Data Mining Case Study

ECAC 2020,
DM-G14

Eduardo Silva - up201603135@fe.up.pt
Francisco Ferreira - up201605660@fe.up.pt

Domain Description

A particular bank wants to improve their services by:

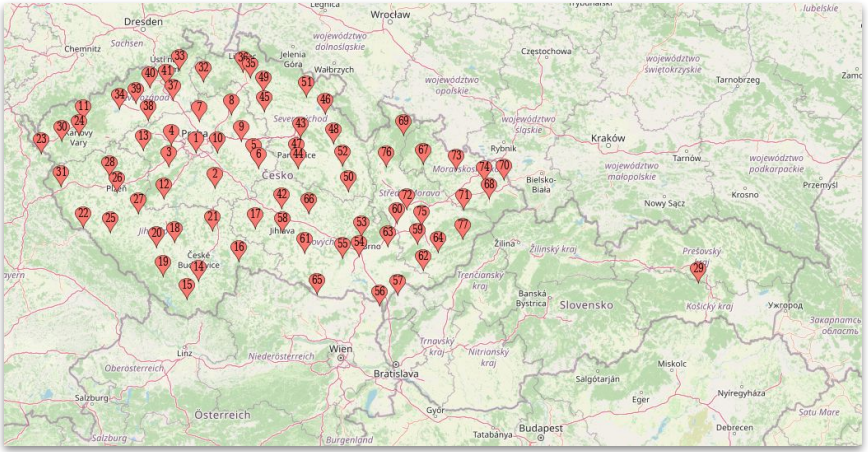
- Offering additional ones to “good” clients.
- Carefully watching “bad” ones so as to minimize losses.

The bank stores information relative to its customers, namely regarding their demographic data, dispositions and associated credit cards, accounts, previous loans and transactions.

The bank managers hope to improve their understanding of customers and seek specific actions to improve services by exploring this data.

For that purpose, it is to be developed a machine-learning tool able to not only perform exploratory analysis but to predict a loan’s outcome with high performance levels.

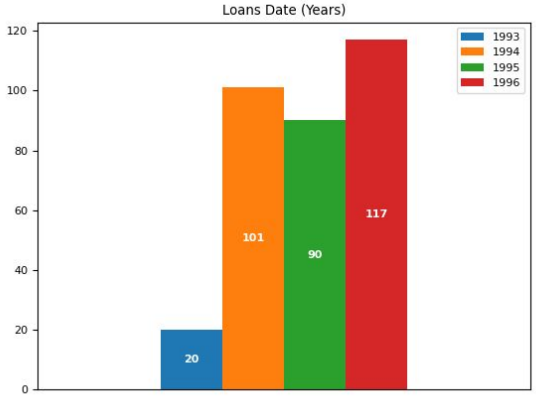
Exploratory Data Analysis

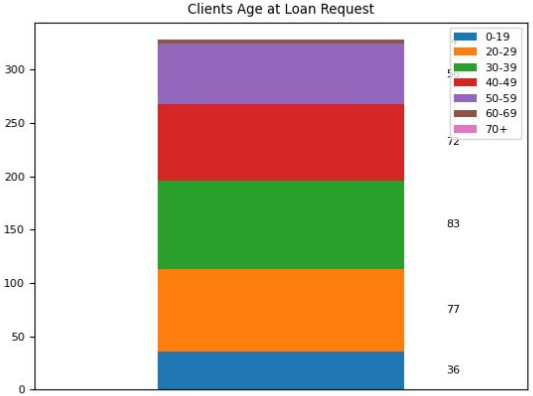


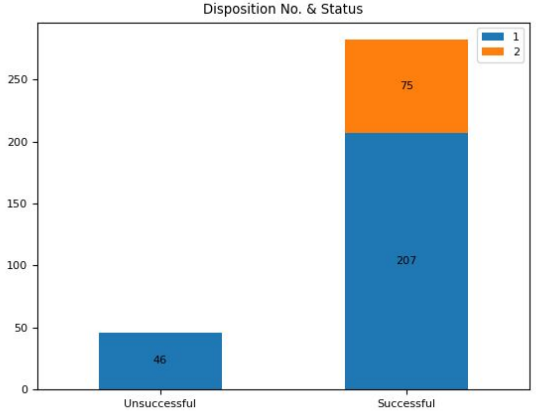
Main findings:

- All districts are in Czech Republic.
- Dates range from:
  - 1993 to 1997 in account data;
  - 1993 to 1996 in card training data and from 1996 to 1998 in the test data;
  - 1995 to 1996 in district data;
  - 1993 to 1996 in loan and transaction training data and from 1997 to 1998 in the loan test data;
  - 1995 to 1998 in transaction test data;
- All clients have only one account.
- All clients have only one associated disposition.
- The maximum number of dispositions associated with an account is 2.
- The most common age range for clients that request a loan is 30-39 years old.
- There is an almost equal proportion of male and female clients and loan takers

- Every account which has issued a loan and has 2 dispositions associated had success in paying said loan.
- Loans that have ended successfully make up almost 86% of all the loans.
- The maximum number of loans associated with an account is one.
- The most popular card type is “Classic”.
- Just over half of the transactions have missing K symbols.
- The transactions with missing operations correspond to those which have “Interest credited” as their K symbol.







Predictive Task - To loan, or not to loan

Problem Definition

Business Goal: To predict whether a loan request will end successfully or not, so as to make an informed decision on which action to take: to loan or not to loan.

ML Goals: Given the available data regarding past loans and other associated information, build, test and evaluate several pipelines, models and/or techniques to classify a loan's outcome in one of two categories: successful (negative class) or unsuccessful (positive class) with an acceptable performance. As such, it can be considered a binary classification problem.

Adopted methodology: CRISP-DM:

1. Business Understanding - See Domain Analysis and Business Goal.
2. Data Understanding - See exploratory data analysis and annex.

Data Preparation

1. Calculation of statistical properties of data to be used for filtering and missing values replacement, such as average, mode, minimum, maximum and quartiles.
2. Data cleaning:
    a. Clients: Normalize date of birth and add extra column with gender information.
    b. Dispositions: Change disposition type (“owner” or “disponent”) to binary form.
    c. Districts: Replace “?” and missing values in the ‘95 crime and unemployment rates columns with their respective average.
    d. Transactions: Replace some missing values with new categories (see annex).
3. Data Normalization (when using K-NN and SVM): Scaled column values between 0 and 1, so that the maximum absolute value of each feature is scaled to unit size.
4. **Outlier removal** (optional) in the loans and transactions files.
5. Data aggregation based on ID matches (loans, accounts, dispositions, ...)
6. Feature Engineering
7. Removal of **correlated predictive attributes** (optional)

Data Preparation: Feature Engineering

The generation of new features was paramount in the models' predictive success. Around 55 new features were introduced, however, not every single one proved useful. Some of the most interesting (both positively and negatively) were:

High Importance

- Average sanction value.
- Average monthly income.
- Number of times the account had a negative balance.
- Last balance registered before the loan request.

Low Importance

- Number of dispositions associated with an account.
- Number of each card type associated with an account.
- Account owner's card type.
- Account owner's number of associated accounts.

Experimental Setup / Modeling

1. Load data into memory.
2. **Stratified train** / test split of dataset.
   OR
   Split data based on loan request date (**Sliding Window** (SW)).
3. Balance training dataset using **Borderline SMOTE** to upsample minority class, followed by **random undersampling** of the majority class until both are equally balanced.
4. Hyperparameter **random** and / or **grid search** (optional).
5. Model generation, **repeated 10-Fold cross-validation** and training.
   - Depending on the algorithm used, **feature selection** was either performed manually before this step or automatically at this point when supported by the algorithm.
6. Testing and evaluation.
   - Main evaluation metric used was **AUC-ROC**, however, when these values were very close between experiments, attention was paid to other metrics, mainly **F1** since it represents a combination of both precision and recall.
   - The accuracy was not given too much importance during evaluation due to the class imbalance.

Results / Evaluation

	Training					Test				
	Accuracy	Precision	Recall	F1	**AUC**	Accuracy	Precision	Recall	F1	**AUC**
Random Forest	0.90	0.86	0.95	0.90	0.978	0.91	0.95	0.95	0.95	0.941
Decision Tree	0.86	0.87	0.89	0.88	0.880	0.92	0.96	0.95	0.96	0.911
SVM	0.86	0.86	1.00	0.92	0.892	0.86	0.86	1.00	0.93	0.844
K-NN	0.78	0.84	0.70	0.76	0.827	0.68	0.97	0.65	0.78	0.835

Note: Throughout this document, the positive class is “Successful Loan”, or 1, and the negative class “Unsuccessful Loan”, or -1.

Results / Evaluation

- The best model overall was the **Random Forest** based one, achieving the highest **AUC** score in both train and test sets while using:
  - **Borderline SMOTE** followed by **Random Undersampling** for data balancing.
  - Automatic Feature Selection (square root of n).

- The **Decision Tree** classifier achieved the best **F1** score on the test data set due to the better combination of precision (just 0.01 higher than the Random Forest one) and recall while using **SMOTE** only for class balancing.

- The **SVM** model had the highest **recall** of 1.0 on the test set, however it also had the lowest precision there, showing a bias towards the positive class, all of this while using **random undersampling** only for class balancing.

- The **K-NN was** the worst performing model in general, although it achieved the best **precision** on the test data set, but an abysmal (and worst) recall (bias towards the negative class), while using **random undersampling** only for class balancing.

Conclusions, limitations and future work

- Overall, a fairly good performance was achieved, with the best model (**Random Forest**) registering an AUC of **0.941** on the test dataset. As such, both ML and business goals were achieved.
- Overfitting was carefully avoided by the close monitoring of produced metrics during training and testing, namely cross-validation average scores.
- Limitations:
  - Since our pipeline was manually assembled and different operations needed to be performed for each algorithm, it was given special focus to those that were expected to generally perform well on this problem.
  - Due to time constraints, model improvement was largely based on performance improvement. In future work, other strategies might prove useful to explore.
  - Limited data also played a role, with the risk of overfitting being substantially high. As such, operations that further diminished the number of training examples were minimized.
- Future Work
  - Train separate model to predict missing K values (multiclass classification).
  - Test and evaluate other algorithms and architectures, such as deep learning networks.

Annex

Project Management

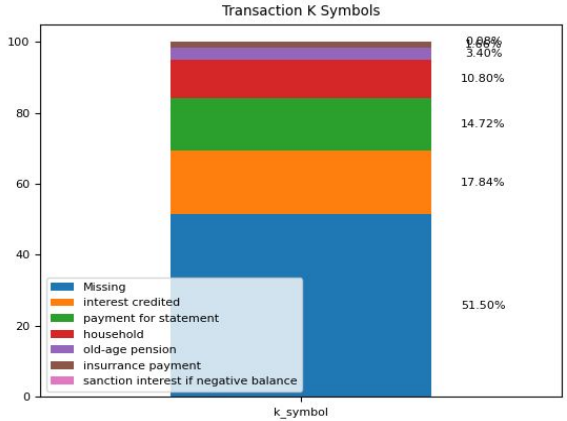
As mentioned in the problem definition, for the predictive task the adopted methodology was **CRISP-DM**. As such, it was planned early on to heavily focus efforts on the **data understanding and preparation** phases. A full pipeline was soon established afterwards in order to begin modeling simpler classifiers and evaluating early results.

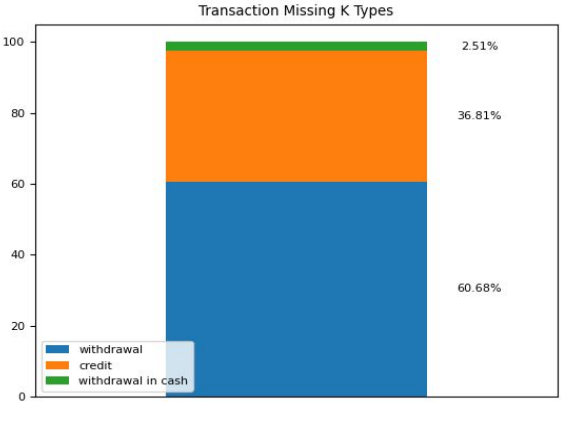
From this point, a plan was drafted weekly in order to better coordinate development focus, which, for the most part, was concentrated in **feature engineering and model optimizations**.

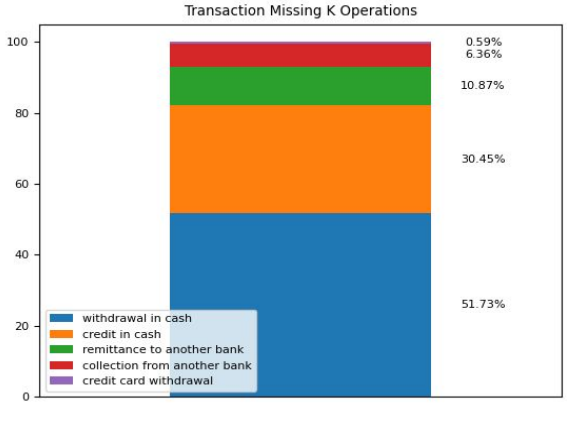
Data Cleaning: Transactions

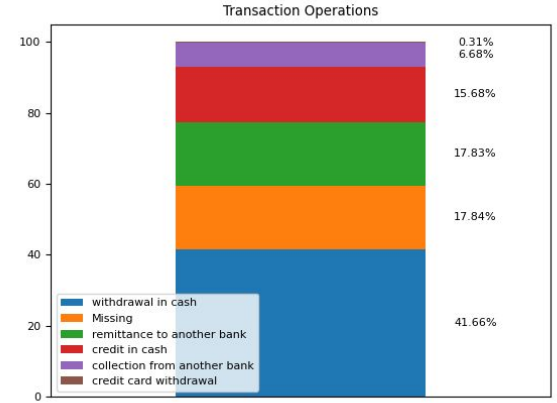
Missing (relevant) values:

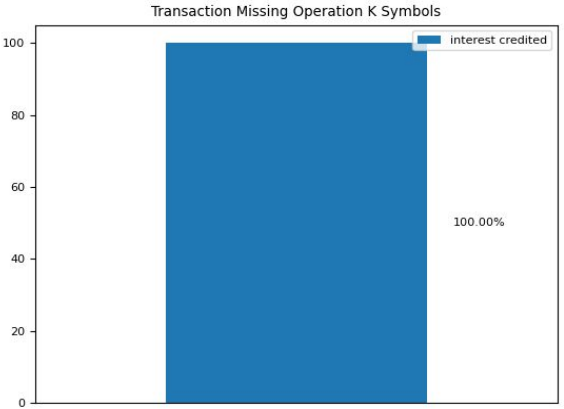
- K Symbol
  - **51.50%** of all transactions are missing k symbols.
  - The transactions with missing K symbols are of almost every type and operation.
  - Doesn't make sense to replace with mode or to remove these objects.
  - **Solution:** Create new 'missing' category.
- Operation
  - **17.84%** of all transactions are missing the operation.
  - Operation mode is "withdrawal in cash", however, all transactions with missing operations are of credit type.
  - All transactions with a missing operation have "interest credited" as their K symbol, and all transactions with 'interest credited' K symbol have a missing operation: **full match**,
  - **Solution:** Create new 'interest' operation since there are no other matching operations.

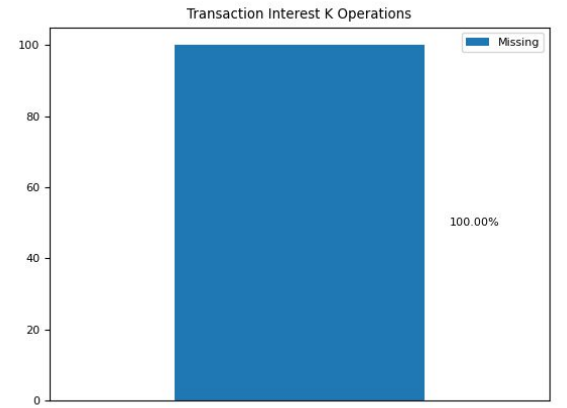

Transaction K Symbols
Missing
interest credited
payment for statement
household
old-age pension
insurrance payment
sanction interest if negative balance
0.88%
1.55%
3.40%
10.80%
14.72%
17.84%
51.50%
k_symbol


Transaction Missing K Types
withdrawal
credit
withdrawal in cash
2.51%
36.81%
60.68%


Transaction Missing K Operations
withdrawal in cash
credit in cash
remittance to another bank
collection from another bank
credit card withdrawal
0.59%
6.36%
10.87%
30.45%
51.73%

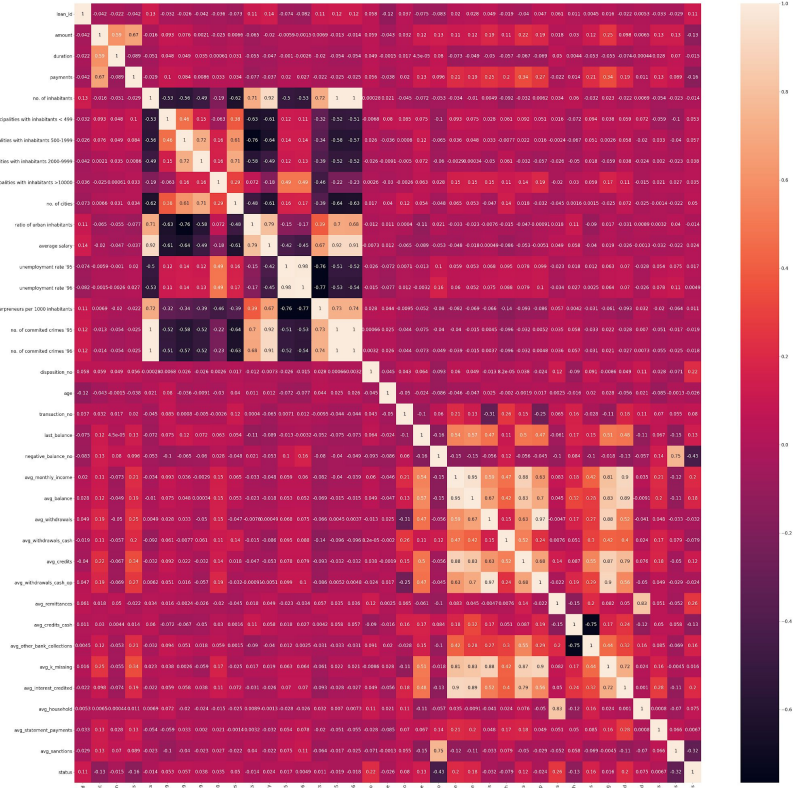

Transaction Operations
withdrawal in cash
Missing
remittance to another bank
credit in cash
collection from another bank
credit card withdrawal
0.31%
6.68%
15.68%
17.83%
17.84%
41.66%


Transaction Missing Operation K Symbols
interest credited
100.00%


Transaction Interest K Operations
Missing
100.00%

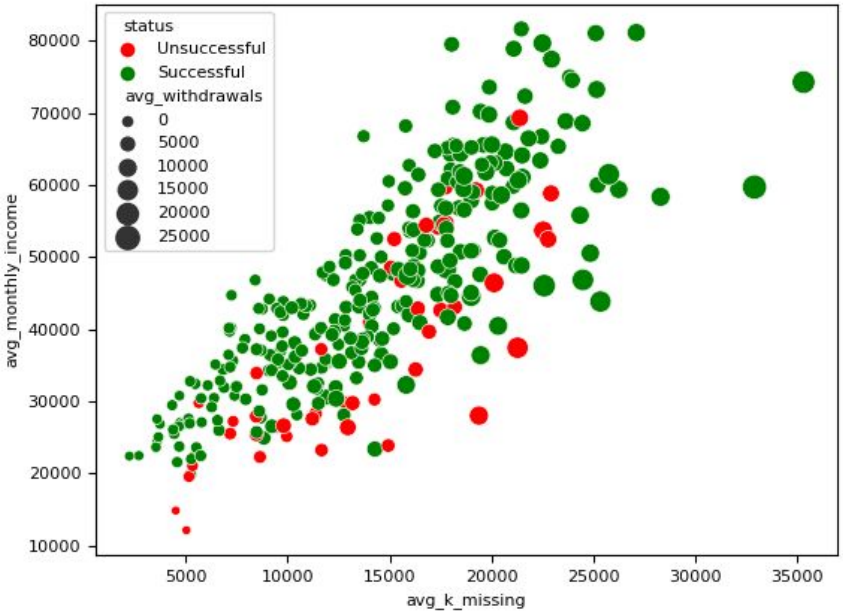
Data Preparation: Correlation Matrix

- Most district attributes highly correlated with each other.
- Some generated features present high (and sometimes unusual) correlation between themselves (lower right corner).
- High **negative correlation** between district data (upper left corner).
- No outstanding correlations between status and other attributes.



Note: Full resolution image in zip file

Data Preparation: Interesting Correlation



The average transactional amount with missing K symbols increases with the average monthly income, as well as with the average transactional amount of withdrawals.

Experimental Setup: Optional Operations

- Date Sliding Window
  - A particular experiment conducted was to use several time windows to increase the algorithm predictive performance, however, its results did not improve the AUC comparatively to the regular train test split, probably due to the smaller amount of training data available this way.
- Hyper Parameter Optimization
  - Random search for broad parameter ranges and high parameter count.
  - Grid search for narrowed down parameter ranges.
  - Manual experimentation guided by algorithm performance and type.

Results: Random Forest

	Training					Test				
	Accuracy	Precision	Recall	F1	**AUC**	Accuracy	Precision	Recall	F1	**AUC**
SMOTE + RU + AFS	0.90	0.86	0.95	0.90	0.978	0.91	0.95	0.95	0.95	0.941
SMOTE + RU + AFS + OR	0.95	0.93	0.95	0.94	0.974	0.94	0.96	0.96	0.96	0.919
SMOTE + AFS	0.94	0.91	0.98	0.94	0.983	0.89	0.92	0.96	0.94	0.918
RU + AFS	0.71	0.68	0.84	0.72	0.826	0.88	0.96	0.89	0.93	0.940
SMOTE + RU + MFS	0.87	0.85	0.9	0.87	0.945	0.88	0.93	0.93	0.93	0.916
SMOTE + RU + AFS + SW	0.87	0.83	0.94	0.87	0.952	0.88	0.90	0.91	0.91	0.778

Results: Decision Tree

	Training					Test				
	Accuracy	Precision	Recall	F1	**AUC**	Accuracy	Precision	Recall	F1	**AUC**
SMOTE + RU	0.84	0.84	0.84	0.84	0.860	0.91	0.93	0.96	0.95	0.905
SMOTE + RU + OR	0.91	0.91	0.92	0.91	0.928	0.88	0.94	0.91	0.93	0.779
SMOTE	0.86	0.87	0.89	0.88	0.880	0.92	0.96	0.95	0.96	0.911
RU	0.68	0.68	0.75	0.67	0.747	0.68	0.68	0.75	0.67	0.747

Note how outlier removal (OR, see **glossary** at the end) didn't improve results both in here and in the Random Forest ensemble.

Results: Support Vector Machine

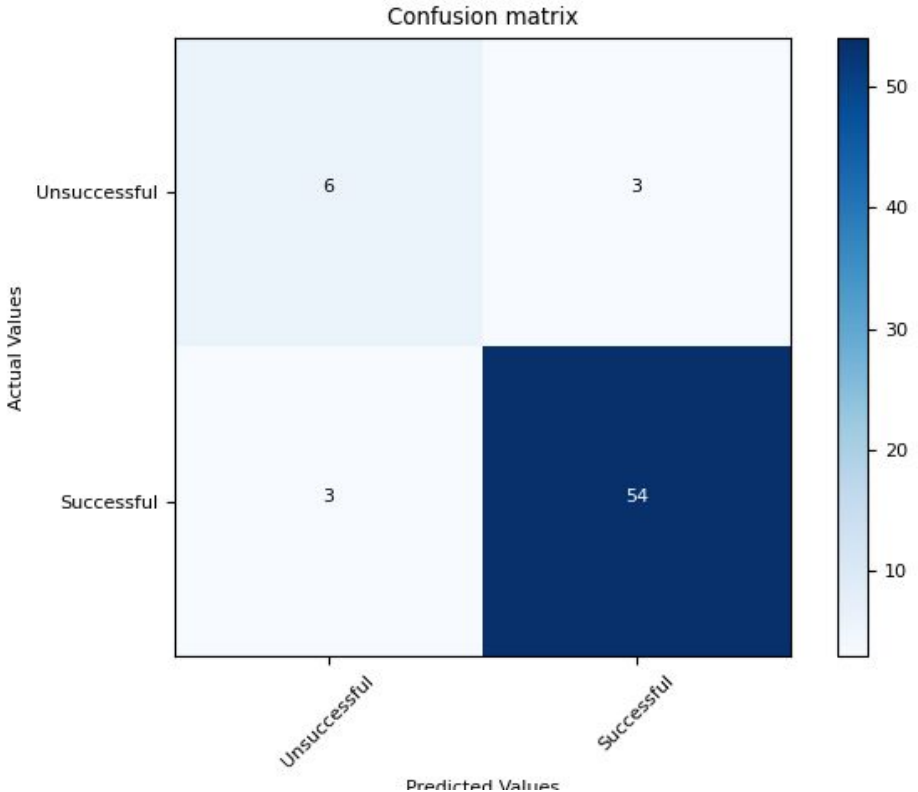
	Training					Test				
	Accuracy	Precision	Recall	F1	**AUC**	Accuracy	Precision	Recall	F1	**AUC**
SMOTE + RU	0.71	0.69	0.74	0.71	0.789	0.68	0.95	0.67	0.78	0.780
SMOTE	0.69	0.69	0.69	0.69	0.811	0.71	0.95	0.70	0.81	0.791
RU	0.86	0.86	1.00	0.92	0.892	0.86	0.86	1.00	0.93	0.844

Results: K-Nearest Neighbors

	Training					Test				
	Accuracy	Precision	Recall	F1	**AUC**	Accuracy	Precision	Recall	F1	**AUC**
SMOTE + RU	0.91	0.97	0.94	0.90	0.993	0.73	0.93	0.74	0.82	0.710
SMOTE	0.92	0.97	0.86	0.91	0.985	0.70	0.93	0.70	0.80	0.727
RU	0.78	0.84	0.70	0.76	0.827	0.68	0.97	0.65	0.78	0.835

Results: Random Forest Confusion Matrix

- In some of the results gathered, the precision, recall and F1 score have the same value.
- Analysis of the confusion matrix of these cases reveals that it is because the number of **false positives** was equal to the number of **false negatives**.



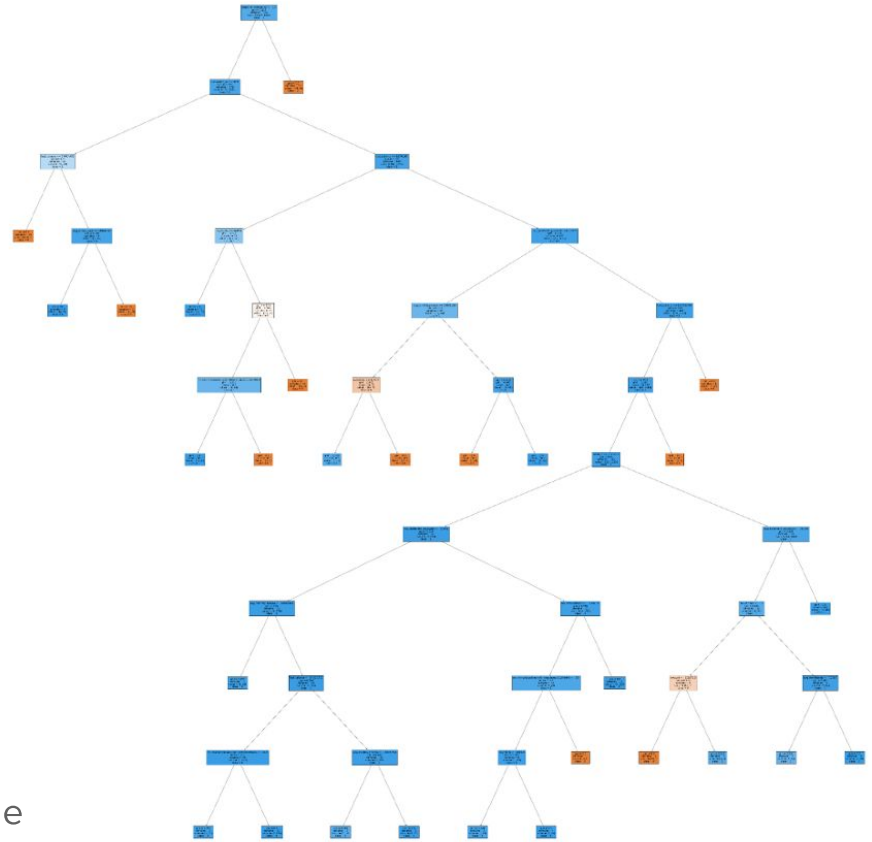
Results: Algorithm’s Most Influential Hyper Parameters

- Decision Tree / Random Forest:
  - **Max Features (RF)** - Maximum number of predictive attributes to be considered when looking for the best split. Overall, **the square root or log2 of n**, with n being the total number of features, yielded the best results, since when considering all the attributes a certain degree of overfitting would take place, mainly due to the poor correlation between some of them and a loan’s status.
  - **No. estimators (RF)** - Number of trees in the forest. The predictive performance tended to increase with this parameter up to a certain point, where it stagnated (around 500).
  - **Class weight** - The weights associated with the classes. This parameter in combination with the class balancing, namely through the use of upsampling / SMOTE, helped the algorithm to favor the “Successful”’ class (weight 7) since it would be the most common in a real scenario, without losing essential information regarding the minority class (weight 1).

Results: Algorithm's Most Influential Hyper Parameters

- K-NN
  - **Leaf Size** - The better results were given by the default value of 30.
  - **Power Parameter (p)** - Power parameter for the Minkowski metric. Best power parameter was 1, which is the equivalent of using Manhattan distance.
  - **Number of Neighbors** - 5 was the best number of neighbors to use.
- SVM
  - **Regularization parameter (C)** - The strength of the regularization is inversely proportional to C, with the value 0.1 being the best option.
  - **Gamma** - Kernel coefficient for 'poly', the best being the value 1,
  - **Kernel** - Specifies the kernel type used in the algorithm. "Poly" was the most suited type according both to the grid and to the random search.
- Other
  - **Sampling strategy** in balancing algorithms (Random Undersampling and SMOTE) - The ratio of the number of samples in the minority class over the number of samples in the majority class after resampling. As expected, the more classes were undersampled, the more information was lost, leading to overfitting at certain points. Results were better when using SMOTE followed by undersampling, however, if the upsampling rate was too high, performance started to decrease.

Results: Decision Tree From Random Forest



Note: Full resolution image in zip file

Results: Decision Tree From Random Forest

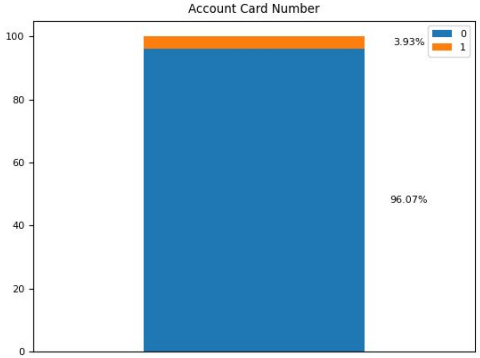
From this example, we can easily interpret the algorithm's decision making rules and draw some interesting conclusions:

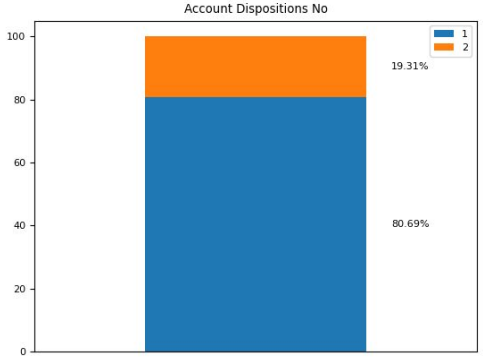
- Clients who had at least one time a negative balance in their accounts are untrustworthy.
- Clients with less than 18 transactions and a current balance inferior to 37697.46 m.u are not to be trusted.
- Clients older than 39 years old with more than 17 transactions and a balance - loan payment value ratio inferior to 5.75 (last balance inferior to 18741.45 m.u and payments greater than 3258.5 m.u) are not reliable payers.
- However, if the client is 39 years old or younger, with the other previous conditions still present, and the number of entrepreneurs in its residential district is inferior to 156, it can be expected of him to pay the loan.

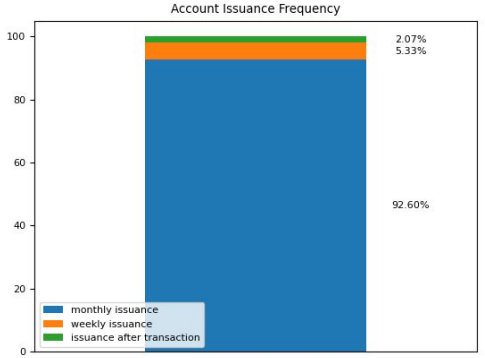
Technology & Used Tools

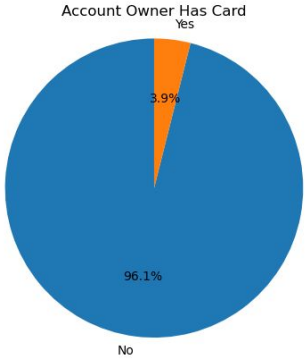
- Project Collaboration
  - Git
  - Google Docs
- Project Development
  - Python
- Data Manipulation Libraries
  - Pandas
  - NumPy
  - Imbalanced-Learn
- Algorithm and Evaluation Libraries
  - Scikit-Learn
- Data Visualization Libraries
  - Matplotlib
  - Seaborn
- Data Storage
  - CSV files

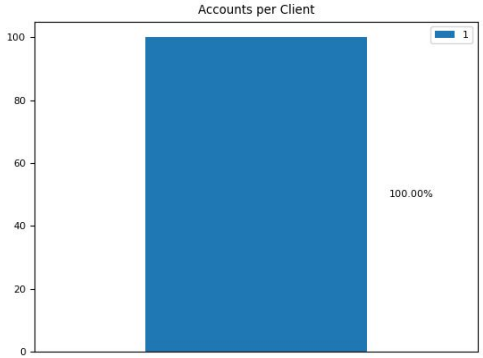
Other plots: Accounts

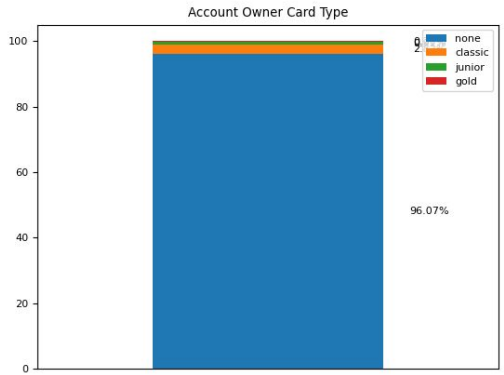




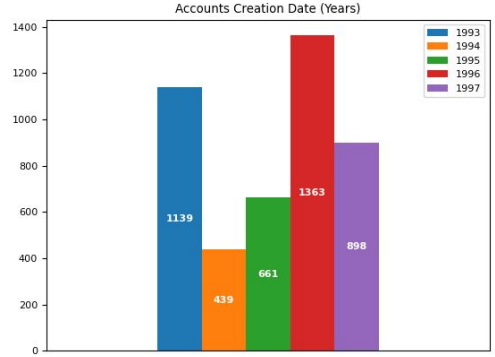


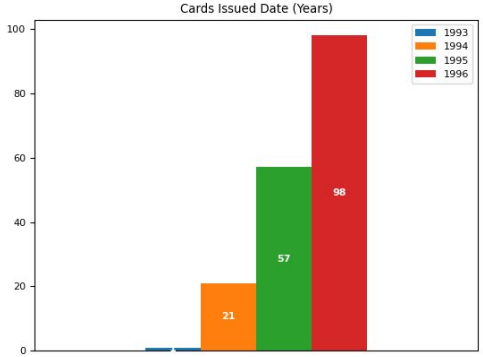


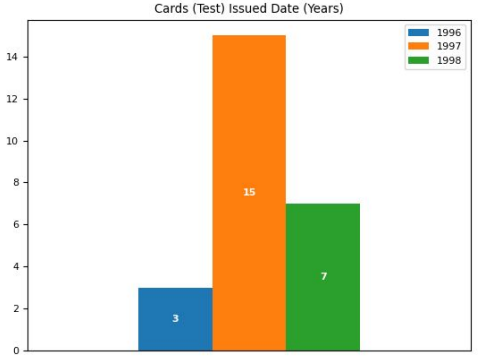


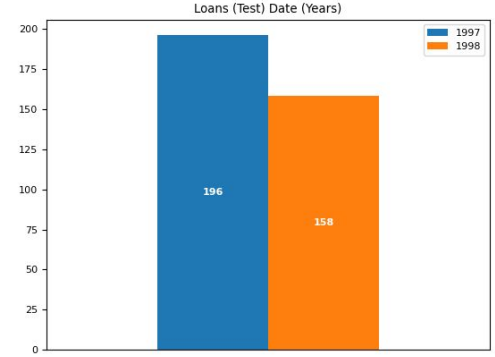


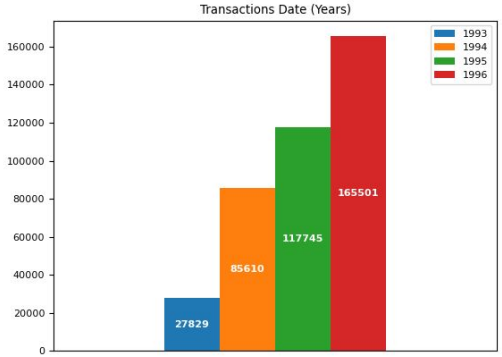
Other plots: Date Distributions

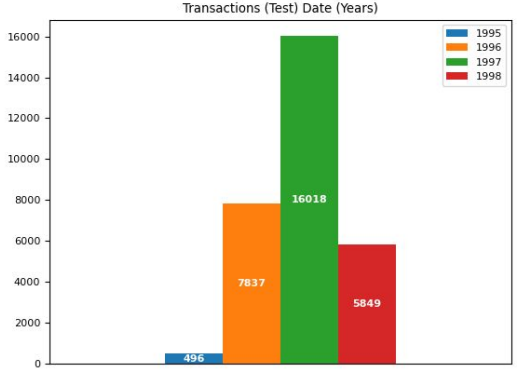






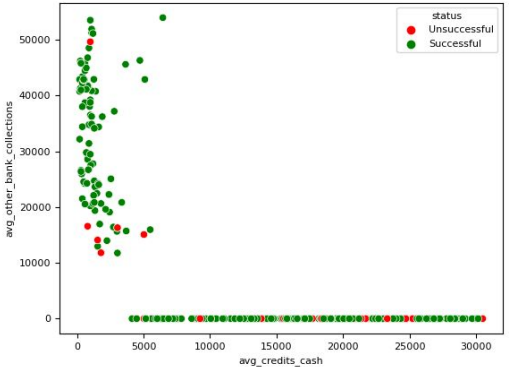


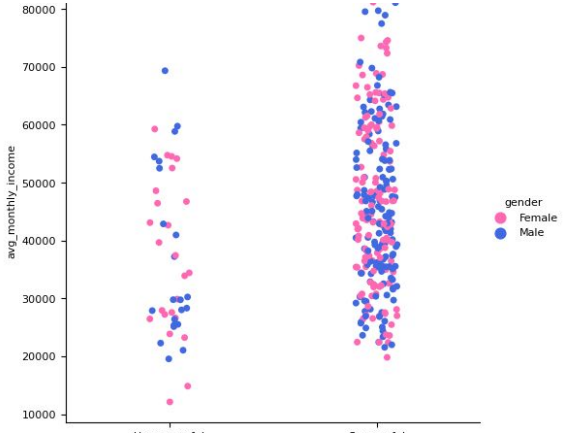


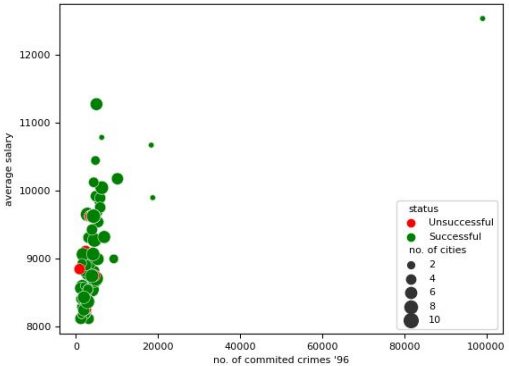


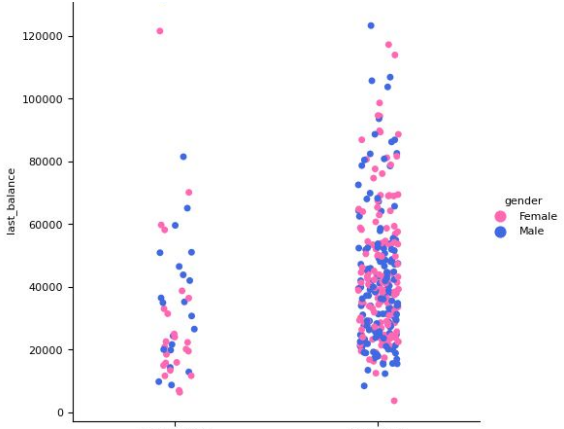
Other plots: Correlations (or lack thereof)

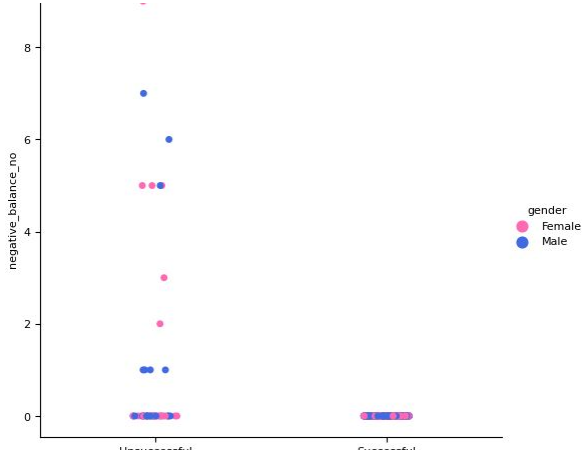
Note: On the categorical scatter plots (middle and right columns), left side represents the unsuccessful loans, and the right successful ones.



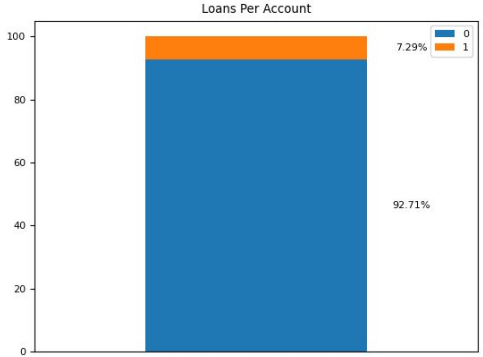


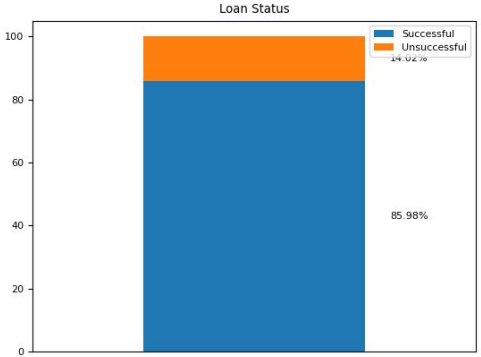


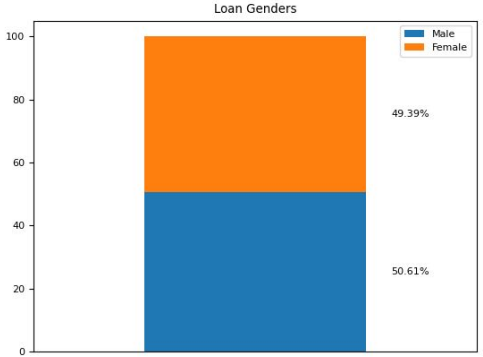


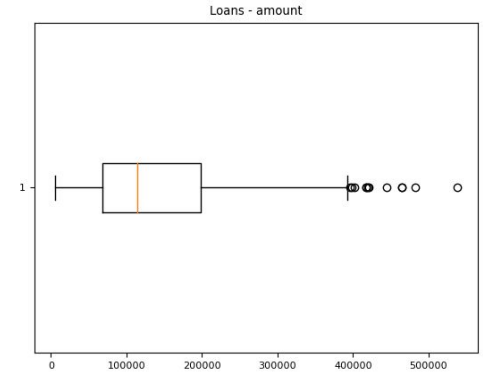


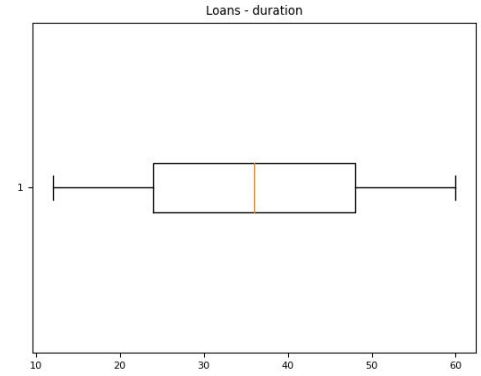
Other plots: Loans

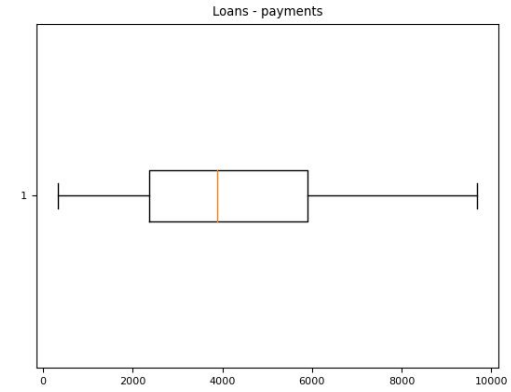




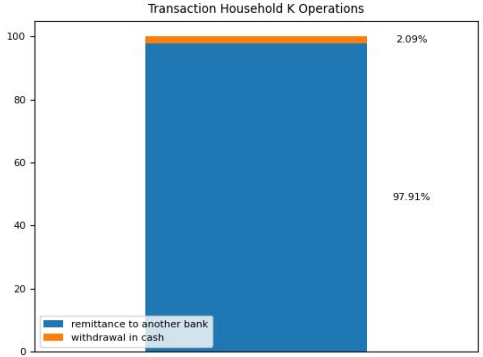


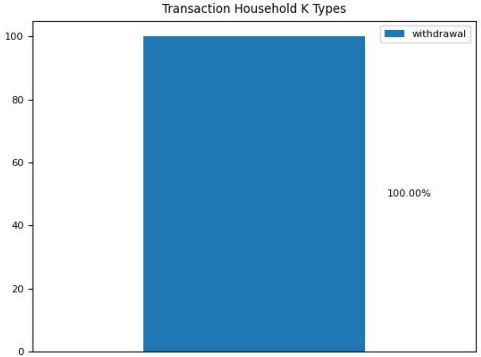


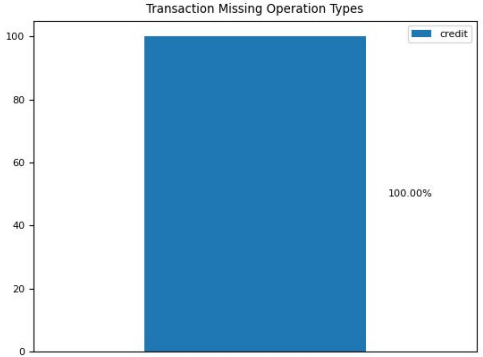


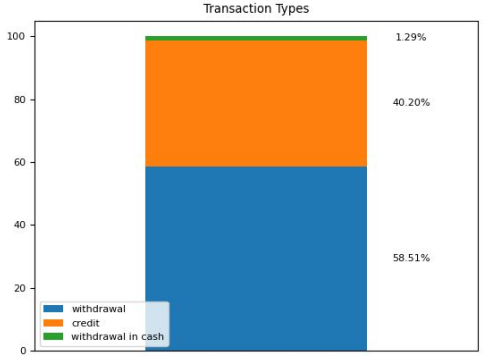


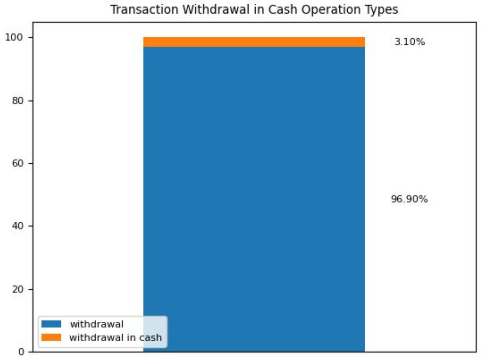
Other plots: Transactions

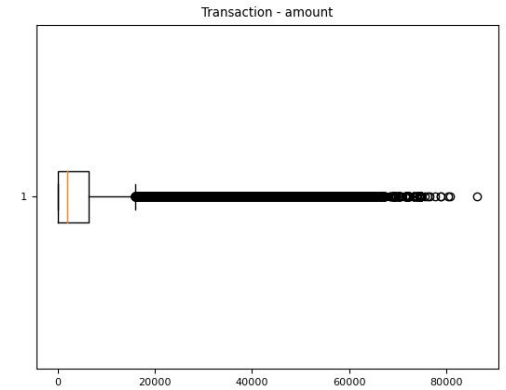




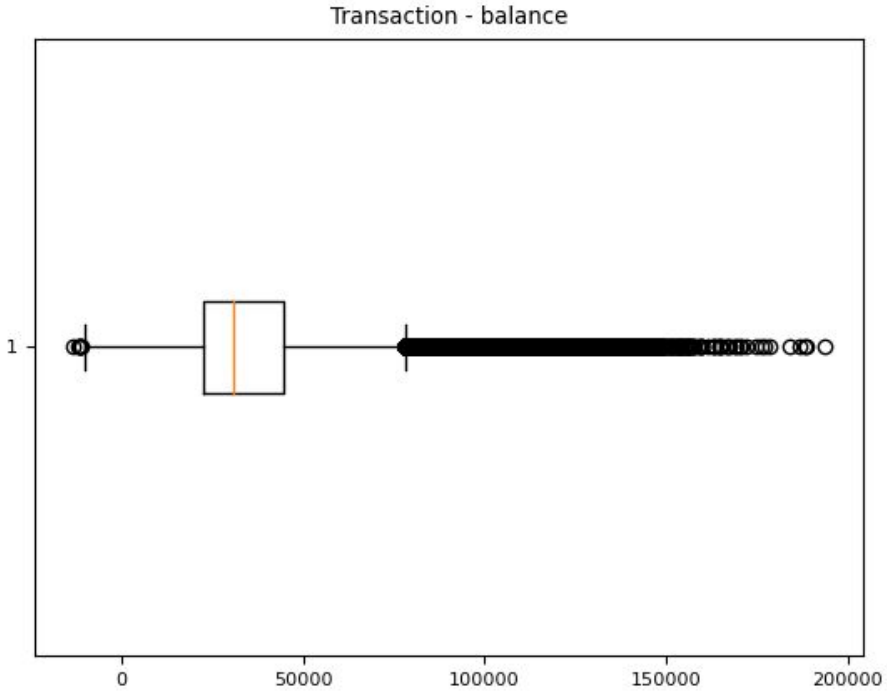






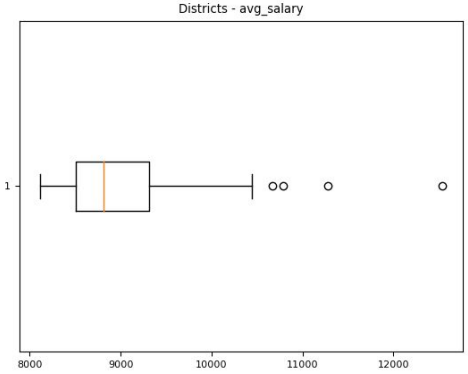


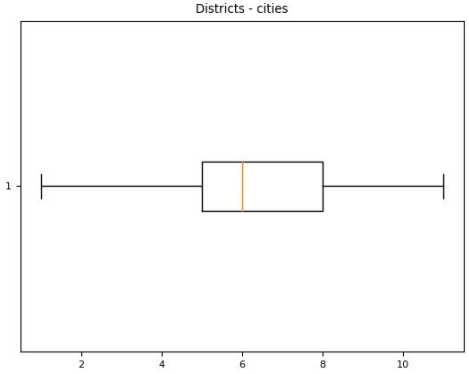
Other plots: Transaction Balance

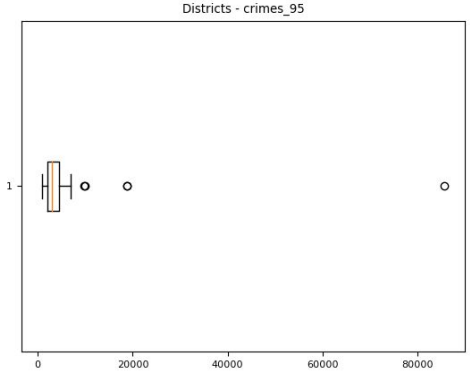


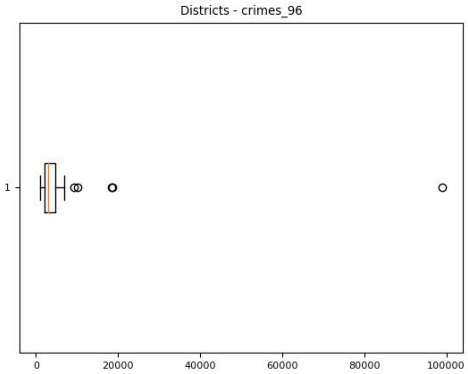
Notice the large number of outliers both here and in the transaction amount distribution (previous slide, bottom right image)

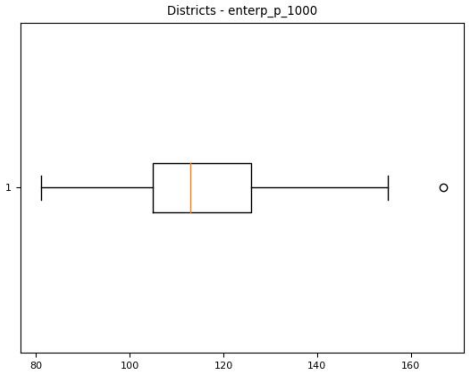
Other plots: Districts

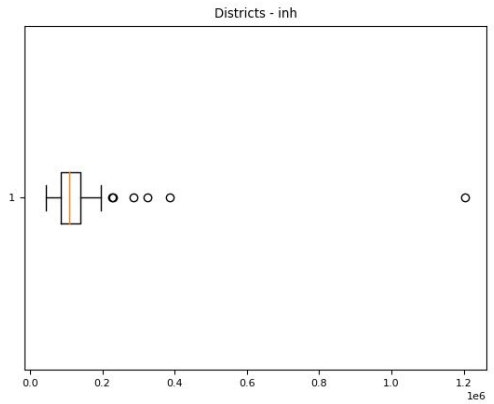




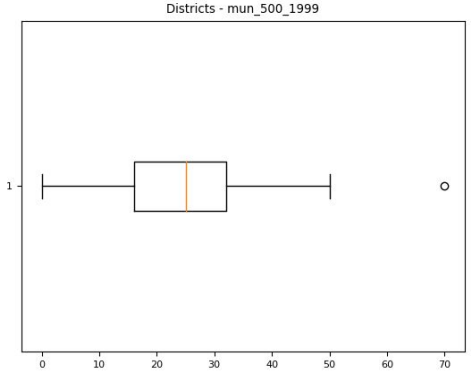


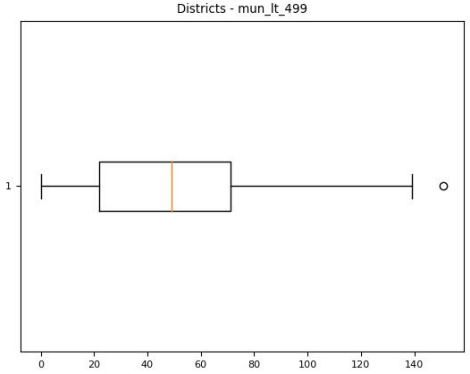


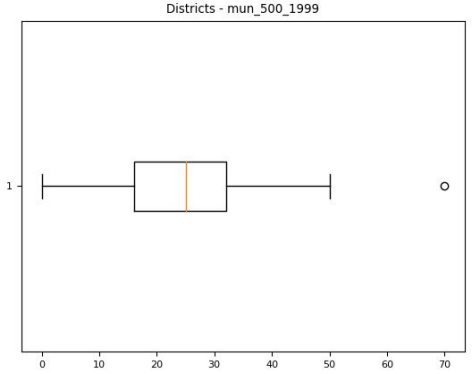


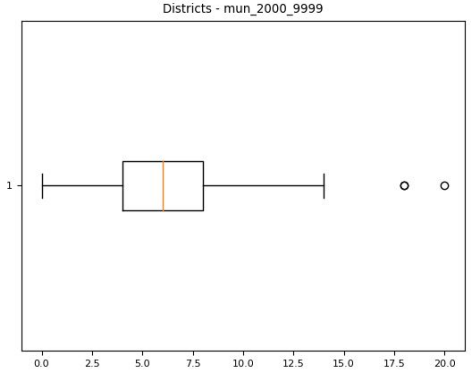


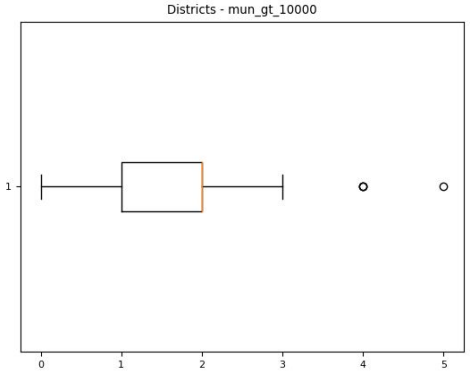
Other plots: Districts

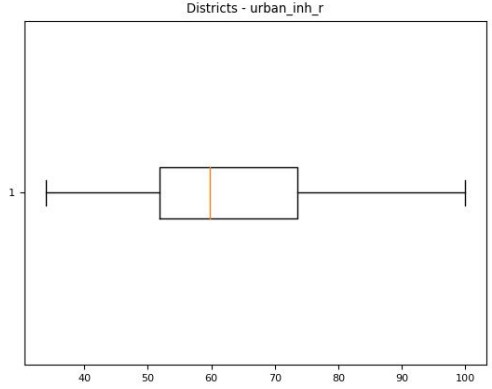




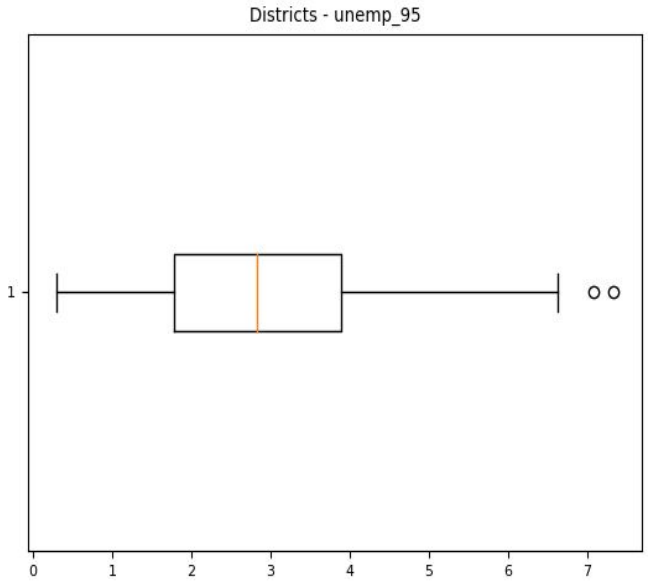


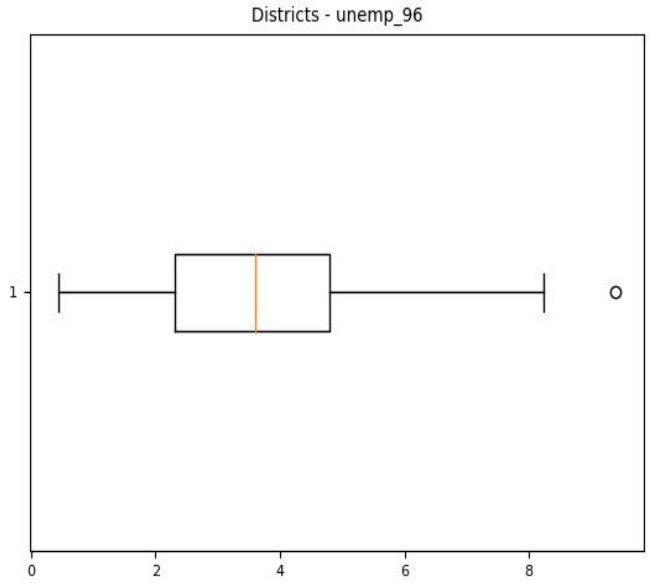




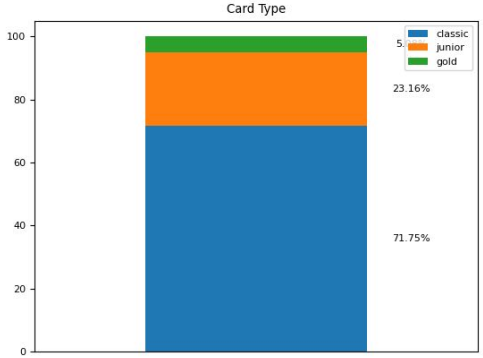


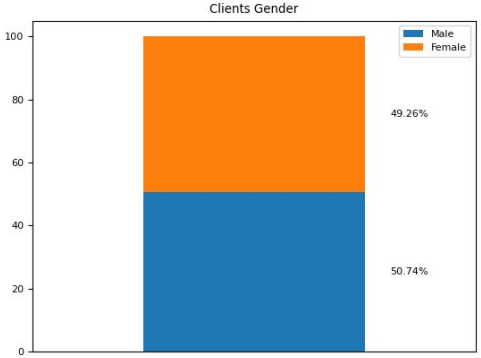
Other plots: District Unemployment Rates

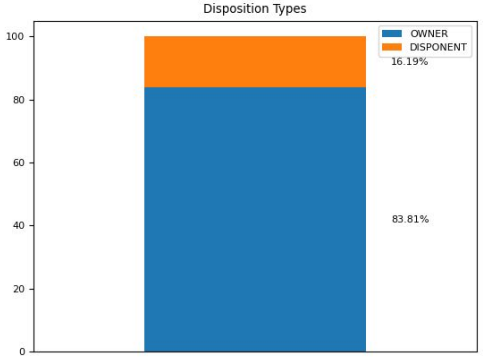


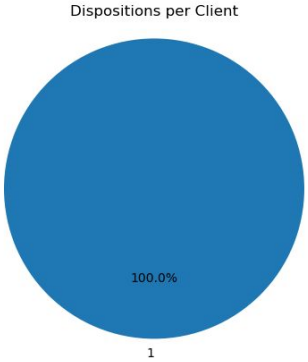


Other plots: Miscellaneous Distributions









Glossary

- AFS - Automatic Feature Selection
- FN - False Negative
- FP - False Positive
- MFS - Manual Feature Selection
- OR - Outlier Removal
- RU - Random Undersampling
- SW - Sliding Window