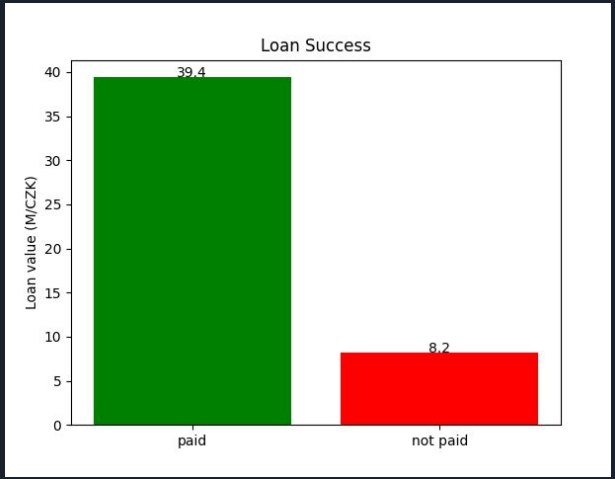
To loan or not to loan

Machine Learning 2022/23 - Data Mining Project

A bank wants to improve their loan system by identifying good clients (whom to offer some additional services) and bad clients (who might cause losses).
It's also useful for the bank to identify if, when given a loan, a client would actually repay it in the end.

Group: 69
Bruno Rosendo, up201906334
João Mesquita, up201906682
Rui Alves, up201905853

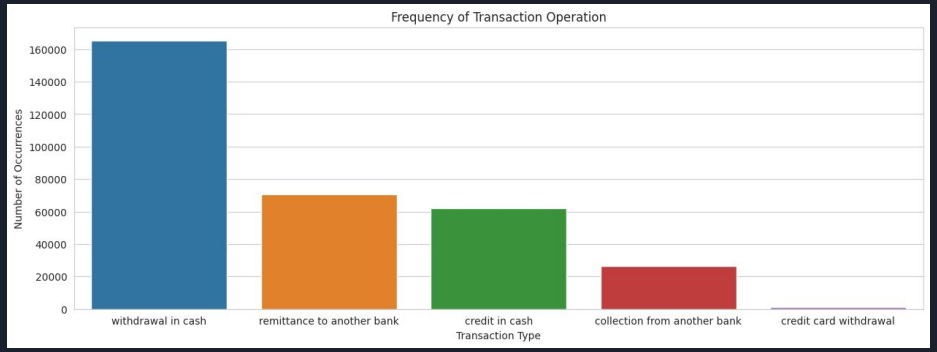
Business Description



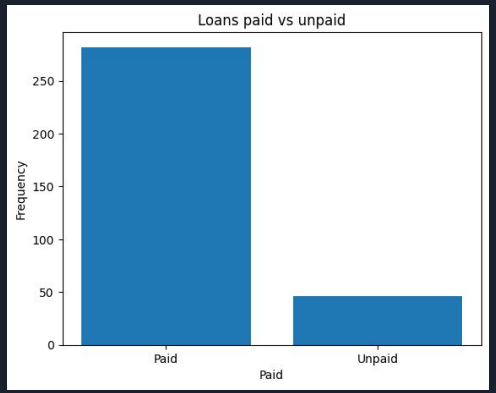
- A bank wants to use the system to improve their business' profits. For that, the system should correctly predict whether a loan will be paid or not, based on the client's data.
- However, it's important to prioritize the loans that will not be paid, since those will lead to financial losses.
- In the considered time interval, the bank lost 8.2 million CZK on unpaid loans (14% of loans and 17.2% of the investment).
- The goal is to reduce the amount of unpaid loans to at most 5%, while maintaining the profits made with successful loans.

Data Analysis

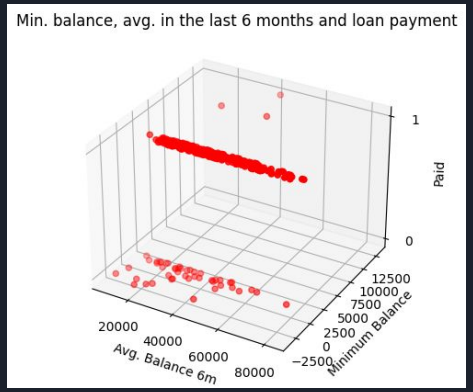
The transaction operation gives detailed information about how the transaction was made and processed



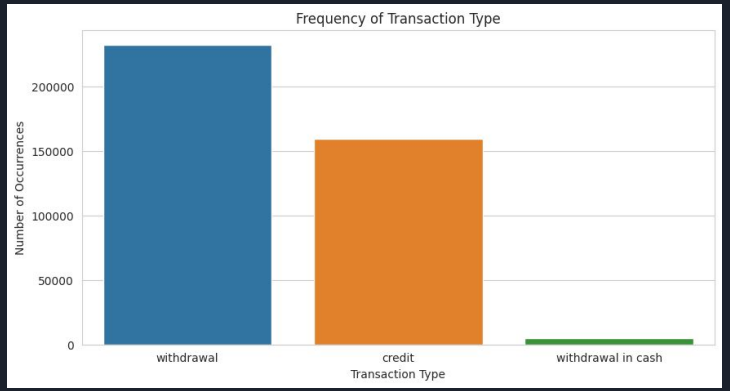
Only around 14% of the loans were not paid, meaning that accuracy isn't a good metric to optimize for.



Clients with low minimum balance tend not to pay the loan, even if the recent average balance is higher.

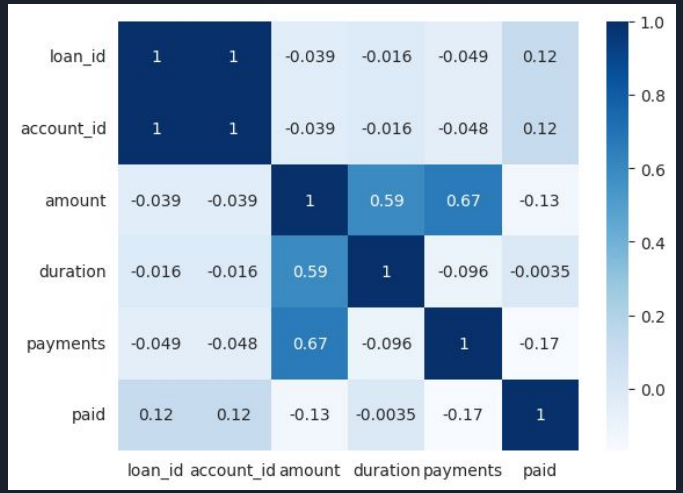


The transaction type gives the same information as the operation, but less detailed and containing inconsistencies.

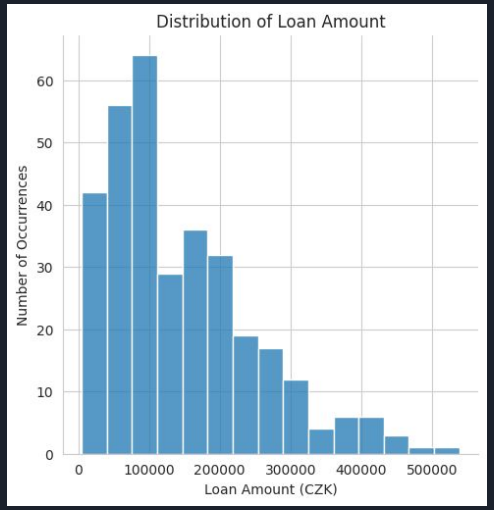


Data Analysis

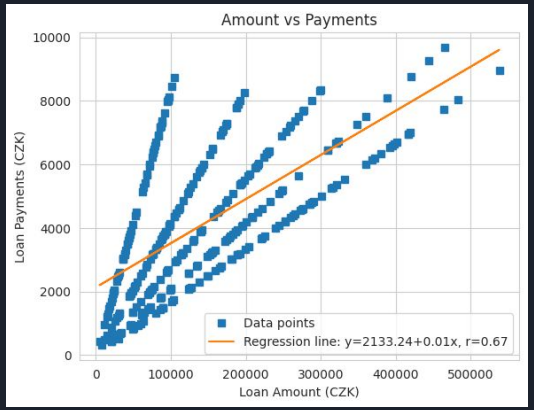
Some attributes of the loans table show interesting correlation. For example, we can see that the amount, duration and the payments are intrinsically connected.



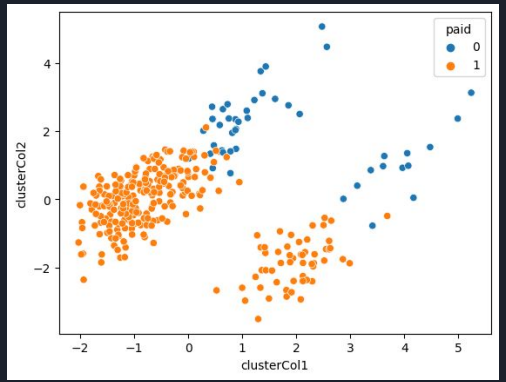
Loans with lower amounts are more common, while greater loans are scarce.



The amount of the loans and the respective payments show great correlation.



KMeans was used to generate clusters on PCA components between clients who paid the loans or not.



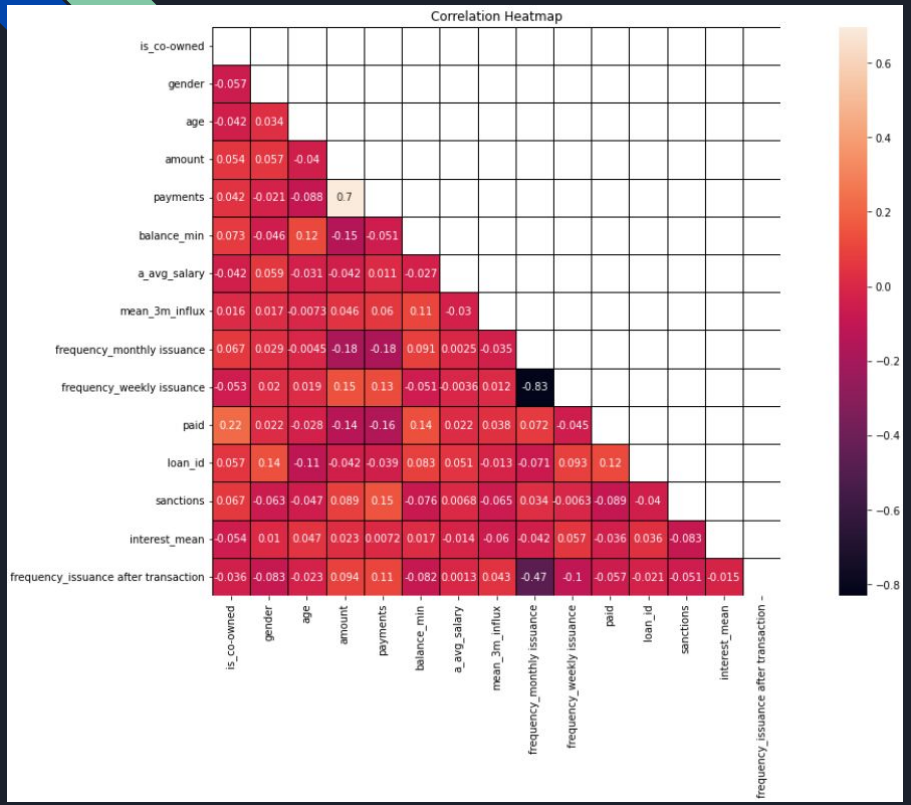
Dimensions of Data Quality

1. **Completeness** ❌
   There are some mandatory fields missing in the districts table and the transactions table contains attributes with a large number of null values (e.g. *operation* and *bank*). Therefore, the data is not considered complete.
2. **Consistency** ❌
   A prime example of the lack of consistency is the overlapping info between the columns *type* and *operation* of the transactions table.
3. **Conformity** ❌
   In general, dates follow the "yy-mm-dd" format. However, the client's birth date adds 50 to the month part of the date, so the data is not considered conformant.
4. **Accuracy** ✅
   The data is extracted directly from a real bank so the information is considered accurate.
5. **Integrity** ❌
   Although the goal of the work is to predict whether a loan will end successfully, most of the accounts do not have associated loan requests - There are orphaned records, so the data lacks integrity.
6. **Timeliness** ❌
   The present data is considerably outdated (more than 20 years old), so it does not achieve timeliness.

Data Preparation -> Feature Engineering

- The tables making up the dataset need to be joined into a single table, ready to be used by machine learning models.
- Duplicate entries are then dropped, since there are many-to-one and many-to-many relationships.
- Tables with a great amount of information are generalized by simple columns (e.g. average account balance). Some other columns weren't considered important and were discarded (e.g. district name and card).
- New features include: average salary, crime ratio, household mean, interest mean, sanctions for negative balance, average balance in the previous year, semester and month, minimum, maximum and average balance, and mean influx in the past year, semester and trimester.

Feature Selection



- Columns with missing values are either dropped (e.g. transaction type) or replaced using linear regression.
- The client's birth date is converted into their age at the time of the loan.
- Categorical data is transformed using CatBoost and One-hot Encoding.
- Some highly correlated attributes are dropped.
- Wrapper methods were used to choose the final attributes, namely forward, backward and bi-directional selection.
- The amount, payment, balance min and whether it's co-owned are the most correlated features with the target variable.

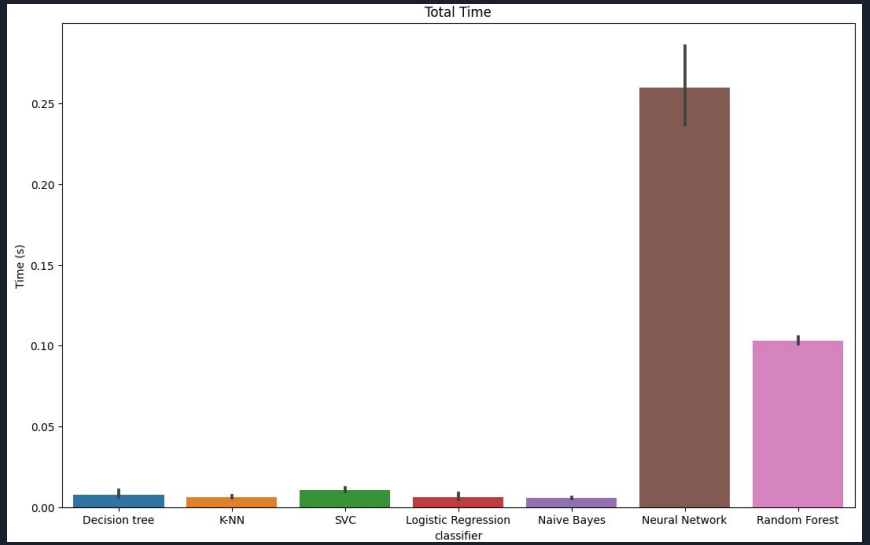
Experimental Setup

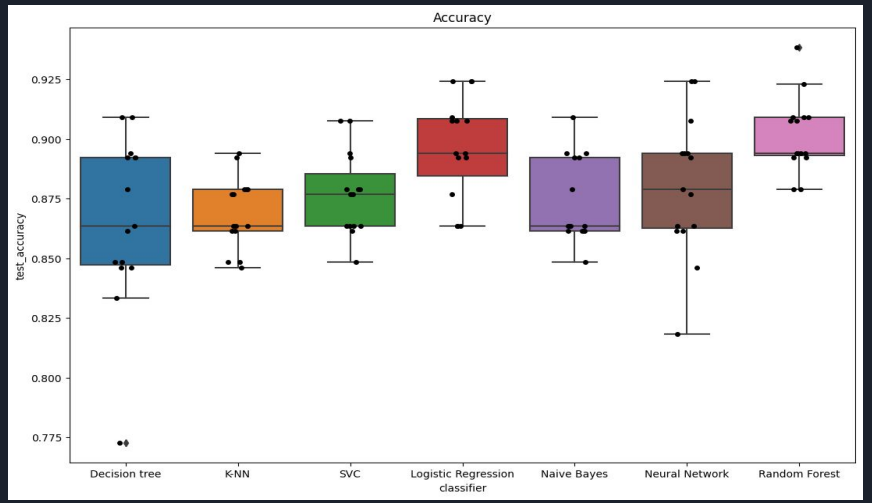
- The model is tested with various classifiers, from which Random Forest, Logistic Regression and Neural Network stand out.
- Z-Score normalization is applied on distance-based models like K-NN to remove bias derived from numerical scale.
- Analyzed evaluation metrics: Accuracy, Precision, Recall, F1-score and ROC-AUC.
- To find the best parameters for each classifier, there is a Parameter Tuning step using Grid Search with Cross Validation.
- Dataset is oversampled with SMOTE to compensate its dimension.
- Detection of outliers using DBSCAN was also tested.

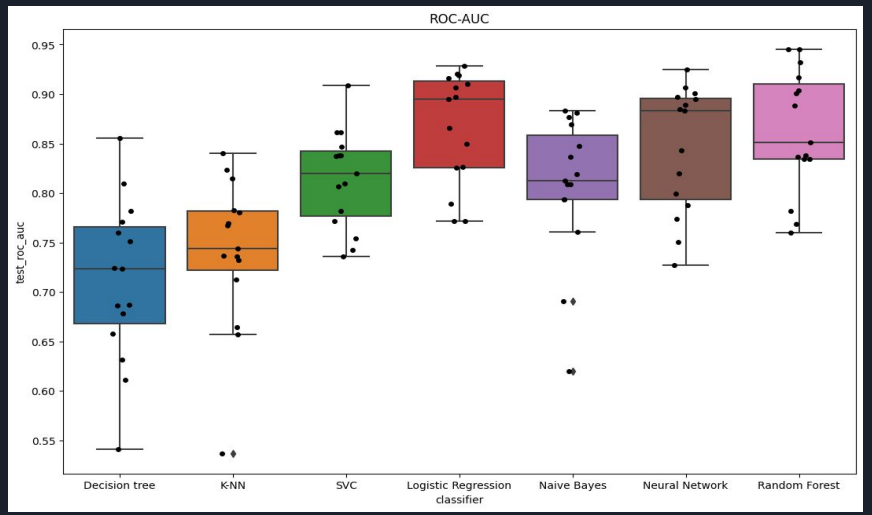
Results

AUC of 96.3% on the public score and 92.3% on the private score.

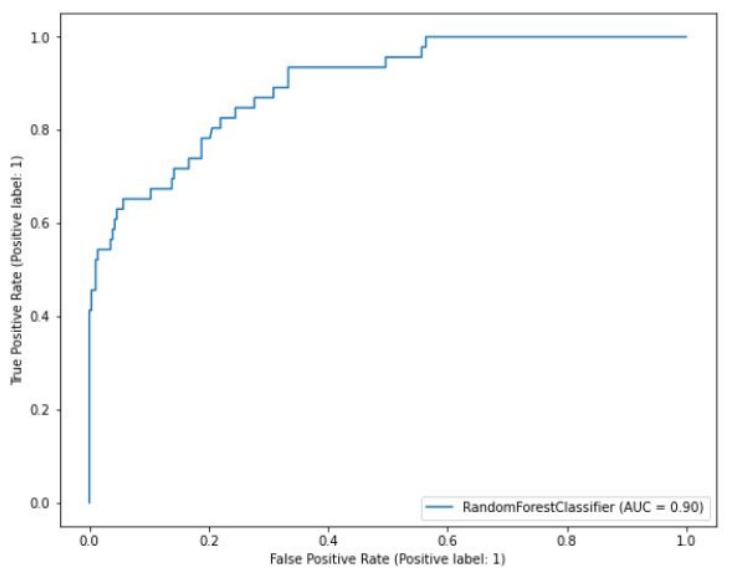
Submission and Description	Private Score	Public Score
submission (1).csv Complete · João Mesquita · 2h ago	0.92345	0.96255



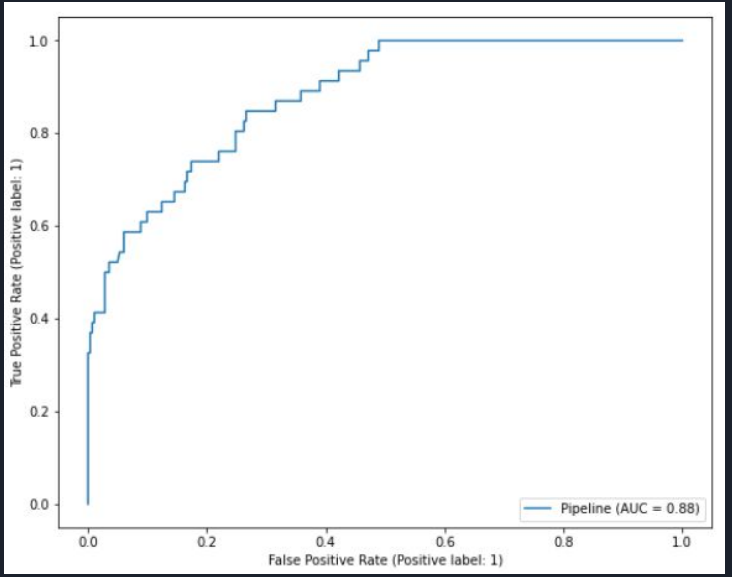




Smote Results



AUC without SMOTE



AUC with SMOTE

Conclusions, Limitations and Future Work

- The dataset used for training is small and has a lot of flaws, which makes it hard to develop a robust model for this use case.
- Random Forest was the model which overall performed the best with the tested dataset.
- Oversampling techniques performed unexpectedly poorly.
- Feature selection is a crucial step and could be further explored to achieve better results.
- Hyperparameter tuning helped improve the system but it’s not as important as feature selection.
- The defined business goal was achieved, with around 4% of false positives (clients who were expected to pay the loan but didn’t), leading to saving around 5.8M CZK and 88% of the investment returned.

Annexes

Data Mining Goals

The goal of this project is to determine when a Loan, associated with an Account, won't end successfully, that is, won't be totally paid. However, the data mining to be done on this project should prioritize the reduction of unpaid loans instead of maximizing the number of paid loans, as described in the Business Description. Therefore, the data mining goals can be defined as follows:

- Collect a dataset containing a significant amount of loans, clients, and other bank information.
- Analyze the data in order to better understand it. Generate relevant statistics, tables, and plots.
- Prepare and process the data based on the analysis done previously, so we can use it correctly on the predictive models.
- Build a machine learning model based on the processed data, evaluate its results and review the whole process, taking the newly acquired info into account.

Missing Values

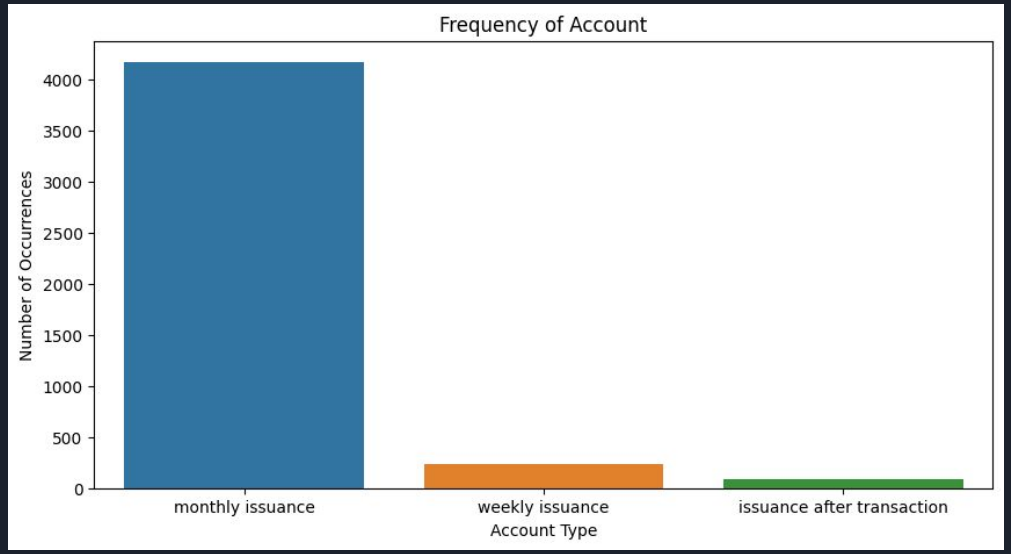
All data tables were analyzed and the following missing values were identified:

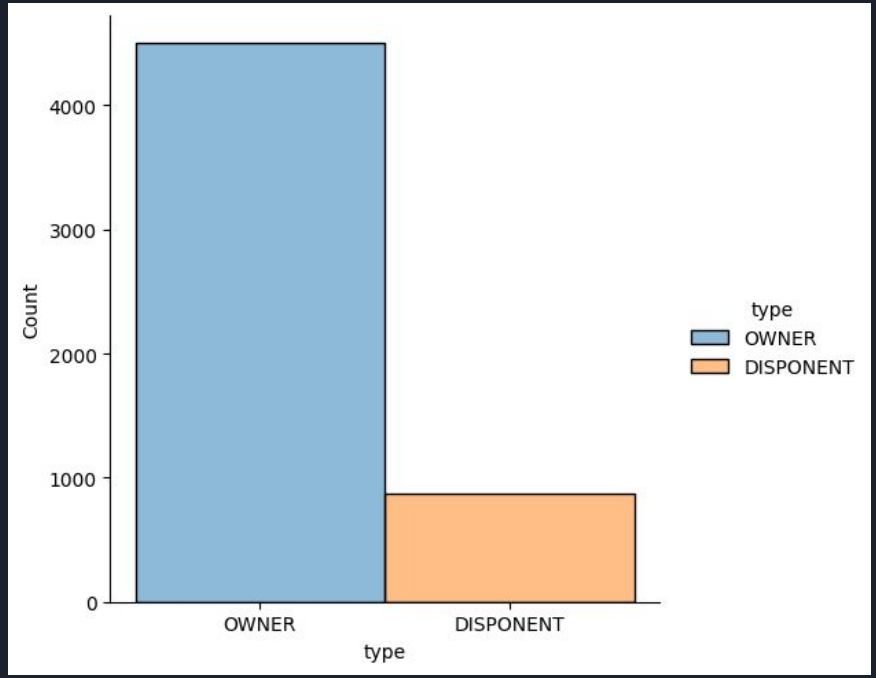
- Transactions
  - operation: 70761
  - k_symbol: 185244
  - bank: 299443
  - account: 294456
- Districts
  - unemp_rate95: 1
  - num_crimes95: 1

In the *transactions* table, the *k_symbol*, *bank*, and *account* columns are categorical, so the missing values are replaced below with an *unknown* value. As for the *operation* column, it proves to be highly correlated with the *type* column, as we check in Slide 17, so we discarded it.

Regarding the *district* table, there is one row with missing values in the columns *unemp_rate95* and *num_crimes95*. In these cases, since they are ratios, we can replace them using linear regression.

Account Type

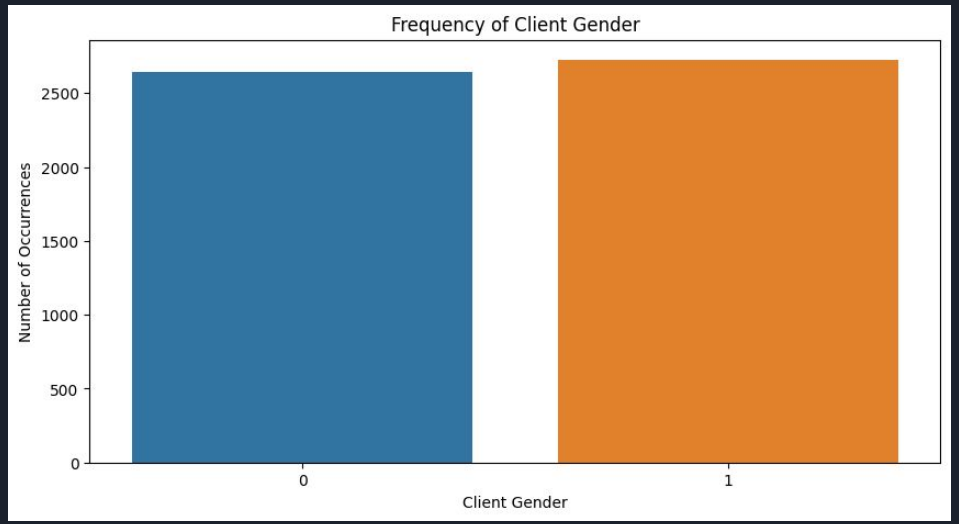




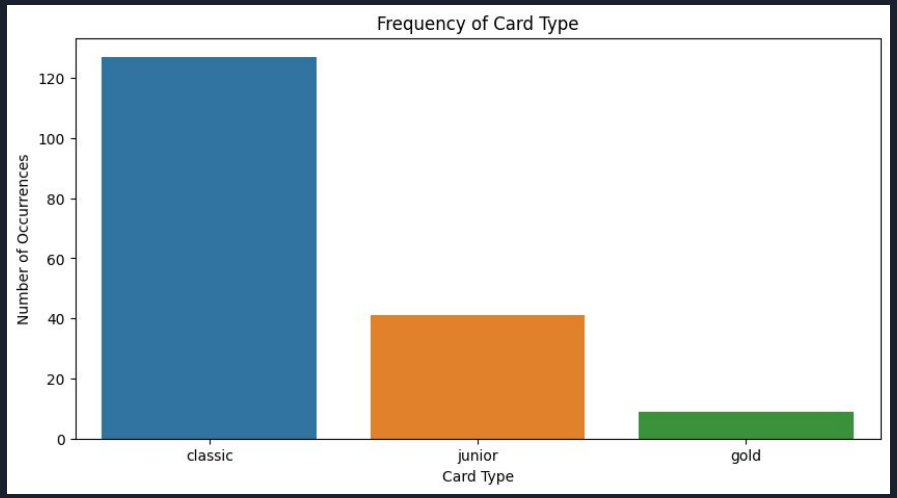
All Accounts have at least 1 Disposition. There are more Dispositions than Accounts, since some clients are **owners** while others are **disponent owners.**

Therefore, it might be useful to create an attribute on the Account table that reflects whether the account is co-owned.

Clients and Cards



The dataset is fairly balanced when it comes to the clients' genders (0 - female, 1 - male).



Almost no client in the dataset owns a card, which might reflect on the importance of this entity.

The type of Card can help us rank clients. A client with a gold card is more involved than a client with a junior card. On the other hand, the **issuance date** allows us to only consider cards issued before a loan.

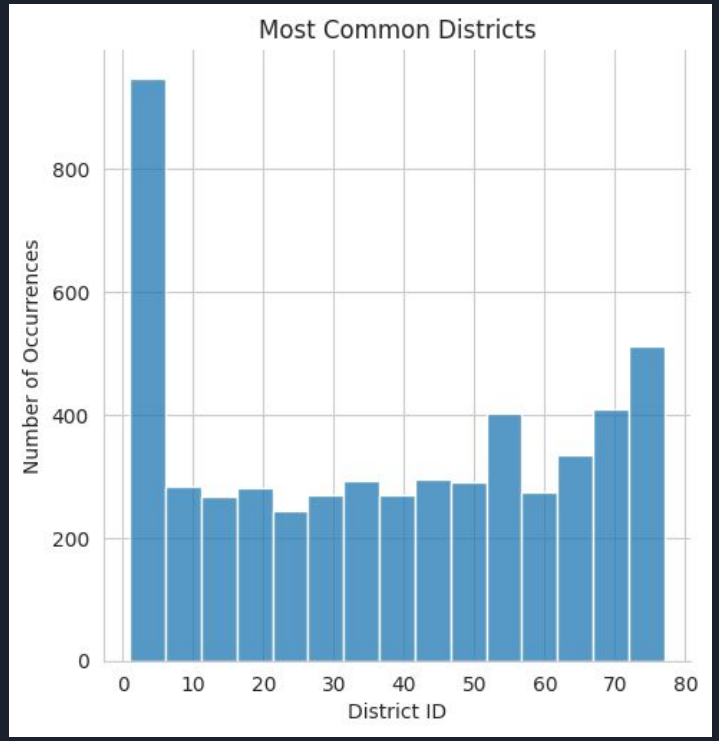
Transaction Type and Operation

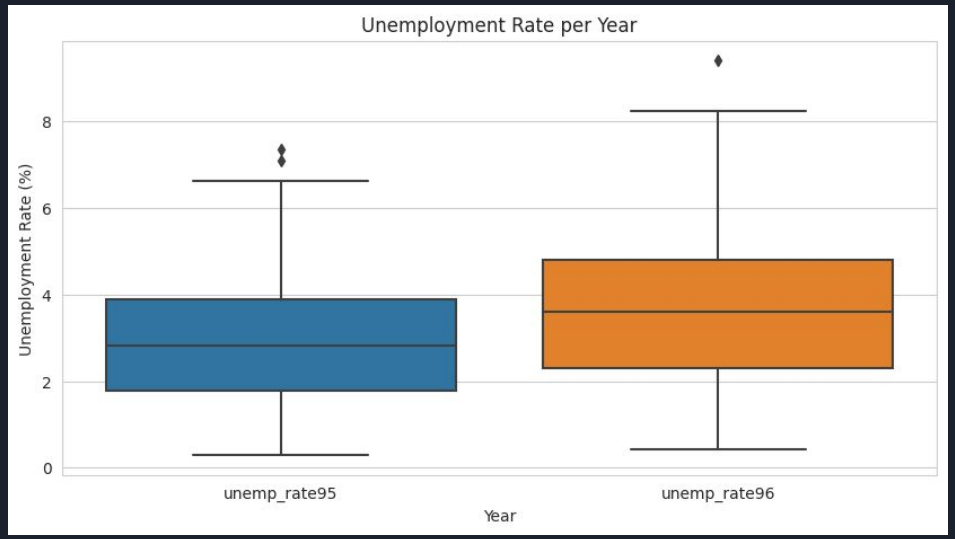
The columns *type* and *operation*, in the *transactions* table, transmit identical information except the *operation* being more detailed. Thus, after investigating the dataset, the following mapping was found:

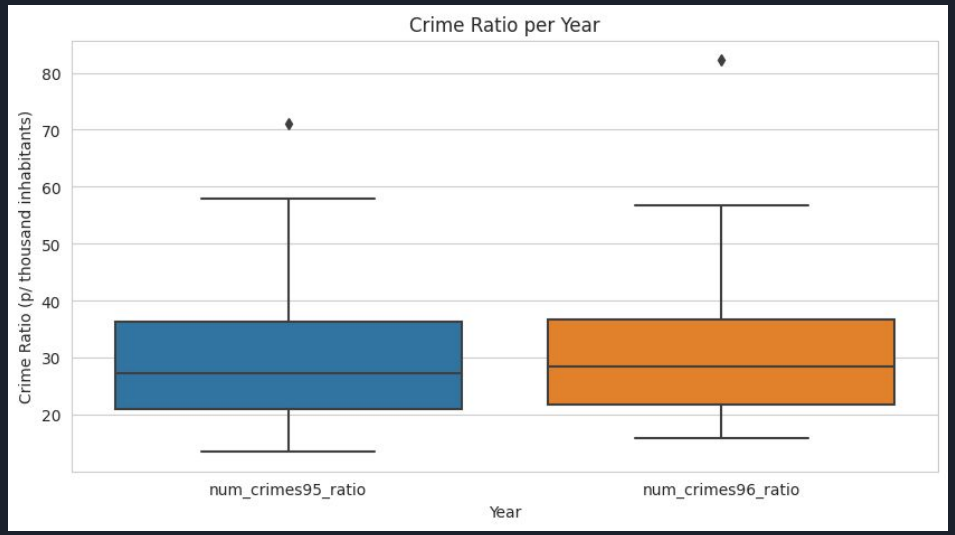
Type	Operation
credit	unknown, collection from another bank, credit in cash
withdrawal	credit card withdrawal; remittance to another bank; withdrawal in cash
withdrawal in cash	withdrawal in cash

To confirm this, we did a chi-square test, concluding with 95% confidence that those two features are correlated, so we dropped the operation column since the withdrawal fields are inconsistent in the type column.

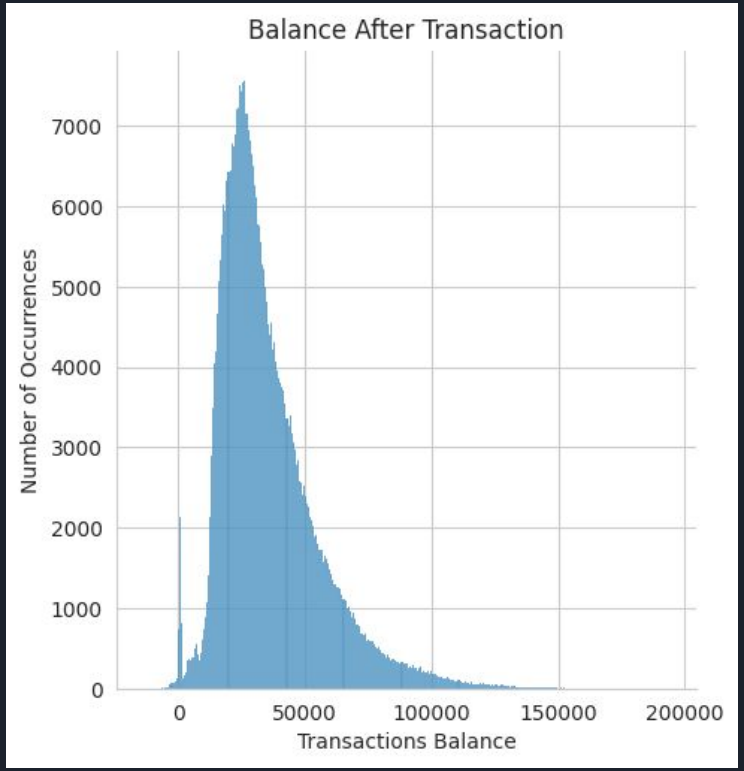
District Information

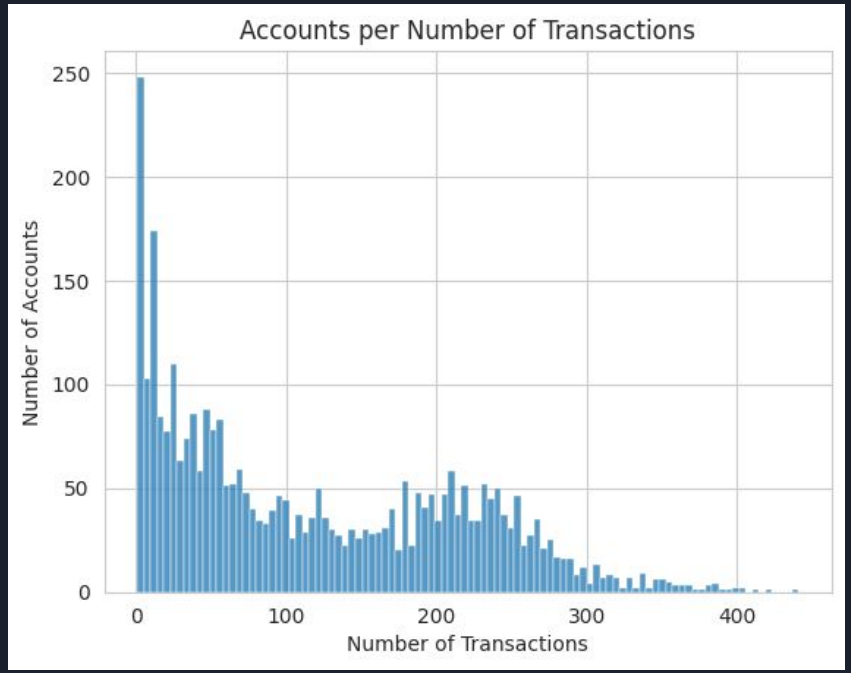






Transactions





	Amount (CZK)	Balance (CZK)
Standard Deviation	9190	19692
Variance	8.4e+07	3.9e+08
Interquartile Range	6373	2237

Data Understanding

To study the data, we proceeded to get the:

- Frequency of the values of nominal data
- Min, Max, Mode and Mean values
- Q1, Q2 and Q3
- Standard deviation, variance and interquartile range
- Contingency tables
- Covariance Matrix
- Correlation Matrix
- Pearson Correlation Coefficient between quantitative data

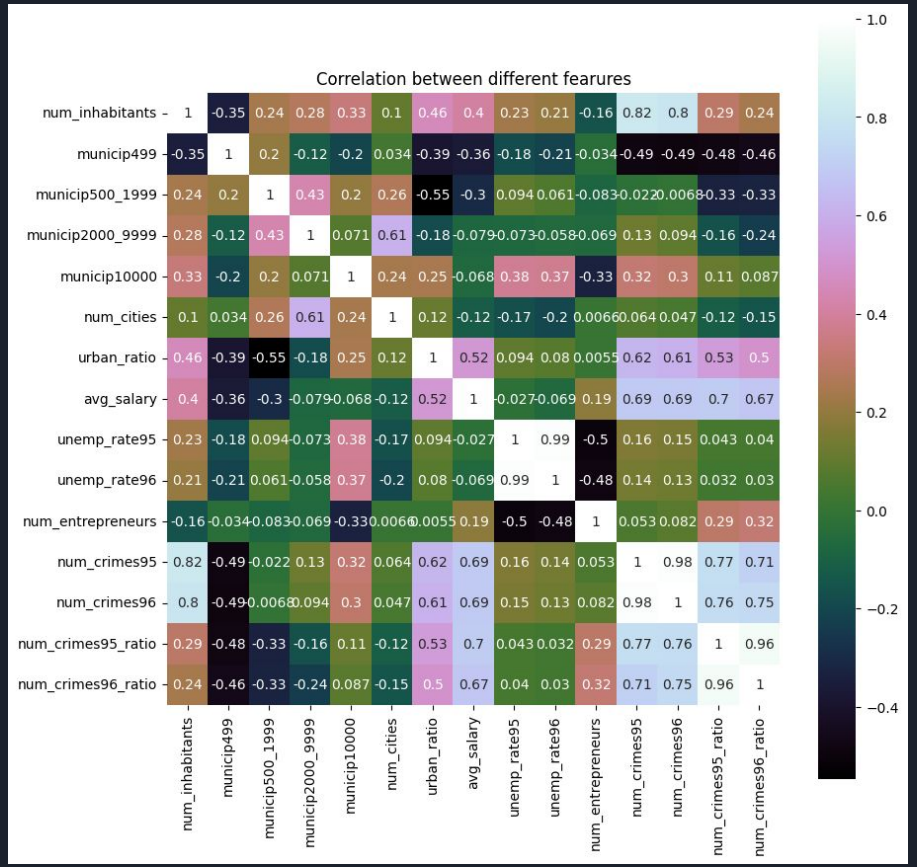
With these calculations, we managed to decide when to use or not to use a feature from our initial dataset, which is explained in the feature engineering section.

Data Understanding - Districts

To study the data, we proceeded to calculate the correlations between the features of each table.

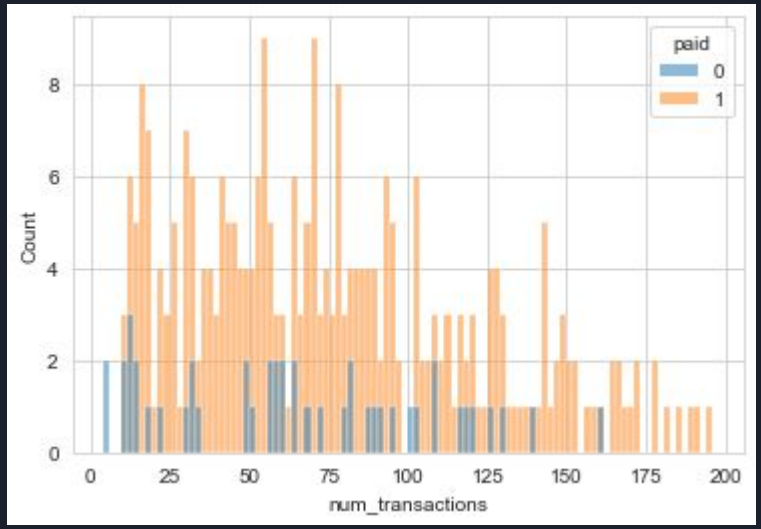
From here, we noticed that there was a significant relationship between the num_crimes95, and num_crimes96 columns and their respective ratios.

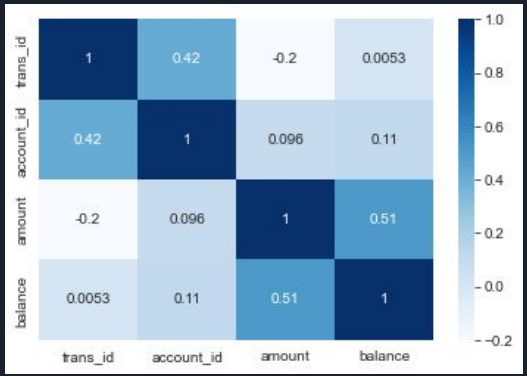
We also see that this relationship exists with the avg_salary feature, so we decided to do a spearman correlation test to confirm it.



Data Understanding - Transactions

Analysing the transactions data with the account and the respective loan, despite not seeing a clear pattern on the distribution of the number of transactions for an account and the status of the loan (paid = 1, not paid = 0), we can identify that the best clients would be the ones that make more transactions, as they are active clients and trustworthy!





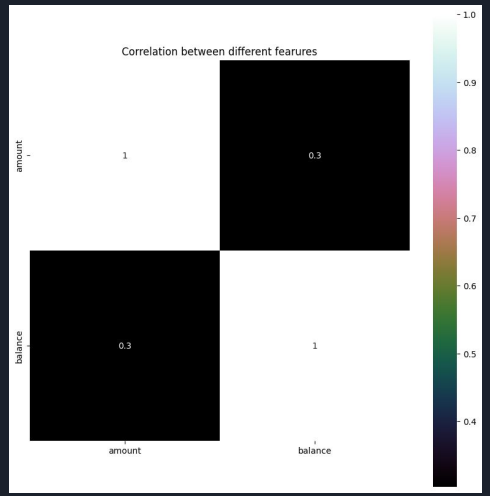
On transactions, we clearly see that the amount and balance are highly correlated, leading us to study in more detail these types of variables (amount of transactions and balance of the account) by extracting features related to them in Feature Engineering.

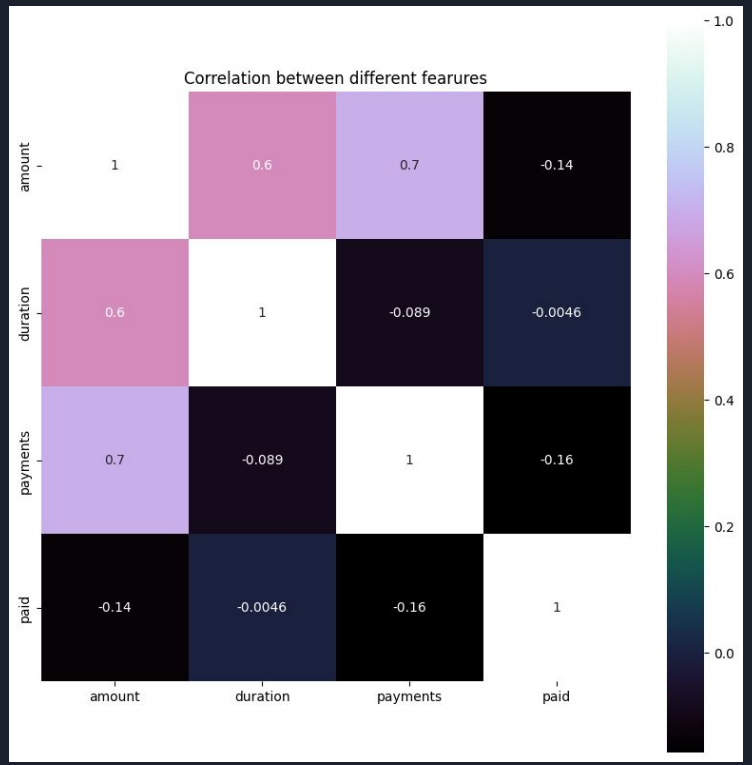
DU and Feature Selection - Loan and Transaction

Analysing the loans table, we see that the amount, duration, and payments columns are highly correlated, so we should discard one of them.

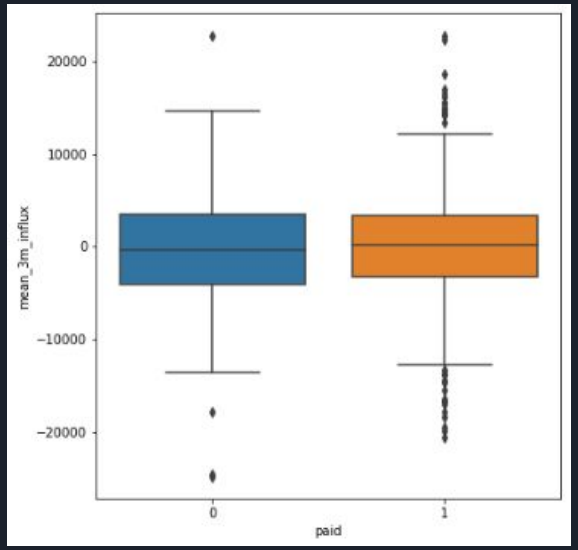
Since payments and amounts provide more information and are more correlated with the target column (paid), we decided to keep them.

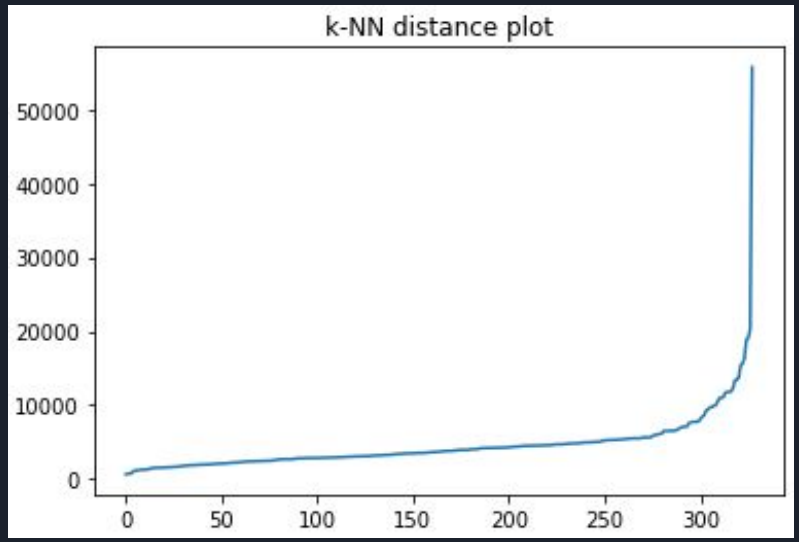
From the analysis of the transactions table, we couldn't draw any conclusions.
We proceeded to calculate the correlation with the target column after they're joined so we can retrieve more information.





Data Preparation - Outliers Detection





Statistical and Clustering-based outlier detection were used to analyze outliers.

The DBSCAN algorithm detected 19.8% of outliers. K-NN distance plot is calculated to determine the radius of neighborhoods.

Removing the outliers would make the dataset contain only 263 rows. Thus, outlier removal did not improve the predictions and was not a correct approach. The resulting high percentage of outliers can be associated with the dimension of the dataset. Nevertheless, since this work is associated with detecting irregular behavior that may cause a loan to be unsuccessful, removing values simply because they deviate from the norm is not the best solution, as they can provide specific events that lead to unpaid loans.

Feature Engineering

sanctions: number of times the client was sanctioned for negative balance.

household_mean: mean of how much a client pays for their house.

interest_mean: mean of how much a client earns in account interest.

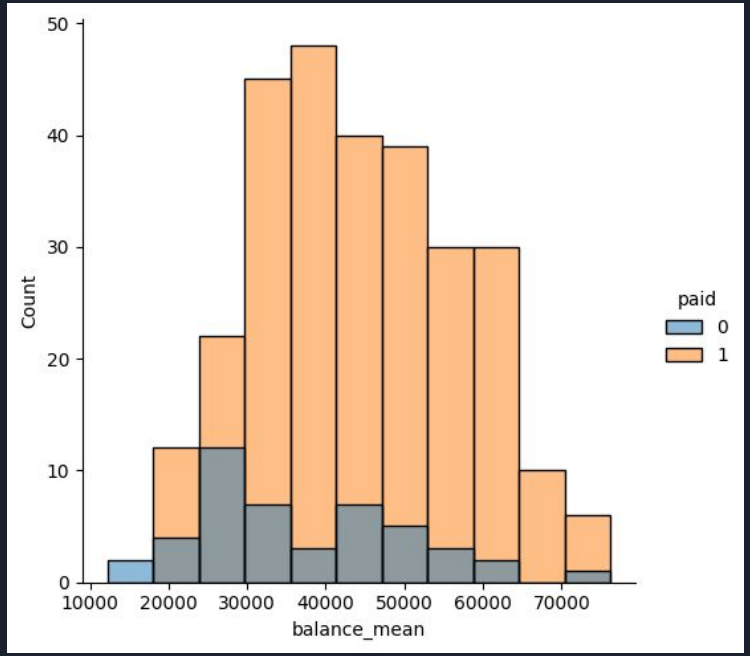
prev_<1m,6m,1y>_balance: client's account balance in a defined time before the loan (1 month, 6 months and 1 year).

balance_<mean,max,min>: client's minimum, maximum and mean account balance.

mean_<1m,6m,1y>_influx: client's balance variation in the last month, 6 months and 1 year.

num_crimes<95,96>_ratio: ratio between the number of crimes and inhabitants in a district.

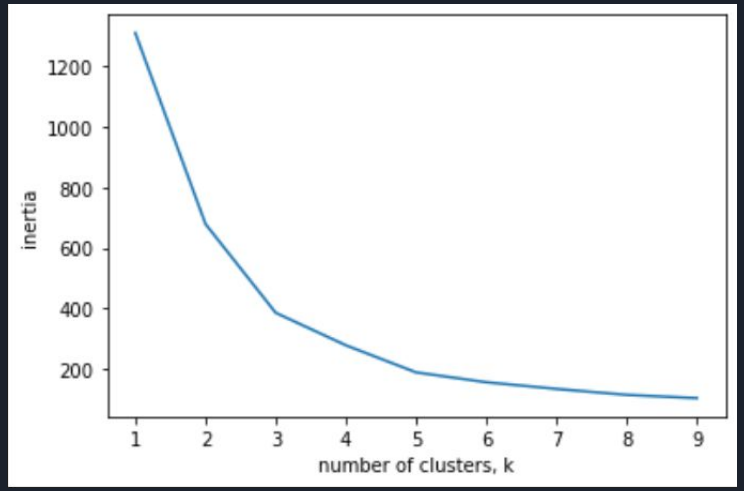
is_co-owned: whether a client's account is co-owned by another client.

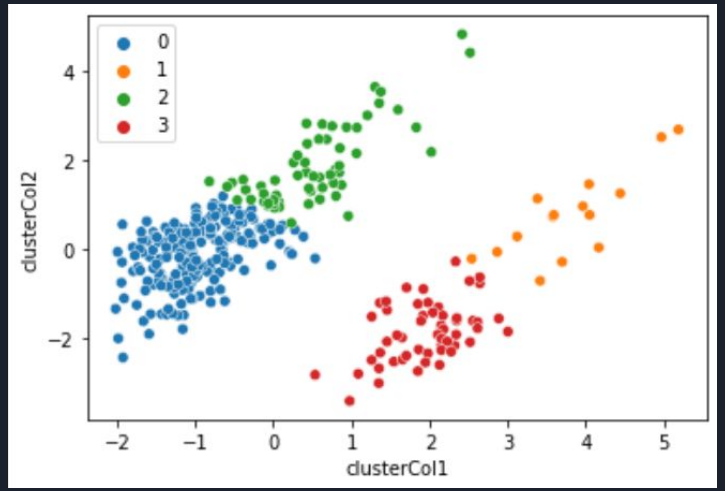


- Correlation matrices were used to identify redundant features.
- Wrapper methods were used to select features using ML algorithms. We tested forward, backward and bi-directional selection.

Clustering

- In order to find clusters of clients or transactions, we used PCA alongside KMeans. PCA is used to reduce the dimensionality of our features. KMeans was used to generate the clusters because it's fast and uses a stochastic approach that frequently works well.



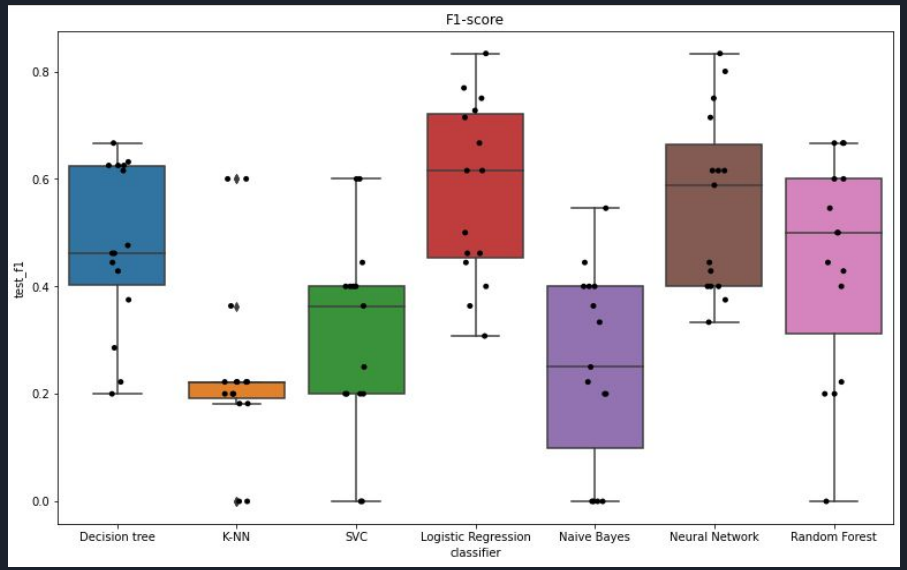


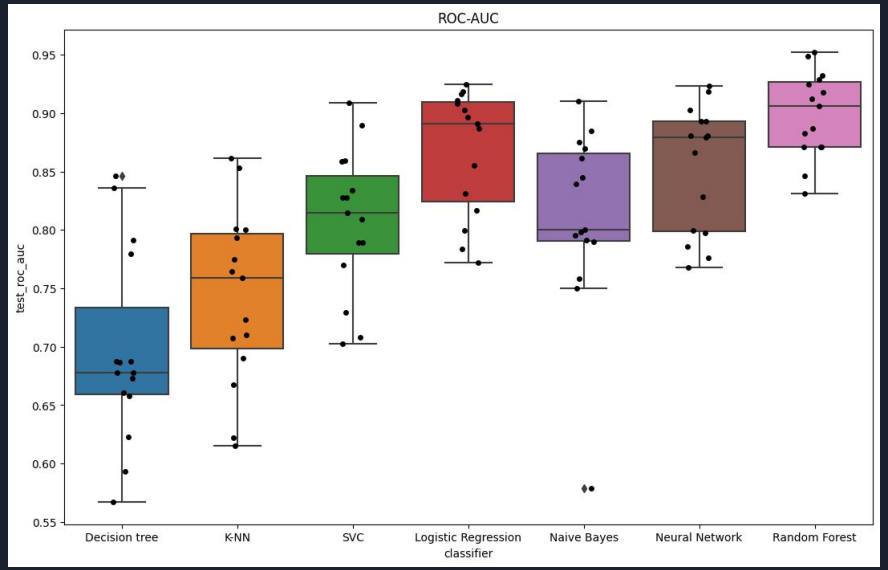
ML Models - Overview

The best-performing models for this predictive task were Random Forest, Logistic Regression, and Neural Network.

These were tested with Repeated Stratified K-Folds and Cross-Validation to guarantee that the ratio between the positive and the negative class was consistent with the training data and no overfitting occurs.

Since our dataset is unbalanced, we also applied SMOTE on the training data in each iteration of the Cross Validation and Repeated Stratified K-Fold.





Parameter Tuning - non-linear SVM

Support Vector Machine is primarily used for classification problems, such as the one being tackled, and the main goal of the algorithm is to create the best decision boundary that can separate the target variable.

The analyzed parameters are:

C - Controls the trade-off between accurate classification and smooth decision boundary.

degree - Degree of the polynomial kernel function used to split the data.

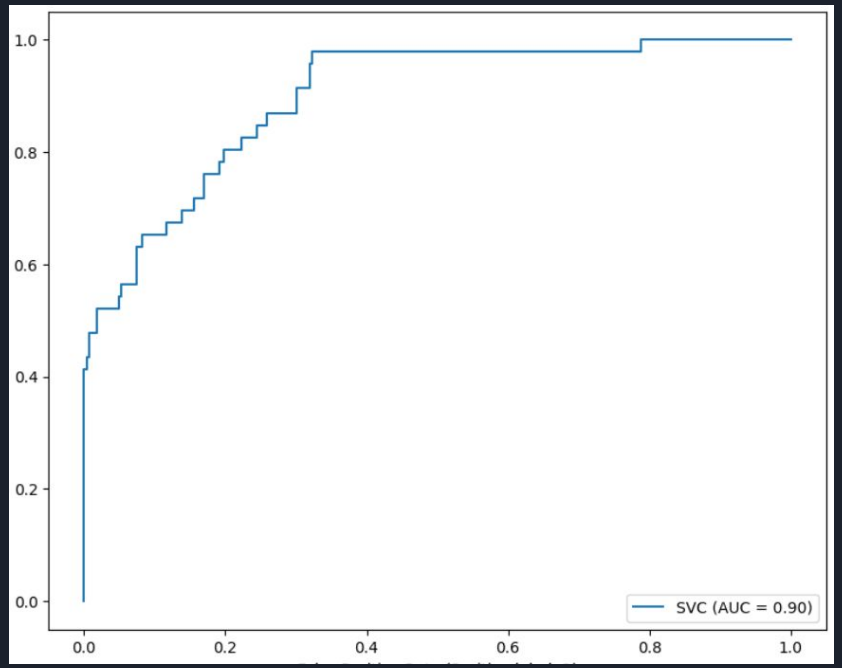
kernel - Kernel type to be used in the algorithm (E.g *linear*, *rbf*, etc.)

gamma - Kernel coefficient used by certain kernel functions.

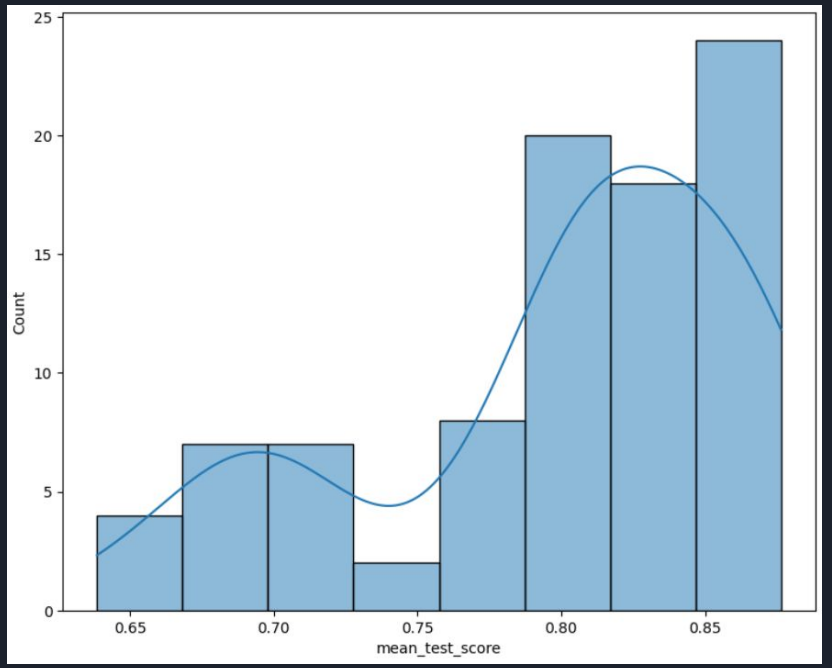
Best Parameters: **{'C': 0.01, 'degree': 2, 'gamma': 'auto', 'kernel': 'linear'}**

Beware that increasing *C* may lead to overfitting, since the classifier adapts more precisely to the training data.

Parameter Tuning - non-linear SVM



Best parameters ROC-AUC



Grid Search ROC-AUC distribution

Parameter Tuning - Random Forest

Random Forest is based on the concept of **ensemble learning**. The algorithm contains a number of decision trees on various subsets of the dataset and performs the average to improve the predictive accuracy of the model.

The analyzed parameters are:

n_estimators - Number of trees in the forest.

criterion - Function to measure the quality of a split.

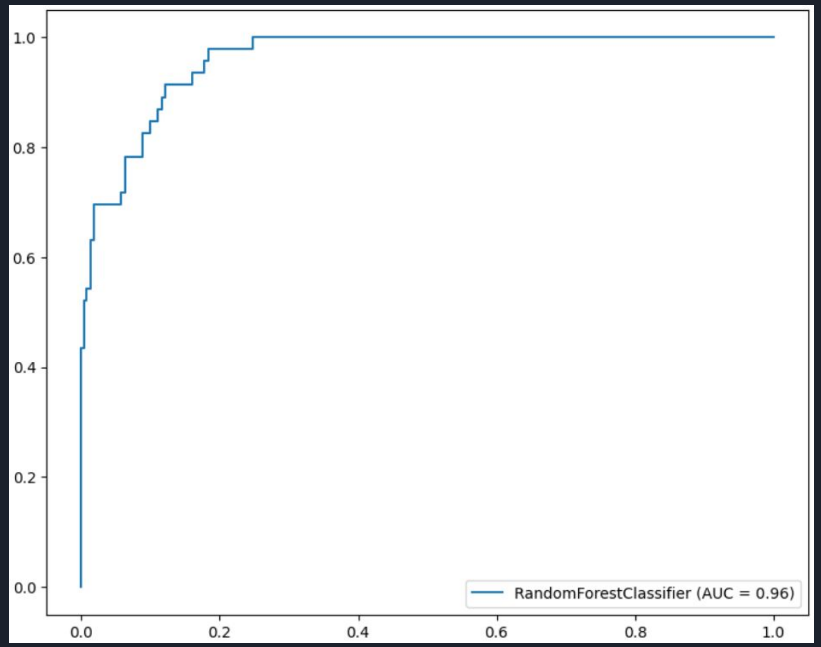
max_depth -Maximum depth of the tree. If none, nodes are expanded until all leaves are pure.

max_features - Number of features to consider during splits.

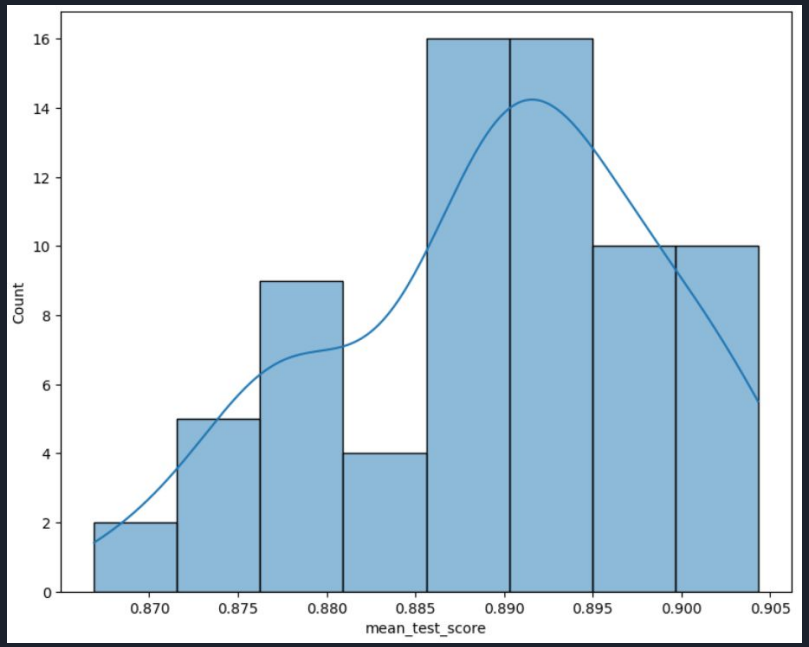
Best Parameters: **{'criterion': 'entropy', 'max_depth': 2, 'max_features': 'sqrt', 'n_estimators': 100}**

The *max_depth* parameter must be controlled to prevent overfitting. On the other hand, increasing the number of trees in the forest leads to higher accuracy and avoids overfitting.

Parameter Tuning - Random Forest



Best parameters ROC-AUC



Grid Search ROC-AUC distribution

Parameter Tuning - Logistic Regression

Logistic Regression is one of the most popular machine learning algorithms. It is used for predicting the categorical dependent variable using a given set of independent variables. Additionally, it calculates the probability of belonging to the target class, instead of simply classifying it.

The analyzed parameters are:

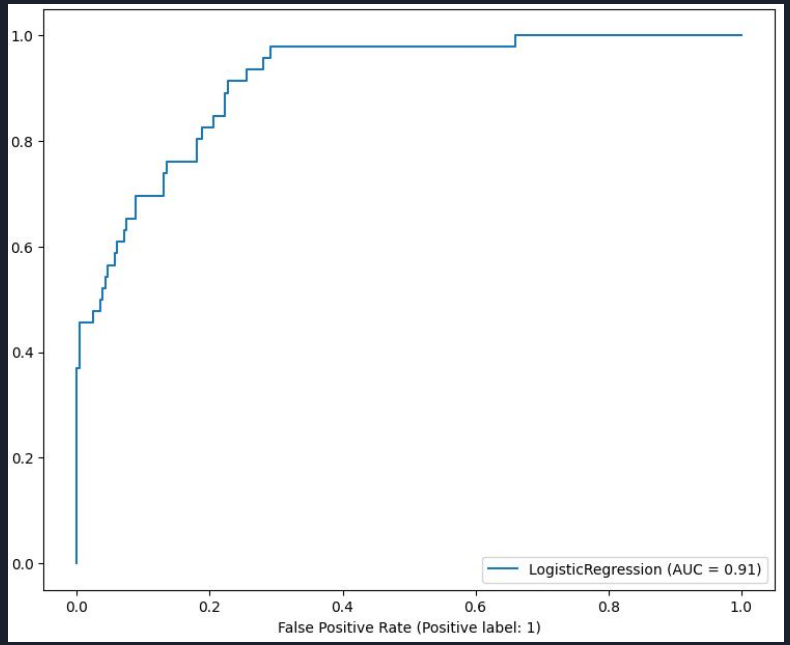
C - Inverse of regularization strength. Regularization makes the model more flexible, as to avoid overfitting.

solver- Algorithm used in the optimization problem.

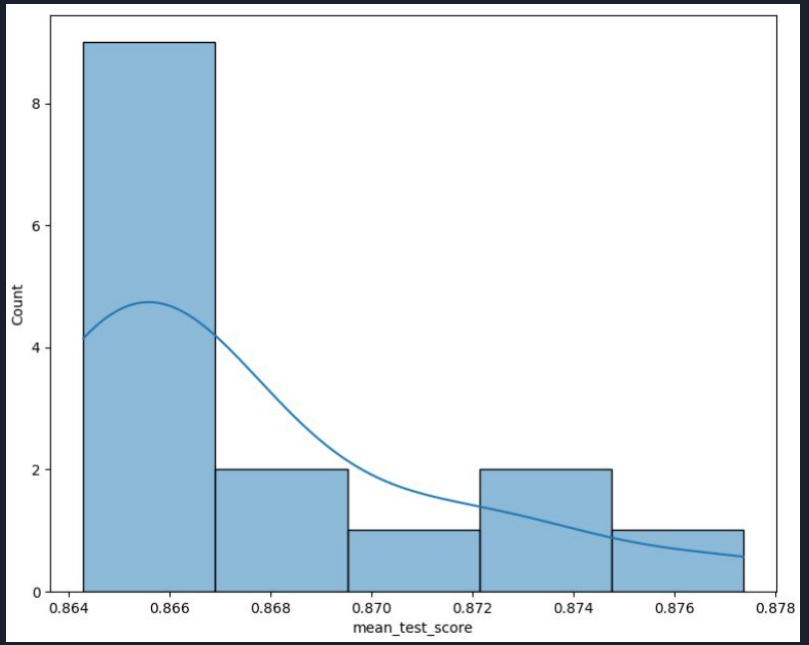
Best Parameters: **{'C': 1000, 'solver': 'newton-cg'}**

Analysing the results, this classifier has a high variance resulting in very good and very bad predictions. Hence, it's not reliable.

Parameter Tuning - Logistic Regression



Best parameters ROC-AUC



Grid Search ROC-AUC distribution

Parameter Tuning - Artificial Neural Network

The Multi-layer Perceptron algorithm is capable of solving non-linear classification problems. Consequently, it's a good alternative to Logistic Regression, as the later is restricted to linear problems. Due to its diversity of configurations, it requires parameter tuning.

The analyzed parameters are:

activation - Activation function for the hidden layer. One of {*identity*, *logistic*, *tanh*, *relu*}.

solver- Solver for weight optimization. One of {*lbfgs*, *sgd*, *adam*}.

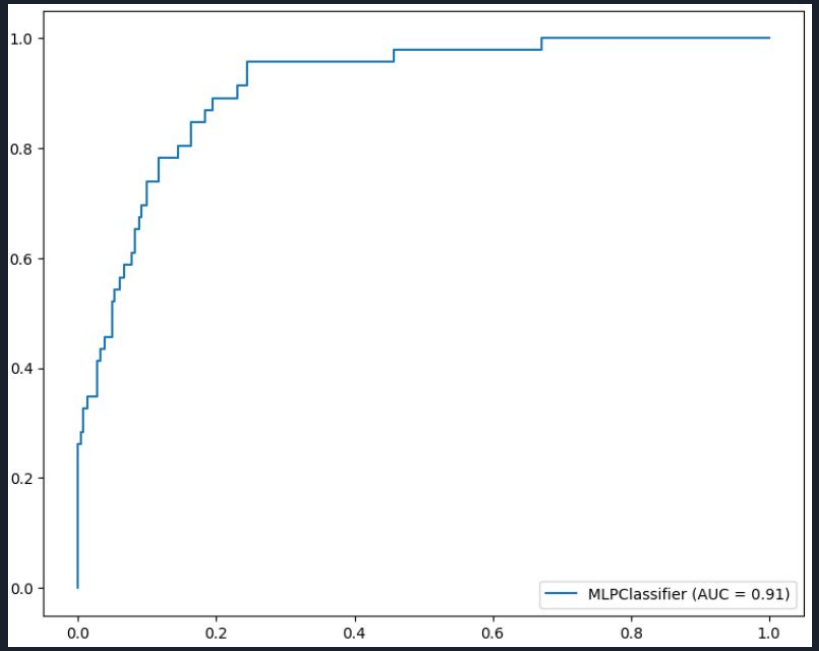
alpha - Strength of the L2 regularization term.

learning_rate - Learning rate scheduled for weight updates. This is only used with the *sgd* solver.

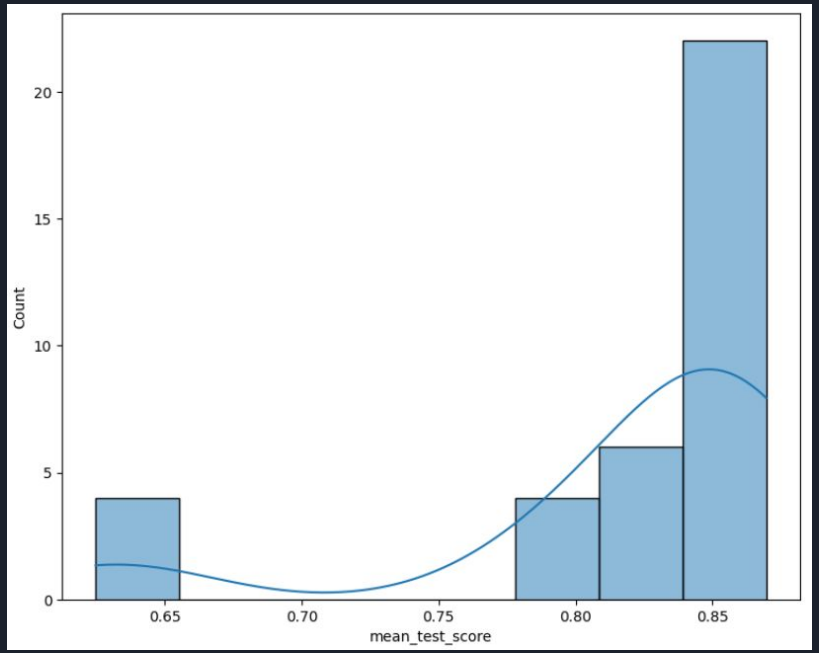
Best Parameters: **{'activation': 'tanh', 'alpha': 0.0001, 'learning_rate': 'constant', 'solver': 'adam'}**

The Artificial Neural Network provided one of the best results, falling short only to the *Random Forest* algorithm. This is an indicator that our predictive task is more complex than a linear problem and could benefit from more data refinement.

Parameter Tuning - Artificial Neural Network



Best parameters ROC-AUC



Grid Search ROC-AUC distribution

Results Analysis

As expected, Random Forest was the top performer on this predictive task, since it is the industry's most used algorithm for the identification of loan risk.

Tuning the parameters of the classifier had a significant influence on the final results, but the main impact was caused by the features used in the model.

That said, meticulous data analysis and feature engineering are essential to produce good results with these models, especially in a domain-specific work like this one (identifying clients that wouldn't pay a loan).

The features that returned the best results were: is_co-owned, gender, age, amount, payments, balance_min, a_avg_salary, mean_3m_influx, frequency_monthly issuance, frequency_weekly issuance, sanctions, interest_mean and frequency_issuance after transaction. Accordingly, banks should give special attention to these properties to guarantee their max efficiency.

This is not a big surprise, since these features have significant business meaning that relates to the bank, its clients, and the inherent financial value.

Results Analysis

We kept some features that didn't have a high linear correlation with the target. Interestingly, this led to better results, because this task is more complex than a linearly-separated problem. This conclusion is also supported by the disparity of the results between the linear and non-linear classifiers presented before.

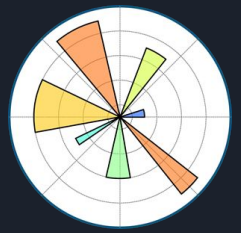
Analysing the engineered features, we saw that co-owned accounts of younger people with monthly issuance (which could mean they have a stable income and are less prone to financial issues) lead to a high loan success rate.

We also noticed that the mean_influx and interest_mean features helped determine whether a loan will or won't be paid, leading to better results.

Precision is not a suitable evaluation metric for the loan risk problem, since nearly 86% of loans were paid in the training dataset. Recall and ROC-AUC are much better indicators of the performance of a classifier.

Technologies

Pandas



Member Contribution

- Bruno Rosendo - 1/3
- João Mesquita - 1/3
- Rui Alves - 1/3

References

- https://matplotlib.org/
- https://www.javatpoint.com/classification-algorithm-in-machine-learning
- https://pandas.pydata.org/
- https://scikit-learn.org/stable/index.html
- https://towardsdatascience.com/how-to-train-test-split-kfold-vs-stratifiedkfold-281767b93869
- https://towardsdatascience.com/smote-fdce2f605729
- https://towardsdatascience.com/the-right-way-of-using-smote-with-cross-validation-92a8d09d00c7
- Curricular Unit's (Machine Learning) lecture slides