Bank Loan Prediction

MEIC - Aprendizagem Computacional 2022/2023

Duarte Sardão - up201905497@up.pt | Gabriel Martins - up201906072@up.pt | Miguel Lopes - up201704590@up.pt | Sérgio Estêvão - up201905680@up.pt

Domain Description

Dataset Composition

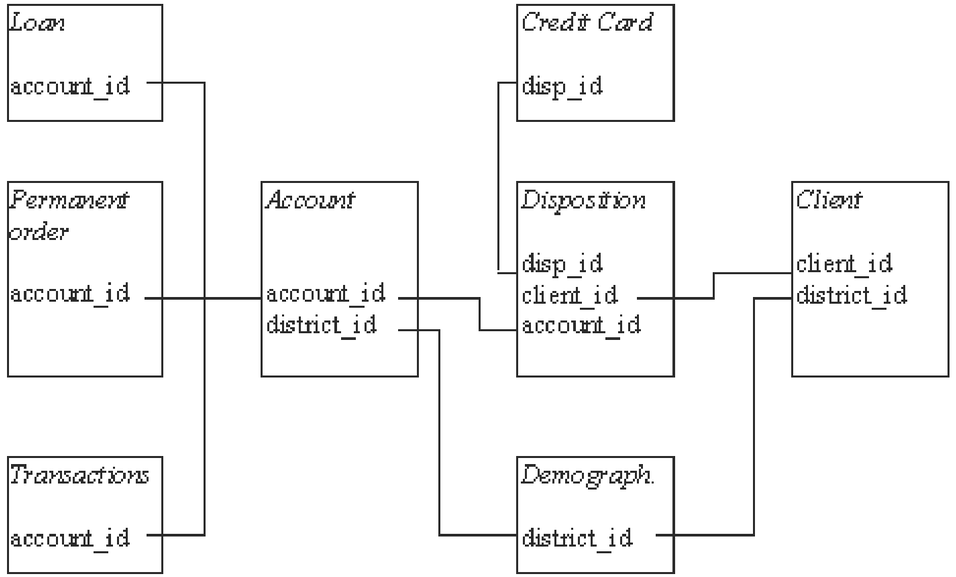
Dataset

The dataset is composed of a series of information describing a Czech bank's activity during the 90s. This includes account, client, credit cards, transaction, loan information, and information regarding the districts where the bank's clients reside.

The records of the Czech bank contain the following entries:

- 4500 accounts
- 5369 clients
- 5369 dispositions
- 396 685 transactions
- 328 loans
- 177 cards
- 77 districts with demographic data

Relational Model of the Dataset

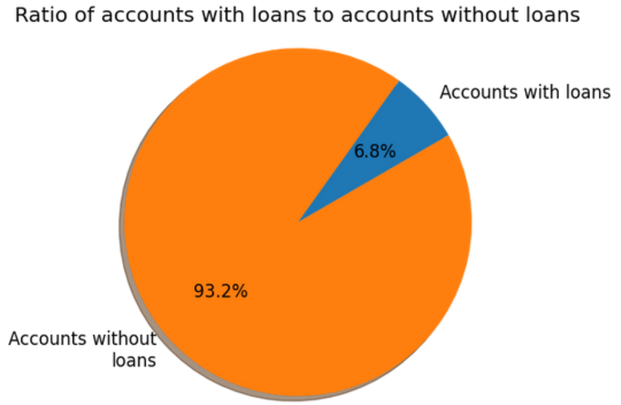


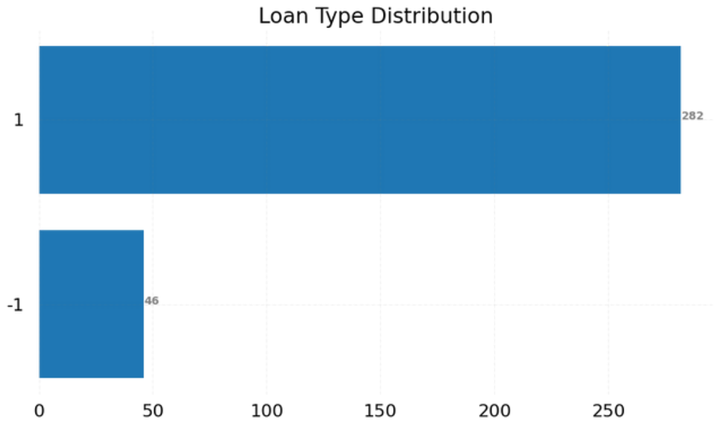
Exploratory Data Analysis

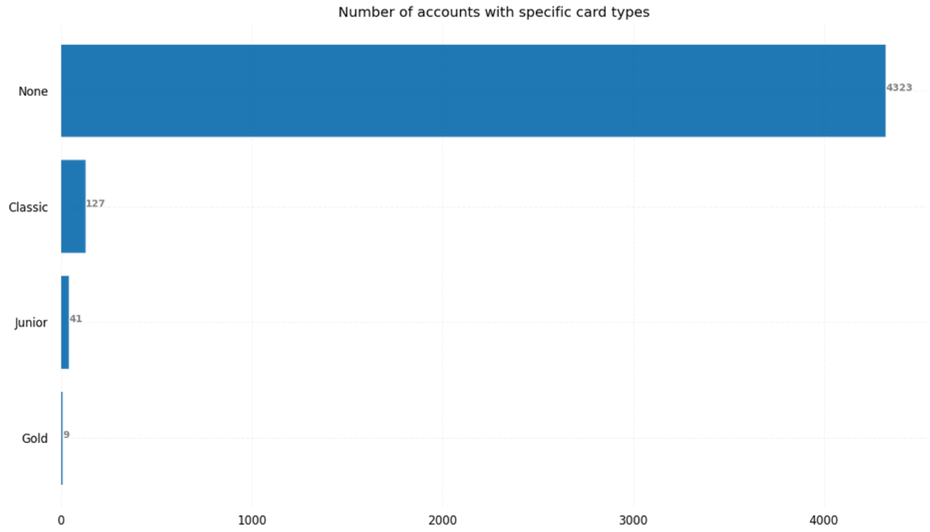
Jupyter Notebook (Python3 & Libraries, e.g. Matplotlib, Pandas, Seaborn), Excel

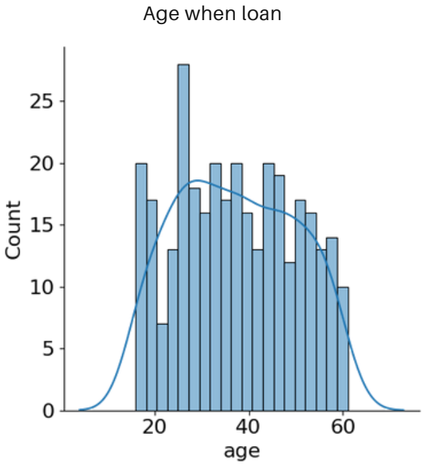
Findings

- The majority of the accounts didn’t have loans, which meant that most of the data couldn't be used
- The loan distribution is unbalanced. Only 14% of the loans were ‘-1’, which hinders the prediction model results
- The great majority of the accounts didn’t have cards associated with them
- The age at which the client takes a loan follows approximately a normal distribution, as expected. The most common age group is between 24 and 40 years old







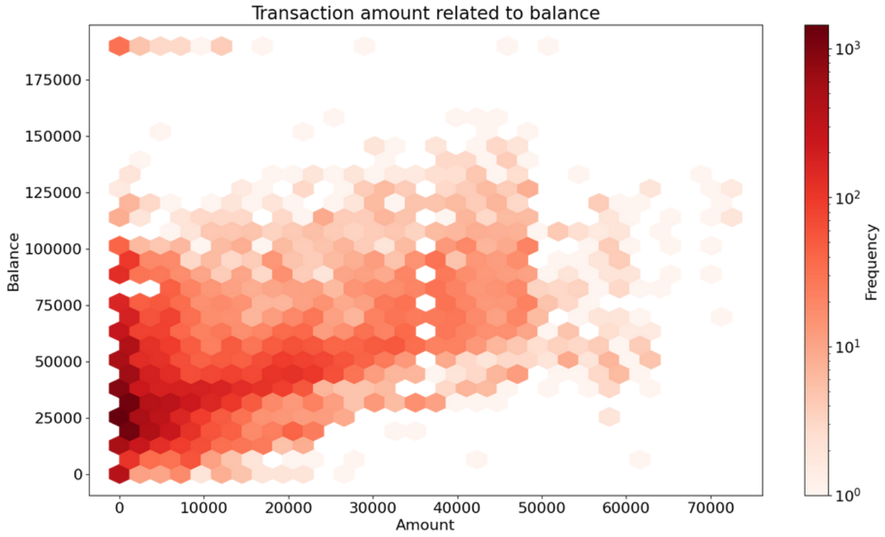


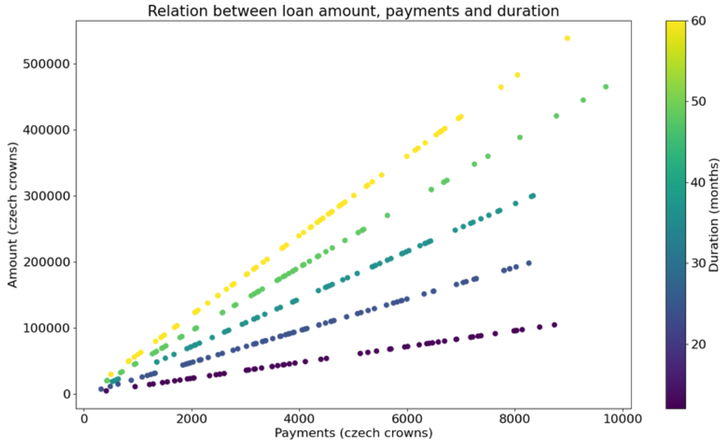
Exploratory Data Analysis

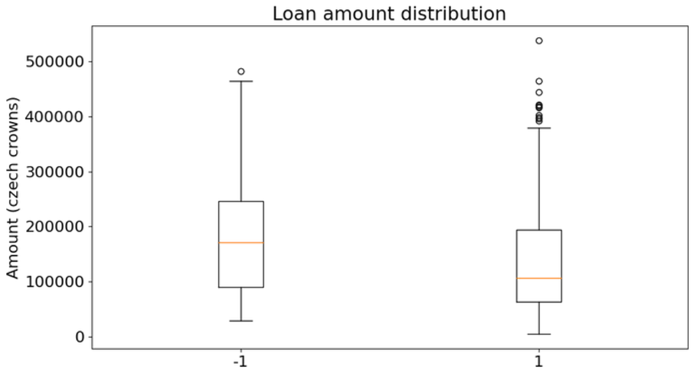
Jupyter Notebook (Python3 & Libraries, e.g. Matplotlib, Pandas, Seaborn), Excel

Findings

- The transactions follow a downward trend. Most are of lower amounts from accounts with a lower balance
- Some outliers can be seen: transactions of lower amounts from accounts with a higher balance and transactions of higher amounts from accounts with a medium balance
- A gap in the transactions of amounts rounding 35.000 <-> 37.500 Kč, no apparent cause
- As expected, a correlation between the loan amount, duration, and payments
- No interest rate is applied by the bank (loan amount = payments * duration)
- The loan amounts related to the paid loans (1) were, on average, smaller and had a higher amount of outliers
- No relevant correlation was found between gender and loan results or balance amounts







Problem Definition

Business Understanding and Data Mining Goals

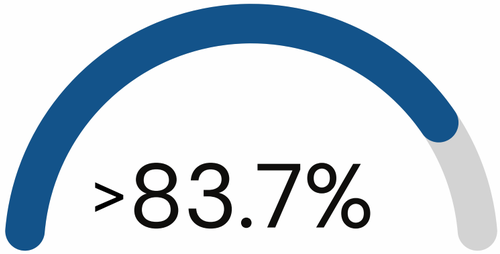
The bank wishes to increase their operating income by using data mining to predict the outcome the loans they grant

1. **Reduce the number of employees and time spent on loan risk analysis**
2. **Increase the number of loans that are paid in full**
3. **Decrease the number of loan defaults**

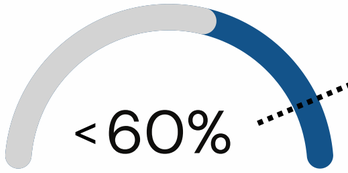
16.3%
Current default rate*

≈8 million Kč
Amount lost from defaulted loans*
(in Czech Korunas)

Our goal is to **decrease the number of actual defaults** (priority), by achieving a higher NPV value, whilst at the same time reducing the number of loans incorrectly marked as defaults (false positives), by achieving a lower FPR value



Negative Predictive Value (NPV)**



False Positive Rate (FPR)**

60%
average denial rate for credit cards in the US.

* Rate calculated with the data provided initially | ** Considering loan defaults as positives and payed loans as negatives

Data Preparation

Data Preparation Pipeline - Main Operations Performed

Removing information that has no relation to loans

Although we were given a substantial amount of data, a lot of it was about clients that had no loans nor any relation to the loan bearer

Extract the client's birth date and gender from the "birth_number"

The column birth_number contained information regarding both the birthdate and the client's gender

Extracted new parameters from several of the datasets

Created columns like crime rate, recent balance, household payments, insurance payments, sanction payment counter, to name just a few

Converted dates into time intervals, considering "1999-01-01" as the current date

Since most classification models don't accept a date as an input type, we converted dates like, the client's birthdate, the account creation date or the date of issuance of the credit cards into a time delta (age of the account/card)

Using K-Nearest Neighbor algorithm to fill in empty values

Some values related to the number of crimes committed in 95 and 96 were missing, so we used the K-Nearest Neighbour algorithm to fill the gaps

Determining the correlation between values and removing redundant features

Having multiple values with high correlations between them can have a negative impact on the training of the classification model, for that reason we analyzed the values of both **Pearson** and **Spearman** correlations and removed values with a correlation value above 0.95

Combined all the data into a single dataset

Merged the extracted data into the Clients data frame and followed this by merging the new Clients dataset with the Loans and District datasets. After this, it was also added 3 new columns to the dataset (the age of the client at the time of the loan, the age of the account at the time of the loan, and the age of the card at the time of the loan)

Data Preparation

Data Preparation Pipeline - Main Operations Performed

Converting columns with categorical attributes into integers

Columns like sex, name, and others that had string attributes were converted into integers to make it possible to use them in algorithms

Remove unnecessary or ineligible information

Remove columns that were either used for merging porpuses (ID values) or contained data that was too sparse to be used

Outliers removal

Having a dataset that combines all the initial datasets, we define a function for removing outliers, based on the 1.5x inter-quartile rule. After that, we apply it to a selection of columns that follow a near-normal distribution - the amount, average_balance_fluctuation_per_month, and avg_balance columns

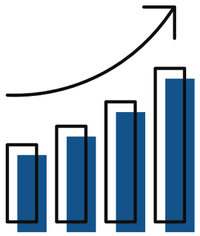
Experimental Setup

Classification Model Pipeline

1. **Standardizing the features** by retrieving the mean and scaling to unit variance (standard score given by z=[x - u]/s, where u is mean and s is std deviation)
2. Addressing the unbalance in the dataset using **oversampling** (we tested SMOTE, SMOTENC, SMOTETOMEK and we also tested undersampling)
3. Used a meta-transformer to **select features** based on their importance weights (with a Linear Support Vector Classification estimator)
4. **Testing different classification models** (SVC, Random Forest, Logistic Regression and Naives Bayes) **with cross-validation**
5. **Hyperparameters are tuned** by doing a cross-validated search over parameter settings
6. **Analyzing the results** - with a main focus on the AUC values but also with the help of plots (precision-recall curve, ROC curve) and the confusion matrix
7. **Selecting our final settings** for classifying the provided test dataset

Results

Results in the Experimental Setup and Kaggle Competition



When testing the data in the experimental context we found that the changes we made reflected very little in the scores and metrics we used to evaluate the performance, so it was hard to determine which algorithm would perform best in the Kaggle competition. The table on the right includes our best results in both the experimental setup and the Kaggle public tests.

Classification Model	Best Results				
	AUC Score Experimental Setup	AUC Score Kaggle Public Tests	AUC Score Kaggle Private Tests	Negative Predictive Value⋆	False Positive Rate⋆
Random Forest	0.8696	0.9376	0.8971	89.7%	6.0%
Logistic Regression	0.8941	0.9481	0.8390	98.7%	11.9%

98.7%

Our best Negative Predictive Value used SMOTE-Tomek for balancing the dataset, outlier removal using the 1.5 IQR method, feature selection, hyperparameter tuning, and Logistic Regression as the classification model. Both this value and the associated False Positive Rate exceed the goals set out in our problem definition

⋆ Values calculated in the experimental context

Results

Result Interpretation

Worst Performers

Undersampling

Undersampling proved disappointing, with AUC scores averaging between 60% to 70%, when this method was used

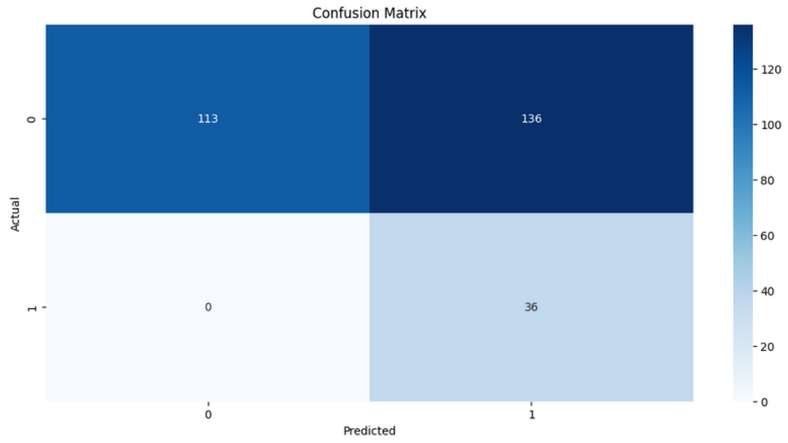
Naives Bayes

Naives Bayes produced consistently mediocre scores when compared to its counterparts

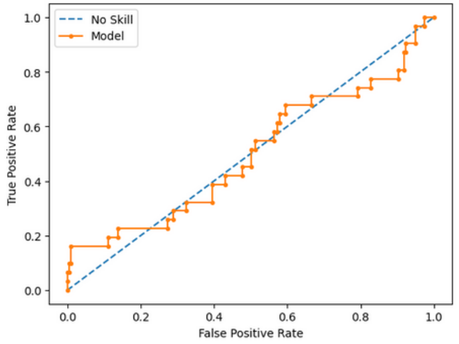
Biggest Impact

Hyperparameter Tuning and Feature Selection

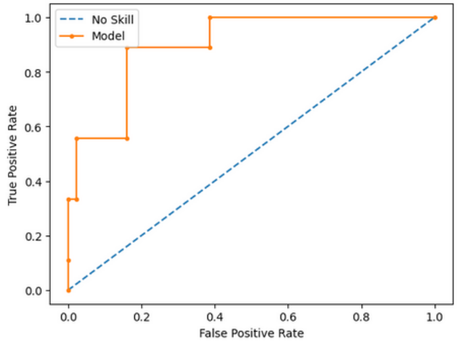
As can be seen from the ROC graphics shown below, when using hyperparameter tuning and feature selection, the quality of the results was higher - there is a clear increase in the value of the area under the curve (AUC)



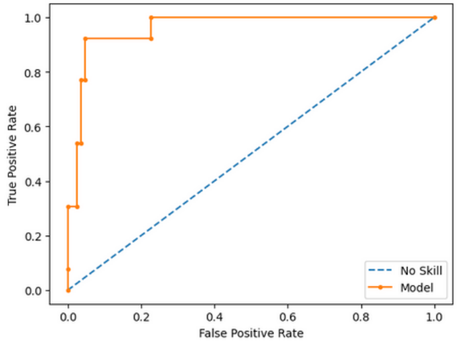
Confusion Matrix - Naives Bayes w/ Undersampling



ROC Curve - Logistic Regression



ROC Curve - Logistic Regression w/ SMOTE and Hyperparameter Tuning



ROC Curve - Logistic Regression w/ SMOTE, Hyperparameter Tuning and Feature Selection

Conclusions, Limitations and Future Work

Conclusions

- Oversampling with SMOTE didn't perform as expected on the private Kaggle tests, even though it had a significant influence on the improvement of the results in the public Kaggle tests and the experimental setup
- Hyperparameter tuning and outlier removal had the most substantial effect on the results
- The goal established at the beginning of this project was accomplished as a higher NPV value was achieved
- We were surprised to find that the Logistic Regression algorithm that had the best AUC score in both the experimental setup and the public Kaggle tests did not perform as well in the private tests. On the other hand, the random forest algorithm had an improvement in performance that was not predicted in the public and experiment setup tests

Limitations and Dificulties

- The dataset was fragmented and not very useful as only a minority of the clients had loans, and even in this group, the data was unbalanced. In the end, we ended up working with only a number around 300 entries
- The initial description of the dataset was misleading as it contained wrong information

Future Work

- Explore new classification models
- Closer analysis of the datasets to see if there is any relevant information left that can be extracted