To Loan Or Not To Loan

T7G3

Afonso Caiado up201806789@up.pt
Elias Lambrecht up202102122@up.pt
José Miguel Maçães up201806622@up.pt
Luís Miguel Afonso Pinto up201806206@up.pt

Domain description

The human being has always learned by observing patterns, formulating hypothesis and testing them to discover new rules. Data mining does exactly that. It is the art and science of transforming raw data into useful information. Software is used to search for patterns and associations to determine connections between different variables, and even to create new ones.

Data mining is used to better understand the clients, their patterns and motivations. By exploring large amounts of data, predictions for the future can be made.

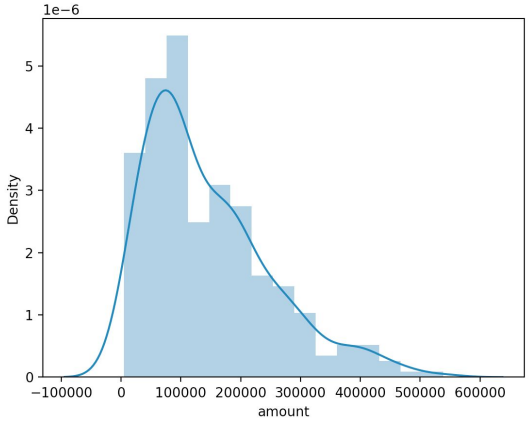
In our particular case, it is not a totally innovative idea, as we found 2 similar machine learning projects, concerning bank loans:

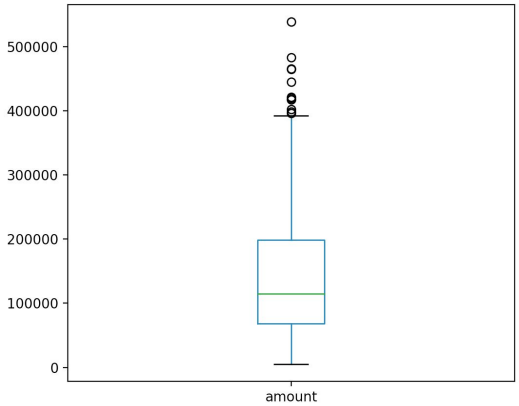
Machine learning: Predicting Bank Loan defaults and **Factors that affect loan giving decision of banks** (both links can be found in the annexes)

Exploratory data analysis

We did an extensive data analysis on the loan dataset:

- Histograms to analyse each columns density
- Boxplots to analyse and detect eventual outliers in each column
- Calculation of the percentage of valid vs invalid loans
- Average loan amount and average monthly loan amount by status analysis






```
percentage of valid(+1) and unvalid(-1) loans and plot:
  1    0.859756
-1    0.140244
```

Problem definition

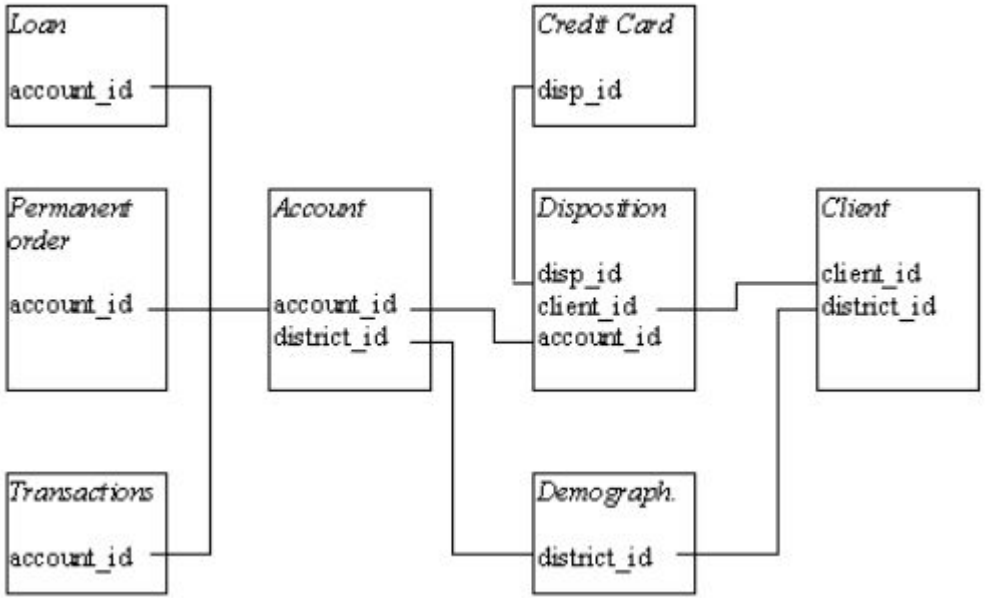
Data about clients, accounts and loans is provided.

We need to predict whether a loan will end successfully, in order for the bank to be able to improve their understanding of customers and seek specific actions to improve services and prevent money losses with default loans.

To do so, we need to analyze the information we have, prepare the data accordingly, build our model and make our predictions.

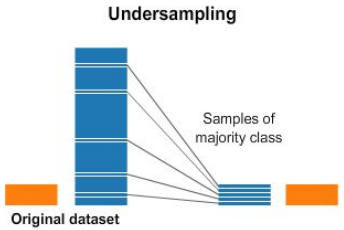
Data preparation

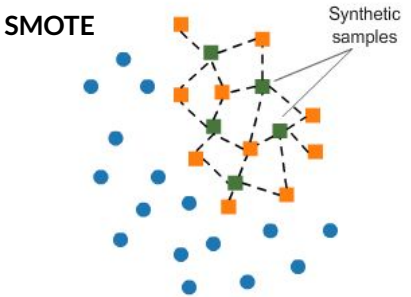
1. Join loan, account and district
   - Categorical to numeric
   - Missing values -> mean
2. Join transactions
   - 1 to N -> 1 to 1 relationship by aggregate of balance
3. Join card
   - Left join
   - Categorical to numeric
4. Join client (date and gender)
   - Map account_id with client_id
   - Decipher birth_number



Data preparation

- Remove unuseful columns
- Feature engineering
  - effort (=amount / average balance)
  - salary_effort (=amount/ average salary)
  - monthly effort (=payments / average salary)
- Imbalanced data
  - undersampling
  - SMOTE





Experimental setup

- Cross Validation: data splitting with a 70% / 30% ratio
- Data split taking the dates into account (train with pre-96 loans to predict post-96 loans)
- AUC used for performance measure
- 4 different algorithms tested: Decision Tree, Random Forest, Gaussian Naive Bayes and Logistic Regression
- Python was used for the whole process

Results

- Gaussian Naive Bayes results:
  - 0.53 AUC -> considers every feature to be independent, has a simple approach to the problem
- Decision Tree Classifier results:
  - 0.52 AUC -> similar approach to Gaussian NB (tree was illogical for some parameters; gold card owners were given worse probabilities than junior card owners)
- Both algorithms are prone to overfitting, hence the bad results.

Results

- Random Forest Classifier:
  - 0.65 AUC -> the best parameters were found to be max_depth=5, n_estimators=41, random_state=5 (random search and grid search)
- Logistic Regression:
  - 0.70 AUC -> with a maximum number of iterations of 1000
- Both algorithms have a smaller chance of overfitting than the other presented. RF aggregate results from multiple trees, which reduces the variance and the probability of overfitting.

Conclusion

We addressed the data mining problem of a bank's loan prediction by preparing and analyzing the available data.

We found it difficult to take what we learned in the data analysis into our data preparation.

Many different tests and experiments have been left for the future due to lack of time (i.e. treatment of the detected outliers). We would have liked to have made a better data analysis and treatment. It could be interesting for example to have calculated a *balance_nearest_loan_date* variable, that would tell us the client's account balance nearest to the loan date.

Individual factors: We believe that the work was well divided between the members of our group, as everyone would have an individual factor of 1.

Annexes:

Domain Analysis

- **Machine learning: Predicting Bank Loan defaults:** *https://towardsdatascience.com/machine-learning-predicting-bank-loan-defaults-d48bffb9aee2*
- **Factors that affect loan giving decision of banks:** *https://deepnote.com/@rhishab-mukherjee/Loan-Prediction-Project-TermPaper-VPSOpiywSu6FZeN2fK8fug*

Data understanding

As mentioned in the presentation part, we did an in depth analysis of the loan dataset. We started by calculating the mean values for the most important columns of the dataset which are the date, amount, duration and payments.

We also calculated the percentage of valid or unvalid loans and realized that the amount of valid loans is much higher than the invalid loans, as the percentage of valid loans is almost 6 times higher. We can conclude that we will have to utilize some method of over / under sampling in the future to balance the data.

```
Average of useful columns:
average_date = 949989.125
average_amount = 145308.6219512195
average_duration = 35.853658536585364
average_payments = 4150.932926829269
```

```
Percentage of valid(+1) and unvalid(-1) loans and plot:
  1    0.859756
-1    0.140244
```

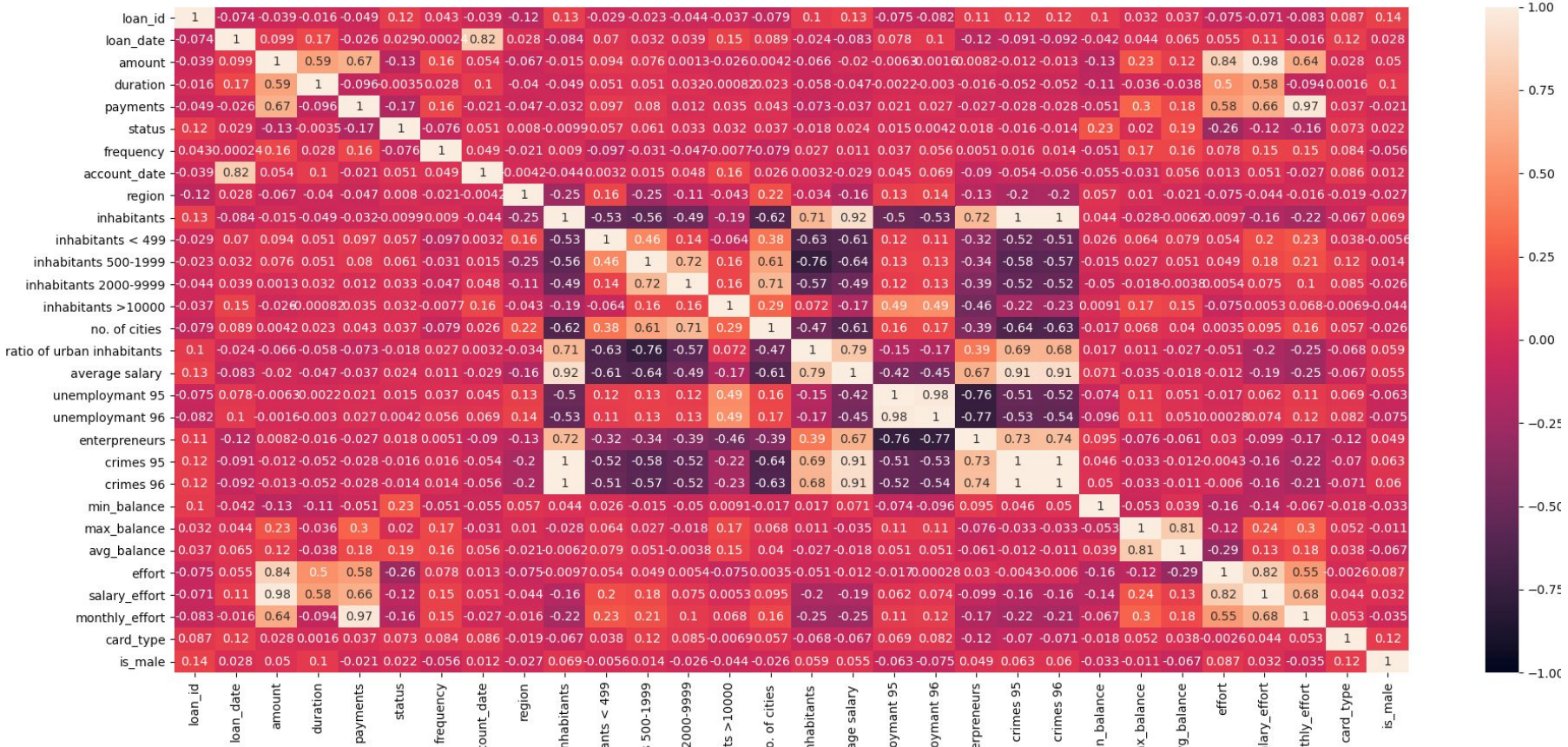
Data understanding: correlations

In the next slide, correlations between all variables can be found in a heatmap, which can be explored in detail in our program.

More specifically, to get an idea of which variables have a bigger impact on the status, we calculated the correlations of all the variables with the status of the loan.

loan_date	0.029130
amount	-0.128237
duration	-0.003537
payments	-0.168436
status	1.000000
frequency	-0.076336
account_date	0.050820
region	0.008025
inhabitants	-0.009869
inhabitants < 499	0.057297
inhabitants 500-1999	0.060763
inhabitants 2000-9999	0.033489
inhabitants >10000	0.031808
no. of cities	0.036856
ratio of urban inhabitants	-0.018391
average salary	0.023895
unemploymant 95	0.014989
unemploymant 96	0.004233
enterpreneurs	0.017960
crimes 95	-0.015744
crimes 96	-0.014080
min_balance	0.226679
max_balance	0.019865
avg_balance	0.193985
effort	-0.262296
salary_effort	-0.120022
monthly_effort	-0.161797
card_type	0.072986
is_male	0.022487

Annexes:


loan_id
loan_date
amount
duration
payments
status
frequency
account_date
region
inhabitants
inhabitants < 499
inhabitants 500-1999
inhabitants 2000-9999
inhabitants >10000
no. of cities
ratio of urban inhabitants
average salary
unemployment 95
unemployment 96
enterpreneurs
crimes 95
crimes 96
min_balance
max_balance
avg_balance
effort
salary_effort
monthly_effort
card_type
is_male
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

Data visualization

Regarding the data visualization part, we did different types of plots to analyse different aspects of the dataset, all of which will be presented in the slides 17-18.

To further emphasize the point that resampling will have to be done in the future, a boxplot was done analyzing the number of loans with each status (1 or -1), the same conclusion was drawn.

We then made some histograms to analyse the density / quantity of the values in the date, amount, duration and payments columns:

- The date plot turned out to be somewhat skewed, as more loans existed in the 1996 year.
- The amount histogram is also generally skewed towards the lower values, which is normal because there will obviously exist less loans of very high amounts.
- The duration plot informs us that although the duration values are just some specific values, the data is mostly well distributed.
- Regarding the payments data, the same conclusion as for the amount can be drawn.

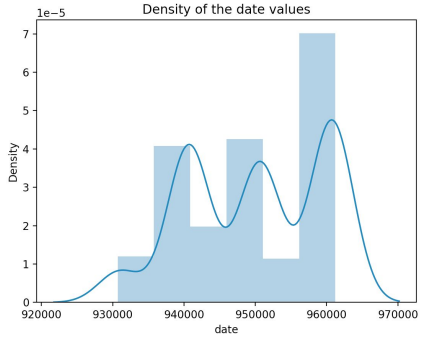
Data visualization

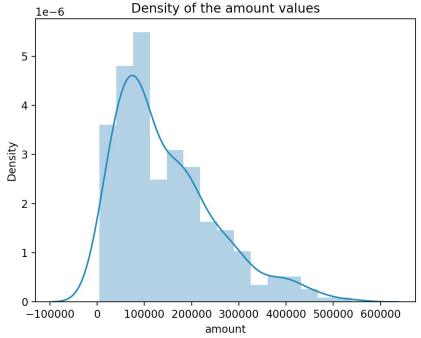
Next, we tried to visualize possible existing outliers in the data by creating boxplots for the same columns we just described, to also provide us a summary of the different variable's distribution. The only data that showed some existing outliers was the amount plot, as many outliers appeared above the Q3 + 1.5*IQR mark (Q3 = third quartile, IQR = interquartile range). Once again, this does make sense as there may be loans of very high amounts, unusual to most of the other values for the amount of the loan.

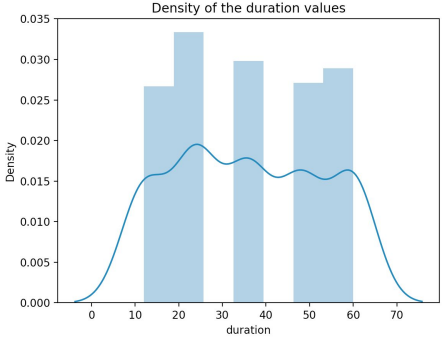
We also did a line plot to analyse the amount according to the date variation. No significant difference was noted from year to year, as the values normally have the same range in each year.

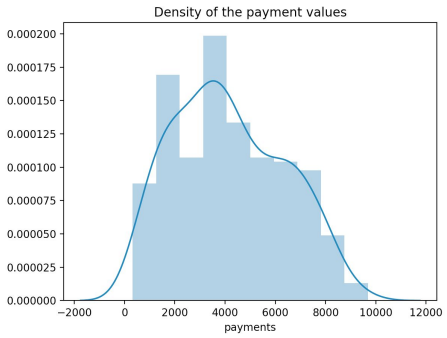
A couple of barplots were also done to analyse the difference in the average loan amount or monthly loan amount according to the status of the loan. We realized that the unsuccessful loans (those that were not paid) usually have a slightly higher average loan amount than the successful loans.

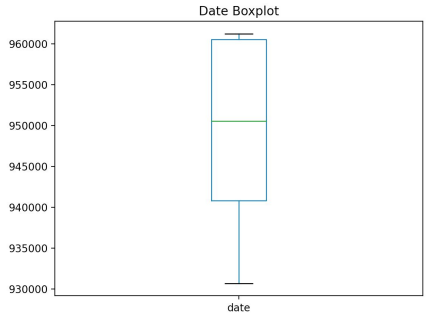
Annexes:

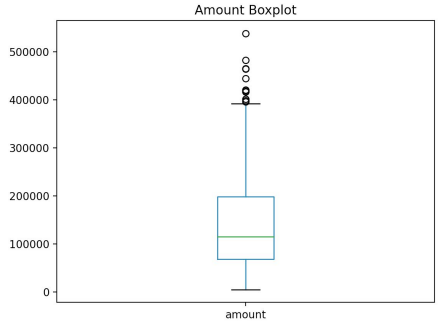

Density of the date values
1e−5
Density
date

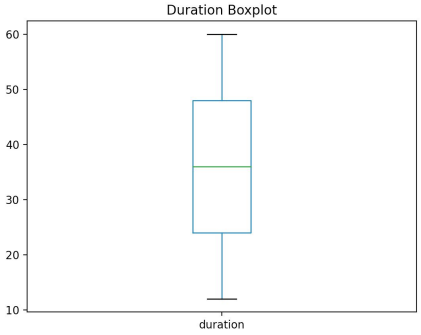

Density of the amount values
1e−6
Density
amount

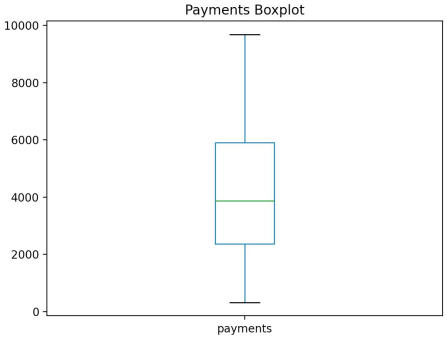

Density of the duration values
Density
duration


Density of the payment values
payments

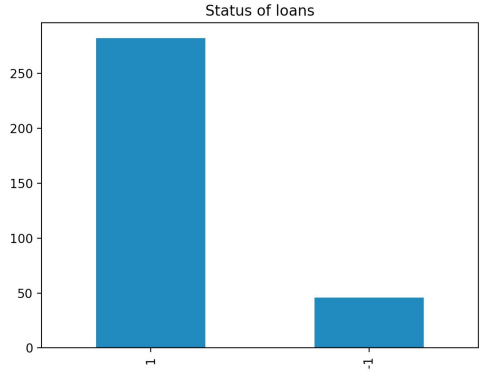

Date Boxplot
date

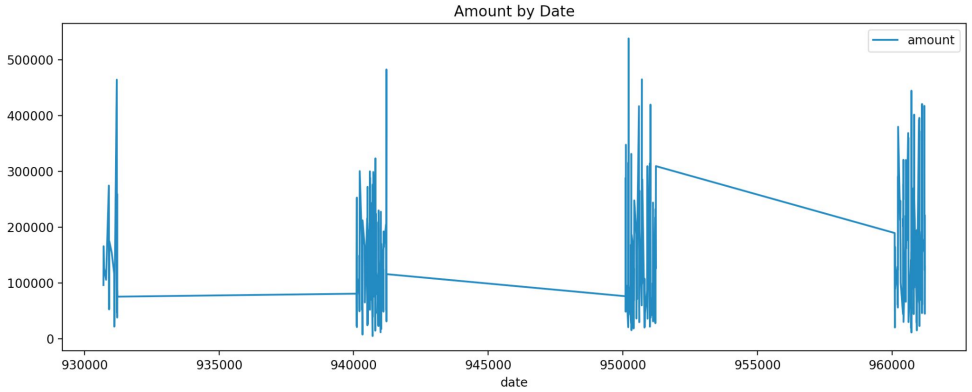

Amount Boxplot
amount

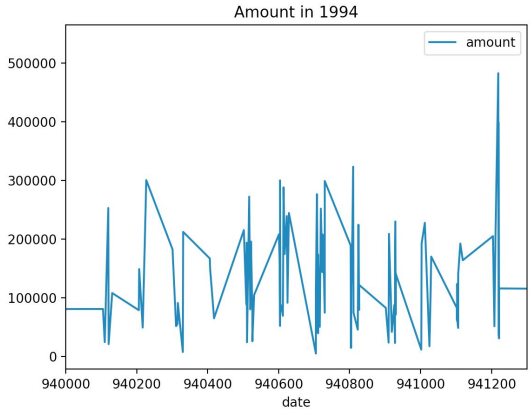

Duration Boxplot
duration

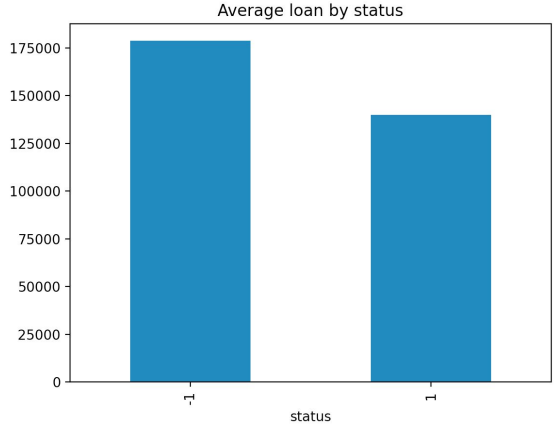

Payments Boxplot
payments

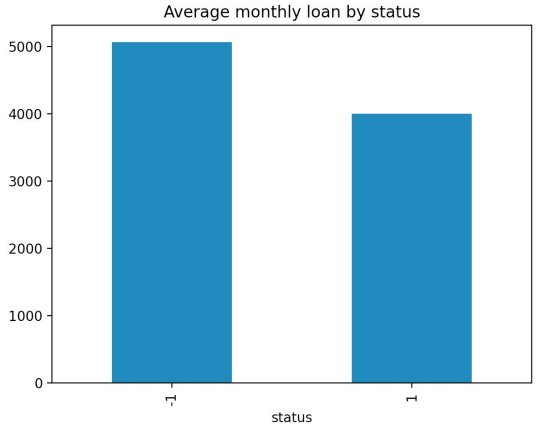
Annexes:


Status of loans
250
200
150
100
50
0
1
-1


Amount by Date
amount
500000
400000
300000
200000
100000
0
930000
935000
940000
945000
950000
955000
960000
date


Amount in 1994
amount
500000
400000
300000
200000
100000
0
940000
940200
940400
940600
940800
941000
941200
date


Average loan by status
175000
150000
125000
100000
75000
50000
25000
0
-1
1
status


Average monthly loan by status
5000
4000
3000
2000
1000
0
-1
1
status

Assessment of dimensions of data quality

- **Accuracy**: The given data is hard check it accuracy but an overview shows that the values and attributes are plausible
- **Completeness:** Some values were missing but they were not considered mandatory (some accounts don't necessarily have credit cards). Some attributes on the district table had missing values, so the data was not complete
- **Consistency**: Given the data, it was hard to make an assessment on the data consistency. However we checked for weird birth dates and none were found.
- **Validity**: Even though the birth dates followed the given rules they could be considered invalid because they didn't follow any standard. The dates needed process because they stored both the birth date and the person gender which is an odd way of storing that kind of information
- **Uniqueness**: No repeated values were found

Data Integration, cleaning and transformation

All tables were joined to obtain one single final table, one loan per row:

1. Loan, Account and District were joined, replacing missing values by the mean of that column and also replacing categorical variables by numerical values, such as Account Frequency and District Region;
2. Joined Transaction, also collapsing all transactions from one account into one single average transaction, based on the average balance;
3. Card table was joined, also replacing categorical variable Type by numerical values;(junior cards were given a value of 1 where gold cards were given a value of 3, their names didn't show their hierarchy)
4. Joined Client table, deciphering the birthdate and creating isMale column.

Data Cleaning: Redundancy

We removed unnecessary columns, based on our analysis:

- District: id, code and name;
- Transaction: id, type, operation, k-symbol, bank account and amount;
- Card: id and issued;
- Client: client_id, disp_type, and district_id

Feature Engineering

We created a few new features based on what we thought would help our model:

- Minimum and Maximum Balance columns;
- Effort column, which divides loan amount by average account balance;
- Salary Effort column, which divides loan amount by average district salary;
- Monthly Effort column, which divides monthly payment value by average district salary.
- isMale column, to tell if a loan is associated with a male or female client

Imbalanced data and sampling

From the data understanding we learned there was an imbalance of the positive and negative class. To deal with this we tried undersampling and SMOTE.

With undersampling we deleted records of the majority class randomly until there was a balance. This lead to an even smaller dataset, therefore we looked into oversampling methods.

We used a python library to implement SMOTE, this method oversamples the minority class using k-NN. We used this technique for the complete dataset.

Training and testing data splitting

We tried and compared multiple methods of data spitting the most effective ones were using stratification and the holdout method.

- Using the **Holdout method** the training and testing was split based on the Date feature. Taking into account we were trying to predict future loans, we tried building a model trained on earlier loans and tested on more recent loans. In our case, the training data was comprised of loans with dates before 1996 and the test data of loans from 1996. This meant we had 70% of the data in the training set and 30% in the test set.
- With the right stratification method we made sure that the ratio of two classes was equal in the testing and training sets.

Algorithms

We started off testing multiple algorithms without parameter tuning to get an idea of the performance of each of the models.

- Decision Tree
- Random Forest
- GaussianNB
- LogisticRegression
- SVC
- Perceptron

Parameter tuning

We found Random Forest and Logistic Regression to have the best performance with basic parameters so we focused the parameter tuning on these two models.

- **Random Forest:** To get a general idea of the best parameters we ran a random search and followed it up with grid search to more narrowly search for the optimal parameters
- **Logistic Regression:** We tried around with different options for the multi_class, max_iter and class_weight and selected the best combination based on the performance

Performance Estimation

To measure performance, AUC was used, as we calculated the probability of predicting an unpaid loan.

We tried different combinations of our pipeline, comparing the local results to evaluate each model and then making a submission to the competition to gauge how good our model was.

We always made sure to run the algorithms on the same data in order to compare the performance correctly.

Model improvement

Our model was constantly improved based on experimenting different permutations and combinations of classifiers and their parameters:

- **Gaussian Naive Bayes**: this algorithm considers every feature to be important, but it is prone to overfitting, so we obtained a **0.53** AUC score with it
- **Decision Tree**: this classifier has a similar approach as Gaussian Naive Bayes, but we found the tree to make what we considered illogical decisions, such as Gold card owners being given a worse probability of obtaining a loan than Junior card owners. It is also prone to overfitting and obtained a **0.52** AUC score;
- **Random Forest Classifier**: this classifier is not prone to overfitting, as it aggregates results from multiple trees, which reduces the variance and the probability of overfitting and gave us a **0.65** AUC. We used random search and grid search to find the best parameters to be max_depth=5, n_estimators=41 and random_state=5;
- **Logistic Regression**: this algorithm is also not prone to overfitting and, with a the maximum number of iterations set at 1000, we obtained a **0.70** AUC score.

Feature importance

The attributes were analysed in a more manual way regarding our knowledge about the theme and some features were considered to be more meaningful than others.

Birth Date: It is easier for one to have a loan proposal accepted after the 25 and before 55 years old. People younger than 25 usually have a worse economic stability. After 55 longer loans may also be more frequently declined because of the higher risk of loss per death.

Loan Duration: loan duration when combined with loan amount gives us the amount per month that the loan taker will be paying. Higher amounts per month represent a higher risk, hence the higher chance of the loan to default.

Project Management Methodology

Although we did start the project development by just doing our own intuitive methodology, we quickly realized that we were using some concepts from the CRISP-DM methodology, so we decided that it was better to properly apply it. Next, we describe what we did (or not) in each phase of the CRISP-DM methodology:

- **Business Understanding:** As said before, event though at this point we hadn't yet realized that we were applying the CRISP methodology, we started the project by assessing and understanding the project's goals and objectives with the project's description in moodle. The first thing we did was create a project in our github repository (detailed in a further slide) and plan what were the main tasks we would have to do in this project.
- **Data Visualization:** As described in the presentation we did an extensive analysis of the loan dataset to not only explore the data that was given to us, but also to verify the data quality (missing values, imbalanced data, etc.). cf. slide 3.
- **Data Preparation:** Cleaning, joining and formatting of the data. For further details cf. slides 5-6.
- **Modelling:** For further details cf. slide 7.
- **Evaluation:** Results evaluation and review. For further details cf. slides 8-9.
- **Deployment:** Finally, regarding the deployment phase, as for every group, we made this presentation / report and presented it to the class.

Project Management Plan

Our initial plan was to do the following tasks:

- Expand data visualization and understanding
- Deal with outliers and imbalanced data
- Replace missing values
- Join data
- Create separate modules for classifiers
- Expand modules with optimized parameters
- Create base model (knn) to compare other models to
- Keep track of how models with different data and parameters score

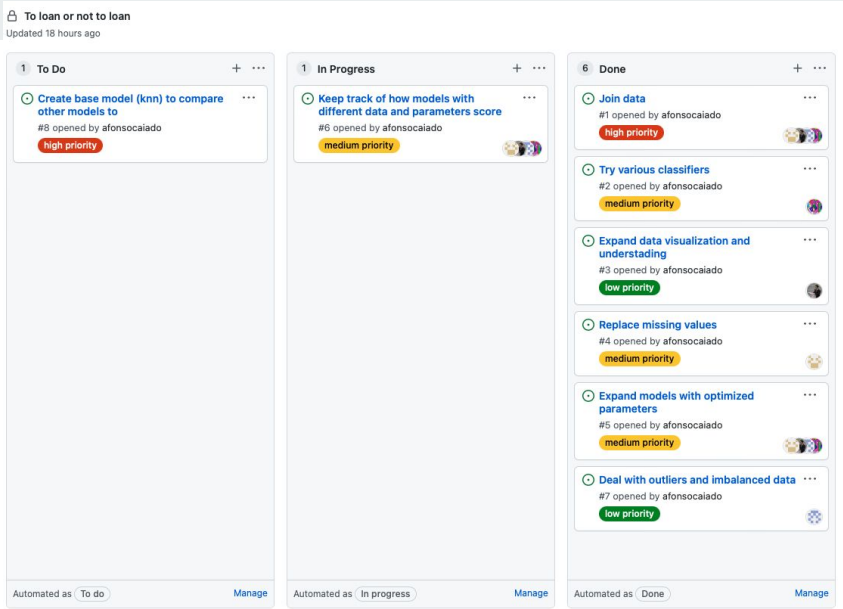
As the project developed, we mostly followed this plan, as we found it to be adequate.

Annexes:

Project Management Tools

We used **github projects** to better plan our project's workflow. It allows every member of the group to know which task (issues) still remain to be done and to assign themselves to one of those tasks, so that the other developers know what is being worked on or not.

We also gave a different level of priority to each issue, so that the most important issues can be done first, in case of lack of time to do all tasks.



Overall, our communication and collaboration as a group was ensured via a **github** repository and a **discord** chat.

Other Tools

Regarding other tools our group used for the realization of the project, we mainly used python and it's different available libraries. Even though at the start we tried using rapidminer, we figured that it would be more beneficial for us to use python, as none of us were particularly comfortable with RM.

For the data visualization part, we used pandas and seaborn.

We also used matplotlib, numpy, sklearn and imblearn throughout the project.