MACHINE LEARNING

to loan or not to loan

T H A T I S T H E Q U E S T I O N

group 52

João Baltazar up201905616
Nuno Costa up201906272
Pedro Correia up201905348

Domain Description

Records of bank accounts and their clients from 1993 to 1998.

4500 accounts
5369 clients
5369 dispositions
77 districts
202 cards *(177 dev)*
426885 transactions *(396685 dev)*
682 loans *(328 dev)*

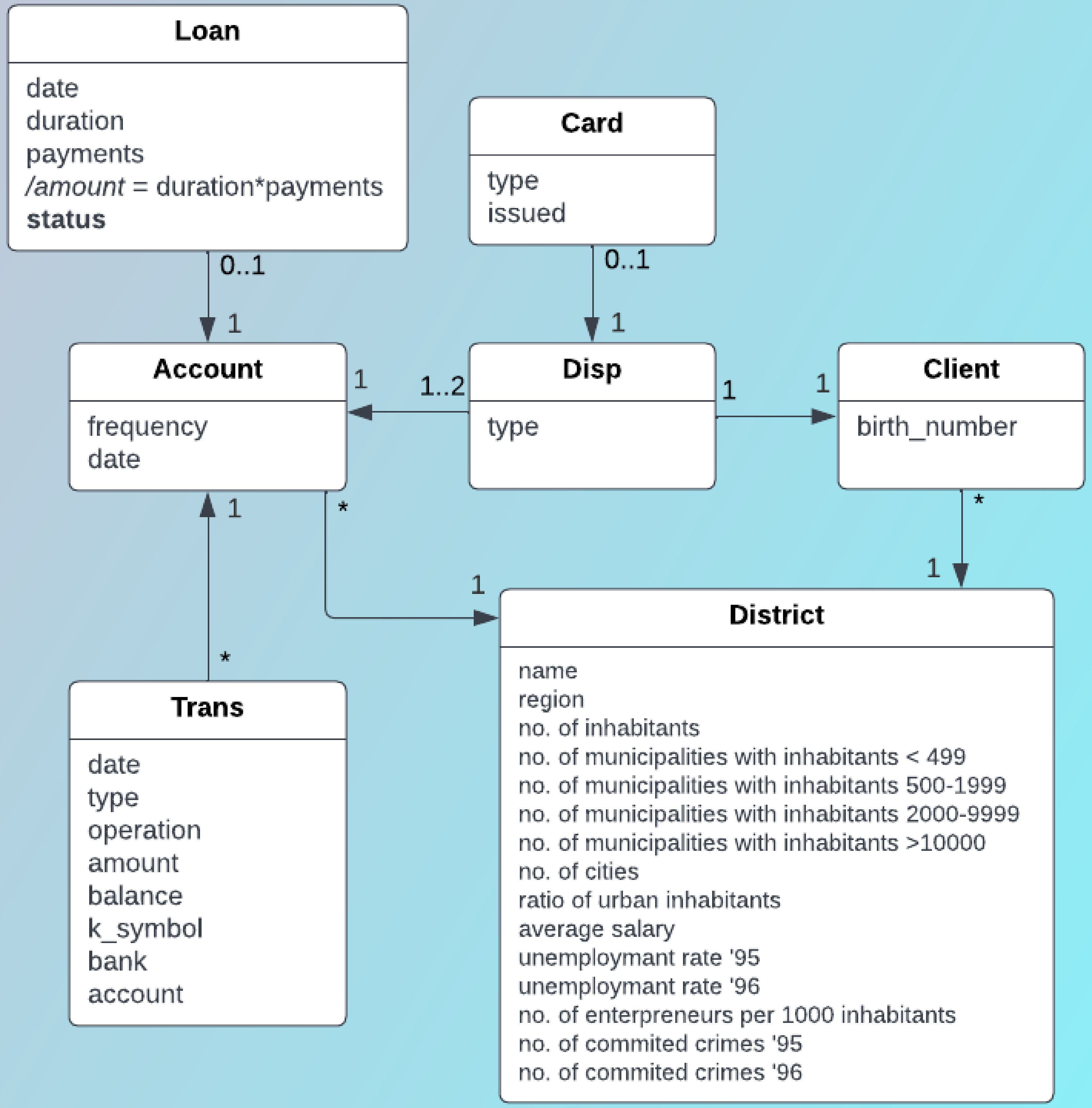
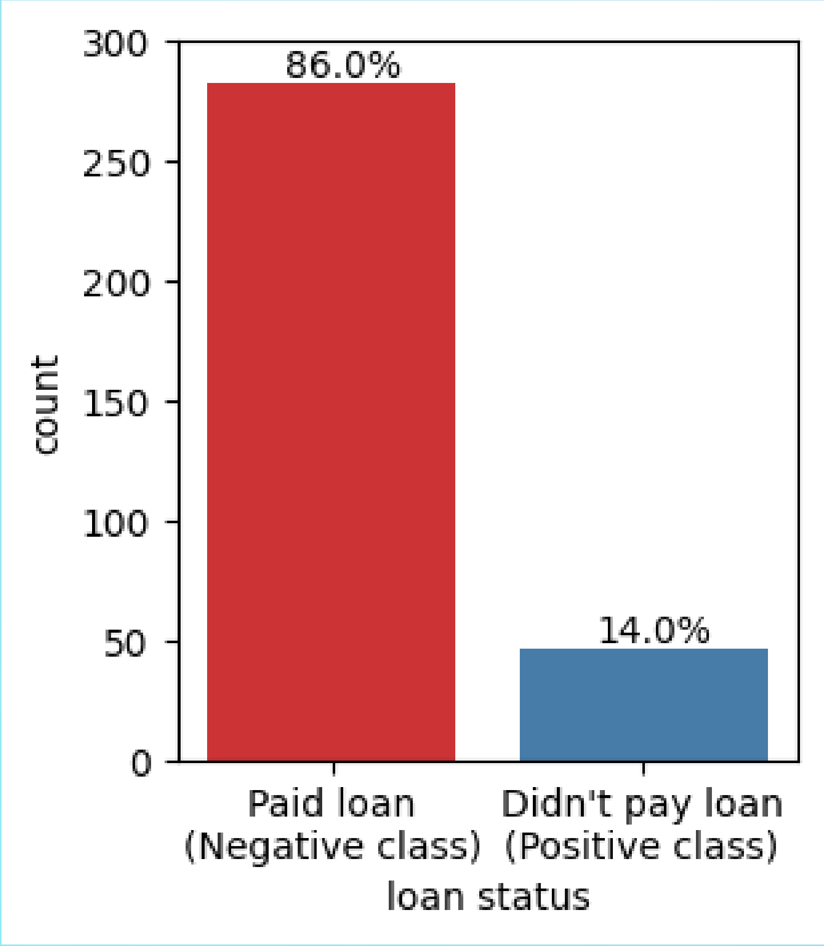


Fig.1 - Relational Model of the 1999 Czech Financial Dataset

group
52

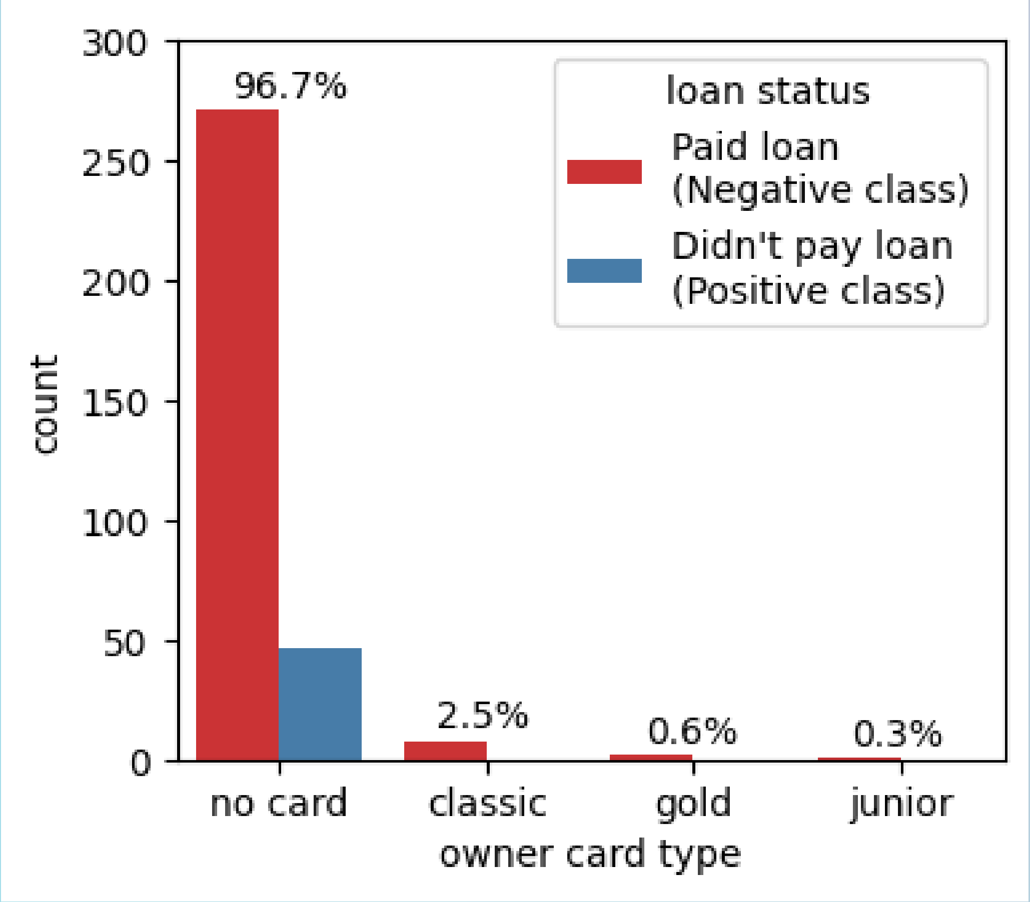
Exploratory Data Analysis

Fig.2 - Imbalanced distribution of loan status



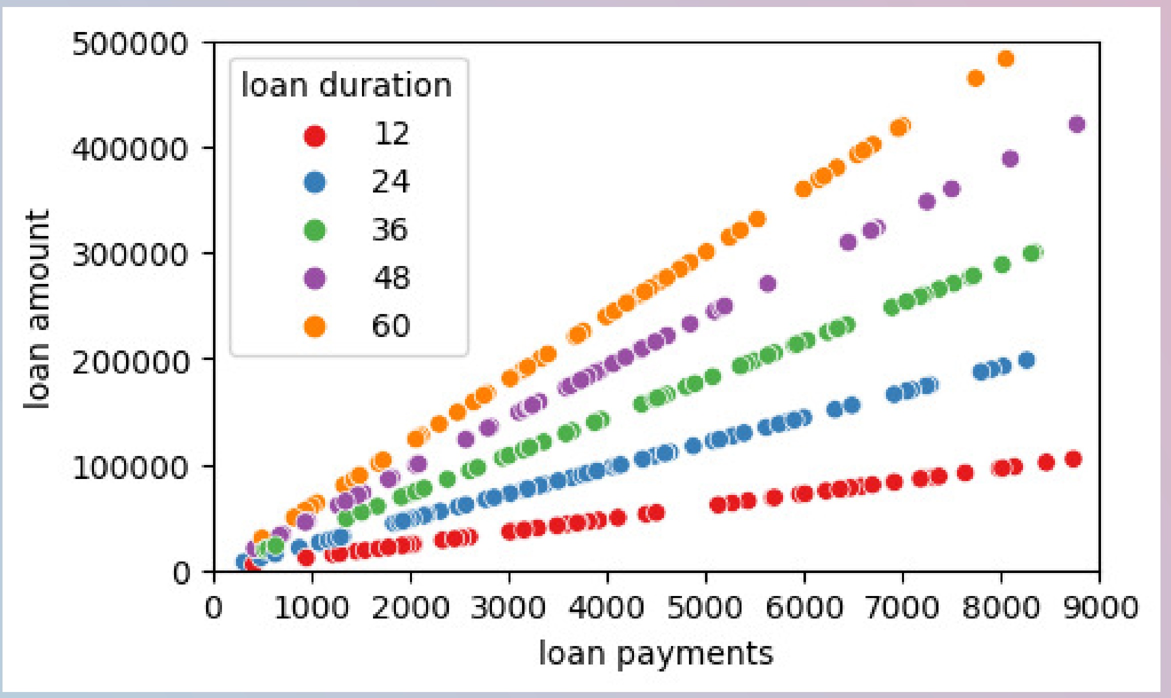
86% of the loans with known outcome were paid off, while *14%* were not.

Fig.3 - Card types of owners of accounts who made a loan



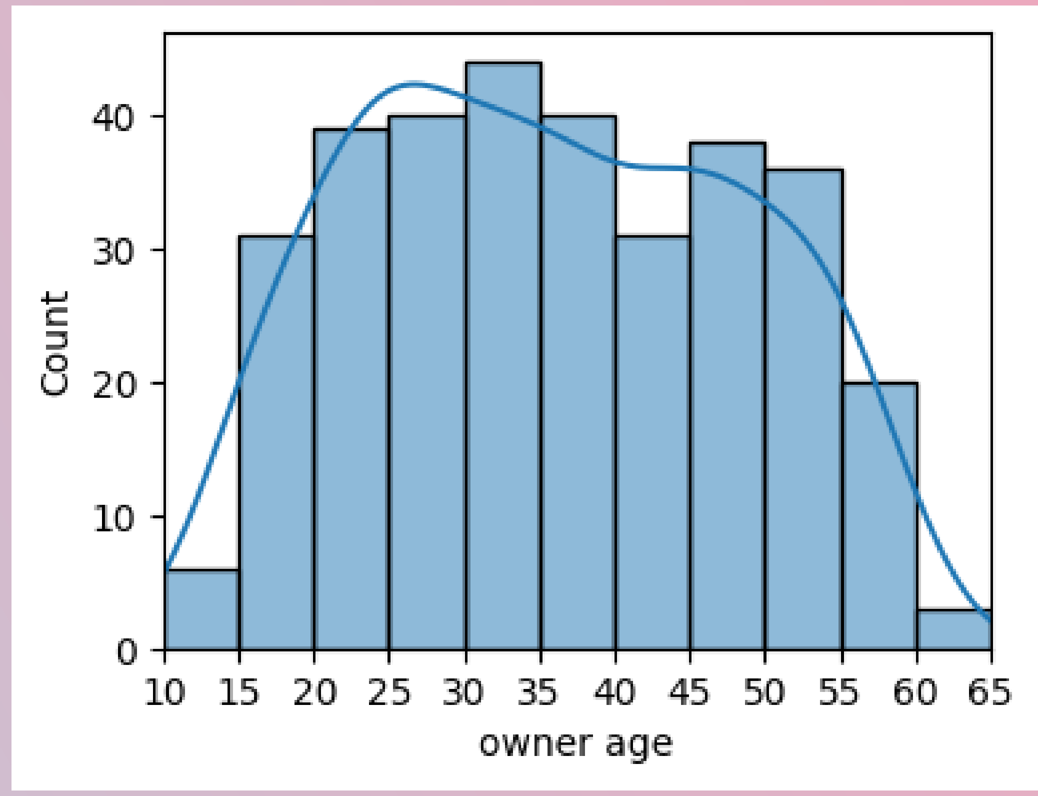
Most clients don't have credit cards, and all who have are owners of an account. The most common card type was *classic*.

Fig.4 - Loan amount is the product of loan payments and loan duration



The bank did not apply interest rate.
(*amount = duration×payments*)

Fig.5 - Age of owners of accounts who made a loan

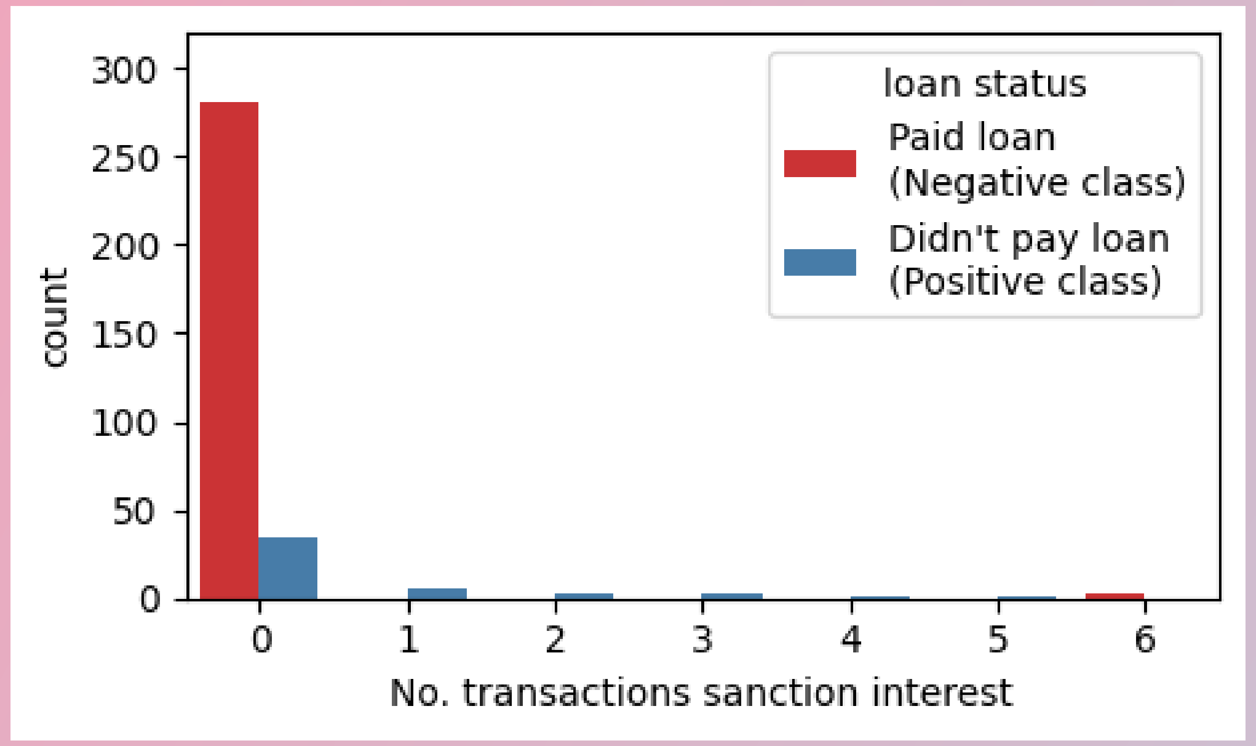


Some loaners are underage and don't have a disponent older than *18* in their account.

group 52

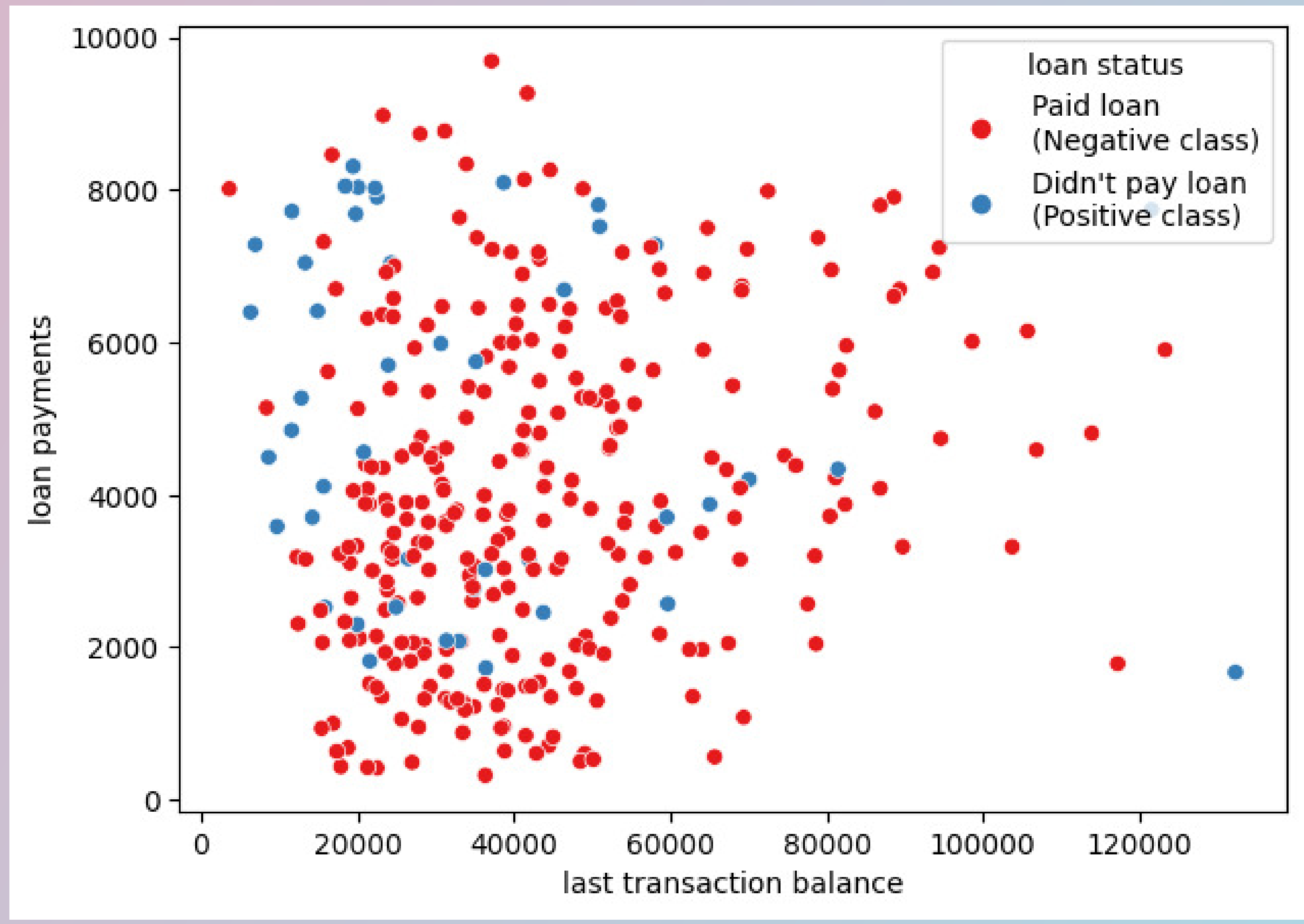
Exploratory Data Analysis

Fig.6 - Number of transactions of sanction interest because of negative balance



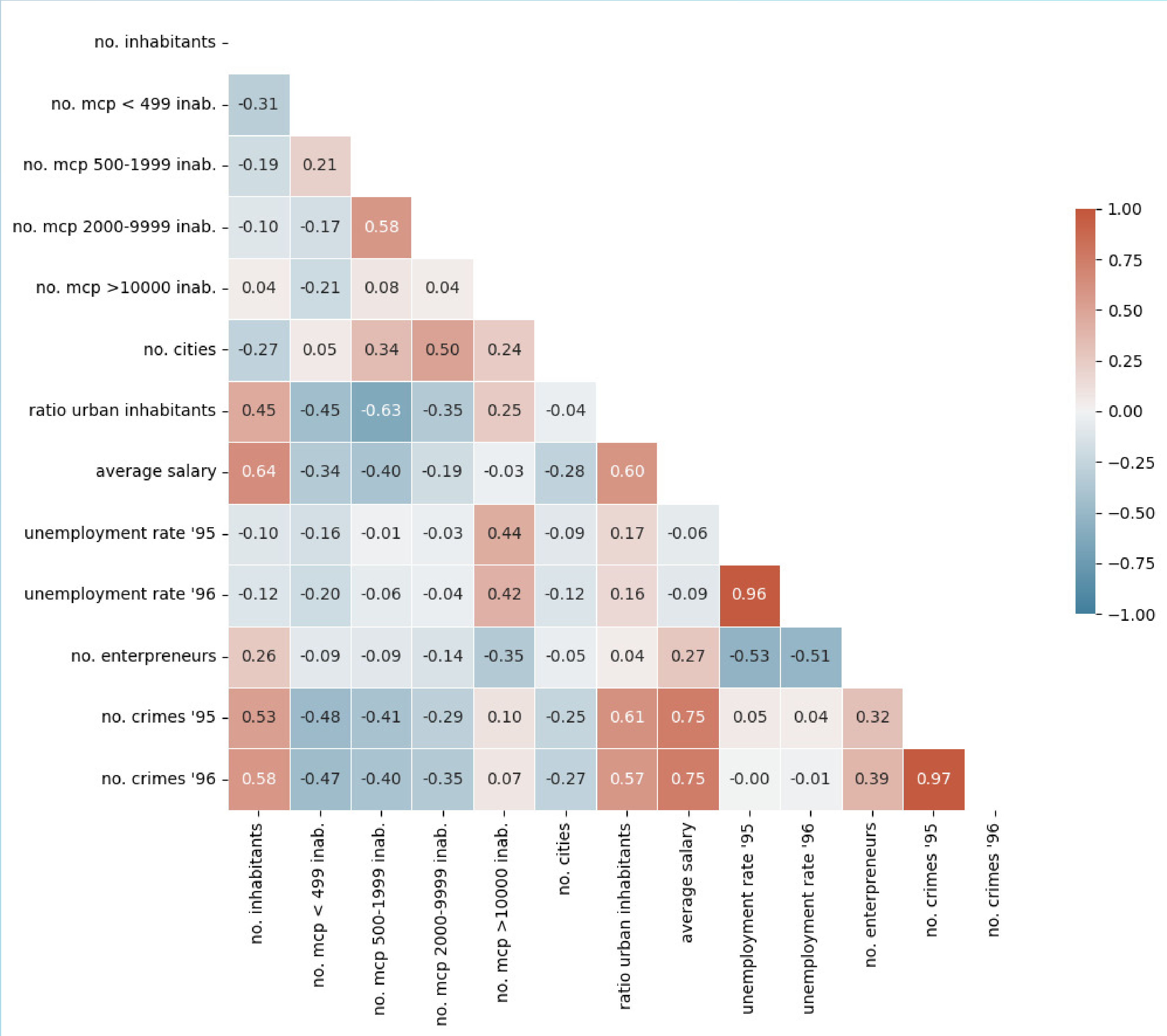
"sanction interest if negative balance" is a good predictor of an unpaid loan.

Fig.7 - Monthly loan payments as a function of balance after last transaction



If the loan is big and the balance of the account after its last transaction is low, it's likely the client will *fail to pay* the rest.

Fig.8 - Correlation matrix of district attributes



Uncovers some nuanced correlations, e.g. a positive correlation between *no. of crimes* and *average salary* — highly populated areas have ***better salaries*** and ***more absolute*** crime.

group 52

Predictive Problem

DEFINITION

Task

Predict whether a **future** loan will fail to be paid
Most loans are paid off → **imbalanced dataset**
Positive class: **not paid**

A bank wants to improve their loaning services.

Better customer understanding
More thorough and efficient credit analysis
Confident prediction of loan fulfillment

Experience

Loan records until *1996*

Performance Measure

AUC from predictions on loans from *1997* and *1998*

Predictive Problem

DATA PREPARATION

Join tables based on IDs

Join *loan*, *account*, *disposition* (*owner*), *card*, *client* and *district*
Take time into account so *loan records* don't include information from the future

Feature Engineering

Extract *birthdate* and *gender* from client *birthnumber*
Use *birthdate* and *loan date* to get *age of the client* (at the *loan*)
Aggregate features from *transactions* (*counts, means, last transaction balance*)

Missing Values

Remove columns with **mostly NaN values** (eg. *card issue date*)
Replace with **mean** — *crime* and *unemployment rate*
Replace with **0** — *mean withdrawal amount when no transactions*
Replace *card type* with **"no card"** when there is no *card*

'count_trans_credits',
'count_trans_withdrawals',
'count_trans_credit_cash',
'count_trans_withdrawal_cash',
'count_trans_withdrawal_card', 'count_trans_collection_other_bank',
'count_trans_remittance_other_bank',
'count_trans_ksymbol_interest_credited',
'count_trans_ksymbol_household',
'count_trans_ksymbol_payment_for_statement',
'count_trans_ksymbol_insurance_payment',
'count_trans_ksymbol_sanction_interest_if_negative_balance',
count_trans_ksymbol_oldage_pension',
'last_trans_balance',
'mean_trans_balance',
'mean_trans_amount_absolute',
'mean_trans_amount_credit',
'mean_trans_amount_withdrawal',
'mean_trans_amount_signed',
'owner_male',
'owner_age',
'account_age_months',
'has_disponent',
'owner_profile'

Fig.9 -Engineered Features List

group 52

Predictive Problem

D A T A P R E P A R A T I O N

Transformation

Standardization — *centering* and *scaling*
Non-linear transformation — for *lognormal/chi-squared* like distributions
Discretization — *categorical feature encoding* and *binarization*

Feature Selection

Remove **redundant** attributes using *correlation threshold* (*>0.8*)
Recursive Feature Selection — *RFECV* and
SequentialFeatureSelector (*backward*)

Outlier Detection

Standard Deviation — *upper bound = 4 × std*
Z-Score — *upper_bound = z-score + 4*

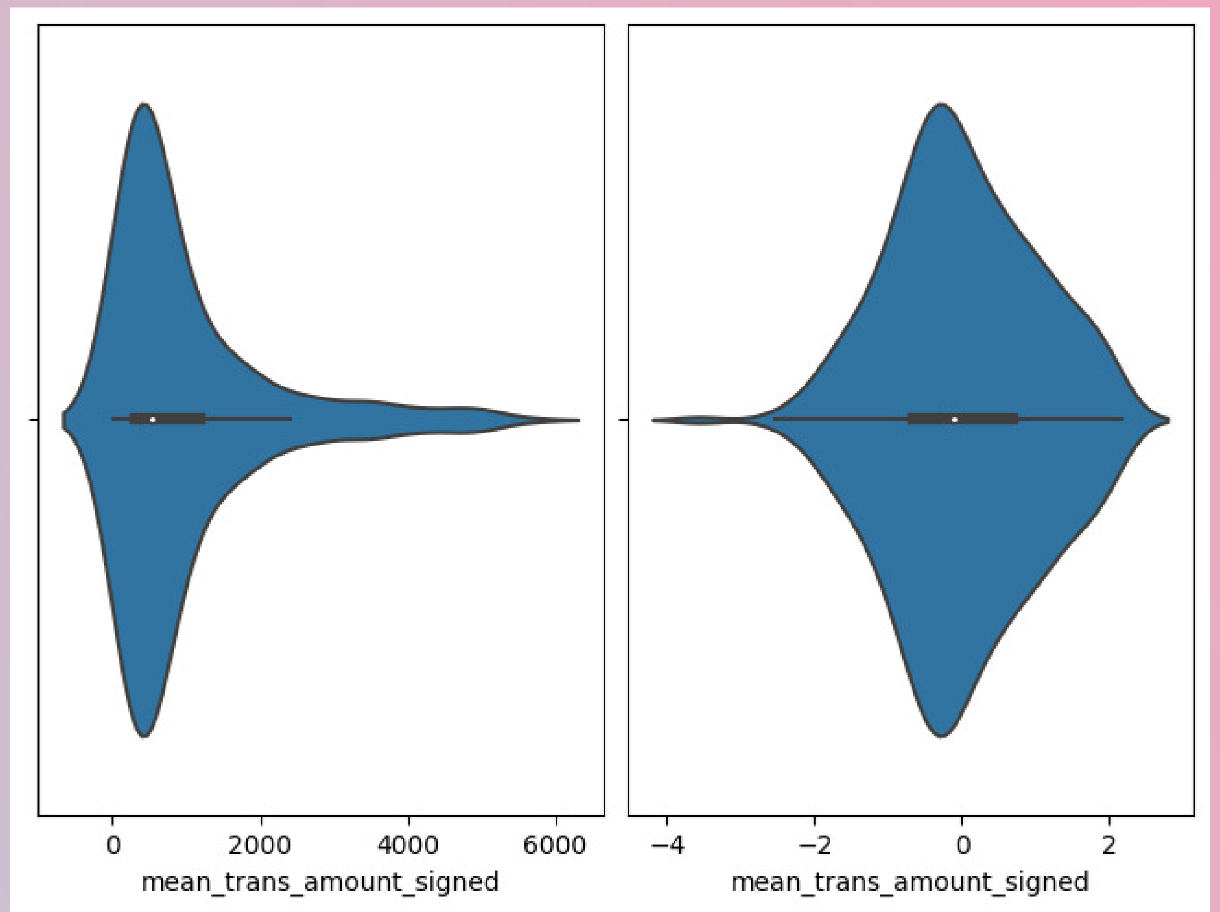


Fig.10 - Mean transaction signed amount before and after PowerTransform non-linear transformation

group 52

Predictive Problem

EXPERIMENTAL SETUP

Feature Definition

Take all previous results into account
Define the input columns and target label

Model Tuning

Past-future train-test split - *TimeSeriesSplit*
Function wrapper to train different models under parameterized conditions

Scoring Metric (the used one was *roc_auc_score*)
Exhaustive (*Grid*) or *Randomized Search* for **hyperparameters** (given the *param_grid*)
Whether to use a Resampler or not (*SMOTETomek*)
used within a **Pipeline** for proper resampling

Try a diversity of Models

DecisionTree (baseline)
RandomForest (robust)
GaussianNB (good for clustered datasets, not the best since it isn't)
LogisticRegression (good for partitioned datasets, PCA indicated it isn't)
GradientBoosting (good to avoid bias)
AdaBoost (good for imbalanced data)
XGB and *LGBM Boost* (not the best given small dataset)
StackingClassifier (good to reassure the output of two or more models)

Calculate and Plot Metrics

Learning Curve, *ROC_AUC* and *Confusion Matrix* plots
Classification Report, *Accuracy* and *ROC_AUC* scores

Predictive Problem

RESULTS

Best Results: Stacking Classifier

> Uses *AdaBoost* and *RandomForest*
> Final estimator is *Logistic Regression*

The recall problem

> Since our positive class is so small, models will struggle to explore the positive prediction. Performance during testing was mostly dictated by the model's ability to overcome this imbalance — *AdaBoost is great in these scenarios*. Resampling made it worse - while the produced model would be more general, the FPR would increase so much so that it would lead to lower *roc_auc* scores in already robust models — simpler ones as *Decision Tree* did get a noticeable bump in performance.

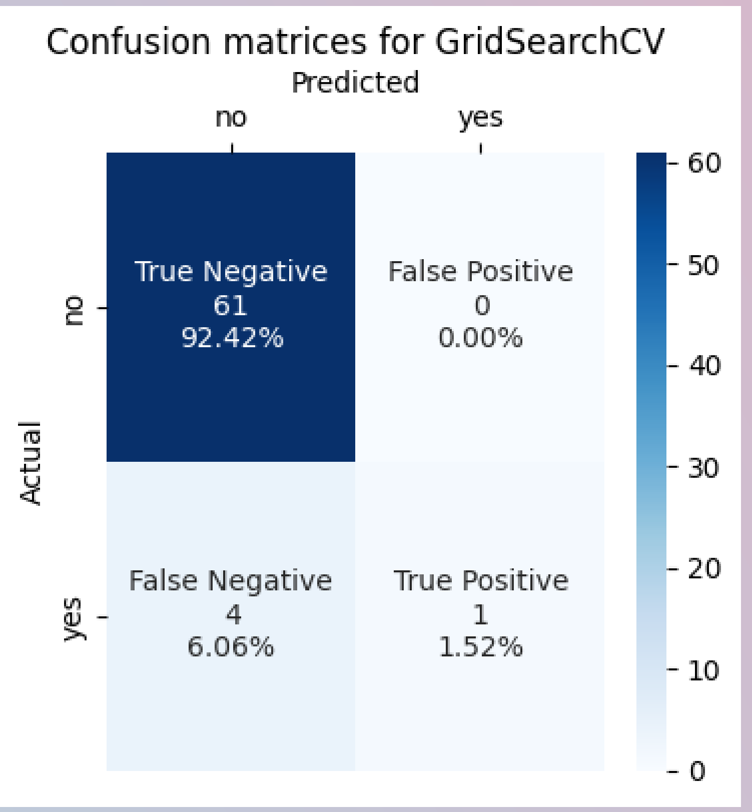


Fig.11 - Confusion matrix for the best classifier obtained

AUC = 0.9115, Accuracy = 0.9394

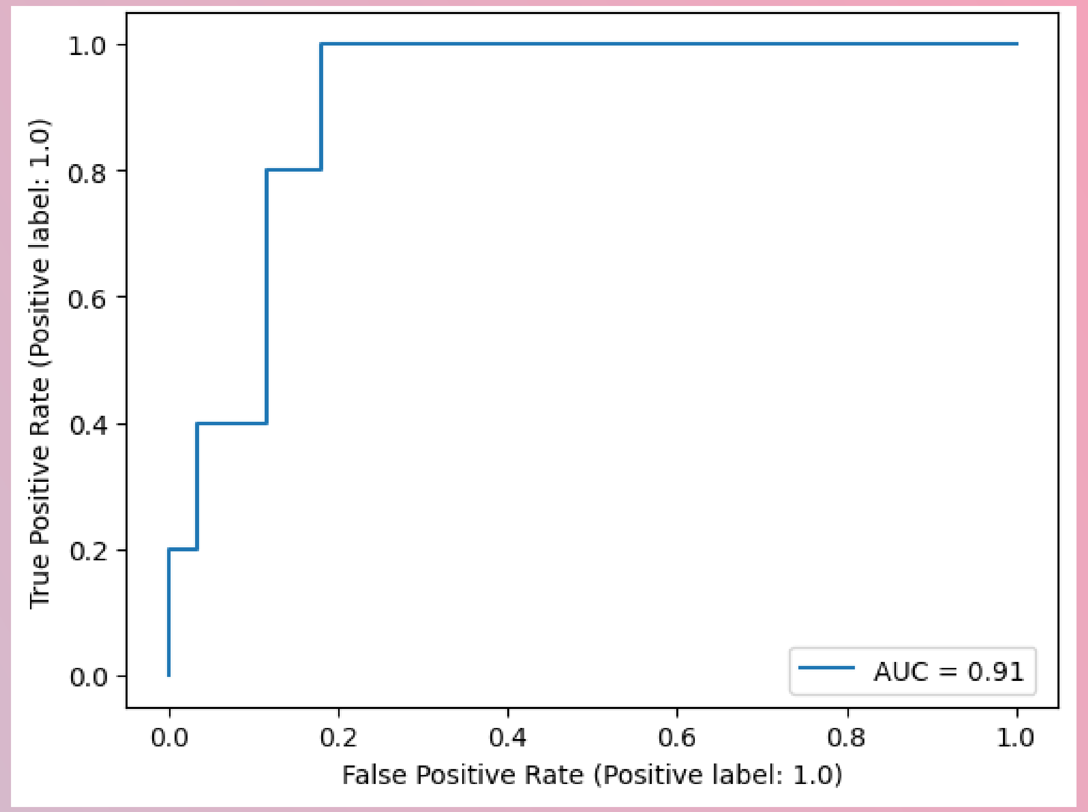


Fig.12 - ROC curve for the best classifier obtained

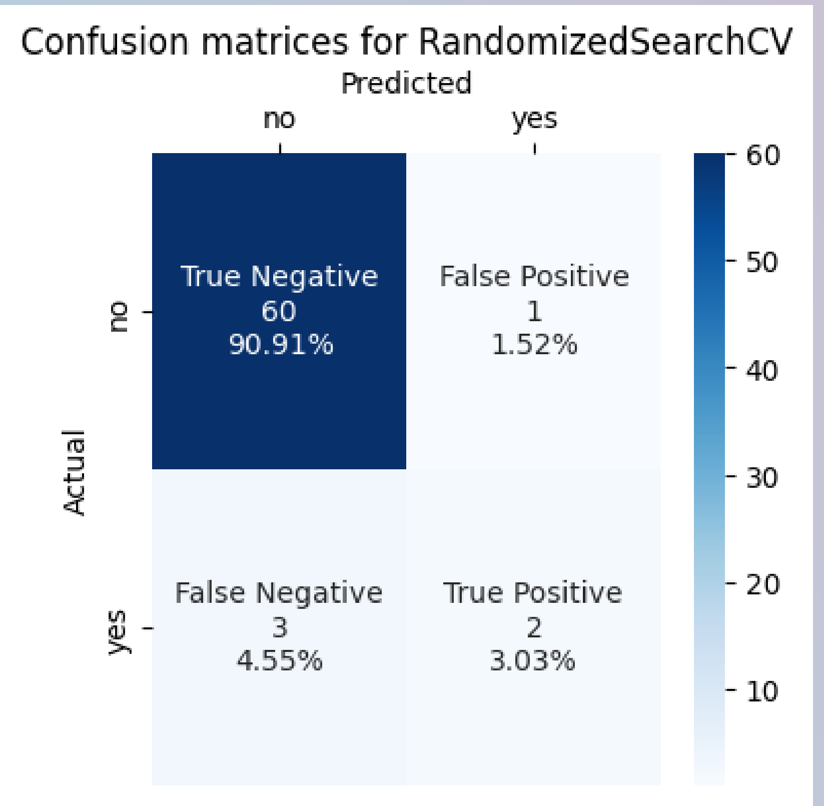


Fig.13 - Confusion matrix for AdaBoostClassifier (no oversampling)

AUC = 0.8721, Accuracy = 0.9394

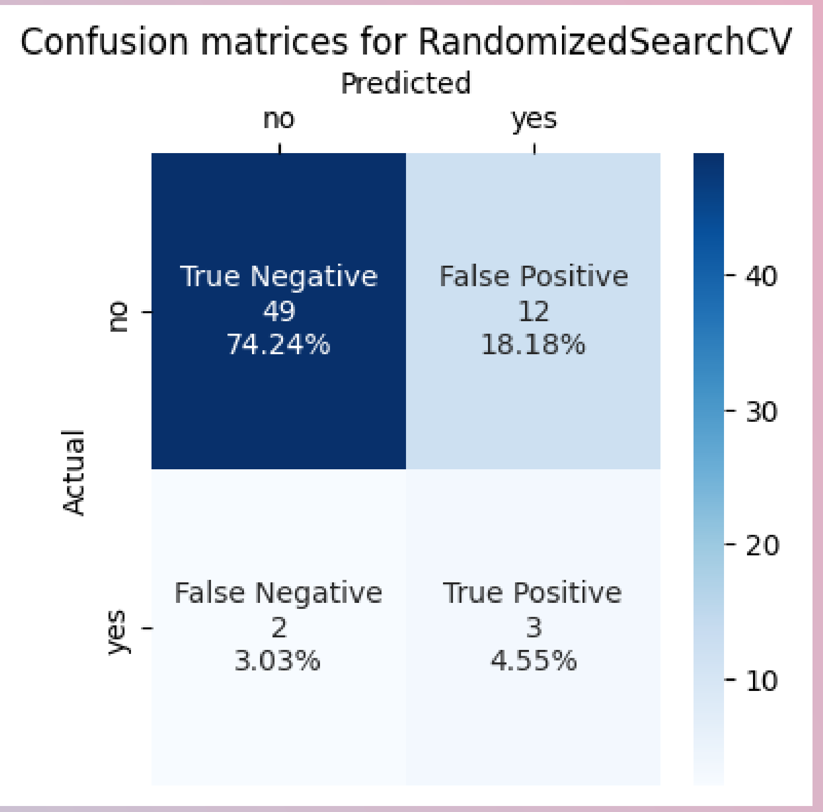


Fig.14 - Confusion matrix for AdaBoostClassifier (oversampled)

AUC = 0.8705, Accuracy = 0.7879

group 52

Predictive Problem

R E S U L T S

Performance of other Models

RandomForest gave good performance out of the box
GradientBoosting, AdaBoost and LGBM gave reasonable scores
DecisionTree required a lot of processing to give reasonable scores
GaussianNB, LogisticRegression and XGB did not give good results

Impact of Preprocessing

Oversampling helped generalize the model — *good for less robust models*
Transforming the data had a good impact on the result — *made the data easier for the model to interpret*
SequentialFeatureSelector improved performance, while all other feature selection techniques were worse than using all available features
Outlier detection did not produce noticeable changes

> given regular thresholds, the ‘outlier’ sample would be too big, impacting the model negatively
>
> given smaller thresholds, the sample would be smaller but the impact was unoticeable

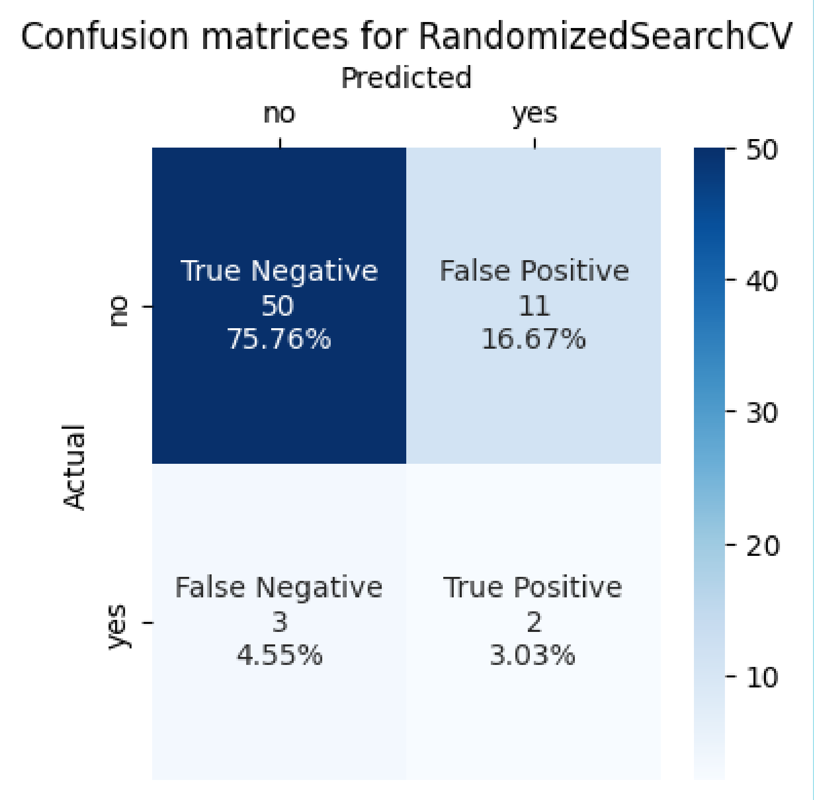


Fig.15 - Confusion matrix for GaussianNB

AUC = 0.7246 Accuracy = 0.7879

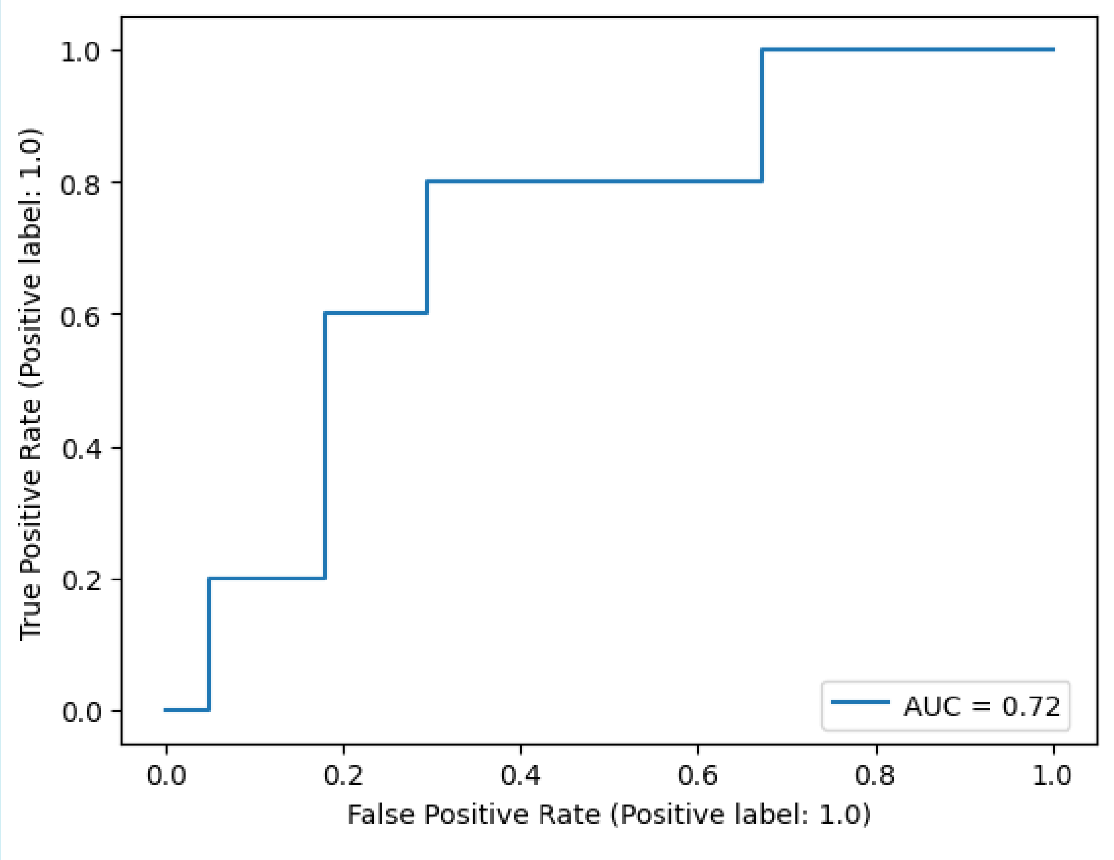


Fig.16 - ROC curve for GaussianNB

group 52

Descriptive Problem

Socio-demographic profile of account owners

Metric: *Euclidean*

Algorithms

- KMeans
- KMedoids (PAM)
- AgglomerativeClustering (average-link) — **Best results**

Tuning number of clusters

- Silhouette method
- Elbow method — **Best results**

Evaluation

- Total average silhouette score
- Variance ratio criterion
- Davies-Bouldin score

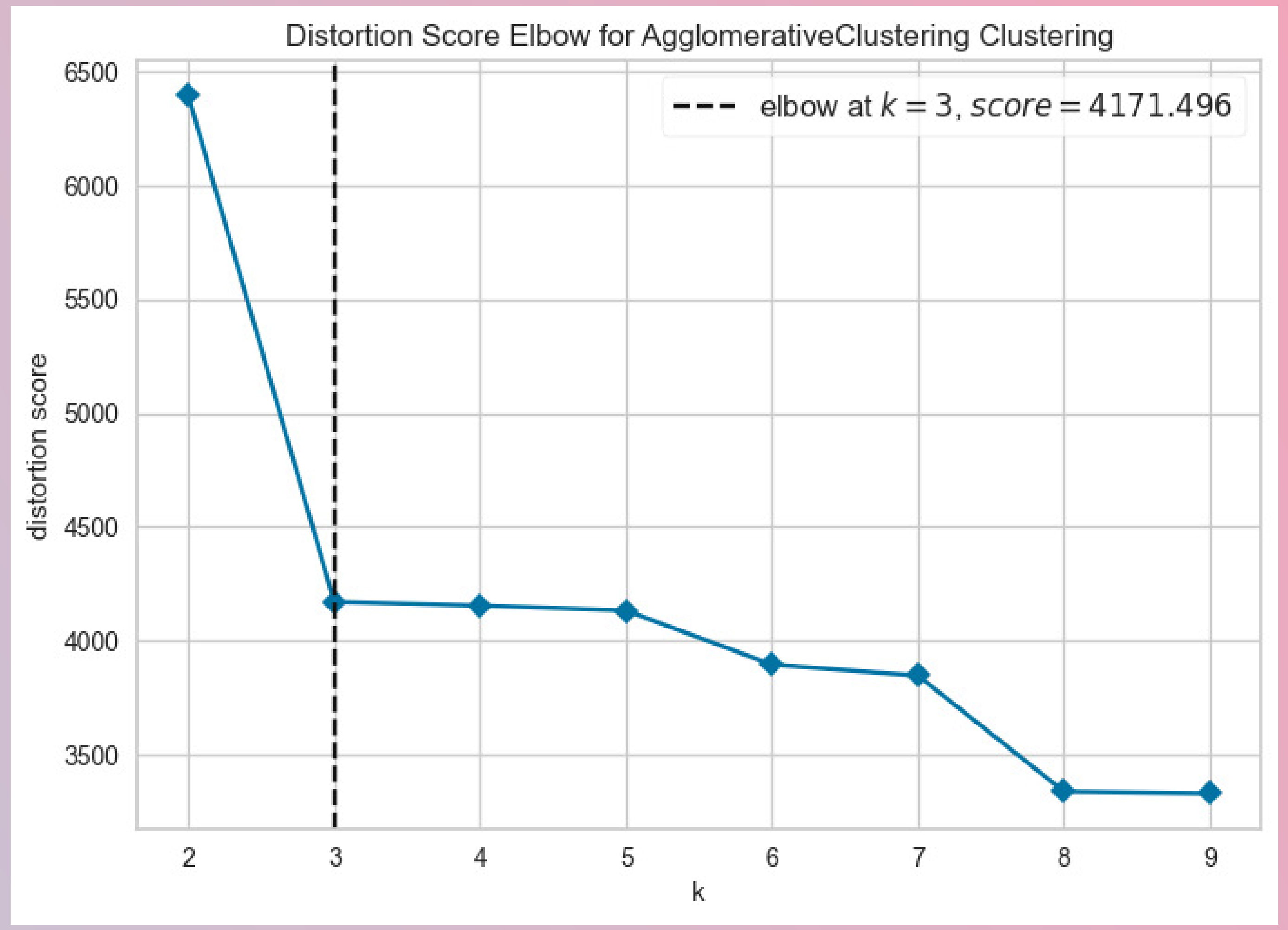


Fig.17 - Distortion Score Elbow for AgglomerativeClustering

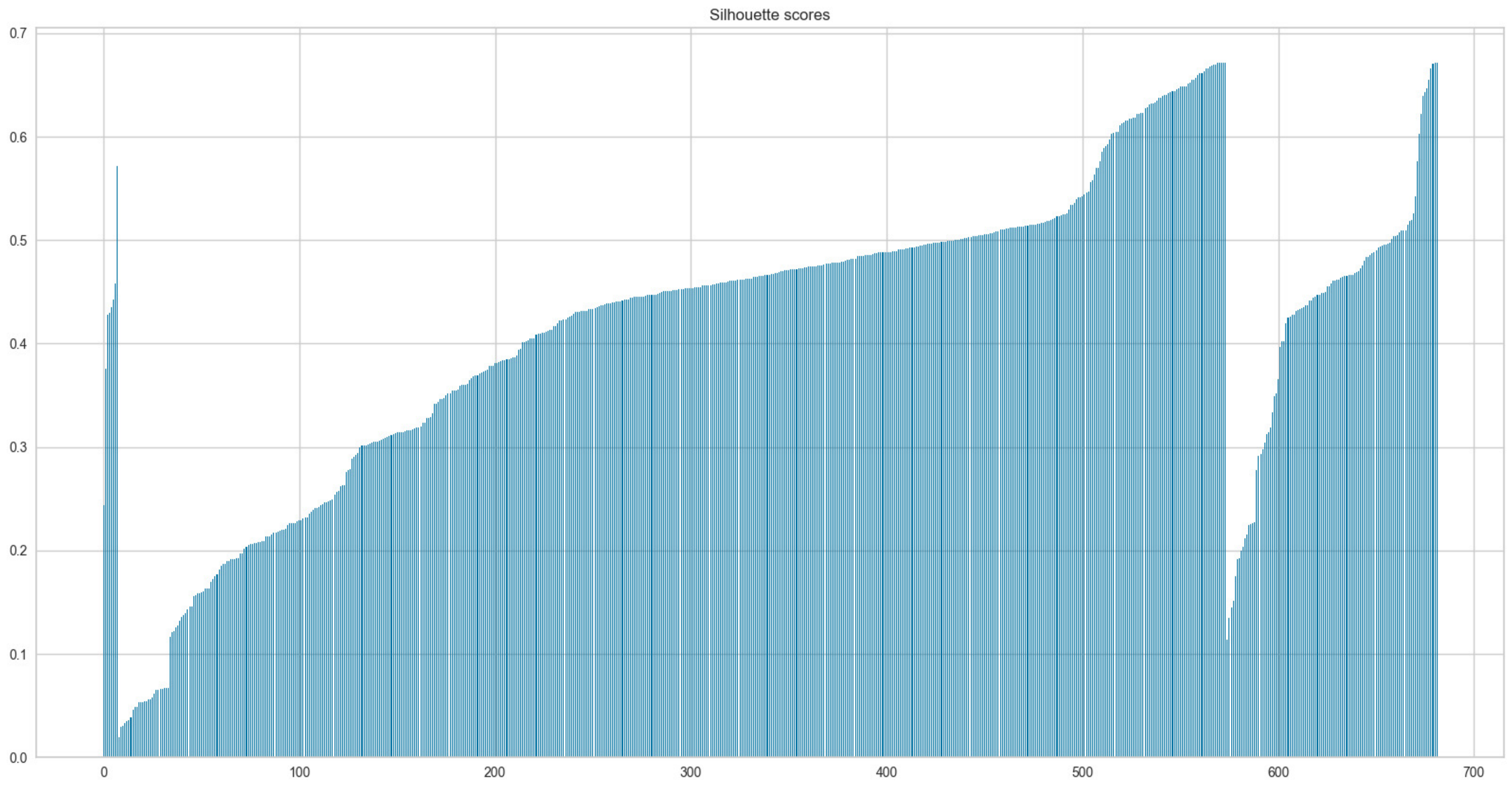


Fig.18 - Silhouette scores for AgglomerativeClustering (3 clusters)

group 52

Conclusions, Limitations And Future Work

DM goals were achieved

> *customer characterization* and *loan prediction*
> good results on Kaggle — *0.96213* (public) and *0.93209* (private)

Iteration is essential

> knowledge about the project and its data evolves over time, so refining previous steps is key

Data is king

> processing can only do so much, and enhancements often lead to decreases in performance because there is very little data, and heavily imbalanced. Complementing this dataset would be one of the top priorities for the future.