SZKOŁA GŁÓWNA HANDLOWA W WARSZAWIE
Kolegium Analiz Ekonomicznych

Data Science in Business

Analysis and Comparison of Credit Risk Assessment: Variable Selection and Explainability

Dominika Matusiak
Album no: 134260

Final Thesis written under the supervision of
Dr. Karol Przanowski

Warszawa 2024

Introduction....2
Transformation of Credit Risk Assessment....2
Objective of the Thesis....3
1. Data Preparation....3
1.1. Data Collection and Preprocessing....3
1.2. Data Conversion....4
1.3. Variable Binning....5
1.4. Calculation of Predictive Power Using the Gini Coefficient....6
1.4.1. Information Gain....7
1.4.2. Chi-square Test....8
1.4.3. Fisher Score....9
1.5. Results of Information Gain, Chi-square Test, and Fisher Score....9
2. RFE Models and Predictive Variable Selection....11
2.1. Introduction....11
2.2 Logistic Regression....12
2.3. Logistic Regression with L2 Penalty (Ridge)....15
2.4. Decision Tree....17
2.5. Las Losowy....21
2.6. XGBoost....24
2.7. Comparison of Model Evaluation Criteria....27
3. Analysis of Selected Variables....30
3.1 Comparison of Variable Analysis Using SHAP at a Global Level....31
3.2. Comparison of Variable Analysis Using SHAP and LIME at a Local Level....34
4. Discussion of Results....36
5. Summary....37
Business Implications....38
List of Figures....39
List of Tables....41
Literature....42

Introduction

Transformation of Credit Risk Assessment

Credit risk is one of the key challenges that banks and financial institutions must address. It is defined as the risk of financial loss incurred by the lender due to the borrower's failure to fulfill financial obligations. Due to the asymmetry of information between the lender and the borrower, the credit risk assessment process is complex and requires advanced tools and analytical techniques. Information asymmetry is particularly problematic in the case of new clients, where the bank does not have a credit history of the new client, making it difficult to assess their creditworthiness. This issue also arises when there is a change in the financial situation of a client who previously regularly repaid loans but may suddenly lose their job or encounter other financial difficulties, which the bank is not immediately informed about. Additionally, some clients may deliberately not disclose full information about their obligations or credit history, leading to an underestimation of risk. Credit assessment models are a key tool in managing credit risk, allowing for the prediction of the likelihood of borrowers defaulting on their obligations. This process includes the analysis of a range of demographic, financial, and behavioral characteristics of the borrower, enabling banks to better understand their clients and more accurately estimate the level of credit risk. Effective credit risk management is essential for the financial stability of lending institutions and for protecting the interests of all parties involved in the lending process. In recent years, technology has played a key role in transforming credit risk assessment methods. Traditional approaches, based on manual data analysis by experts, have given way to automated systems. The development of technology and the increase in the availability of large datasets (big data) have allowed for the creation of more precise and efficient predictive models, significantly improving the efficiency and accuracy of credit decisions. In a dynamic financial environment, it is crucial that these decisions are made quickly – within minutes or seconds. The automation of the credit assessment process and the use of modern technologies enable this by analyzing vast amounts of data and identifying key patterns and dependencies indicating potential insolvency risk. For this reason, it is worth focusing on identifying particularly important variables that significantly impact the predictive power of the model and reducing computational complexity, which can be costly. Effective variable selection is key to ensuring that predictive models are both accurate and computationally efficient.

Objective of the Thesis

The aim of this thesis is to find optimal estimators (models) used in the Recursive Feature Elimination (RFE) method for the analyzed dataset and to determine the point at which the quality

of the models no longer significantly improves. The analysis includes the evaluation of indicators such as the Gini coefficient and Type I and Type II errors, depending on the number and type of selected variables. After subjectively selecting the optimal variables, an analysis was conducted using SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) methods to understand the explainability of variables both globally (at the model level) and locally (at the level of individual observations). The thesis aims to provide useful insights into the effectiveness of different approaches in the context of credit risk assessment for the analyzed dataset and to describe the mechanisms of variable functioning in predictive models, which is a standard in the field of credit risk management. Local explainability was presented on the first observation from the test set. The above analysis aims to answer the following questions:

- Which models, considering the selected evaluation criteria, achieve the best results?
- How significant can the reduction of the number of variables be while maintaining a high Gini coefficient?
- How do different RFE estimators affect the explainability of models and observations from the ABT using SHAP and LIME analyses? In the following stages of the thesis, a data preparation pipeline will be presented, including binning and early filtration stages such as Gini coefficient analysis for individual variables and methods like Information Gain, Chi-test, and Fisher-score. The second chapter will focus on the theoretical description of variable selection methods and an attempt to answer the first two research questions.

1. Data Preparation

1.1. Data Collection and Preprocessing

The dataset selected for analysis is sourced from the Kaggle platform, titled "Development of Credit Risk Model & Scorecard". It contains information on over 420,000 consumer loans issued in 2015, encompassing 74 features, including the current loan status and various attributes related to borrowers and their payment behaviors. The dataset was chosen due to the large number of available variables, allowing for a more comprehensive analysis and feature selection. The table prepared for analysis stores basic information about each observation, such as a unique identifier, loan amount (loan_amnt), interest rate (int_rate), loan term (term), and the target variable. The target variable was created based on the 'loan_status' column, which indicates whether the client met their credit obligations. For analysis purposes, the statuses "Charged Off", "Default", "Late (31-120 days)", and "Does not meet the credit policy. Status: Charged Off" were assigned a value of 1, indicating default. All other statuses were assigned a value of 0, indicating no default. The table does not include clients (observations) who are in the process of repaying. After this operation, the 'loan_status' column was removed from the dataset to avoid redundancy. The target

variable is therefore a binary variable, where 0 indicates no default, and 1 indicates default. The data is organized in a single table that contains information about borrowers, their payment history, and current obligations. Other variables in the dataset include, among others, Interest Rate, Installment, Employment Title (Emp Title), Employment Length (Emp Length), Home Ownership status, Annual Income, Purpose of the loan, State of residence (Address State), Debt-to-Income Ratio, number of delinquencies in the last 2 years (Delinquency 2 Years), the earliest credit line (Earliest Credit Line), number of credit inquiries in the last 6 months (Inquiries Last 6 Months), total number of accounts (Total Accounts), total payment (Total Payment), total principal payment (Total Principal Payment), and total interest payment (Total Interest Payment).

1.2. Data Conversion

To ensure data consistency for analysis, a series of transformations were performed. The 'emp_length' column contained information in a text format, such as "10+ years" and "< 1 year". These values were converted to numbers by replacing the corresponding text with numerical values, and missing values were filled with zeros, indicating a lack of employment history. This conversion enables subsequent quantitative analyses, eliminating issues related to the diversity of data formats. This transformation was necessary to make the data homogeneous and ready for statistical analysis and modeling, as the diversity of text formats could lead to errors in calculations and analyses.

Date columns were also converted to datetime format, and the difference in months was calculated. Several date columns (such as 'earliest_cr_line', 'issue_d', 'last_pymnt_d', 'last_credit_pull_d') were converted to datetime format. Then, the difference between the loan issue date and the dates in these columns was calculated, creating new columns representing the number of months from the given date to the loan issue date. Negative values that could arise from this operation were replaced with the maximum value in the respective column to avoid errors in the analysis. Negative values may indicate erroneous data or data outside the observation period. These transformations were necessary to ensure temporal consistency of the data and enable accurate time-based analyses. Differences in dates are crucial for understanding the credit history and payment behaviors of clients.

These data transformations are essential for preparing a high-quality dataset that allows for precise modeling and analysis of credit risk. These steps help avoid issues related to data heterogeneity and ensure more accurate and reliable analysis results and predictive models.

1.3. Variable Binning

Initial binning stages - transforming input variables into a limited number of categories (bins) - were based on code proposed by Dr. Karol Przanowski. This code was included due to its

universality, correctness, and Dr. Przanowski's expertise. Key variable selection algorithms were proposed by the author (Chapters 2 and 3). The analytical table was divided into training, testing, and validation sets in proportions of 60%, 30%, and 10%. This division allows for proper model training, validation, and testing. Continuous and categorical variables were then extracted and binned, where the minimum category share was set at 1%. Binning of variables is typically used during the creation of scorecards, and in our case, to simplify modeling and primarily for simplified comparison of analysis results of used models. Binning continuous variables, such as age or income, allows for reducing their variance and focusing on important value ranges that are significant for the predictive model. For categorical variables, such as employment status or home ownership type, binning allows for grouping rare categories into meaningful clusters, increasing the model's stability. This process not only reduces dimensionality but also facilitates result interpretation. Each bin can be easily interpreted by business experts, which is particularly important in a regulatory context where transparency is required. Additionally, binning helps identify and eliminate insignificant variable values, which in turn improves the model's performance. The maximum number of bins in this work was set to 4 for simplification of analysis and more convenient interpretation. Tests were also conducted on groups containing 7 and 10 categories, but they were not included due to disproportionate distribution of variables, making consistency and interpretation of results difficult. Binning is not the main focus of this work, but a stage in the data preparation pipeline. The main goal is to compare RFE results for different models. However, experimenting with different numbers of bins could further contribute to understanding predictive dependencies. For continuous variables, a decision tree classifier based on the Gini criterion was used to determine interval points, detecting optimal thresholds that maximize the difference between groups. This process includes calculating the minimum share of variable values in bins and adjusting thresholds based on this value. Choosing the Gini criterion allows for effective differentiation of groups based on data variability. More details about the decision tree will be discussed in Chapter 2. For nominal variables, agglomerative clustering was used, grouping categories based on similarity, allowing for more meaningful grouping of nominal variables. This process includes analyzing category shares and the mean target variable in each category, then clustering using the agglomerative method. Based on established thresholds, appropriate bins are assigned to variable values in the dataset. In the next stage, the Information Value variable was created to assess the impact of explanatory variables on credit risk. Additionally, at this stage, a logistic regression model was created to check the predictive power of the model on the test and training sets. Due to the predominant number of target 0 variables, undersampling was performed, balancing the number of variables with target = 0 and target = 1 to 9894. This procedure, in subsequent analysis stages, minimally reduced Type I and Type II error values.

Variable	Condition	BR	Share	All	Bad	Good	Logit	GRP	Type	Bad share	Good share	Information Value
mths_since_last_pymnt_d	55.5 <= mths_since_last_pymnt_d < 57.5	0.3159666844	0.02233471598	5643	1783	3860	-0.7723698144	0	INT	0.3814719726	0.01556564589	1.168276454
mths_since_last_pymnt_d	57.5 <= mths_since_last_pymnt_d < 59.5	0.2895984036	0.01586742448	4009	1161	2848	-0.8973352398	1	INT	0.2483953787	0.01148470454	0.7263061973
mths_since_last_pymnt_d	59.5 <= mths_since_last_pymnt_d	0.2112418831	0.0195047812	4928	1041	3887	-1.317455792	2	INT	0.2227214377	0.01567452476	0.5482549337
mths_since_last_credit_pull_d	54.5 <= mths_since_last_credit_pull_d < 55.	0.1177027453	0.01254274587	3169	373	2796	-2.014366451	0	INT	0.07980316645	0.0112750119	0.1335885427
out_prncp	out_prncp < 15.245	0.1069897363	0.06092869356	15394	1647	13747	-2.121865114	0	INT	0.3523748395	0.05543547516	0.5487312465
out_prncp_inv	out_prncp_inv < 15.245	0.1069897363	0.06092869356	15394	1647	13747	-2.121865114	0	INT	0.3523748395	0.05543547516	0.5487312465
next_pymnt_d	<OTHERS>	0.1068573286	0.060893072	15385	1644	13741	-2.123251714	0	NOM	0.351732991	0.05541127985	0.5471788903
int_rate	21.575 <= int_rate	0.08	0.03186150339	8050	644	7406	-2.442346894	0	INT	0.1377834831	0.02986507085	0.1647241397
sub_grade	<OTHERS>	0.07569386039	0.02823602052	7134	540	6594	-2.502346114	0	NOM	0.1155327343	0.02659063964	0.1303985525

Table 1
Table Fragment with Gini Coefficients for Training and Test Sets

1.4. Calculation of Predictive Power Using the Gini Coefficient

In the next stage, the individual predictive power of variables was assessed using the Gini coefficient, which is a measure of distribution inequality and evaluates the predictive power of a variable. The Gini coefficient can be calculated as:

$$Gini = 2AUC - 1 \quad (1.1)$$

Where:
AUC is the area under the ROC (Receiver Operating Characteristic) curve.

$$Gini = \frac{a_p}{a_r} \quad (1.2)$$

Where:
a_p = This represents the area between the ROC curve and the line of no-discrimination, a_r = This represents the total area under the ROC curve.

Alternatively, the Gini coefficient can be calculated using the difference between the proportion of concordant pairs (P_c) and the proportion of discordant pairs (P_d):

$$Gini = P_c - P_d \quad (1.3)$$

Where:

P_c = proportion of concordant pairs: cases where the predicted risk value for a bad loan is greater than for a good loan ($P_i > P_j$).

P_d = proportion of discordant pairs: cases where the predicted risk value for a bad loan is less than for a good loan ($P_i < P_j$).

A higher Gini value indicates that the variable is more valuable in differentiating credit risk. For preselection, a minimum Gini threshold of 1% was established. To avoid the dominance of single variables in feature selection, variables with a Gini value above 40% were removed. The author acknowledges potential quality reduction in the model, noting that these variables often dominated analyses, limiting the diversity of selected variables for sets containing fewer than 7 variables.

Variable	Gini train	Gini test	R. Gini	Information Value	Missing percent	Number of distinct	Mode	P. mode	Type
out_prncp	0.3091197745	0.2988972577	0.0330697601	0.6731226245	0	82878	0	0.0608851561	INT
total_rec_prncp	0.1994829176	0.1976282612	0.009297319391	0.1615094828	0	76152	0	0.04202552087	INT
initial_list_status	0.1800368691	0.1721503997	0.04380474606	0.1323481736	0	2	w	0.6347721804	NOM
total_pymnt	0.1301639215	0.1436318649	0.103469097	0.0997046223	0	119636	0	0.04187907669	INT
total_rec_int	0.1240951371	0.1412931351	0.138587203	0.05244020419	0	96671	0	0.04223529226	INT
inq_last_6mths	0.1272061042	0.1356336305	0.06625095815	0.06423228068	0	7	0	0.6094452536	INT
total_rev_hi_lim	0.1197779154	0.1235338446	0.03135744373	0.05079073695	0	7650	13000	0.003285099107	INT
home_ownership	0.1028415465	0.1173797694	0.1413652692	0.03884804017	0	4	MORTGAGE	0.4931764929	NOM
title	0.1147910305	0.1134975776	0.01126789158	0.05670581602	0.0003443417136	22	Debt consolidati	0.5940436079	NOM

Table 2
Table Fragment with Gini Coefficients for Training and Test Sets

1.4. Additional Filter Methods

The next step in the preselection process was the removal of highly collinear variables (above 90%) to reduce potential multicollinearity. High collinearity between variables can lead to issues with model interpretation and unstable parameter estimates, making it essential to minimize. This process involved iteratively removing one variable from each pair of highly collinear variables. For each pair of variables with a correlation coefficient exceeding 90%, the variable that had more pairs of high collinearity with other variables was removed. If both variables had the same number of highly collinear pairs, the first variable was removed. As a result of this process, 31 variables were selected, which exhibited low collinearity, allowing the model to avoid excessive redundancy. This preselection stage ensured that the selected variables were unique in the information they contributed to the model, thus improving its performance and reliability in predicting credit risk.

There are many other methods for filtering variables, three of which are discussed below. The drawback of these methods is the lack of a relationship between the feature selection process and the performance of classification algorithms. Therefore, these methods will not be applied restrictively but will serve to familiarize with the variables and remove only those that perform poorly in all filtering stages, i.e., Information Gain, Chi-square Test, and Fisher-Score.

1.4.1. Information Gain

Information Gain (IG) is a measure of the amount of information that a given feature provides about the target variable. It is calculated as the difference between the entropy before and after splitting the dataset based on the given feature. Entropy is a measure of uncertainty or randomness in the dataset, and a reduction in entropy indicates an increase in information. IG is defined as:

$$IG(Y, X) = H(Y) - H(Y|X) \tag{2.1}$$

Where:

$H(Y)$ = entropy of the target variable Y,

$H(Y|X)$ = conditional entropy of Y given the value of feature X.

The entropy H of variable Y is defined as jest:

$$H(Y) = -\sum_{i=1}^{n} P(y_i) log P(y_i) \quad (2.2)$$

The conditional entropy H(Y|X) is calculated as:

$$H(Y|X) = -\sum_{x \in X} P(x) \sum_{x \in X} P(y|x) log P(y|x) \quad (2.3)$$

Where:

$P(y_i)$ = probability of occurrence of class y_i, P(y|x) = conditional probability of occurrence of class y given the value of feature x.

1.4.2. Chi-square Test

The Chi-square Test is a statistical method used to assess the relationship between two categorical variables. In the context of feature selection, the Chi-square Test evaluates how strongly a given feature is associated with the target variable. The Chi-square Test compares observed frequencies with expected frequencies under the assumption of independence and is defined as:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \quad (3)$$

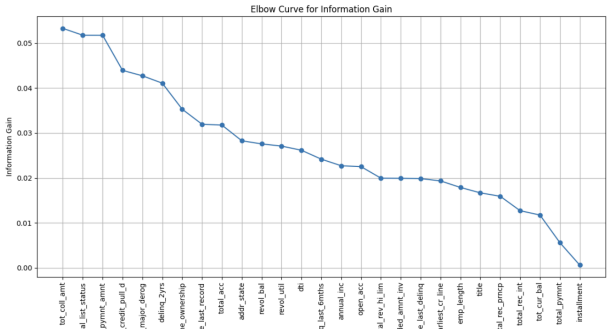
Where:
O_i = observed frequency,
E_i = expected frequency.

1.4.3. Fisher Score

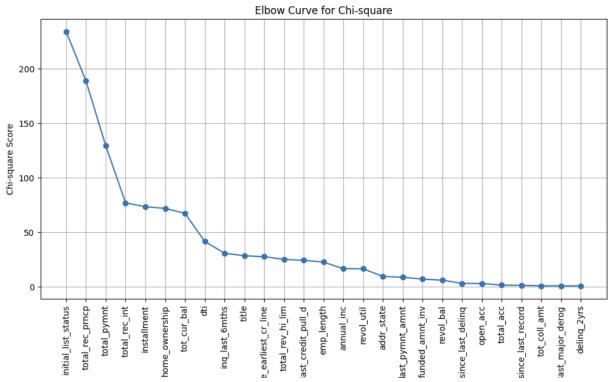
The Fisher Score evaluates variables based on their ability to distinguish classes, measuring the ratio of inter-class variance to intra-class variance for each variable. The higher the Fisher Score, the better a given feature distinguishes between the classes.

$$F_i = \sum \frac{Variance\ between\ classes}{Variance\ within\ classes} \quad (4)$$

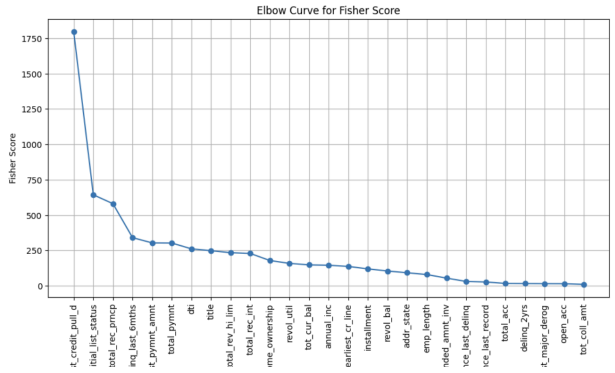
1.5. Results of Information Gain, Chi-square Test, and Fisher Score



Higher IG values indicate that the feature provides more information. Figure 1a shows that IG values decrease after the first three variables. The plot does not have a clear breaking point but shows a local flattening of the curve at the level of 0.02.



Higher Chi-square values indicate a stronger relationship between the feature and the target variable. The Chi-square results for individual variables are shown in Figure 1b, which indicates a global breaking point at the fourth variable and a flattening after the eighth variable.



The F-value plot for individual variables is shown in Figure 1c. The plot highlights the first three variables, followed by a stable flattening from the fourth variable onward, indicating a decreasing informativeness of these variables.

Graph 1
Elbow Curves: 1a - Information Gain, 1b - Chi-square, 1c - Fisher Score

The comparison of the results of the Information Gain, Chi-square Test, and Fisher Score methods allowed for the identification of key variables before feature selection. Variables that obtained high values in all three methods, such as **initial_list_status** and **mths_since_last_credit_pull_d**,

were considered particularly important for the credit risk assessment model. Variables with low values in all three methods, such as **tot_coll_amt**, **open_acc**, and **delinq_2yrs**, were deemed less useful and considered for removal. The results of the analyses are presented in Table 3. Variables with low values in all three measures were removed.

Information Gain		Chi-square test		F-Score	
Feature	**Results**	**Feature**	**Results**	**Feature**	**Results**
tot_coll_amt	0.053267	initial_list_status	234.252518	mths_since_last_credit_pull_d	1796.900702
initial_list_status	0.051745	total_rec_prncp	188.862672	initial_list_status	643.014658
last_pymnt_amnt	0.051731	total_pymnt	129.305988	total_rec_prncp	580.180151
mths_since_last_credit_pull_d	0.043927	total_rec_int	76.884505	inq_last_6mths	340.149671
mths_since_last_major_derog	0.042723	installment	73.345697	last_pymnt_amnt	303.068757
delinq_2yrs	0.041085	home_ownership	71.812308	total_pymnt	302.205358
home_ownership	0.035301	tot_cur_bal	67.287694	dti	260.985800
mths_since_last_record	0.031948	dti	41.526415	title	247.957323
total_acc	0.031762	inq_last_6mths	30.608623	total_rev_hi_lim	233.777470
addr_state	0.028272	title	28.293920	total_rec_int	228.542678
revol_bal	0.027572	mths_since_earliest_cr_line	27.470763	home_ownership	178.402264
revol_util	0.027089	total_rev_hi_lim	25.024607	revol_util	158.307989
dti	0.026163	mths_since_last_credit_pull_d	24.106512	tot_cur_bal	147.676827
inq_last_6mths	0.024183	emp_length	22.466025	annual_inc	145.125858
annual_inc	0.022720	annual_inc	16.481159	mths_since_earliest_cr_line	136.715964
open_acc	0.022513	revol_util	16.295863	installment	119.305995
total_rev_hi_lim	0.019954	addr_state	9.337571	revol_bal	104.399861
funded_amnt_inv	0.019922	last_pymnt_amnt	8.548848	addr_state	91.891834
mths_since_last_delinq	0.019859	funded_amnt_inv	6.861281	emp_length	79.291794
mths_since_earliest_cr_line	0.019341	revol_bal	5.897921	funded_amnt_inv	54.234579
emp_length	0.017889	mths_since_last_delinq	2.904720	mths_since_last_delinq	30.806569
title	0.016690	open_acc	2.844259	mths_since_last_record	26.916898
total_rec_prncp	0.015929	total_acc	1.319860	total_acc	16.642737
total_rec_int	0.012724	mths_since_last_record	1.138847	delinq_2yrs	15.898100
tot_cur_bal	0.011754	tot_coll_amt	0.681908	mths_since_last_major_derog	15.225210
total_pymnt	0.005623	mths_since_last_major_derog	0.638046	open_acc	14.760494

Information Gain		Chi-square test		F-Score	
installment	0.00062	delinq_2yrs	0.593575	tot_coll_amt	10.082580

Table 3
Comparison of Variable Importance across Information Gain, Chi-square Test, and Fisher Score

2. RFE Models and Predictive Variable Selection

2.1. Introduction

In this chapter, the theoretical foundations and results for metrics such as the Gini coefficient, Type I and Type II errors, and the Kolmogorov-Smirnov (KS) statistic are presented to address the first research question: which models, considering the selected evaluation criteria, achieve the best results?

As discussed in Chapter 1, after the variable filtering stage, approximately 24 significant features were extracted, with variables having a Gini coefficient below 40% excluded. Despite the relatively small number of variables, it was possible to identify the most important features for individual models, which is presented in this chapter. In real-world applications where models encompass thousands of variables, feature selection is even more crucial to reduce computation time.

This chapter focuses on the theoretical presentation of the applied models and experimentation with their hyperparameters to find those that achieve the best results according to the selected criteria. After identifying the best models from each group (logistic regression, logistic regression with L2 penalty, random forest, decision tree, XGBoost), the optimal number of features for which the metrics stabilize or do not significantly improve was determined.

The goal of feature selection in this work was to present the proposed approach rather than to find one universal method. The suggested number of variables may vary depending on specific datasets and tables, known as Analytical Base Tables (ABT).

Feature Selection Process

- Logistic Regression and Ridge Logistic Regression

Variables were removed based on model coefficients. For logistic regression, features with the smallest beta coefficients were removed, while for ridge logistic regression, an additional L2 penalty was applied.

- Tree-based Models (Decision Tree and Random Forest)

  Feature importance was assessed based on splitting criteria. Features with the least impact on tree splits were removed in subsequent iterations.

- XGBoost

  Feature importance was based on information gain. Features that contributed the least to information gain were eliminated in each iteration.

This process was repeated until only one variable remained. For each number of selected variables, the chosen criteria were discussed. Based on this, conclusions were drawn regarding the optimal model and the optimal number of features that should be used for credit risk prediction.

This analysis provided insights into which features are most important for our models and which feature selection approach is most effective in the context of credit risk prediction. The analysis will be conducted on various model parameters. Chapter 3 compares variables using SHAP values.

2.2 Logistic Regression

The first analysis in our study involves applying logistic regression as the estimator in the RFE method. Logistic regression is a statistical model used to predict the probability of belonging to one of two classes, based on the logistic function. This model can be described by the following equation:

$$P(y = 1|X) = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_n x_n)}} \quad (5)$$

Where:

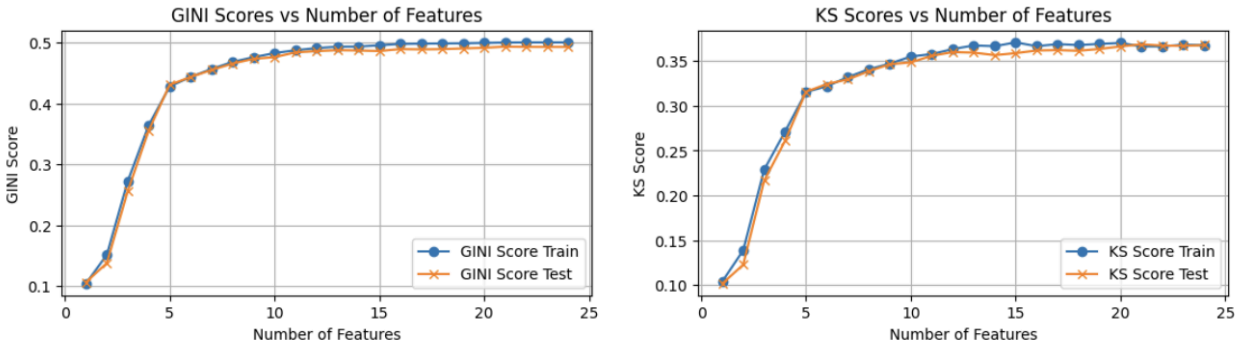
$P(y=1 \mid X)$ = probability of belonging to class 1 for a given feature vector X,
β_0 = intercept,
$\beta_1, \beta_2, \ldots, \beta_n$ = regression coefficients for variables $x_1, x_2, \ldots, x_n$.

Logistic regression is valued for its simplicity, interpretability, and ability to easily compute probabilities.

Figure 2 presents the Gini coefficient and Kolmogorov-Smirnov (KS) statistic plots based on the number of selected variables for training and test sets. The KS statistic assesses the model's ability to separate the two classes by measuring the maximum difference between the cumulative probability distributions of these classes.

On the first plot, there is a rapid increase in the Gini coefficient initially, stabilizing after about five features. The same pattern is observed with the KS Score, indicating that adding more features only brings marginal benefits. The results for the training and test sets are similar for both criteria, indicating good model generalization.

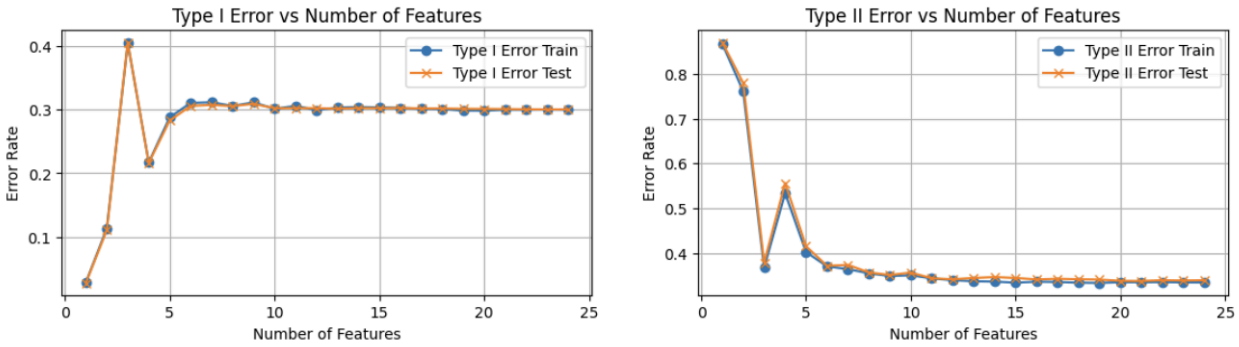
For five features, the Gini coefficient is approximately 0.43 (moderate predictive power), and for ten features, it increases to 0.48 (moderately high predictive power). For five features, the KS Score is around 0.32 (moderate class separation ability), and for ten features, it rises to 0.35 (moderately high class separation ability).



Graph 2
Wyniki współczynnika Giniego (a) oraz statystyki KS (b) dla modelu RFE z estymatorem Regresji Logistycznej.

Figure 3 presents the plots of Type I and Type II errors based on the number of selected variables for the training and test sets. Type I errors, also known as false positives, occur when the model incorrectly classifies customers who did not default (true class is 0) as those who did default (prediction is 1). The plot shown in Figure 3 indicates that initially, when the number of features is small, the number of Type I errors is high. A surprising spike is visible at three variables, which requires further analysis. After selecting around five features, the number of Type I errors stabilizes, indicating an improvement in model accuracy. For five features, the Type I error is approximately 28.3% for the test set, which represents a moderate level of false positives. For ten features, the Type I error is around 30.2% for the test set, showing that adding more variables does not improve the outcome. Type II errors, also known as false negatives, occur when the model incorrectly classifies customers who did default (true class is 1) as those who did not default (prediction is 0). The plot shows that the number of Type II errors quickly decreases as the number of features increases to about five, then stabilizes. The spike at four variables corresponds to the

spike observed with Type I errors. Stabilization occurs after selecting about five to six features. For five features, the Type II error is approximately 41.5% for the test set, indicating a moderate level of false negatives. For ten features, the Type II error decreases to about 35.6% for the test set, which is a moderately low level of false negatives. The values of Type I and Type II errors stabilize after considering around five features, suggesting that further addition of features brings marginal benefits. The results for the training and test sets are similar, indicating good model generalization. For five features, the Type I error is approximately 28.3%, representing a moderate level of false positives, while the Type II error is around 41.5%, representing a moderate level of false negatives. For ten features, the Type I error is approximately 30.2%, which also represents a moderate level of false positives, and the Type II error is around 35.6%, which is a moderately low level of false negatives.



Graph 3
Results of Type I Error and Type II Error for the RFE Model with Logistic Regression Estimator

Number of Features	GINI Score Train	GINI Score Test	Type I Error Train	Type I Error Test	Type II Error Train	Type II Error Test	KS Score Train	KS Score Test
5	0.427818	0.430582	0.287762	0.282655	0.402225	0.415141	0.315576	0.315847
10	0.482739	0.475999	0.301027	0.301941	0.350449	0.356338	0.355156	0.348666

Table 4
Evaluation Criteria for RFE with Logistic Regression Estimator for 5 and 10 Variables

2.3. Logistic Regression with L2 Penalty (Ridge)

This chapter focuses on the application of logistic regression with L2 penalty. An analysis was conducted for different values of the hyperparameter C to determine which parameters could lead to the highest Gini coefficient results. Ridge Regression in the context of logistic regression is a

method that adds an L2 penalty to the cost function to promote more stable models by reducing variance. The cost function in logistic regression with L2 penalty is defined as:

$$J(\beta) = -\frac{1}{n}\sum_{i=1}^{n}(y_i\ log(y\hat{}_i + (1 - y_i)log(1 - y\hat{}_i) + \lambda \sum_{j=1}^{p} \beta_j^2 \quad (10)$$

Where:

y_i = target value for the i-th observation,

$y\hat{}_i$ = predicted probability,

p = number of features,

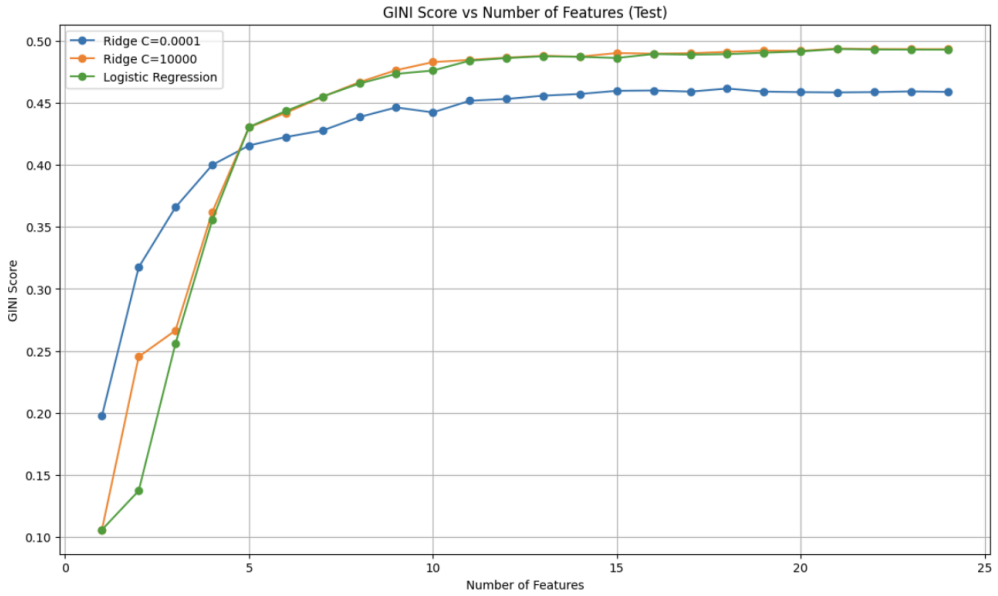
β_j = regression coefficient,

n = number of samples,

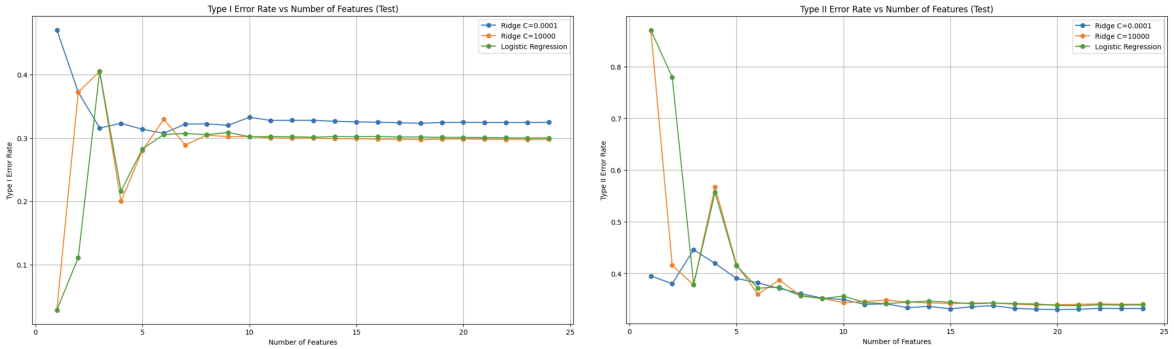
λ = hyperparameter controlling the size of the L2 penalty.

The regularization parameter λ in the Ridge model controls the strength of the penalty; at λ=0 model redukuje się do klasycznej regresji liniowej. the model reduces to classical linear regression. As α, increases, the coefficients are shrunk, simplifying the model and reducing variance. Note that the hyperparameter C is the inverse of λ ($C = \frac{1}{\lambda}$), meaning that higher values of C correspond to less L2 regularization. In the analysis, Ridge regression was used instead of Lasso regression to avoid uncontrolled variable elimination. Instead, RFE with the Ridge Regression estimator ranks variables and eliminates those with the smallest coefficients, allowing for more stable modeling.

Figure 4 presents the Gini coefficient results for hyperparameter values C=0.0001C=0.0001C=0.0001, C=1000C=1000C=1000 in the Ridge regression model, and for classical logistic regression. For higher values of C (less L2 penalty), the model approaches classical logistic regression without L2 penalty, which is visible from the fifth variable onward. Results from five variables show that the Gini coefficient for the Ridge model with C=0.0001C=0.0001C=0.0001 is 0.407, while for models with C=1000C=1000C=1000 and classical logistic regression, the Gini coefficient is around 0.42, indicating better fit with higher values of C. Results for ten features show a similar trend, where the Gini coefficient for the Ridge model with C=0.0001C=0.0001C=0.0001 is 0.415, and for models with C=1000C=1000C=1000 and classical logistic regression, it is around 0.48. However, for fewer than five variables, the penalty with coefficient C=0.0001C=0.0001C=0.0001 models better, as observed in **Figure 4** (higher Gini) and **Figure 5** (lower proportion of false negative and false positive errors to all variables). For the dataset under consideration, a higher C coefficient (i.e., less L2 penalty) increases the Gini coefficient for the test set. Since the further analysis focuses on five and ten variables, and logistic regression without L2 penalty achieves higher predictive values (Gini), it was decided to omit the RFE estimator with regularization.



Graph 4
Gini results for the test set for the RFE model with Logistic Regression estimator with L2 penalty for the test set.



Graph 5
Results of Type I and Type II errors for the RFE model with Logistic Regression estimator with L2 penalty for the training and test sets.

Number of Features	GINI Score Train	GINI Score Test	Type I Error Train	Type I Error Test	Type II Error Train	Type II Error Test	KS Score Train	KS Score Test
C = 0.0001								
1	0.182726	0.197628	0.484168	0.47071	0.385537	0.39507	0.140565	0.142879
2	0.297346	0.317435	0.378049	0.372514	0.380616	0.380282	0.241335	0.247204
3	0.351256	0.365817	0.32991	0.315864	0.451433	0.445775	0.249465	0.261155
4	0.389151	0.399926	0.337184	0.323064	0.418485	0.419366	0.286906	0.289403
C = 1000								
1	0.103664	0.105961	0.028883	0.028147	0.867351	0.870423	0.103766	0.10143
2	0.250711	0.2454	0.368635	0.37238	0.411425	0.415493	0.21994	0.212127
3	0.281289	0.266578	0.403937	0.4053	0.367137	0.378169	0.237698	0.228326
4	0.371106	0.361942	0.200257	0.199867	0.547497	0.567254	0.277279	0.266615
C=0								
1	0.103664	0.105961	0.028883	0.028147	0.867351	0.870423	0.103766	0.10143
2	0.151111	0.137429	0.113607	0.110942	0.761232	0.779577	0.138639	0.122576
3	0.272285	0.25596	0.403937	0.405219	0.367351	0.378521	0.228712	0.216444
4	0.364328	0.355552	0.216945	0.216324	0.53359	0.556338	0.271502	0.261074

Table 5
Evaluation Criteria for RFE with Ridge Logistic Regression Estimator for up to 4 variables.

2.4. Decision Tree

A Decision Tree is a machine learning algorithm that creates a predictive model based on a tree structure. Each node of the tree represents a decision made based on the value of a feature, and each leaf of the tree represents the prediction outcome. A Decision Tree minimizes the cost function defined as the sum of classification errors for all nodes:

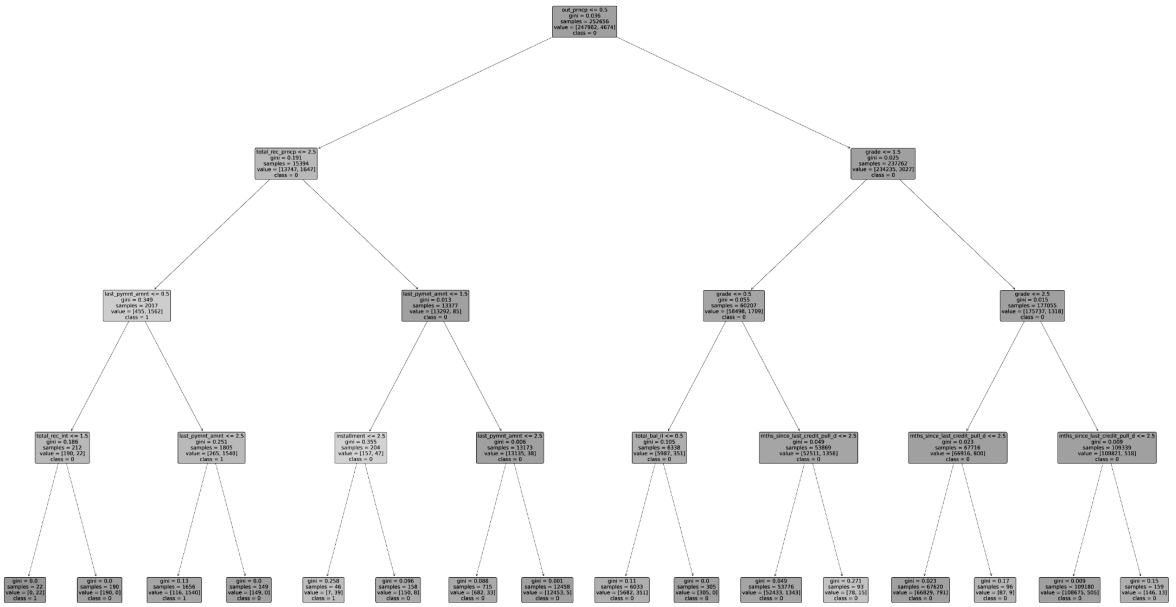
$$J = \sum_{i=1}^{n} (y_i - \hat{y}i)^2 \quad (6)$$

Where:

y_i = target value for the i-th observation,

$\hat{y}i$ = predicted value for the i-th observation,

n = number of observations.



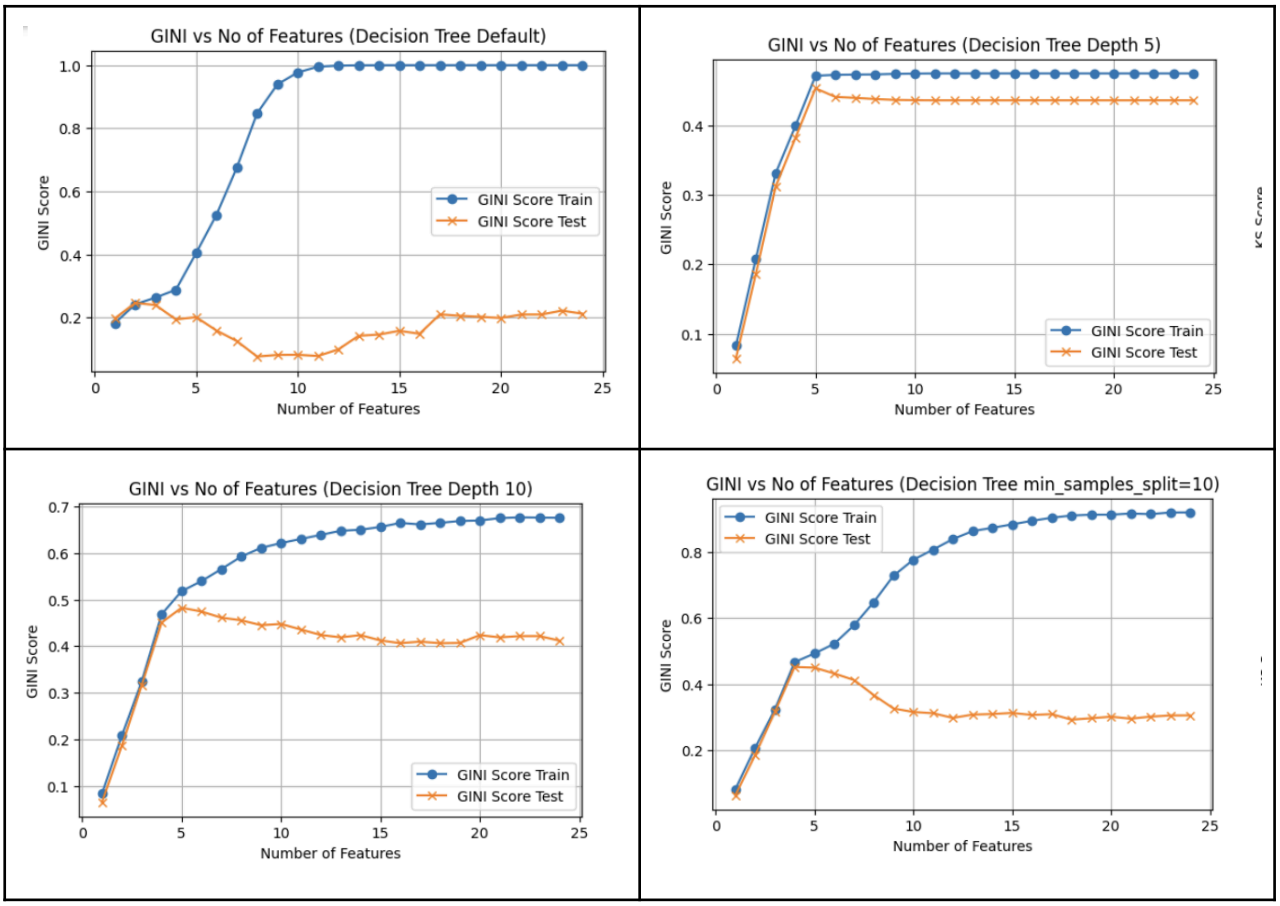
Graph 6
Decision Tree Plot with Default Hyperparameters Pruned to 4 Nodes

In the following analysis, the Gini coefficient results are presented based on the number of selected variables for different settings of decision tree hyperparameters. The configurations included in the plots (Table 5) are as follows:

- Decision Tree Default
- Decision Tree Depth 5
- Decision Tree Depth 10
- Decision Tree min_samples_split=10
- Decision Tree min_samples_leaf=5
- Decision Tree max_features=lo

By comparing the Gini coefficients for the test and training sets (Table 6), it can be observed that up to 4-5 variables, the training and test models achieve similar results, indicating that the more variables, the more complex the model becomes. Additionally, models with smaller tree depth achieve the best balance between fit and generalization. For example, the model with tree depth 5 (Decision Tree Depth 5) achieves a Gini coefficient of about 0.5 for the training set and about 0.45 for the test set, indicating good generalization. In the model with tree depth 10 (Decision Tree Depth 10), the Gini coefficient for the training set increases to 0.7, and for the test set, it achieves results similar to the model with depth 5, and even higher up to the tenth variable.

The model with default settings (Decision Tree Default) and the model with max_features=log2 achieve a Gini coefficient close to 1 for the training set, but for the test set, the Gini coefficient reaches a maximum value of about 0.25, indicating clear overfitting and highlighting the importance of testing and validating models. Similarly, models with min_samples_split=10 and min_samples_leaf=5 achieve a Gini coefficient of about 0.8 for the training set and about 0.4 for the test set, indicating moderate overfitting. Analyzing the results for the Decision Tree Depth 5 and Depth 10 models on the test set (Graph 5), we notice that for the model with tree depth 5 (Decision Tree Depth 5), the Gini coefficient for the test set with 5 features is 0.452, and with 10 features, it drops to 0.435942. Despite a larger delta between the Gini coefficients for the training and test sets, this model shows better performance for a smaller number of variables. For the Decision Tree Depth 10 model, the Gini coefficient for the test set with 5 features is 0.474445, and with 10 features, it drops to 0.43577.



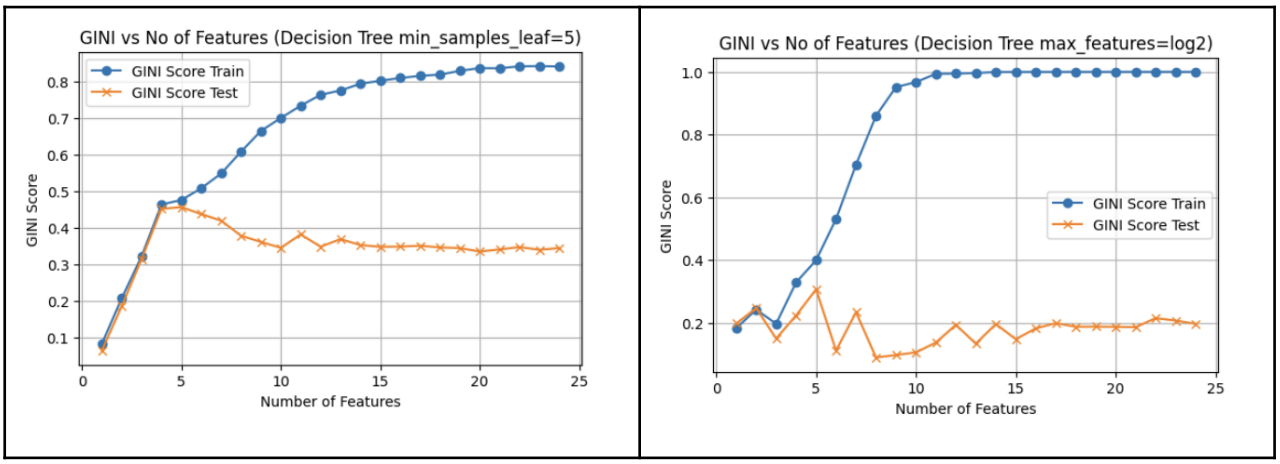
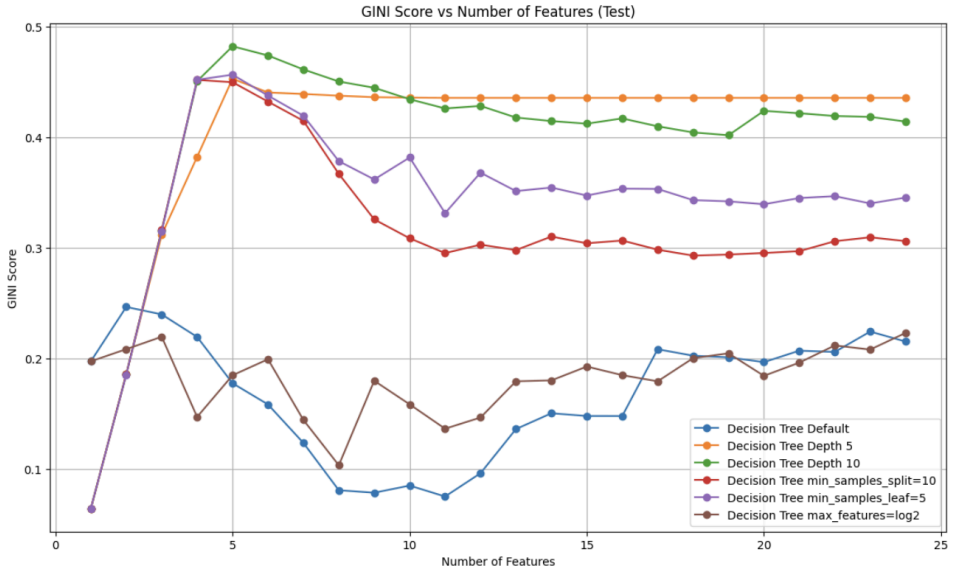


Table 6

Results of Gini Coefficient for Training and Test Sets for RFE Model with Decision Tree Estimators



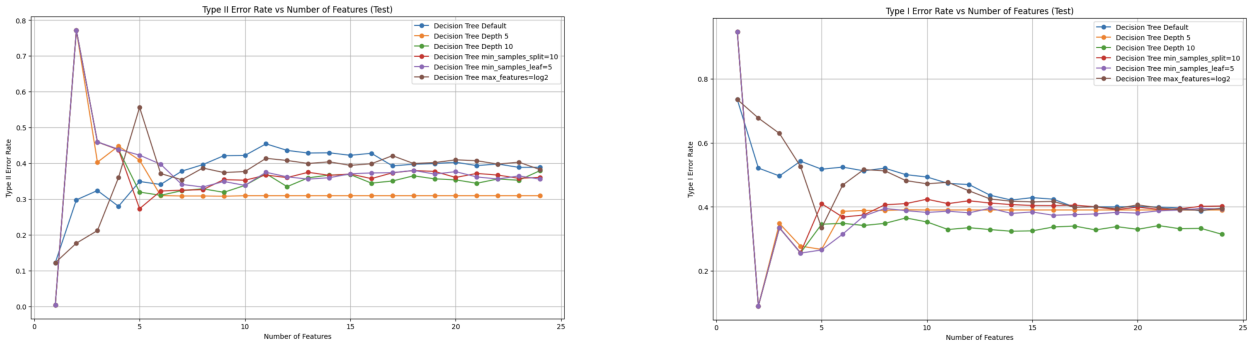
Graph 7

Gini Results for the Test Set for the RFE Model with Decision Tree Estimator for the Test Set

Figure 6 presents the plots of Type I and Type II errors for different decision tree models, calculated as the proportions of false classifications relative to the total number of observations. The Decision Tree Default and max_features=log2 models show higher Type I errors, stabilizing at around 0.4, indicating overfitting. The Decision Tree Depth 5 model stabilizes Type I errors at around 0.3, which is a better result compared to the Decision Tree Depth 10, which stabilizes at

around 0.35. Models with min_samples_split=10 and min_samples_leaf=5 show similar Type I error values, stabilizing around 0.35-0.4.

In the case of Type II errors, the Decision Tree Depth 5 and Depth 10 models stabilize at around 0.3, which is more favorable than the Decision Tree Default and max_features=log2 models, which stabilize at around 0.35. Models with min_samples_split=10 and min_samples_leaf=5 stabilize around 0.3-0.35. The Decision Tree Depth 5 model achieves the lowest Type I and Type II error values, indicating its best generalization ability compared to other models. Additionally, as described above, this model achieves high results in the Gini analysis on the test set and is the most stable with an increasing number of variables, thus it was chosen for further analysis.



Graph 8
The results of Type I and Type II errors for the RFE model with a Decision Tree estimator for the training and test sets.

2.5. Las Losowy

Random Forest is an algorithm based on decision trees that creates a set of multiple decision trees to improve prediction accuracy. A key element of Random Forest is the use of the bagging (bootstrap aggregating) method, which randomly selects data samples for training each tree and then aggregates their results. The cost function in Random Forest considers the average prediction errors from all trees in the forest:

$$J = \frac{1}{m}\sum_{k=1}^{m}\sum_{i=1}^{n}(y_i - \hat{y}ik)^2 \qquad (7)$$

Where:

m = number of trees in the forest,

$\hat{y}$ik = predicted value for the iii-th observation by the k-th tree

The algorithm effectively reduces the risk of overfitting by averaging the results of multiple decision trees, which increases the stability and accuracy of predictions. Random Forest handles a large number of variables well and is less susceptible to multicollinearity than a single decision tree. The trade-off for high accuracy is the loss of interpretability at the level of an individual tree, although visualizing individual trees can help better understand the model's workings.

Table 7 and Figure 7 present the Gini index depending on the number of features for different Random Forest model configurations. The Random Forest models with Depth 10 and min_samples_leaf=3 achieve the highest Gini indices for the test set, stabilizing at values around 0.5. Despite higher Gini indices for the training set, which are about 0.7 for Depth 10 and above 0.9 for min_samples_leaf=3, the Depth 10 model shows a better balance between fit and generalization.

The Random Forest models with n_estimators=200 and min_samples_split=5 show clear overfitting, with a Gini index close to 1 for the training set and about 0.4 for the test set, indicating a lower ability to generalize compared to the Depth 10 model. In the case of the Random Forest models with n_estimators=200 and min_samples_split=5, the differences between the Gini indices for the training and test sets are larger than in the Depth 10 model, indicating greater overfitting in these models.

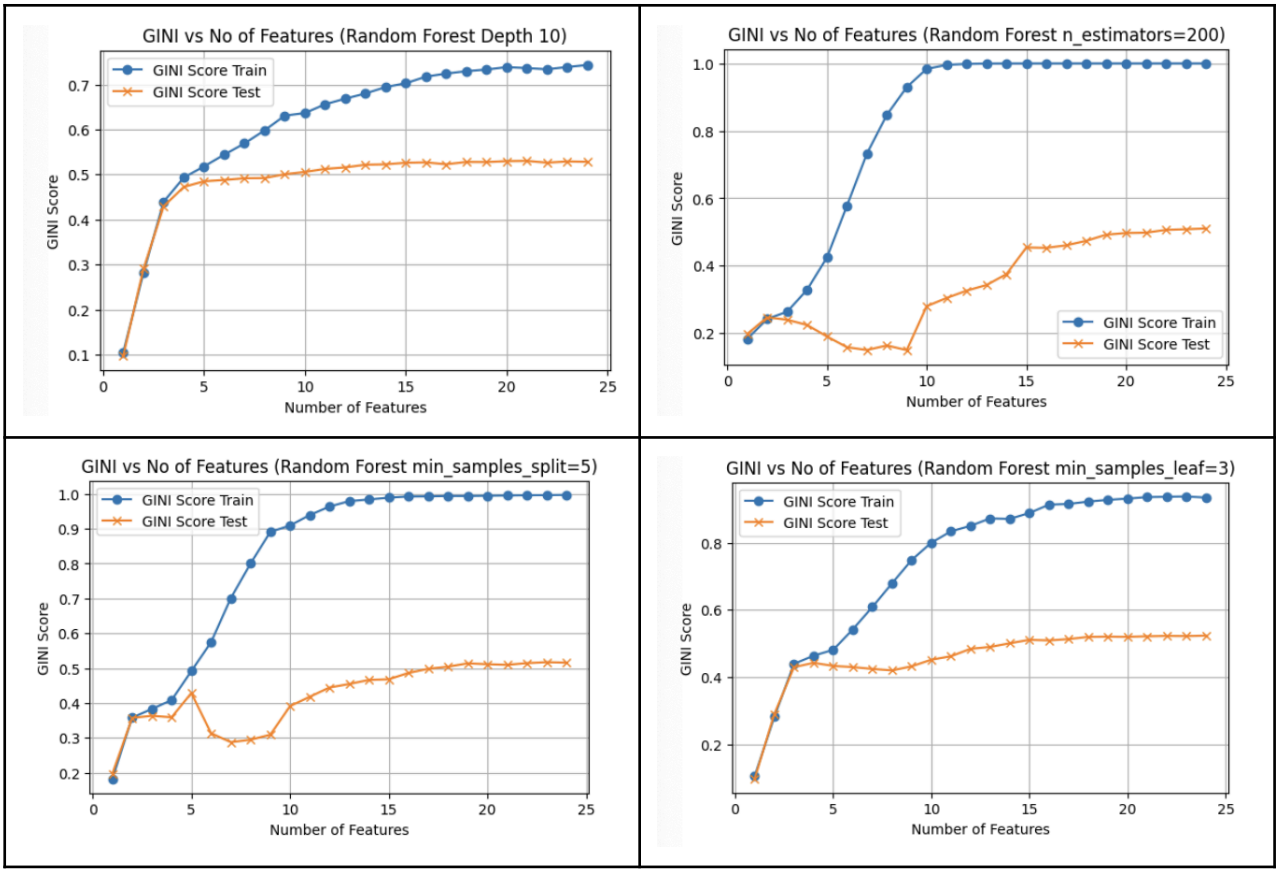
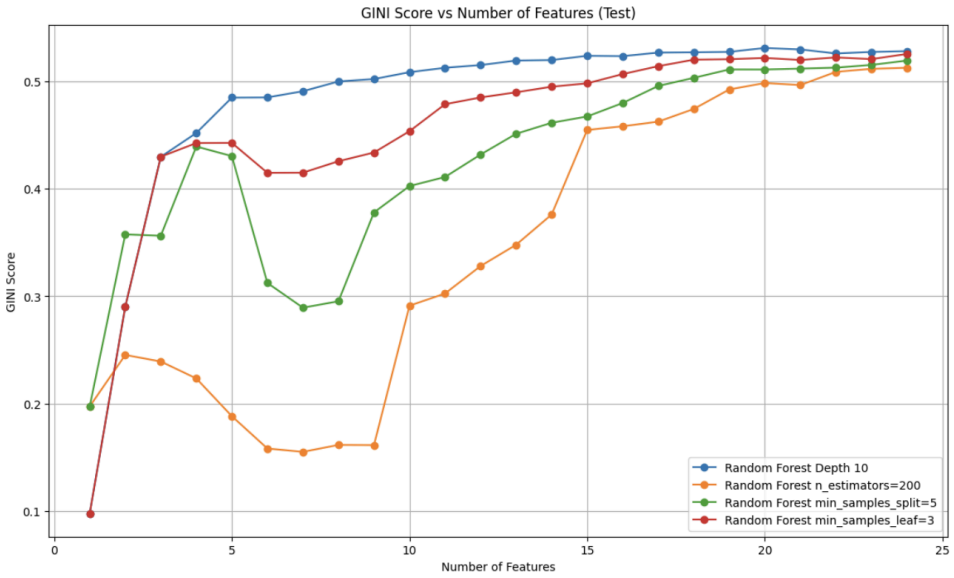
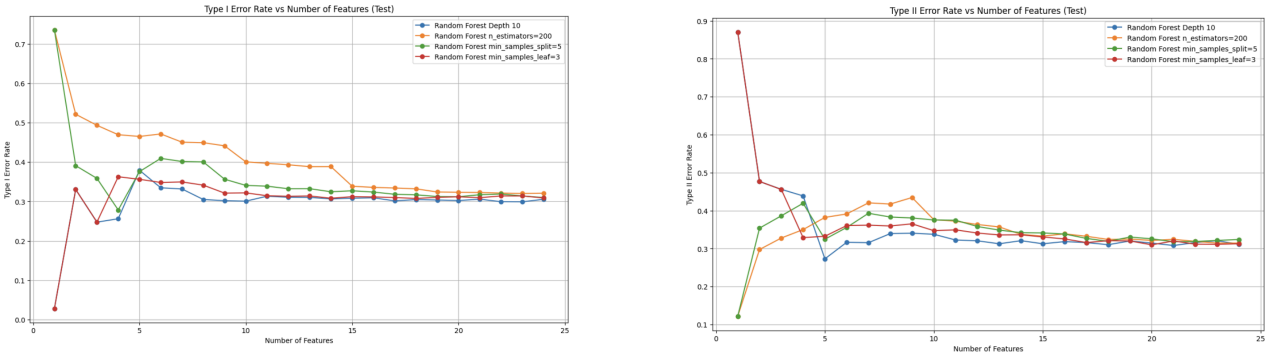


Table 7
The results of the Gini coefficient for the training and test sets for the RFE model with Random Forest estimators.

In Figure 8, in the analysis of Type I and Type II errors, the Random Forest model with Depth 10 again demonstrates the best performance, achieving the lowest values of Type I and Type II errors, which stabilize at approximately 0.3 for both Type I and Type II errors. In comparison, the Random Forest models with n_estimators=200 and min_samples_split=5 have Type I and Type II errors stabilizing at around 0.35-0.4, indicating higher error values compared to the Depth 10 model.



Graph 9
The Gini results for the test set for the RFE model with a Random Forest estimator for the test set.



Graph 10
The results of Type I and Type II errors for the RFE model with a Random Forest estimator for the training and test sets.

2.6. XGBoost

XGBoost (Extreme Gradient Boosting) is an extension of the Gradient Boosting method, which builds sequential models where each new model attempts to correct the errors made by the previous ones. The cost function in XGBoost is defined as the sum of the predictive losses and penalties for model complexity:

$$J(\Theta) = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(h_k) \quad (8)$$

Where:

$L(y_i, \hat{y}_i)$ = loss function for the iii-th observation,

$\Omega(hk)$ = penalty for the complexity of the kkk-th tree (model).

The penalty for tree complexity considers the number of leaves in the tree and the sum of the squared values of the leaf weights:

$$\Omega(h_k) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2 \quad (9)$$

Where:

γ i λ = regularization parameters,

T = number of leaves,

w_j = weight value in the j-th leaf.

The results of the Gini index depending on the number of selected features for different configurations of the XGBoost model are presented in Figure 9. The analyzed configurations include: Depth 5, n_estimators=200, learning_rate=0.1, and colsample_bytree=0.8. These models demonstrate varying levels of performance on the test set. The XGBoost Depth 5 and XGBoost learning_rate=0.1 models stabilize their Gini index at around 0.54 after considering about five features, which is a better result compared to the XGBoost n_estimators=200 and XGBoost colsample_bytree=0.8 models, which stabilize at around 0.48-0.49.

The analysis of Type I (false positives) and Type II (false negatives) errors presented in Figure 10 shows that the XGBoost Depth 5 and XGBoost learning_rate=0.1 models have lower Type I error values, stabilizing at around 0.30. In contrast, the XGBoost n_estimators=200 and XGBoost colsample_bytree=0.8 models stabilize at values around 0.32-0.33. For Type II errors, the XGBoost Depth 5 and XGBoost learning_rate=0.1 models also show lower values, stabilizing at

around 0.30, which is a better result compared to the XGBoost n_estimators=200 and XGBoost colsample_bytree=0.8 models, which stabilize at around 0.33-0.34.

Table 7 and Figures 9 and 10 present detailed results for the analyzed models, confirming that XGBoost learning_rate=0.1 is the most effective among the other XGBoost models in the context of our dataset and number of features.

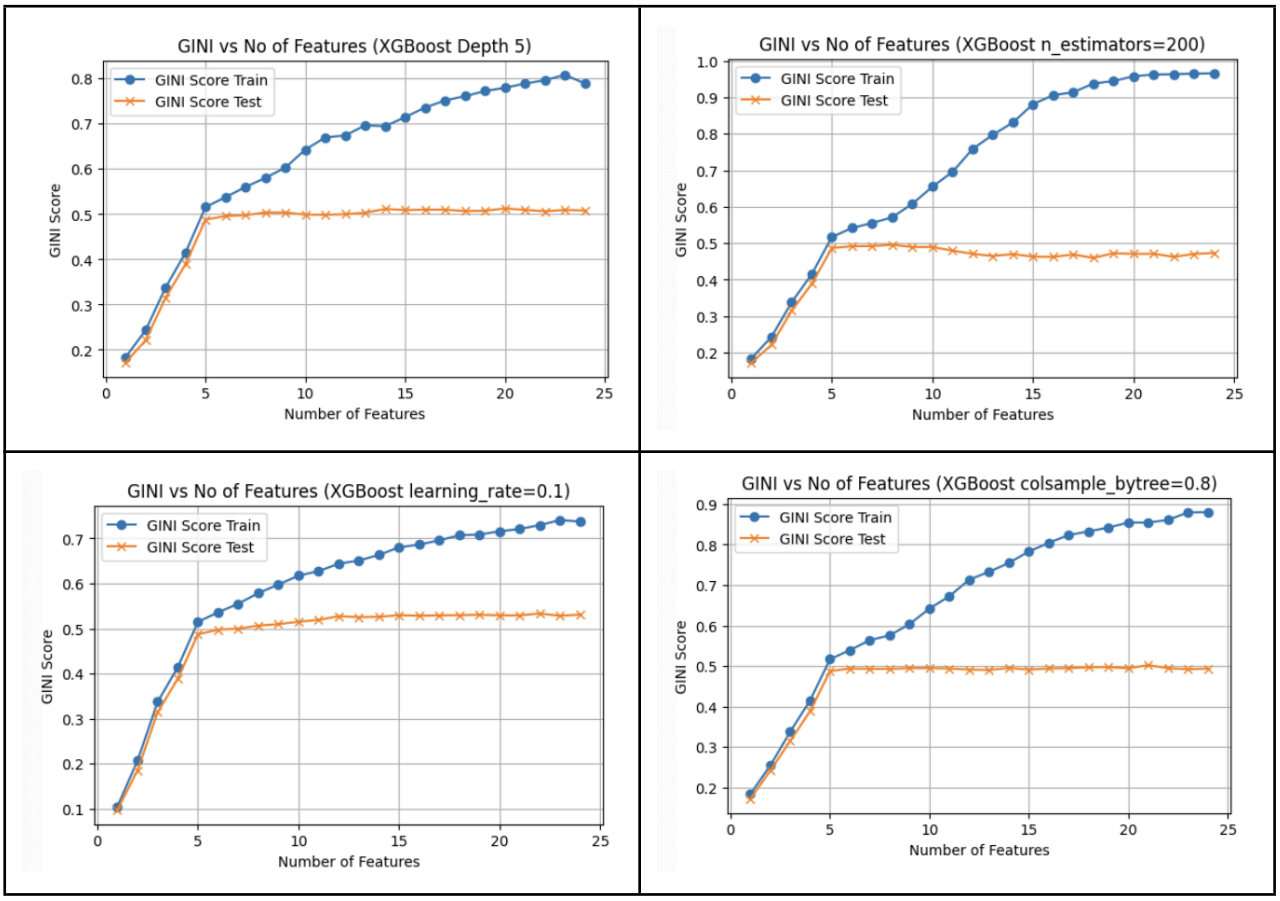
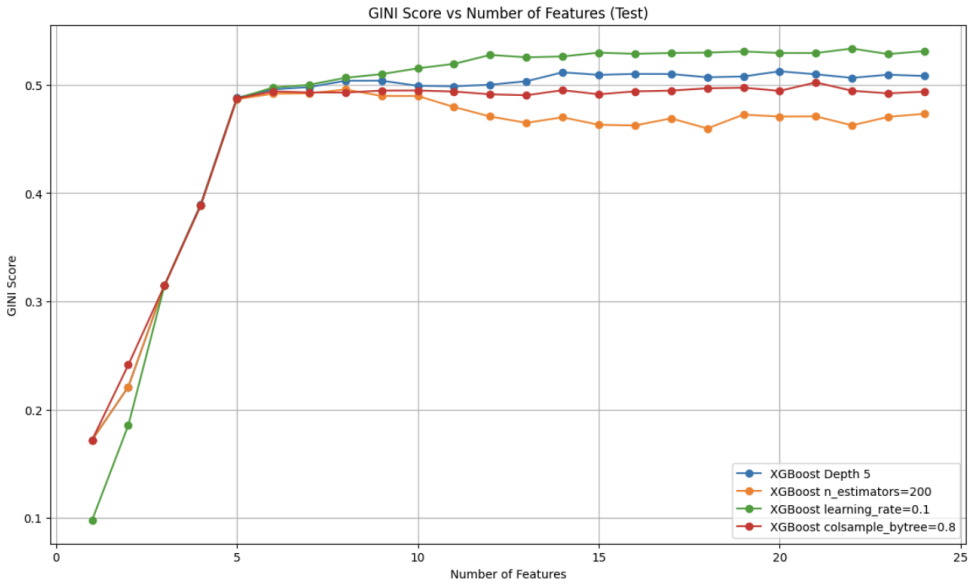
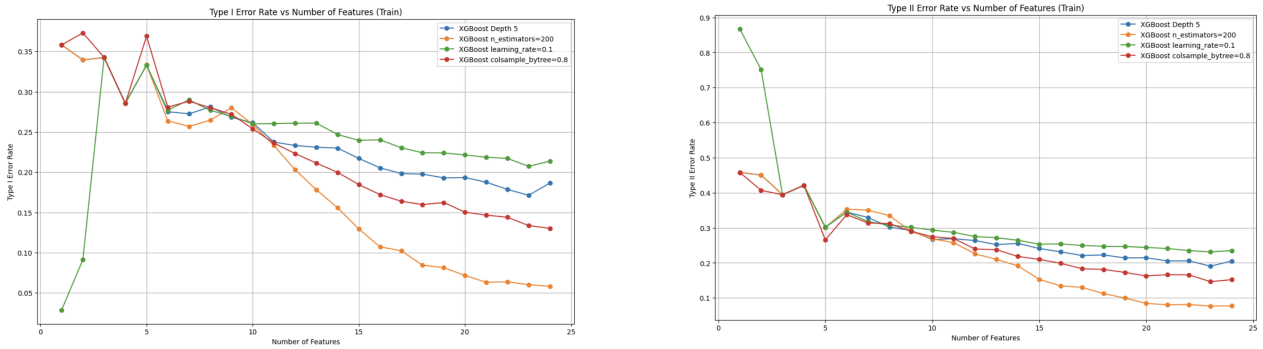


Table 7
Gini Coefficient Results for the Training and Test Sets for the RFE Model with XGBoost Estimators



Graph 11
Gini Coefficient Results for the Training and Test Sets for the RFE Model with XGBoost Estimators



Graph 12
Gini Coefficient Results for the Test Set for the RFE Model with XGBoost Estimator

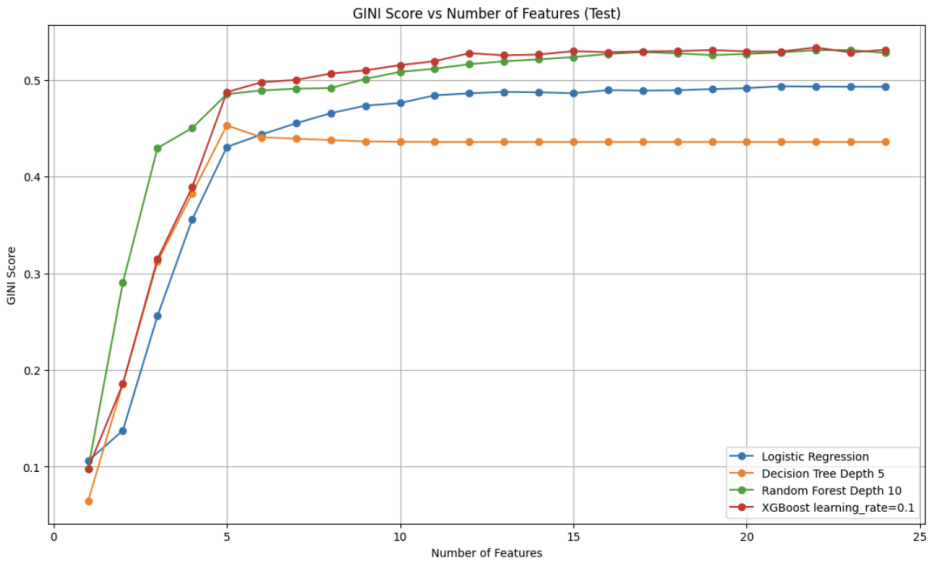
2.7. Comparison of Model Evaluation Criteria

In this section, the focus is on comparing different types of classification models used to assess credit risk. These models were evaluated using the RFE method to prioritize and select variables, as chosen in the previous part of the study. The models are as follows:

- **Logistic Regression**: Classical logistic regression without regularization.
- **Decision Tree**: Decision tree with a maximum depth of 5.
- **Random Forest**: Random forest with a maximum tree depth set to 10.
- **XGBoost**: Extreme gradient boosting with a learning_rate parameter set to 0.1.

As shown in Figure 13, XGBoost and Random Forest consistently outperform the other models in terms of the Gini coefficient (Graph 1). The XGBoost model achieves the highest Gini coefficient values for most numbers of features, indicating its high effectiveness in predicting credit risk. Random Forest also shows good results, especially with a smaller number of features, suggesting that it is effective in modeling with a limited amount of variables. Logistic Regression and Decision Tree achieve lower Gini coefficient values, indicating their limited ability to distinguish between high-risk and low-risk credit clients compared to XGBoost and Random Forest.

.

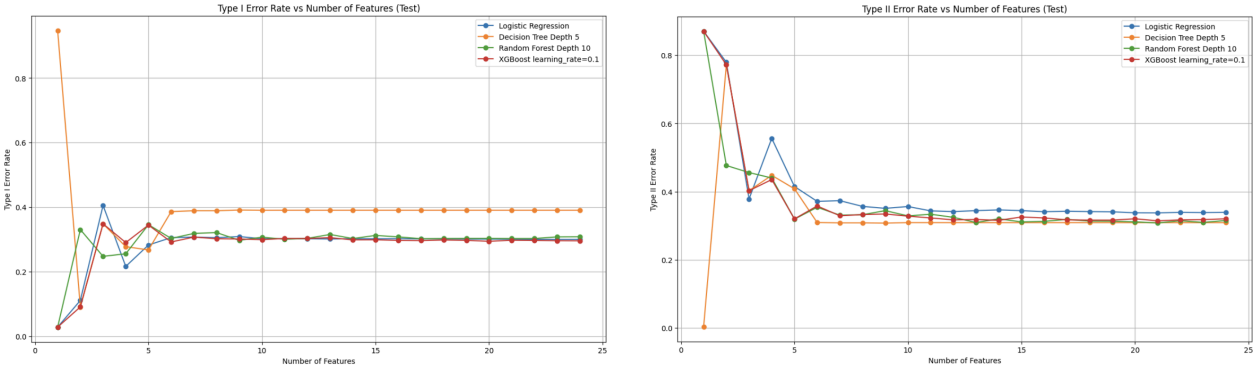


Graph 13
Gini Results for Selected RFE Models with Logistic Regression, XGBoost, Decision Tree, and Random Forest Estimators

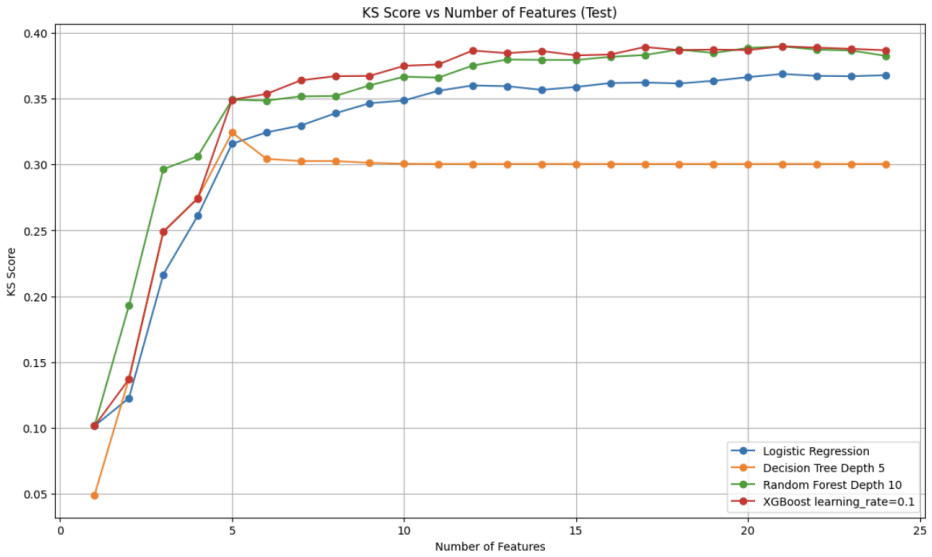
This analysis provides additional insights into the strengths and weaknesses of the individual models (Figures 2 and 3). The XGBoost and Random Forest models stand out again, achieving the lowest values of both Type I and Type II errors. This indicates that these models are more effective in both minimizing false alarms (Type I errors) and detecting high-risk clients (Type II errors). Interestingly, the Decision Tree model, despite having lower Gini values, shows a low number of Type II errors with a small number of variables, making it potentially useful in applications where minimizing false negatives is a priority.

XGBoost and Random Forest models achieve the highest values of KS (Graph 15) and Lift (Graph 16), which means they are more effective in recognizing differences between groups with different risk levels. Additionally, it is noteworthy that the optimal number of features for all models is around 10, with significant improvement in evaluation criteria (Gini, KS - Graph 15, Lift - Graph 16) up to five variables. Beyond this number, the improvement in model performance is minimal, which may suggest the risk of overfitting.

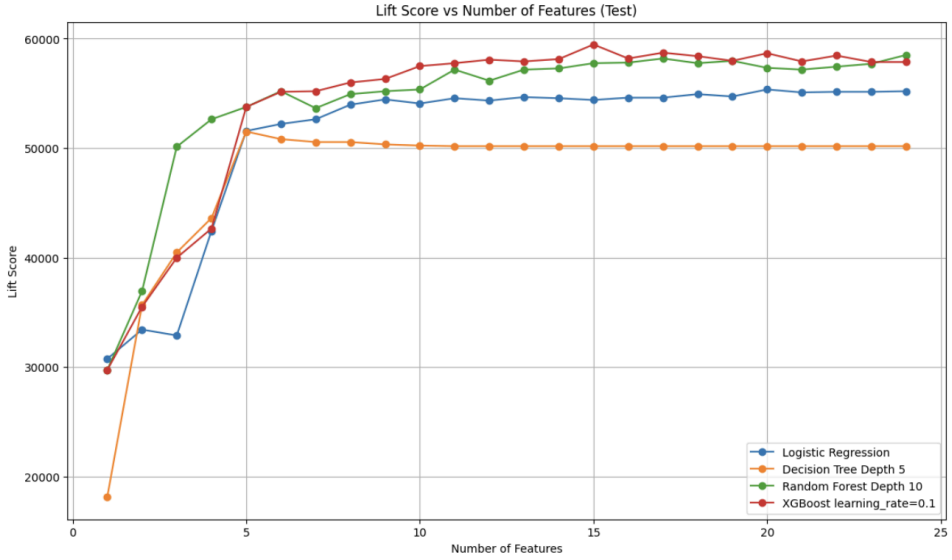
Logistic Regression and Decision Tree, although less effective, may be useful in specific cases where model simplicity and interpretability are priorities.



Graph 14
Results of Type I and Type II Errors for Selected RFE Models with Logistic Regression, XGBoost, Decision Tree, and Random Forest Estimators



Graph 15
Results of KS for Selected RFE Models with Logistic Regression, XGBoost, Decision Tree, and Random Forest Estimators

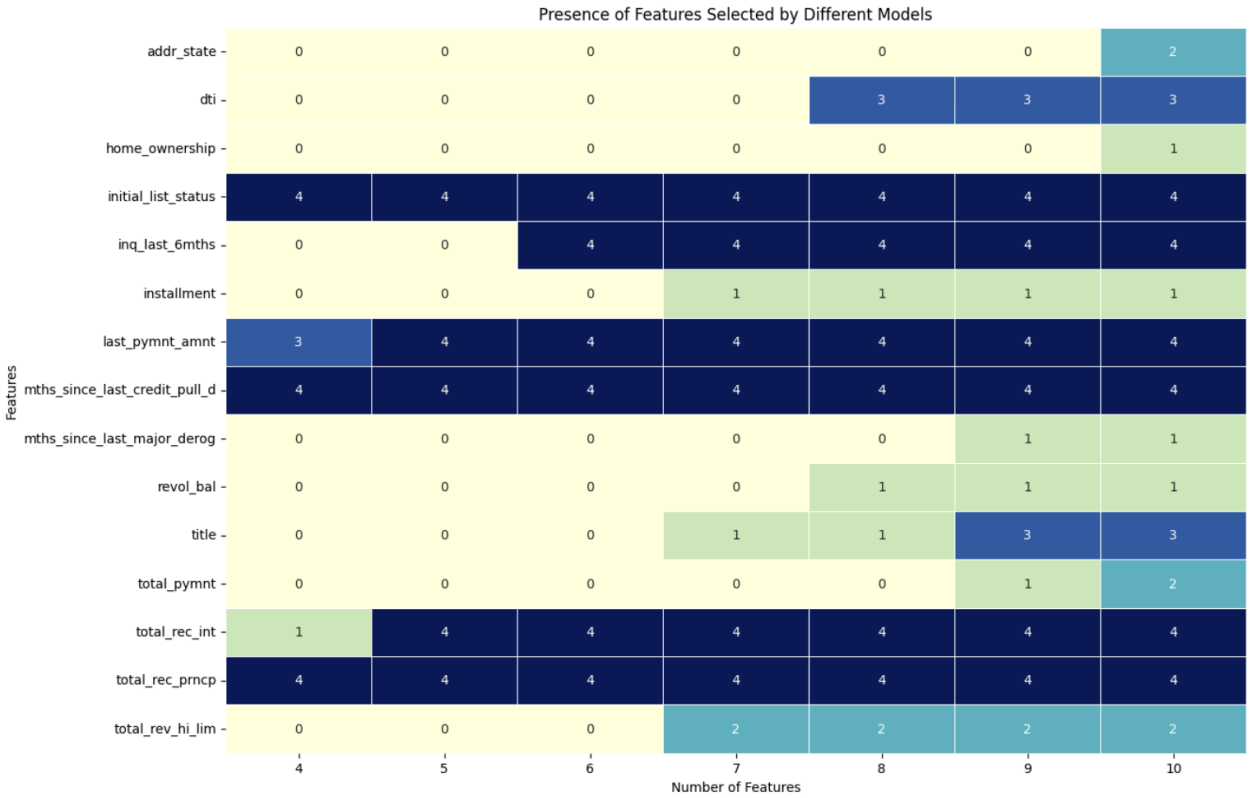


Graph 16
Results of Lift for Selected RFE Models with Logistic Regression, XGBoost, Decision Tree, and Random Forest Estimators

3. Analysis of Selected Variables

In this chapter, we analyze the variables selected using the four chosen estimators, as presented in Figure 17 and the table in Appendix 1. Variables such as initial_list_status, total_rec_prncp, last_pymnt_amnt, and mths_since_last_credit_pull_d are frequently selected by most models, indicating their importance. The analysis also revealed unique variable choices with a larger number of features. For instance, the decision tree often selects variables such as installment and revol_bal, which are not preferred by other models. Conversely, XGBoost frequently selects total_rev_hi_lim with a larger number of features, which is not observed in other models. These differences highlight the impact of different feature selection algorithms on variable choice.

This introduction lays the foundation for further analysis in Chapter 4, where the interpretation of selected variables using LIME and SHAP methods is discussed in detail.



Graph 17
Frequency of Variables Selected by the RFE Models with Logistic Regression, XGBoost, Decision Tree, and Random Forest Estimators

3.1 Comparison of Variable Analysis Using SHAP at a Global Level

SHAP (SHapley Additive exPlanations) is a method for interpreting machine learning model outcomes based on Shapley values from game theory. SHAP explains the impact of each variable on model predictions, allowing for a better understanding of how individual features influence model results. This section compares logistic regression and XGBoost models using SHAP for 5 and 10 selected variables.

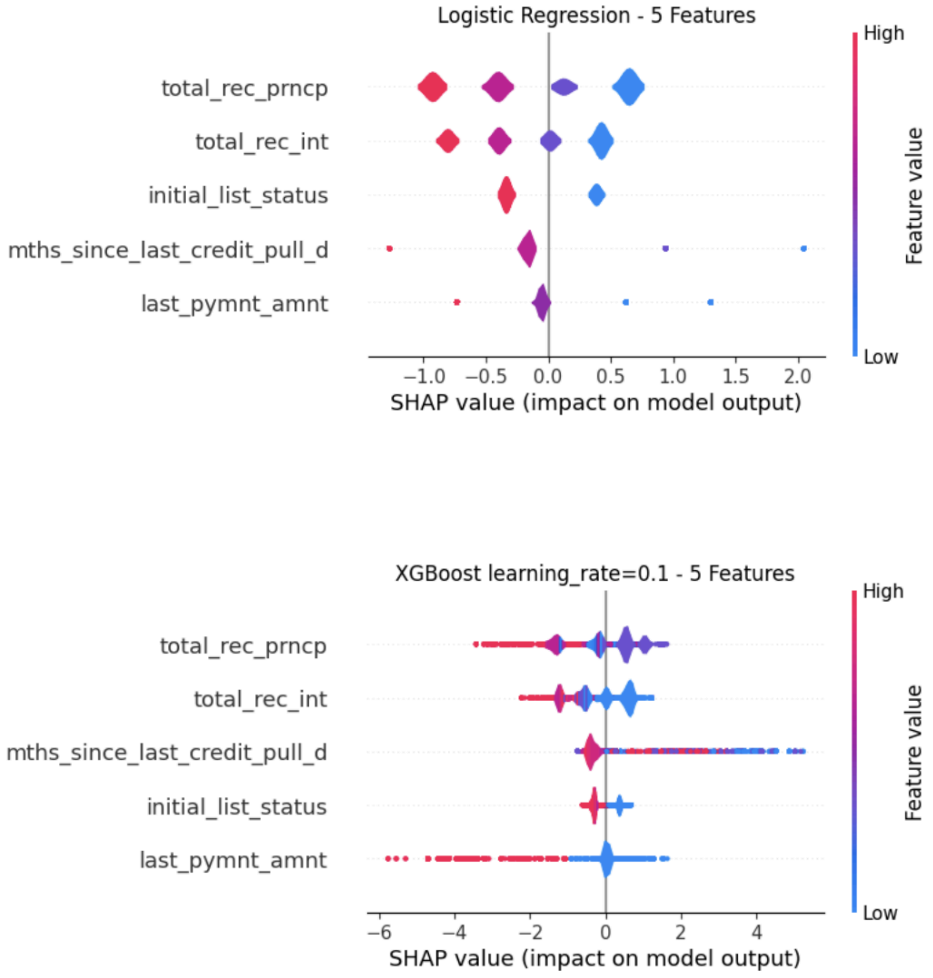
The XGBoost model was chosen due to its highest Gini coefficient values in the test sets - 0.49 for 5 variables and 0.52 for 10 variables - suggesting that XGBoost performs better with the data by capturing more complex patterns. For comparison, logistic regression was also selected for its simplicity. The Gini coefficient for logistic regression is 0.43 for 5 variables and 0.48 for 10 variables. In both models, a larger number of variables (10) achieved better results across most metrics compared to 5 variables.

Analyzing the logistic regression and XGBoost models using 5 variables (Graph 18), it was noted that total_rec_prncp and total_rec_int have the greatest impact on both models' results. In logistic regression, these variables, along with initial_list_status, mths_since_last_credit_pull_d, and last_pymnt_amnt, have a more focused and predictable impact, owing to the simplicity of the linear model. SHAP plots for logistic regression are less scattered, indicating consistency in the variables' influence on the model. In the XGBoost model, the variables' impact is more dispersed, resulting from the model's greater complexity and flexibility. XGBoost better captures complex patterns in the data, leading to greater variation in the influence of individual features.

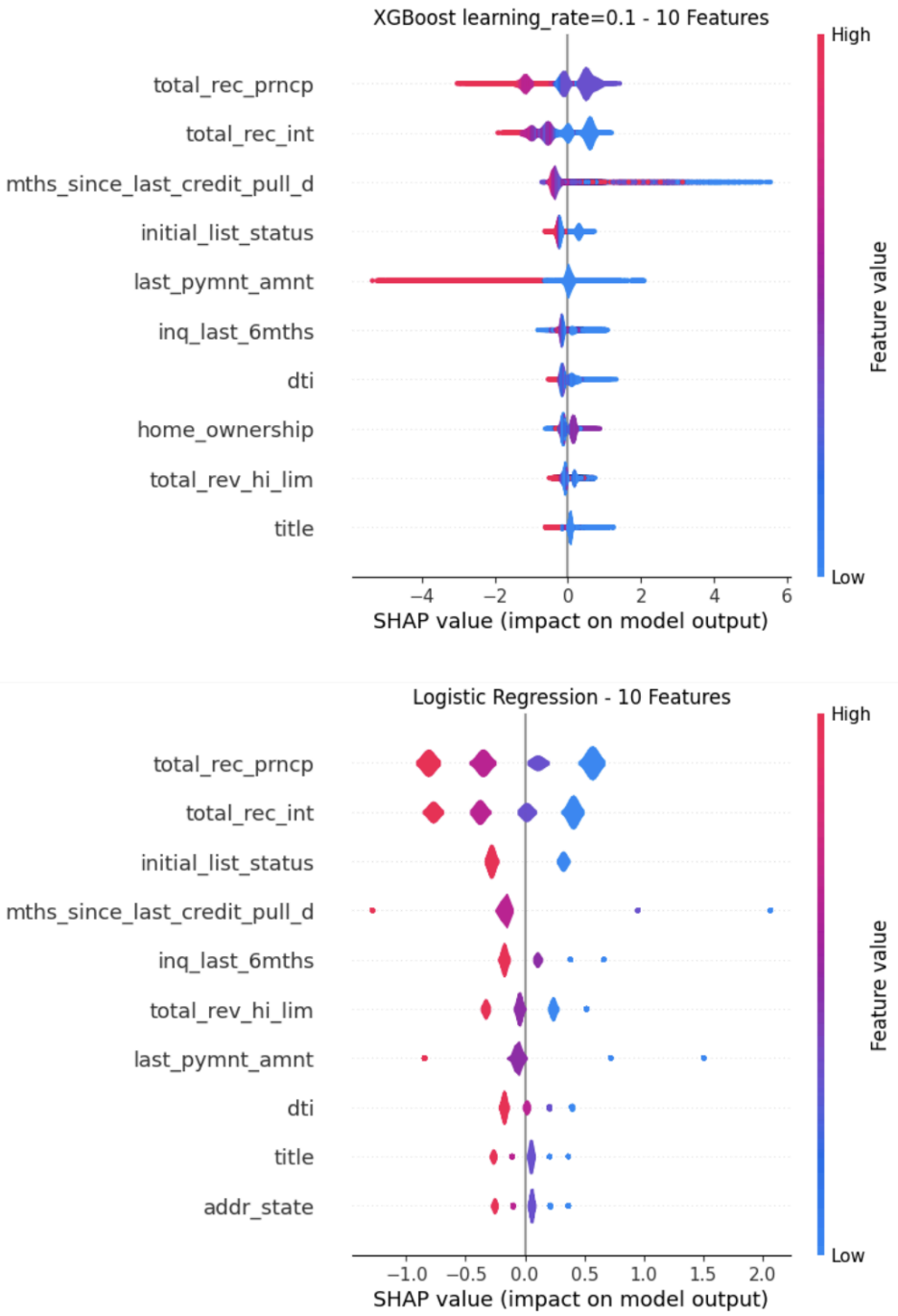
When analyzing models with 10 variables (Graph 19), total_rec_prncp and total_rec_int remain the most important variables in both models. In logistic regression, additional variables such as initial_list_status, mths_since_last_credit_pull_d, inq_last_6mths, total_rev_hi_lim, last_pymnt_amnt, dti, title, and addr_state introduce some variation, but their impact remains relatively focused. For XGBoost, the larger number of variables causes even more dispersion in the impacts, underscoring the model's ability to capture complex relationships in the data. This model shows clearer differences in variable impacts, suggesting better adaptation to complex data patterns.

To better illustrate the average impact of each variable on model results, bar charts were created showing SHAP values as the average absolute value, enabling easy comparison of variable importance. For example, in Figure 20 for the logistic regression model with 5 variables, total_rec_int has an average SHAP value of about 0.45, while initial_list_status, mths_since_last_credit_pull_d, and last_pymnt_amnt have smaller impacts. In the analysis of models with 10 variables (Graph 21), it is shown that in the logistic regression model, total_rec_prncp and total_rec_int remain the most important variables with high average SHAP values of about 0.5 and 0.4, respectively.

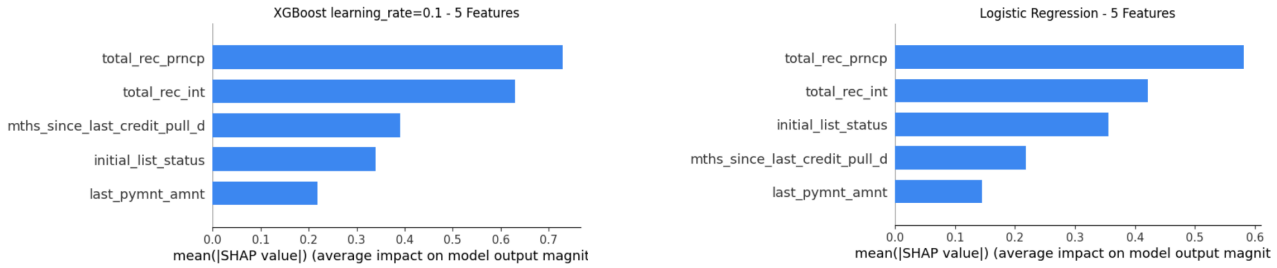
Additional variables, such as initial_list_status, mths_since_last_credit_pull_d, inq_last_6mths, total_rev_hi_lim, last_pymnt_amnt, dti, title, and addr_state, introduce some variation but their impact is relatively smaller and more focused. In the XGBoost model, while total_rec_prncp and total_rec_int also dominate, the impact of other variables is more dispersed. Variables such as mths_since_last_credit_pull_d, initial_list_status, last_pymnt_amnt, inq_last_6mths, dti, home_ownership, total_rev_hi_lim, and title are shown to have varied impacts on the model.



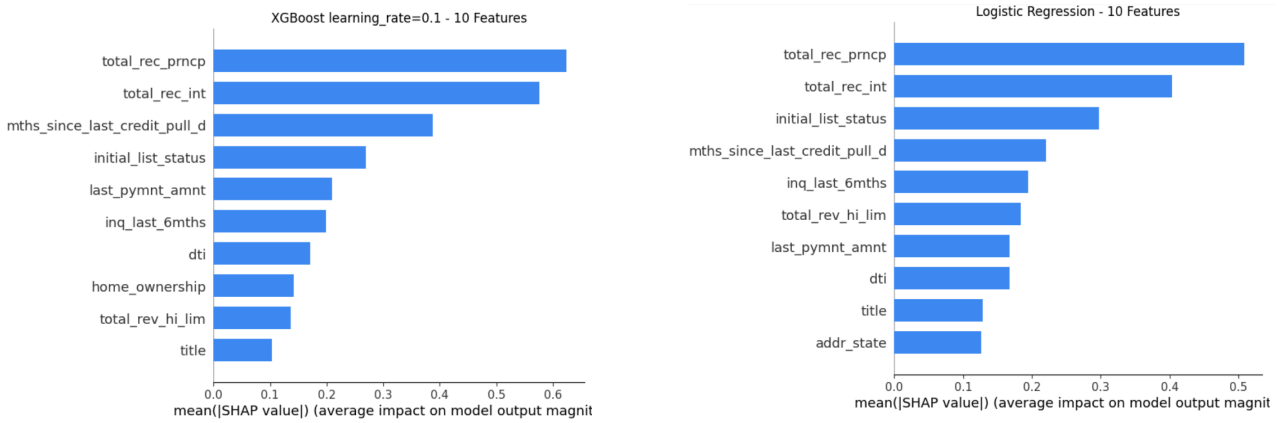
Graph 18
Global SHAP Analysis for XGBoost and Logistic Regression Models with 5 Variables (Violin Plot)



Graph 19
Global SHAP Analysis for XGBoost and Logistic Regression Models with 10 Variables (Violin Plot)



Graph 20
Global SHAP Analysis for XGBoost and Logistic Regression Models with 5 Variables (Bar Plot)



Graph 21
Global SHAP Analysis for XGBoost and Logistic Regression Models with 10 Variables (Bar Plot)

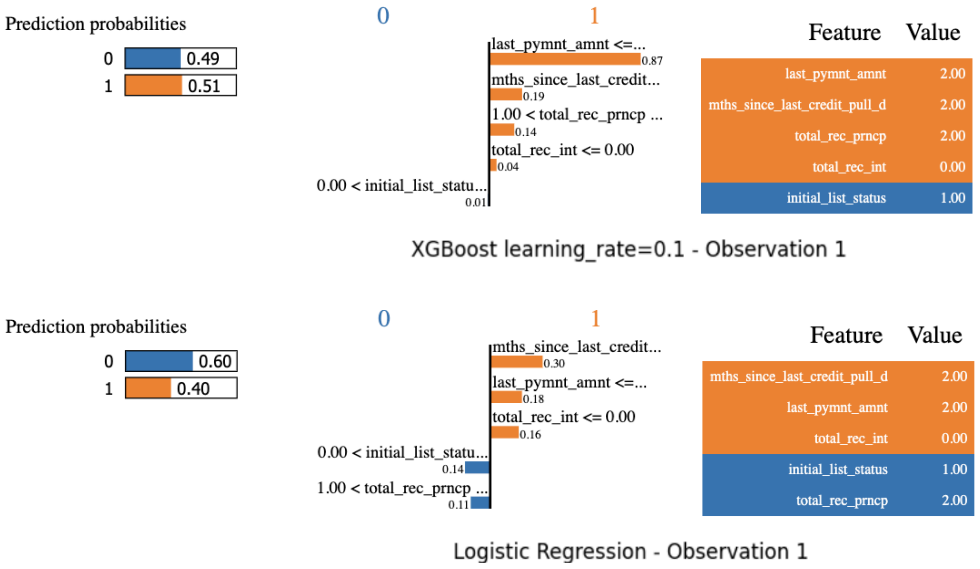
3.2. Comparison of Variable Analysis Using SHAP and LIME at a Local Level

In this section, we compare SHAP and LIME methods in the context of interpreting the influence of individual variables at a local level, that is, for specific observations in the dataset. SHAP is a popular method due to its ability to provide explanations both globally and locally, whereas LIME is primarily used at the local level, i.e., for single observations.

LIME operates by generating perturbations of the original input data, creating modified versions of these data, which are then used to make predictions by the model. LIME then builds a simpler, interpretable model (e.g., linear regression) based on these modified data and their predictions, which allows for identifying which features of the data have the greatest impact on the model's outcome at that specific point (Qiu et al., 2022; Munkhdalai et al., 2019).

Figure 22 shows the results of local LIME analysis for XGBoost and logistic regression models for Observation 1. For the XGBoost model, the prediction for class 0 (no default) is 0.49, whereas for class 1 (default) it is 0.51, indicating a slight prediction in favor of default. Conversely, the logistic regression model predicts a value of 0.60 for class 0 and 0.40 for class 1, indicating a prediction of no default for this observation.

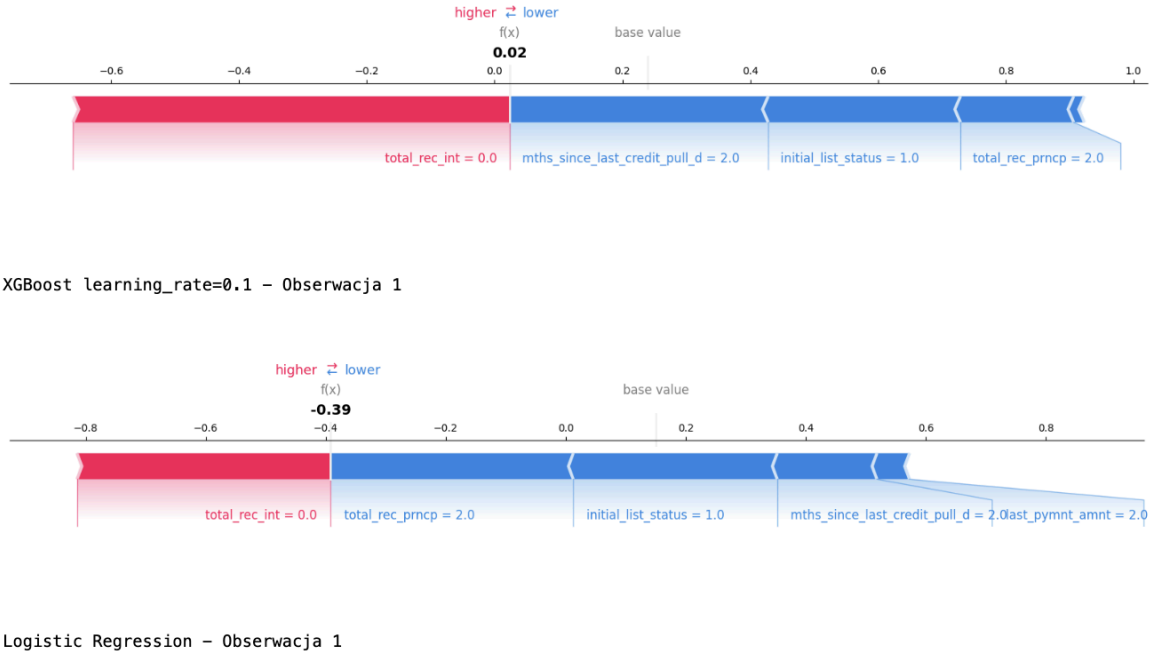
Analyzing the impact of individual features on predictions, it was noted that in the XGBoost model, the features last_pymnt_amnt, mths_since_last_credit_pull_d, and total_rec_prncp have the greatest impact on the prediction of default. For logistic regression, mths_since_last_credit_pull_d and last_pymnt_amnt also have a significant impact on the prediction, but in the opposite direction—the model leans towards no default. The difference in the impact of variables, such as total_rec_prncp, which contributes to default in XGBoost and to no default in logistic regression, is crucial. LIME plots indicate the bins to which the variables belong, which helps visualize the client's profile. For example, initial_list_status is in bin 1 in both models and contributes to predicting no default.



Graph 22
Local LIME Analysis for XGBoost and Logistic Regression Models with 5 Variables (Observation 1)

In the case of SHAP, in Figure 23, the plots also show how each variable contributes to the final model prediction. For the XGBoost model, the value f(x)=0.02, which means the model is slightly inclined towards classifying as non-default. However, for logistic regression, f(x)=−0.39, indicating a stronger inclination towards classifying as default. The SHAP plots show that in logistic regression, only the variable total_rec_int has an opposite impact compared to the results obtained using LIME.

Analyzing the results of both methods, it can be observed that LIME and SHAP mostly agree on the influence of individual variables on the model. However, for logistic regression, the variable total_rec_int in SHAP has an opposite impact compared to the results obtained using LIME, which is due to the differences between the model explanation methods.



Graph 23
Local SHAP Analysis for XGBoost and Logistic Regression Models with 5 Variables (Observation 1)

4. Discussion of Results

Recall that the primary objective of this study was to investigate which models achieve the best results in the context of credit risk assessment, how the reduction of the number of variables affects model quality, and how different RFE estimators impact the explainability of models and observations from ABT using SHAP and LIME analyses. The discussion includes a detailed analysis of the obtained results and references to the selected research questions.

Which models, considering the selected evaluation criteria, achieve the best results?

The analysis conducted in this study allowed for comparing various models in terms of their effectiveness in predicting credit risk. The best results were achieved by the XGBoost models with a learning rate of 0.1 and Random Forest with a tree depth of 10. The XGBoost model achieved the highest Gini coefficient values for most numbers of features, indicating its high efficiency in distinguishing between high and low credit risk clients. Random Forest also showed good

performance, especially with fewer features, suggesting its effectiveness in situations where data is limited. Logistic Regression and Decision Tree models achieved lower Gini coefficient values, indicating their limited ability to distinguish credit risk compared to the aforementioned estimators.

How does significant reduction in the number of variables allow for maintaining a high Gini coefficient?

The analysis results indicate that reducing the number of variables to about 10 features allows for maintaining a high Gini coefficient without a significant loss in model quality. The optimal number of features was around 5 to 10, with most models achieving stable and high Gini coefficient values with this number of variables. Beyond this number, further addition of features brought marginal benefits and could even lead to model overfitting. As demonstrated, reducing the number of variables can be crucial for increasing computational efficiency and model explainability, which is particularly important in the context of credit risk management.

How do different RFE estimators affect the explainability of models and observations from ABT using SHAP and LIME analyses?

Different RFE (Recursive Feature Elimination) estimators significantly affect model explainability. The analysis using SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) allowed for a deeper understanding of the impact of individual variables on model outcomes. The XGBoost and Random Forest models, which achieved the highest Gini coefficient values, also demonstrated more complex and dispersed variable impacts, resulting from their flexibility in capturing intricate patterns in the data. Using SHAP and LIME made it possible to identify key variables such as total_rec_prncp and total_rec_int, which had the greatest impact on model predictions.

In local analysis, LIME and SHAP plots for individual observations showed agreement on the impact of most variables. LIME uses data perturbation and variable binning methods, presenting impacts in table form. In contrast, SHAP provides more detailed explanations of variable impacts both globally and locally, allowing for a better understanding of model mechanisms. There is a trend towards increasing the popularity of SHAP due to its ability to explain models at both levels.

5. Summary

In this study, a detailed analysis of various predictive models used in credit risk assessment was conducted, focusing on their performance and interpretability using SHAP and LIME methods. The research results indicate that XGBoost and Random Forest models achieve the highest Gini coefficient values, demonstrating their high effectiveness in distinguishing between high and low credit risk clients. The optimal number of features for these models ranges from 5 to 10, allowing for high prediction quality without significant loss in accuracy.

The analysis using SHAP and LIME enabled a deeper understanding of the impact of individual variables on model outcomes. The XGBoost and Random Forest models showed more complex and dispersed variable impacts, resulting from their flexibility in capturing intricate patterns in the data. Key variables such as total_rec_prncp and total_rec_int had the greatest impact on model predictions. LIME and SHAP plots for individual observations showed agreement on the impact of most variables, with LIME using data perturbation and variable binning methods to present impacts in table form, while SHAP provides more detailed explanations at both global and local levels. There is a trend towards increasing the popularity of SHAP due to its ability to explain models at both levels.

Business Implications

The results of this analysis have significant implications for credit risk management in financial institutions. Effective variable selection is crucial for increasing computational efficiency and model interpretability, which is particularly important in a dynamic financial environment. Predictive models that are both accurate and computationally efficient enable quick and precise credit decision-making. Automating the credit assessment process and using modern technologies allows for analyzing large amounts of data and identifying key patterns and relationships indicating potential insolvency risk. This enables banks and financial institutions to better manage credit risk, minimize financial losses, and increase operational efficiency. Implementing advanced predictive models along with explanation methods like SHAP and LIME can significantly improve the credit risk assessment process and contribute to better business outcomes.

List of Figures

1. Elbow Curve Plots for Different Criteria:

    1a: Information Gain

    1b: Chi-square

    1c: Fisher Score

2. Gini Coefficient Results (a) and KS Statistic (b) for the RFE Model with Logistic Regression Estimator

3. Type I and Type II Error Results for the RFE Model with Logistic Regression Estimator

4. Gini Results for the Test Set for the RFE Model with Logistic Regression Estimator with L2 Penalty

5. Type I and Type II Error Results for the RFE Model with Logistic Regression Estimator with L2 Penalty for Training and Test Sets

6. Decision Tree Plot with Default Hyperparameters Reduced to 4 Nodes

7. Gini Results for the Test Set for the RFE Model with Decision Tree Estimator for the Test Set

8. Type I and Type II Error Results for the RFE Model with Decision Tree Estimator for Training and Test Sets

9. Gini Results for the Test Set for the RFE Model with Random Forest Estimator for the Test Set

10. Type I and Type II Error Results for the RFE Model with Random Forest Estimator for Training and Test Sets

11. Gini Results for the Test Set for the RFE Model with XGBoost Estimator for the Test Set

12. Type I and Type II Error Results for the RFE Model with XGBoost Estimator for Training and Test Sets

13. Gini Results for Selected RFE Models with Logistic Regression, XGBoost, Decision Tree, and Random Forest Estimators

14. Type I and Type II Error Results for Selected RFE Models with Logistic Regression, XGBoost, Decision Tree, and Random Forest Estimators

15. KS Results for Selected RFE Models with Logistic Regression, XGBoost, Decision Tree, and Random Forest Estimators

16. Lift Results for Selected RFE Models with Logistic Regression, XGBoost, Decision Tree, and Random Forest Estimators

17. Frequency of Variables Selected by RFE Models with Logistic Regression, XGBoost, Decision Tree, and Random Forest Estimators

18. Global SHAP Analysis for XGBoost and Logistic Regression Models with 5 Variables (Violin Plot)

19. Global SHAP Analysis for XGBoost and Logistic Regression Models with 10 Variables (Violin Plot)

20. Global SHAP Analysis for XGBoost and Logistic Regression Models with 5 Variables (Bar Plot)

21. Global SHAP Analysis for XGBoost and Logistic Regression Models with 10 Variables (Bar Plot)

22. Local LIME Analysis for XGBoost and Logistic Regression Models with 5 Variables (Observation 1)

23. Local SHAP Analysis for XGBoost and Logistic Regression Models with 5 Variables (Observation 1)

List of Tables

1. Table Fragment with Indicators: Good Share, Bad Share, Logit, Information Value
2. Table Fragment with Gini Indicators for Training and Test Sets
3. Comparison of Variables Using Information Gain, Chi-square Test, and Fisher Score Methods
    - Key variables highlighted in blue
    - Variables recommended for removal marked in gray
4. Evaluation Criteria for RFE with Logistic Regression Estimator for 5 and 10 Variables
5. Evaluation Criteria for RFE with Logistic Regression Estimator with Ridge Regularization for Up to 4 Variables
6. Gini Coefficient Results for Training and Test Sets for RFE Model with Decision Tree Estimators
7. Gini Coefficient Results for Training and Test Sets for RFE Model with Random Forest Estimators

Literature

1. Credit Scoring Series Part Four: Variable Selection. (n.d.). Default. Retrieved June 18, 2024, from https://altair.com/blog/articles/credit-scoring-series-part-four-variable-selection

2. Ernesto, & Hashmi, M. (2021, May 29). How to Interpret a Machine Learning Model with Python's SHAP Library? Dr. Ernesto Lee. https://ernesto.net/shap-how-to-interpret-machine-learning-models-with-python/

3. Kaszynski, D., Kaminski, B., & Szapiro, T. (2020). Credit Scoring in context of interpretable machine learning theory and practice: Vol. ISBN 978-83-8030-424-6 (First Edition). SGH.

4. Koç, O., Ugur, Ö., & Kestelc, A. S. (2023). The impact of featrue selection and transformation on machine learning methods in determining the credit scoring. https://arxiv.org/pdf/2303.05427

5. Laborda, J., & Ryoo, S. (2021). Feature Selection in a Credit Scoring Model. Mathematics, 9(7), 746. https://doi.org/10.3390/math9070746

6. Munkhdalai, L., Wang, L., Park, H., & Ryu, K. (2019). Advanced Neural Network Approach, Its Explanation with LIME for Credit Scoring Application. , 407-419. https://doi.org/10.1007/978-3-030-14802-7_35.

7. Przanowski, K. (2014). Credit Scoring w erze Big-data. Techniki modelowania z wykorzystaniem generatora losowych danych portfela Consumer Finance (Wydanie I).

8. Qiu, L., Yang, Y., Cao, C., Zheng, Y., Ngai, H., Hsiao, J., & Chen, L. (2022). Generating Perturbation-based Explanations with Robustness to Out-of-Distribution Data. Proceedings of the ACM Web Conference 2022. https://doi.org/10.1145/3485447.3512254.

9. Tan, F. (2019, December 10). Feature Selection in Credit Scoring. Medium. https://medium.com/@finntanweelip/feature-selection-in-credit-scoring-b0eee604cd51

10. Trivedi, S. K. (2020). A study on credit scoring modeling with different feature selection and machine learning approaches. Technology in Society, 63, 101413. https://doi.org/10.1016/j.techsoc.2020.101413

11. Zhou, Y., Uddin, M. S., Habib, T., Chi, G., & Yuan, K. (2021). Feature selection in credit risk modeling: an international evidence. Economic Research-Ekonomska Istraživanja, 1–31. https://doi.org/10.1080/1331677x.2020.1867213

Appendix 1

F. Count	Logistic Regression	Decision Tree Depth 5	Random Forest Depth 10	XGBoost learning_rate=0.1
4	['initial_list_status', 'total_rec_prncp', 'last_pymnt_amnt', 'mths_since_last_credit_pull_d']	['initial_list_status', 'total_rec_prncp', 'last_pymnt_amnt', 'mths_since_last_credit_pull_d']	['initial_list_status', 'total_rec_prncp', 'total_rec_int', 'mths_since_last_credit_pull_d']	['initial_list_status', 'total_rec_prncp', 'last_pymnt_amnt', 'mths_since_last_credit_pull_d']
5	['initial_list_status', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'mths_since_last_credit_pull_d']	['initial_list_status', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'mths_since_last_credit_pull_d']	['initial_list_status', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'mths_since_last_credit_pull_d']	['initial_list_status', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'mths_since_last_credit_pull_d']
6	['inq_last_6mths', 'initial_list_status', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'mths_since_last_credit_pull_d']	['inq_last_6mths', 'initial_list_status', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'mths_since_last_credit_pull_d']	['inq_last_6mths', 'initial_list_status', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'mths_since_last_credit_pull_d']	['inq_last_6mths', 'initial_list_status', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'mths_since_last_credit_pull_d']
7	['inq_last_6mths', 'initial_list_status', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'total_rev_hi_lim', 'mths_since_last_credit_pull_d']	['installment', 'inq_last_6mths', 'initial_list_status', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'mths_since_last_credit_pull_d']	['title', 'inq_last_6mths', 'initial_list_status', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'mths_since_last_credit_pull_d']	['inq_last_6mths', 'initial_list_status', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'total_rev_hi_lim', 'mths_since_last_credit_pull_d']
8	['dti', 'inq_last_6mths', 'initial_list_status', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'total_rev_hi_lim', 'mths_since_last_credit_pull_d']	['installment', 'inq_last_6mths', 'revol_bal', 'initial_list_status', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'mths_since_last_credit_pull_d']	['title', 'dti', 'inq_last_6mths', 'initial_list_status', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'mths_since_last_credit_pull_d']	['dti', 'inq_last_6mths', 'initial_list_status', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'total_rev_hi_lim', 'mths_since_last_credit_pull_d']
9	['title', 'dti', 'inq_last_6mths', 'initial_list_status', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'total_rev_hi_lim', 'mths_since_last_credit_pull_d']	['installment', 'inq_last_6mths', 'revol_bal', 'initial_list_status', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'mths_since_last_major_derog', 'mths_since_last_credit_pull_d']	['title', 'dti', 'inq_last_6mths', 'initial_list_status', 'total_pymnt', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'mths_since_last_credit_pull_d']	['title', 'dti', 'inq_last_6mths', 'initial_list_status', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'total_rev_hi_lim', 'mths_since_last_credit_pull_d']
10	['title', 'addr_state', 'dti', 'inq_last_6mths', 'initial_list_status', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'total_rev_hi_lim', 'mths_since_last_credit_pull_d']	['installment', 'inq_last_6mths', 'revol_bal', 'initial_list_status', 'total_pymnt', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'mths_since_last_major_derog', 'mths_since_last_credit_pull_d']	['title', 'addr_state', 'dti', 'inq_last_6mths', 'initial_list_status', 'total_pymnt', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'mths_since_last_credit_pull_d']	['home_ownership', 'title', 'dti', 'inq_last_6mths', 'initial_list_status', 'total_rec_prncp', 'total_rec_int', 'last_pymnt_amnt', 'total_rev_hi_lim', 'mths_since_last_credit_pull_d']