Università
degli Studi
di Padova

BANK TARGETING CAMPAIGN

STATISTICAL LEARNING - FINAL PROJECT

INTRODUCTION

Dataset Presentation and Research Objective

Dataset summary

The dataset describes profitable and risk customers for a common bank institution. It contains 4117 observations, defined by 16 variables, of which 6 categorical and 10 continuous.

ID	Unique identifier of the customer	*Employed_days*	Number of days the customer has been employed
Age	Age	*Mortgage*	If the customer has a mortgage (Flag variable)
Income	Annual Income	*Mortgage_amount*	The amount of the mortgage (if active)
Gender	Gender	*Storecar*	Number of store cards headed to the customers
Marital	Marital status of the customer (married/single/divsepwid)	*Loans*	Number of active loans
Numkids	Number of kids	*Loan_amount*	Total amount of all the active loans
Numcards	Number of credit cards headed	*Credit_score*	Score (0.00,1.00) given by the financial instution to the customer's spending behavior
Howpaid	Frequency of the wage payment	*Risk*	Typology of credit risk

Purpose

Aim of the research was to classify profitable customer to be targeted, while trying to avoid risky ones.

CLASSIFICATION TASK
Main steps

Variable Analysis

Each feature was analyzed through a qualitative approach (i.e., graphical perspective) and a quantitative approach (*chi-squared* for categorical variables and *ttest* for continuous variable).
Eventually, a first dimensionality reduction was computed by dropping meaningless variables.

Data Subdivision

The dataset was divided in three partitions: training set, in which the model will be built, validation set, in which the model will be evaluated in order to tune specific parameters, and test set, where the final model will be applied and reviewed.

Model Building

To obtain more consolidated results, four algorithms were employed and compared:

- Logistic Regression
- Linear Discriminant Analysis
- Quadratic Discriminant Analysis
- K - Nearest Neighbors

1 - DATA PREPARATION

IDA - Initial Data Analysis

Missing Data

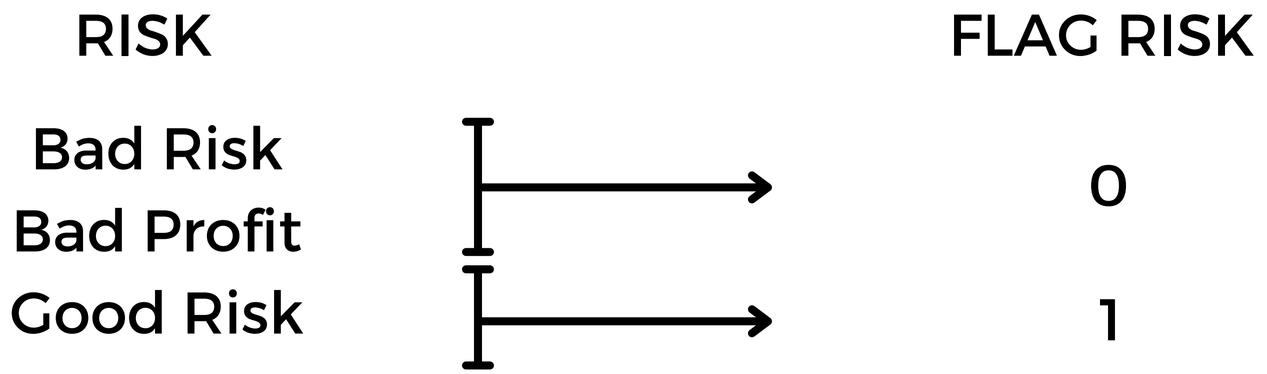
Firstly we checked for missing data, which may arise several problems during the analysis.
Luckily, our dataset did not contain any.

Format Conversion

Even if some variables (*Numkids*, *Numcards*, *Storecar* and *Loans*) appear as numerical, their range is so short that classifying them into categories would be more efficient for us. On the other hand, for a problem of formatting, Credit_Score was initialized as a factor. It represents the trust attributed to a single customer based on its past behavior; assuming values from 0.00 to 1.00, it was "formatted" as double.

Target Transformation

Our target variable *Risk* is a three-level categorical variable, which has been transformed into a binary variable for classification purposes.

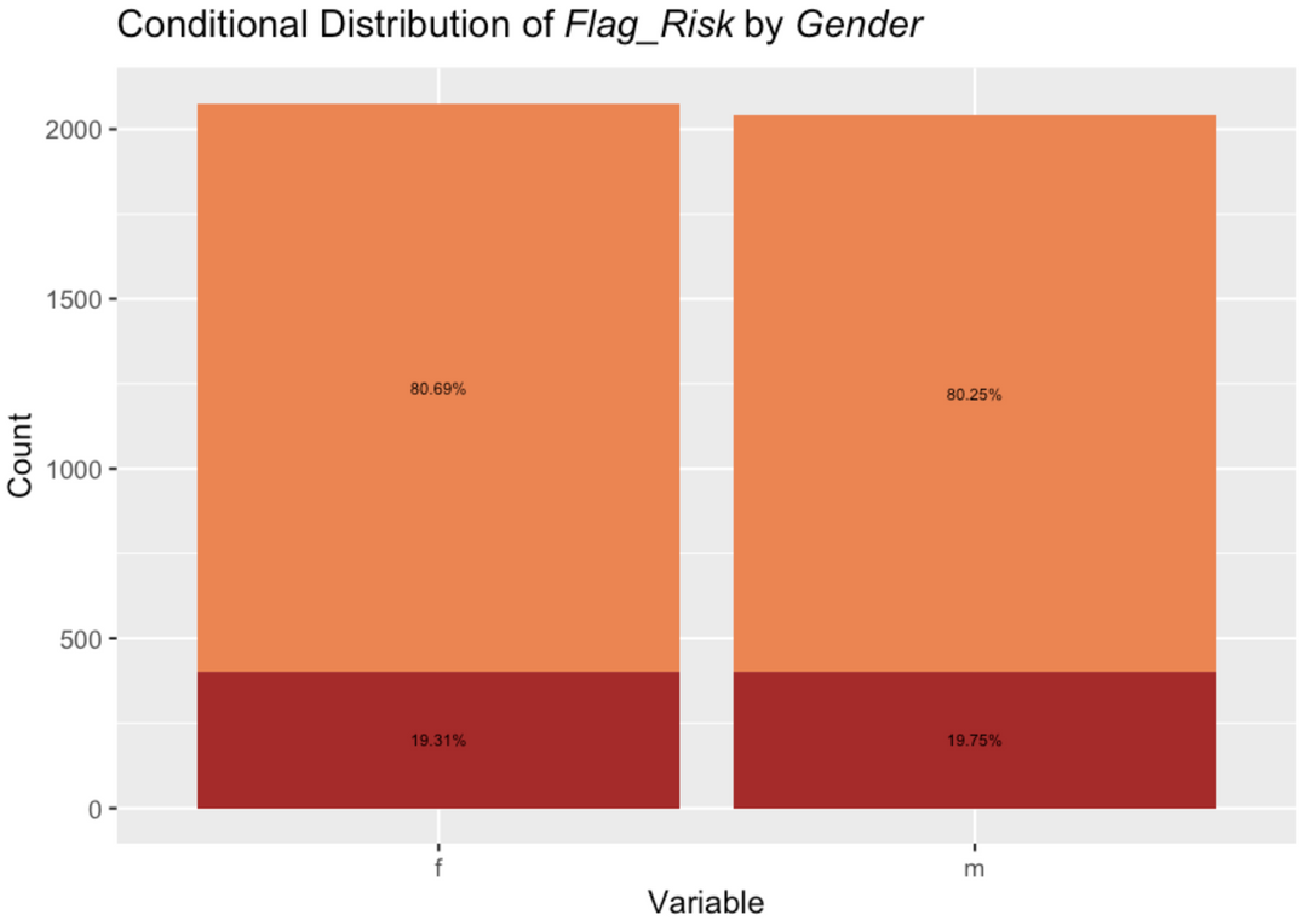


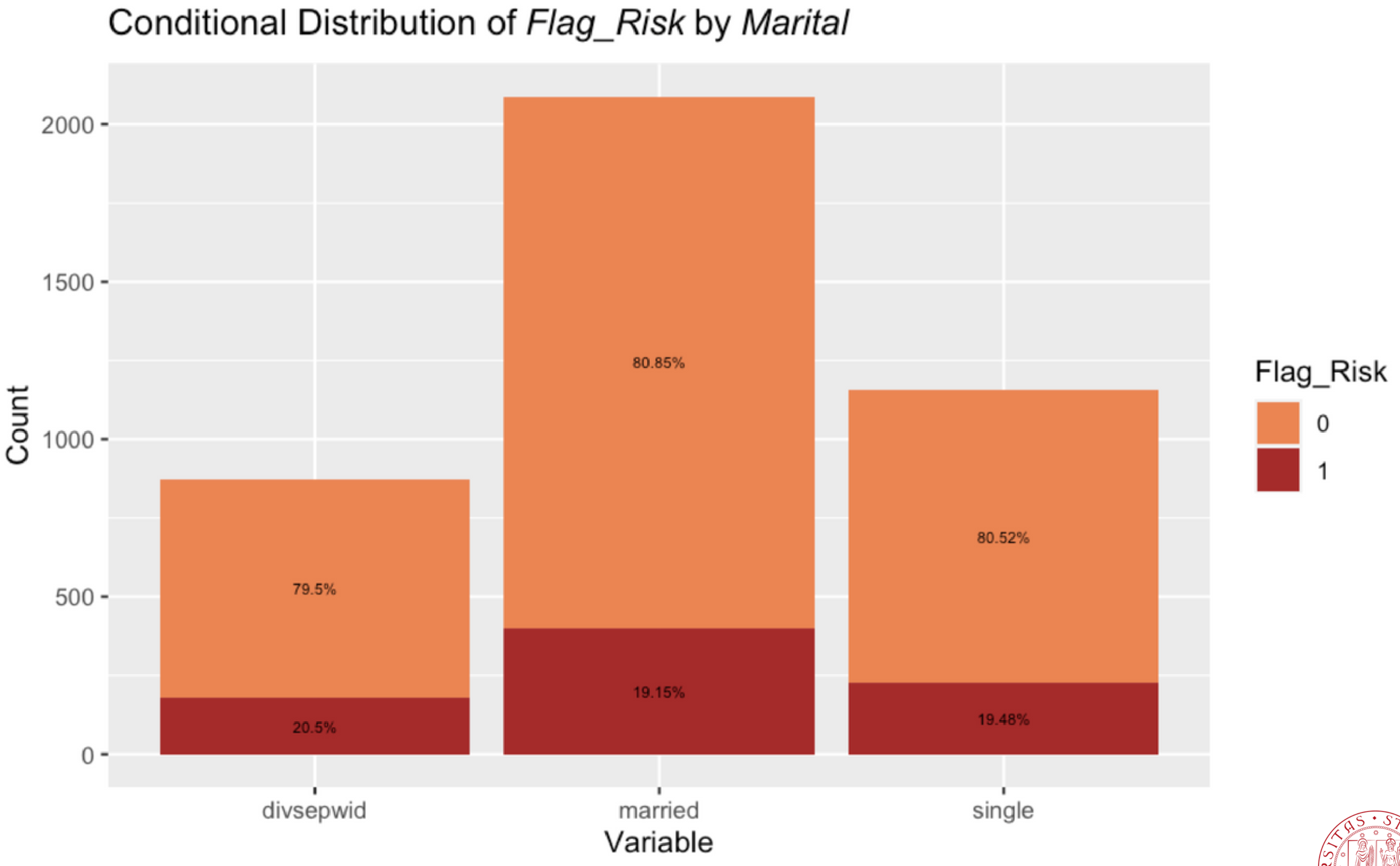
2 - DATA ANALYSIS

Exploratory Data Analysis and Statistics

CATEGORICAL VARIABLES

We directly started by analyzing the distribution of variables with respect to the response variable. For what concern the categorical regressors, *Gender* and *Marital* did not show any dependence. As we can notice from the graph below, the percentages of profitable customers do not change between the different levels of the two variables; this results are confirmed by the chi-squared test, where $p<0.05$.


Conditional Distribution of *Flag_Risk* by *Gender*


Conditional Distribution of *Flag_Risk* by *Marital*

CATEGORICAL VARIABLES

The other variables showed instead a varying level of dependency wrt the target. A brief summary is listed:

- *Numkids*

This variable behaves strangely, with the highest percentage of profitable customer (~40%) linked to the ones with no children; the percentage halves for clients with 4 children (~20%) and it's even lower for clients with less children (1 to 3).

- *Numcards*

The percentage of good customers oscillates around 20% for almost all the values, while touching bottom for clients that possess 3 or 4 credit cards (~4%). This regressor seems partly explanatory, but the logic behind is still unclear.

- *How_paid*

This variable seems to be highly correlated with the profitability of a customer, with a percentage of good customers 3 times higher for the ones who get paid monthly. This aspect reflects stability in the customer "lifestyle", therefore we could expect such outcome.

- *Storecar*

Customers with no store cards are mostly profitable for the bank, with a percentage of ~80%. Such levels drops as the number of store cards increases, reaching a ~15% of good customer when store car = 5.

- *Loans*

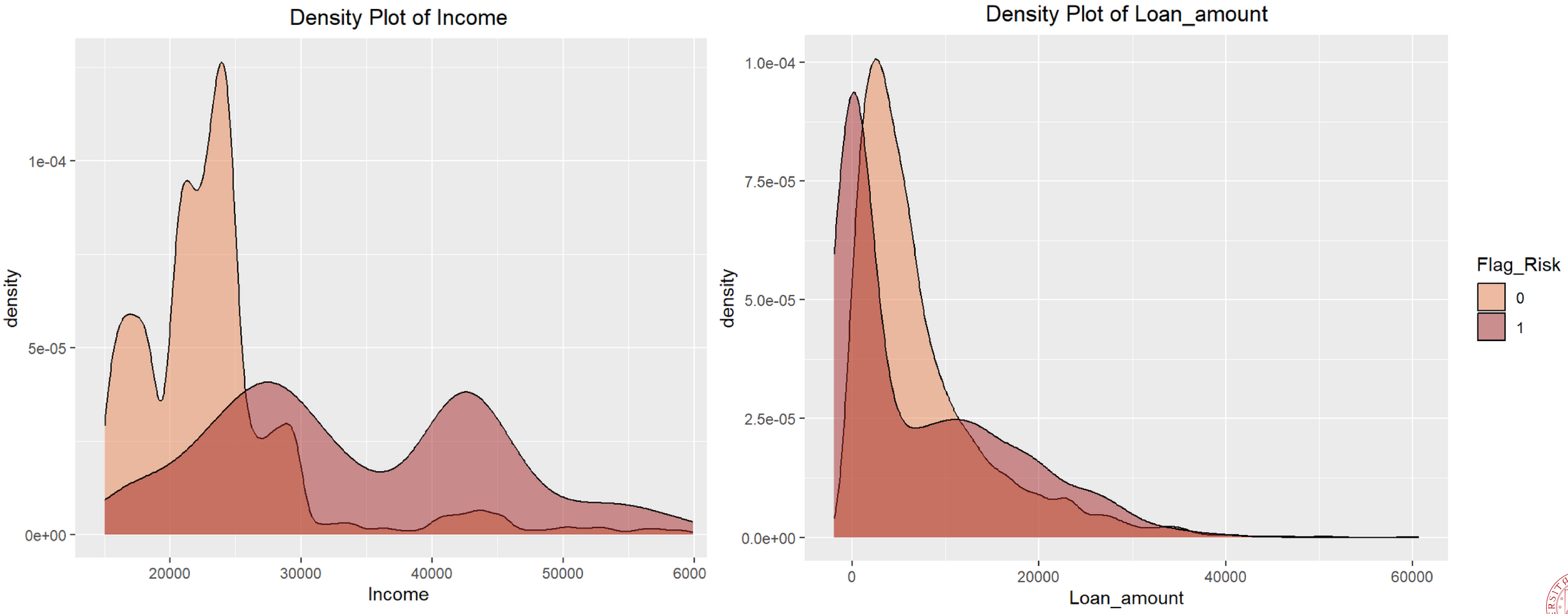
Not having a loan is a synonym of "avoiding risks" for a bank, hence the 70% of customers within this subset is profitable. Even if more loans means more risk, clients with 3 loans have a lower percentage of riskiness wrt the ones with 1 or 2.

- *Mortgage*

Having a mortgage implies a percentage of good risk 2 times higher than not having it.

CONTINUOUS VARIABLES

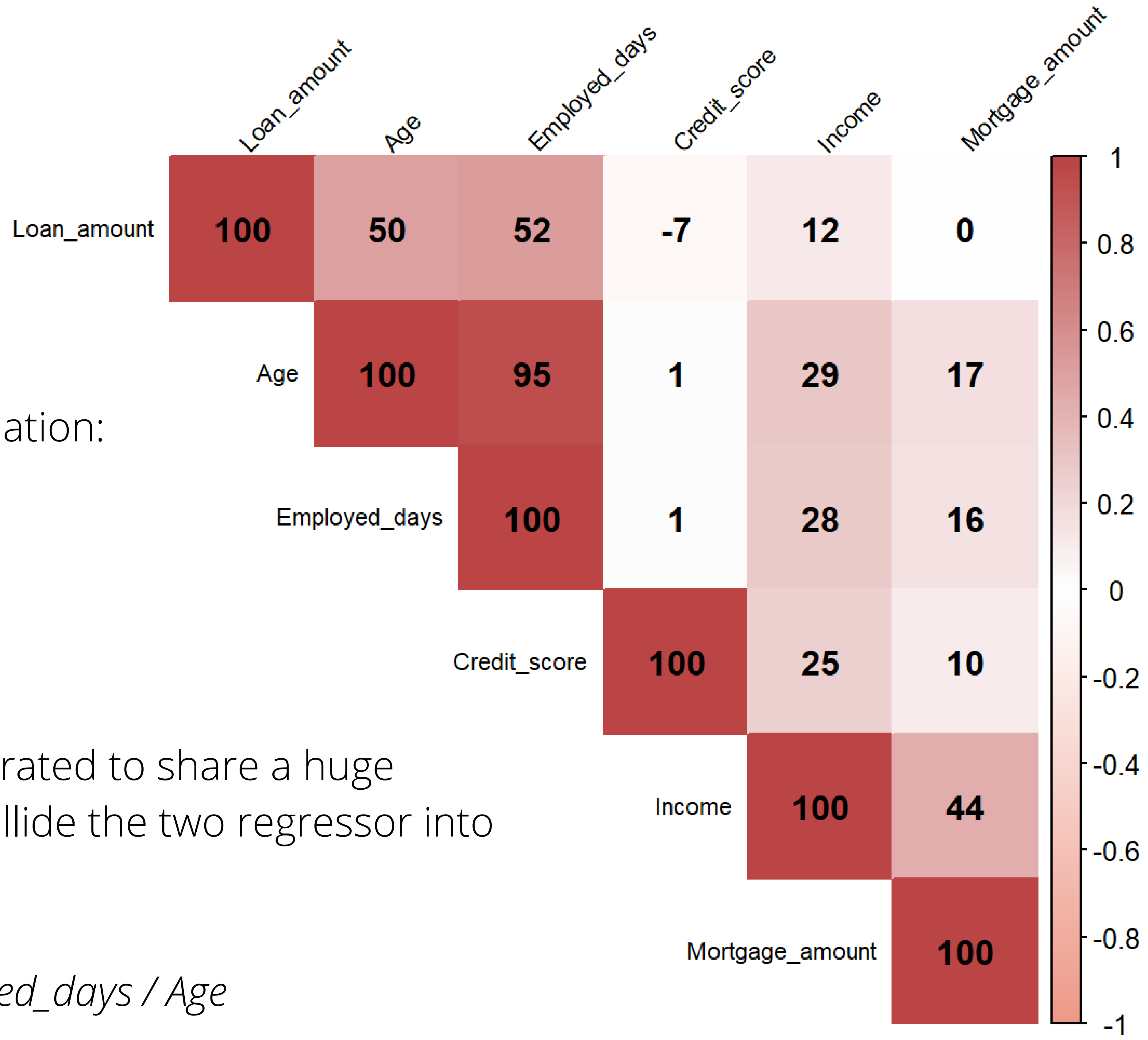
For what concern continuous variables we plotted their distribution depending on their association to the target variable. Subsequently, a ttest was performed to measure such relationship.
The major part were significantly related to *Flag_Risk*; the only exception is *Loan_Amount*.



Density Plot of Income

Density Plot of Loan_amount

CORRELATIONS



	Loan_amount	Age	Employed_days	Credit_score	Income	Mortgage_amount
Loan_amount	100	50	52	-7	12	0
Age		100	95	1	29	17
Employed_days			100	1	28	16
Credit_score				100	25	10
Income					100	44
Mortgage_amount						100

Some variables showed significant level or correlation:

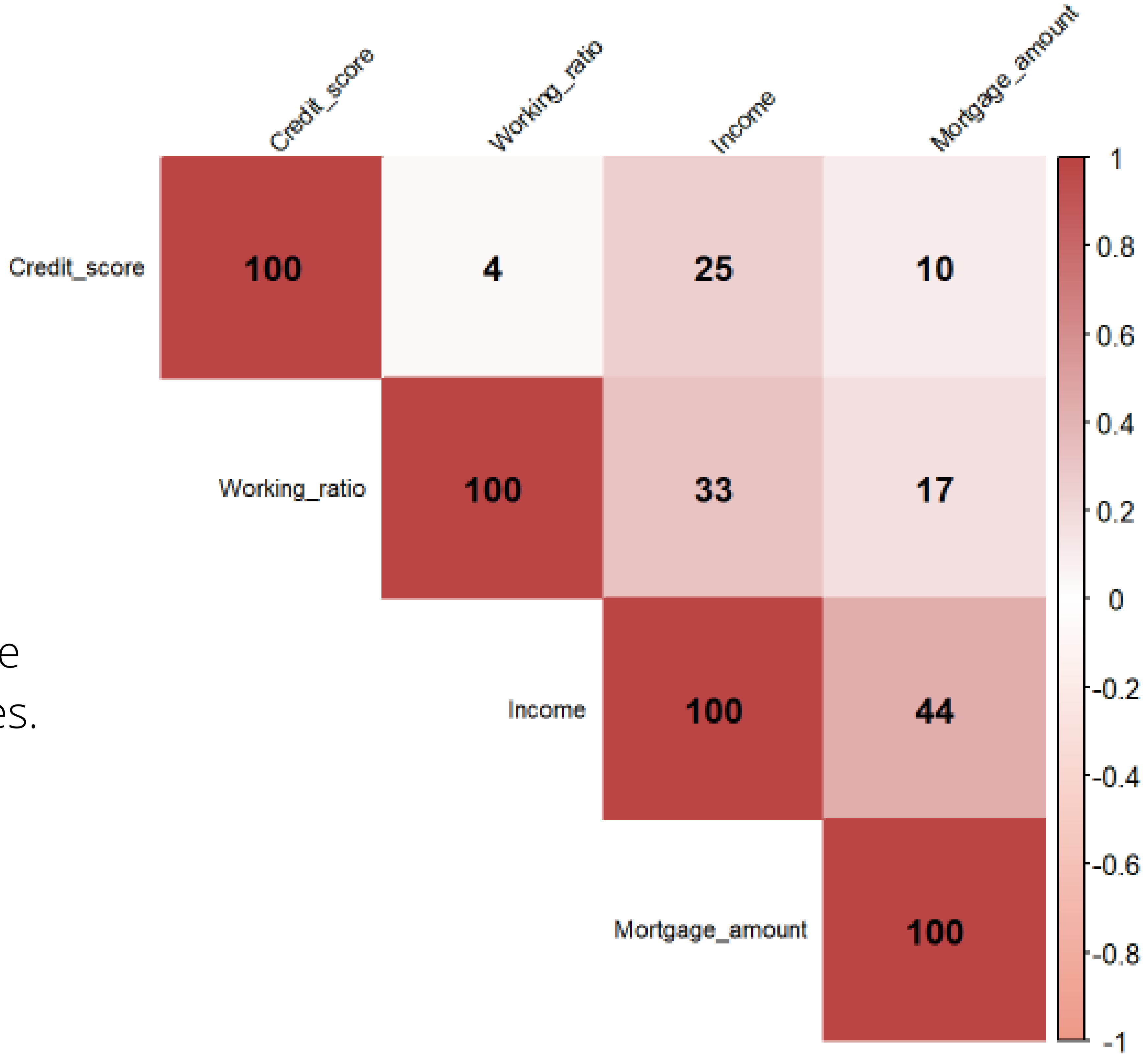
- *Loan Amount - Age*
- *Loan Amount -Employed_day*
- *Age - Employed_days*

Eventually, only *Age* and *Employed_days* demonstrated to share a huge amount of variance. Therefore, we decided to collide the two regressor into a single one, by computing their ratio.

Working_Ratio = Employed_days / Age

CORRELATIONS - 2

After merging together the 2 variables displaying the highest correlation value, the corrleation matrix displays moderate values.



	Credit_score	Working_ratio	Income	Mortgage_amount
Credit_score	100	4	25	10
Working_ratio		100	33	17
Income			100	44
Mortgage_amount				100

3 - MODEL BUILDING

PARAMETRIC ALGORITHMS

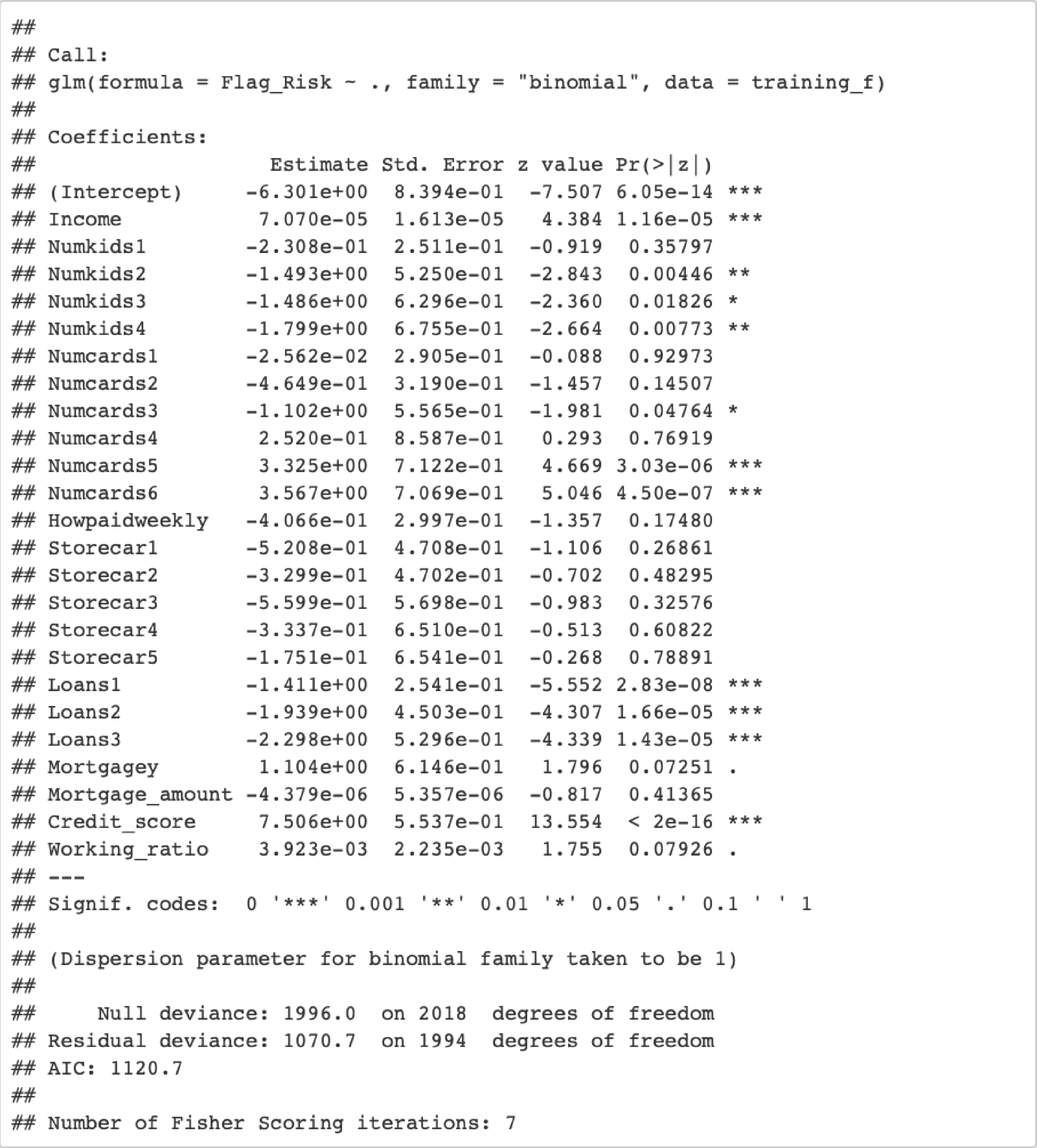
ALGORITHMS

To classify our observations we initially exploited three different algorithms:

- Logistic Regression →
- Linear Discriminant Analysis
- Quadratic Discriminant Analysis

From the LR summary we notice how 2 variables are not significant towards the explanation of the target variable:

- *Storecar*
- *Mortgage_amount*



```
##
## Call:
## glm(formula = Flag_Risk ~ ., family = "binomial", data = training_f)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -6.301e+00  8.394e-01  -7.507 6.05e-14 ***
## Income           7.070e-05  1.613e-05   4.384 1.16e-05 ***
## Numkids1        -2.308e-01  2.511e-01  -0.919  0.35797
## Numkids2        -1.493e+00  5.250e-01  -2.843  0.00446 **
## Numkids3        -1.486e+00  6.296e-01  -2.360  0.01826 *
## Numkids4        -1.799e+00  6.755e-01  -2.664  0.00773 **
## Numcards1       -2.562e-02  2.905e-01  -0.088  0.92973
## Numcards2       -4.649e-01  3.190e-01  -1.457  0.14507
## Numcards3       -1.102e+00  5.565e-01  -1.981  0.04764 *
## Numcards4        2.520e-01  8.587e-01   0.293  0.76919
## Numcards5        3.325e+00  7.122e-01   4.669 3.03e-06 ***
## Numcards6        3.567e+00  7.069e-01   5.046 4.50e-07 ***
## Howpaidweekly   -4.066e-01  2.997e-01  -1.357  0.17480
## Storecar1       -5.208e-01  4.708e-01  -1.106  0.26861
## Storecar2       -3.299e-01  4.702e-01  -0.702  0.48295
## Storecar3       -5.599e-01  5.698e-01  -0.983  0.32576
## Storecar4       -3.337e-01  6.510e-01  -0.513  0.60822
## Storecar5       -1.751e-01  6.541e-01  -0.268  0.78891
## Loans1          -1.411e+00  2.541e-01  -5.552 2.83e-08 ***
## Loans2          -1.939e+00  4.503e-01  -4.307 1.66e-05 ***
## Loans3          -2.298e+00  5.296e-01  -4.339 1.43e-05 ***
## Mortgagey        1.104e+00  6.146e-01   1.796  0.07251 .
## Mortgage_amount -4.379e-06  5.357e-06  -0.817  0.41365
## Credit_score     7.506e+00  5.537e-01  13.554  < 2e-16 ***
## Working_ratio    3.923e-03  2.235e-03   1.755  0.07926 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1996.0  on 2018  degrees of freedom
## Residual deviance: 1070.7  on 1994  degrees of freedom
## AIC: 1120.7
##
## Number of Fisher Scoring iterations: 7
```

ALGORITHMS - 2

We plot now the confusion matrix of the different models.

LOGISTIC

	Reference	
Prediction	**0**	**1**
0	672	65
1	24	103

	Metrics
Accuracy	89,70%
Sensitivity	61,31%
Specifity	96,55%
Precision	81,10%

Treshold	**0.5**

LDA

	Reference	
Prediction	**0**	**1**
0	648	52
1	48	116

	Metrics
Accuracy	88,43%
Sensitivity	69,05%
Specifity	93,10%
Precision	70,73%

Treshold	**0.3**

QDA

	Reference	
Prediction	**0**	**1**
0	591	36
1	105	132

	Metrics
Accuracy	83,68%
Sensitivity	78,57%
Specifity	84,91%
Precision	55,70%

Treshold	**0.7**

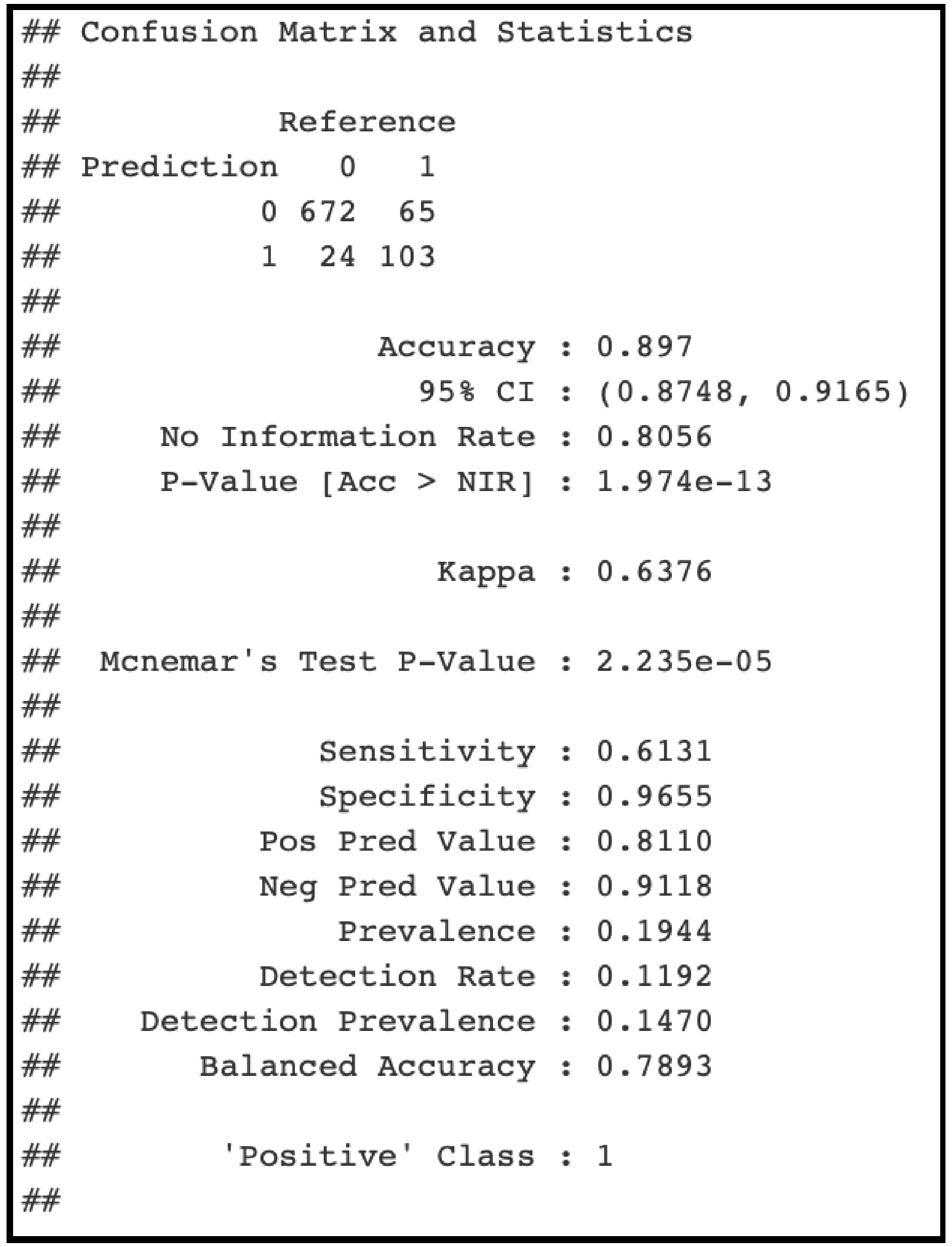
STEP-WISE APPROACH

We implemented a step-wise approach to perform feature selection on our dataset.
3 different approaches were implemented:

- Forward
- Backward
- Bi-directional

When applied to the configured models, the backward direction returned better results. As a matter of fact, the forward approach kept all the variables in each run, being penalized for this; alternatively, the combined approach (“both”), took always into consideration the backward approach, emulating it.

The results confirmed the inital assumptions after the first LR:
Storecar and **Mortage_amount** can be removed



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 672  65
##          1  24 103
##
##                Accuracy : 0.897
##                  95% CI : (0.8748, 0.9165)
##     No Information Rate : 0.8056
##     P-Value [Acc > NIR] : 1.974e-13
##
##                   Kappa : 0.6376
##
##  Mcnemar's Test P-Value : 2.235e-05
##
##             Sensitivity : 0.6131
##             Specificity : 0.9655
##          Pos Pred Value : 0.8110
##          Neg Pred Value : 0.9118
##              Prevalence : 0.1944
##          Detection Rate : 0.1192
##    Detection Prevalence : 0.1470
##       Balanced Accuracy : 0.7893
##
##        'Positive' Class : 1
##
```

BALANCING

To reduce the risk of over-estimating the majority class we balanced the training set.
The validation and the test set, instead, will be kept with the original proportion, since they must reflect the reality.

We adopted the *SMOTE* - *Synthetic Minority Oversampling Technique,* which artificially generates new observations close to the original ones. We use the term "close" because SMOTE is able to point in the space data via the geometrical representation of its features, thus defining a subspace over which generating new observations (it exploits the KNN algorithm).

20/80 → 50/50

CROSS - VALIDATION

Subsequently, to obtain results more robust, we introduced the cross-validation method by evaluating the model's performance on multiple folds of the data.

− DATA VARIABILITY

+ GENERALIZATION

To do this we wrote a function in which k iterations are run, and consequently k different partitions are generated. Even if we do not generate mutual exclusive folds as in the k-fold cv approach, we're pretty confident that a high number of iteration will still guarantee statistically reliable results.

Eventually, at each iteration we saved three useful metrics for the upcoming model evaluation/comparison:

- **AUC**
- **PRECISION**
- **PROFITS**

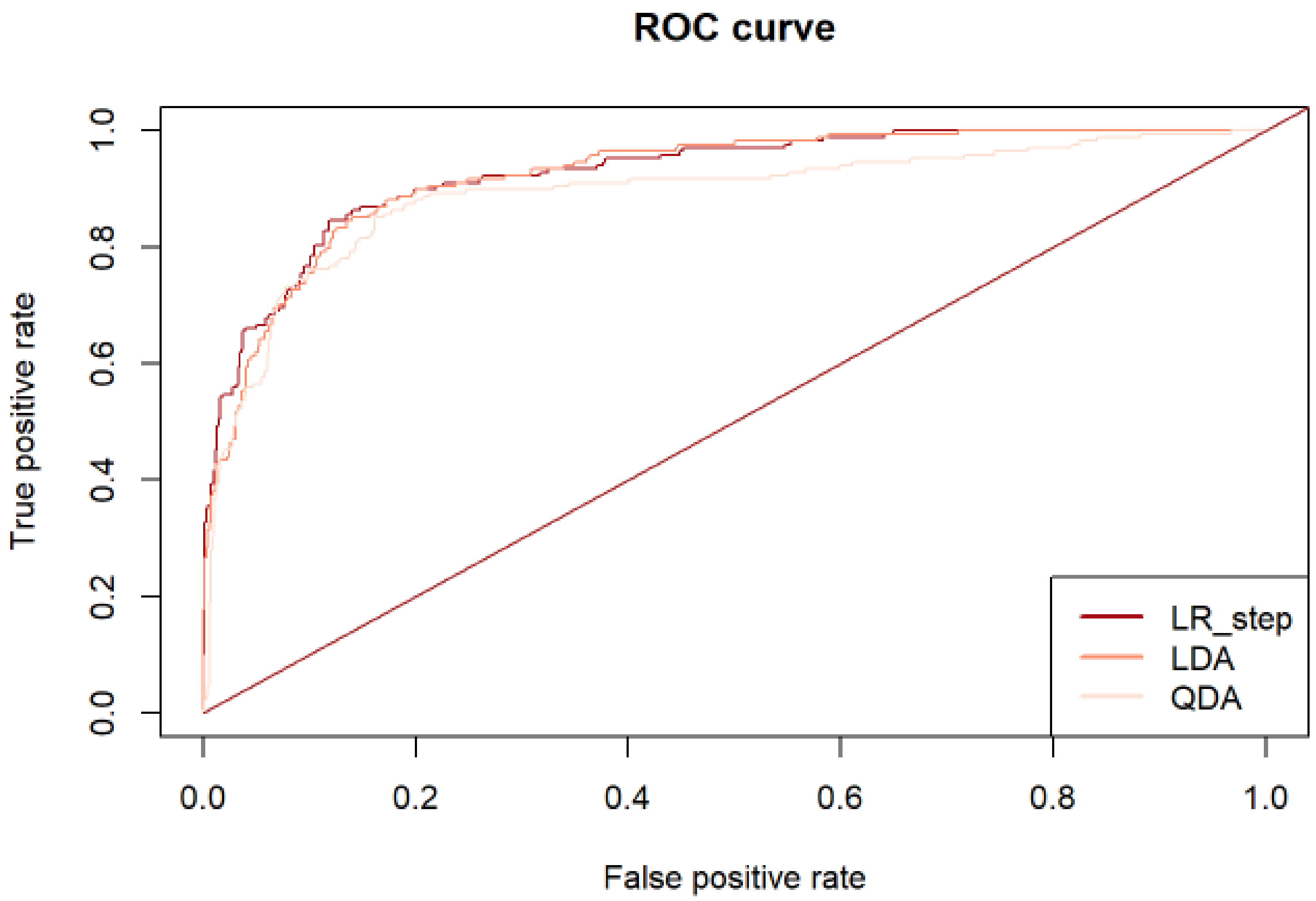
UNIVERSITÀ DEGLI STUDI DI PADOVA

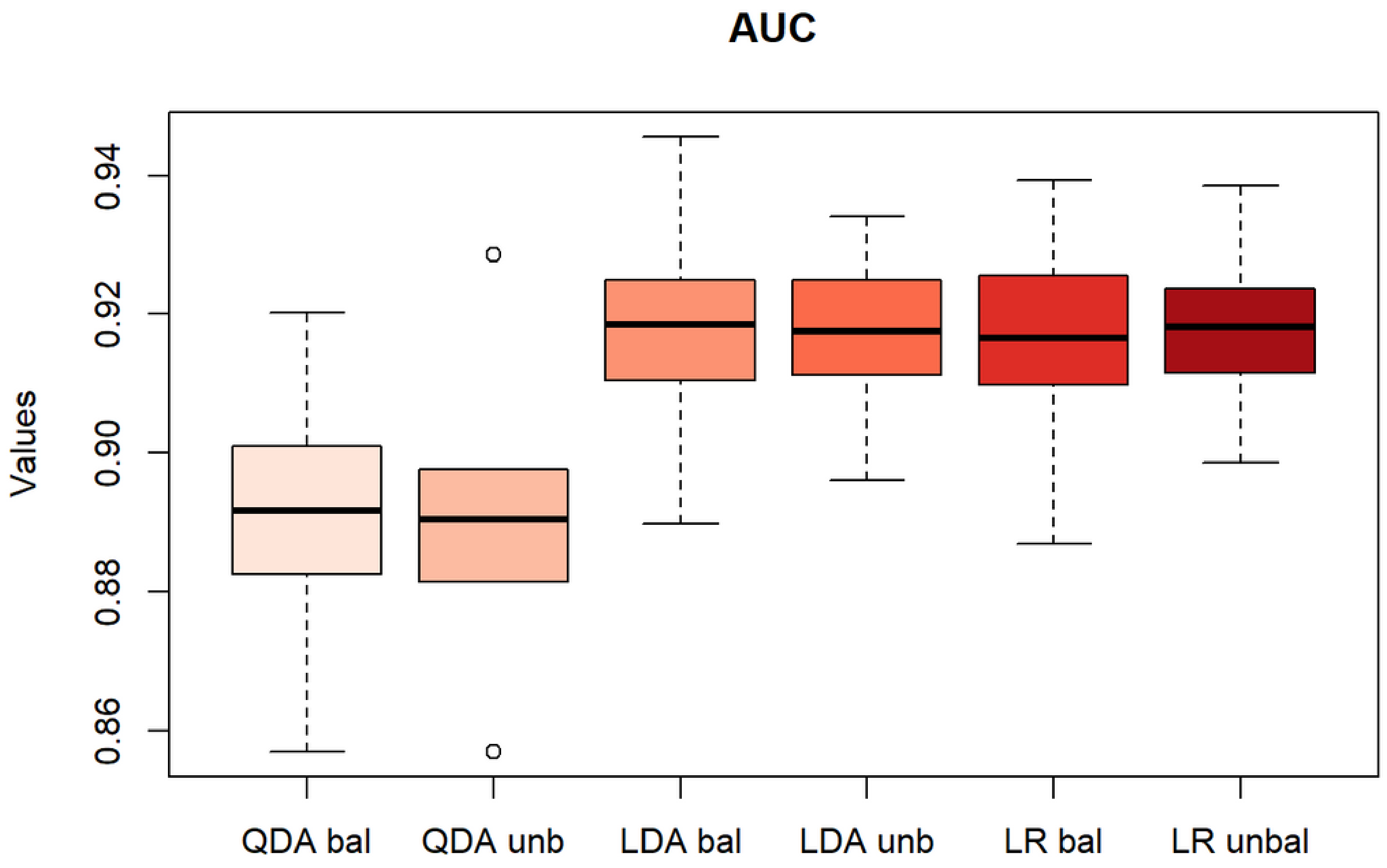
4- 1ST MODEL COMPARISON

METRICS

ROC - AUC

By considering the area included below the ROC curve, we can measure the overall performance of the model across all possible thresholds. The 100 runs of CV gives us more stable results under different training and validation sets.

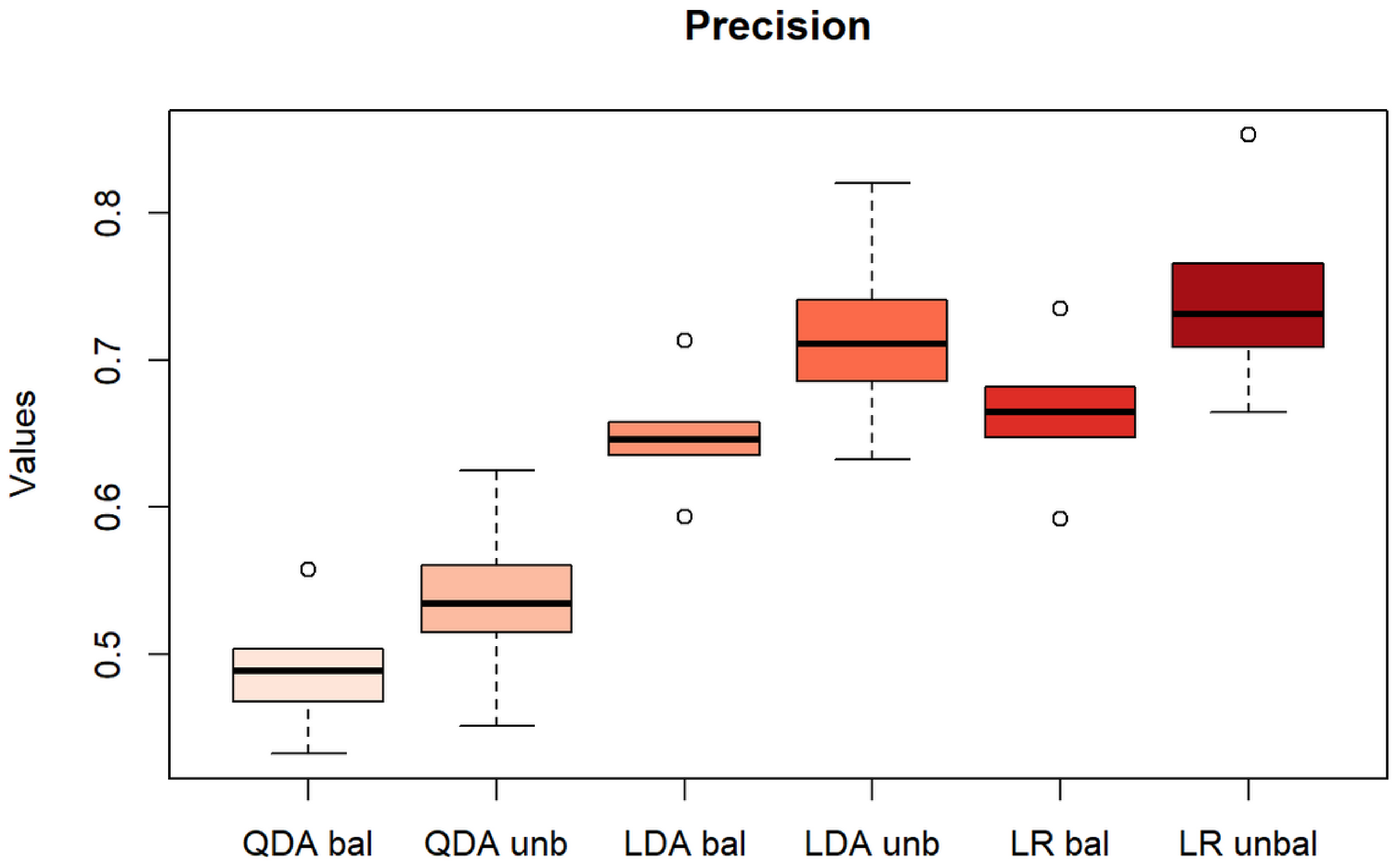




METRICS

Precision

Since we had no direction towards which directing our study, we assumed a scenario in which the company has a low budget and prefers to target few but profitable customers, rather than targeting a larger pool but with less attention. Therefore, we centered our analysis on improving the Positive Predicted Value (or precision).

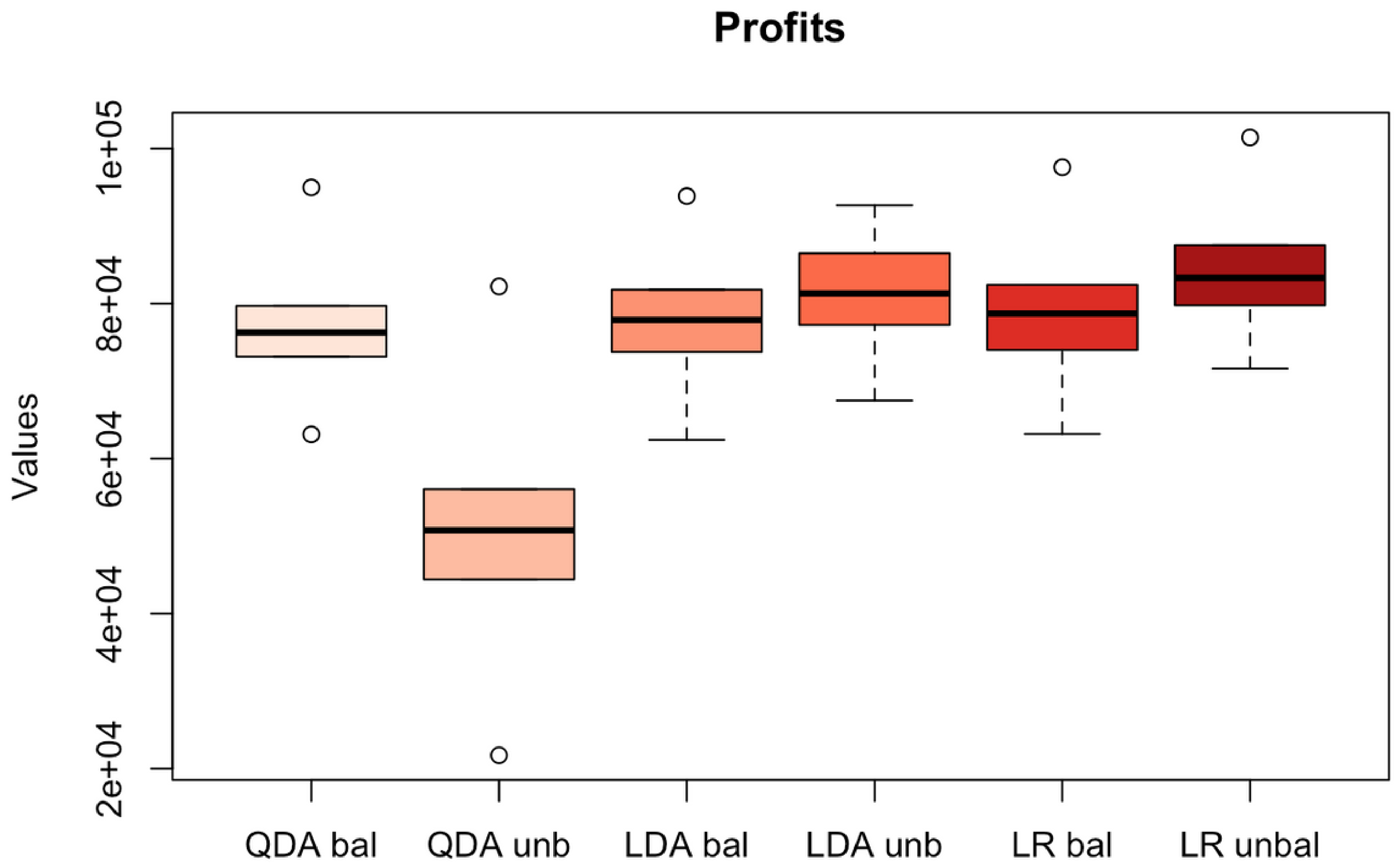


METRICS

Profits

In order to have a more concrete point of view about our results, as well as to facilitate the decision making process, we introduced a function to evaluate the monetary advantages offered by the model.
This function is strictly related to the precision of our model, and calculates the profits as:

$$Profits = -10 * PP + 1000 * TP - 700 * FP$$



Not having any clue on the real monetary aspects of such campaign we used arbitrary value; the institution will then be able to proceed to define it according to its needs.

As we can notice, the reduced LR trained on the original training set (i.e.: not balanced), achieved the highest performances among all the other models.

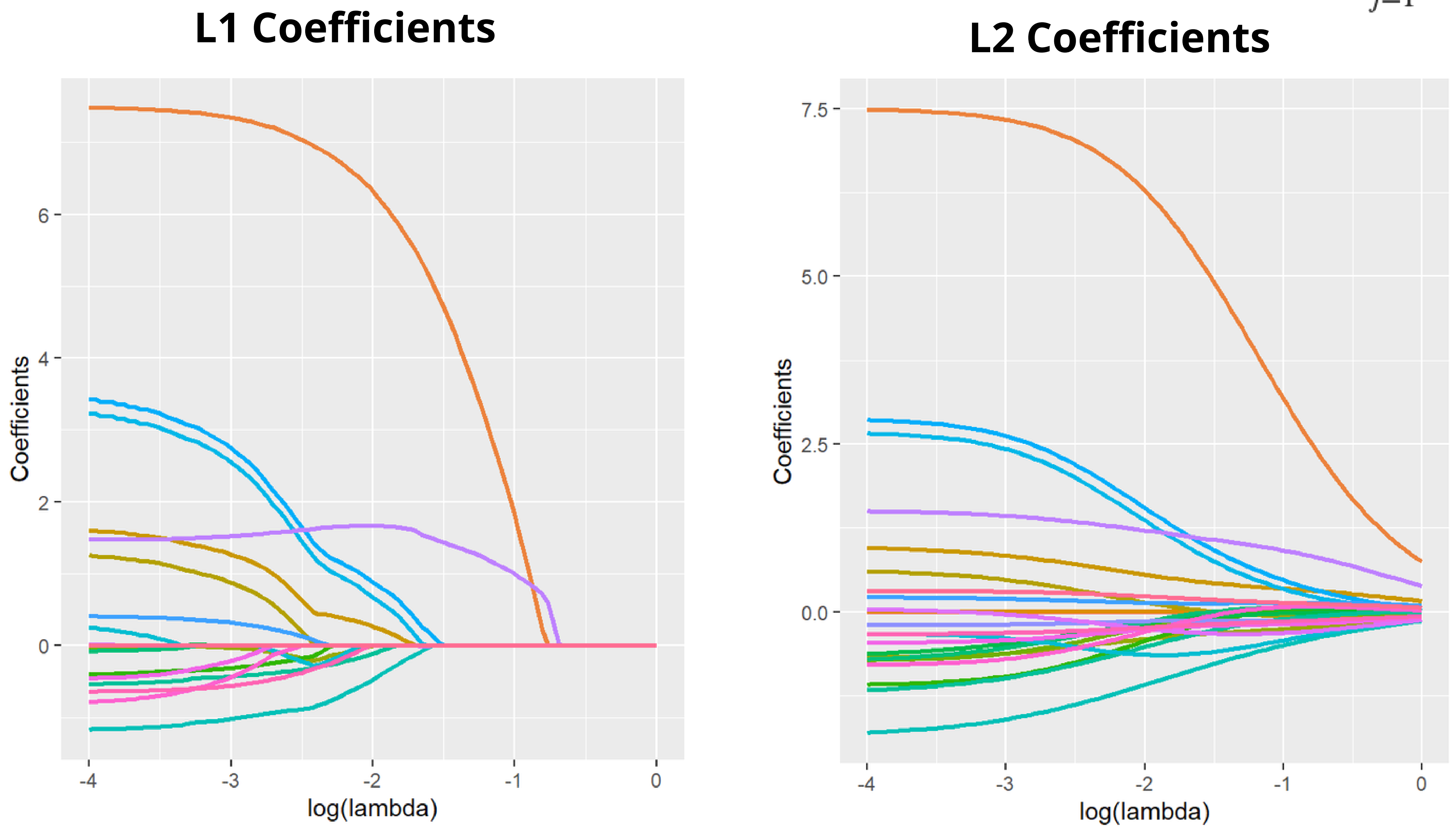
5 - MODEL IMPROVEMENT

Logistic Regression - Regularization

REGULARIZED REGRESSION

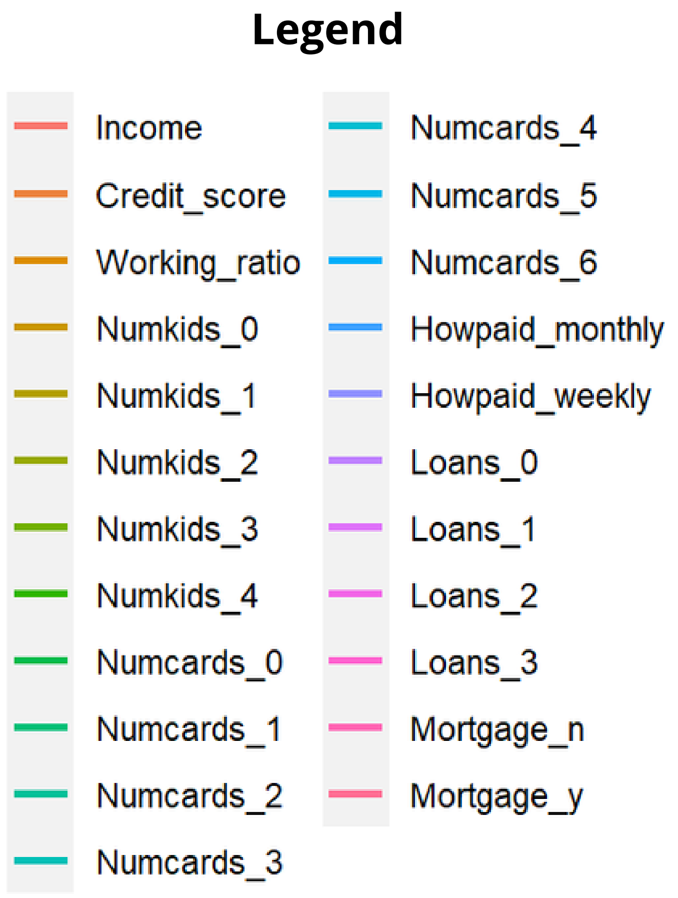
We proceed now to regularize the resulting logistic regression model by implementing two different regularization techniques:

- **LASSO** $$\hat{\beta}^L = arg\ min_b\ \{(y - X\beta)^T(y - X\beta) + \lambda \sum_{j=1}^{p} |\beta_j|\}$$
- **RIDGE** $$\hat{\beta}^R = arg\ min_b\ \{(y - X\beta)^T(y - X\beta) + \lambda \sum_{j=1}^{p} \beta_j^2\}$$



L1 Coefficients

L2 Coefficients

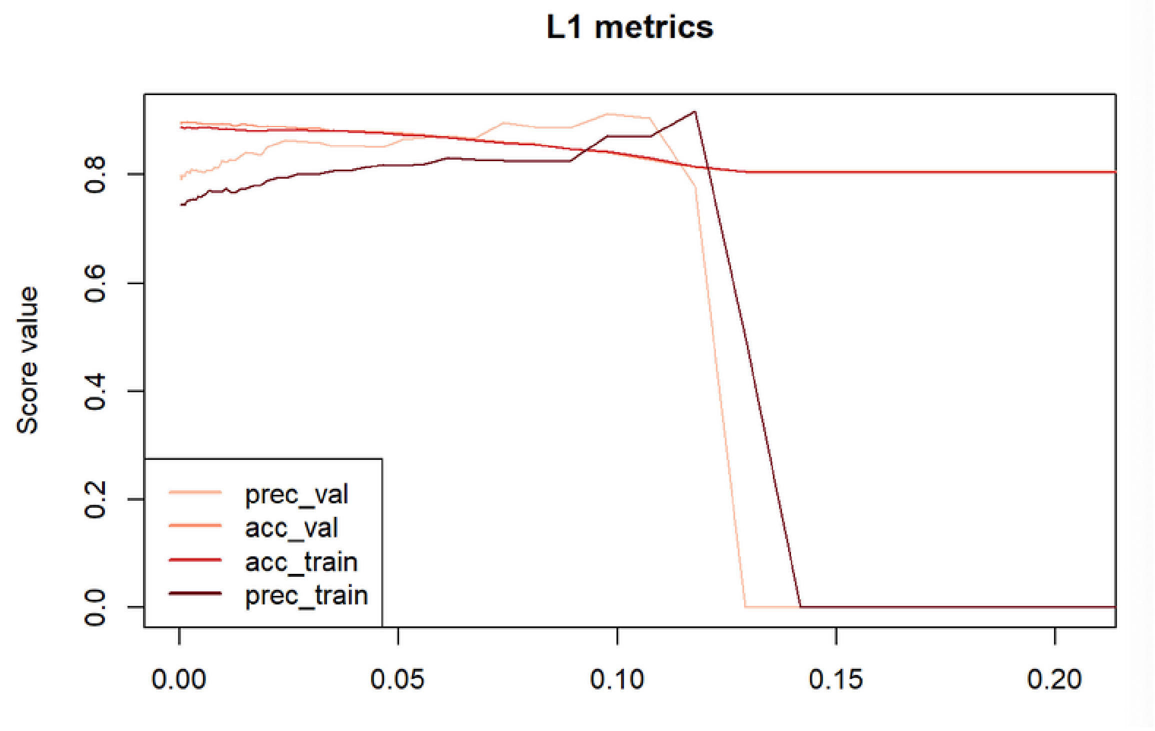


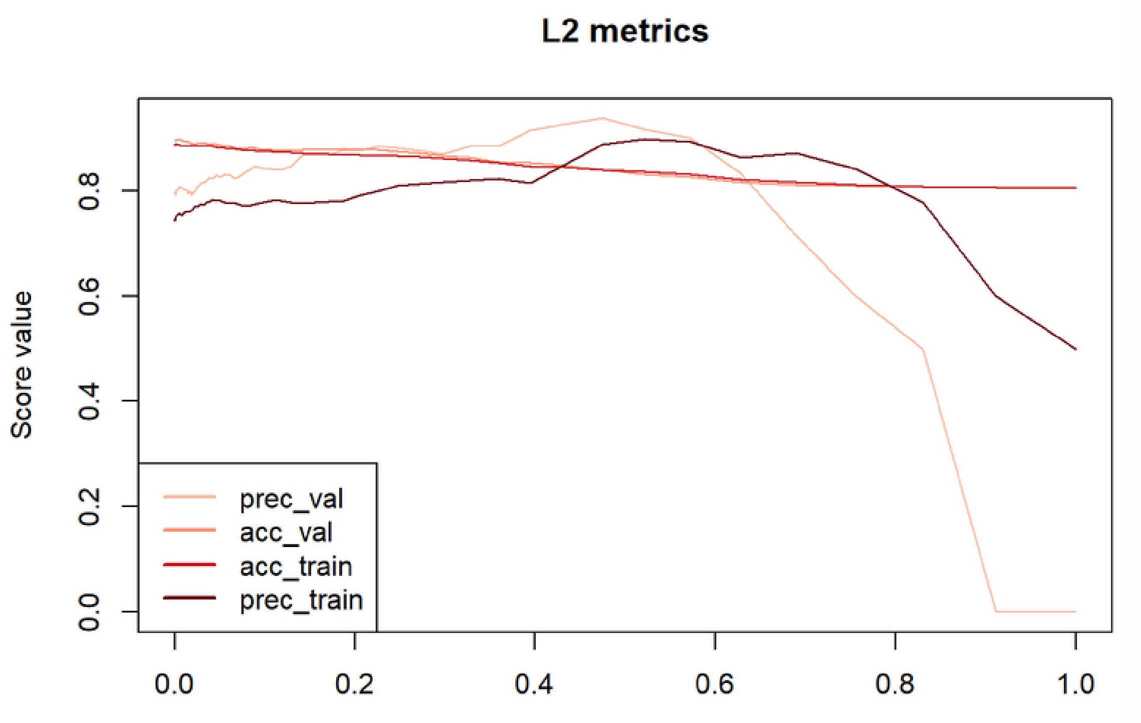
Robust coefficients

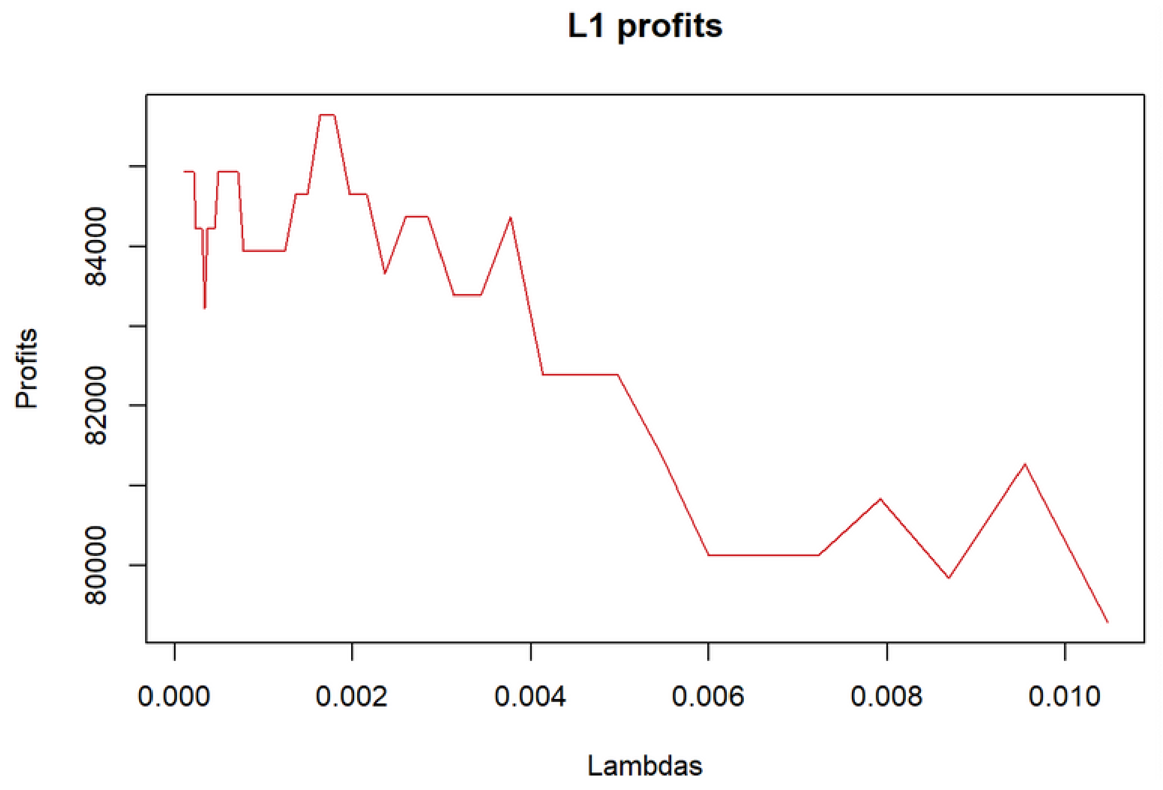
Income - credit_score - loans_0
working_ratio - numcards_6

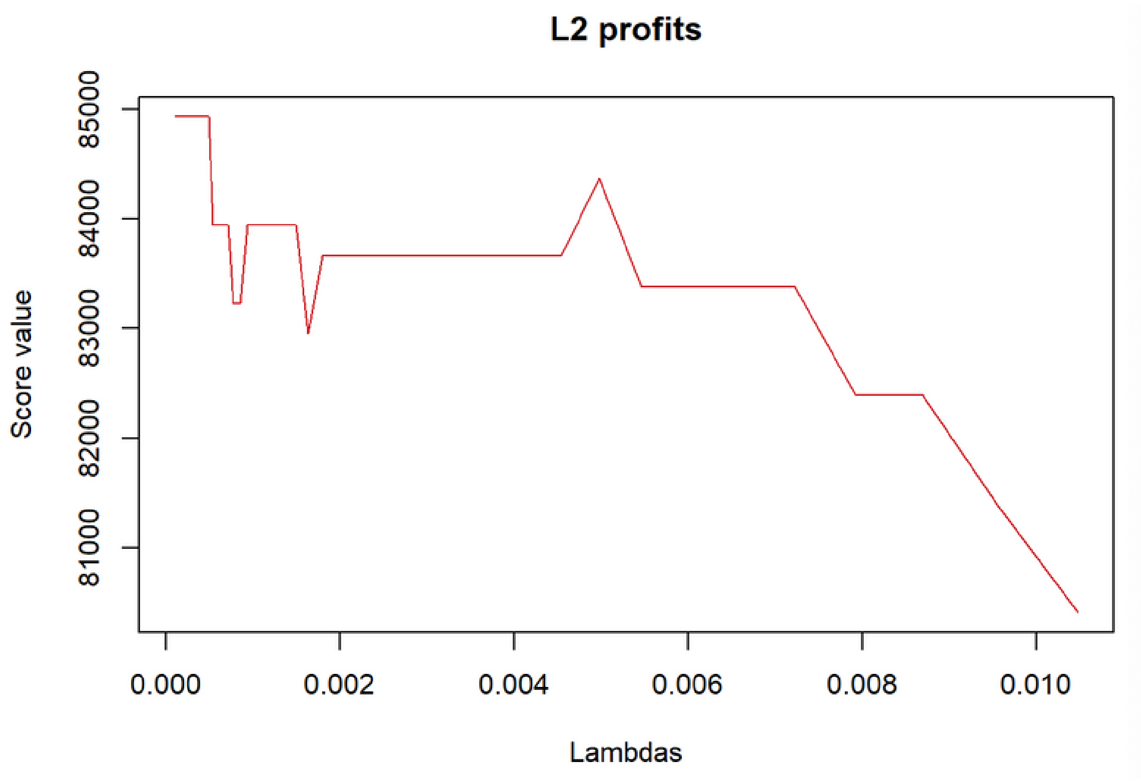
REGULARIZED REGRESSION - 2

The best lambda was chosen according to precision and profit maximization. Graphs are cut only to the relevant area. For both regularization techniques we implemente cross validation to find the best lambda









Max_profits : **85640** Max_precision: **91.18%**
Profits - Max precision : **18110** Lambda: **1.79e-3**

Max_profits : **84930** Max_precision: **93.75%**
Profits - Max precision : **29700** Lambda: **4.86e-4**

UNIVERSITÀ DEGLI STUDI DI PADOVA

6 - MODEL BUILDING

NON-PARAMETRIC ALGORITHM

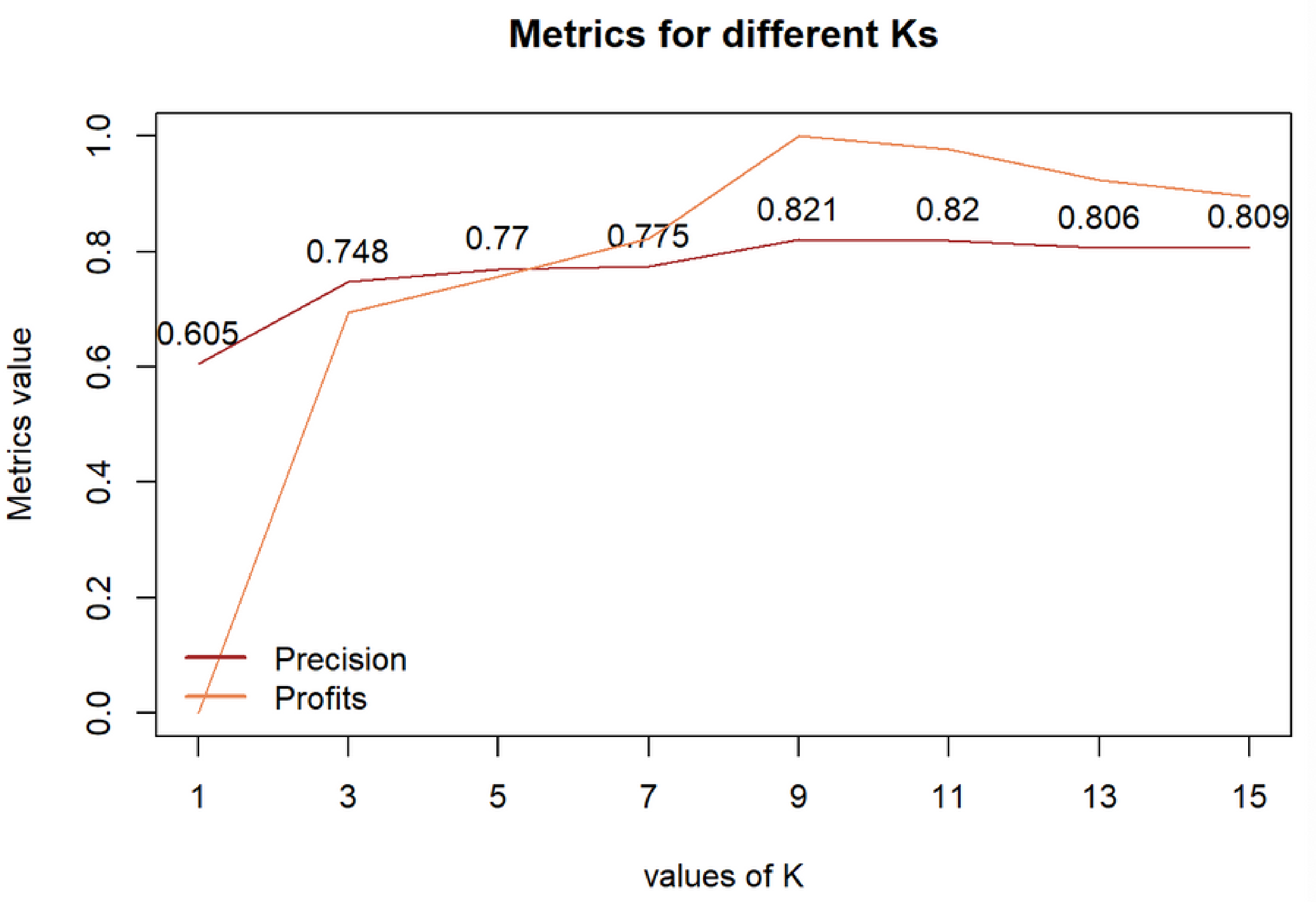
ALGORITHM

Consequently, we decided to perform the classification task using a non-parametric model:

- K - Nearest Neighbors

K-NN is a distance based algorithm, therefore we had to transform our variables:

- categorical variables ⟶ dummy variables
- continuous variables ⟶ scaling

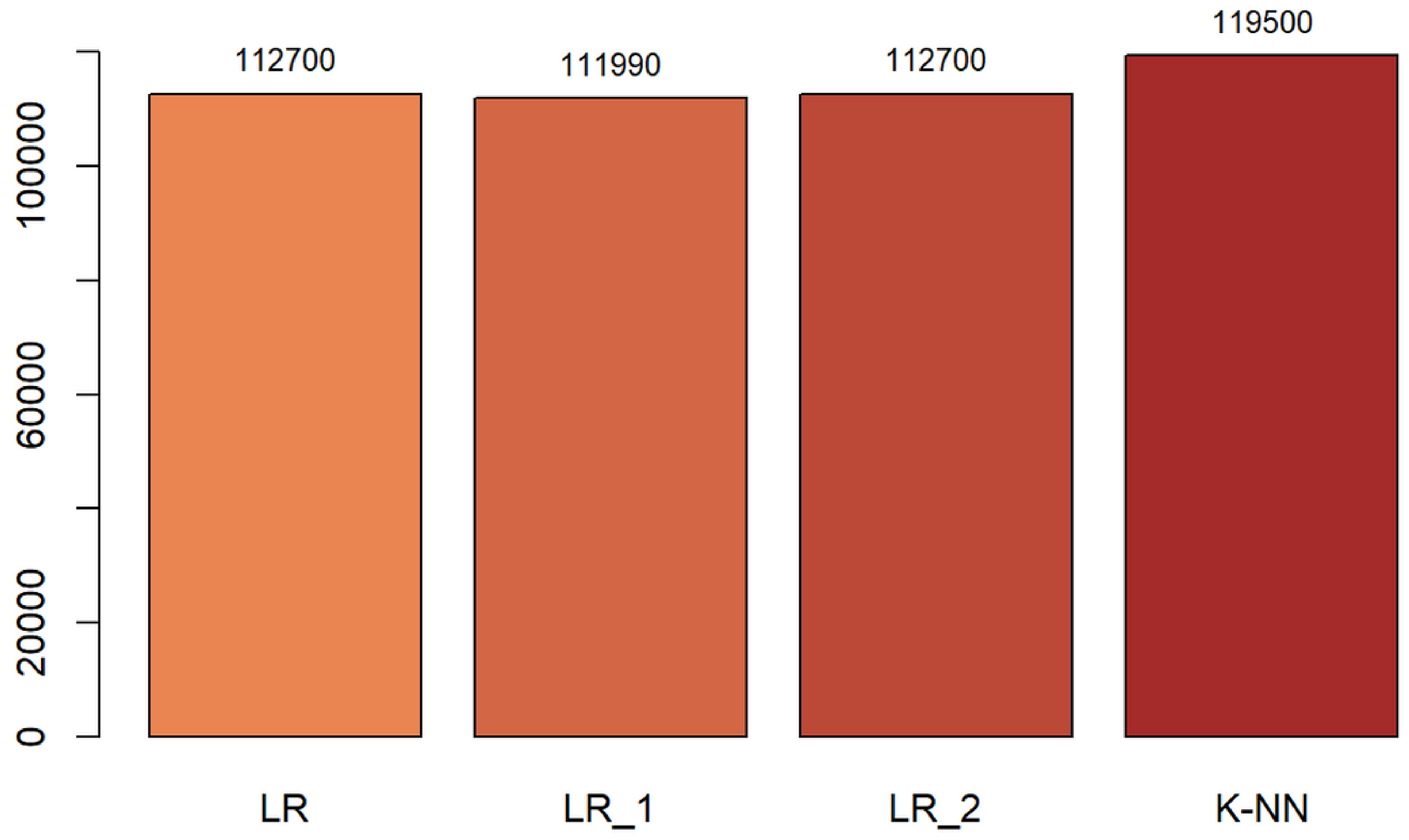


K = 9

6- 2ND MODEL COMPARISON

COMPARISON

Eventually, we compared the logistic regression (normal and regularized) with the KNN just defined. Following the profits related to each algorithm.



The KNN model performs slightly better than the logistic regression.
Therefore, we suggest to take into consideration the aforementioned model as a basis for further study.

7 - FINAL CONSIDERATIONS

FINAL CONSIDERATIONS

- A classificative model for future predictions
- Suggestions to support the decision making process
- Areas for improvements

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

SCATTO GIACOMO 2079421
VOLPATO PIETRO 2079419

Thanks