CREDIT RISK ANALYSIS OF GERMAN CREDIT DATASET USING LOGISTICS AND BOOSTING TECHNIQUES IN MACHINE LEARNING

BUI PHUONG THAO

Abstract

Credit risk analysis has long been an attention-capturing area for any financial institutions since it allows them to reach as objective as possible decisions when it comes to determine the default possibility of a borrower. Regarding the recent difficulties the global economic is encountering, the appropriate allocation of resources is of high priority, thus it is understandable why financial institutions like banks place an emphasis on identifying the legible loaners. The question of interest is to figure out which methods bring about higher efficiency and whether different approaches of feature engineering would result in significant changes in the overall performance of the same model. Within the scope of this research, a variety of common machine learning methods are utilized, along with trials for different methods when preprocessing data. It has been notable that XGBoost stands out with highest accuracy

I. Introduction

Credit risk analysis, as commonly understood and defined, is the process of assessing the probability of a customer defaulting on a payment. Accurately determining the creditworthiness of an individual is a complicated yet crucial task for financial institutions since failing to do so would result in that institution suffering from severe loss (Emmanuel et al, 2022). The banking system normally categorizes applicants into good and bad, the former class represents those who have a better reputation for paying on time while the opposite is true for the latter.

One of the most prevalent statistical methods implemented for the past works of default assessment is Logistic Regression (W. *et al 2022)* . That is the reason why this paper's work has chosen the mentioned model to be the baseline. However, Wang et al 2020 found out that Logistic Regression could not outperform other machine learning models in terms of accuracy, conventional techniques such a Support Vector Machines, k-Nearest Neighbor, Random Forest. Therefore, those models were also implemented to provide some comparison with the performance of the baseline in this paper.

Another worth considering issue when working with credit data is the fact that the number of non-default transactions outnumber the default ones, which means predictors have to face the problem of imbalanced dataset. Preceding studies show that machine learning models working on imbalanced data are prone to poor performance, to mitigate the problem, studying ensemble methods is required (Lu Wang 2021).

Within the scope of this research, the main focus is placed on the application of Logistic Regression and other machine learning models with a view to record the difference in the

final performance. I also want to see if there are some clear changes when the features of German Credit dataset is preprocessed differently. To achieve further improvement in the performance, ensemble models are also applied. It is expected that the conclusions derived from this paper can contribute to the literature on the application of machine learning in the field of credit risk analysis.

II. Theoretical background

1. Models

1.1. Logistic Regression

Logistic regression is a statistical method used for binary classification, where the results are typically represented as either 0 or 1. It estimates the probability a given observation belongs to one of the two categories, returning a value lies between 0 and 1. The conversion of predictor variable into probabilities is made possible by the logistic function, also known as the sigmoid function, which has an S-shaped curve that maps real-valued number to the range [0,1].

The model has the following form:

$$log\frac{\rho}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

Where X_i with i ∈(1,k) are the independent variables, p is the probability and β_i are regression coefficients. Researchers normally pursue the value of coefficients that could minimize the error in the predicted probabilities.

1.2. K-Nearest Neighbors (KNN)

KNN is a straightforward machine learning algorithm used for classification and regression, which assumes that similar data points would have similar output, thus classifies a new data point by examining the class labels of its K nearest neighbors. To be more specific, KNN calculates the distance between a new data point with the rest of the training dataset, then the top K neighbors with shortest distances are selected. After that, the majority voting of those neighbors' classes is also the one associated to the new data point. The metric used for computing the distance may vary from Euclidean to Manhattan, the choice of which can have significance impact on the algorithm performance.

1.3. Decision Trees

Decision Trees are known as intuitive and interpretable models for decision-making tasks in machine learning. Basically, Decision Trees provide a hierarchical structure made of interior nodes and edges. Each node corresponds to a variable and leads to a child representing a possible value of that variable. The leaves lie at the bottom of the tree, implying the predicted value of the target variable. A key advantage of Decision

Trees lies in their interpretability as well as its flexibility in handling both numerical and categorical data.

1.4. Ensemble Classifiers

Ensembles are sets of machine learnings whose outputs are combined before reaching the final decision of the overall system. To be more specific, boosting is an ensemble technique whose aim is to improve the predictions resulting from a decision tree. This process is carried out by using base models, also known as weak learners. The trees are grown sequentially so that each tree is grown using information derived from the previous ones, trying to reduce the errors made from existing trees.

The ensemble techniques applied during this paper's work are mainly tree-based ensembles (TBE), including light gradient boosting (LightGBM), categorical boosting (CatBoost), Adaptive Boosting (AdaBoost) and Extreme Gradient Boosting (XGBoost).

In the timeline of tree-based ensemble methods, AdaBoost emerges as the pioneer in 1996. The achievement process can be described with the following steps: Initially, each given data sample is assigned with equal weight, after each iteration, misclassified samples have their weight increased, the opposite is true for the correctly classified cases, thus, subsequent weak learners would focus more on the incorrect samples from preceding iterations.

After AdaBoost laid the groundwork for boosting algorithms, XGBoost recorded its debut in 2014. It employs a regularized gradient boosting framework that minimizes loss function by adding decision trees to the ensemble in each iteration. During which, XGBoost uses the first and second derivative to reduce the risk of overfitting.

LightGBM then emerged as a boosting technique that prioritizes speed and efficiency for large datasets through sampling and bundling techniques. Particularly, instances that contribute most to the loss function are selectively sampled while exclusive features are bundled together, which helps reduce memory usage thus accelerating training speed.

Finally, CatBoost was designed to specialize in handling categorical features without preprocessing. With ordered boosting, a random permutation is carried out and the sample weights are adjusted based on the error of the previous iteration.

Each method has its own approach, but they share the same goal is to improve the overall predictive accuracy.

2. Weight of Evidence and Information Value

2.1. Weight of Evidence (WOE)

When encountering binary classifications such as the probability of default in credit risk, there are plenty of variables, each has a different range of values, thus appropriate variable selection is necessary. To achieve that, a benchmark is required, and WOE (Weight of Evidence) - a transformation technique is implemented. WOE measures the strength of the relationship between an independent variable and the target one. Mathematically, WOE is defined as follows:

$$\text{Weight of Evidence} = ln\left(\frac{\%good}{\%bad}\right)$$

A positive WOE indicates that this variable is more likely to be associated with non-events, while a negative WOE suggests a higher likelihood of events. The closer to zero a WOE value, the less predictive power of a variable.

WOE can be applicable for both continuous and categorical features. If the variable is continuous, data should be split into 10 bins, it is recommended that each bin should contain at least 5% cases. After that, the numbers of good and bad classes are counted to find the percentage for each of them before calculating WOE with the aforementioned formula.

2.2. Information Value (IV)

Information value is a metric that quantifies the ability of a variable to distinguish between outcomes of the target variable. It is calculated by summing the differences between the proportion of events and non-events for each independent variable before multiplying by the corresponding WOE values. A high IV value indicates higher importance of a variable, according to the following table:

IV	Statistical Strength
<= 0.02	Insignificant Predictive Power
0.02 - 0.1	Weak Predictive Power
0.1 - 0.3	Medium Predictive Power
0.3 - 0.5	Strong Predictive Power
>0.5	Suspicious; too good to be true

III. Data

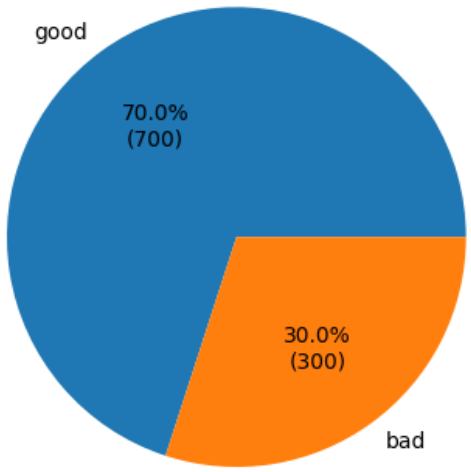
3.1. Data description

The dataset used in this paper is named German Credit Data, provided by Dr. Hans Hofmann, it classifies individuals (each with a set of both categorical and numeric attributes) as good or bad credit risk. The dataset consists of 1000 instances with 20 columns, and there are no missing values. The dataset can be accessed here.

3.2. Exploratory data analysis (EDA)

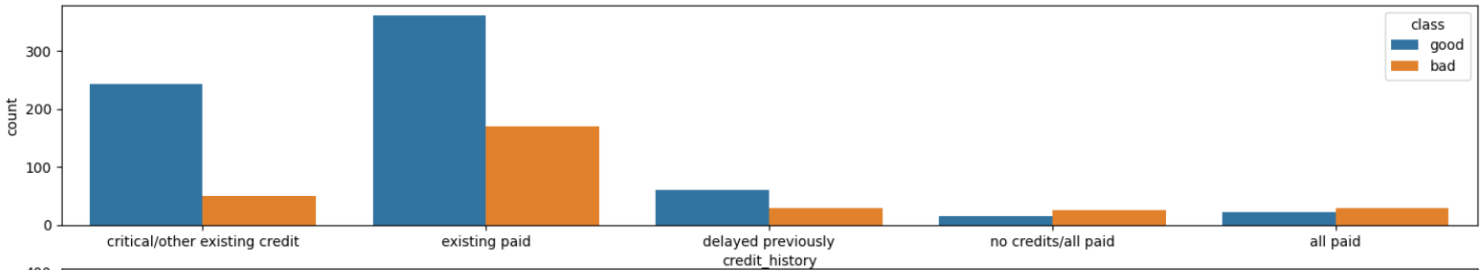
It can be observed from the target variable, which is the "class" column that there is quite a disproportion between the good and bad credit risks. The number of bad credit account for 30%, almost half as much as the good one. This implies that we have to deal with an imbalanced dataset.

Figure 1. Bad credit ratio



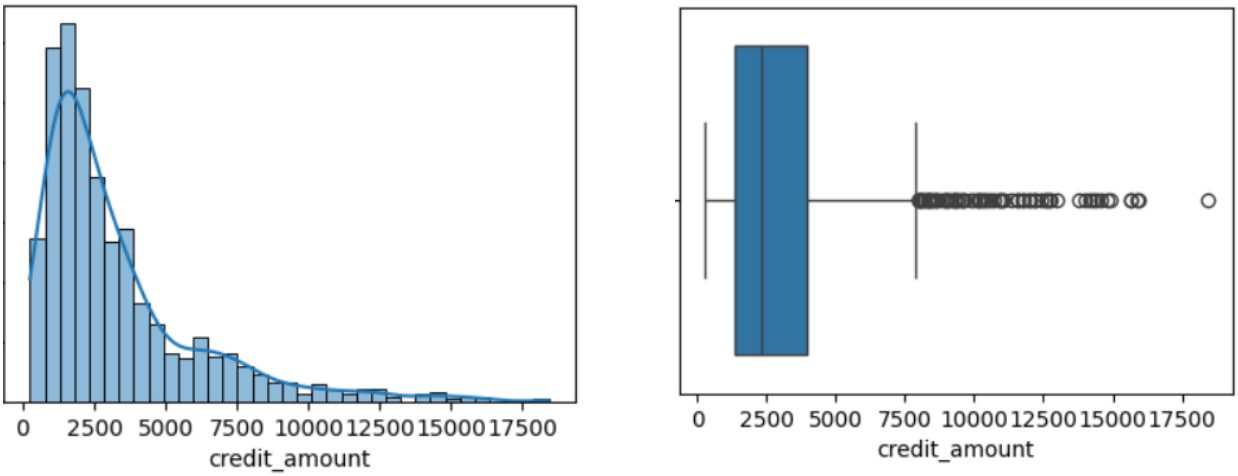
When examining "credit_history", it is noticeable that individuals whose records are no credits or all paid are more likely to be classified as bad credit. It may be understandable for a financial institution to deny a completely new customer since there is no record of their transactions. However, it is quite puzzling to me when individuals who fully paid have higher chance of being denied, it is assumed that this happens because of interest rate (the more the customer pays well, the less the bank will earn over interest rate).

Figure 2: Bar plot of credit history on class



When it comes to numeric features, it is noticeable that “credit_amount” witnesses skewness in its distribution, indicating that there are several outliers need dealing with.

Figure 3. Histogram and boxplot of credit amount



3.3. Data preprocessing

3.3.1. Feature selections with IV

Features that are assumed to have great contribution to the predictive power are filtered using the IV technique. The IV value for each variable is presented in the following table:

Table1. IV values of variables

VAR_NAME	IV
age	0.146137
checking_status	0.666012
credit_amount	0.218303
credit_history	0.293234
duration	0.295094
employment	0.086434
existing_credits	0.013267

foreign_worker	0.043877
housing	0.083293
installment_commitment	0.026322
job	0.008763
num_dependents	0.000043
other_parties	0.032019
other_payment_plans	0.057615
own_telephone	0.006378
personal_status	0.044671
property_magnitude	0.112638
purpose	0.169195
residence_since	0.003589

It has been decided that 0.02 is the threshold used to opt out unimportant variables, which are ‘existing_credits’, ‘job’, ‘num_dependents’, ‘own_telephone’, ‘residence_since’.

3.3.2. Encoding categorical variables

Encoding is a crucial step in data preprocessing since it allows consistency representation of all variables. Meanwhile, most models are only capable of working with numeric input, thus mapping categorical values into numeric one should be done with considerate approach.

Within this research, Ordinal Encoder and Label Encoder are initially implemented for specific cases. While Label Encoder assigns a unique integer to each category in a categorical variable and is typically used when the categories have no inherent order or hierarchy, Ordinal Encoder assigns integers to categories based on their orders.

Accordingly, categorical features having specific orders including “checking_status”, “saving_status” and “employment” were encoded with Ordinal, the others with Label Encoder.

Besides, WOE encoder was also implemented for comparison purpose since it is known to be a commonly technique used in credit scoring. Compared to conventional encoding techniques, WOE has the advantage of being able to deal with missing values and outliers.

3.3.3. Data transformation

Standardizing data is a prevalent technique in data preprocessing, it rescales features to one same scale, improving performance of models. Two scaling methods are applied in this paper, the first one is Standard Scaler, which standardizes features by removing the mean (mean equals 0) and scaling to unit variance (equals 1) since it assumes that data follows the normal distribution. The other scaling method is

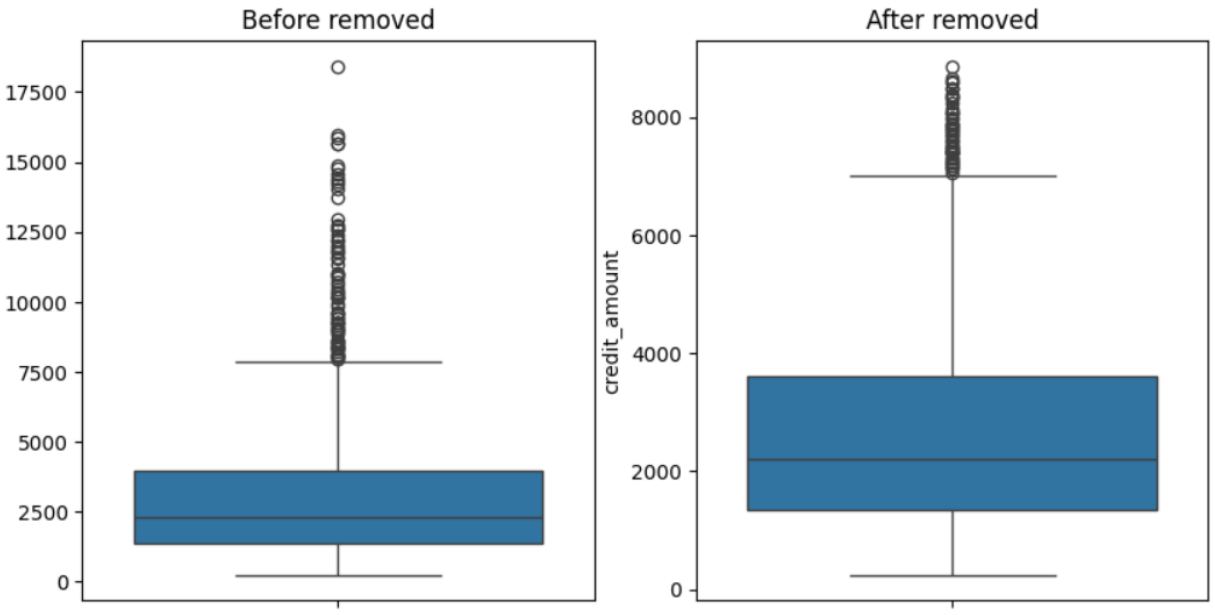
MinMax Scaler, which scales features to a range between 0 and 1. MinMax Scaler preserves the shape as well as the relationships between data points.

The above two scaling techniques are applied with all models in every test case possible with a view to conclude which technique is more likely to be suitable for the problem of credit risk prediction.

3.3.4. Outliers Removal

Manipulating outliers is an essential step in preparing data for credit risk modelling since outliers can significantly affect the performance of predictive models, leading to biased results. There are different techniques to handle outliers including trimming (remove them from the dataset) or winsorization (replacing extreme values with predefined percentile values). Outliers of the data used in this paper mostly occur in the "credit_amount" features, and we have removed those with z-score value larger than 2.

Figure 4. Boxplot of credit_amount before and after outliers removal



3.3.5. Rebalance data

Data imbalance is not something uncommon in reality, it occurs when the percentage of one class outweighs the other, especially in binary classification task. When the minority class is of higher interest, accuracy can be misleading due to the high dominance of the majority class. The same applies for the case when the majority class is of greater concern, accuracy might still be high even when the model performs poorly on the minority class.

A proposed approach to tackle this problem is to apply balancing data methods, one of which is Synthetic Minority Over-sampling Technique (SMOTE). SMOTE works by generating synthetic sample for the minority class, thus balancing the overall class distribution. To be more specific, individual instances from the minority class are randomly selected before k nearest neighbors of each are identified. New instances are then created along the line segment that connects the nearest neighbors. The whole process is repeated when the desired balance of both classes is achieved.

3.3.6. Modelling

The dataset after the preprocessing stage is split into a training and testing set with the ratio of 80/20. The number of rows in training set is 800, testing set has 200. The 7 models including : Logistic Regression, K-Nearest Neighbors, Decision Trees, LightGBM, AdaBoost, CatBoost, XGBoost are utilized on the training set.

IV. Findings and Discussion

4.1. Evaluation metrics

To measure the performance of each model, the following metrics are employed in this study : Accuracy, F1 score, Precision rate, Recall rate, and AUC value. These evaluation metrics are derived from the original four values of a confusion matrix, which is a tabular of detail breakdown of a model's prediction ability.

The four values constituting a confusion matrix are True Positive (TP) : the number of samples that were correctly classified as positive , True Negative (TN) : the number of samples that were correctly classified as negative, False Positive (FP): the number of samples that were negative but classified as positive (Type I error), and False Negative (NG): the number of samples that were positive but classified as negative (Type II error).

Accuracy has been a prevalent metric when it comes to assess the performance of a modal, it represents the ratio of correctly predicted instances to the total number of instances. However, considering the priority of identifying default customers in risk analysis, along with the case of normally observed imbalanced dataset, precision and recall should receive more attention.

Recall is defined as the ratio of TP over the sum of TP and FN, while precision represents the ratio of TP and TP and FP. Thus, there is likely to be a discrepancy between these two values, F1 score is the value that factors in both of them.

$$\text{F1} - \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

It can be inferred that a high F1 score indicates good performance in terms of minimizing false positives and false negatives. As a result, this study has chosen F1 score to be the main metric for evaluating the performance of models.

Besides, AUC-ROC values are also calculated for each model since it measures the ability to discriminate between default and non-default instances.

4.2. Results

Since this study focuses on identifying the more suitable techniques in terms of scaling and encoding methods in combination with different models, the following table only provides the metric results of the highest case for each model in terms of F1-score.

Table 2. Models, preprocessing techniques and metric results

model	remove_outlier	rebalance_data	encode_method	scale_method	accuracy	recall	f1_score	precision	roc_auc
Logistics	no	yes	woe	standard	0.76	0.779661	0.657143	0.567901	0.810314
KNN	no	yes	woe	standard	0.69	0.677966	0.56338	0.481928	0.726169
CatBoost	yes	yes	woe	standard	0.815	0.661017	0.678261	0.696429	0.829186
XGBoost	yes	no	ordinal	minmax	0.81	0.59322	0.648148	0.714286	0.848299
AdaBoost	yes	yes	ordinal	minmax	0.83	0.779661	0.730159	0.686567	0.832552
LightGBM	yes	yes	ordinal	minmax	0.825	0.59322	0.666667	0.76087	0.84313

Regard F1-score, AdaBoost stands out to own the highest value while XGBoost shows the largest ROC-AUC value. In terms of accuracy, AdaBoost still surpasses other model, meanwhile CatBoost makes the least false positive prediction. From an overall perspective, AdaBoost is one with least incorrect predictions.

For a more careful look at the difference of using different techniques, the given bar plot indicates a slight preference of models for using WOE for encoding method and MinMax Scaler for scaling data.

Figure 5. Encoding techniques used by each model

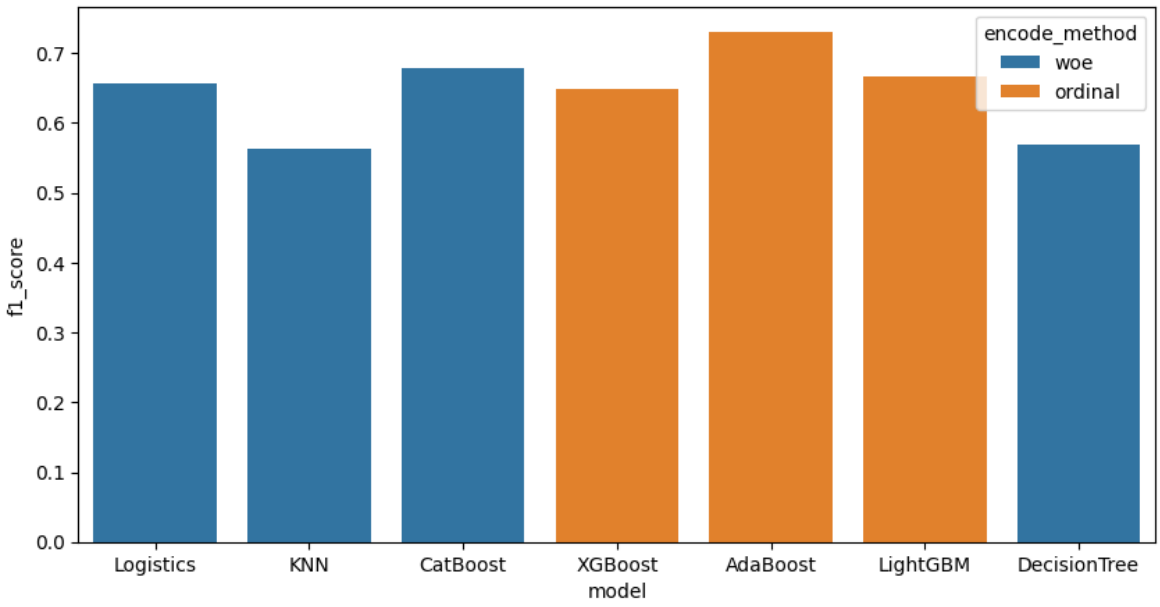
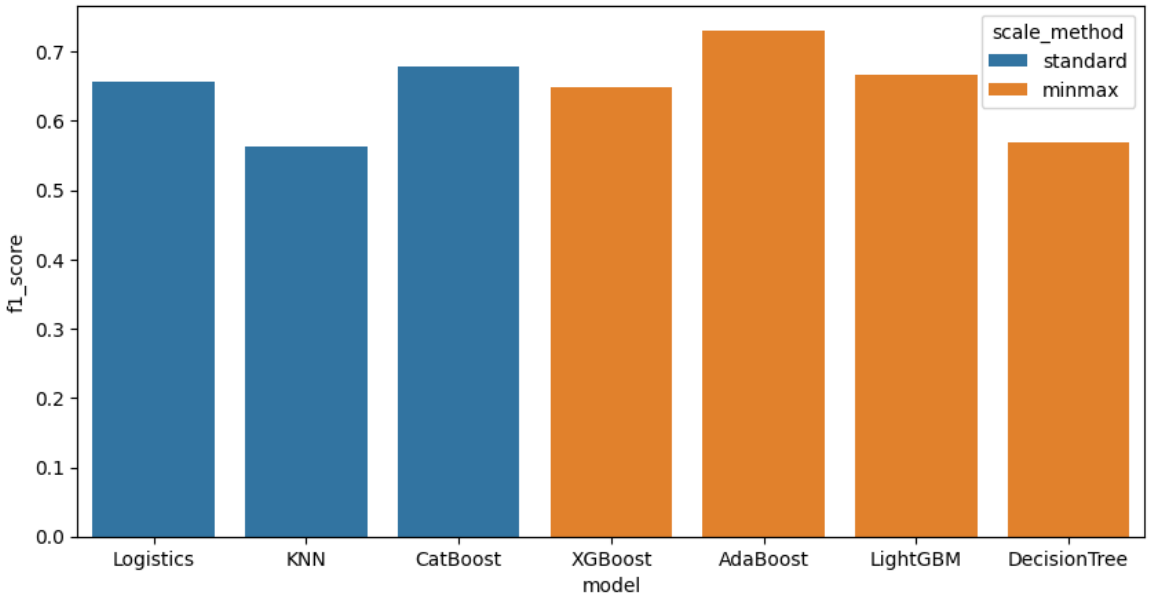


Figure 6. Scaling techniques used by each model



In addition, the below graphs also consolidate the role of rebalancing data as well as handling outliers, especially when we want to enhance the performance of model.

Figure 7. Models work well with rebalanced data

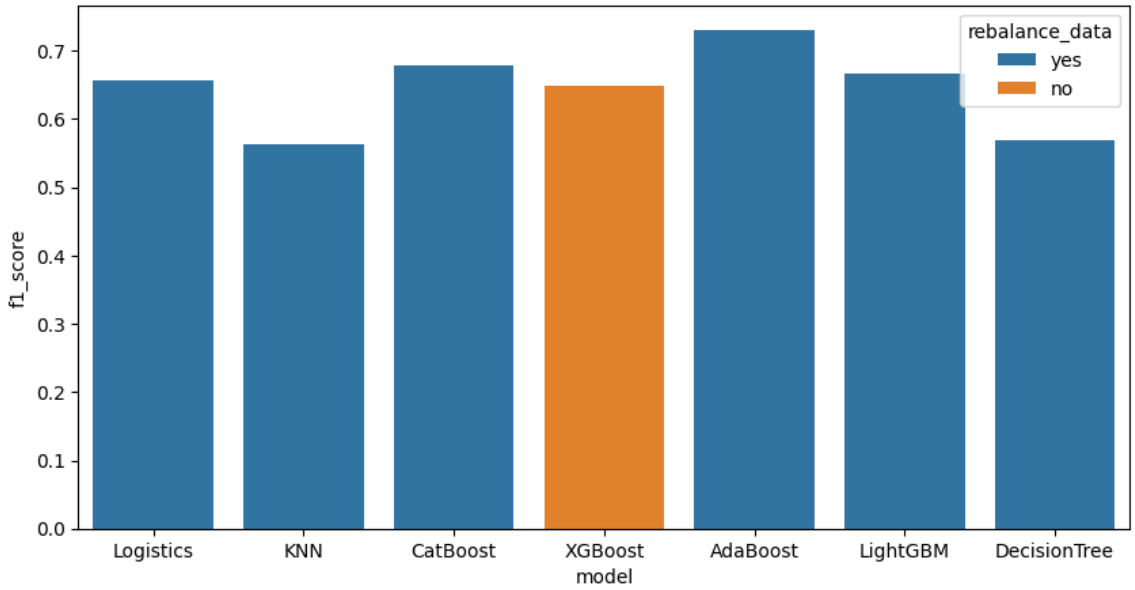
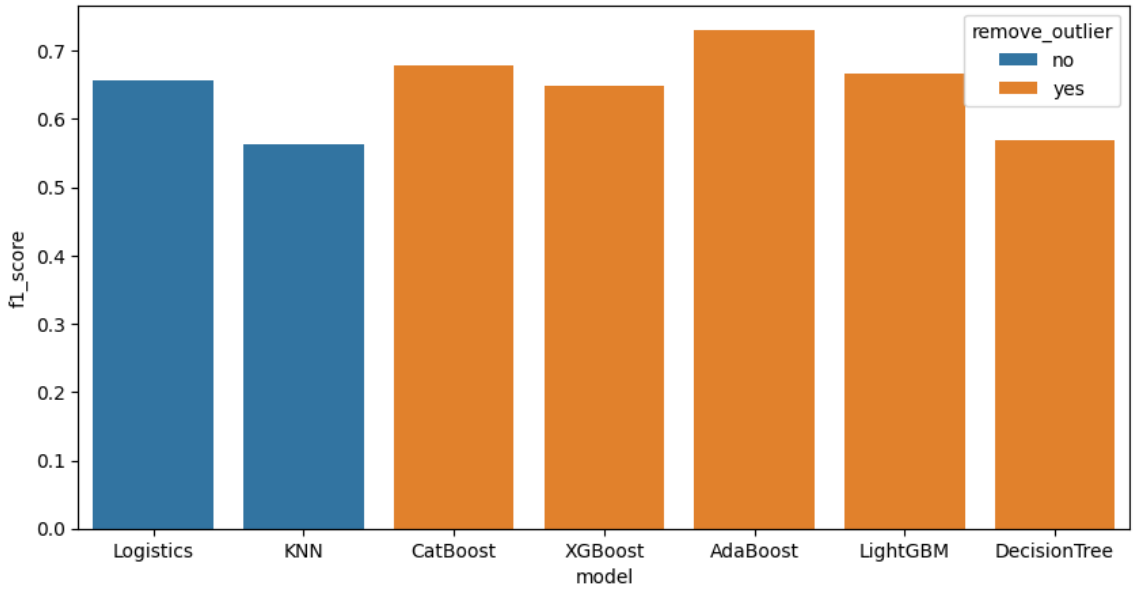


Figure 8. Models work well with extreme value-trimmed data



V. Conclusions

During this study, I have attempted to carry out a comprehensive analysis and comparison of how credit risk differentiates with a variety of machine learning models including Logistic Regression as the starting point before leveraging to other ensemble methods. The study is conducted on the German Credit dataset and the performance of each model is evaluated mostly based on F1 score. The results consolidate the necessity of meticulous and considerate selection of techniques applied during preprocessing stage, and the fact

that AdaBoost outperforms other models also highlight the power gradient boosting can bring about in credit risk analysis.

Further perspective to achieve higher performance as well as discover more meaningful insight in this field might involve the need of hyperparameter tunning and the use of advanced methods such as Deep Learning and Neural Networks. As observed from this study, the performance of machine learning models relies on the quality and relevance of features. Future exploration into feature selection and domain-specific feature engineering could pose potential improvements in the predictive powers.

In conclusion, this study underscores the effectiveness of various machine learning techniques in credit risk analysis, emphasizing the cruciality of methodical experiment and model interpretation. Incorporating advanced methods and refining conventional techniques will be essential to for the field to evolve.

References

Emmanuel, I., Sun, Y. & Wang, Z. A machine learning-based credit risk prediction engine system using a stacked classifier and a filter-based feature selection method. *J Big Data* **11**, 23 (2024).

Shi, S., Tse, R., Luo, W. *et al.* Machine learning-driven credit risk: a systemic review. *Neural Comput & Applic* **34**, 14327–14339 (2022).

Yuelin Wang, Yihan Zhang, Yan Lu, Xinran Yu, A Comparative Assessment of Credit Risk Model Based on Machine Learning ——a case study of bank loan data,Procedia Computer Science, Volume 174, 2020,Pages 141-149, SSN 1877-0509.

Lu Wang, Imbalanced credit risk prediction based on SMOTE and multi-kernel FCM improved by particle swarm optimization, Applied Soft Computing, Volume 114, 2022,108153,ISSN 1568-4946.

Coşkun, S. B. and Turanli, M.. "Credit risk analysis using boosting methods" Journal of Applied Mathematics, Statistics and Informatics, vol.19, no.1, 2023, pp.5-18.

Si, Z., Niu, H., & Wang, W. (2022). Credit Risk Assessment by a Comparison Application of Two Boosting Algorithms. *School of Economics and Management, University of Science and Technology Beijing*.

Zhang, T., & Li, B. (2018). Loan Prediction Model Based on AdaBoost and PSO-SVM. In *Advances in Intelligent Systems Research, volume 147, International Conference on Network, Communication, Computer Engineering (NCCE 2018)*. Department of Informatics, Beijing University of Technology, Beijing, 100124, China.

Chopra, A., & Bhilare, P. (2018). Application of Ensemble Models in Credit Scoring Models. *Business Perspectives and Research*, 6(2), 1–12. © 2018 K.J. Somaiya Institute of Management Studies and Research, SAGE Publications.

Baesens, Bart and Smedts, Kristien, Boosting Credit Risk Models (July 28, 2023). British Accounting Review, Forthcoming.

Wang, Y., Zhang, Y., Lu, Y., & Yu, X. (2020). A Comparative Assessment of Credit Risk Model Based on Machine Learning ——a case study of bank loan data. *Procedia Computer Science, 174*, 141-149.

J. G. Ponsam, S. V. J. Bella Gracia, G. Geetha, S. Karpaselvi and K. Nimala, "Credit Risk Analysis using LightGBM and a comparative study of popular algorithms," *2021 4th International Conference on Computing and Communications Technologies (ICCCT)*, Chennai, India, 2021, pp. 634-641

Sudhansu R. Lenka, Sukant Kishoro Bisoy, Rojalina Priyadarshini, Mangal Sain, "Empirical Analysis of Ensemble Learning for Imbalanced Credit Scoring Datasets: A

Systematic Review", *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 6584352, 18 pages, 2022.

Appendix

The code of this paper can be accessed here