Loan Default Prediction Report

Arne Lescrauwaet[1]

[1]852617312

January 2024

1 Introduction

For the 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 prediction 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 algorithm 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 prediction to see which algorithm is the best for 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 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

I started the data exploration process by examining the overall characteristics of the dataset. It comprises 5,930 entries and includes the anticipated 13 features. Notably, the feature values appeared to be on diverse scales. Among the features, two were of the object type, specifically 'REASON' and 'JOB,' while the remaining features were of numeric types. Most features, excluding 'BAD' and 'LOAN,' exhibited missing values, with 'DEBTINC' having a significantly higher count of missing values compared to others which is visible in Figure 1. Additionally, rows indexed at 3 and 1405 were mostly empty, containing only 'BAD' and 'LOAN' values.

3.2 Outliers

Following the initial analysis, I examined potential outliers within the dataset. Several features, including 'VALUE,' 'CLAGE,' and 'DEBTINC,' exhibited noteworthy outliers as is evident in Figure 2. However, upon closer inspection, these outliers seemed reasonable, representative of real-world situations, and not indicative of errors in the data entry process, thus I decided to leave them as they were.

3.3 Correlation

I proceeded to examine the correlations, utilizing a correlation matrix to assess the relationships between the features. It was anticipated that 'MORTDUE' and 'VALUE' would exhibit a high correlation, and indeed, they did as can

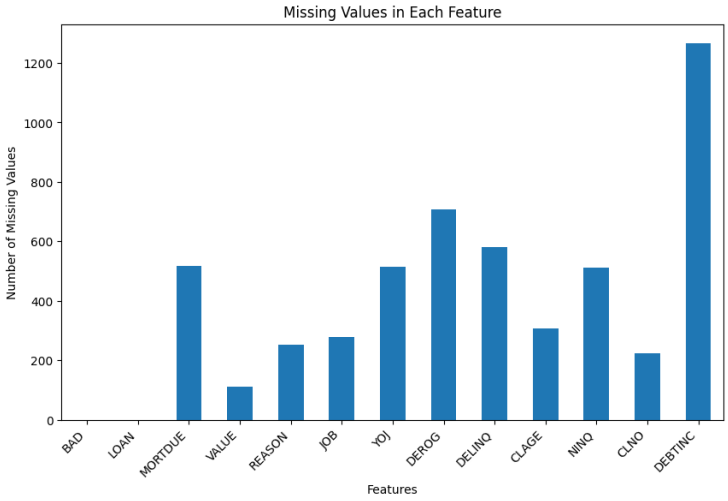

Figure 1: Missing values

be seen in Figure 3. However, aside from this expected correlation, no other significant correlations were observed among the features.

3.4 Features

Following the analysis of the overall dataset characteristics, I delved into the specifics of individual features. Since a common obstacle in loan default prediction is imbalanced data, I first looked at the distribution of the target feature, confirming the expected imbalance of roughly 4:1. Additionally, the dataset contained two features of object type, namely 'JOB' and 'REASON'. For these features, I further investigated their categories and respective distributions.

In the first case, there were entries without a job category but who had 'x' years at their current job. I interpreted this as a data entry error. Thus, in the data engineering section, I plan to impute these with a job category labeled 'missing'. Some entries had a job but were missing their years of experience, which also seemed like an error since earlier statistics of that feature revealed the minimum value was 0 years of experience.
For the following section, there were some assumptions made regarding definitions. The features 'DEROG,' 'DELINQ,' and 'CLAGE' are related. Specifically, the 'NINQ' feature denotes recent credit lines, indicating that an entry cannot possess recent credit lines without also having general credit lines, as represented by the 'CLNO' feature. This observation aligns with the dataset, as no entries were found where 'NINQ' was missing while 'CLNO' was present. Conversely, it is possible for an entry to lack recent credit lines ('NINQ') but still have general credit lines ('CLNO'). The definition of 'recent' is not explicitly provided in the dataset source; however, for analytical purposes, a cut-off of 12

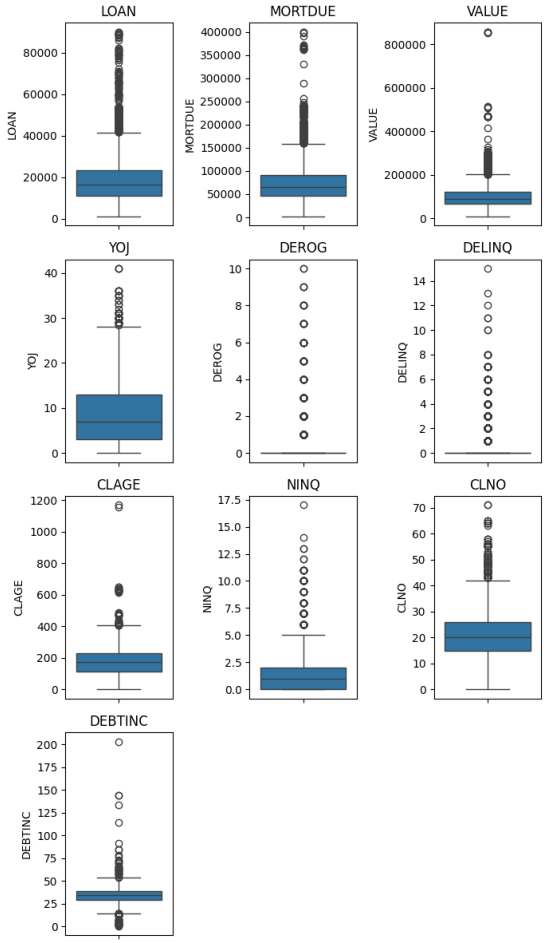

Figure 2: Outliers

months is adopted for 'recent.' In cases where the oldest credit line ('CLAGE') is younger than 12 months, we can utilize this information to impute potential missing values for 'NINQ'.

4 Feature Engineering

During the data analysis, I found that rows 3 and 1405 were empty except for the 'BAD' and 'LOAN' features, so I decided to drop them. We saw that JOB and REASON are categorical features, so I'll first make dummy variables for these. I'll also add a new category 'missing' to be used as an imputation for the null values so we don't lose the information of them being null. In the correlation section of the data analysis, it appeared that the 'MORTDUE' and

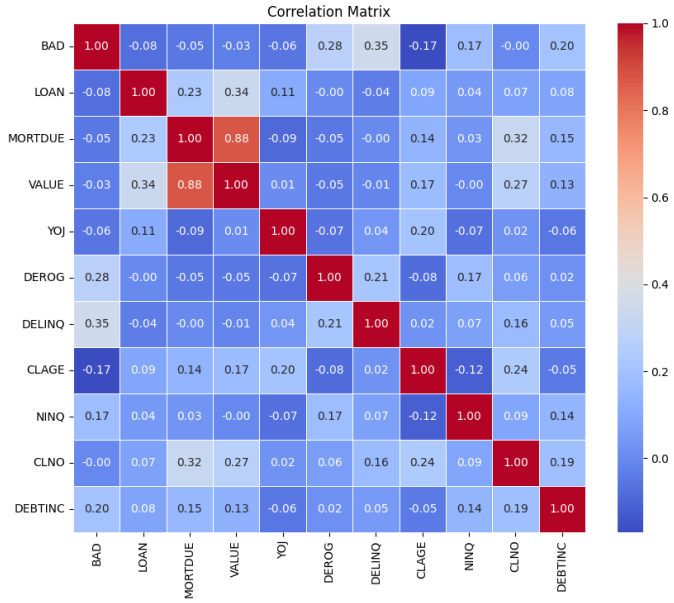

Figure 3: Correlations

'VALUE' features have a high correlation (88%) meaning that they produce a lot of redundant information, so I'll drop 'MORTDUE'.

Before moving on to the next data engineering steps, the data needs to be split into Train and Test sets to prevent information leakage. I opted for the standard 80-20 Train-Test split. For the imputations, I started with imputing the missing credit lines as mentioned in the data analysis section about CLAGE, DELINQ. The following columns were imputed with the sklearn IterativeImputer: 'YOJ', 'VALUE', 'DEROG', 'DELINQ', 'CLAGE', 'NINQ', 'CLNO', 'DEBTINC'. Since the features have different scales, they need to be standardized because we are working with algorithms that are affected by the scale of the features. For this, I used the sklearn StandardScaler.

5 Methodology and Implementation

5.1 Imbalanced Data

As mentioned in the data analysis section, loan default prediction usually suffers from imbalanced data. There are several ways of dealing with imbalanced data, some of the more common approaches are oversampling, undersampling, assigning weights to classes, or leaving the classes imbalanced. I tried some

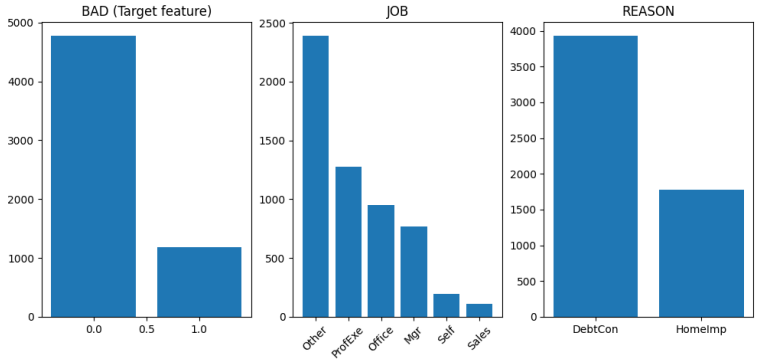

Figure 4: plot of BAD, JOB, and REASON

of these techniques in the 'imbalance_comparison.ipynb' notebook, where I created three different training sets. One set used SMOTE to balance the classes, another used weights to balance the classes, and the third left the classes imbalanced. I then trained three instances of non-fine-tuned Logistic Regression and Random Forest models on these datasets and compared their train and test scores as can be seen in Figure 5.

Figure 5: Comparison of imbalance approaches

From the comparisons, we see that the SMOTE approach performs best in training, which is to be expected since we essentially make the classification problem easier by providing more examples of the other class. However, this advantage does not seem to translate to the test set. The clear winner on the test set appears to be leaving the classes imbalanced, which was not what I expected. Thus, this is the approach I will use when comparing all the models. For the sake of thoroughness, the weighted method was included in the sub-comparison.

However, its application in the actual comparison was not possible, as our chosen algorithms, specifically LDA and KNN, do not support the use of class weights.

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.

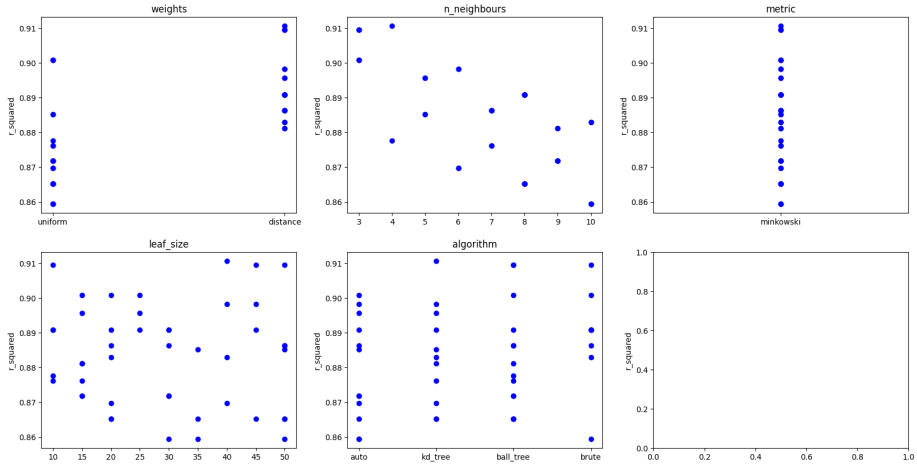

Figure 6: Example of random grid search plot

The final model parameters were found via the best_params method on the trained GridSearchCV object. The models with the corresponding parameters were then saved via joblib to be used in the evaluation section. The KNN and Random Forest models scored best on the training set, and the LDA model scored the worst.

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

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

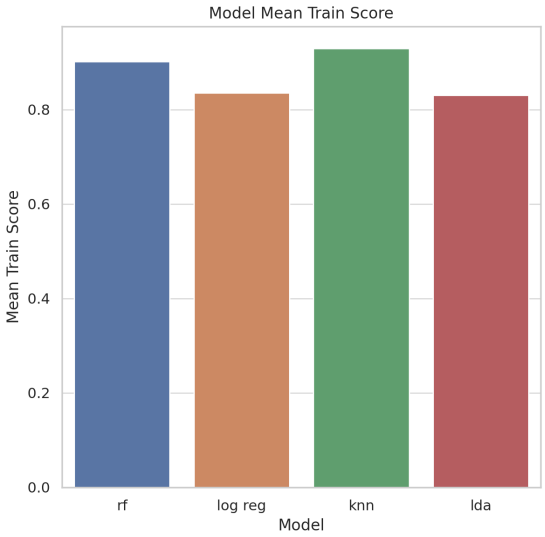

Figure 7: Model training scores

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, it is important that the models don'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 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.

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

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 his 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 ".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

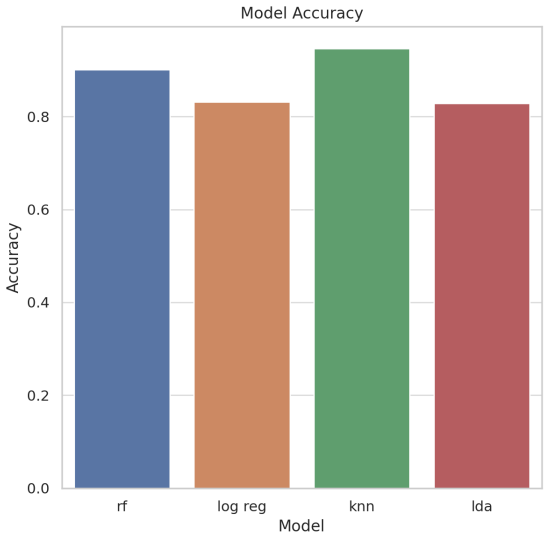

Figure 8: Evaluation Accuracy scores

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.

6.3.3 Interpretation of the results

There is no one default method when dealing with class imbalances, SMOTE or a variant of SMOTE is often used in loan default prediction but for this project, it turned out the best approach was ignoring the imbalance. The K nearest-neighbor algorithm scored best in both accuracy and false negative count beating the more popular logistic regression and ensemble techniques such as Random Forest. The Linear discriminant Analysis and Logistic Regression techniques had the best interpretability because of the ability to analyze their coefficients. Despite methods such as Random Forest and KNN being less interpretable, the LIME framework evens the playing field by allowing all models to have a certain degree of interpretability via feature importance regardless of their intrinsic methods.

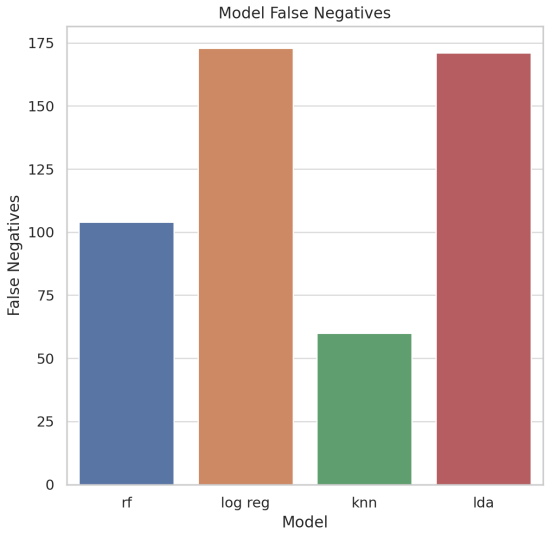

Figure 9: False Negative scores

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 some standards. 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.