Lab Assignment 1: Classification

Pablo Rodríguez, Ángel Visedo and José Carlos Riego

Machine Learning I - MSc in Big Data

October 2024

Abstract

This study applies machine learning to predict repayment behavior for Home Equity Lines of Credit (HELOCs). Using a dataset of credit history features, we have developed and compared several classification models, including Logistic Regression, tree-based models, and neural networks. After data preprocessing and analysis, the Multi-Layer Perceptron (MLP) outperformed others, achieving the highest F1-score and balancing high- and low-risk borrower identification. While simpler models performed well, the MLP's accuracy justified its selection. This study emphasizes the importance of feature selection and hyperparameter tuning, advocating for these models as tools to support, not replace, human judgment in lending decisions.

1 Introduction

Predicting whether borrowers will repay their loans is essential for financial institutions, especially when dealing with Home Equity Lines of Credit (HELOCs).

A HELOC is a form of secured credit that allows homeowners to borrow against the equity in their homes, providing a flexible credit line for large expenditures or consolidating higher-interest debt, such as credit card balances. HELOCs often come with lower interest rates than other loans, and the interest might be tax-deductible in some cases.[2]

The dataset used in this project provides a variety of features related to borrowers' credit histories, including risk estimates, credit utilization, and trade histories. These factors offer valuable insights into the financial habits of applicants and help predict their ability to repay loans. This analysis will begin with some Data Preprocessing, including a wide Exploratory Data Analysis (EDA) to understand the data, followed by the application of several classification models. These models will be evaluated an compared based on their key performance metrics such as accuracy, precision, recall, and F1-score, among others.

Additionally, the primary objective of this study is to gain a deeper understanding of how these models function and address problems from a financial perspective. The focus is on assessing the practical applicability of these models in predicting financial risk, providing valuable insights for their use in real-world lending decisions.

2 Dataset Overview

2.1 FICO Score

The FICO Score is a three-digit number based on the information in your credit reports. It helps lenders determine how likely you are to repay a loan. This, in turn, affects how much you can borrow, how many months you have to repay, and how much it will cost (the interest rate).[1]

In our case, we will focus on the likelihood of repayment within a specified timeframe. Understanding the underlying factors that contribute to a FICO Score is crucial for predicting loan repayment behavior.

2.2 Description of the Dataset

The variables considered for the analysis are as follows:

Inputs:

1. **ExternalRiskEstimate**: Measure of borrower's riskiness from external data sources.
2. **NetFractionRevolvingBurden**: Proportion of current credit usage compared to maximum allowed credit.
3. **AverageMInFile**: Average duration, in months, of trades in a borrower's credit file.
4. **MSinceOldestTradeOpen**: Age, in months, of the borrower's oldest credit account.
5. **PercentInstallTrades**: Percentage of credit accounts with fixed payment terms.
6. **NumSatisfactoryTrades**: Count of trades where obligations were met satisfactorily.
7. **NumTotalTrades**: Total number of credit accounts.
8. **MSinceMostRecentInqexcl7days**: Months since the last credit inquiry, excluding the past week.
9. **PercentTradesNeverDelq**: Percentage of trades with no delinquency history.

Output:

10. **Risk Performance**: Flag indicating whether the borrower paid as negotiated (12-36 months). Class variable (0 = pays, 1 = does not pay).

2.3 Special Characters

The dataset contains some special characters which correspond to the following situations:

- **-9**: No Bureau Record or No Investigation
- **-8**: No Usable/Valid Trades or Inquiries
- **-7**: Condition not Met (e.g. No Inquiries, No Delinquencies)

We have determined that these values do not provide significant information for our models. Therefore, we have omitted any clients exhibiting one of these special situations. Given the sufficiently large size of our dataset, we can afford to exclude these entries without compromising the integrity of our analysis.

3 Data preprocessing

After importing the data, we have conducted its preprocessing, modifying various relevant parameters to train the models with the highest quality data possible.

3.1 Handling Missing Values

There are missing values in our dataset, and its proportion in `RiskPerformance` (output) is around 30 %. We have tried to train models that support missing values, such as Decision Tree and Random Forest, with them and, although we see an improvement for these two models, they lag behind other models. Therefore, we have opted to discard those data with missing values when training the models.

3.2 Identification of Outliers

Through the analysis of histograms, Q-Q plots, and box-plots of the variables, we can conclude that although outliers seem to be present, two factors would justify considering them.

First, in certain cases (e.g., 1), these outliers seem to simply fall within the tail of the skewed distribution, making it unreasonable to exclude them, as they would provide valuable insights into the phenomenon under study.

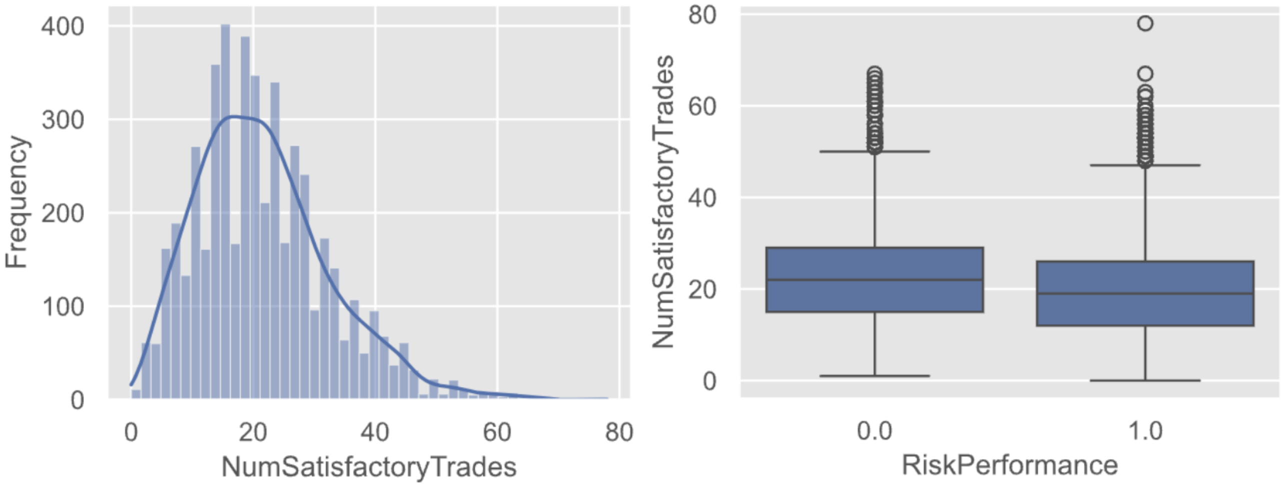

Figure 1: Histogram and boxplot of `NumSatisfactoryTrades`.

Second, there are other points that appear beyond the extremes of the box (more than 3 standard deviations away from the mean) because most of the data is concentrated within a narrow range of the variable (e.g., 2). However, since we cannot conclusively considerate them scientifically invalid, we have also chosen to include them in our analysis.

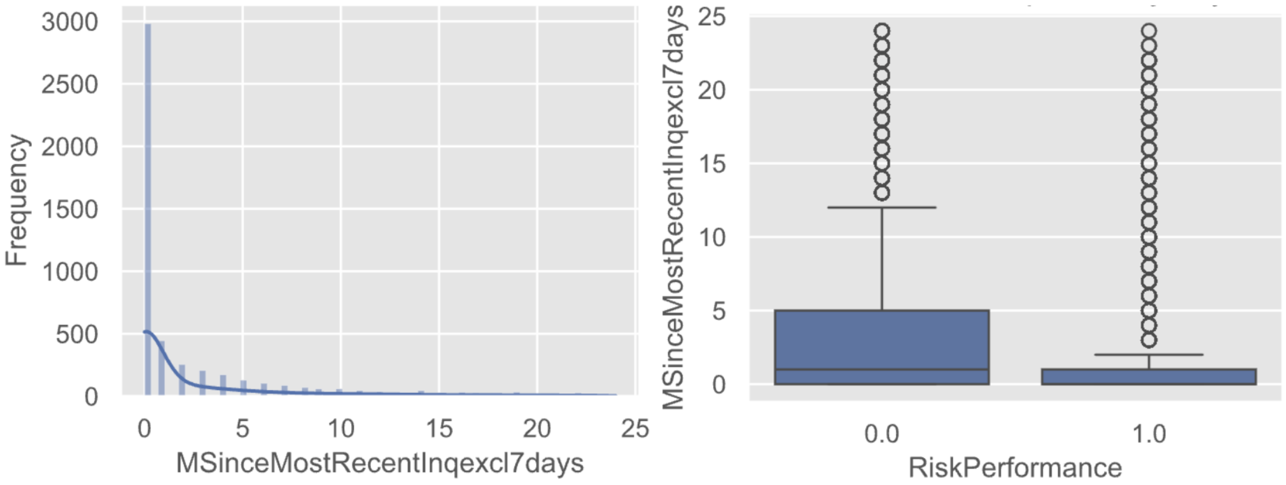

Figure 2: Histogram and boxplot of `MSinceMostRecentInqexcl7days`.

3.3 Exploratory Data Analysis (EDA)

3.3.1 Skewness transformation

Given that some of the input variables exhibit an asymmetric distribution (e.g. `NumTotalTrades`, `PercentTradesNeverDelq`), we have attempted to apply Box-Cox transformations to make the distribution of the variables as normal as possible. However, after observing no improvements in the results after several trials, we decided to discard these modifications.

3.3.2 Collinearity

To analyze the collinearity between variables, we have plotted the correlation matrix (Figure 3), from which we have observed the following significant correlations:

- `NumSatisfactoryTrades` with `NumTotalTrades` (0,93)
- `AverageMinFile` with `MSinceOldestTradeOpen` (0,74)
- `ExternalRiskEstimate` with `NetFractionRevolvingBurden` (-0,62)
- `NetFractionRevolvingBurden` with `PercentTradesWBalance` (0,60)

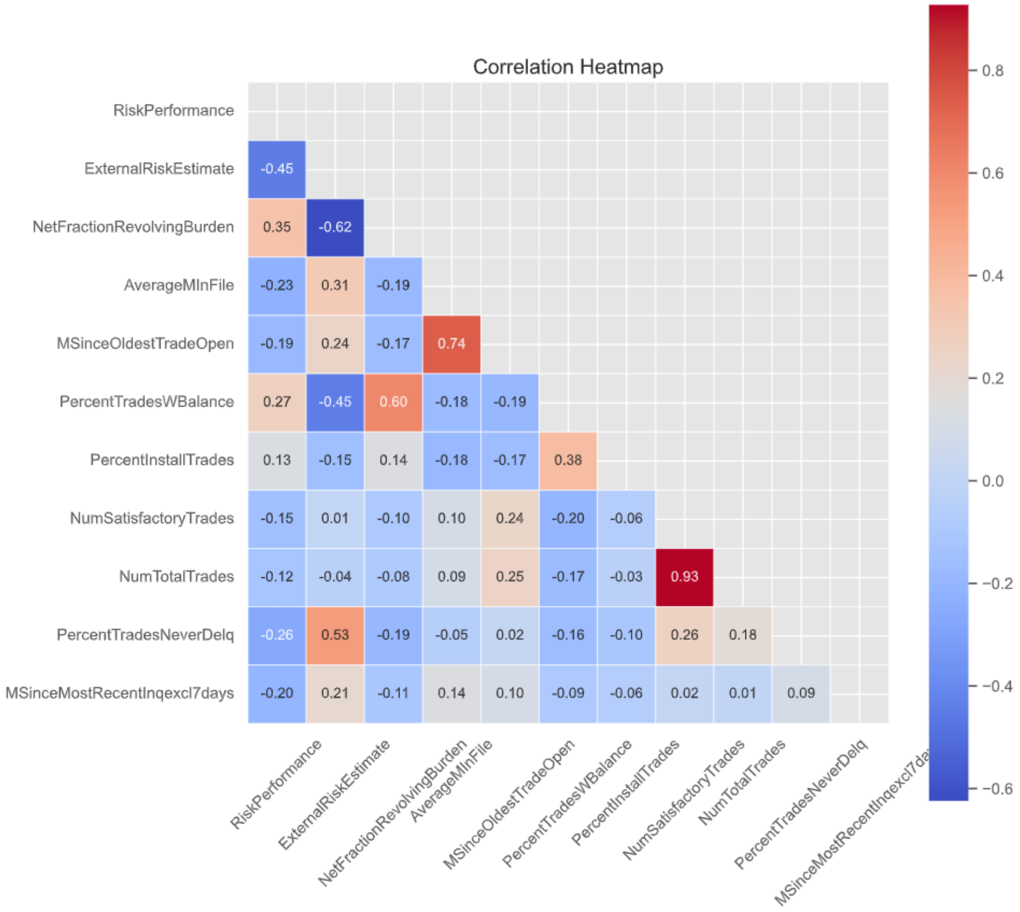

Figure 3: Correlation matrix of predictors.

3.3.3 Pre-training feature importance

Before training the models, we considered it essential to study the importance of each variable using certain statistical methods, with the objective to reduce and take only the important ones. This helps us understand how sensitive the output is to different input variables. Below, we highlight some of the techniques used to evaluate feature importance during preprocessing, including the Chi-square test and PCA.

3.3.4 Chi-Square Test

The Chi-Square test helps in feature selection by assessing the association between categorical variables. It compares observed and expected frequencies, and if the p-value is below a certain threshold (0.05), it suggests a significant relationship in the input data.

3.3.5 PCA (Principal Component Analysis)

Principal Component Analysis (PCA) reduces dimensionality by transforming original variables into new, uncorrelated components. The first few components capture the most variance, helping to focus on the features that hold the most information while minimizing noise.

3.3.6 Final variable selection

After training the models with different combinations of variables, and based on the feature importance metrics previously discussed and the MLP model feature selection method (see 5.2), we have chosen to retain all inputs except `MSinceOldestTradeOpen` and `NumTotalTrades`, mainly because of the low sensitivity and the strong correlation they have with other inputs. Chi-square test and PCA were not taken into account in the end as they did not provide any extra information.

These variables enhance the F1 score (which also has been set as the parameter to maximize in cross-validation) and the sensitivy (TPR), which are the main metrics we have selected to focus on, due to the reasons below.

First, the F1 score is a sort of average between sensitivity (TPR) and specificity (TNR), providing a balance between these two metrics. A high F1 score, therefore, indicates both a high TPR and TNR, which implies minimizing the number of clients who are granted credit when they should not be, as well as the number of clients who are denied credit when they should receive it.

Second, sensitivity (TPR) focuses on indicating the proportion of clients who were classified as low risk for repayment when they should have been classified as high risk (FN). Paying attention to this metric is crucial, as this scenario poses the greatest financial risk to the HELOC credit line.

3.4 Class imbalance and Data splitting

After completing all the data preprocessing, which involved removing missing values and excluding clients with special circumstances (see section 2.3), we assessed whether there was any class imbalance. The output distribution (`RiskPerformance`) is approximately 51% - 49%, indicating that there is no significant imbalance.

Regarding the dataset split into training and test sets, we decided on an 80 - 20 split, opting not to use a validation set, as we did not consider it necessary given the size of the dataset.

4 Model Selection and Fitting

In this section we will briefly describe the models used, as well as the hyperparameters that have been adjusted, choosing the ones that maximize the F1-score in cross-validation, as commented previously.

4.1 Logistic Regression (LR)

A classification model that uses a sigmoid function to estimate the probability that an instance belongs to one of two classes, based on a linear combination of the input features.

This model does not contain hyperparameters to be adjusted.

4.2 K-Nearest Neighbors (kNN)

A classification algorithm that assigns a class to an instance based on the majority classes of its nearest neighbors in the feature space.

The hyperparameter of this model considered for training is `knn__n_neighbors`, with an optimal value of 71 neighbors.

4.3 Decision Tree (DT)

A Decision Tree splits the data into subsets based on feature values to make predictions. It represents decisions in a tree-like structure, where each node corresponds to a feature, each branch represents a decision rule, and each leaf node represents an outcome or class label.

The hyperparameter optimized for this model is `DT__min_impurity_decrease = 0.0045`, which refers to the minimum impurity (in our case, the Gini) that must decrease at each split or node.

4.4 Random Forest (RF)

This model combines multiple decision trees by voting to improve accuracy and reduce overfitting. Each tree is trained with a random subset of the data and features.[3]

In this case, the best hyperparameters are the following:

- `RFmax_depth = 5`
- `RFn_estimators = 10`

where `RFmax_depth` refers to the maximum depth of each tree, and `RFn_estimators` to the number of trees.

4.5 Support Vector Machines (SVM)

It consists of finding an optimal hyperplane that maximizes the separation between different classes, using points close to the margin.

The hyperparameter of this model considered for training is `SVC__C`, this hyperparameter represents the cost term, which balances the maximization of the margin and the minimization of the classification error. A small value allows more errors (more generalization), while a large value penalizes errors and seeks a tighter margin (risk of overfitting).

The most optimal value we have observed is `SVC__C = 10`.

4.6 Naive Bayes (NB)

A classification model that uses Bayes' theorem to calculate the probability that an instance belongs to a class. It assumes that the features are independent, allowing the joint probability to be calculated as the product of the individual probabilities. It then assigns the class with the highest posterior probability. This model does not contain hyperparameters to be adjusted.[4]

4.7 Gradient Boosting Classifier (GBC)

Generates sequential weak models, each correcting the errors of the previous one. It uses decision trees and minimizes a loss function to optimize the final model.[5]

We have optimized these hyperparameters:

- GB__n_estimator = 10.
- GB__learning_rate = 0.01.
- GB__max_depth = 3.

  where GB__n_estimator is the number of trees, GB__learning_rate controls how much trees are adjusted to the errors and GB__max_depth being the maximum depth of trees.

4.8 MLP Model

It is an artificial neural network consisting of multiple layers of nodes (neurons), where each one applies a nonlinear transformation to perform classification.

We have optimized these hyperparameters:

- MLP__alpha = 1e-08.
- MLP__solver = 'adam'.
- MLP__hidden_layer_sizes = (4,4,4), 3 layers with 4 neurons each.
- MLP__activation = 'relu'.

where MLP__alpha is the regularization L2, which prevents overfitting, MLP__solver determinates how weights are optimized, MLP__hidden_layer_sizes, which are the number of layer and neurons in the model and MLP__activation is the activation function to use.

5 Comparative Analysis

5.1 Comparison of models

As we can see from the barplots, the model achieves a balance between sensitivity and specificity (see F1-score), which is crucial in credit risk assessment. High sensitivity allows us to identify most clients who might struggle with repayments, while good specificity helps avoid unnecessarily denying loans to creditworthy clients.

In this context, we prioritized metrics beyond just accuracy, focusing on those that are more relevant to credit risk, such as F1-score, specificity, and sensitivity, since it is more important to balance the identification of risky clients with minimizing false rejections of eligible ones. This careful consideration of multiple metrics underscores the MLP as the most effective model for our HELOC repayment prediction task, offering reliable risk assessment for lending decisions.

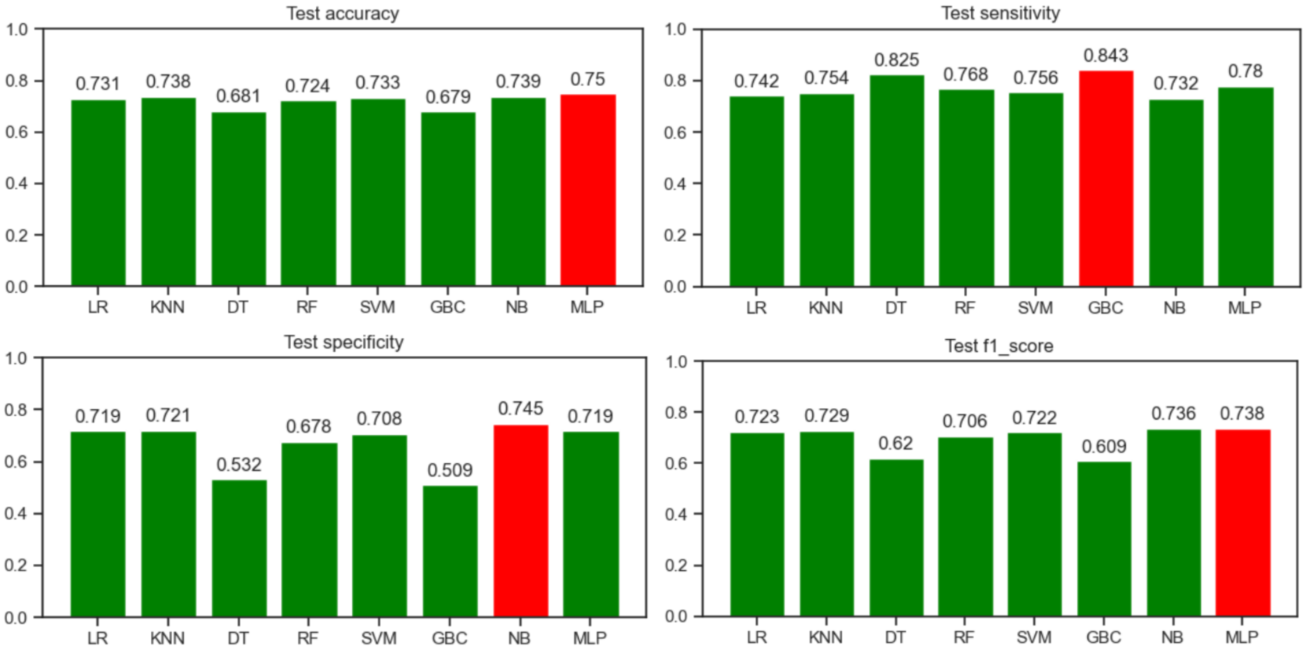

Figure 4: Important metrics barplots of differents models.

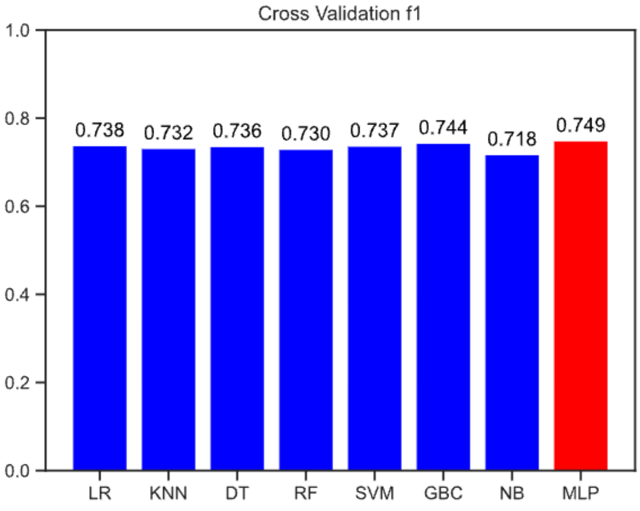

Figure 5: Cross validation F1-score.

5.2 Analysis of the best model

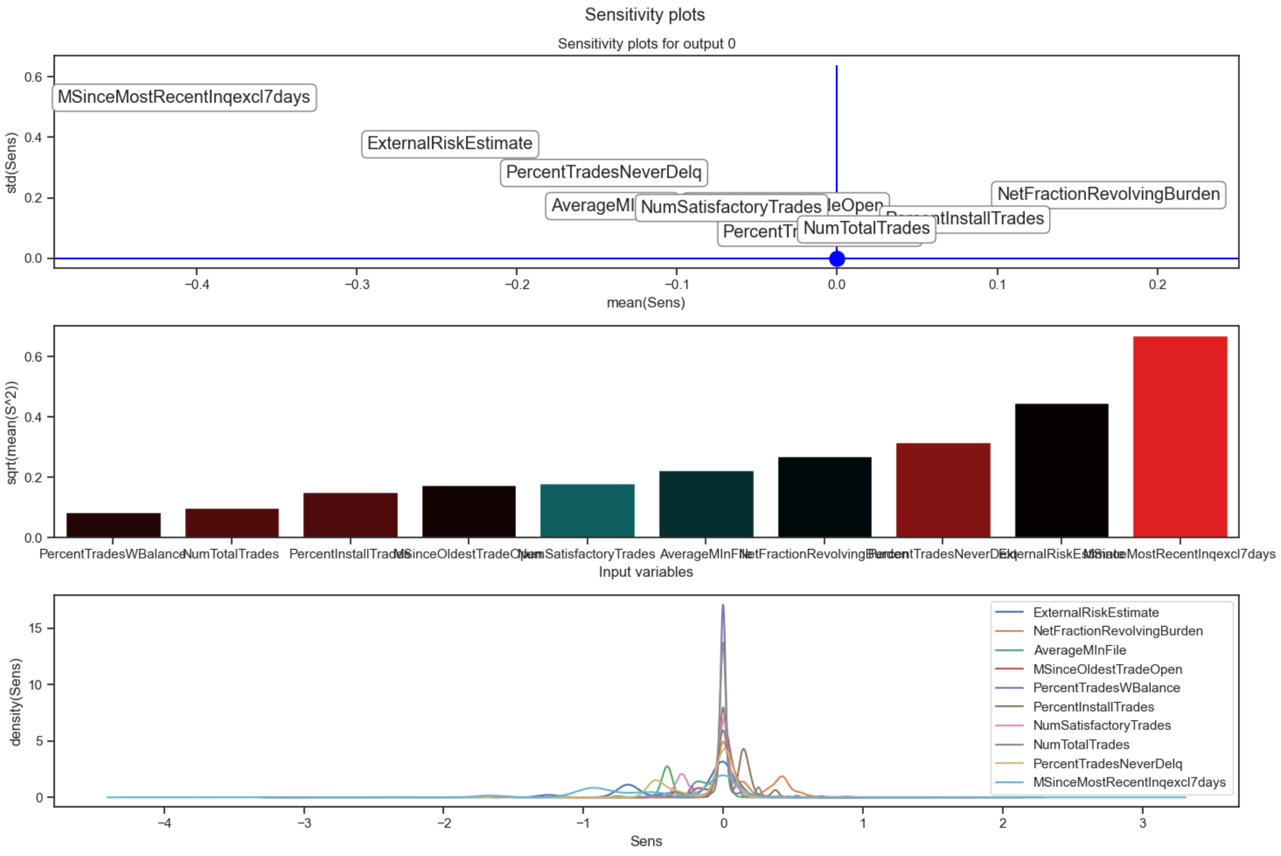

Figure 6: Feature importance using neuralsens

As shown in Figure 6, some variables have a lower impact on the model. Among the least relevant are `NumTotalTrades`, `MSinceOldestTradeOpen`, and `PercentTradesWBalance`. These variables exhibit low sensitivity values, indicating that their influence on credit risk prediction is limited compared to the most important variables, such as `MSinceMostRecentInqexcl7days` and `ExternalRiskEstimate`.

This contributed to the variable election in Section 3.3.6, where we rejected the variables `NumTotalTrades` and `MSinceOldestTradeOpen`, as they were also highly correlated with other features and did not provide significant value to our model.

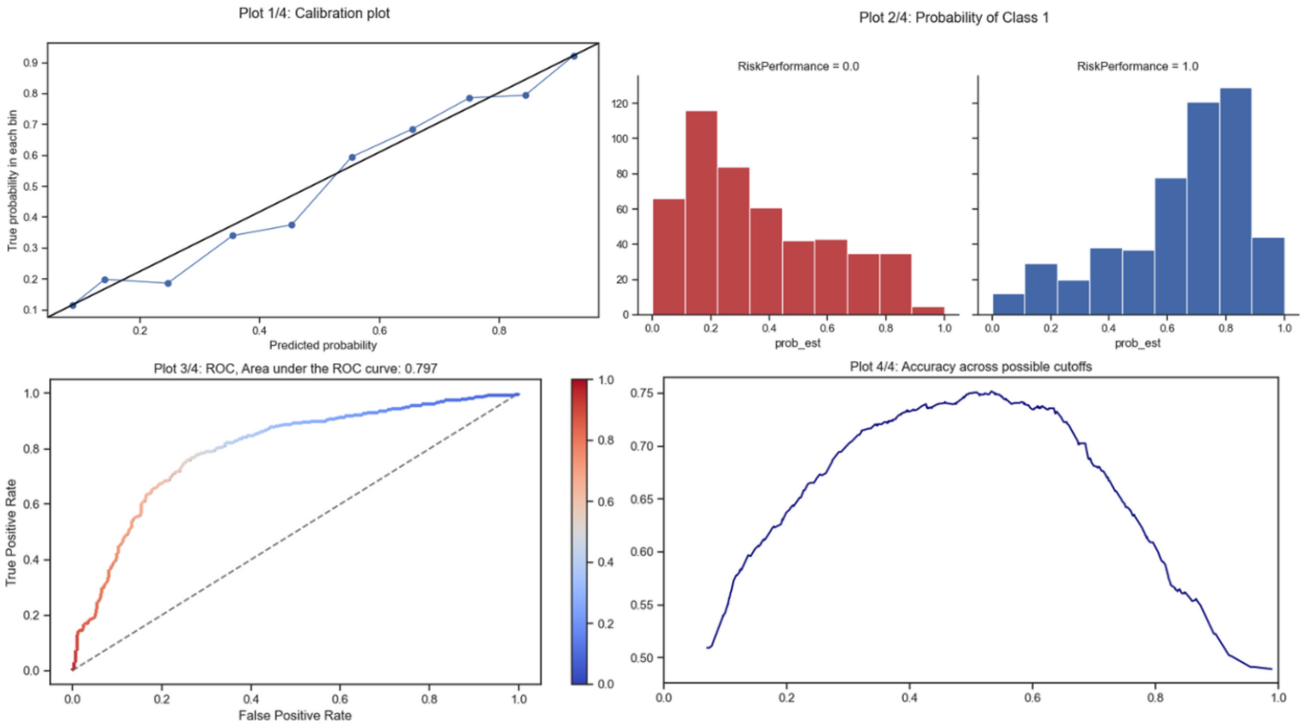

Figure 7: Class performance plots of MLP model

Looking at the calibration plot, we see that the curve follows the diagonal line quite closely. This indicates that the predicted probabilities align well with the actual probabilities, which means that the model is well calibrated. Turning to the histograms, we see that both are asymmetric, so they seem to separate well when to give credit and when not to give credit. Moving on to the ROC curve, it shows a good performance with an Area Under the Curve (AUC) of 0.797. This indicates that the model has a good discriminative ability. Finally, the accuracy curve shows a clear peak around 0.75, indicating that the model achieves good overall accuracy. The symmetric shape suggests a balanced performance across different probability thresholds.

6 Conclusions

Our study on predicting repayment of Home Equity Lines of Credit (HELOCs) has shown that machine learning techniques are indeed useful for assessing credit risk. After cleaning the data, analyzing it thoroughly, and testing several classification models, we have developed a robust method to predict whether borrowers will repay their loans. The Multi-Layer Perceptron (MLP) model turned out to be the best, with the highest F1 score and sensitivity. This means it is especially good at identifying both high-risk and low-risk borrowers, which is crucial for banks to minimize losses and be able to lend to more people.

We have seen that choosing the data characteristics well and adjusting the model parameters greatly improves the results. However, it is important to remember that, while our model is very useful, it should not completely replace human judgment in lending decisions. In the future, we could improve the model by including more data, such as general economic indicators or alternative credit information.

It's worth noting that although the MLP achieved the best results, simpler models like Logistic Regression (LR) and tree-based models such as Random Forest (RF) and Gradient Boosting Classifier (GBC) also performed quite well. If greater simplicity or interpretability were desired, these models could be a valid option, sacrificing only a small loss in accuracy. Our choice of MLP was primarily based on the performance metrics mentioned earlier, prioritizing accuracy over model explainability. This decision will ultimately depend on the specific needs of the financial institution and the desired balance between precision and simplicity.

References

[1] myFICO. (n.d.). *What is a FICO Score?* Retrieved September 26, 2024, from `https://www.myfico.com/credit-education/what-is-a-fico-score`

[2] Bank of America. (n.d.). *What is a home equity line of credit (HELOC)?* Retrieved September 26, 2024, from `https://www.bankofamerica.com/mortgage/learn/what-is-a-home-equity-line-of-credit/`

[3] DataCamp. (2024). *Random forest classification with Scikit-Learn.* Retrieved October 1, 2024, from `https://www.datacamp.com/tutorial/random-forests-classifier-python`

[4] DataCamp. (2023). *Naive Bayes classification tutorial using Scikit-learn.* Retrieved October 1, 2024, from `https://www.datacamp.com/tutorial/naive-bayes-scikit-learn`

[5] Cienciadedatos. (2023). *Gradient boosting con Python.* Retrieved October 1, 2024, from `https://cienciadedatos.net/documentos/py09_gradient_boosting_python`