To loan or not to loan

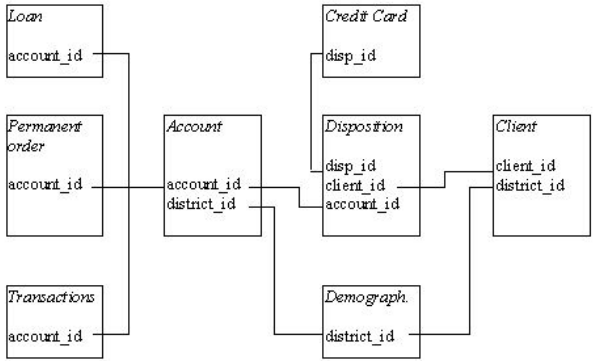
Data Mining Project

Beatriz Mendes - up201806551@up.pt
Henrique Pereira - up201806538@up.pt
Hugo Guimarães - up201806490@up.pt

Domain Description

Each table of the schema is correspondent to one of the provided files

- **4500 accounts**, that can be accessed by clients, which can be owners or disponents, with different types of credit cards
- Demographic data about the district, regarding the bank (accounts) and the clients
- **426885 transactions**, characterised by date, type, amount and operation.
- **682 loans**, 328 of which with known loan status

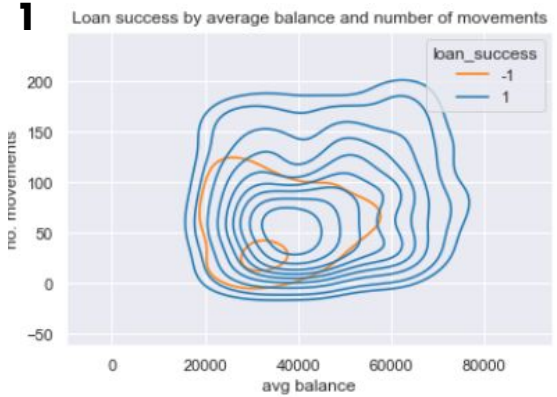


Exploratory Data Analysis

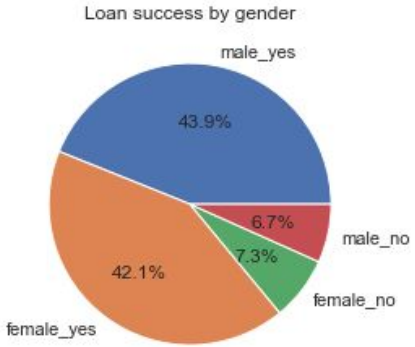
1. There is a very positive correlation between the pair of attributes *no. movements* and *avg balance* and the success of a loan.
2. Gender is not correlated to loan success
3. Loan success rate is lower when the loan amount is higher
4. There is no correlation between account age on loan and loan success
5. Over 80% of clients that ask for loans have no credit card
6. Whenever a client has a credit card, their loan is always successful
7. From the distribution of the status on the loans where it is known - 86% ended successfully - we can conclude that this attribute is unbalanced.
8. No client has more than one loan
9. No transactions occur on an account after a loan is made
10. The condition amount = payment * duration is always true

Exploratory Data Analysis

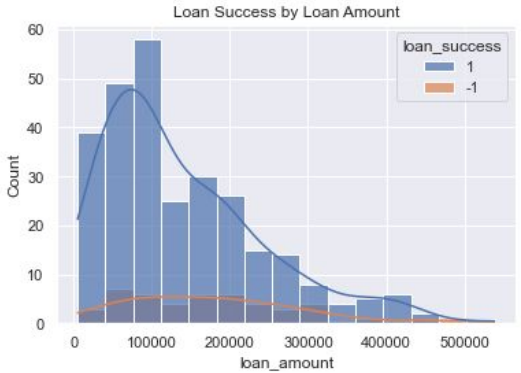
1



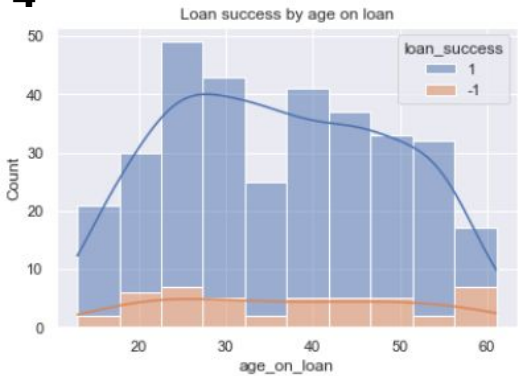
2



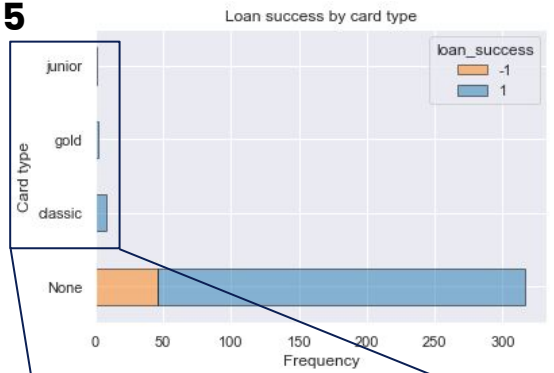
3



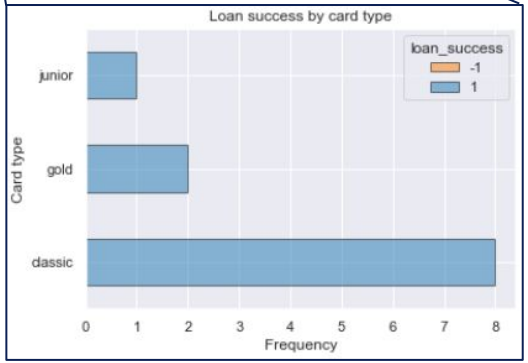
4



5



6

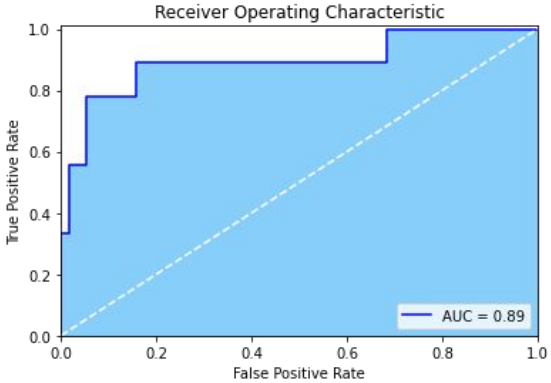


Problem Definition

Goal: Predict when loan will be unsuccessful (-1)

Problem Type: Classification problem where the variable to be predicted is either 1 or -1 (successful or unsuccessful, respectively)

Evaluation: Area Under the Curve (AUC) is a performance measurement for the classification problems at various threshold settings. It shows the rate of false positives and true positives, giving an insight on the performance of the model.



Data Preparation

Data Cleaning

- Transform dates into the format DD-MM-YYYY
- Replace empty values in *no. of commited crimes /'95* and *unemployment rate /'95* with the values from the next year ('96)
- Deal with categorical classes:
    - Label encoding to numerical in the same column
    - Creating dummy columns with binary (0 or 1) values for each category
- Deal with outliers (no outliers were detected, hence none were removed)
- Drop columns (**feature selection**) from *transactions* that have too many missing values (more than 80%)
- Replace missing values in the *operation* column of *transactions* with the most common occurrence.
- Consider only the owner for each account (drop disponents)

Data Preparation

Feature Engineering

- Client birthdate and gender calculated from *birth_number*
- Client age on loan calculated from birthdate and loan date
- District crime growth and unemployment growth between ‘95 and ‘96
- Total crime in ‘95 and ‘96
- Transactions statistics:
    - Minimum, maximum and average balance after transaction
    - Minimum, maximum and average transaction amount
    - Total number of transactions from each account

Experimental Setup

After the pre-processing of the data, the project has the following steps, for any algorithm:

- Flag Selection
    a. Boolean values - OVERSAMPLE, DUMMIES, CATEGORY_ENCONDING, MIN_MAX_SCALER
    b. Numerical values - SPLIT_RATIO, N_COLUMNS, N_SPLITS
- Prediction
    a. Select the most influencing attributes using the **selectKbest** function from sklearn (number of selected attributes can be changed by a flag)
    b. Split the datasets, using one of the following options:
        - **train_test_split** (default 80/20, changed by the flag)
        - **stratifiedKFold** - Split the data into K folds (value can be changed by a flag)
    c. Apply oversample technique using **SMOTE** (enabled/disabled by a flag)
    d. Apply **grid search** to evaluate the best parameters for the algorithm
    e. Apply the **chosen algorithm** with the obtained grid search parameters to get the predicted values
    f. Evaluate the model Score and AUC curve

Results

- When using the ***RandomForest*** algorithm, an average score of 86% was obtained
- The final result of the *Kaggle* competition was slightly lower, with approximately 84%
- After several experimentations, we have concluded that the ***Oversampling*** technique has resulted in better performances
- Several algorithms were used, such as ***RandomForest***, ***Logistic Regression***, ***Decision Tree*** and ***KNN***, and better results were obtained by using the first two, with the ***Random Forest*** achieving better predictions in the public leaderboard, while the best result in the private leaderboard was obtained through the ***Logistic Regression***
- The two splitting methods used belong to the *sklearn* library: ***train_test_split*** and ***stratifiedKfold***. The first one provided higher maximum scores, but the second obtained more consistent results. It is plausible to conclude that the first method was causing an overfit, maybe due to the provided split ratio.

Conclusions, Limitations and Future Work

- Conclusions
  - Feature Engineering and selection were the most important steps that lead to a better AUC curve
  - There are many algorithms that can be applied as a classification model, and each has their different parameters and require different pre-processing
  - We cannot blindly trust in an AUC curve drawn without knowledge about what are doing, because we can easily enter a state of overfitting

Conclusions, Limitations and Future Work

- Limitations
  - The loan training dataset was too small and imbalanced, which inadvertently lead to some increased overfitting, making us believe that our result were better than they actually were
- Future Work
  - Clustering
  - More feature Engineering/Selection - We believe that adding more features to our project may lead us to better results. The transactions table can be further explored.
  - Experiment other algorithms
  - Try different parameters in each algorithm

Annexes