Loan Default Prediction Report

Arne Lescrauwaet[1]

[1]852617312

January 2024

1 Introduction

For this project, I decided to use the HMEQ dataset to predict which customers may fail to repay their loan and which ones will successfully pay it back. Loan default prediction is a classification problem and involves using algorithms to classify potential borrowers into two groups: those at risk of default and those considered safe. Much of the literature on this topic focuses on a select subset of algorithms: Ensemble Methods, Logistic Regression, and Decision Trees [1, 3, 4]. I opted to compare some of these algorithms (Random Forest and Logistic Regression) with lesser-used algorithms (K Nearest Neighbor and Linear Discriminant Analysis). I also went beyond just looking at the metrics and incorporated interpretability in the comparison. The code accompanying this report can be found in the following GitHub repo.

2 Goal

The goal of this project is to compare the Random Forest, Logistic Regression, K Nearest Neighbor, and Linear Discriminant Analysis algorithms on the task of loan default prediction to see which algorithm is the best for automated loan default prediction. What constitutes 'best' can be a bit vague since many metrics can be compared. That's why I will compare the algorithms on accuracy, false negative count, and interpretability.
As a prerequisite to the actual training of the algorithms, we need to deal with imbalanced data. There are numerous ways of dealing with imbalances between classes: undersampling, oversampling, or assigning weights to name a few. To see which approach works best for the HMEQ dataset, I will conduct a swift comparison.

3 Data analysis

The dataset used in this project is the Home Equity dataset. This dataset contains loan performance information for 5,960 recent home equity loans.

It contains the following columns:

Column	Description
BAD	1: client defaulted on the loan, 0: loan repaid
LOAN	Amount of the loan request
MORTDUE	Amount due on existing mortgage
VALUE	Value of the current property
REASON	DebtCon: debt consolidation, HomeImp: home improvement
JOB	Six occupational categories
YOJ	Years at present job
DEROG	Number of major derogatory reports
DELINQ	Number of delinquent credit lines
CLAGE	Age of the oldest trade line in months
NINQ	Number of recent credit lines
CLNO	Number of credit lines
DEBTINC	Debt-to-income ratio

The "BAD" column is the target feature we try to predict.

3.1 General dataset information

The initial stage of data exploration involved a comprehensive examination of the dataset's characteristics. This dataset consists of 5,960 entries, encompassing 13 features as expected. A notable observation was the variation in feature scales and data types. Specifically, 'REASON' and 'JOB' were identified as object-type features, while the rest were numeric. A majority of the features, except 'BAD' and 'LOAN,' had missing values. 'DEBTINC', in particular, displayed a noticeably higher frequency of missing values, as illustrated in Figure 1. Additionally, it was observed that rows 3 and 1405 were largely empty, only populated with 'BAD' and 'LOAN' values.

3.2 Outliers

After the preliminary analysis, I proceeded to investigate potential outliers in the dataset. Notable outliers were present in several features, such as 'VALUE', 'CLAGE', and 'DEBTINC', as clearly shown in Figure 2. Closer examination suggested that these outliers were realistic and reflective of actual scenarios, rather than being anomalies or errors in data entry. Consequently, I opted to retain these outliers in their original form.

3.3 Correlation

Moving forward, I analyzed the interrelationships among features using a correlation matrix. As expected, a high correlation between 'MORTDUE' and 'VALUE' was observed, which is evident in Figure 3. Beyond this anticipated

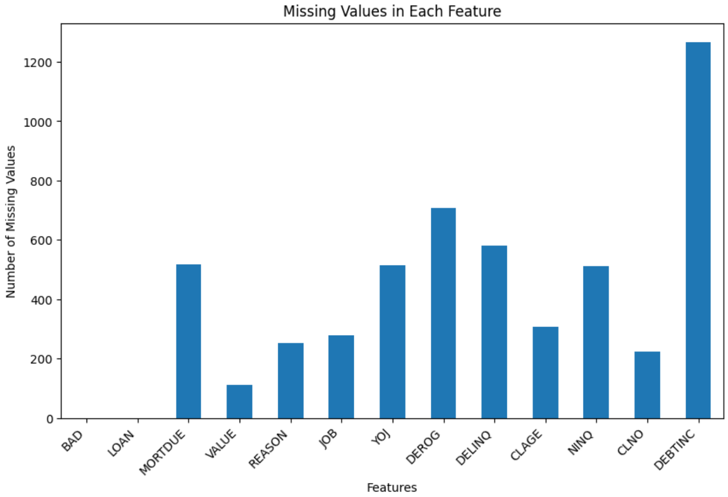


Figure 1: Missing values

correlation, there were no other notable correlations detected between the features. The feature that appears to have the strongest correlation with the target variable is ”DELINQ”.

3.4 Features

After evaluating the general characteristics of the dataset, I focused on a detailed examination of individual features. Recognizing that imbalanced data is a typical challenge in predicting loan defaults, I began by analyzing the distribution of the target feature, which confirmed an anticipated imbalance ratio of approximately 4:1. The dataset also included two object-type features, ’JOB’ and ’REASON’. I conducted an in-depth exploration of the categories within these features and their respective distribution patterns.

In the initial scenario, I encountered entries that lacked a job category yet had ’x’ years recorded for their current job. This was perceived as a data entry inconsistency. Therefore, in the data engineering phase, I intend to address these by assigning a ’missing’ label to the job category for such entries. Additionally, there were instances where entries had a specified job but lacked information on years of experience. This was also considered erroneous, especially given that previous data statistics indicated the lowest recorded experience was 0 years.
In the subsequent analysis, certain assumptions regarding the definitions of features were necessary. The attributes ’DEROG,’ ’DELINQ,’ and ’CLAGE’ have interrelated aspects. Notably, ’NINQ’ signifies recent credit lines, implying that the presence of recent credit lines (as indicated by ’NINQ’) necessitates the existence of general credit lines, as denoted by ’CLNO’. This was corroborated by the dataset, where no instances were observed with ’NINQ’ absent while

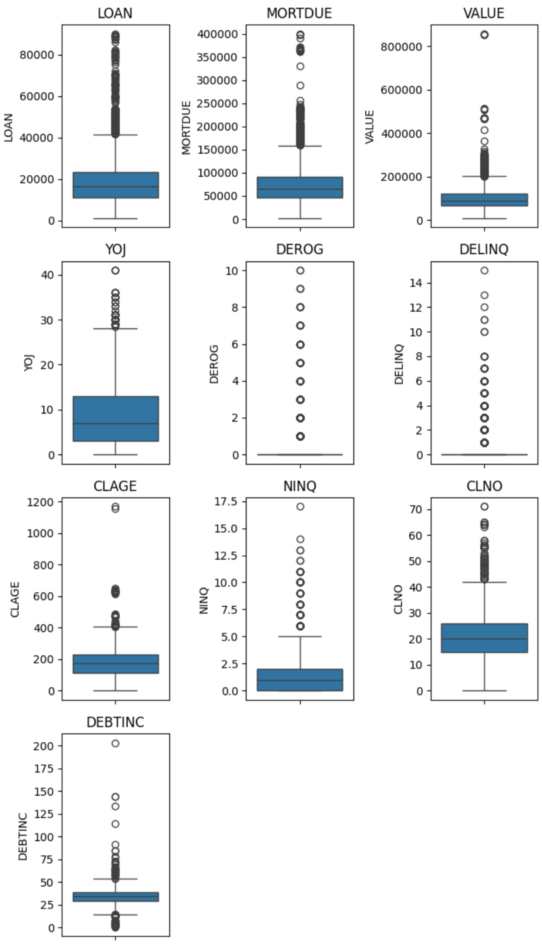


Figure 2: Outliers

'CLNO' was recorded. Conversely, it is feasible for an entry to have general credit lines ('CLNO') without recent ones ('NINQ'). The dataset doesn't clearly define what constitutes 'recent', so for this analysis, 12 months was used as the threshold. Therefore, in situations where the oldest credit line ('CLAGE') is less than 12 months old, this criterion was applied to impute any missing 'NINQ' values.

4 Feature Engineering

In my data analysis, it was observed that rows 3 and 1405 contained data only in the 'BAD' and 'LOAN' features, leading me to remove these rows. Further analysis revealed that 'JOB' and 'REASON' are categorical, prompting

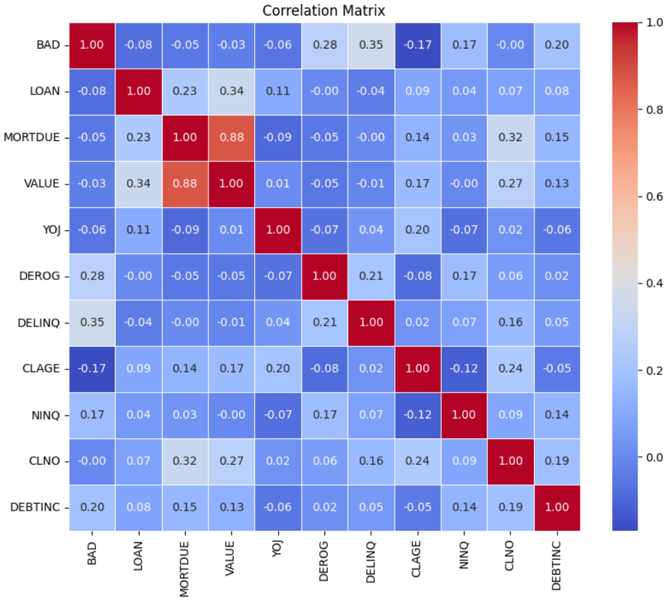


Figure 3: Correlations

me to create dummy variables for these categories. Additionally, to effectively handle null values without losing their informational value, I introduced a 'missing' category for imputation purposes. The correlation analysis highlighted a significant correlation (88%) between 'MORTDUE' and 'VALUE', suggesting redundancy in the information they provide. Consequently, I decided to eliminate the 'MORTDUE' feature from further analysis.

Before advancing with further data engineering processes, it was necessary to divide the data into Training and Testing sets, a crucial step to avoid information leakage. Accordingly, I adopted the conventional 80-20 split for the Train-Test division. Addressing the imputation needs, I began by filling in missing values for credit lines, particularly as discussed under the 'CLAGE' and 'DELINQ' sections of data analysis. The columns 'YOJ', 'VALUE', 'DEROG', 'DELINQ', 'CLAGE', 'NINQ', 'CLNO', and 'DEBTINC' were then imputed using sklearn's IterativeImputer. Given the varied scales across different features, standardization was imperative to ensure the efficacy of the algorithms, which are sensitive to feature scales. To achieve this, I employed the sklearn StandardScaler.

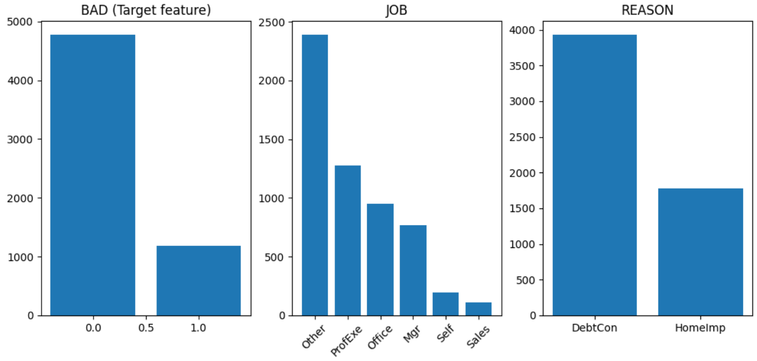


Figure 4: plot of BAD, JOB, and REASON

5 Methodology and Implementation

5.1 Imbalanced Data

As highlighted in the data analysis section, the challenge of imbalanced data is prevalent in loan default prediction. Various strategies exist to address this issue, including oversampling, undersampling, class weighting, or maintaining the imbalance. In the 'imbalance_comparison.ipynb' notebook, I experimented with these approaches by creating three distinct training sets. One utilized SMOTE for class balancing, another applied weights for the same purpose, and the third retained the existing class imbalance. Following this, I trained instances of Logistic Regression and Random Forest models, without fine-tuning, on each of these datasets. The performance of these models, in terms of training and testing scores, is detailed in Figure 5.

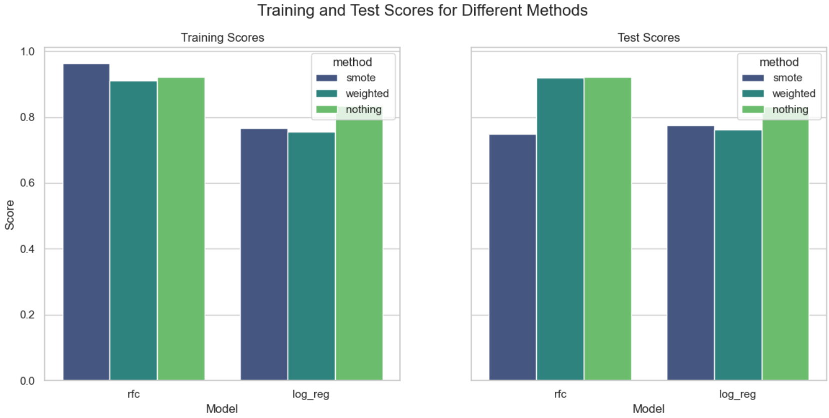


Figure 5: Comparison of imbalance approaches

The comparative analysis reveals that the SMOTE technique yields the best results during the training phase, which aligns with expectations as it simplifies the classification task by augmenting the minority class. However, this benefit did not carry over to the test set. Interestingly, the most effective method on the test set turned out to be maintaining the class imbalance, with Logistic Regression and Random Forest models performing optimally under this condition. Consequently, I have decided to employ this approach for the overall model comparison. While the weighted method was initially considered for a more comprehensive comparison, its practical implementation was limited due to the constraints of our selected algorithms, namely LDA and KNN, which are incompatible with class weight utilization.

5.2 Trainig

The models were trained in the following manner. First, a random search was done to narrow down the potential model parameters via sklearn's RandomizedSearchCV. After the random search, the parameter results were plotted, and the most promising parameters with slight deviations were used in a grid search via sklearn's GridSearchCV.

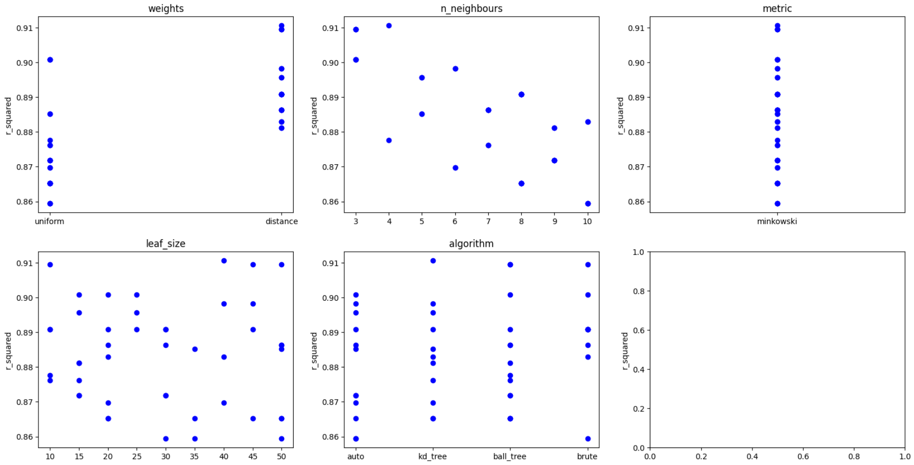


Figure 6: Example of random search plot

Optimal model parameters were determined using the 'best_params' attribute of the trained GridSearchCV object. Models fine-tuned with these parameters were subsequently saved using joblib for utilization in the evaluation phase. Notably, the KNN and Random Forest models exhibited superior performance on the training set, whereas the LDA model showed the least favorable results.

The parameter configuration for the models after the grid search can be seen in the Fine-Tuned Model Parameters table.

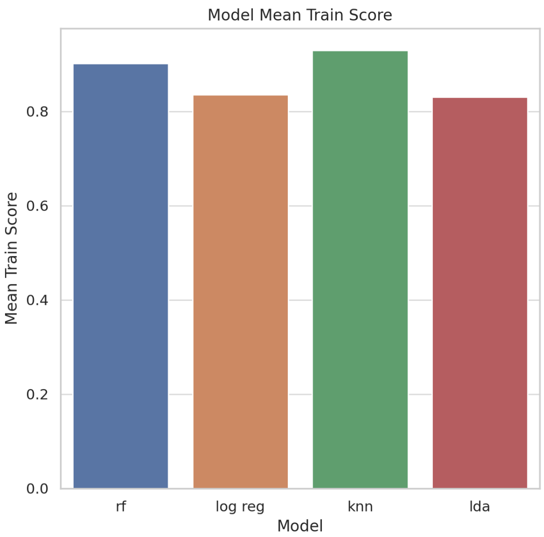


Figure 7: Model training scores

6 Evaluation and Results

The models were compared on their accuracy on the test set, the amount of false negative predictions, and their interpretability.

6.1 Accuracy

Accuracy is a metric that measures the overall correctness of a model's predictions by comparing the number of correctly classified instances to the total number of instances in the dataset. It is not the be-all end-all as discussed earlier, but it is still an important metric to get an overall idea of how well the model is predicting. The fine-tuned models were evaluated on the test set via the "clf.score(X_test, y_test)" function. The overall scores were quite good with the lowest being 83.1% from the Logistic Regression model and the highest 94.7% from the KNN model. Surprisingly the lesser-used KNN model scored better than the popular models such as logistic Regression and Random Forrest.

6.2 False Negatives

A false negative occurs in classification when the model incorrectly predicts a negative outcome (e.g., no default on a loan) when the actual outcome is positive (e.g., the client defaults on the loan). In this use case, the models mustn't predict that the client is going to repay their loan (BAD:0) if in actuality they will default (BAD:1), thus false negatives are of great interest.

We get the models' false negative scores by using the sklearn confusion matrix. Despite the accuracy of the models being fairly similar, there are rather

Model	Hyperparameter	Value
RFC	criterion	entropy
	max_depth	10
	max_features	sqrt
	max_leaf_nodes	414
	min_samples_leaf	1
	min_samples_split	2
	n_estimators	140
LDA	shrinkage	auto
	solver	lsqr
	store_covariance	True
KNN	algorithm	kd_tree
	leaf_size	40
	metric	minkowski
	n_neighbors	2
	weights	distance
Log_Reg	C	0.1
	dual	False
	penalty	l1
	solver	liblinear

Table 1: Fine-Tuned Model Parameters

larger discrepancies between their false negative counts. The KNN model scores best with only 60 false negative predictions, followed by the random forest model with 104 false negatives. The Logistic regression and LDA models respectively predicted 173 and 171 false negatives.

6.3 Interpretability

Model interpretability is the ability to interpret the decisions and predictions made by a machine learning model. This is an important part of machine learning when models are used in important decisions. Different models have different properties that make them more or less interpretable or explainable. Not only do we rely on the models themselves, but there are also frameworks such as LIME that help uncover a model's workings.

6.3.1 Internal properties

Random Forrest: The estimators used in the trained random forest model can be accessed via the "rfc.estimators_" attribute. This gives us access to the individual decision tree estimators allowing us to analyze the features it considers most important (they will be close to the root), and how they collectively contribute to the ensemble's predictions.

Linear Discriminant Analysis and Logistic Regression: Using the

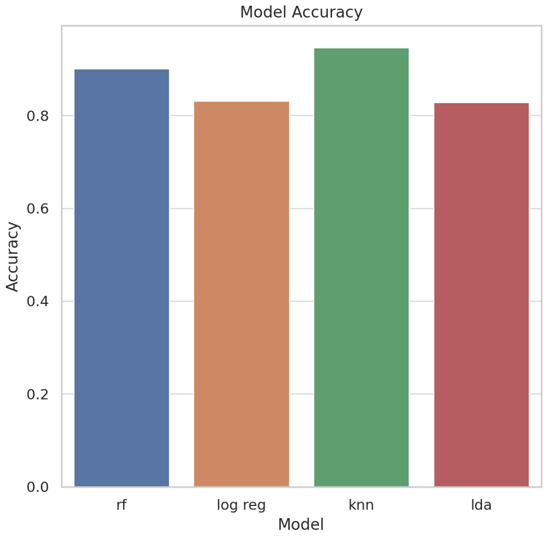


Figure 8: Evaluation Accuracy scores

".coef_" attribute of a trained model we can access the learned coefficients. Using these we can see how the features influence the predictions. The magnitude of the coefficient reflects its importance. The sign (positive/negative) indicates that an increase in the corresponding feature value contributes to a higher/ lower likelihood of the data point belonging to a specific class.

K nearest neighbors: KNN is a distance-based algorithm thus we don't get any insights into the contribution of the features to the predictions.

6.3.2 LIME

The Local Interpretable Model-agnostic Explanations (LIME) [2] framework is a technique used to interpret the predictions of machine learning models. Via LIME we get insights into the importance of different features for the specific prediction. We can easily see which features are important for the "will default" or "will repay the loan" prediction and the overall probability of predicting one class or the other.

In Figure 10 we can see the most important features and their magnitude for all of the algorithms. The features with the blue background are important for the "loan repaid" prediction and the features with an orange background are important for the "will default" prediction. We see that the "DELINQ" feature is an important factor when predicting who will default on their loan which is no surprise since it had the highest correlation (35%) with the target variable.

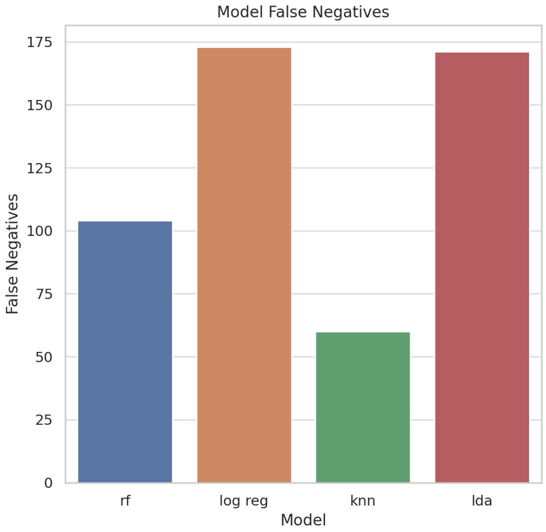
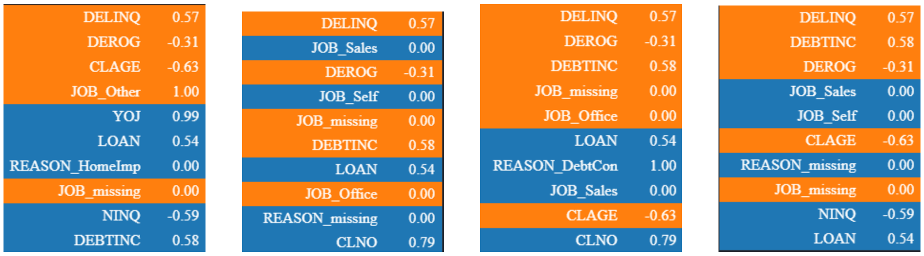


Figure 9: False Negative scores



DELINQ	0.57
DEROG	-0.31
CLAGE	-0.63
JOB_Other	1.00
YOJ	0.99
LOAN	0.54
REASON_HomeImp	0.00
JOB_missing	0.00
NINQ	-0.59
DEBTINC	0.58

(a) Lime KNN

DELINQ	0.57
JOB_Sales	0.00
DEROG	-0.31
JOB_Self	0.00
JOB_missing	0.00
DEBTINC	0.58
LOAN	0.54
JOB_Office	0.00
REASON_missing	0.00
CLNO	0.79

(b) Lime LDA

DELINQ	0.57
DEROG	-0.31
DEBTINC	0.58
JOB_missing	0.00
JOB_Office	0.00
LOAN	0.54
REASON_DebtCon	1.00
JOB_Sales	0.00
CLAGE	-0.63
CLNO	0.79

(c) Lime Log Reg

DELINQ	0.57
DEBTINC	0.58
DEROG	-0.31
JOB_Sales	0.00
JOB_Self	0.00
CLAGE	-0.63
REASON_missing	0.00
JOB_missing	0.00
NINQ	-0.59
LOAN	0.54

(d) Lime RFC

Figure 10: Lime feature importance

6.3.3 Interpretation of the results

In addressing class imbalances, there is no one-size-fits-all solution. While SMOTE or its variants are commonly utilized for loan default prediction, this project found that disregarding the imbalance yielded the most favorable results. The K Nearest Neighbor algorithm outperformed more established methods like Logistic Regression and Random Forest in both accuracy and false negative count. In terms of interpretability, Linear Discriminant Analysis and Logistic Regression stood out due to their coefficient analysis capabilities. However, less interpretable methods like Random Forest and KNN were brought to a comparable level of interpretability through the application of the LIME framework, which provided a way to interpret all models based on feature importance, irrespective of their inherent methodologies.

7 Conclusion and Discussion

Predicting loan defaults effectively is a critical challenge in the financial industry. This issue holds significant importance as inaccurate predictions can lead to considerable risks for banking institutions. It's essential to identify the most effective models that can accurately distinguish between potential defaulters and reliable borrowers. In the realm of finance, the ideal model surpasses mere accuracy or low false positive rates. Regulatory demands necessitate that such models be not only accurate but also interpretable, as they play a pivotal role in crucial decision-making processes. Given the inherent imbalance often found in loan default datasets, various methods like SMOTE and its variants have been employed to address this imbalance. However, there's no one-size-fits-all solution in this domain.

In the case of this project leaving the class imbalance as is seemed to be the best approach after having compared this method to SMOTE and class weighting. In loan default prediction Logistic regression is a popular model because of its interpretability and Random Forest models and other ensemble techniques are popular because of their performance. But on this dataset, the less common model KNN beat both of these in accuracy score and false negative count. Although KNN is inherently less interpretable due to its reliance on distance measurements, integrating it with the LIME framework allows for a better interpretation of its predictions through the analysis of feature importance.

In contexts where automated decision-making impacts humans, the model must remain unbiased toward specific groups. Typically, biases emerge from data containing identifiable features like race, religion, or country of origin. In our study, this concern was not applicable as our dataset did not include such sensitive attributes.

7.1 Limitations

In the data engineering step, some assumptions were made about how the data was gathered and about some standards i.e. the definition of recent. Ideally, we would have more information about the data-gathering process but that's not always feasible for online datasets. In the assessment only a small subset of machine learning models were used thus we can't confidently say that the best-performing model found is the best model for this task in general. For future works, it could be interesting to explore the different data balancing approaches further and compare them across datasets.

References

[1] Lili Lai. "Loan Default Prediction with Machine Learning Techniques". In: *2020 International Conference on Computer Communication and Network Security (CCNS)*. 2020, pp. 5–9. DOI: 10.1109/CCNS50731.2020.00009.

[2] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “”Why Should I Trust You?”: Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. 2016, pp. 1135–1144.

[3] Junhui Xu, Zekai Lu, and Ying Xie. “Loan default prediction of Chinese P2P market: a machine learning methodology”. In: *Scientific Reports* 11.1 (2021), p. 18759.

[4] Jing Zhou et al. “Default prediction in P2P lending from high-dimensional data based on machine learning”. In: *Physica A: Statistical Mechanics and its Applications* 534 (2019), p. 122370.