα και συμβάλλουν εξίσου στις προβλέψεις του μοντέλου. Αυτές οι τεχνικές είναι ιδιαίτερα σημαντικές όταν χρησιμοποιούνται μοντέλα που βασίζονται σε απόσταση, όπως **k-πλησιέστεροι γείτονες (KNN)** ή **μηχανές διανυσμάτων υποστήριξης (SVM),** οι οποίες είναι ευαίσθητες στην κλιμάκωση χαρακτηριστικών.

Label Encoding

Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

→

One Hot Encoding

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

ΕΙΚΟΝΑ 1.4: κωδικοποίηση ετικετών και κωδικοποίηση One-hot (OHE)

1.3.3 Τεχνικές επιλογής χαρακτηριστικών

Η επιλογή χαρακτηριστικών είναι ένα κρίσιμο βήμα για τη μείωση των διαστάσεων του συνόλου δεδομένων και τη βελτίωση της απόδοσης του μοντέλου. **Η ανάλυση συσχέτισης**, **η ανάλυση κύριων συνιστωσών (Principal component analysis, PCA)** και η **αναδρομική εξάλειψη χαρακτηριστικών (recursive feature elimination, RFE)** είναι συνήθως χρησιμοποιούμενες τεχνικές για τον εντοπισμό των πιο ενημερωτικών χαρακτηριστικών και την απόρριψη περιττών ή άσχετων . Στο πλαίσιο της πρόβλεψης πιστωτικού κινδύνου, χαρακτηριστικά όπως το ποσό του δανείου, το πιστωτικό αποτέλεσμα και το προηγούμενο ιστορικό αθέτησης τείνουν να έχουν υψηλή προγνωστική ισχύ, άλλα μπορεί να εισάγουν θόρυβο στο μοντέλο.

1.3.4 Εξαγωγή χαρακτηριστικών Συγκεκριμένου Τομέα

Πέρα από τις τυπικές τεχνικές, η εξαγωγή χαρακτηριστικών συγκεκριμένου τομέα είναι απαραίτητη για τη δημιουργία προσαρμοσμένων λειτουργιών που αποτυπώνουν μοναδικά οικονομικά μοτίβα. Για παράδειγμα, ο υπολογισμός **του λόγου χρέους προς εισόδημα**, **του λόγου δανείου προς αξία** ή η δημιουργία χαρακτηριστικών βάσει χρόνου, όπως η **ηλικία του πιστωτικού λογαριασμού,** μπορεί να παρέχει στο μοντέλο πιο λεπτές πληροφορίες σχετικά με την οικονομική συμπεριφορά ενός δανειολήπτη. Αυτά τα μηχανικά χαρακτηριστικά συχνά έχουν περισσότερη προγνωστική ισχύ από τις ακατέργαστες μεταβλητές.

1.4 Αξιολόγηση Απόδοσης σε Μοντέλα Πρόβλεψης Πιστωτικού Κινδύνου

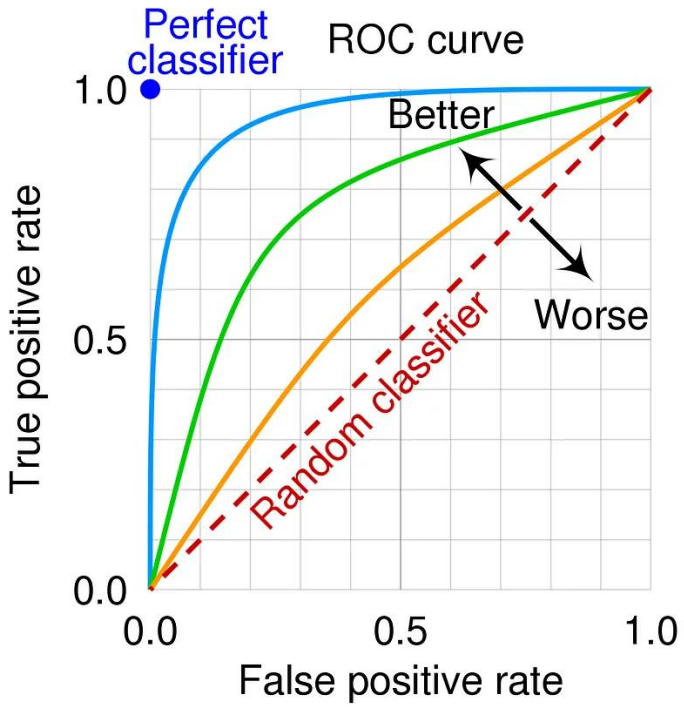
Η αξιολόγηση της απόδοσης των μοντέλων πρόβλεψης πιστωτικού κινδύνου αποτελεί κρίσιμο βήμα για τη διασφάλιση της αξιοπιστίας και της αποτελεσματικότητας του μοντέλου πριν από την ανάπτυξή του σε σενάρια πραγματικού κόσμου. Μια διεξοδική αξιολόγηση περιλαμβάνει τη χρήση διαφόρων μετρήσεων, τεχνικών και μεθόδων επικύρωσης προσαρμοσμένων για την αντιμετώπιση των ειδικών προκλήσεων που είναι εγγενείς στην πρόβλεψη πιστωτικού κινδύνου, όπως τα μη ισορροπημένα δεδομένα και η ανάγκη για ερμηνευσιμότητα.

1.4.1 Μετρήσεις αξιολόγησης

Διάφορες μετρήσεις χρησιμοποιούνται συνήθως για την αξιολόγηση της απόδοσης των μοντέλων μηχανικής μάθησης στην πρόβλεψη πιστωτικού κινδύνου. Αυτές οι μετρήσεις είναι: accuracy, precision, recall, και F1 score. Ωστόσο, δεδομένης της τυπικά μη ισορροπημένης φύσης των συνόλων δεδομένων πιστωτικού κινδύνου - όπου οι αθετήσεις είναι σχετικά σπάνιες - δείκτες όπως **η AUC-ROC (Area Under the Receiver Operating Characteristic Curve)** και η **Precision-Recall AUC** είναι πιο κατάλληλες. Αυτές οι μετρήσεις παρέχουν καλύτερη αξιολόγηση του βαθμού στον οποίο το μοντέλο μπορεί να κάνει διάκριση μεταξύ υπερήμερων και μη υπερήμερων δανειοληπτών, ακόμη και όταν υπάρχει ταξική ανισορροπία.

Ειδικότερα, η καμπύλη AUC-ROC χρησιμοποιείται ευρέως για την αξιολόγηση των μοντέλων ταξινόμησης στην πρόβλεψη πιστωτικού κινδύνου, διότι αξιολογεί τον συμβιβασμό μεταξύ αληθώς θετικών (σωστά ταυτοποιημένων κακοπληρωτών) και ψευδώς θετικών αποτελεσμάτων (μη υπερήμεροι που ταξινομούνται εσφαλμένα ως κακοπληρωτές) σε διαφορετικά όρια [7]. Τα μοντέλα με υψηλότερες τιμές AUC-ROC αποδίδουν καλύτερα στην πρόβλεψη του πιστωτικού κινδύνου, καθιστώντας αυτή τη μέτρηση απαραίτητη για την αξιολόγηση αλγορίθμων βαθμολόγησης πιστοληπτικής ικανότητας.

Μια άλλη μέτρηση που είναι ζωτικής σημασίας για την αξιολόγηση των μοντέλων πιστωτικού κινδύνου είναι ο **συντελεστής Gini (Gini coefficient)**, ο οποίος είναι μια κανονικοποιημένη έκδοση του AUC-ROC και παρέχει έναν μόνο αριθμό για την αξιολόγηση της απόδοσης του μοντέλου. Ο συντελεστής Gini χρησιμοποιείται συχνά σε χρηματοπιστωτικά ιδρύματα επειδή παρέχει ένα διαισθητικό μέτρο της εξουσίας διακρίσεων του μοντέλου . Ο συντελεστής Gini προτιμάται στον τραπεζικό τομέα λόγω της ερμηνευσιμότητάς του και της άμεσης εφαρμογής του στις εκτιμήσεις χρηματοοικονομικού κινδύνου.



ΕΙΚΟΝΑ 1.5: ROC Curve

1.4.2 Μάθηση και ερμηνευσιμότητα ευαίσθητη στο κόστος

Στην πρόβλεψη πιστωτικού κινδύνου, το κόστος της εσφαλμένης ταξινόμησης μπορεί να διαφέρει σημαντικά μεταξύ των κατηγοριών. Ο εσφαλμένος χαρακτηρισμός ενός δανειολήπτη υψηλού κινδύνου ως δανειολήπτη χαμηλού κινδύνου θα μπορούσε να οδηγήσει σε σημαντικές οικονομικές ζημίες για τα δανειοδοτικά ιδρύματα, ενώ το αντίστροφο (άρνηση χορήγησης πίστωσης σε δανειολήπτη χαμηλού κινδύνου) επισύρει μικρότερη χρηματική ποινή. **Συχνά εφαρμόζονται προσεγγίσεις μάθησης ευαίσθητες ως προς το κόστος** , όπως η προσαρμογή του κατώτατου ορίου απόφασης ή η θέσπιση συντελεστών στάθμισης ποινής για ψευδώς αρνητικά, για να ληφθούν υπόψη αυτές οι άνισες δαπάνες .

Εκτός από την απόδοση, η **ερμηνευσιμότητα** του μοντέλου είναι κρίσιμη για την πρόβλεψη πιστωτικού κινδύνου, ειδικά σε ρυθμιζόμενους κλάδους όπως ο χρηματοπιστωτικός τομέας. Ενώ πολύπλοκα μοντέλα όπως **τυχαία δάση** ή **νευρωνικά δίκτυα** μπορεί να παρέχουν υψηλότερη προγνωστική ακρίβεια, απλούστερα μοντέλα όπως **η λογιστική παλινδρόμηση** ή **τα δέντρα αποφάσεων** συχνά ευνοούνται λόγω της διαφάνειας και της ευκολίας εξήγησής τους. Υπάρχει μια αυξανόμενη τάση προς την ανάπτυξη ερμηνεύσιμων μοντέλων μηχανικής μάθησης, ιδιαίτερα σε τομείς υψηλού ρίσκου όπως η βαθμολόγηση πιστοληπτικής ικανότητας .

1.4.3 Σύγκριση διαφορετικών μοντέλων

Μια συγκριτική αξιολόγηση διαφόρων μοντέλων, συμπεριλαμβανομένης της **λογιστικής παλινδρόμησης**, **των δέντρων αποφάσεων**, **των μηχανών διανυσμάτων υποστήριξης (SVM)** και των **νευρωνικών δικτύων**, είναι συχνά απαραίτητη για τον προσδιορισμό του μοντέλου με τις καλύτερες επιδόσεις για την πρόβλεψη πιστωτικού κινδύνου. Μελέτες έχουν δείξει ότι ενώ οι παραδοσιακές μέθοδοι όπως η λογιστική παλινδρόμηση εξακολουθούν να χρησιμοποιούνται ευρέως στα χρηματοπιστωτικά ιδρύματα λόγω της ερμηνευσιμότητάς τους, πιο σύνθετα μοντέλα όπως οι **μηχανές ενίσχυσης κλίσης (GBM)** και **τα τυχαία δάση** συχνά τα ξεπερνούν όσον αφορά την προγνωστική ακρίβεια. Για παράδειγμα, σε μια ολοκληρωμένη μελέτη συγκριτικής αξιολόγησης από **τους Lessmann et al. (2015)** [4] , τα μοντέλα συνόλων που βασίζονται σε δέντρα όπως **το XGBoost** και **τα τυχαία δάση** βρέθηκαν να ξεπερνούν τα παραδοσιακά μοντέλα λογιστικής παλινδρόμησης όσον αφορά τη βαθμολογία AUC-ROC και F1.

2. Μεθοδολογία έρευνας

2.1 Σχεδιασμός έρευνας

Ο ερευνητικός σχεδιασμός αυτής της μελέτης ακολουθεί μια **ποσοτική προσέγγιση**, αξιοποιώντας τεχνικές μηχανικής μάθησης για την πρόβλεψη του κινδύνου πιστωτικής αθέτησης χρησιμοποιώντας το σύνολο δεδομένων Home Credit Default Risk. Αυτός ο ποσοτικός ερευνητικός σχεδιασμός είναι δομημένος για να αξιολογήσει την αποτελεσματικότητα διαφόρων αλγορίθμων μηχανικής μάθησης, με ιδιαίτερη έμφαση στα **δίκτυα Kolmogorov-Arnold (KAN),** στην πρόβλεψη πιθανοτήτων αθέτησης δανείου. Αυτή η ενότητα θα περιγράφει λεπτομερώς τις διαδικασίες βήμα προς βήμα που εμπλέκονται στην ερευνητική διαδικασία, συμπεριλαμβανομένης της συλλογής δεδομένων, της προεπεξεργασίας, της μηχανικής χαρακτηριστικών, της επιλογής μοντέλων, της εκπαίδευσης και της επικύρωσης.

Η μελέτη ακολουθεί ένα **εποπτευόμενο πλαίσιο μάθησης**, όπου ο στόχος είναι να προβλεφθεί μια δυαδική μεταβλητή-στόχος (αθέτηση δανείου: ναι ή όχι) με βάση διάφορα χαρακτηριστικά εισόδου. Η έρευνα περιλαμβάνει πολλαπλά στάδια, ξεκινώντας με **την απόκτηση δεδομένων** από το σύνολο δεδομένων Home Credit Default Risk, ακολουθούμενη από μια εκτεταμένη φάση προεπεξεργασίας δεδομένων. Η προεπεξεργασία δεδομένων θα περιλαμβάνει το χειρισμό τιμών που λείπουν, την ανίχνευση ακραίων τιμών και την κανονικοποίηση των χαρακτηριστικών για να διασφαλιστεί ότι τα δεδομένα βρίσκονται σε βέλτιστη κατάσταση για μοντέλα μηχανικής μάθησης.

Η εξαγωγή χαρακτηριστικών θα είναι ένα άλλο κρίσιμο βήμα στον σχεδιασμό της έρευνας. Ο σκοπός της μηχανικής χαρακτηριστικών είναι να εξαγάγει τα πιο ενημερωτικά χαρακτηριστικά από το σύνολο δεδομένων, καθώς μελέτες έχουν δείξει ότι τα καλά κατασκευασμένα χαρακτηριστικά μπορούν να βελτιώσουν σημαντικά την απόδοση των προγνωστικών μοντέλων . Θα δημιουργηθούν νέα χαρακτηριστικά, όπως ο λόγος εισοδήματος προς δάνειο και η διάρκεια του πιστωτικού ιστορικού, ακολουθώντας προσεγγίσεις παρόμοιες με αυτές που περιγράφουν οι **Lessmann et al. (2015) [4].**
Στη συνέχεια, η μελέτη θα επικεντρωθεί στην **επιλογή μοντέλων**. Τα δίκτυα Kolmogorov-Arnold (KAN) θα είναι το κύριο μοντέλο που χρησιμοποιείται σε αυτή την ανάλυση λόγω της θεωρητικής του ικανότητας να προσεγγίζει μη γραμμικές συναρτήσεις με υψηλή ακρίβεια. Η έρευνα θα συγκρίνει την απόδοση του KAN με άλλα μοντέλα, συμπεριλαμβανομένης της λογιστικής παλινδρόμησης και των μηχανών ενίσχυσης κλίσης (GBM). Η επιλογή των μοντέλων για σύγκριση βασίζεται στην επικρατούσα χρήση τους στη μοντελοποίηση πιστωτικού κινδύνου.

Μετά την επιλογή του μοντέλου, η φάση **εκπαίδευσης και επικύρωσης** θα χρησιμοποιήσει τεχνικές διασταυρούμενης επικύρωσης για να διασφαλίσει ότι τα μοντέλα γενικεύονται καλά σε νέα δεδομένα. Θα εφαρμοστεί μια **μέθοδος διασταυρούμενης επικύρωσης k-fold (k-fold cross-validation)** για τον διαχωρισμό των δεδομένων σε σύνολα εκπαίδευσης και δοκιμών. Ο συντονισμός υπερπαραμέτρων θα πραγματοποιηθεί επίσης χρησιμοποιώντας **αναζήτηση πλέγματος (grid search).**
Συνοπτικά, αυτή η έρευνα έχει σχεδιαστεί για να διερευνήσει την προγνωστική δύναμη των δικτύων Kolmogorov-Arnold στην αξιολόγηση πιστωτικού κινδύνου, παρέχοντας παράλληλα μια συγκριτική ανάλυση με άλλους γνωστούς αλγόριθμους μηχανικής μάθησης. Η έρευνα θα καθοδηγείται από αυστηρές τεχνικές προεπεξεργασίας, μηχανικής χαρακτηριστικών και αξιολόγησης μοντέλων για τη διασφάλιση της ευρωστίας και της αξιοπιστίας των αποτελεσμάτων.

2.2 Επισκόπηση συνόλου δεδομένων

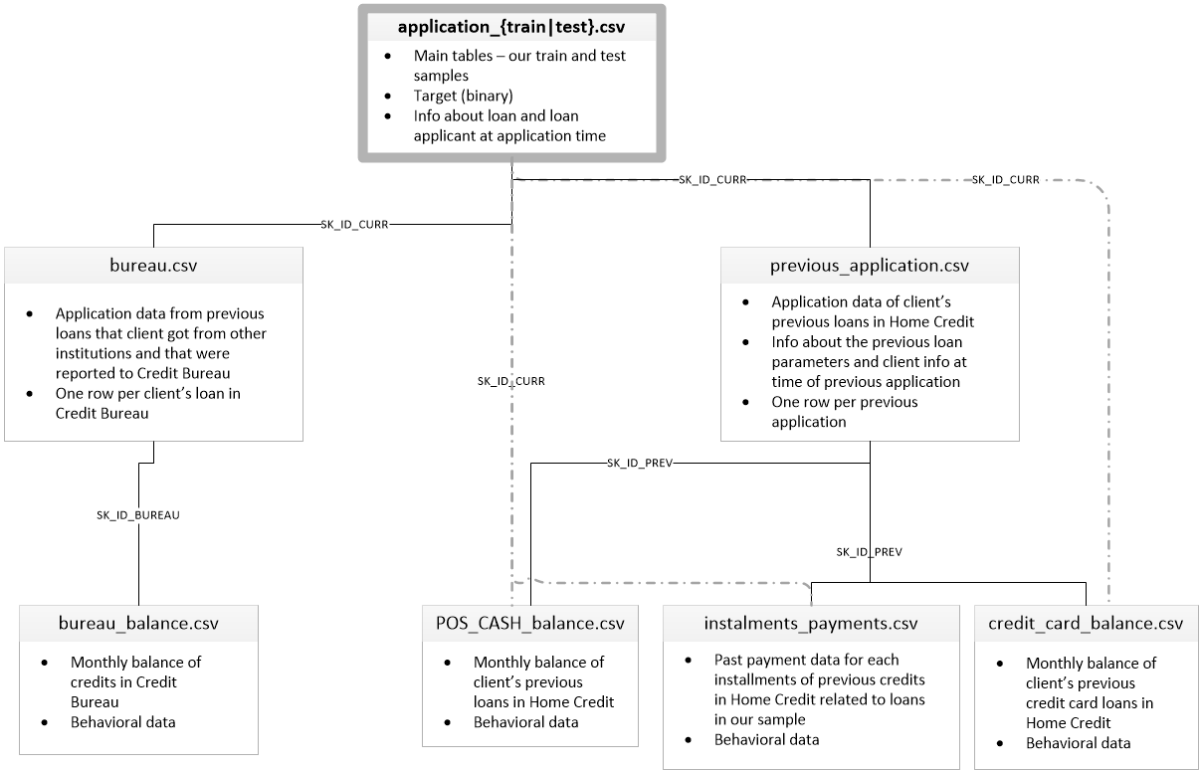
Το σύνολο δεδομένων που χρησιμοποιείται για αυτή την έρευνα είναι το Home Credit Default Risk dataset, το οποίο διατίθεται στο κοινό μέσω της πλατφόρμας Kaggle. Αυτό το σύνολο δεδομένων έχει σχεδιαστεί ειδικά για την ανάπτυξη μοντέλων μηχανικής μάθησης που προβλέπουν τον κίνδυνο αθέτησης δανείου με βάση ένα ευρύ φάσμα χαρακτηριστικών δανειολήπτη. Περιέχει εκτενείς πληροφορίες από αιτήσεις δανείων που υποβάλλονται στην Home Credit, ένα χρηματοπιστωτικό ίδρυμα που επικεντρώνεται στον δανεισμό ατόμων με περιορισμένη πρόσβαση σε παραδοσιακές τραπεζικές υπηρεσίες. Δεδομένης της ποικιλομορφίας των χαρακτηριστικών του, το σύνολο δεδομένων προσφέρει μια πλούσια και σύνθετη δομή για τη μοντελοποίηση πιστωτικού κινδύνου.

2.2.1 Σύνθεση συνόλου δεδομένων

Το σύνολο δεδομένων αποτελείται από πολλαπλά διασυνδεδεμένα αρχεία που παρέχουν συλλογικά μια λεπτομερή εικόνα του οικονομικού και προσωπικού υπόβαθρου κάθε αιτούντος. Αυτά τα αρχεία περιέχουν διάφορους τύπους δεδομένων, συμπεριλαμβανομένων δημογραφικών πληροφοριών, πιστωτικού ιστορικού, προηγούμενων αρχείων δανείων και συμπεριφορών πληρωμής. Ο κύριος πίνακας που χρησιμοποιήθηκε για τη μελέτη αυτή είναι ο «application_train.csv», ο οποίος περιέχει πάνω από **300.000 αιτήσεις δανείων** από πελάτες της Home Credit. Κάθε εγγραφή σε αυτόν τον φάκελο αντιστοιχεί σε μια μεμονωμένη αίτηση δανείου, με μια δυαδική μεταβλητή στόχου («TARGET») που υποδεικνύει εάν ο πελάτης αποπλήρωσε επιτυχώς το δάνειο («0») ή αθέτησε το δάνειο («1»).

Εκτός από το πρωτογενές σύνολο δεδομένων, τα ακόλουθα συμπληρωματικά αρχεία χρησιμοποιούνται για τη βελτίωση του μοντέλου πρόβλεψης παρέχοντας περισσότερο πλαίσιο για τους αιτούντες:
- **bureau.csv**: Αυτό το αρχείο περιλαμβάνει δεδομένα σχετικά με προηγούμενες πιστώσεις που έχουν λάβει οι πελάτες από άλλα ιδρύματα, προσφέροντας μια ευρύτερη προοπτική για το πιστωτικό ιστορικό του πελάτη πέρα από το Home Credit.
- **bureau_balance.csv**: Μια πιο λεπτομερής ανάλυση τους κατάστασης κάθε δανείου πιστωτικού γραφείου με την πάροδο του χρόνου, αναφέροντας την Κατάσταση του για κάθε μήνα για τον οποίο ήταν ενεργό.
- **previous_application.csv**: Αρχεία προηγούμενων αιτήσεων δανείου που υποβλήθηκαν από τον πελάτη με Home Credit, παρέχοντας πληροφορίες σχετικά με την προηγούμενη συμπεριφορά δανεισμού του πελάτη και το ιστορικό έγκρισης.
- **installments_payments.csv**: Περιέχει λεπτομερείς πληροφορίες σχετικά με τους πληρωμές δόσεων προηγούμενων δανείων, συμπεριλαμβανομένου του εάν οι πληρωμές έγιναν εγκαίρως ή καθυστέρησαν.
- **POS_CASH_balance.csv**: Πληροφορίες σχετικά με το υπόλοιπο και την κατάσταση προηγούμενων δανείων με μετρητά στο σημείο πώλησης.
- **credit_card_balance.csv**: Ιστορικά δεδομένα υπολοίπου πιστωτικών καρτών, συμπεριλαμβανομένων ιστορικών πληρωμών και υπολοίπου για κάθε μήνα.

Κάθε ένα από αυτά τα αρχεία προσθέτει ένα επίπεδο πολυπλοκότητας και βάθους στη διαδικασία μοντελοποίησης, επιτρέποντας τη δημιουργία χαρακτηριστικών που μπορούν να συλλάβουν τους μακροπρόθεσμες τάσεις στη συμπεριφορά των πελατών και να αξιολογήσουν τον κίνδυνο αποπληρωμής του δανείου με μεγαλύτερη ακρίβεια.



Διάγραμμα 2.1: Οι πίνακες δεδομένων και οι συσχετίσεις τους

2.2.2 Προκλήσεις με το σύνολο δεδομένων

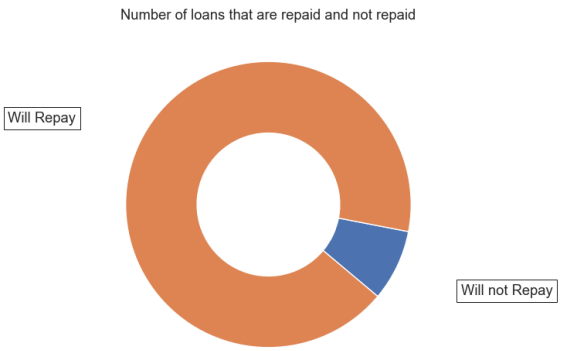
Παρά τον πλούτο του συνόλου δεδομένων για τον κίνδυνο αθέτησης εγχώριας κατανάλωσης, παρουσιάζει αρκετές προκλήσεις που πρέπει να αντιμετωπιστούν κατά τη διαδικασία προετοιμασίας των δεδομένων:

1. **Δεδομένα που λείπουν**: Ένα σημαντικό μέρος του συνόλου δεδομένων περιέχει τιμές που λείπουν, ιδιαίτερα σε στήλες που σχετίζονται με το πιστωτικό ιστορικό και τα στοιχεία απασχόλησης. Για παράδειγμα, τα χαρακτηριστικά «OCCUPATION_TYPE» και «EXT_SOURCE_1» έχουν μεγάλο ποσοστό καταχωρήσεων που λείπουν, γεγονός που θα μπορούσε να μεροληπτήσει το μοντέλο εάν δεν αντιμετωπιστεί σωστά. Τεχνικές όπως ο **καταλογισμός (imputation)**, όπου οι τιμές που λείπουν αντικαθίστανται από εκτιμήσεις, ή η απόρριψη στηλών με υπερβολικά ελλιπή δεδομένα μπορεί να είναι απαραίτητες.

	missing_count	missing_ratio
BASEMENTAREA_MEDI	179943	0.585160
BASEMENTAREA_AVG	179943	0.585160
BASEMENTAREA_MODE	179943	0.585160
EXT_SOURCE_1	173378	0.563811
NONLIVINGAREA_MEDI	169682	0.551792
NONLIVINGAREA_MODE	169682	0.551792
NONLIVINGAREA_AVG	169682	0.551792
ELEVATORS_MEDI	163891	0.532960
ELEVATORS_MODE	163891	0.532960
ELEVATORS_AVG	163891	0.532960

ΕΙΚΟΝΑ 2.1: Ποσοστό ελλιπών στοιχειών στον πίνακα application_train

2. **Μη ισορροπημένο σύνολο δεδομένων**: Η μεταβλητή-στόχος («TARGET») παρουσιάζει σοβαρές διαταραχές, με περίπου το **92%** των αιτούντων να αποπληρώνουν επιτυχώς τα δάνειά τους, ενώ μόνο περίπου **8%** αθετούν τις υποχρεώσεις τους. Αυτή η ανισορροπία μπορεί να οδηγήσει σε μεροληπτική απόδοση του μοντέλου, όπου το μοντέλο τείνει να προβλέπει την πλειοψηφική τάξη (μη προεπιλογή) πιο συχνά.



Διάγραμμα 2.2: Κατανομή μεταβλητής TARGET

3. **Feature Engineering**: Το σύνολο δεδομένων περιέχει ένα μείγμα κατηγορικών και αριθμητικών χαρακτηριστικών, πολλά από τα οποία πρέπει να μετασχηματιστούν πριν από την εκπαίδευση μοντέλου. Για παράδειγμα, κατηγορικά χαρακτηριστικά όπως «CODE_GENDER» και «NAME_EDUCATION_TYPE» απαιτούν κωδικοποίηση (π.χ. κωδικοποίηση one-hot ή κωδικοποίηση ετικετών). Η εξαγωγή χαρακτηριστικών είναι κρίσιμη για να διασφαλιστεί ότι το μοντέλο Kolmogorov-Arnold Networks (KANs) συλλαμβάνει σημαντικές σχέσεις μεταξύ μεταβλητών.

4. **Ακραίες τιμές**: Ορισμένα χαρακτηριστικά περιέχουν ακραίες τιμές που μπορούν να στρεβλώσουν τη διαδικασία εκμάθησης του μοντέλου. Για παράδειγμα, το χαρακτηριστικό «DAYS_EMPLOYED» περιέχει ανώμαλες τιμές όπου ορισμένοι αιτούντες καταγράφονται ως έχοντες τιμές που υπερβαίνουν τη διάρκεια ζωής του ανθρώπου. Αυτές οι ακραίες τιμές πρέπει είτε να διορθωθούν είτε να αφαιρεθούν.

2.3 Προεπεξεργασία δεδομένων

Ο στόχος της προεπεξεργασίας δεδομένων και της μηχανικής χαρακτηριστικών είναι η μετατροπή πρωτογενών δεδομένων σε δομημένη μορφή που μεγιστοποιεί την απόδοση των μοντέλων μηχανικής μάθησης. Σε αυτή τη μελέτη, το **σύνολο δεδομένων Home Credit Default Risk** υποβλήθηκε σε διάφορα στάδια προεπεξεργασίας, συμπεριλαμβανομένου του χειρισμού των τιμών που λείπουν, του εντοπισμού και της επεξεργασίας ακραίων τιμών, της κλιμάκωσης αριθμητικών χαρακτηριστικών και της κωδικοποίησης κατηγορικών μεταβλητών. Αυτά τα βήματα είναι κρίσιμα για να διασφαλιστεί ότι το σύνολο δεδομένων είναι καλά προετοιμασμένο και ότι τα μοντέλα πρόβλεψης μπορούν να εξαγάγουν σημαντικά μοτίβα από τα δεδομένα.

2.3.1 Χειρισμός δεδομένων που λείπουν

Ο χειρισμός των δεδομένων που λείπουν είναι ένα από τα πιο κρίσιμα βήματα προεπεξεργασίας σε οποιοδήποτε σύνολο δεδομένων και το σύνολο δεδομένων Home Credit περιέχει πολλά χαρακτηριστικά με ποικίλους βαθμούς έλλειψης. Η προσέγγιση των ελλειπόντων δεδομένων προσαρμόστηκε με βάση την ποσότητα των ελλείψεων σε κάθε στήλη:

1. **Κατάργηση γραμμών με μικρές ποσότητες δεδομένων που λείπουν**: Για στήλες όπου έλειπε ένα μικρό ποσοστό (κάτω του 5%) τιμών, οι επηρεαζόμενες γραμμές καταργήθηκαν. Ενώ θα μπορούσε να χρησιμοποιηθεί καταλογισμός, η κατάργηση αυτών των γραμμών βοηθά στην αποφυγή εισαγωγής δυνητικά ανακριβών τιμών, ειδικά όταν ο αριθμός των τιμών που λείπουν είναι αμελητέος σε σχέση με το μέγεθος του συνόλου δεδομένων.

2. **Κατάργηση στηλών με υπερβολικά ελλιπή δεδομένα**: Για στήλες όπου έλειπε περισσότερο από το 90% των δεδομένων, καταργήθηκε ολόκληρη η στήλη. Η διατήρηση τέτοιων στηλών θα παρείχε λίγες χρήσιμες πληροφορίες για το μοντέλο, καθώς θα αποτελούνταν κυρίως από ελλείπουσες ή τεκμαρτές τιμές. Η κατάργηση αυτών των στηλών απλοποιεί επίσης το σύνολο δεδομένων και μειώνει την πολυπλοκότητα του μοντέλου.

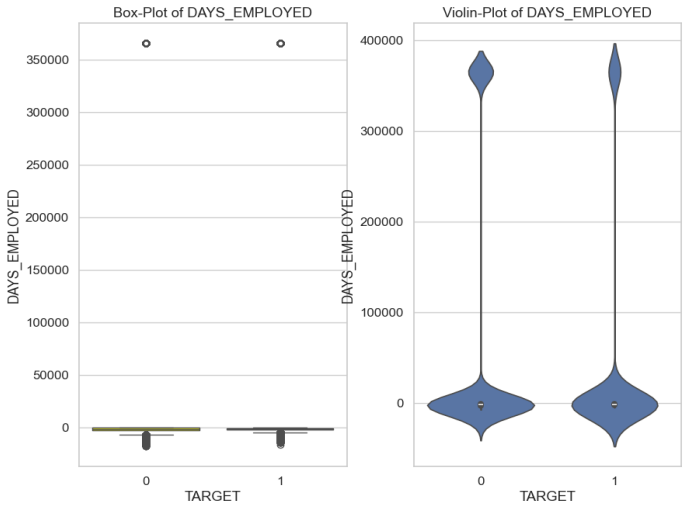
3. **Χειρισμός κατηγορικών μεταβλητών**: Για κατηγορικά χαρακτηριστικά με τιμές που λείπουν, οι καταχωρίσεις που λείπουν αντικαταστάθηκαν με μια ειδική κατηγορία με την ένδειξη **"XNA",** υποδεικνύοντας ότι τα δεδομένα δεν είναι διαθέσιμα. Αυτή η προσέγγιση επιτρέπει στα μοντέλα να αντιμετωπίζουν την έλλειψη ως ειδική κατηγορία χωρίς να στρεβλώνουν την κατηγορική δομή. Η χρήση ενός συμβόλου κράτησης θέσης για κατηγορίες που λείπουν αποτρέπει την απώλεια πληροφοριών και αποφεύγει τη μεροληψία που εισάγεται από μεθόδους καταλογισμού, όπως η αντικατάσταση λειτουργίας.

4. **Χειρισμός αριθμητικών μεταβλητών**: Για αριθμητικά χαρακτηριστικά, οι τιμές που λείπουν αντικαταστάθηκαν με **0**. Η απόφαση αυτή ελήφθη για να αποφευχθεί η εισαγωγή μεροληψίας που θα μπορούσε να προκύψει από στατιστικό τεκμαρτό υπολογισμό (όπως η αντικατάσταση των ελλειπόντων τιμών με τη μέση τιμή ή τη διάμεσο), η οποία θα μπορούσε να μεταβάλει την κατανομή των δεδομένων. Η αντικατάσταση των τιμών που λείπουν με 0 είναι μια απλή και διαφανής μέθοδος, ιδιαίτερα σε περιπτώσεις όπου τα ελλείποντα δεδομένα μπορεί να σημαίνουν την απουσία μιας συναλλαγής ή ενός γεγονότος, κάτι που έχει νόημα στη μοντελοποίηση πιστωτικού κινδύνου.

2.3.2 Ανίχνευση και διαχείριση ακραίων τιμών

Οι ακραίες τιμές που αποκλίνουν σημαντικά από άλλες παρατηρήσεις, μπορούν να έχουν δυσανάλογο αντίκτυπο στα μοντέλα μηχανικής μάθησης, ειδικά σε σύνολα δεδομένων που περιλαμβάνουν οικονομικά δεδομένα. Σε αυτή τη μελέτη, εντοπίστηκαν και αφαιρέθηκαν ακραίες τιμές για να διασφαλιστεί ότι τα μοντέλα δεν επηρεάστηκαν αδικαιολόγητα από ακραίες τιμές.

- **Ανίχνευση ακραίων τιμών**: Τα boxplots χρησιμοποιήθηκαν για τον εντοπισμό ακραίων τιμών σε αριθμητικές μεταβλητές όπως **DAYS_ENDDATE_FACT** και **DAYS_EMPLOYED**. Τα boxplots αντιπροσωπεύουν οπτικά την κατανομή των δεδομένων επισημαίνοντας τυχόν τιμές που βρίσκονται εκτός 1,5 φορές του διατεταρτημοριακού εύρους (IQR), καθιστώντας τα ιδανικά για τον εντοπισμό ακραίων τιμών. Αυτά τα γραφήματα είναι ιδιαίτερα χρήσιμα για τον εντοπισμό ακραίων τιμών που θα μπορούσαν να στρεβλώσουν τα αποτελέσματα των μοντέλων μηχανικής μάθησης.



Διάγραμμα 2.3: Boxplot και violinplot της μεταβλητής Days_Employed σε σχέση με την μεταβλητή target

- **Επεξεργασία ακραίων τιμών**: Οι ακραίες τιμές που εντοπίστηκαν μέσω boxplots αφαιρέθηκαν εξ ολοκλήρου από το σύνολο δεδομένων. Η αφαίρεση των ακραίων τιμών διασφαλίζει ότι τα μοντέλα εκπαιδεύονται σε δεδομένα που είναι αντιπροσωπευτικά των τυπικών αιτούντων δάνεια, αντί να στρεβλώνονται από ακραίες περιπτώσεις. Αυτή η μέθοδος αντιμετώπισης ακραίων τιμών είναι κοινή στα σύνολα οικονομικών δεδομένων, καθώς οι ακραίες τιμές μπορούν να στρεβλώσουν τις προβλέψεις των μοντέλων και να μειώσουν την ακρίβεια. Με την αφαίρεσή τους, το σύνολο δεδομένων γίνεται πιο ισορροπημένο, οδηγώντας σε βελτιωμένη γενίκευση μοντέλων.

2.3.3 Κλιμάκωση χαρακτηριστικών

Η κλιμάκωση χαρακτηριστικών είναι ζωτικής σημασίας στη μηχανική μάθηση, καθώς διασφαλίζει ότι τα χαρακτηριστικά με διαφορετικά εύρη δεν επηρεάζουν δυσανάλογα τις προβλέψεις του μοντέλου. Για παράδειγμα, σε αυτό το σύνολο δεδομένων, μεταβλητές όπως **AMT_INCOME_TOTAL** (εισόδημα) και **DAYS_EMPLOYED** (διάρκεια απασχόλησης) έχουν πολύ διαφορετικά εύρη.

- **Μέθοδος κλιμάκωσης**: Η κλιμάκωση χαρακτηριστικών εφαρμόστηκε διαιρώντας κάθε αριθμητικό χαρακτηριστικό με τη μεγαλύτερη απόλυτη τιμή του, ομαλοποιώντας αποτελεσματικά το εύρος όλων των μεταβλητών ώστε να βρίσκονται εντός **[-1, 1]**. Αυτή η μέθοδος, γνωστή ως **κλιμάκωση max-abs**, διασφαλίζει ότι όλα τα χαρακτηριστικά συμβάλλουν εξίσου στο μοντέλο, διατηρώντας παράλληλα τις σχετικές σχέσεις μεταξύ των σημείων δεδομένων. Η κλιμάκωση max-abs είναι ιδιαίτερα επωφελής σε μοντέλα μηχανικής μάθησης που βασίζονται σε τεχνικές βελτιστοποίησης με βάση την κλίση, όπως τα δίκτυα Kolmogorov-Arnold, καθώς εμποδίζει χαρακτηριστικά με μεγαλύτερο εύρος να κυριαρχήσουν στη διαδικασία μάθησης.

Η κλιμάκωση των δεδομένων σε εύρος [-1, 1] είναι ιδανική όταν ασχολούμαστε με σύνολα δεδομένων που περιέχουν τόσο θετικές όσο και αρνητικές τιμές, καθώς διατηρεί το πρόσημο των δεδομένων και δεν παραμορφώνει το σχετικό μέγεθος των αρχικών μεταβλητών. Διασφαλίζοντας ότι

κάθε χαρακτηριστικό είναι σε συγκρίσιμη κλίμακα, μπορούμε να αποτρέψουμε το μοντέλο από το να είναι μεροληπτικό προς χαρακτηριστικά με μεγαλύτερα αριθμητικά εύρη.

2.3.4 Κωδικοποίηση κατηγορικών μεταβλητών

Το σύνολο δεδομένων περιέχει πολλές κατηγορικές μεταβλητές, όπως **CODE_GENDER**, **NAME_EDUCATION_TYPE** και **OCCUPATION_TYPE**, οι οποίες πρέπει να μετατραπούν σε αριθμητική μορφή για χρήση σε μοντέλα μηχανικής μάθησης. Σε αυτή τη μελέτη, η κωδικοποίηση ετικετών χρησιμοποιήθηκε για κατηγορικά χαρακτηριστικά.

- **Κωδικοποίηση ετικετών**: Σε αντίθεση με **την κωδικοποίηση one hot**, η οποία δημιουργεί πολλές δυαδικές στήλες για κάθε κατηγορία, η κωδικοποίηση ετικετών εκχωρεί έναν μοναδικό ακέραιο αριθμό σε κάθε κατηγορία. Η κωδικοποίηση ετικετών επιλέχθηκε επειδή **η κωδικοποίηση one hot** θα εισήγαγε σημαντικό αριθμό νέων στηλών, ιδιαίτερα για χαρακτηριστικά με υψηλή πληθικότητα, όπως το **OCCUPATION_TYPE**. Αυτό θα αυξήσει τη διαστασιολόγηση του συνόλου δεδομένων και ενδεχομένως θα μειώσει την αποτελεσματικότητα και τις επιδόσεις των μοντέλων μηχανικής μάθησης. Η κωδικοποίηση ετικετών είναι υπολογιστικά αποτελεσματική και λειτουργεί καλά όταν δεν υπάρχει τακτική σχέση μεταξύ των κατηγοριών, όπως συμβαίνει με τα περισσότερα κατηγορικά χαρακτηριστικά στο σύνολο δεδομένων Home Credit.

Η χρήση κωδικοποίησης ετικετών βοηθά στη διατήρηση ενός διαχειρίσιμου αριθμού χαρακτηριστικών, μειώνοντας τον υπολογιστικό φόρτο στα μοντέλα χωρίς να χάνονται πολύτιμες κατηγορικές πληροφορίες. Ενώ η κωδικοποίηση one hot είναι κατάλληλη για μοντέλα που μπορούν να χειριστούν δεδομένα υψηλών διαστάσεων, η κωδικοποίηση ετικετών προσφέρει μια πιο αποτελεσματική εναλλακτική λύση για πολύπλοκα σύνολα δεδομένων υψηλής πληθικότητας.

2.3.5 Εξαγωγή χαρακτηριστικών

Η εξαγωγή χαρακτηριστικών διαδραματίζει κεντρικό ρόλο στην ενίσχυση της προγνωστικής ισχύος των μοντέλων μηχανικής μάθησης, ιδιαίτερα όταν πρόκειται για σύνθετα σύνολα δεδομένων όπως το **Home Credit Default Risk dataset**. Αυτή η διαδικασία περιλαμβάνει τη δημιουργία νέων δυνατοτήτων από υπάρχοντα δεδομένα για να βοηθήσει το μοντέλο να καταγράψει πιο λεπτές σχέσεις. Για αυτή τη διατριβή, πολλά νέα χαρακτηριστικά σχεδιάστηκαν σε βασικούς πίνακες του συνόλου δεδομένων, συμπεριλαμβανομένων των **application**, **bureau**, **credit_card_balance**, **installments_payments** και **previous_application**. Αυτά τα χαρακτηριστικά σχεδιάστηκαν για να αντιπροσωπεύουν κρίσιμους χρηματοοικονομικούς δείκτες και άλλες μετρήσεις που μπορούν να παρέχουν πληροφορίες σχετικά με την ικανότητα ενός δανειολήπτη να αποπληρώσει τα δάνειά του.

Ο πίνακας **application** περιέχει πληροφορίες σχετικά με τους αιτούντες δάνειο και διάφορα νέα χαρακτηριστικά σχεδιάστηκαν για να αντιπροσωπεύουν καλύτερα την οικονομική κατάσταση των αιτούντων:

- **ANNUITY_INCOME_PERCENT**: Αυτό το χαρακτηριστικό δημιουργήθηκε διαιρώντας το ποσό της προσόδου με το συνολικό εισόδημα του αιτούντος. Αντιπροσωπεύει το ποσοστό του εισοδήματος που αφιερώνεται στις πληρωμές προσόδων, παρέχοντας πληροφορίες σχετικά με το βάρος του δανείου στα οικονομικά του αιτούντος. Μια υψηλότερη αξία μπορεί να υποδηλώνει υψηλότερο κίνδυνο, καθώς υποδηλώνει ότι σημαντικό μέρος του εισοδήματος του αιτούντος δεσμεύεται για την αποπληρωμή του δανείου.

- **FAMILY_COUNT_INCOME_PERCENT**: Αυτό το χαρακτηριστικό διαιρεί το συνολικό εισόδημα με τον αριθμό των μελών της οικογένειας. Αντιπροσωπεύει το κατά κεφαλήν εισόδημα της οικογένειας και δίνει μια σαφέστερη εικόνα της οικονομικής ευημερίας των μεγαλύτερων οικογενειών. Ένα χαμηλότερο κατά κεφαλήν εισόδημα θα μπορούσε να συνεπάγεται υψηλότερο κίνδυνο αθέτησης υποχρεώσεων, καθώς το νοικοκυριό έχει περισσότερα εξαρτώμενα άτομα να στηρίξει.

- **CREDIT_TERM (ANNUITY/CREDIT)**: Αυτό το χαρακτηριστικό υπολογίζει τη διάρκεια της πίστωσης διαιρώντας τις συνολικές πληρωμές προσόδων με το ποσό της πίστωσης. Αντικατοπτρίζει τη διάρκεια κατά την οποία αναμένεται να αποπληρωθεί το δάνειο, η οποία είναι χρήσιμη για την κατανόηση των μακροπρόθεσμων οικονομικών δεσμεύσεων του δανειολήπτη.

- **CREDIT_INCOME_PERCENT**: Αυτό το χαρακτηριστικό είναι ο λόγος του ποσού πίστωσης προς το συνολικό εισόδημα του αιτούντος. Μετρά το σχετικό μέγεθος του δανείου σε σύγκριση με τα κέρδη του δανειολήπτη, παρέχοντας έναν δείκτη του χρέους του αιτούντος.

- **BIRTH_EMPLOYED_PERCENT**: Δημιουργήθηκε διαιρώντας τον αριθμό των ημερών απασχόλησης του αιτούντος με την ηλικία του σε ημέρες (DAYS_EMPLOYED/DAYS_BIRTH), αυτό το χαρακτηριστικό μετρά το ποσοστό της ζωής του αιτούντος που πέρασε στην απασχόληση. Μπορεί να υποδηλώνει σταθερότητα, καθώς τα άτομα με μεγαλύτερο ιστορικό απασχόλησης σε σχέση με την ηλικία τους συχνά θεωρούνται πιο οικονομικά σταθερά.

- **CREDIT_ANNUITY_PERCENT**: Αυτό το χαρακτηριστικό είναι ο λόγος του πιστωτικού ποσού προς την πρόσοδο. Δείχνει πόσο από το δάνειο καλύπτεται από κάθε πληρωμή προσόδου, παρέχοντας πληροφορίες για τους όρους και τη δομή του δανείου.

- **CHILDREN_COUNT_INCOME_PERCENT**: Αυτό το χαρακτηριστικό διαιρεί το συνολικό εισόδημα με τον αριθμό των παιδιών στο νοικοκυριό. Αξιολογεί πόσο εισόδημα είναι διαθέσιμο ανά παιδί, το οποίο αποτελεί χρήσιμο δείκτη οικονομικού άγχους, ειδικά για τις πολύτεκνες οικογένειες.

Ο πίνακας **bureau** περιέχει δεδομένα σχετικά με το προηγούμενο πιστωτικό ιστορικό του αιτούντος από άλλα χρηματοπιστωτικά ιδρύματα. Αρκετά χαρακτηριστικά σχεδιάστηκαν από αυτόν τον πίνακα για να συλλάβουν σημαντικές πτυχές της πιστωτικής συμπεριφοράς του αιτούντος:

- **CREDIT_DURATION**: Αυτή η δυνατότητα υπολογίζει τη διάρκεια κάθε πίστωσης αφαιρώντας την ημερομηνία λήξης της πίστωσης από την ημερομηνία έναρξης. Ένα μεγαλύτερο πιστωτικό ιστορικό μπορεί να είναι ενδεικτικό της χρηματοπιστωτικής σταθερότητας, αλλά θα μπορούσε επίσης να υποδηλώνει υψηλότερη έκθεση στο χρέος.

- **DEBT_PERCENTAGE**: Αυτό το χαρακτηριστικό είναι ο λόγος του συνολικού χρέους προς το συνολικό ποσό πίστωσης. Παρέχει πληροφορίες σχετικά με το ποσοστό της δανειακής πίστωσης που εξακολουθεί να εκκρεμεί, όπου ένα υψηλότερο ποσοστό μπορεί να σηματοδοτήσει αυξημένο χρηματοοικονομικό κίνδυνο.

- **DEBT_CREDIT_DIFF**: Αυτή η δυνατότητα υπολογίζει τη διαφορά μεταξύ της συνολικής πίστωσης και του συνολικού χρέους. Μια μεγαλύτερη διαφορά μπορεί να υποδηλώνει ότι ο αιτών διαχειρίζεται καλά το χρέος του, ενώ μια μικρότερη διαφορά ή αρνητική αξία θα μπορούσε να υποδηλώνει υψηλότερο κίνδυνο αθέτησης υποχρεώσεων.

Ο πίνακας **credit_card_balance** παρακολουθεί τις συναλλαγές και τα υπόλοιπα πιστωτικών καρτών των αιτούντων. Δημιουργήθηκαν διάφοροι δείκτες για την αξιολόγηση της οικονομικής υγείας των αιτούντων με βάση τη χρήση της πιστωτικής τους κάρτας:

- **Percentage_of_limit_drawn**: Αυτή η λειτουργία διαιρεί το υπόλοιπο της πιστωτικής κάρτας με το πιστωτικό όριο, αντιπροσωπεύοντας το ποσό της διαθέσιμης πίστωσης που έχει χρησιμοποιηθεί. Ένα υψηλότερο ποσοστό θα μπορούσε να υποδηλώνει επικίνδυνη συμπεριφορά, καθώς υποδηλώνει ότι ο αιτών είναι κοντά στο να μεγιστοποιήσει το πιστωτικό του όριο.

- **Percentage_of_min_payment**: Αυτή η δυνατότητα υπολογίζει το ποσοστό της ελάχιστης πληρωμής διαιρώντας την πραγματική πληρωμή με την ελάχιστη απαιτούμενη πληρωμή. Οι αιτούντες που πληρώνουν σταθερά μόνο το ελάχιστο ενδέχεται να είναι πιο πιθανό να αθετήσουν τις υποχρεώσεις τους.

- **Percentage_of_receivable_principal**: Αυτό το χαρακτηριστικό είναι ο λόγος του εισπρακτέου κεφαλαίου προς το υπόλοιπο της πιστωτικής κάρτας, δείχνοντας πόσο από το οφειλόμενο υπόλοιπο είναι στην πραγματικότητα κεφάλαιο, σε αντίθεση με τόκους ή τέλη.

- **Amount_per_drawing**: Αυτή η λειτουργία υπολογίζει το μέσο ποσό που λαμβάνεται σε κάθε συναλλαγή, το οποίο μπορεί να δώσει πληροφορίες σχετικά με τα πρότυπα δαπανών του αιτούντος. Μεγαλύτερες μέσες αναλήψεις θα μπορούσαν να υποδηλώνουν μια πιο σημαντική εξάρτηση από την πίστωση, η οποία θα μπορούσε να σηματοδοτήσει οικονομική πίεση.

Στον **πίνακα installments_payments** καταγράφονται οι δόσεις που κατέβαλαν οι αιτούντες για προηγούμενα δάνεια. Δημιουργήθηκαν βασικά χαρακτηριστικά για τη μέτρηση του έγκαιρου χαρακτήρα και της συνέπειας αυτών των πληρωμών:

- **DAYS_LATE_PAYMENT**: Αυτή η λειτουργία υπολογίζει τον αριθμό των ημερών καθυστέρησης μιας πληρωμής, συγκρίνοντας την προγραμματισμένη ημερομηνία πληρωμής με την πραγματική ημερομηνία πληρωμής. Οι συνεχείς καθυστερήσεις πληρωμών θα μπορούσαν να υποδηλώνουν μεγαλύτερη πιθανότητα αθέτησης μελλοντικών δανείων.

- **PAYMENT_AMOUNT_MISSING**: Αυτή η λειτουργία υπολογίζει το ποσό που λείπει από μια πληρωμή (δηλαδή, αν ο η πραγματική πληρωμή είναι λιγότερη από την ελάχιστη αναγκαία). Οι μη καταβληθείσες πληρωμές αποτελούν ισχυρούς δείκτες οικονομικής δυσχέρειας και αυξημένου κινδύνου.

Ο πίνακας **previous_application** περιέχει πληροφορίες σχετικά με τις προηγούμενες αιτήσεις δανείου του αιτούντος. Αρκετά χαρακτηριστικά σχεδιάστηκαν για να συλλάβουν τα αποτελέσματα και τις οικονομικές συμπεριφορές από προηγούμενες αιτήσεις πίστωσης:

- **AMT_DECLINED**: Αυτή η λειτουργία παρακολουθεί τη διαφορά μεταξύ του ποσού πίστωσης που ζητήθηκε και του ποσού που χορηγήθηκε. Μια σημαντική απόκλιση μπορεί να υποδηλώνει ότι οι προηγούμενες αιτήσεις του αιτούντος θεωρήθηκαν επικίνδυνες από άλλους δανειστές.

- **AMT_CREDIT_GOODS_DIFF**: Αυτή η δυνατότητα υπολογίζει τη διαφορά μεταξύ του ποσού της πίστωσης και της αξίας των αγαθών ή των υπηρεσιών που αγοράστηκαν με το δάνειο. Αυτή η διαφορά μπορεί να δώσει μια εικόνα για το αν ο αιτών τείνει να υπερδανείζεται σε σχέση με τις πραγματικές ανάγκες του, γεγονός που μπορεί να υποδηλώνει οικονομική ανευθυνότητα.

- **CREDIT_DOWNPAYMENT_RATIO**: Αυτό το χαρακτηριστικό είναι ο λόγος της προκαταβολής προς το συνολικό ποσό πίστωσης. Ένας υψηλότερος δείκτης προκαταβολής μπορεί να υποδηλώνει χαμηλότερο κίνδυνο αθέτησης υποχρεώσεων, καθώς ο αιτών έχει περισσότερα ίδια κεφάλαια στο δάνειο.

2.3.6 Συγκέντρωση βασικών πινάκων

Η συγκέντρωση δεδομένων από πολλαπλούς πίνακες είναι ζωτικής σημασίας για την καταγραφή διαφόρων πτυχών του οικονομικού ιστορικού και της συμπεριφοράς ενός δανειολήπτη. Συνοψίζοντας βασικά χαρακτηριστικά από διαφορετικές πηγές δεδομένων και ενσωματώνοντάς τα σε ένα ολοκληρωμένο σύνολο δεδομένων, διασφαλίζουμε ότι το μοντέλο μηχανικής μάθησης έχει πρόσβαση σε λεπτές πληροφορίες, οδηγώντας σε καλύτερες προβλέψεις. Οι ακόλουθες συγκεντρώσεις πραγματοποιήθηκαν στους πίνακες, **bureau_balance**, **bureau**, **POS_cash_balance**, **credit_card_balance**, **installments_payments** και **previous_application** για την εξαγωγή σημαντικών πληροφοριών.

Ο πίνακας **bureau_balance,** ο οποίος παρακολουθεί τη μηνιαία πιστωτική κατάσταση για κάθε αιτούντα, συγκεντρώθηκε δημιουργώντας έναν πίνακα διασταύρωσης της **δυνατότητας STATUS**. Αυτός ο διασταυρούμενος πίνακας καταγράφει τη συχνότητα κάθε κατάστασης δανείου (π.χ. τρέχουσα, ληξιπρόθεσμη κ.λπ.) για κάθε αιτούντα. Αυτό στη συνέχεια **ενώθηκε** αριστερά με τον **πίνακα του bureau**, επιτρέποντάς μας να συνδέσουμε συνοπτικά δεδομένα πιστωτικής κατάστασης με τα κύρια αρχεία δανείων.

Ο πίνακας **bureau** παρέχει λεπτομέρειες σχετικά με το πιστωτικό ιστορικό ενός αιτούντος σε άλλα χρηματοπιστωτικά ιδρύματα. Βασικά συγκεντρωτικά χαρακτηριστικά περιελάμβαναν:

- **CUSTOMER_LOAN_COUNT**: Ο συνολικός αριθμός δανείων που είχε ο αιτών με εξωτερικούς δανειστές.
- **CREDIT_ACTIVE crosstable**: Ένας ενδιάμεσος πίνακας για το αν ένα πιστωτικό όριο είναι ενεργό, κλειστό ή σε αθέτηση.
- **DAYS_CREDIT_mean, μέγιστη, ελάχιστη**: Αντιπροσωπεύει τις μέσες, μέγιστες και ελάχιστες ημέρες από τη λήψη της πίστωσης.
- **AMT_CREDIT_SUM_mean, μέγιστη, ελάχιστη**: Αποτυπώνει το συνολικό πιστωτικό άνοιγμα του αιτούντος, συγκεντρωτικό ώστε να αντικατοπτρίζει διάφορα μέτρα (μέση τιμή, μέγιστη, ελάχιστη).

Αυτά τα συγκεντρωτικά στοιχεία **ενώθηκαν** αριστερά με τον πινακα **application**, συνδέοντας το εξωτερικό πιστωτικό ιστορικό με το προφίλ του αιτούντος.

Στον πίνακα **POS_cash_balance,** ο οποίος παρακολουθεί τις πληρωμές για δάνεια δόσεων στο σημείο πώλησης, συγκεντρώσαμε τα ακόλουθα χαρακτηριστικά:

- **SK_DPD_mean, μέγιστη**: Μέσες και μέγιστες ημέρες καθυστέρησης.
- **SK_DPD_DEF_mean, μέγ:** Ημέρες καθυστέρησης κατά τις οποίες παρουσιάστηκε η αθέτηση.

Στη συνέχεια, τα συγκεντρωτικά χαρακτηριστικά **ενώθηκαν** αριστερά με τον **πίνακα previous_application**, επιτρέποντας την ανάλυση των προτύπων πληρωμών που σχετίζονται με προηγούμενα δάνεια.

Ο πίνακας **credit_card_balance** καταγράφει τις συναλλαγές με πιστωτικές κάρτες.

- **AMT_BALANCE_mean**: Η μέση ισορροπία με την πάροδο του χρόνου.
- **Percentage_of_limit_drawn_mean**: Το μέσο ποσοστό του πιστωτικού ορίου που αναλαμβάνεται.
- **Percentage_of_min_payment_mean**: Το μέσο ποσοστό της ελάχιστης πληρωμής που πραγματοποίησε ο αιτών.
- **SK_DPD_mean, μέγ.**: Ο αριθμός των ημερών καθυστέρησης.

Αυτά τα συγκεντρωτικά στοιχεία ενώθηκαν αριστερά με τον **πίνακα previous_application** για να ενσωματώσουν τη συμπεριφορά της πιστωτικής κάρτας στο πλαίσιο της αίτησης δανείου.

Ο πίνακας **installments_payments** περιέχει στοιχεία για προηγούμενες πληρωμές δόσεων δανείου. Τα βασικά συγκεντρωτικά χαρακτηριστικά περιλαμβάνουν:

- **DAYS_LATE_PAYMENT_mean**: Οι μέσες ημέρες καθυστέρησης μιας πληρωμής.
- **PAYMENT_AMOUNT_MISSING_mean**: Το μέσο ποσό πληρωμής που λείπει.

Αυτά τα χαρακτηριστικά **ενώθηκαν** αριστερά με τον **πίνακα previous_application** για να παρέχουν πληροφορίες σχετικά με τη συνέπεια της συμπεριφοράς αποπληρωμής δανείου.

Ο πίνακας **previous_application** περιέχει πληροφορίες σχετικά με προηγούμενες αιτήσεις πίστωσης. Δημιουργήθηκαν διάφορα συγκεντρωτικά στοιχεία, μεταξύ των οποίων:

- Πίνακες διασταύρωσης για **NAME_CONTRACT_TYPE**, **NAME_CONTRACT_STATUS** και **CODE_REJECT_REASON**, που καταγράφουν τους τύπους δανείων, την κατάστασή τους και τους λόγους απόρριψης.
- Συγκεντρωτικά χαρακτηριστικά όπως **AMT_ANNUITY_mean, max, min**, **AMT_APPLICATION_mean, max, min, AMT_BALANCE_mean_max, mean, min, AMT_CREDIT_max, mean, min** και **AMT_GOODS_PRICE_mean, max, min** καταγράφουν βασικές οικονομικές μετρήσεις που σχετίζονται με προηγούμενες αιτήσεις δανείων.
- Χαρακτηριστικά όπως **DAYS_FIRST_DRAWING_max, min** και **DAYS_LAST_DUE_max, min** συνοψίζουν σημαντικά χρονικά πλαίσια στη διαδικασία δανεισμού.
- Χαρακτηριστικά όπως **AMT_DECLINED_max, min, mean** αντιπροσωπεύουν ποσά που απορρίφθηκαν σε προηγούμενες αιτήσεις δανείου.
- Επιπλέον, οι **AMT_CREDIT_GOODS_DIFF_max, min, mean** και **CREDIT_DOWNPAYMENT_RATIO_max, min, mean συγκεντρώθηκαν** για να εκτιμηθούν οι διαφορές μεταξύ των πιστωτικών ποσών και της αξίας των αγαθών, καθώς και ο λόγος της προκαταβολής προς την πίστωση.

Αυτά τα συγκεντρωτικά στοιχεία ενώθηκαν αριστερά με τον **πίνακα αιτήσεων**, επιτρέποντάς μας να ενσωματώσουμε λεπτομερείς πληροφορίες σχετικά με προηγούμενες αιτήσεις δανείων στο κύριο σύνολο δεδομένων.

2.4 Επιλογή μοντέλου

Η επιλογή μοντέλων είναι μια κρίσιμη πτυχή της οικοδόμησης ενός συστήματος πρόβλεψης για τον κίνδυνο αθέτησης δανείων. Για αυτή τη μελέτη, επικεντρωθήκαμε σε τρία μοντέλα μηχανικής μάθησης: **Kolmogorov-Arnold Network (KAN), Logistic Regression** και **XGBoost**. Κάθε ένα από αυτά τα μοντέλα έχει διακριτά πλεονεκτήματα στη διαχείριση δομημένων χρηματοοικονομικών δεδομένων και επιλέχθηκε με βάση το θεωρητικό τους υπόβαθρο και την πρακτική επιτυχία στην πρόβλεψη πιστωτικού κινδύνου.

2.4.1 Δίκτυο Kolmogorov-Arnold (KAN)

Τα δίκτυα Kolmogorov-Arnold (KAN) είναι μια σχετικά νέα κατηγορία νευρωνικών δικτύων που βασίζεται στο θεώρημα υπέρθεσης Kolmogorov-Arnold, το οποίο υποθέτει ότι οποιαδήποτε συνεχής συνάρτηση μπορεί να αναπαρασταθεί ως άθροισμα απλούστερων μονομεταβλητών συναρτήσεων. Στο πλαίσιο της πρόβλεψης πιστωτικού κινδύνου, τα δίκτυα KAN προσφέρουν τη δυνατότητα να χειρίζονται αποτελεσματικά τις μη γραμμικές σχέσεις μεταξύ των χαρακτηριστικών, διατηρώντας παράλληλα την ερμηνευσιμότητα σε σύγκριση με τα παραδοσιακά βαθιά νευρωνικά δίκτυα. Τα KAN έχουν αποδειχθεί ότι αποδίδουν καλά σε οικονομικά πλαίσια, ειδικά όταν υπάρχουν πολύπλοκες αλληλεπιδράσεις μεταξύ μεταβλητών.

Τα μοντέλα KAN μειώνουν επίσης την ανάγκη για εκτεταμένη μηχανική χαρακτηριστικών, καθώς είναι σε θέση να μάθουν εξαιρετικά μη γραμμικές σχέσεις απευθείας από τα δεδομένα. Αυτό το χαρακτηριστικό είναι απαραίτητο για τα οικονομικά δεδομένα, όπου οι αλληλεπιδράσεις μεταξύ χαρακτηριστικών όπως το εισόδημα, το πιστωτικό ιστορικό και το μέγεθος του δανείου ενδέχεται να μην ακολουθούν απλά γραμμικά μοτίβα .

2.4.2 Λογιστική παλινδρόμηση

Η λογιστική παλινδρόμηση είναι ένα από τα πιο καθιερωμένα μοντέλα σε προβλήματα δυαδικής ταξινόμησης, καθιστώντας το μια δημοφιλή επιλογή για την πρόβλεψη πιστωτικού κινδύνου. Μοντελοποιεί την πιθανότητα αθέτησης ως συνάρτηση γραμμικών συνδυασμών των χαρακτηριστικών εισόδου. Παρά την απλότητά του, το Logistic Regression παρέχει ερμηνεύσιμα

αποτελέσματα, τα οποία είναι εξαιρετικά πολύτιμα σε ρυθμιζόμενες βιομηχανίες όπως η χρηματοδότηση, όπου η διαφάνεια είναι απαραίτητη .

Αυτό το μοντέλο υποθέτει μια γραμμική σχέση μεταξύ των ανεξάρτητων μεταβλητών και των λογαριθμικών αποδόσεων της εξαρτημένης μεταβλητής (αθέτηση ή μη αθέτηση). Ενώ αυτή η υπόθεση μπορεί να περιορίσει την αποτελεσματικότητά της με εξαιρετικά μη γραμμικά δεδομένα, η ευρωστία και η ευκολία ερμηνείας της την καθιστούν ένα αξιόπιστο βασικό μοντέλο για σύγκριση με πιο σύνθετα μοντέλα όπως το KAN και το XGBoost .

Μελέτες έχουν δείξει με συνέπεια ότι η Λογιστική Παλινδρόμηση αποδίδει ανταγωνιστικά σε εφαρμογές βαθμολόγησης πιστοληπτικής ικανότητας, ειδικά όταν συνδυάζεται με αποτελεσματική μηχανική χαρακτηριστικών και προεπεξεργασία δεδομένων.

2.4.3 XGBoost

Το XGBoost (Extreme Gradient Boosting) είναι μια ισχυρή τεχνική εκμάθησης συνόλου που έχει κερδίσει σημαντική έλξη σε διαγωνισμούς μηχανικής μάθησης και πρακτικές εφαρμογές λόγω της υψηλής απόδοσης και ευελιξίας της. Δημιουργεί μια σειρά από δέντρα αποφάσεων διαδοχικά, όπου κάθε δέντρο προσπαθεί να διορθώσει τα λάθη του προκατόχου του εστιάζοντας στις πιο δύσκολες να προβλεφθούν περιπτώσεις . Η XGBoost είναι ιδιαίτερα έμπειρη στο χειρισμό μεγάλων, δομημένων συνόλων δεδομένων με τιμές που λείπουν και θορυβώδη δεδομένα, τα οποία είναι κοινά σε σύνολα δεδομένων πιστωτικού κινδύνου.

Ένα από τα βασικά πλεονεκτήματα του XGBoost είναι η ικανότητά του να χειρίζεται τη σημασία των χαρακτηριστικών αυτόματα και ισχυρά μέσω του πλαισίου ενίσχυσης κλίσης. Αυτό είναι κρίσιμο για την πρόβλεψη του πιστωτικού κινδύνου, όπου ορισμένα χαρακτηριστικά (π.χ. πιστωτικό ιστορικό, ποσό δανείου) ενδέχεται να έχουν δυσανάλογο αντίκτυπο στον κίνδυνο αθέτησης υποχρεώσεων. Επιπλέον, το XGBoost προσφέρει εγγενή υποστήριξη για regularization, η οποία βοηθά στην αποφυγή υπερβολικής τοποθέτησης, μια συχνή πρόκληση όταν ασχολείστε με οικονομικά δεδομένα υψηλών διαστάσεων.

Πολλές μελέτες έχουν επισημάνει την απόδοση της XGBoost στην πιστοληπτική βαθμολόγηση. Οι Chen και Guestrin (2016) [8] έδειξαν ότι το XGBoost συχνά ξεπερνά τα παραδοσιακά μοντέλα μηχανικής μάθησης όπως η λογιστική παλινδρόμηση και τα δέντρα αποφάσεων όταν ασχολείται με μη ισορροπημένα σύνολα δεδομένων, κάτι που είναι σύνηθες στην πρόβλεψη αθέτησης δανείου.

Σύγκριση και αιτιολόγηση

Η επιλογή αυτών των τριών υποδειγμάτων παρέχει μια ολοκληρωμένη προσέγγιση για την πρόβλεψη του πιστωτικού κινδύνου. Το δίκτυο KAN χρησιμοποιείται για την καταγραφή σύνθετων, μη γραμμικών αλληλεπιδράσεων μεταξύ χαρακτηριστικών που ενδέχεται να μην είναι εμφανή μέσω παραδοσιακών μοντέλων. Η λογιστική παλινδρόμηση χρησιμεύει ως μια ισχυρή, ερμηνεύσιμη βάση, διασφαλίζοντας ότι η διαδικασία λήψης αποφάσεων του μοντέλου είναι διαφανής. Τέλος, το XGBoost επιλέγεται για την ικανότητά του να χειρίζεται μη ισορροπημένα σύνολα δεδομένων και πολύπλοκες αλληλεπιδράσεις χαρακτηριστικών, προσφέροντας παράλληλα υψηλή προγνωστική ακρίβεια.

Δοκιμάζοντας αυτά τα μοντέλα στο ίδιο σύνολο δεδομένων, στοχεύουμε να συγκρίνουμε την απόδοσή τους όσον αφορά την προγνωστική ακρίβεια, την ερμηνευσιμότητα και την υπολογιστική απόδοση. Αυτή η προσέγγιση θα μας επιτρέψει να καθορίσουμε το καταλληλότερο μοντέλο για αυτό το έργο πρόβλεψης πιστωτικού κινδύνου, διασφαλίζοντας τόσο την ακρίβεια όσο και τη διαφάνεια.

2.5 Εκπαίδευση και επικύρωση μοντέλων

Σε αυτή την ενότητα, εμβαθύνουμε στις διαδικασίες που χρησιμοποιούνται για την εκπαίδευση και την επικύρωση των μοντέλων που επιλέχθηκαν στην **Ενότητα 2.4**: Δίκτυα Kolmogorov-Arnold (KAN), Λογιστική παλινδρόμηση και XGBoost. Η εκπαιδευτική διαδικασία σχεδιάστηκε για να ελαχιστοποιήσει την υπερβολική τοποθέτηση, να εξασφαλίσει ισχυρή απόδοση σε άγνωστα δεδομένα και να παρέχει ακριβείς προβλέψεις κινδύνου αθέτησης δανείου. Τα επιλεγμένα μοντέλα εκπαιδεύτηκαν χρησιμοποιώντας **διασταυρούμενη επικύρωση K-fold** που είναι ενσωματωμένη σε όλα τα παραπάνω μοντέλα και βελτιστοποιήθηκαν μέσω **συντονισμού υπερπαραμέτρων**, με προσεκτική εξέταση των μετρήσεων απόδοσής τους, AUC-ROC και πίνακα σύγχυσης.

2.5.1 Στρατηγική διαχωρισμού δεδομένων

Το σύνολο δεδομένων χωρίστηκε σε **σύνολα εκπαίδευσης** και **δοκιμών**, με αναλογία 80% έως 20%. Αυτή είναι μια τυπική προσέγγιση στη μηχανική μάθηση για να διασφαλιστεί ότι τα μοντέλα εκπαιδεύονται στην πλειοψηφία των δεδομένων, αφήνοντας ένα μέρος στην άκρη για αμερόληπτη αξιολόγηση σε άγνωστα δεδομένα. Για την περαιτέρω βελτίωση της ανθεκτικότητας του μοντέλου και τη μείωση του κινδύνου υπερβολικής τοποθέτησης, **εφαρμόστηκε διασταυρούμενη επικύρωση K-fold** κατά τη διάρκεια της φάσης εκπαίδευσης. Συγκεκριμένα, χρησιμοποιήθηκε **5-πλάσια διασταυρούμενη επικύρωση**, η οποία είναι ενσωματωμένη στα μοντέλα, όπου το εκπαιδευτικό σύνολο χωρίστηκε σε πέντε υποσύνολα ίσου μεγέθους. Για κάθε πτυχή, το μοντέλο εκπαιδεύτηκε σε τέσσερα από τα υποσύνολα και επικυρώθηκε στο υπόλοιπο. Αυτή η διαδικασία επαναλήφθηκε πέντε φορές, διασφαλίζοντας ότι κάθε υποσύνολο χρησιμοποιήθηκε για επικύρωση μία φορά.

Η **στρατηγική διασταυρούμενης επικύρωσης** ήταν ζωτικής σημασίας για την αξιολόγηση της γενίκευσης του μοντέλου και τη μείωση της διακύμανσης στις εκτιμήσεις απόδοσης του μοντέλου. Χρησιμοποιώντας αυτήν την τεχνική, διασφαλίσαμε ότι η αξιολόγηση των επιδόσεων δεν βασίστηκε αποκλειστικά σε έναν διαχωρισμό εκπαίδευσης-δοκιμής, αλλά ήταν αντιπροσωπευτική σε πολλές πτυχές.

2.5.2 Εκπαίδευση Λογιστικής Παλινδρόμησης

Η λογιστική παλινδρόμηση είναι ένα ευρέως χρησιμοποιούμενο μοντέλο στην πρόβλεψη πιστωτικού κινδύνου λόγω της απλότητας και της ερμηνευσιμότητάς του. Σε αυτό το έργο, η λογιστική παλινδρόμηση υλοποιήθηκε χρησιμοποιώντας τον **ταξινομητή Stochastic Gradient Descent Classifier (SGDClassifier)** από τη βιβλιοθήκη **scikit-learn**. Αυτή η μέθοδος είναι γνωστή για την υπολογιστική αποτελεσματικότητά της, ειδικά όταν ασχολείται με μεγάλα σύνολα δεδομένων, κάτι που ήταν κρίσιμο δεδομένου του μεγέθους του συνόλου δεδομένων Home Credit. Εφαρμόσαμε την **τακτοποίηση L2 (L2 regularization),** για να αποτρέψουμε την υπερβολική τοποθέτηση τιμωρώντας μεγάλες εκτιμήσεις συντελεστών. Η κανονικοποίηση βοηθά στον έλεγχο της πολυπλοκότητας του μοντέλου, αποθαρρύνοντας το μοντέλο από την προσαρμογή θορύβου στα δεδομένα. Η **υπερπαράμετρος alpha**, η οποία ελέγχει την ισχύ της κανονικοποίησης, συντονίστηκε χρησιμοποιώντας **αναζήτηση πλέγματος** σε ένα εύρος τιμών. Η διαδικασία αναζήτησης πλέγματος αξιολογεί συστηματικά πολλαπλές τιμές άλφα και επιλέγει εκείνη που μεγιστοποιεί την απόδοση με βάση τα αποτελέσματα διασταυρούμενης επικύρωσης. Η καλύτερη τιμή για το άλφα επιλέχθηκε για να επιτευχθεί ισορροπία μεταξύ μεροληψίας και διακύμανσης.

Επιπλέον, χρησιμοποιήσαμε **το CalibratedClassifierCV** με τη **σιγμοειδή μέθοδο** για να βελτιώσουμε τις εκτιμήσεις πιθανότητας του μοντέλου. Η λογιστική παλινδρόμηση συχνά παράγει αξιόπιστα αποτελέσματα δυαδικής ταξινόμησης, αλλά ο βαθμονομημένος ταξινομητής ενισχύει την πιθανοτική ερμηνεία των αποτελεσμάτων, καθιστώντας τον πιο κατάλληλο για εργασίες εκτίμησης κινδύνου όπου οι ακριβείς προβλέψεις πιθανότητας είναι απαραίτητες.

2.5.3 Εκπαίδευση XGBoost

Το XGBoost (eXtreme Gradient Boosting) επιλέχθηκε για την αποδεδειγμένη αποτελεσματικότητά του στο χειρισμό δομημένων δεδομένων και την ικανότητά του να καταγράφει πολύπλοκες μη γραμμικές σχέσεις. Το XGBoost δημιουργεί ένα σύνολο δέντρων αποφάσεων, καθένα από τα οποία βελτιώνει τα λάθη του προηγούμενου, μειώνοντας έτσι την προκατάληψη και τη διακύμανση.

Οι βασικές υπερπαράμετροι **max_depth** και **min_child_weight** βελτιστοποιήθηκαν χρησιμοποιώντας **αναζήτηση πλέγματος**. Η παράμετρος max_depth ελέγχει το μέγιστο βάθος κάθε δέντρου. Οι υψηλότερες τιμές επιτρέπουν στο μοντέλο να καταγράφει πιο λεπτομερείς αλληλεπιδράσεις στα δεδομένα, αν και αυτό ενέχει τον κίνδυνο υπερβολικής προσαρμογής. **Min_child_weight** καθορίζει το ελάχιστο άθροισμα των βαρών στιγμιότυπων που απαιτούνται σε έναν κόμβο φύλλου, διασφαλίζοντας ότι οι διαχωρισμοί συμβαίνουν μόνο όταν υπάρχουν επαρκή δεδομένα για να το δικαιολογήσουν.

2.5.4 Εκπαίδευση δικτύου Kolmogorov-Arnold (KAN)

Το δίκτυο Kolmogorov-Arnold (KAN), ένας τύπος νευρωνικού δικτύου που υπερέχει στην προσέγγιση συνεχών συναρτήσεων, επιλέχθηκε για τις μη γραμμικές δυνατότητες προσέγγισης, οι οποίες είναι ιδιαίτερα επωφελείς για την καταγραφή πολύπλοκων σχέσεων σε δεδομένα πιστωτικού κινδύνου. Για να προσδιορίσουμε την καλύτερη αρχιτεκτονική για αυτό το μοντέλο, πειραματιστήκαμε με τρία διαφορετικά πλάτη (τον αριθμό των νευρώνων σε κάθε κρυφό στρώμα).

Ο KAN εκπαιδεύτηκε χρησιμοποιώντας το **Adam optimizer**, μια παραλλαγή στοχαστικής κλίσης που προσαρμόζει το ρυθμό μάθησης καθ 'όλη τη διάρκεια της προπόνησης, εξασφαλίζοντας ταχύτερη σύγκλιση και καλύτερη απόδοση παρουσία αραιών κλίσεων.

Η διασταυρούμενη επικύρωση ήταν το κλειδί για τον προσδιορισμό της βέλτιστης αρχιτεκτονικής KAN και της καλύτερης τιμής για το λάμδα. Εκτελώντας 5πλάσια διασταυρούμενη επικύρωση, διασφαλίσαμε ότι το μοντέλο θα μπορούσε να γενικευτεί καλά σε διαφορετικά υποσύνολα δεδομένων, μειώνοντας τον κίνδυνο υπερβολικής τοποθέτησης και υποπροσαρμογής. Το τελικό μοντέλο KAN επικυρώθηκε στο σετ δοκιμών, όπου η απόδοσή του συγκρίθηκε τόσο με το Logistic Regression όσο και με το XGBoost.

2.6 Εργαλεία και τεχνολογίες

Αυτή η ενότητα περιγράφει τα εργαλεία και τις τεχνολογίες που χρησιμοποιούνται κατά τη διάρκεια της ερευνητικής διαδικασίας, ειδικά για την προεπεξεργασία δεδομένων, την κατάρτιση μοντέλων και την αξιολόγηση της απόδοσης. Η επιλογή αυτών των εργαλείων έγινε με γνώμονα την αποτελεσματικότητά τους και την ευκολία ενσωμάτωσής τους στους στόχους του έργου.

2.6.1 Γλώσσα προγραμματισμού: Python

Η Python επιλέχθηκε ως η κύρια γλώσσα προγραμματισμού για αυτό το έργο λόγω του τεράστιου οικοσυστήματος βιβλιοθηκών για ανάλυση δεδομένων και μηχανική μάθηση. Η ευελιξία της Python επέτρεψε την απρόσκοπτη εκτέλεση της προεπεξεργασίας δεδομένων, της μηχανικής χαρακτηριστικών και της ανάπτυξης μοντέλων μηχανικής μάθησης.

- **Pandas**: Η βιβλιοθήκη Pandas χρησιμοποιήθηκε για χειρισμό δεδομένων, καθαρισμό και ανάλυση. Η ισχυρή δομή του DataFrame διευκόλυνε τον χειρισμό μεγάλων συνόλων δεδομένων, όπως το σύνολο δεδομένων Home Credit Default Risk, καθιστώντας πιο αποτελεσματικές λειτουργίες όπως το φιλτράρισμα, η συγχώνευση και η συγκέντρωση δεδομένων.

- **Numpy**: Το Numpy χρησιμοποιήθηκε για το χειρισμό πινάκων και πινάκων αριθμητικών δεδομένων, παρέχοντας εργαλεία υψηλής απόδοσης για επιστημονικούς υπολογισμούς.

2.6.2 Βιβλιοθήκες οπτικοποίησης δεδομένων: Seaborn

Η οπτικοποίηση δεδομένων ήταν ένα ουσιαστικό βήμα για την κατανόηση των μοτίβων και των τάσεων στο σύνολο δεδομένων. Για το σκοπό αυτό, το Seaborn, χτισμένο στην κορυφή του Matplotlib, ήταν η κύρια βιβλιοθήκη που χρησιμοποιήθηκε για τη σχεδίαση.

- **Seaborn**: Αυτή η βιβλιοθήκη επέτρεψε τη δημιουργία διαφόρων απεικονίσεων, όπως box-plots, correlation heatmaps. Αυτές οι απεικονίσεις ήταν ζωτικής σημασίας για τη διερεύνηση των σχέσεων μεταξύ των χαρακτηριστικών και τον εντοπισμό ακραίων τιμών.

2.6.3 Πλαίσια Μηχανικής Μάθησης: Scikit-learn, XGBoost και Efficient-KAN

Για την κατασκευή και την αξιολόγηση μοντέλων μηχανικής μάθησης, **οι Scikit-learn, XGBoost** και **Efficient-KAN** ήταν οι βιβλιοθήκες που χρησιμοποιήθηκαν:

- **Scikit-learn**: Μια ολοκληρωμένη βιβλιοθήκη για μηχανική μάθηση, η Scikit-learn παρείχε εργαλεία για διαχωρισμό δεδομένων, διασταυρούμενη επικύρωση και αξιολόγηση μοντέλων. Τα API του χρησιμοποιήθηκαν για την επιλογή μοντέλων, την εκπαίδευση και τη σύγκριση μετρήσεων απόδοσης, όπως ακρίβεια, ανάκληση και βαθμολογία ROC-AUC. Συγκεκριμένα, χρησιμοποιήθηκε για το χειρισμό συμβατικών αλγορίθμων μηχανικής μάθησης όπως η λογιστική παλινδρόμηση.

- **XGBoost:** Μια αποτελεσματική εφαρμογή δέντρων αποφάσεων με ενίσχυση κλίσης, το XGBoost χρησιμοποιήθηκε για την υψηλή απόδοση και την επεκτασιμότητα του στο χειρισμό μεγάλων συνόλων δεδομένων και την ισχυρή προγνωστική ακρίβεια.

- **Efficient-KAN** : Το Efficient-KAN χρησιμοποιήθηκε για την υλοποίηση των δικτύων Kolmogorov-Arnold (KAN), το οποίο είναι γνωστό για τη μοντελοποίηση πολύπλοκων μη γραμμικών συστημάτων με λιγότερες παραμέτρους από τα παραδοσιακά νευρωνικά δίκτυα. Αυτή η βιβλιοθήκη παρέχει μια βελτιστοποιημένη και πιο επεκτάσιμη εφαρμογή του KAN, βοηθώντας στον εξορθολογισμό της διαδικασίας εκπαίδευσης και πρόβλεψης.

2.6.4 Περιβάλλον εκπαίδευσης μοντέλων: Kaggle Cloud Platform

Η εκπαίδευση και η αξιολόγηση των μοντέλων πραγματοποιήθηκαν στην **πλατφόρμα Cloud της Kaggle**, ένα ευρέως χρησιμοποιούμενο διαδικτυακό περιβάλλον για την επιστήμη των δεδομένων και τη μηχανική μάθηση. Το Kaggle προσφέρει πρόσβαση σε GPU και TPU, τα οποία ήταν απαραίτητα για την επιτάχυνση της εκπαίδευσης μοντέλων μηχανικής μάθησης.

- **Kaggle Notebooks**: Αυτή η πλατφόρμα παρείχε ένα περιβάλλον Jupyter Notebook, επιτρέποντας τη διαδραστική κωδικοποίηση και ανάλυση δεδομένων. Υποστήριζε εύκολη πρόσβαση σε σύνολα δεδομένων, επιτρέποντας την εισαγωγή δεδομένων διαγωνισμού απευθείας στο περιβάλλον. Επιπλέον, οι πόροι cloud computing της Kaggle διευκόλυναν την εκπαίδευση μοντέλων μεγάλης κλίμακας χωρίς την ανάγκη τοπικών πόρων υλικού.

Η υποδομή του Kaggle ήταν απαραίτητη για την κλιμάκωση του έργου, επιτρέποντας επαναληπτικό συντονισμό μοντέλων και γρήγορο πειραματισμό με διαφορετικές υπερπαραμέτρους στα μοντέλα μηχανικής μάθησης, συμπεριλαμβανομένων των δικτύων Kolmogorov-Arnold τα οποία απαιτούν αρκετά περισσότερους υπολογιστικούς πόρους σε σχέση με τα υπόλοιπα μοντέλα.

3. Αποτελέσματα Έρευνας

3.1 Επισκόπηση των αποτελεσμάτων του πειράματος

Η διαδικασία πειραματισμού για αυτή τη διατριβή ακολούθησε μια δομημένη και επαναληπτική προσέγγιση για την αξιολόγηση της προγνωστικής ισχύος διαφόρων μοντέλων μηχανικής μάθησης για το σύνολο δεδομένων Home Credit Default Risk. Ο απώτερος στόχος ήταν να καθοριστεί ποιο μοντέλο θα μπορούσε να προβλέψει καλύτερα εάν ένας πελάτης θα αποπληρώσει επιτυχώς ένα δάνειο, με ιδιαίτερη έμφαση στην απόδοση της Kolmogorov-Arnold Networks (KAN). Η μεθοδολογία περιελάμβανε προεπεξεργασία δεδομένων, μηχανική χαρακτηριστικών, εκπαίδευση μοντέλων, επικύρωση και αξιολόγηση απόδοσης χρησιμοποιώντας διάφορες βασικές μετρήσεις.

3.1.1 Προεπεξεργασία δεδομένων και μηχανική χαρακτηριστικών

Το σύνολο δεδομένων που χρησιμοποιήθηκε σε αυτή τη μελέτη, που παρέχεται από την Home Credit, περιέχει πολλούς πίνακες, συμπεριλαμβανομένων δεδομένων αιτήσεων πελατών, πληροφοριών πιστωτικού γραφείου και δεδομένων υπολοίπου πιστωτικών καρτών. Πριν από τη μοντελοποίηση, εφαρμόστηκαν σημαντικά βήματα προεπεξεργασίας για το χειρισμό τιμών που λείπουν, ακραίων τιμών και κατηγορικών μεταβλητών.

- **Χειρισμός δεδομένων που λείπουν**: Το σύνολο δεδομένων είχε σημαντικές τιμές που έλειπαν. Οι γραμμές με μικρές ποσότητες δεδομένων που λείπουν καταργήθηκαν. Για στήλες με τιμές που λείπουν πάνω από 90%, ολόκληρη η στήλη απορρίφθηκε. Για τις κατηγορικές μεταβλητές, τα δεδομένα που έλειπαν αντικαταστάθηκαν με μια νέα τιμή, "XNA", υποδεικνύοντας ότι οι πληροφορίες απουσίαζαν. Για τα αριθμητικά δεδομένα, οι τιμές που έλειπαν αντικαταστάθηκαν με 0 για να αποφευχθεί η μεροληψία που εισήχθη από τεκμαρτές τιμές.

- **Ανίχνευση και θεραπεία ακραίων τιμών**: Τα boxplots χρησιμοποιήθηκαν για την ανίχνευση ακραίων τιμών στις αριθμητικές μεταβλητές. Οι ακραίες τιμές, ειδικά σε μεταβλητές όπως το εισόδημα και το πιστωτικό ποσό, αφαιρέθηκαν εντελώς για να αποφευχθεί η στρέβλωση των αποτελεσμάτων του μοντέλου. Αυτή η προσέγγιση επιλέχθηκε έναντι του καταλογισμού για τη διατήρηση της ακεραιότητας των δεδομένων.

- **Μηχανική χαρακτηριστικών**: Δημιουργήθηκαν αρκετά νέα χαρακτηριστικά για τη βελτίωση της προβλεπτικής ισχύος των μοντέλων. Για παράδειγμα, χαρακτηριστικά όπως *ANNUITY_INCOME_PERCENT*, *CREDIT_TERM* (ο λόγος της ετήσιας προσόδου προς το ποσό πίστωσης) και *CREDIT_INCOME_PERCENT* δημιουργήθηκαν στον πίνακα *application_train* για να καταγράψουν τη βασική δυναμική που σχετίζεται με το δάνειο. Παρόμοιες τεχνικές μηχανικής χαρακτηριστικών εφαρμόστηκαν σε άλλους πίνακες όπως *το bureau*, όπου *υπολογίστηκαν* CREDIT_DURATION *και* DEBT_PERCENTAGE.

3.1.2 Επιλογή και εκπαίδευση μοντέλων

Ο πρωταρχικός στόχος της ανάλυσης ήταν η απόδοση της Kolmogorov-Arnold Networks (KAN). Ωστόσο, για τη συγκριτική αξιολόγηση της απόδοσης του KAN, δοκιμάστηκαν και άλλα μοντέλα μηχανικής μάθησης. Αυτά τα μοντέλα περιελάμβαναν:

- **Λογιστική παλινδρόμηση**: Ένα παραδοσιακό, ερμηνεύσιμο μοντέλο που χρησιμοποιείται συχνά στην πρόβλεψη πιστωτικού κινδύνου.

- **XGBoost**: Ένα ισχυρό πλαίσιο ενίσχυσης κλίσης που έχει γίνει δημοφιλές στους διαγωνισμούς Kaggle λόγω της υψηλής απόδοσής του σε δεδομένα πίνακα.

Κάθε μοντέλο εκπαιδεύτηκε χρησιμοποιώντας 5πλάσια διασταυρούμενη επικύρωση για να διασφαλιστεί η στιβαρότητα στην απόδοση και να αποφευχθεί η υπερβολική τοποθέτηση. Ο συντονισμός υπερπαραμέτρων πραγματοποιήθηκε χρησιμοποιώντας αναζήτηση πλέγματος για τη βελτιστοποίηση της απόδοσής των μοντέλων.

3.1.3 Μετρήσεις αξιολόγησης μοντέλων

Η απόδοση του μοντέλου αξιολογήθηκε χρησιμοποιώντας διάφορες βασικές μετρήσεις, καθεμία από τις οποίες επιλέχθηκε για να αντιμετωπίσει διαφορετικές πτυχές της απόδοσης ταξινόμησης:

- **Accuracy**: Το ποσοστό των σωστών προβλέψεων σε ολόκληρο το σύνολο δεδομένων. Ενώ η μέτρηση accuracy είναι μια απλή μέτρηση, μπορεί να είναι παραπλανητική όταν πρόκειται για μη ισορροπημένα σύνολα δεδομένων όπως τα δεδομένα εγχώριας πίστωσης, όπου οι περιπτώσεις αθέτησης είναι πολύ λιγότερες από τις περιπτώσεις μη αθέτησης υποχρεώσεων.

- **Precision, Recall και βαθμολογία F1**: Αυτές οι μετρήσεις χρησιμοποιήθηκαν για την αξιολόγηση της απόδοσης των μοντέλων στην πρόβλεψη της προεπιλεγμένης κλάσης (που είναι η κλάση μειοψηφίας). Η μέτρηση Precision αξιολογεί πόσες από τις προβλεπόμενες αθετήσεις ήταν σωστές, ενώ η Recall μετρά πόσες πραγματικές αθετήσεις εντοπίστηκαν. Το F1-Score, ο αρμονικός μέσος όρος ακρίβειας και ανάκλησης, παρείχε μια ισορροπία μεταξύ αυτών των δύο μετρήσεων.

- **ROC-AUC (Receiver Operating Characteristic - Area Under the Curve)**: Αυτή η μέτρηση ήταν κεντρική στην αξιολόγηση, καθώς αντικατοπτρίζει την ικανότητα του μοντέλου να διακρίνει μεταξύ κλάσεων (προεπιλογή έναντι μη προεπιλογής). Μια υψηλότερη βαθμολογία ROC-AUC δείχνει καλύτερη απόδοση.

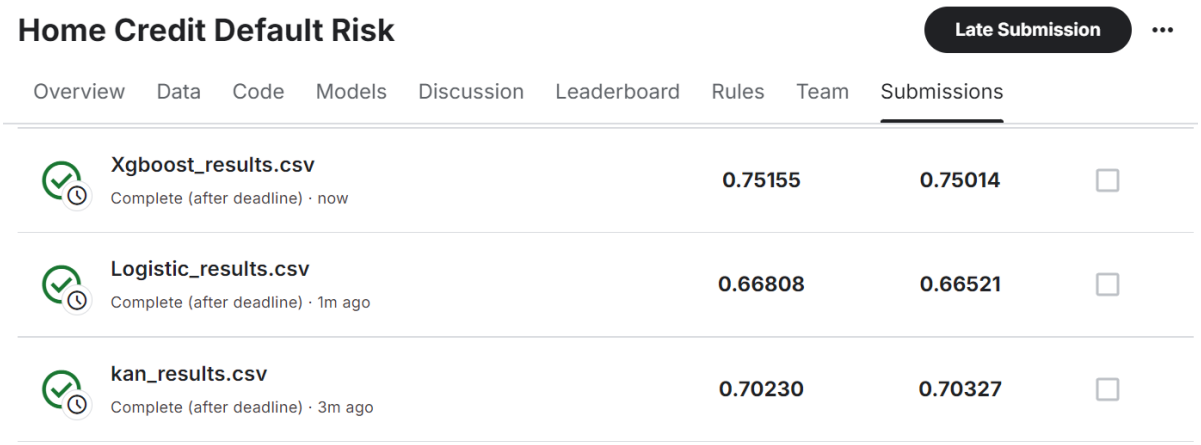
Επιπλέον χρησιμοποιήθηκε η αξιολόγηση μέσω του διαγωνισμού Home Credit Default Risk στο Kaggle, ο οποίος αναθέτει ένα σκορ σε κάθε υποβολή, η οποία περιέχει τις προβλέψεις του μοντέλου με βάση τον πίνακα test_application.

3.1.4 Σύνοψη των αποτελεσμάτων

- **Λογιστική παλινδρόμηση**: ROC-AUC 0,74, Kaggle competion = 0.67, Precision = 0.16, Recall = 0.67, F1 Score = 0.26, Accuracy = 0.70.

- **XGBoost**: ROC-AUC 0,77, Kaggle competion = 0.76, Precision = 0.17, Recall = 0.71, F1 Score = 0.27, accuracy = 0.70.

- **Δίκτυα Kolmogorov-Arnold (KAN)**: ROC-AUC 0,71, Kaggle competion = 0.7, Precision = 0.15, Recall = 0.65, F1 Score = 0.25, accuracy = 0.68.



ΕΙΚΟΝΑ 3.1: Αποτελέσματα του διαγωνισμού στο Kaggle

3.2 Αξιολόγηση μοντέλων

Αυτή η ενότητα αξιολογεί την απόδοση των τριών μοντέλων - Logistic Regression, XGBoost και Kolmogorov-Arnold Networks (KAN). Κάθε μοντέλο αξιολογήθηκε χρησιμοποιώντας ένα σύνολο βασικών μετρήσεων: ROC-AUC, μέτρηση του διαγωνισμού Kaggle, Precision, Recall, F1 Score, and Accuracy. Αυτές οι μετρήσεις παρέχουν μια ολοκληρωμένη εικόνα της αποτελεσματικότητας κάθε μοντέλου στον χειρισμό του προβλήματος πρόβλεψης πιστωτικής αθέτησης υποχρεώσεων.Για λόγους σύγκρισης, το νικητήριο μοντέλο Kaggle σε αυτόν τον διαγωνισμό είχε βαθμολογία 0,81, η οποία χρησιμεύει ως σημείο αναφοράς.

3.2.1 Λογιστική παλινδρόμηση

Η λογιστική παλινδρόμηση είναι μια βασική αλλά ισχυρή μέθοδος για την κατανόηση των σχέσεων μεταξύ των χαρακτηριστικών και της μεταβλητής-στόχου. Για αυτό το σύνολο δεδομένων, παρείχε τα ακόλουθα αποτελέσματα:

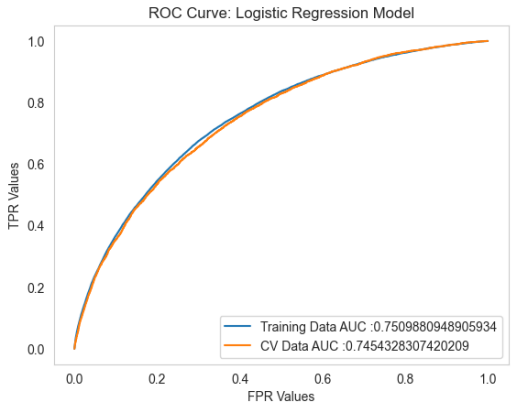
- **ROC-AUC = 0,74**: Αυτή η βαθμολογία δείχνει ότι το μοντέλο είναι αρκετά ικανό να διακρίνει μεταξύ αθέτησης και μη αθέτησης. Αν και δεν είναι το υψηλότερο, δείχνει ένα μέτριο επίπεδο εξουσίας διακρίσεων.

- **Kaggle Competition Metric = 0,67**: Αυτή η μέτρηση, η οποία λαμβάνει υπόψη τους συγκεκριμένους επιχειρηματικούς στόχους του διαγωνισμού, υποδηλώνει ότι το γραμμικό μοντέλο ήταν μόνο μέτρια επιτυχημένο. Δεδομένου ότι η βαθμολογία νίκης ήταν 0,81, αυτό το αποτέλεσμα αντικατοπτρίζει περιορισμούς στην ικανότητα του μοντέλου να συλλάβει πολύπλοκες σχέσεις.

- **Precision = 0,16, Recall = 0,67, βαθμολογία F1 = 0,26**: Αυτές οι μετρήσεις δείχνουν ότι ενώ το μοντέλο είχε υψηλή μέτρηση Recall (ικανότητα αναγνώρισης αληθινών θετικών), η μέτρηση Precision του (σωστή αναγνώριση θετικών) ήταν χαμηλή. Η βαθμολογία F1, μια ισορροπία μεταξύ ακρίβειας και ανάκλησης, είναι σχετικά χαμηλή στο 0,26, υποδεικνύοντας ότι το μοντέλο δυσκολεύτηκε να κάνει ακριβείς προβλέψεις με συνέπεια.

- **Ακρίβεια = 0,70**: Η συνολική ακρίβεια δείχνει ότι το μοντέλο προέβλεψε σωστά το 70% των περιπτώσεων, κάτι που είναι λογικό αλλά όχι βέλτιστο για την πρόβλεψη πιστωτικού κινδύνου.

Τα αποτελέσματα δείχνουν ότι η λογιστική παλινδρόμηση μπορεί να χρησιμεύσει ως βάση αναφοράς, αλλά δεν διαθέτει την πολυπλοκότητα για να συλλάβει τις ιδιαιτερότητες στα δεδομένα πιστωτικού κινδύνου, όπως οι μη γραμμικές αλληλεπιδράσεις και η σημασία των χαρακτηριστικών.



Διάγραμμα 3.1: καμπύλη ROC λογιστικής παλινδρόμησης

```
Logistic regression results
Precision = 0.16554455445544555
Recall = 0.6735146022155085
F1 Score = 0.26576594476455395
Accuracy = 0.6995707456668075
```

ΕΙΚΟΝΑ 3.2: Αποτελέσματα Λογιστικής παλινδρόμησης

3.2.2 XGBoost

Ο XGBoost, ένας αλγόριθμος ενίσχυσης κλίσης, είναι γνωστός για την απόδοσή του σε δομημένα προβλήματα δεδομένων και διαγωνισμούς. Ξεπέρασε τα άλλα μοντέλα σε αυτή την αξιολόγηση:

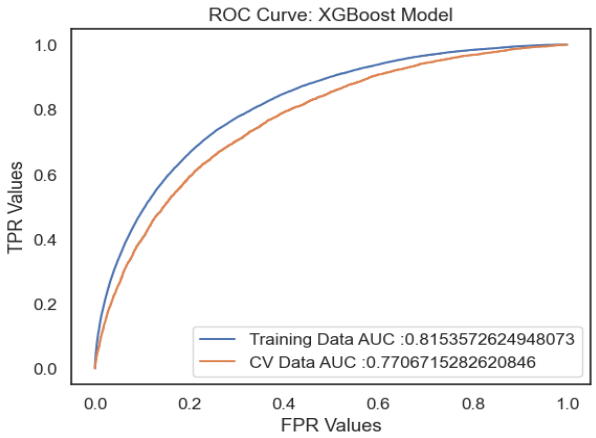
- **ROC-AUC = 0,87**: Αυτή η υψηλή βαθμολογία αποδεικνύει την ικανότητα του XGBoost να διακρίνει μεταξύ αθετήσεων και μη αθετήσεων, αντανακλώντας τη δύναμή του στο χειρισμό πολύπλοκων αλληλεπιδράσεων χαρακτηριστικών και μη γραμμικών σχέσεων.

- **Kaggle Competition Metric = 0,76**: Αυτό το αποτέλεσμα, αν και χαμηλότερο από το νικητήριο σκορ του 0,81, δείχνει ότι το XGBoost ήταν εξαιρετικά αποτελεσματικό στον διαγωνισμό. Υποδηλώνει ότι το XGBoost ήταν καλύτερα ευθυγραμμισμένο με τους συγκεκριμένους στόχους του διαγωνισμού σε σύγκριση με το Linear Regression και το KAN.

- **Precision = 0,17, Recall = 0,71, βαθμολογία F1 = 0,27**: Αυτές οι τιμές δείχνουν ισορροπημένη απόδοση. Η μέτρηση Recall είναι υψηλή, υποδεικνύοντας καλή ευαισθησία στους παραβάτες, ενώ η Precision και η βαθμολογία F1 είναι καλύτερες από αυτές της Γραμμικής Παλινδρόμησης, αν και εξακολουθούν να είναι μέτριες.

- **Ακρίβεια = 0,70**: Η ακρίβεια είναι η ίδια με τη γραμμική παλινδρόμηση, αλλά με σημαντικά καλύτερη απόδοση σε άλλες μετρήσεις, γεγονός που υποδηλώνει ότι το XGBoost εντόπισε σωστά πιο λεπτά μοτίβα που χάθηκαν από το γραμμικό μοντέλο.

Συνολικά, η ανώτερη απόδοση της XGBoost σε πολλαπλές μετρήσεις επιβεβαιώνει την αποτελεσματικότητά της σε αυτό το έργο πρόβλεψης πιστωτικού κινδύνου. Η ικανότητά του να χειρίζεται πολύπλοκα, υψηλής διάστασης δεδομένα το καθιστά κατάλληλο για τέτοια προβλήματα.



Διάγραμμα 3.2: καμπύλη ROC XGBoost

```
XGBoost results
Precision = 0.17060745866974242
Recall = 0.7150050352467271
F1 Score = 0.27548209366391185
accuracy = 0.6963838574355306
```

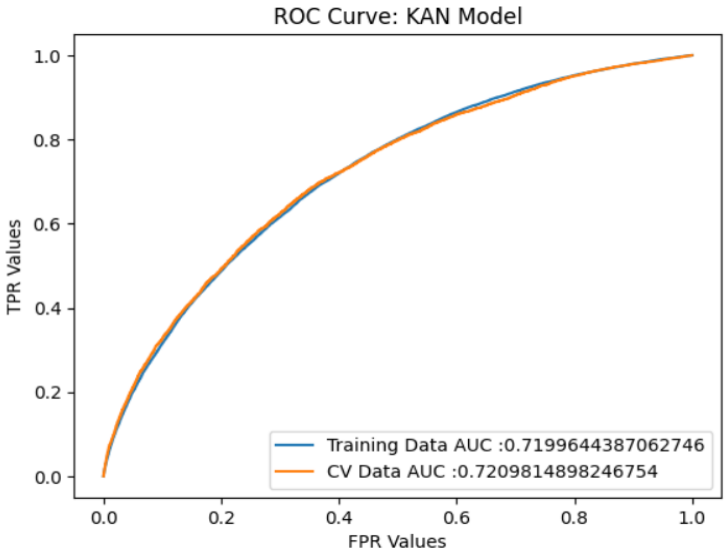
ΕΙΚΟΝΑ 3.3: Αποτελέσματα XGBoost

3.2.3 Δίκτυα Kolmogorov-Arnold (KAN)

Το KAN, ένα θεωρητικά ισχυρό μοντέλο για την αποτύπωση σύνθετων, μη γραμμικών μοτίβων, είχε χαμηλές επιδόσεις σε αυτή την αξιολόγηση:

- **ROC-AUC = 0,71**: Η μέτρηση είναι χαμηλότερη τόσο από το XGBoost όσο και από τη λογιστική παλινδρόμηση. Αν και το KAN είναι θεωρητικά ικανό να μοντελοποιεί πολύπλοκες σχέσεις, η απόδοσή του εδώ ήταν υποβέλτιστη, πιθανώς λόγω προκλήσεων στον συντονισμό του μοντέλου ή στη μηχανική χαρακτηριστικών.
- **Kaggle Competition Metric = 0,60**: Το KAN σημείωσε 0,70 στη μέτρηση ανταγωνισμού Kaggle, ξεπερνώντας ελαφρώς τη γραμμική παλινδρόμηση, αλλά εξακολουθεί να βρίσκεται πίσω από το XGBoost. Αυτό δείχνει ότι, ενώ το KAN ήταν σε θέση να καταγράψει ορισμένες σημαντικές πτυχές του συνόλου δεδομένων, δεν είχε την απαραίτητη ακρίβεια για να ανταγωνιστεί το XGBoost.
- **Ακρίβεια = 0,15, Ανάκληση = 0,65, Βαθμολογία F1 = 0,25**: Αυτές οι μετρήσεις υπογραμμίζουν σημαντικές προκλήσεις. Ενώ η ανάκληση είναι λογική, η ακρίβεια και η βαθμολογία F1 είναι αρκετά χαμηλές, υποδεικνύοντας ότι ο KAN συχνά ταξινομεί εσφαλμένα τους κακοπληρωτές ως μη κακοπληρωτές και αντίστροφα.
- **Ακρίβεια = 0,63**: Η χαμηλότερη ακρίβεια μεταξύ των μοντέλων που αξιολογήθηκαν, Αν και η ακρίβεια δεν είναι το καλύτερο μέτρο για μη ισορροπημένα σύνολα δεδομένων, παρέχει περαιτέρω στοιχεία ότι το KAN δεν απέδωσε όσο αναμενόταν σε αυτό το σενάριο..

Παρά τα θεωρητικά πλεονεκτήματα του KAN, η εφαρμογή του σε αυτό το πρόβλημα δεν ήταν επιτυχής, πιθανώς λόγω δυσκολιών στη διαμόρφωση του μοντέλου και τη ρύθμιση για αυτόν τον συγκεκριμένο τύπο δεδομένων. Τα αποτελέσματα υποδηλώνουν ότι απαιτούνται περαιτέρω εργασίες για τη βελτιστοποίηση του KAN για δομημένα δεδομένα πιστωτικού κινδύνου.



Διάγραμμα 3.3: καμπύλη ROC KAN

```
KAN results
Precision = 0.15391848056038737
Recall = 0.6520613423620792
F1 Score = 0.24904914042294235
accuracy = 0.6789697896003382
```

ΕΙΚΟΝΑ 3.4: Αποτελέσματα KAN

3.2.4 Περίληψη και σύγκριση

Το XGBoost αναδείχθηκε ως η καλύτερη απόδοση, με ROC-AUC 0,87 και μέτρηση διαγωνισμού Kaggle 0,76, λίγο πίσω από τη βαθμολογία του νικητή του διαγωνισμού 0,81. Η γραμμική παλινδρόμηση, αν και απλούστερη και ευκολότερη στην ερμηνεία, πέτυχε ROC-AUC 0,74 και μέτρηση ανταγωνισμού Kaggle 0,67, υποδεικνύοντας τους περιορισμούς της στο χειρισμό σύνθετων, μη γραμμικών σχέσεων. Το KAN, από την άλλη πλευρά, έδειξε τη χαμηλότερη απόδοση σε όλες τις μετρήσεις, υποδηλώνοντας ότι δεν ήταν κατάλληλο για αυτό το έργο στην τρέχουσα υλοποίησή του.

Τα αποτελέσματα δείχνουν ότι ενώ τα παραδοσιακά μοντέλα όπως η γραμμική παλινδρόμηση παρέχουν μια χρήσιμη βάση, προηγμένα μοντέλα όπως το XGBoost προσφέρουν σημαντικές βελτιώσεις στην προγνωστική απόδοση. Παρά τις πολλά υποσχόμενες θεωρητικές του βάσεις, το KAN απαιτεί περαιτέρω βελτίωση για να είναι ανταγωνιστικό σε δομημένα καθήκοντα πρόβλεψης όπως ο πιστωτικός κίνδυνος. Οι μελλοντικές εργασίες θα μπορούσαν να επικεντρωθούν στη βελτίωση των διαδικασιών μηχανικής χαρακτηριστικών και συντονισμού για το KAN ή στη διερεύνηση υβριδικών μοντέλων που συνδυάζουν τα πλεονεκτήματα διαφορετικών προσεγγίσεων για την επίτευξη καλύτερων αποτελεσμάτων σε σύνθετα σύνολα οικονομικών δεδομένων.

3.2.5 Μηχανική Χαρακτηριστικών και Απόδοση KAN

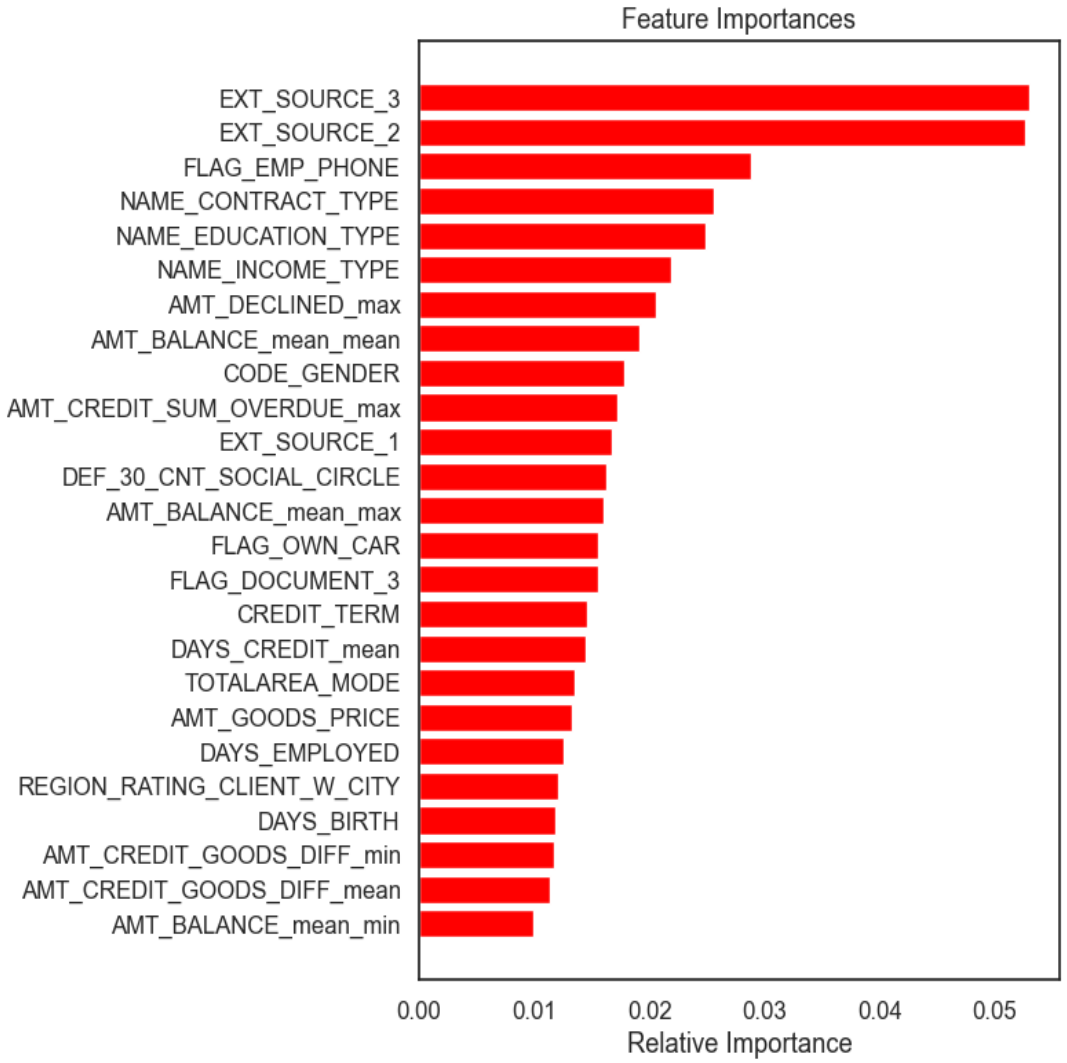
Μία από τις βασικές πτυχές που επηρέασαν την απόδοση του KAN ήταν η στρατηγική μηχανικής χαρακτηριστικών που χρησιμοποιήθηκε. Το σύνολο δεδομένων περιείχε πολλά σύνθετα χαρακτηριστικά που δημιουργήθηκαν μέσω ειδικών γνώσεων, όπως αναλογίες μεταξύ εισοδήματος και πίστωσης, προσόδων και ιστορικού απασχόλησης. Ενώ το KAN μπορεί θεωρητικά να μοντελοποιήσει πολύπλοκες σχέσεις μεταξύ αυτών των χαρακτηριστικών, η επιτυχία του εξαρτάται σε μεγάλο βαθμό από το πόσο καλά ευθυγραμμίζονται τα χαρακτηριστικά με τους συγκεκριμένους τύπους μη γραμμικοτήτων που είναι κατάλληλο να συλλάβει το δίκτυο.

Σε αυτό το πείραμα, η μηχανική χαρακτηριστικών επικεντρώθηκε στη δημιουργία χαρακτηριστικών αλληλεπίδρασης (π.χ. **ANNUITY_INCOME_PERCENT**, **CREDIT_TERM**, **CREDIT_INCOME_PERCENT** κ.λπ.), τα οποία αποδείχθηκαν ευεργετικά για το XGBoost. Ωστόσο, το KAN ενδέχεται να απαιτεί

διαφορετικούς τύπους αλληλεπιδράσεων ή μετασχηματισμών για την καλύτερη εκμετάλλευση του συνόλου δεδομένων. Αυτό υπογραμμίζει έναν κρίσιμο τομέα όπου το KAN θα μπορούσε ενδεχομένως να βελτιωθεί: την ενίσχυση της συμβατότητάς του με μεθόδους μηχανικής χαρακτηριστικών προσαρμοσμένες στον πιστωτικό κίνδυνο.

3.3 Ανάλυση Σημασίας Χαρακτηριστικών

Αυτή η ενότητα παρέχει μια συνοπτική επισκόπηση των πιο σημαντικών χαρακτηριστικών του **μοντέλου XGBoost** που χρησιμοποιείται για την πρόβλεψη πιστωτικού κινδύνου. **EXT_SOURCE_3** αναδειχθεί ως το πιο κρίσιμο χαρακτηριστικό, αντανακλώντας τις βαθμολογίες εξωτερικού πιστωτικού κινδύνου από τρίτους οργανισμούς, οι οποίες είναι απαραίτητες για την αξιολόγηση της πιθανότητας αθέτησης ενός δανειολήπτη. Χαρακτηριστικά όπως **AMT_GOODS_PRICE** και **CREDIT_TERM** υπογραμμίζουν τη σημασία των χρηματοοικονομικών μεταβλητών που σχετίζονται άμεσα με το μέγεθος του δανείου και τους όρους αποπληρωμής. Άλλοι σημαντικοί παράγοντες περιλαμβάνουν **το TOTALAREA_MODE**, το οποίο συνδέει το μέγεθος της ιδιοκτησίας με τη χρηματοπιστωτική σταθερότητα, και **AMT_BALANCE_mean_max**, αντανακλώντας τα υπόλοιπα πιστωτικών καρτών που σηματοδοτούν οικονομική πίεση. Χαρακτηριστικά χαμηλότερης κατάταξης, όπως **CODE_GENDER**, **FLAG_OWN_CAR** και **FLAG_EMP_PHONE** παρέχουν δημογραφικούς δείκτες και δείκτες αξιοπιστίας, οι οποίοι συμβάλλουν αλλά είναι λιγότερο προβλεπτικοί.



Διάγραμμα 3.4: Τα πιο Σημαντικά Χαρακτηριστικά

Βασικά χαρακτηριστικά:

1. **EXT_SOURCE_3** - Το πιο σημαντικό εξωτερικό πιστωτικό αποτέλεσμα.
2. **AMT_GOODS_PRICE** – Άμεσα συνδεδεμένο με το μέγεθος και τον κίνδυνο του δανείου.

3. **CREDIT_TERM** – Συσχετίζει τους όρους αποπληρωμής του δανείου με τον πιστωτικό κίνδυνο.

4. **AMT_BALANCE_mean_max** - Τα υπόλοιπα πιστωτικών καρτών ως δείκτης οικονομικής πίεσης.

Η ανάλυση της σημασίας των χαρακτηριστικών στο μοντέλο XGBoost υπογραμμίζει τον κρίσιμο ρόλο των εξωτερικών πιστωτικών βαθμών και των χρηματοοικονομικών μεταβλητών που σχετίζονται με τα ποσά και τους όρους των δανείων. Αυτοί οι παράγοντες κυριαρχούν στις προβλέψεις, προσφέροντας σαφείς πληροφορίες για τους σημαντικότερους παράγοντες του πιστωτικού κινδύνου. Η κατανόηση της βαρύτητας που αποδίδεται σε αυτά τα χαρακτηριστικά παρέχει διαφάνεια στη διαδικασία λήψης αποφάσεων του μοντέλου, βοηθώντας τα ιδρύματα να επικεντρωθούν στους πιο σχετικούς παράγοντες κατά την αξιολόγηση των αιτήσεων δανείων. Μέσω της ανάλυσης, είναι επίσης σαφές ότι πολλά από τα μηχανικά χαρακτηριστικά κατατάσσονται σε υψηλή σημασία, δείχνοντας τον κρίσιμο ρόλο που διαδραματίζει η μηχανική χαρακτηριστικών στη δημιουργία μοντέλων πρόβλεψης.

3.4 Περιορισμοί της μελέτης

Παρά τις επιτυχίες που επιτεύχθηκαν στην πρόβλεψη πιστωτικού κινδύνου χρησιμοποιώντας διάφορα μοντέλα μηχανικής μάθησης, αυτή η μελέτη έχει αρκετούς περιορισμούς που πρέπει να αντιμετωπιστούν για μελλοντικές εργασίες. Αυτοί οι περιορισμοί σχετίζονται κυρίως με την ποιότητα των δεδομένων, την επιλογή μοντέλων και τη μηχανική χαρακτηριστικών.

1. **Ποιότητα δεδομένων και τιμές που λείπουν**

   Ένας από τους κύριους περιορισμούς είναι ο χειρισμός των τιμών που λείπουν. Ενώ τα ελλείποντα δεδομένα αντιμετωπίστηκαν μέσω στρατηγικών καταλογισμού και αφαίρεσης, όπως η αντικατάσταση κατηγορικών τιμών που λείπουν με "XNA" και αριθμητικών τιμών που λείπουν με 0, αυτές οι προσεγγίσεις ενδέχεται να εισάγουν μεροληψία στο μοντέλο. Ο καταλογισμός τιμών που λείπουν μπορεί να αποδυναμώσει το πραγματικό σήμα μέσα στα δεδομένα, ιδιαίτερα όταν λείπει ένα σημαντικό τμήμα ενός χαρακτηριστικού. Πιο προηγμένες τεχνικές, όπως πολλαπλοί καταλογισμοί ή μέθοδοι βαθιάς μάθησης όπως οι αυτοκωδικοποιητές, θα μπορούσαν να βελτιώσουν τον χειρισμό των δεδομένων που λείπουν.
2. **Ανίχνευση και αντιμετώπιση ακραίων τιμών**

   Οι ακραίες τιμές αντιμετωπίστηκαν αφαιρώντας τις εντοπισμένες ακραίες τιμές με βάση την ανάλυση boxplot, η οποία μπορεί να είναι μια απλή αλλά αμβλεία προσέγγιση. Αν και αυτή η μέθοδος απέτρεψε πιθανή παραμόρφωση δεδομένων από ακραίες τιμές, η πιο προηγμένη επεξεργασία ακραίων τιμών, όπως η **ισχυρή κλιμάκωση** ή η χρήση **κλιμάκωσης βάσει ποσοτήτων (quantile-based scaling)**, θα μπορούσε να βελτιώσει την ικανότητα του μοντέλου να γενικεύει σε μελλοντικά σύνολα δεδομένων, όπου οι ακραίες τιμές θα μπορούσαν να παρέχουν πολύτιμες πληροφορίες για σπάνιες περιπτώσεις πιστωτικού κινδύνου.
3. **Πολυπλοκότητα και ερμηνευσιμότητα μοντέλου**

   Το δίκτυο Kolmogorov-Arnold (KAN), αν και καινοτόμο, παρουσίασε χαμηλότερες επιδόσεις σε σύγκριση με ευρύτερα χρησιμοποιούμενα μοντέλα όπως το XGBoost. Αυτό το χάσμα απόδοσης μπορεί να οφείλεται σε ανεπαρκή συντονισμό ή στην εγγενή πολυπλοκότητα του KAN, η οποία μπορεί να μην γενικευτεί καλά σε αυτόν τον τύπο δομημένων οικονομικών δεδομένων. Επιπλέον, τα μοντέλα KAN δεν διαθέτουν την ερμηνευσιμότητα που απαιτείται συχνά στις χρηματοπιστωτικές βιομηχανίες, όπου οι ρυθμιστικοί φορείς απαιτούν σαφείς εξηγήσεις για τις αποφάσεις μοντέλου.
4. **Μηχανική χαρακτηριστικών**

   Η μηχανική χαρακτηριστικών επικεντρώθηκε σε μεγάλο βαθμό στη δημιουργία δεικτών και ποσοστών με βάση τα διαθέσιμα οικονομικά δεδομένα. Ενώ αυτό ενίσχυσε την προγνωστική ισχύ του μοντέλου, η εξάρτηση από χειροκίνητα δημιουργημένα χαρακτηριστικά μπορεί να παραβλέψει πολύπλοκα μοτίβα που θα μπορούσαν να ανακαλυφθούν αυτόματα χρησιμοποιώντας τεχνικές βαθιάς μάθησης, όπως **νευρωνικά δίκτυα** σχεδιασμένα να μαθαίνουν ιεραρχίες χαρακτηριστικών απευθείας από ανεπεξέργαστα δεδομένα. Επιπλέον, χαρακτηριστικά όπως **CREDIT_TERM** και **CREDIT_ANNUITY_PERCENT** ενδέχεται να μην αποτυπώνουν την πλήρη πολυπλοκότητα της οικονομικής συμπεριφοράς ενός πελάτη.
5. **Γενίκευση σε διαφορετικούς πληθυσμούς**

Το σύνολο δεδομένων που χρησιμοποιείται σε αυτήν την ανάλυση είναι ιδιαίτερα συγκεκριμένο για τις αιτήσεις δανείων ενός χρηματοπιστωτικού ιδρύματος και τα μοντέλα ενδέχεται να μην γενικεύονται καλά σε άλλες αγορές ή πληθυσμούς με διαφορετικές οικονομικές συνθήκες. Απαιτείται περισσότερη δουλειά για την επικύρωση αυτών των ευρημάτων σε διαφορετικές γεωγραφικές περιοχές και δημογραφικά στοιχεία.

4. Συμπεράσματα

4.1 Σύνοψη των στόχων

Πρωταρχικός στόχος της παρούσας διπλωματικής εργασίας ήταν η διερεύνηση της εφαρμογής τεχνικών μηχανικής μάθησης, συγκεκριμένα της Kolmogorov-Arnold Networks (KAN), στην πρόβλεψη του πιστωτικού κινδύνου χρησιμοποιώντας το **σύνολο δεδομένων Home Credit Default Risk**. Η μελέτη είχε ως στόχο να συγκρίνει την απόδοση του KAN με παραδοσιακά μοντέλα όπως η γραμμική παλινδρόμηση και πιο προηγμένοι αλγόριθμοι όπως το XGBoost, και να αξιολογήσει την προγνωστική ισχύ των μοντέλων μέσω μετρήσεων όπως **η ROC-AUC** και η **βαθμολογία του διαγωνισμού Kaggle**.

Καθ' όλη τη διάρκεια της ανάλυσης, εφαρμόσαμε με επιτυχία μοντέλα μηχανικής μάθησης που αξιολόγησαν την πιθανότητα αποπληρωμής του δανείου των πελατών με βάση ένα ευρύ φάσμα οικονομικών και δημογραφικών παραγόντων. Η εφαρμογή της μηχανικής χαρακτηριστικών συνέβαλε σημαντικά στη βελτίωση της απόδοσης του μοντέλου, ιδιαίτερα για το μοντέλο XGBoost, το οποίο έδειξε την υψηλότερη ακρίβεια πρόβλεψης.

4.2 Μεθοδολογικά και πρακτικά προβλήματα

Αρκετές μεθοδολογικές προκλήσεις αντιμετωπίστηκαν κατά τη διάρκεια αυτής της έρευνας. Ο χειρισμός των **ελλειπόντων δεδομένων** και των **ακραίων τιμών** δημιουργούσε πρακτικές δυσκολίες, όπως αναλύεται στο κεφάλαιο 2. Παρόλο που χρησιμοποιήσαμε στρατηγικές όπως ο καταλογισμός τιμών που λείπουν με "XNA" για κατηγορικά δεδομένα και μηδέν για αριθμητικά δεδομένα, αυτή η προσέγγιση πιθανότατα εισήγαγε κάποια μεροληψία. Πιο εξελιγμένες μέθοδοι, όπως **ο πολλαπλός καταλογισμός** ή οι τεχνικές βαθιάς μάθησης, θα μπορούσαν να έχουν παράγει πιο ακριβή αποτελέσματα. Ο εντοπισμός ακραίων τιμών ήταν μια άλλη πρόκληση, καθώς η απλή αφαίρεση μέσω boxplots μπορεί να οδήγησε στην απώλεια πολύτιμων πληροφοριών σχετικά με ακραίες περιπτώσεις κινδύνου αθέτησης υποχρεώσεων. Προηγμένες τεχνικές όπως **η ισχυρή κλιμάκωση** θα μπορούσαν να μετριάσουν αυτό το ζήτημα σε μελλοντικές μελέτες.

Επιπλέον, το μοντέλο KAN είχε χαμηλή απόδοση σε σύγκριση με το XGBoost, πιθανώς λόγω της εγγενούς πολυπλοκότητας του συνόλου οικονομικών δεδομένων. Αυτό υποδηλώνει ότι τα μοντέλα KAN μπορεί να μην είναι κατάλληλα για δομημένα δεδομένα σε μορφή πίνακα χωρίς περαιτέρω συντονισμό ή υβριδισμό με άλλες μεθόδους μηχανικής μάθησης.

4.3 Επίτευξη στόχων

Παρά τις προκλήσεις αυτές, η έρευνα κατάφερε σε μεγάλο βαθμό να εκπληρώσει τους αρχικούς της στόχους. Τα μοντέλα αξιολογήθηκαν μέσω **του ROC-AUC** και της **βαθμολογίας του διαγωνισμού Kaggle**, με το XGBoost να επιτυγχάνει τα καλύτερα αποτελέσματα, με ROC-AUC **0,77** και βαθμολογία Kaggle **0,76** - κοντά στην κορυφαία βαθμολογία του διαγωνισμού **0,81**. Το KAN, αν και καινοτόμο, σημείωσε χαμηλότερο **ROC-AUC 0,71** και βαθμολογία Kaggle **0,70**. Η γραμμική παλινδρόμηση έμεινε πιο πίσω. Αυτά τα ευρήματα δείχνουν ότι ενώ το KAN έχει δυνατότητες, η εφαρμογή του σε αυτό το πλαίσιο μπορεί να απαιτήσει περαιτέρω βελτίωση για να επιτευχθεί ισοτιμία με πιο καθιερωμένα μοντέλα όπως το XGBoost.

4.4 Καινοτόμες συνεισφορές και Μελλοντικές Ερευνητικές Κατευθύνσεις

Αυτή η έρευνα συμβάλλει στον αυξανόμενο όγκο βιβλιογραφίας σχετικά με τη χρήση των **δικτύων Kolmogorov-Arnold** για προγνωστική μοντελοποίηση σε χρηματοοικονομικές εφαρμογές. Ενώ το μοντέλο KAN δεν ξεπέρασε το XGBoost σε αυτή τη μελέτη, η εφαρμογή του αντιπροσωπεύει μια καινοτόμο προσέγγιση στην **πρόβλεψη πιστωτικού κινδύνου**. Το μοντέλο έδειξε κάποιες δυνατότητες όταν αξιολογήθηκε με απλούστερες μετρήσεις μηχανικής μάθησης, γεγονός που υποδηλώνει ότι οι μελλοντικές βελτιώσεις θα μπορούσαν να βελτιώσουν την προγνωστική του ισχύ.

Διάφορες οδοί για μελλοντική έρευνα προκύπτουν από αυτό το έργο. Πρώτον, θα μπορούσαν να

διερευνηθούν πιο προηγμένες τεχνικές για το χειρισμό **δεδομένων που λείπουν** και **ακραίων τιμών** για τη μείωση της μεροληψίας και τη βελτίωση της ακρίβειας του μοντέλου. **Ο πολλαπλός καταλογισμός** ή **οι αυτόματοι κωδικοποιητές** μπορεί να προσφέρουν πιο εκλεπτυσμένες μεθόδους για την αντιμετώπιση των τιμών που λείπουν, ενώ η πιο εξελιγμένη ανίχνευση ακραίων τιμών θα μπορούσε να διασφαλίσει ότι οι ακραίες τιμές αντιμετωπίζονται κατάλληλα χωρίς να διακυβεύεται η ακεραιότητα του συνόλου δεδομένων. Δεύτερον, η περαιτέρω βελτίωση και εξερεύνηση των **δικτύων Kolmogorov-Arnold** θα μπορούσε να αποφέρει καλύτερα αποτελέσματα. Η μελλοντική έρευνα θα μπορούσε να επικεντρωθεί στη βελτιστοποίηση των υπερπαραμέτρων και στην ενσωμάτωση του KAN σε μεθόδους συνόλων.

Τέλος, η εφαρμογή αυτών των μοντέλων σε **διαφορετικά σύνολα δεδομένων** και σε διάφορες **γεωγραφικές περιοχές** θα διασφαλίσει ότι τα μοντέλα γενικεύονται αποτελεσματικά σε άλλες αγορές. Μια ευρύτερη, διακρατική μελέτη θα παρείχε βαθύτερες πληροφορίες σχετικά με την παγκόσμια δυνατότητα εφαρμογής αυτών των υποδειγμάτων πιστωτικού κινδύνου.

Συμπερασματικά, ενώ αυτή η μελέτη έκανε σημαντικά βήματα στην κατανόηση και την εφαρμογή μοντέλων μηχανικής μάθησης στην πρόβλεψη πιστωτικού κινδύνου, εξακολουθούν να υπάρχουν αρκετοί τομείς για μελλοντική βελτίωση. Με τη βελτίωση των μεθόδων χειρισμού δεδομένων, τη διερεύνηση των δυνατοτήτων του KAN και την ενίσχυση της εξηγησιμότητας του μοντέλου, η μελλοντική έρευνα μπορεί να βασιστεί στα θεμέλια που τέθηκαν από αυτό το έργο.

Πίνακας ορολογίας

Ξενόγλωσσος όρος	Ελληνικός Όρος
Gradient boosting	Ενίσχυση κλίσης
Logistic regression	Λογιστική παλινδρόμηση
Decision trees	Δέντρα αποφάσεων
Support vector machines	Μηχανές διανυσμάτων υποστήριξης
Neural networks	Νευρωνικά δίκτυα
Linear Discriminant Analysis	Γραμμική διακριτική ανάλυση
Random forests	Τυχαία δάση
Overfitting	Υπερβολική τοποθέτησης
Feature engineering	Εξαγωγή χαρακτηριστικών
K-nearest neighbors imputation	Καταλογισμός k-πλησιέστερων γειτόνων
One-hot encoding	κωδικοποίηση One-hot
Label encoding	κωδικοποίηση ετικετών
min-man scaling	κλιμάκωση min-max
z-score normalization	κανονικοποίηση z-score
Principal component analysis	ανάλυση κύριων συνιστωσών
recursive feature elimination	αναδρομική εξάλειψη χαρακτηριστικών
k-fold cross-validation	μέθοδος διασταυρούμενης επικύρωσης k-fold
grid search	αναζήτηση πλέγματος
max-abs scaling	κλιμάκωση max-abs
L2 regularization	τακτοποίηση L2
Gini coefficient	Συντελεστής Gini

Πίνακας συντμήσεων-αρτικόλεξων-ακρονυμίων

KAN	Δίκτυα Kolmogorov-Arnold
LDA	Γραμμική διακριτική ανάλυση
GBM	Μηχανές ενίσχυσης κλίσης
SVM	Μηχανές διανυσμάτων υποστήριξης
SMOTE	Synthetic Minority Over-sampling Technique
OHE	κωδικοποίηση One-hot
KNN	k-πλησιέστεροι γείτονες
PCA	ανάλυση κύριων συνιστωσών
RFE	αναδρομική εξάλειψη χαρακτηριστικών
AUC-ROC	Area Under the Receiver Operating Characteristic Curve

Βιβλιογραφία

[1] Home Credit Group - Home Credit Default Risk. Available at: https://www.kaggle.com/competitions/home-credit-default-risk

[2] Research on credit risk of commercial banks based on multiple logistic model- Jiexin Lu, Yongzhen Tong Available at: https://francis-press.com/papers/4291

[3] Machine Learning for Credit Risk Prediction: A Systematic Literature Review by Jomark Pablo Noriega,Luis Antonio Rivera, andJosé Alfredo Herrera Available at: https://www.mdpi.com/2306-5729/8/11/169

[4] Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research By Stefan Lessmann, Bart Baesens, Hsin-Vonn Seow, Lyn Thomas Available at: https://www.researchgate.net/publication/276280838_Benchmarking_state-of-the-art_classification_algorithms_for_credit_scoring_An_update_of_research

[5] S. Maiorov & A. Pinkus. Lower Bounds for Approximation by Kolmogorov Superpositions. Neural Networks, 1999 Available at: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=dfaa71334abffcfc1a6fccc59e2e25e0649dd2cd

[6] K. Hornik, M. Stinchcombe, & H. White. Multilayer Feedforward Networks are Universal Approximators. Neural Networks, 1991. Available at: https://www.cs.cmu.edu/~epxing/Class/10715/reading/Kornick_et_al.pdf

[7] AUC: a Better Measure than Accuracy in Comparing Learning Algorithms Charles X. Ling, Jin Huang Available at: https://www.site.uottawa.ca/~stan/csi7162/presentations/William-presentation.pdf

[8] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System Available at: https://arxiv.org/pdf/1603.02754

[9] KAN: Kolmogorov-Arnold Networks by Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle Available at: https://arxiv.org/html/2404.19756v

[10] ΕΙΚΟΝΑ 1.1: Λογιστική παλινδρόμηση Available at: https://wisdomml.in/logistic-regression-explained-definition-and-examples/

[11] ΕΙΚΟΝΑ 1.2: Εκπαίδευση μηχανής ενίσχυσης κλίσης Available at: https://vitalflux.com/gradient-boosting-algorithm-concepts-example/

[12] ΕΙΚΟΝΑ 1.3: Σύγκριση δικτύου KAN με Multilayer perceptron Available at: https://arxiv.org/pdf/2404.19756

[13] ΕΙΚΟΝΑ 1.4: κωδικοποίηση ετικετών και κωδικοποίηση One-hot (OHE) Available at: https://medium.com/@michaeldelsole/what-is-one-hot-encoding-and-how-to-do-it-

f0ae272f1179

[14] ΕΙΚΟΝΑ 1.5: ROC Curve Available at: https://medium.com/@ilyurek/roc-curve-and-auc-evaluating-model-performance-c2178008b02

[15] ΔΙΑΓΡΑΜΜΑ 2.1: Οι πίνακες δεδομένων και οι συσχετίσεις τους Available at: https://www.kaggle.com/competitions/home-credit-default-risk/data

Παράρτημα Α - Κώδικας Python

```
import warnings
warnings.filterwarnings("ignore")

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import phik
from datetime import datetime
import torch
import torch.nn as nn
import torch.optim as optim
from tqdm import tqdm
from torch.utils.data import DataLoader, Dataset, TensorDataset


from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import SGDClassifier
from sklearn.calibration import CalibratedClassifierCV
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score

from sklearn.metrics import f1_score

from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve,auc
from sklearn.metrics import confusion_matrix

# Method to reduce dataframes memory usage

def reduce_memory_usage(df):

    start_mem = df.memory_usage().sum() / 1024**2
    print('Memory usage of dataframe is {:.2f} MB'.format(start_mem))

    for col in df.columns:
        col_type = df[col].dtype

        if col_type != object:
            c_min = df[col].min()
            c_max = df[col].max()
            if str(col_type)[:3] == 'int':
                if c_min > np.iinfo(np.int8).min and c_max < np.iinfo(np.int8).max:
                    df[col] = df[col].astype(np.int8)
                elif c_min > np.iinfo(np.int16).min and c_max < np.iinfo(np.int16).max:
                    df[col] = df[col].astype(np.int16)
                elif c_min > np.iinfo(np.int32).min and c_max < np.iinfo(np.int32).max:
                    df[col] = df[col].astype(np.int32)
                elif c_min > np.iinfo(np.int64).min and c_max < np.iinfo(np.int64).max:
                    df[col] = df[col].astype(np.int64)
            else:
                if c_min > np.finfo(np.float16).min and c_max < np.finfo(np.float16).max:
```

```
                df[col] = df[col].astype(np.float16)
            elif c_min > np.finfo(np.float32).min and c_max < np.finfo(np.float32).max:
                df[col] = df[col].astype(np.float32)
            else:
                df[col] = df[col].astype(np.float64)

    end_mem = df.memory_usage().sum() / 1024**2
    print('Memory usage after optimization is: {:.2f} MB'.format(end_mem))
    print('Decreased by {:.1f}%'.format(100 * (start_mem - end_mem) / start_mem))

    return df

#method to find amount of missing values

def missing_values(dt):
    # number of missing values
    missing_num = dt.isnull().sum().values
    # total records
    total = dt.shape[0]
    # percentage of missing
    missing_ratio = missing_num/total
    # return a dataframe to show: feature name, # of missing and % of missing
    return pd.DataFrame(data={'missing_count':missing_num, 'missing_ratio':missing_ratio},
index=dt.columns.values).query('missing_ratio>0').sort_values(['missing_ratio'],
ascending=False)

train_data = reduce_memory_usage(pd.read_csv('application_train.csv'))
print("application_train")
print('Number of datapoints: ', train_data.shape[0])
print('Number of features: ', train_data.shape[1])
train_data.head()

missing_values(train_data).head(122)

test_data = reduce_memory_usage(pd.read_csv('application_test.csv'))
print('Number of datapoints: ', test_data.shape[0])
print('Number of features: ', test_data.shape[1])
test_data.head()

bureau_data = reduce_memory_usage(pd.read_csv('bureau.csv'))
print('Number of datapoints: ', bureau_data.shape[0])
print('Number of features: ', bureau_data.shape[1])
bureau_data.head()

missing_values(bureau_data).head(100)

bureau_balance = reduce_memory_usage(pd.read_csv('bureau_balance.csv'))
print('Number of datapoints: ', bureau_balance.shape[0])
print('Number of features: ', bureau_balance.shape[1])
bureau_balance.head()

missing_values(bureau_balance).head(100)

previous_application = reduce_memory_usage(pd.read_csv(previous_application.csv'))
print('Number of datapoints: ', previous_application.shape[0])
print('Number of features: ', previous_application.shape[1])
```

```
previous_application.head()

missing_values(previous_application).head(100)

pos_cash_balance = reduce_memory_usage(pd.read_csv('POS_CASH_balance.csv'))
print('Number of datapoints: ', pos_cash_balance.shape[0])
print('Number of features: ', pos_cash_balance.shape[1])
pos_cash_balance.head()

missing_values(pos_cash_balance).head(100)

installments_payments =
reduce_memory_usage(pd.read_csv('installments_payments.csv'))
print('Number of datapoints: ', installments_payments.shape[0])
print('Number of features: ', installments_payments.shape[1])
installments_payments.head()

missing_values(installments_payments).head(100)

credit_card_balance = reduce_memory_usage(pd.read_csv('credit_card_balance.csv'))
print('Number of datapoints: ', credit_card_balance.shape[0])
print('Number of features: ', credit_card_balance.shape[1])
credit_card_balance.head()

missing_values(credit_card_balance).head(100)

# Analyzing the target column to check the distribution of the dataset

y_value_counts = train_data['TARGET'].value_counts()
print("Number of customers who will not repay the loan on time: ", y_value_counts[1], ", (",
(y_value_counts[1]/(y_value_counts[1]+y_value_counts[0]))*100,"%)")
percentage_unpaid = (y_value_counts[1]/(y_value_counts[1]+y_value_counts[0]))
print("Number of customers who will repay the loan on time: ", y_value_counts[0], ", (",
(y_value_counts[0]/(y_value_counts[1]+y_value_counts[0]))*100,"%)")

fig, ax = plt.subplots(figsize=(6, 6), subplot_kw=dict(aspect="equal"))
recipe = ["Will not Repay", "Will Repay"]

data = [y_value_counts[1], y_value_counts[0]]

wedges, texts = ax.pie(data, wedgeprops=dict(width=0.5),\
                  startangle=-40)

bbox_props = dict(boxstyle="square,pad=0.3", fc="w", ec="k", lw=0.72)
kw = dict(xycoords='data', textcoords='data', arrowprops=dict(arrowstyle="-"),
          bbox=bbox_props, zorder=0, va="center")

for i, p in enumerate(wedges):
    ang = (p.theta2 - p.theta1)/2. + p.theta1
    y = np.sin(np.deg2rad(ang))
    x = np.cos(np.deg2rad(ang))
    horizontalalignment = {-1: "right", 1: "left"}[int(np.sign(x))]
    connectionstyle = "angle,angleA=0,angleB={}".format(ang)
    kw["arrowprops"].update({"connectionstyle": connectionstyle})
    ax.annotate(recipe[i], xy=(x, y), xytext=(1.35*np.sign(x), 1.4*y),
                horizontalalignment=horizontalalignment, **kw)
```

```
ax.set_title("Number of loans that are repaid and not repaid")

plt.show()

train_data.phik_matrix()

#methods to plot every feature with target

def plot_stats(df, feature,label_rotation=False,horizontal_layout=True):
    temp = df[feature].value_counts()
    df1 = pd.DataFrame({feature: temp.index,'Number of loans': temp.values})
    # Calculate the percentage of target=1 per category value
    cat_perc = df[[feature, 'TARGET']].groupby([feature],as_index=False).mean()
    cat_perc.sort_values(by='TARGET', ascending=False, inplace=True)
    if(horizontal_layout):
        fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(12,6))
    else:
        fig, (ax1, ax2) = plt.subplots(nrows=2, figsize=(12,14))
    sns.set_color_codes("pastel")
    s = sns.barplot(ax=ax1, x = feature, y="Number of loans",data=df1)
    if(label_rotation):
        s.set_xticklabels(s.get_xticklabels(),rotation=90)
    s = sns.barplot(ax=ax2, x = feature, y='TARGET', order=cat_perc[feature],
data=cat_perc)
    if(label_rotation):
        s.set_xticklabels(s.get_xticklabels(),rotation=90)
    plt.ylabel('Percent of unpaid loans [%]', fontsize=10)
    plt.tick_params(axis='both', which='major', labelsize=10)
    plt.axhline(percentage_unpaid, color='green')
    plt.show()

def plot_stats_continuous(df, feature, plots = ['distplot', 'CDF', 'box', 'violin'], scale_limits =
None, figsize = (20,8), histogram = True, log_scale = False):
    '''
    Function to plot continuous variables distribution

    Inputs:
        df: DataFrame
            The DataFrame from which to plot.
        feature: str
            Column's name whose distribution is to be plotted.
        plots: list, default = ['distplot', 'CDF', box', 'violin']
            List of plots to plot for Continuous Variable.
        scale_limits: tuple (left, right), default = None
            To control the limits of values to be plotted in case of outliers.
        figsize: tuple, default = (20,8)
            Size of the figure to be plotted.
        histogram: bool, default = True
            Whether to plot histogram along with distplot or not.
        log_scale: bool, default = False
            Whether to use log-scale for variables with outlying points.
    '''
    data_to_plot = df.copy()
    if scale_limits:
        data_to_plot[feature] = df[feature][(df[feature] > scale_limits[0]) & (df[feature] <
```

```
scale_limits[1])]

    number_of_subplots = len(plots)
    plt.figure(figsize = figsize)
    sns.set_style('whitegrid')

    for i, ele in enumerate(plots):
        plt.subplot(1, number_of_subplots, i + 1)
        plt.subplots_adjust(wspace=0.25)

        if ele == 'CDF':
            percentile_values_0 = data_to_plot[data_to_plot.TARGET ==
0][[feature]].dropna().sort_values(by = feature)
            percentile_values_0['Percentile'] = [ele / (len(percentile_values_0)-1) for ele in
range(len(percentile_values_0))]

            percentile_values_1 = data_to_plot[data_to_plot.TARGET ==
1][[feature]].dropna().sort_values(by = feature)
            percentile_values_1['Percentile'] = [ele / (len(percentile_values_1)-1) for ele in
range(len(percentile_values_1))]

            plt.plot(percentile_values_0[feature], percentile_values_0['Percentile'], color = 'red',
label = 'Non-Defaulters')
            plt.plot(percentile_values_1[feature], percentile_values_1['Percentile'], color =
'black', label = 'Defaulters')
            plt.xlabel(feature)
            plt.ylabel('Probability')
            plt.title('CDF of {}'.format(feature))
            plt.legend(fontsize = 'medium')
            if log_scale:
                plt.xscale('log')
                plt.xlabel(feature + ' - (log-scale)')

        if ele == 'distplot':
            sns.distplot(data_to_plot[feature][df['TARGET'] == 0].dropna(),
                         label='Paid', hist = False, color='green')
            sns.distplot(data_to_plot[feature][df['TARGET'] == 1].dropna(),
                         label='Unpaid', hist = False, color='red')
            plt.xlabel(feature)
            plt.ylabel('Probability Density')
            plt.legend(fontsize='medium')
            plt.title("Dist-Plot of {}".format(feature))
            if log_scale:
                plt.xscale('log')
                plt.xlabel(f'{feature} (log scale)')

        if ele == 'violin':
            sns.violinplot(x='TARGET', y=feature, data=data_to_plot)
            plt.title("Violin-Plot of {}".format(feature))
            if log_scale:
                plt.yscale('log')
                plt.ylabel(f'{feature} (log Scale)')

        if ele == 'box':
            sns.boxplot(x='TARGET', y=feature, data=data_to_plot, color='yellow')
            plt.title("Box-Plot of {}".format(feature))
```

```
        if log_scale:
            plt.yscale('log')
            plt.ylabel(f'{feature} (log Scale)')

    plt.show()

def plot_stats_categorical(df, var):
    nrow=var.__len__()
    i = 0

    sns.set_style('whitegrid')
    plt.figure()
    fig, ax = plt.subplots(nrow,2,figsize=(20,6*nrow))

    for feature in var:

        temp = df[feature].value_counts()
        df1 = pd.DataFrame({feature: temp.index,'Number of loans': temp.values})
        # Calculate the percentage of target=1 per category value
        cat_perc = df[[feature, 'TARGET']].groupby([feature],as_index=False).mean()
        cat_perc.sort_values(by='TARGET', ascending=False, inplace=True)

        i += 1
        plt.subplot(nrow,2,i)
        sns.set_color_codes("pastel")

        s = sns.barplot(x = feature, y="Number of loans",data=df1)
        s.set_xticklabels(s.get_xticklabels(),rotation=45)
        s.set_title(feature)

        i += 1
        plt.subplot(nrow,2,i)
        s = sns.barplot(x = feature, y='TARGET', order=cat_perc[feature], data=cat_perc)
        s.set_xticklabels(s.get_xticklabels(),rotation=45)
        s.set_title(feature)

        plt.ylabel('Percent of unpaid loans [%]', fontsize=10)
        plt.tick_params(axis='both', which='major', labelsize=10)
        plt.axhline(percentage_unpaid, color='green')


    plt.show();

categorical_features = ['REG_CITY_NOT_WORK_CITY', 'NAME_EDUCATION_TYPE',
'NAME_HOUSING_TYPE','REG_REGION_NOT_LIVE_REGION','CODE_GENDER',
'CNT_CHILDREN', 'NAME_INCOME_TYPE', 'FLAG_OWN_CAR', 'FLAG_OWN_REALTY',
'NAME_FAMILY_STATUS', 'NAME_CONTRACT_TYPE', 'REG_CITY_NOT_LIVE_CITY',
'REG_REGION_NOT_WORK_REGION','FLAG_DOCUMENT_2','FLAG_DOCUMENT_4','F
LAG_DOCUMENT_10','FLAG_DOCUMENT_12','FLAG_DOCUMENT_20',
'REGION_RATING_CLIENT', 'REGION_RATING_CLIENT_W_CITY']

plot_stats_categorical(train_data, categorical_features)

plot_stats(train_data, 'OCCUPATION_TYPE',True, False)

plot_stats(train_data, 'ORGANIZATION_TYPE',True, False)
```

```
plot_stats_continuous(train_data, 'EXT_SOURCE_1', plots=['distplot', 'box'], figsize=(20,8))

plot_stats_continuous(train_data, 'EXT_SOURCE_2', plots=['distplot', 'box'], figsize=(20,8))

plot_stats_continuous(train_data, 'EXT_SOURCE_3', plots=['distplot', 'box'], figsize=(20,8))

train_data['age_years'] = -train_data['DAYS_BIRTH']/365
plot_stats_continuous(train_data, 'age_years', plots=['distplot', 'violin'], figsize=(20,8))
train_data.pop('age_years');

train_data['years_employed'] = -train_data['DAYS_EMPLOYED']/365
plot_stats_continuous(train_data, 'years_employed', plots=['distplot', 'violin'], figsize=(20,8))
train_data.pop('years_employed');

train_data['years_employed'] = -train_data['DAYS_EMPLOYED']/365
train_data.loc[train_data["years_employed"] < 0, "years_employed"] = np.nan
train_data.loc[train_data["years_employed"] > 80, "years_employed"] = np.nan
plot_stats_continuous(train_data, 'years_employed', plots=['distplot', 'violin'], figsize=(20,8))
train_data.pop('years_employed');

plot_stats_continuous(train_data, 'FLOORSMAX_AVG', plots = ['box', 'violin'], figsize =
(10,8))

plot_stats_continuous(train_data, 'FLOORSMIN_MODE', plots = ['box', 'violin'], figsize =
(10,8))

plot_stats_continuous(train_data, 'DAYS_EMPLOYED', plots = ['box', 'violin'], figsize =
(10,8))

#Cleaning the application data

train_data['DAYS_EMPLOYED'].describe()

print("Minimum value(in years) = ",\
      max(train_data['DAYS_EMPLOYED'].values)/365)

print("Maximum value(in years) = ",\
      min(train_data['DAYS_EMPLOYED'].values)/365)

train_data.replace(max(train_data['DAYS_EMPLOYED'].values), np.nan, inplace=True)

test_data.replace(max(train_data['DAYS_EMPLOYED'].values), np.nan, inplace=True)

train_data['DAYS_REGISTRATION'].describe()
#%%
print("Minimum value(in years) = ",\
      max(train_data['DAYS_REGISTRATION'].values)/365)

print("Maximum value(in years) = ",\
      min(train_data['DAYS_REGISTRATION'].values)/365)

# removing the rows with ‘XNA’ as Gender

train_data = train_data[train_data['CODE_GENDER'] != 'XNA']

test_data = test_data[test_data['CODE_GENDER'] != 'XNA']
```

filling the categorical columns without value with 'XNA' value

```
categorical_columns = train_data.dtypes[train_data.dtypes == 'object'].index.tolist()
train_data[categorical_columns] = train_data[categorical_columns].fillna('XNA')
test_data[categorical_columns] = test_data[categorical_columns].fillna('XNA')

# The following FLAG_DOCUMENT features have one category for almost all data, we will
#remove those

flag_document_to_drop =
['FLAG_DOCUMENT_2','FLAG_DOCUMENT_4','FLAG_DOCUMENT_10','FLAG_DOCUM
ENT_12', 'FLAG_DOCUMENT_20']
train_data = train_data.drop(columns=flag_document_to_drop)
test_data = test_data.drop(columns=flag_document_to_drop)

numerical_columns_train = train_data.dtypes[train_data.dtypes != 'object'].index.tolist()
numerical_columns_test = test_data.dtypes[test_data.dtypes != 'object'].index.tolist()

train_data[numerical_columns_train] = train_data[numerical_columns_train].fillna(0)
test_data[numerical_columns_test] = test_data[numerical_columns_test].fillna(0)

train_data.shape

missing_values(train_data).head(100)

missing_values(test_data).head(100)

# Next we will create some new features based on the already existing ones

def Add_features_to_application(data):
   data['ANNUITY_INCOME_PERCENT'] = data['AMT_ANNUITY'] /
data['AMT_INCOME_TOTAL']
   data['FAMILY_CNT_INCOME_PERCENT'] = data['AMT_INCOME_TOTAL'] /
data['CNT_FAM_MEMBERS']
   data['CREDIT_TERM'] = data['AMT_ANNUITY'] / data['AMT_CREDIT']
   data['CREDIT_INCOME_PERCENT'] = data['AMT_CREDIT'] /
data['AMT_INCOME_TOTAL']
   data['BIRTH_EMPLOYED_PERCENT'] = data['DAYS_EMPLOYED'] /
data['DAYS_BIRTH']
   data['CREDIT_ANNUITY_PERCENT'] = data['AMT_CREDIT'] / data['AMT_ANNUITY']
   data['CHILDREN_CNT_INCOME_PERCENT'] =
data['AMT_INCOME_TOTAL']/data['CNT_CHILDREN']
   return data
#%%
expanded_train_data = Add_features_to_application(train_data)
expanded_train_data.shape
#%%

expanded_test_data = Add_features_to_application(test_data)
expanded_test_data.shape

# Now encode the categorical data, so they can be used by the models

expanded_ohe_train_data = expanded_train_data.copy()
# Create a label encoder object
```

```
le = LabelEncoder()
le_count = 0

# Iterate through the columns
for feature in expanded_ohe_train_data:
    if expanded_ohe_train_data[feature].dtype == 'object':


        # Train on the training data
        le.fit(expanded_ohe_train_data[feature])
        # Transform both training and testing data
        expanded_ohe_train_data[feature] =
le.transform(expanded_ohe_train_data[feature])

        # Keep track of how many columns were label encoded
        le_count += 1

print('%d columns were label encoded.' % le_count)
#%%
expanded_ohe_test_data = expanded_test_data.copy()
# Create a label encoder object
le = LabelEncoder()
le_count = 0

# Iterate through the columns
for feature in expanded_ohe_test_data:
    if expanded_ohe_test_data[feature].dtype == 'object':

        # Train on the training data
        le.fit(expanded_ohe_test_data[feature])
        # Transform both training and testing data
        expanded_ohe_test_data[feature] = le.transform(expanded_ohe_test_data[feature])

        # Keep track of how many columns were label encoded
        le_count += 1

print('%d columns were label encoded.' % le_count)

expanded_ohe_train_data.shape

expanded_ohe_test_data.shape

#Merging the TARGETS from application_train to bureau table.

bureau_data.head()

bureau_target = train_data.iloc[:,:2].merge(bureau_data, on = 'SK_ID_CURR', how = 'right')

bureau_target.head()

bureau_target.shape

missing_values(bureau_target)

categorical_features_bureau = bureau_target.dtypes[bureau_target.dtypes ==
'object'].index.tolist()
```

```
plot_stats_categorical(bureau_target, categorical_features_bureau)

numerical_features_bureau = bureau_target.dtypes[bureau_target.dtypes !=
'object'].index.tolist()
numerical_features_bureau.remove("TARGET")
numerical_features_bureau.remove("SK_ID_CURR")
numerical_features_bureau.remove("SK_ID_BUREAU")
for feature in numerical_features_bureau:
   plot_stats_continuous(bureau_target, feature, plots = ['distplot', 'box'], figsize = (10,8))
   print(bureau_target[feature].describe())

#We notice that there are outliers that negatively affect the column
#DAYS_ENDDATE_FACT , so we will remove them

# IQR
# Calculate the upper and lower limits

Q1 = bureau_target['DAYS_ENDDATE_FACT'].quantile(0.25)
Q3 = bureau_target['DAYS_ENDDATE_FACT'].quantile(0.75)
IQR = Q3 - Q1
lower = Q1 - 1.5*IQR
upper = Q3 + 1.5*IQR

# Create arrays of Boolean values indicating the outlier rows
upper_array = np.where(bureau_target['DAYS_ENDDATE_FACT'] >= upper)[0]
lower_array = np.where(bureau_target['DAYS_ENDDATE_FACT'] <= lower)[0]

# Removing the outliers
bureau_target.drop(index=upper_array, inplace=True)
bureau_target.drop(index=lower_array, inplace=True)
#plot_stats_continuous(bureau_target, 'DAYS_ENDDATE_FACT', plots = ['distplot', 'box'],
figsize = (10,8))
print(bureau_target['DAYS_ENDDATE_FACT'].describe())

#we drop DAYS_CREDIT_UPDATE with positive values and values less than 41.000 days
#(112 years ago)
drop_index = bureau_target.loc[(bureau_target['DAYS_CREDIT_UPDATE'] < -41000) |
(bureau_target['DAYS_CREDIT_UPDATE'] > 0)].index
bureau_target.drop(drop_index , inplace=True)
#plot_stats_continuous(bureau_target, 'DAYS_CREDIT_UPDATE', plots = ['distplot', 'box'],
figsize = (10,8))
print(bureau_target['DAYS_CREDIT_UPDATE'].describe())

#CREDIT_CURRENCY has very few values different from 'currency 1' so we drop them
drop_index = bureau_target.loc[bureau_target.CREDIT_CURRENCY != 'currency 1'].index
bureau_target.drop(drop_index , inplace=True)
bureau_target.drop(columns='CREDIT_CURRENCY')

#bureau_balace
bureau_balance_agg = pd.crosstab(bureau_balance.SK_ID_BUREAU,
bureau_balance.STATUS, normalize='index')
#%%
bureau_balance_agg.columns = [f'{bureau_balance_agg.columns.name}_{label}' for label in
bureau_balance_agg.columns]
bureau_balance_agg.rename_axis(None)
```

```
bureau_balance_agg.reset_index()

bureau_data_merged =  bureau_target.join(bureau_balance_agg, how='left',
on='SK_ID_BUREAU')

for feature in ['STATUS_0', 'STATUS_C', 'STATUS_X']:
   plot_stats_continuous(bureau_data_merged, feature, plots = ['distplot', 'box'], figsize =
(10,8))
   print(bureau_data_merged[feature].describe())

def Add_features_to_bureau(data):
   data['CREDIT_DURATION'] = data['DAYS_CREDIT_ENDDATE'] - data['DAYS_CREDIT']
   data['DEBT_PERCENTAGE'] = data['AMT_CREDIT_SUM'] /
data['AMT_CREDIT_SUM_DEBT']
   data['DEBT_CREDIT_DIFF'] = data['AMT_CREDIT_SUM'] -
data['AMT_CREDIT_SUM_DEBT']


   return data


expanded_bureau_data = Add_features_to_bureau(bureau_data_merged)
expanded_bureau_data.shape


expanded_bureau_data.groupby('SK_ID_CURR')['STATUS_0'].transform('count').head(100
00)
#%%
bureau_data_agg = expanded_bureau_data[['SK_ID_CURR']].copy()
bureau_data_agg['CUSTOMER_LOAN_COUNT'] =
expanded_bureau_data.groupby('SK_ID_CURR')['SK_ID_BUREAU'].transform('count')
bureau_data_agg =
bureau_data_agg.join(pd.crosstab(expanded_bureau_data['SK_ID_CURR'],
expanded_bureau_data['CREDIT_ACTIVE']), on='SK_ID_CURR')
bureau_data_agg['DAYS_CREDIT_mean'] =
expanded_bureau_data.groupby('SK_ID_CURR')['DAYS_CREDIT'].transform('mean') * -1
bureau_data_agg['DAYS_CREDIT_max'] =
expanded_bureau_data.groupby('SK_ID_CURR')['DAYS_CREDIT'].transform('max') * -1
bureau_data_agg['DAYS_CREDIT_min'] =
expanded_bureau_data.groupby('SK_ID_CURR')['DAYS_CREDIT'].transform('min') * -1
bureau_data_agg['CREDIT_DAY_OVERDUE_max'] =
expanded_bureau_data.groupby('SK_ID_CURR')['CREDIT_DAY_OVERDUE'].transform('m
ax')
bureau_data_agg['DAYS_CREDIT_ENDDATE_min'] =
expanded_bureau_data.groupby('SK_ID_CURR')['DAYS_CREDIT_ENDDATE'].transform('
min')
bureau_data_agg['DAYS_CREDIT_ENDDATE_max'] =
expanded_bureau_data.groupby('SK_ID_CURR')['DAYS_CREDIT_ENDDATE'].transform('
max')
bureau_data_agg['AMT_CREDIT_MAX_OVERDUE_max'] =
expanded_bureau_data.groupby('SK_ID_CURR')['AMT_CREDIT_MAX_OVERDUE'].transf
orm('max')
bureau_data_agg['CNT_CREDIT_PROLONG_max'] =
expanded_bureau_data.groupby('SK_ID_CURR')['CNT_CREDIT_PROLONG'].transform('m
ax')
```

```
bureau_data_agg['AMT_CREDIT_SUM_mean'] =
expanded_bureau_data.groupby('SK_ID_CURR')['AMT_CREDIT_SUM'].transform('mean')
bureau_data_agg['AMT_CREDIT_SUM_max'] =
expanded_bureau_data.groupby('SK_ID_CURR')['AMT_CREDIT_SUM'].transform('max')
bureau_data_agg['AMT_CREDIT_SUM_min'] =
expanded_bureau_data.groupby('SK_ID_CURR')['AMT_CREDIT_SUM'].transform('min')
bureau_data_agg['AMT_CREDIT_SUM_DEBT_max'] =
expanded_bureau_data.groupby('SK_ID_CURR')['AMT_CREDIT_SUM_DEBT'].transform('
max')
bureau_data_agg['AMT_CREDIT_SUM_LIMIT_max'] =
expanded_bureau_data.groupby('SK_ID_CURR')['AMT_CREDIT_SUM_LIMIT'].transform('
max')
bureau_data_agg['AMT_CREDIT_SUM_OVERDUE_max'] =
expanded_bureau_data.groupby('SK_ID_CURR')['AMT_CREDIT_SUM_OVERDUE'].transf
orm('max')
bureau_data_agg['DAYS_CREDIT_UPDATE_max'] =
expanded_bureau_data.groupby('SK_ID_CURR')['DAYS_CREDIT_UPDATE'].transform('m
ax') * -1
bureau_data_agg['STATUS_0'] =
expanded_bureau_data.groupby('SK_ID_CURR')['STATUS_0'].transform('sum') /
bureau_data_agg['CUSTOMER_LOAN_COUNT']
bureau_data_agg['STATUS_1'] =
expanded_bureau_data.groupby('SK_ID_CURR')['STATUS_1'].transform('sum') /
bureau_data_agg['CUSTOMER_LOAN_COUNT']
bureau_data_agg['STATUS_2'] =
expanded_bureau_data.groupby('SK_ID_CURR')['STATUS_2'].transform('sum') /
bureau_data_agg['CUSTOMER_LOAN_COUNT']
bureau_data_agg['STATUS_3'] =
expanded_bureau_data.groupby('SK_ID_CURR')['STATUS_3'].transform('sum') /
bureau_data_agg['CUSTOMER_LOAN_COUNT']
bureau_data_agg['STATUS_4'] =
expanded_bureau_data.groupby('SK_ID_CURR')['STATUS_4'].transform('sum') /
bureau_data_agg['CUSTOMER_LOAN_COUNT']
bureau_data_agg['STATUS_5'] =
expanded_bureau_data.groupby('SK_ID_CURR')['STATUS_5'].transform('sum') /
bureau_data_agg['CUSTOMER_LOAN_COUNT']
bureau_data_agg['STATUS_C'] =
expanded_bureau_data.groupby('SK_ID_CURR')['STATUS_C'].transform('sum') /
bureau_data_agg['CUSTOMER_LOAN_COUNT']
bureau_data_agg['STATUS_X'] =
expanded_bureau_data.groupby('SK_ID_CURR')['STATUS_X'].transform('sum') /
bureau_data_agg['CUSTOMER_LOAN_COUNT']

bureau_data_agg = bureau_data_agg.drop_duplicates(subset=['SK_ID_CURR'])

bureau_data_agg.fillna(0, inplace=True)


# POS_CASH_balance

pos_cash_balance_agg = pos_cash_balance[['SK_ID_PREV']].copy()
pos_cash_balance_agg['SK_DPD_mean'] =
pos_cash_balance.groupby('SK_ID_PREV')['SK_DPD'].transform('mean')
pos_cash_balance_agg['SK_DPD_max'] =
pos_cash_balance.groupby('SK_ID_PREV')['SK_DPD'].transform('max')
pos_cash_balance_agg['SK_DPD_DEF_mean'] =
pos_cash_balance.groupby('SK_ID_PREV')['SK_DPD_DEF'].transform('mean')
```

```
pos_cash_balance_agg['SK_DPD_DEF_max'] =
pos_cash_balance.groupby('SK_ID_PREV')['SK_DPD_DEF'].transform('max')

pos_cash_balance_agg =
pos_cash_balance_agg.drop_duplicates(subset=['SK_ID_PREV'])

pos_cash_balance_agg.rename(columns={'Active': 'pos_cash_Active', 'Amortized debt':
'pos_cash_Amortized debt', 'Approved': 'pos_cash_Approved', 'Canceled':
'pos_cash_Canceled', 'Completed': 'pos_cash_Completed', 'Demand': 'pos_cash_Demand',
'Returned to the store': 'pos_cash_Returned to the store', 'Signed': 'pos_cash_Signed',
'XNA': 'pos_cash_XNA'}, inplace=True)
pos_cash_balance_agg.head()


#Credit_Card_Balance

credit_card_balance['Percentage_of_limit_drawn'] =
credit_card_balance['AMT_DRAWINGS_CURRENT'] /
credit_card_balance['AMT_CREDIT_LIMIT_ACTUAL']
credit_card_balance['Percentage_of_min_payment'] =
credit_card_balance['AMT_PAYMENT_TOTAL_CURRENT'] /
credit_card_balance['AMT_INST_MIN_REGULARITY']
credit_card_balance['Percentage_of_receivable_principal'] =
credit_card_balance['AMT_TOTAL_RECEIVABLE'] /
credit_card_balance['AMT_RECEIVABLE_PRINCIPAL']
credit_card_balance['Amount_per_drawing'] =
credit_card_balance['AMT_DRAWINGS_CURRENT'] /
credit_card_balance['CNT_DRAWINGS_CURRENT']


credit_card_balance_agg = credit_card_balance[['SK_ID_PREV']].copy()
credit_card_balance_agg['AMT_BALANCE_mean'] =
credit_card_balance.groupby('SK_ID_PREV')['AMT_BALANCE'].transform('mean')
credit_card_balance_agg['Percentage_of_limit_drawn_mean'] =
credit_card_balance.groupby('SK_ID_PREV')['Percentage_of_limit_drawn'].transform('mea
n')
credit_card_balance_agg['AMT_DRAWINGS_CURRENT_mean'] =
credit_card_balance.groupby('SK_ID_PREV')['AMT_DRAWINGS_CURRENT'].transform('m
ean')
credit_card_balance_agg['Percentage_of_min_payment_mean'] =
credit_card_balance.groupby('SK_ID_PREV')['Percentage_of_min_payment'].transform('me
an')
credit_card_balance_agg['Percentage_of_receivable_principal_mean'] =
credit_card_balance.groupby('SK_ID_PREV')['Percentage_of_receivable_principal'].transfor
m('mean')
credit_card_balance_agg['Amount_per_drawing_mean'] =
credit_card_balance.groupby('SK_ID_PREV')['Amount_per_drawing'].transform('mean')
credit_card_balance_agg['CNT_DRAWINGS_CURRENT_mean'] =
credit_card_balance.groupby('SK_ID_PREV')['CNT_DRAWINGS_CURRENT'].transform('m
ean')
pos_cash_balance_agg['SK_DPD_mean'] =
pos_cash_balance.groupby('SK_ID_PREV')['SK_DPD'].transform('mean')
pos_cash_balance_agg['SK_DPD_max'] =
pos_cash_balance.groupby('SK_ID_PREV')['SK_DPD'].transform('max')
pos_cash_balance_agg['Credit_Card_SK_DPD_DEF_mean'] =
pos_cash_balance.groupby('SK_ID_PREV')['SK_DPD_DEF'].transform('mean')
pos_cash_balance_agg['Credit_Card_SK_DPD_DEF_max'] =
```

```
pos_cash_balance.groupby('SK_ID_PREV')['SK_DPD_DEF'].transform('max')

credit_card_balance_agg =
credit_card_balance_agg.drop_duplicates(subset=['SK_ID_PREV'])

credit_card_balance_agg.fillna(0, inplace=True)


# installments_payments
# Very small amount of rows with missing values, dropping them

installments_payments.dropna(inplace=True);

installments_payments['DAYS_LATE_PAYMENT'] =
installments_payments['DAYS_ENTRY_PAYMENT'] -
installments_payments['DAYS_INSTALMENT']
installments_payments['PAYMENT_AMOUNT_MISSING'] =
installments_payments['AMT_INSTALMENT'] - installments_payments['AMT_PAYMENT']
#%%
installments_payments_agg = installments_payments[['SK_ID_PREV']].copy()
installments_payments_agg['DAYS_LATE_PAYMENT_mean'] =
installments_payments.groupby('SK_ID_PREV')['DAYS_LATE_PAYMENT'].transform('mea
n')
installments_payments_agg['PAYMENT_AMOUNT_MISSING_mean'] =
installments_payments.groupby('SK_ID_PREV')['PAYMENT_AMOUNT_MISSING'].transfor
m('mean')

installments_payments_agg =
installments_payments_agg.drop_duplicates(subset=['SK_ID_PREV'])


# previous_application
# columns RATE_INTEREST_PRIMARY and RATE_INTEREST_PRIVILEGED are
#removed because they are 96% empty

previous_application.drop(columns=['RATE_INTEREST_PRIMARY',
'RATE_INTEREST_PRIVILEGED'])

previous_application_merged =  previous_application.merge(pos_cash_balance_agg,
how='left', on='SK_ID_PREV')
previous_application_merged =
previous_application_merged.merge(credit_card_balance_agg, how='left',
on='SK_ID_PREV')
previous_application_merged =
previous_application_merged.merge(installments_payments_agg, how='left',
on='SK_ID_PREV')
#%%
previous_application_merged.head()

previous_application_target = train_data.iloc[:,:2].merge(previous_application_merged, on =
'SK_ID_CURR', how = 'right')
previous_application_target.head()

previous_application_target.phik_matrix()
```

```
categorical_features_previous =
previous_application_target.dtypes[previous_application_target.dtypes ==
'object'].index.tolist()

plot_stats_categorical(previous_application_target, categorical_features_previous)
#%%
numerical_features_previous =
previous_application_target.dtypes[previous_application_target.dtypes !=
'object'].index.tolist()
numerical_features_previous.remove("TARGET")
numerical_features_previous.remove("SK_ID_CURR")
numerical_features_previous.remove("SK_ID_PREV")
for feature in numerical_features_previous:
   plot_stats_continuous(previous_application_target, feature, plots=['distplot', 'box'],
figsize=(10, 8))
   print(previous_application_target[feature].describe())

#cleaning the data
# days fields should be negative or 0

previous_application_target['DAYS_FIRST_DRAWING'].replace(max(previous_application_
target['DAYS_FIRST_DRAWING'].values), np.nan, inplace=True)
previous_application_target['DAYS_FIRST_DRAWING'].describe()

previous_application_target['DAYS_FIRST_DUE'].replace(max(previous_application_target[
'DAYS_FIRST_DUE'].values),\
                                    np.nan, inplace=True)
previous_application_target['DAYS_FIRST_DUE'].describe()

previous_application_target['DAYS_LAST_DUE_1ST_VERSION'].loc[previous_application_
target['DAYS_LAST_DUE_1ST_VERSION'] > 0] = np.nan
previous_application_target['DAYS_LAST_DUE_1ST_VERSION'].describe()

previous_application_target['DAYS_LAST_DUE'].replace(max(previous_application_target['
DAYS_LAST_DUE'].values),\
                                    np.nan, inplace=True)
previous_application_target['DAYS_LAST_DUE'].describe()

previous_application_target['DAYS_TERMINATION'].replace(max(previous_application_tar
get['DAYS_TERMINATION'].values),\
                                    np.nan, inplace=True)
previous_application_target['DAYS_TERMINATION'].describe()

previous_application_target[categorical_features_previous] =
previous_application_target[categorical_features_previous].fillna('XNA')

# Creating new features based on the analysis
previous_application_target['AMT_DECLINED'] =
previous_application_target['AMT_APPLICATION'] -
previous_application_target['AMT_CREDIT']
previous_application_target['AMT_CREDIT_GOODS_DIFF'] =
previous_application_target['AMT_CREDIT'] -
previous_application_target['AMT_GOODS_PRICE']
previous_application_target['CREDIT_DOWNPAYMENT_RATIO'] =
previous_application_target['AMT_DOWN_PAYMENT'] /
previous_application_target['AMT_CREDIT']
```

```
#Aggregating to merge with train_data

prev_app_agg = previous_application_target[['SK_ID_CURR']].copy()

prev_app_agg['applications_count'] =
previous_application_target.groupby('SK_ID_CURR')['SK_ID_PREV'].transform('count')
prev_app_agg =
prev_app_agg.join(pd.crosstab(previous_application_target['SK_ID_CURR'],
previous_application_target['NAME_CONTRACT_TYPE']), on='SK_ID_CURR')
prev_app_agg['AMT_ANNUITY_mean'] =
previous_application_target.groupby('SK_ID_CURR')['AMT_ANNUITY'].transform('mean') *
-1
prev_app_agg['AMT_ANNUITY_max'] =
previous_application_target.groupby('SK_ID_CURR')['AMT_ANNUITY'].transform('max') * -
1
prev_app_agg['AMT_ANNUITY_min'] =
previous_application_target.groupby('SK_ID_CURR')['AMT_ANNUITY'].transform('min') * -1
prev_app_agg['AMT_APPLICATION_max'] =
previous_application_target.groupby('SK_ID_CURR')['AMT_APPLICATION'].transform('max
')
prev_app_agg['AMT_APPLICATION_min'] =
previous_application_target.groupby('SK_ID_CURR')['AMT_APPLICATION'].transform('min'
)
prev_app_agg['AMT_APPLICATION_mean'] =
previous_application_target.groupby('SK_ID_CURR')['AMT_APPLICATION'].transform('me
an')
prev_app_agg['AMT_BALANCE_mean_max'] =
previous_application_target.groupby('SK_ID_CURR')['AMT_BALANCE_mean'].transform('
max')
prev_app_agg['AMT_BALANCE_mean_min'] =
previous_application_target.groupby('SK_ID_CURR')['AMT_BALANCE_mean'].transform('
min')
prev_app_agg['AMT_BALANCE_mean_mean'] =
previous_application_target.groupby('SK_ID_CURR')['AMT_BALANCE_mean'].transform('
mean')
prev_app_agg['AMT_CREDIT_max'] =
previous_application_target.groupby('SK_ID_CURR')['AMT_CREDIT'].transform('max')
prev_app_agg['AMT_CREDIT_min'] =
previous_application_target.groupby('SK_ID_CURR')['AMT_CREDIT'].transform('min')
prev_app_agg['AMT_CREDIT_mean'] =
previous_application_target.groupby('SK_ID_CURR')['AMT_CREDIT'].transform('mean')
prev_app_agg['AMT_GOODS_PRICE_max'] =
previous_application_target.groupby('SK_ID_CURR')['AMT_GOODS_PRICE'].transform('m
ax')
prev_app_agg['AMT_GOODS_PRICE_min'] =
previous_application_target.groupby('SK_ID_CURR')['AMT_GOODS_PRICE'].transform('mi
n')
prev_app_agg['AMT_GOODS_PRICE_mean'] =
previous_application_target.groupby('SK_ID_CURR')['AMT_GOODS_PRICE'].transform('m
ean')
prev_app_agg['DAYS_FIRST_DRAWING_max'] =
previous_application_target.groupby('SK_ID_CURR')['DAYS_FIRST_DRAWING'].transform
('max') * -1
prev_app_agg['DAYS_FIRST_DRAWING_min'] =
previous_application_target.groupby('SK_ID_CURR')['DAYS_FIRST_DRAWING'].transform
('min') * -1
prev_app_agg['DAYS_LAST_DUE_max'] =
```

```
previous_application_target.groupby('SK_ID_CURR')['DAYS_LAST_DUE'].transform('max')
* -1
prev_app_agg['DAYS_LAST_DUE_min'] =
previous_application_target.groupby('SK_ID_CURR')['DAYS_LAST_DUE'].transform('min')
* -1
prev_app_agg['AMT_DECLINED_max'] =
previous_application_target.groupby('SK_ID_CURR')['AMT_DECLINED'].transform('max')
prev_app_agg['AMT_DECLINED_min'] =
previous_application_target.groupby('SK_ID_CURR')['AMT_DECLINED'].transform('min')
prev_app_agg['AMT_DECLINED_mean'] =
previous_application_target.groupby('SK_ID_CURR')['AMT_DECLINED'].transform('mean')
prev_app_agg['AMT_CREDIT_GOODS_DIFF_max'] =
previous_application_target.groupby('SK_ID_CURR')['AMT_CREDIT_GOODS_DIFF'].transf
orm('max')
prev_app_agg['AMT_CREDIT_GOODS_DIFF_min'] =
previous_application_target.groupby('SK_ID_CURR')['AMT_CREDIT_GOODS_DIFF'].transf
orm('min')
prev_app_agg['AMT_CREDIT_GOODS_DIFF_mean'] =
previous_application_target.groupby('SK_ID_CURR')['AMT_CREDIT_GOODS_DIFF'].transf
orm('mean')
prev_app_agg['CREDIT_DOWNPAYMENT_RATIO_max'] =
previous_application_target.groupby('SK_ID_CURR')['CREDIT_DOWNPAYMENT_RATIO'].
transform('max')
prev_app_agg['CREDIT_DOWNPAYMENT_RATIO_min'] =
previous_application_target.groupby('SK_ID_CURR')['CREDIT_DOWNPAYMENT_RATIO'].
transform('min')
prev_app_agg['CREDIT_DOWNPAYMENT_RATIO_mean'] =
previous_application_target.groupby('SK_ID_CURR')['CREDIT_DOWNPAYMENT_RATIO'].
transform('mean')


prev_app_agg[['Approved_count', 'Canceled_count', 'Refused_count',
'Unused_offer_count']] = pd.crosstab(previous_application_target['SK_ID_CURR'],
previous_application_target['NAME_CONTRACT_STATUS']).reset_index(drop=True)
prev_app_agg[['CLIENT', 'HC', 'LIMIT', 'SCO', 'SCOFR', 'SYSTEM', 'VERIF', 'XAP', 'XNA']] =
pd.crosstab(previous_application_target['SK_ID_CURR'],
previous_application_target['CODE_REJECT_REASON']).reset_index(drop=True)
prev_app_agg = prev_app_agg.drop_duplicates(subset=['SK_ID_CURR'])
prev_app_agg.fillna(0, inplace=True)

train_data_temp1 = expanded_ohe_train_data.merge(prev_app_agg, how='left',
on='SK_ID_CURR')
train_data_temp1 = train_data_temp1.merge(bureau_data_agg, how='left',
on='SK_ID_CURR')

test_data_temp1 = expanded_ohe_test_data.merge(prev_app_agg, how='left',
on='SK_ID_CURR')
test_data_temp1 = test_data_temp1.merge(bureau_data_agg, how='left',
on='SK_ID_CURR')

# the merged data have nan values for the applications that didn't have previous
#applications, so we will remove them

train_data_temp1.fillna(0, inplace=True)
test_data_temp1.fillna(0, inplace=True)
```

```
#saving the clean data for quick access
train_data_temp1.to_csv('train1.csv', index=False)
test_data_temp1.to_csv('test1.csv', index=False)


#max-abs scaling
train_data1_no_target = train_data1_no_target /
train_data1_no_target.abs().max().astype(np.float64)
train_data1_no_target.fillna(0, inplace=True)

test_data1 = test_data1 / test_data1.abs().max().astype(np.float64)
test_data1.fillna(0, inplace=True)


#split the data to train and validation

X_train_final, X_cv_final, Y_train_final, Y_cv_final = train_test_split(train_data1_no_target,
target, test_size=0.20, \
                                                    stratify=target)
print(X_train_final.shape, Y_train_final.shape)
print(X_cv_final.shape, Y_cv_final.shape)

def obtain_threshold(thresholds,tpr,fpr):

    obtain_threshold.best_tradeoff = tpr*(1-fpr)
    ideal_threshold = thresholds[obtain_threshold.best_tradeoff.argmax()]

    return ideal_threshold

def plot_confusion_matrix(test_y, predict_y):

    C = confusion_matrix(test_y, predict_y)

    A =(((C.T)/(C.sum(axis=1))).T)

    B =(C/C.sum(axis=0))

    plt.figure(figsize=(20,4))

    labels = [0,1]

    cmap=sns.light_palette("blue")
    plt.subplot(1, 3, 1)
    sns.set(font_scale=1.1)
    sns.set_style(style='white')

    sns.heatmap(C, annot=True, cmap=cmap, fmt=".10f", xticklabels=labels, \
            yticklabels=labels)
    plt.xlabel('Predicted Class')
    plt.ylabel('Original Class')
    plt.title("Confusion matrix")

    plt.subplot(1, 3, 2)
    sns.heatmap(B,annot=True, cmap=cmap, fmt=".10f", xticklabels=labels, \
            yticklabels=labels)
    plt.xlabel('Predicted Class')
```

```
    plt.ylabel('Original Class')
    plt.title("Precision matrix")

    plt.subplot(1, 3, 3)
    # representing B in heatmap format
    sns.heatmap(A, annot=True, cmap=cmap, fmt=".10f", xticklabels=labels, \
                yticklabels=labels)
    plt.xlabel('Predicted Class')
    plt.ylabel('Original Class')
    plt.title("Recall matrix")

    plt.show()


# Logistic regression

alpha = [10 ** x for x in range(-5, 2)] # hyperparameter for SGD classifier.

train_auc=[]
validation_auc=[]

roc_auc_array=[]

start = datetime.now()

roc_auc_array=[]
for i in alpha:
    classifier = SGDClassifier(alpha=i, penalty='l2', loss='log_loss', random_state=57,
class_weight='balanced')
    classifier.fit(X_train_final, Y_train_final)

    sig_classifier = CalibratedClassifierCV(classifier, method="sigmoid")
    sig_classifier.fit(X_train_final, Y_train_final)

    Y_train_pred = sig_classifier.predict_proba( X_train_final)[:,1]
    Y_cv_pred = sig_classifier.predict_proba( X_cv_final)[:,1]

    train_auc.append(roc_auc_score(Y_train_final,Y_train_pred))
    validation_auc.append(roc_auc_score(Y_cv_final,Y_cv_pred))

    roc_auc_array.append(roc_auc_score(Y_cv_final, Y_cv_pred))
    print('For values of alpha = ', i, "the roc_auc score is:",
roc_auc_score(Y_cv_final,Y_cv_pred))

fig, ax = plt.subplots()
ax.plot(alpha, roc_auc_array,c='g')
for i, txt in enumerate(np.round(roc_auc_array,3)):
    ax.annotate((alpha[i],np.round(txt,3)), (alpha[i],roc_auc_array[i]))
plt.grid()
plt.title("Cross Validation AUC for each alpha")
plt.xlabel("Alpha i's")
plt.ylabel("ROC_AUC Score")
plt.show()

best_alpha = np.argmax(roc_auc_array)
classifier = SGDClassifier(alpha=alpha[best_alpha], penalty='l2', loss='log_loss',
```

```
random_state=42)

classifier.fit(X_train_final,  Y_train_final)
sig_classifier = CalibratedClassifierCV(classifier, method="sigmoid")
sig_classifier.fit(X_train_final, Y_train_final)

predict_y_train = sig_classifier.predict_proba( X_train_final)[:,1]
print('For values of best alpha = ', alpha[best_alpha], "The train roc_auc is:",\
                        roc_auc_score(Y_train_final, predict_y_train))

predict_y_cv = sig_classifier.predict_proba( X_cv_final)[:,1]
print('For values of best alpha = ', alpha[best_alpha], "The cv roc_auc is:",\
                        roc_auc_score(Y_cv_final, predict_y_cv))

print(" "*100)
print("Time taken to run this cell :", datetime.now() - start)

start = datetime.now()

clf = SGDClassifier(alpha=10**-5, penalty='l2', loss='log_loss',
random_state=42,class_weight='balanced')
clf.fit(X_train_final, Y_train_final)

sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(X_train_final, Y_train_final)

train_fpr1, train_tpr1, tr_thresholds1 =
roc_curve(Y_train_final,sig_clf.predict_proba(X_train_final)[:,1])
cv_fpr1,cv_tpr1, cv_thresholds1 = roc_curve(Y_cv_final,
sig_clf.predict_proba(X_cv_final)[:,1])

plt.plot(train_fpr1,train_tpr1, label ="Training Data AUC :" + str(auc(train_fpr1,train_tpr1)))
plt.plot(cv_fpr1,cv_tpr1,label="CV Data AUC :" + str(auc(cv_fpr1,cv_tpr1)))
plt.legend()

plt.xlabel("FPR Values")
plt.ylabel("TPR Values")
plt.title('ROC Curve: Logistic Regression Model')

plt.grid(False)
plt.show()

print('Ideal Threshold for the CV Dataset =',
obtain_threshold(cv_thresholds1,cv_tpr1,cv_fpr1))
print("Time taken to run this cell :", datetime.now() - start)


Y_cv_final_plot = Y_cv_final
Y_cv_final_pred = sig_clf.predict(X_cv_final)

plot_confusion_matrix(Y_cv_final_plot, Y_cv_final_pred)


precision = precision_score(Y_cv_final_plot, Y_cv_final_pred)
recall = recall_score(Y_cv_final_plot, Y_cv_final_pred)
f1score = f1_score(Y_cv_final_plot, Y_cv_final_pred)
```

```
print(f"Precision = {precision}")
print(f"Recall = {recall}")
print(f"F1 Score = {f1score}")



from sklearn.linear_model import LogisticRegression
test_data_temp = test_data1.copy()
for col in test_data_temp.columns:
   if col=='TARGET':
      test_data_temp = test_data_temp.drop(['TARGET'],axis=1)
   if col =='SK_ID_CURR':
      test_data_temp = test_data_temp.drop(['SK_ID_CURR'],axis=1)

logistic_regression = LogisticRegression(C=10**-5,
class_weight='balanced',random_state=42,penalty='l2')
logistic_regression.fit(X_train_final, Y_train_final)

#scaler_test = np.nan_to_num(test_data_temp)
lr_test_predict = logistic_regression.predict_proba(test_data_temp)[:,1]
#%%
lr_test_predict.shape

#saving the logistic regression results to use in the Kaggle competition

test_data_temp['SK_ID_CURR'] = test_data['SK_ID_CURR']
test_data_temp['TARGET'] = lr_test_predict
test_data_temp['SK_ID_CURR'] = test_data_temp['SK_ID_CURR'].apply(lambda x:
np.int32(x))
test_data_temp[['SK_ID_CURR', 'TARGET']].to_csv('Logistic_results.csv', index= False)


#XGBoost
import xgboost as xgb

max_depth = range(3, 11) # hyperparam for SGD classifier.

train_auc=[]
cv_auc=[]

roc_auc_array=[]

start = datetime.now()

roc_auc_array=[]

for i in max_depth:

   import xgboost as xgb
   xgb2 = xgb.XGBClassifier(objective='binary:logistic', max_depth=i, n_jobs=3)
   xgb2.fit(X_train_final,Y_train_final)

   Y_train_pred = xgb2.predict_proba( X_train_final)[:,1]
   Y_cv_pred = xgb2.predict_proba(X_cv_final)[:,1]
```

```
    train_auc.append(roc_auc_score(Y_train_final,Y_train_pred))
    cv_auc.append(roc_auc_score(Y_cv_final,Y_cv_pred))

    roc_auc_array.append(roc_auc_score(Y_cv_final, Y_cv_pred))
    print('For values of max_depth = ', i, "the roc_auc score is:",
roc_auc_score(Y_cv_final,Y_cv_pred))

fig, ax = plt.subplots()
ax.plot(max_depth, roc_auc_array,c='g')
for i, txt in enumerate(np.round(roc_auc_array,3)):
    ax.annotate((max_depth[i],np.round(txt,3)), (max_depth[i],roc_auc_array[i]))
plt.grid()
plt.title("Cross Validation AUC for each alpha")
plt.xlabel("Max_Depth i's")
plt.ylabel("ROC_AUC Score")
plt.show()

best_depth = np.argmax(roc_auc_array)

xgb2 = xgb.XGBClassifier(objective='binary:logistic', max_depth=best_depth,\
                    n_jobs=3)
xgb2.fit(X_train_final,Y_train_final)

predict_y_train = xgb2.predict_proba(X_train_final)[:,1]
print('For values of best max_depth = ', max_depth[best_depth], "The train roc_auc is:",\
                            roc_auc_score(Y_train_final, predict_y_train))

predict_y_cv = xgb2.predict_proba(X_cv_final)[:,1]
print('For values of best max_depth = ', max_depth[best_depth], "The cv roc_auc is:",\
                            roc_auc_score(Y_cv_final, predict_y_cv))

print(" "*100)
print("Time taken to run this cell :", datetime.now() - start)


min_child_weight = range(5, 16, 2) # hyperparam for SGD classifier.

train_auc=[]
cv_auc=[]

roc_auc_array=[]

start = datetime.now()

roc_auc_array=[]

for i in min_child_weight:

    import xgboost as xgb
    xgb1 = xgb.XGBClassifier(objective='binary:logistic', min_child_weight=i,n_jobs=-1)
    xgb1.fit(X_train_final,Y_train_final)

    Y_train_pred = xgb1.predict_proba( X_train_final)[:,1]
    Y_cv_pred = xgb1.predict_proba( X_cv_final)[:,1]

    train_auc.append(roc_auc_score(Y_train_final,Y_train_pred))
```

```
    cv_auc.append(roc_auc_score(Y_cv_final,Y_cv_pred))

    roc_auc_array.append(roc_auc_score(Y_cv_final, Y_cv_pred))
    print('For values of min_child_weight = ', i, "the roc_auc score is:",
roc_auc_score(Y_cv_final,Y_cv_pred))

fig, ax = plt.subplots()
ax.plot(min_child_weight, roc_auc_array,c='g')
for i, txt in enumerate(np.round(roc_auc_array,3)):
    ax.annotate((min_child_weight[i],np.round(txt,3)), (min_child_weight[i],roc_auc_array[i]))
plt.grid()
plt.title("Cross Validation AUC for each alpha")
plt.xlabel("Min_child_weight i's")
plt.ylabel("ROC_AUC Score")
plt.show()

best_weight = np.argmax(roc_auc_array)

xgb1 = xgb.XGBClassifier(objective='binary:logistic', min_child_weight=best_weight,\
                    n_jobs=3)
xgb1.fit(X_train_final,Y_train_final)

predict_y_train = xgb1.predict_proba(X_train_final)[:,1]
print('For values of best min_child_weight = ', min_child_weight[best_weight], "The train
roc_auc is:",\
                            roc_auc_score(Y_train_final, predict_y_train))

predict_y_cv = xgb1.predict_proba(X_cv_final)[:,1]
print('For values of best min_child_weight = ', min_child_weight[best_weight], "The cv
roc_auc is:",\
                            roc_auc_score(Y_cv_final, predict_y_cv))

print(" "*100)
print("Time taken to run this cell :", datetime.now() - start)


#roc curve

start = datetime.now()

xgb_model = xgb.XGBClassifier(objective='binary:logistic',eval_metric = 'auc',
min_child_weight=13, max_depth=4, n_jobs=-1)
xgb_model.fit(X_train_final, Y_train_final)

train_fpr4, train_tpr4, tr_thresholds4 =
roc_curve(Y_train_final,xgb_model.predict_proba(X_train_final)[:,1])
cv_fpr4,cv_tpr4, cv_thresholds4 = roc_curve(Y_cv_final,
xgb_model.predict_proba(X_cv_final)[:,1])

plt.plot(train_fpr4,train_tpr4, label ="Training Data AUC :" + str(auc(train_fpr4,train_tpr4)))
plt.plot(cv_fpr4,cv_tpr4,label="CV Data AUC :" + str(auc(cv_fpr4,cv_tpr4)))
plt.legend()

plt.xlabel("FPR Values")
plt.ylabel("TPR Values")
plt.title('ROC Curve: XGBoost Model')
```

```
plt.grid(False)
plt.show()

print('Ideal Threshold for the CV Dataset =',
obtain_threshold(cv_thresholds4,cv_tpr4,cv_fpr4))
print("Time taken to run this cell :", datetime.now() - start)


Y_cv_final_plot_4 = Y_cv_final
Y_cv_final_pred_4 = xgb_model.predict(X_cv_final)

plot_confusion_matrix(Y_cv_final_plot_4, Y_cv_final_pred_4)


precision = precision_score(Y_cv_final_plot_4, Y_cv_final_pred_4)
recall = recall_score(Y_cv_final_plot_4, Y_cv_final_pred_4)
f1score = f1_score(Y_cv_final_plot_4, Y_cv_final_pred_4)

print(f"Precision = {precision}")
print(f"Recall = {recall}")
print(f"F1 Score = {f1score}")

#feature importance
features = test_data1.columns
importances = xgb_model.feature_importances_
indices = (np.argsort(importances))[-25:]
plt.figure(figsize=(6,8))
plt.title('Feature Importances')
plt.barh(range(len(indices)), importances[indices], color='red', align='center')
plt.yticks(range(len(indices)), [features[i] for i in indices])
plt.xlabel('Relative Importance')
plt.show()



#KAN
input = train_data1_no_target.drop(columns= 'SK_ID_CURR').copy()
test = test_data_norm.drop(columns= 'SK_ID_CURR').copy()

input_arr = np.float32(np.array(input))
target_arr = np.float32(np.array(target))
train_input, val_input, train_target, val_target = train_test_split(input_arr, target_arr,
                                                    test_size=0.2,
                                                    random_state=42)
test_arr = np.float32(np.array(test))

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
print(device)

train_input_tensor = torch.tensor(train_input).to(device)
train_target_tensor = torch.tensor(train_target).to(device)
val_input_tensor = torch.tensor(val_input).to(device)
val_target_tensor = torch.tensor(val_target).to(device)
test_tensor = torch.tensor(test_arr).to(device)
```

```
print(train_input_tensor.shape, val_input_tensor.shape, train_target_tensor.shape,
val_target_tensor.shape, test_tensor.shape)

train_dataset = torch.utils.data.TensorDataset(train_input_tensor, train_target_tensor)
val_dataset = torch.utils.data.TensorDataset(val_input_tensor, val_target_tensor)
train_loader = DataLoader(train_dataset, batch_size=32, shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=32, shuffle=False)


#model 1 width = [195, 100, 20, 1]

# Define model
model = KAN([195, 100, 20, 1])
model.to(device)
# Define optimizer
#optimizer = optim.SGD(model.parameters(), lr=1e-3, weight_decay=1e-4)
optimizer = optim.AdamW(model.parameters(), lr=1e-3, weight_decay=1e-4)
# Define learning rate scheduler
scheduler = optim.lr_scheduler.ExponentialLR(optimizer, gamma=0.8)


#model 2 width = [195, 250, 100, 25, 1]

# Define model
model2 = KAN ([195, 250, 100, 25, 1])
model2.to(device)
# Define optimizer
#optimizer = optim.SGD(model2.parameters(), lr=1e-3, weight_decay=1e-4)
optimizer = optim.AdamW(model2.parameters(), lr=1e-3, weight_decay=1e-4)
# Define learning rate scheduler
scheduler = optim.lr_scheduler.ExponentialLR(optimizer, gamma=0.8)


#model3 width =

# Define model
model3 = KAN ([195, 100, 20, 10, 5, 1])
model3.to(device)
# Define optimizer
#optimizer = optim.SGD(model3.parameters(), lr=1e-3, weight_decay=1e-4)
optimizer = optim.AdamW(model3.parameters(), lr=1e-3, weight_decay=1e-4)
# Define learning rate scheduler
scheduler = optim.lr_scheduler.ExponentialLR(optimizer, gamma=0.8)


#train the model
loss_func = nn.MSELoss()
train_loss_all = []
val_loss_all = []
losses = []
for epoch in range(30):
    # Train
    train_loss = 0
    train_num = 0
    model.train()
    with tqdm(train_loader) as pbar:
```

```
        running_loss = 0.0
        for i, (input, target) in enumerate(pbar):
            input = input.view(-1, 195).to(device)
            optimizer.zero_grad()
            output = model(input)
            loss = loss_func(output, target.to(device))
            optimizer.zero_grad()
            loss.backward()
            optimizer.step()
            train_loss += loss.item()*input.size(0)
            train_num += input.size(0)
            # print('Epoch %d loss: %.3f' % (epoch + 1, running_loss / len(trainloader)))
            pbar.set_postfix(lr=optimizer.param_groups[0]['lr'])
    train_loss_all.append(train_loss/train_num)
    pbar.set_postfix(loss=train_loss/train_num)

    # Validation
    model.eval()
    val_loss = 0
    val_accuracy = 0
    val_num = 0
    with torch.no_grad():
        for input, target in val_loader:
            input = input.view(-1, 195).to(device)
            output = model(input)
            val_loss += loss_func(output, target.to(device)).item()*input.size(0)
            val_num += input.size(0)
    val_loss_all.append(val_loss/val_num)
    # val_accuracy /= len(valloader)

    # Update learning rate
    scheduler.step()

    print(
        f"Epoch {epoch + 1}, Val Loss: {val_loss/val_num}"
    )


#loss curves
fig, ax = plt.subplots()
ax.plot(range(1, epoch + 2), train_loss_all, label='Train Loss')
ax.plot(range(1, epoch + 2), val_loss_all, label='Val Loss')
ax.set_title('Loss Curves')
ax.set_xlabel('Epochs')
ax.set_ylabel('Loss')
ax.legend()
plt.show()

Y_train_pred = model(train_input_tensor)
Y_train_pred = Y_train_pred.cpu().detach().numpy()
Y_cv_pred = model(val_input_tensor)
Y_cv_pred = Y_cv_pred.cpu().detach().numpy()

#roc curve
train_fpr4, train_tpr4, tr_thresholds4 = roc_curve(train_target,Y_train_pred)
cv_fpr4,cv_tpr4, cv_thresholds4 = roc_curve(val_target, Y_cv_pred)
```

```
plt.plot(train_fpr4,train_tpr4, label ="Training Data AUC :" + str(auc(train_fpr4,train_tpr4)))
plt.plot(cv_fpr4,cv_tpr4,label="CV Data AUC :" + str(auc(cv_fpr4,cv_tpr4)))
plt.legend()

plt.xlabel("FPR Values")
plt.ylabel("TPR Values")
plt.title('ROC Curve: KAN Model')

plt.grid(False)
plt.show()

output_kan = model(val_input_tensor)
output_kan_np = output_kan.cpu().detach().numpy()
print(output_kan_np)

print(roc_auc_score(val_target_tensor.cpu().detach().numpy(), output_kan_np))

precision = precision_score(val_target_tensor.cpu().detach().numpy(), output_kan_np,
average='binary')
recall = recall_score(val_target_tensor, output_kan_np, average='binary')
f1score = f1_score(val_target_tensor, output_kan_np, average='binary')

print(f"Precision = {precision}")
print(f"Recall = {recall}")
print(f"F1 Score = {f1score}")


start = datetime.now()

temp_test_data1 = test_data1.copy()

for col in temp_test_data1.columns:
   if col=='TARGET':
      temp_test_data1 = temp_test_data1.drop(['TARGET'],axis=1)
   if col =='SK_ID_CURR':
      temp_test_data1 = temp_test_data1.drop(['SK_ID_CURR'],axis=1)

kan_test_predict = model(test_tensor).detach()

print("Time taken to run this cell :", datetime.now() - start)

#save results for Kaggle competition
temp_test_data1['SK_ID_CURR'] = test_data['SK_ID_CURR']
temp_test_data1['TARGET'] = kan_test_predict.cpu()
temp_test_data1['SK_ID_CURR'] = temp_test_data1['SK_ID_CURR'].apply(lambda x:
np.int32(x))
temp_test_data1[['SK_ID_CURR', 'TARGET']].to_csv('kan_results.csv', index= False)
```