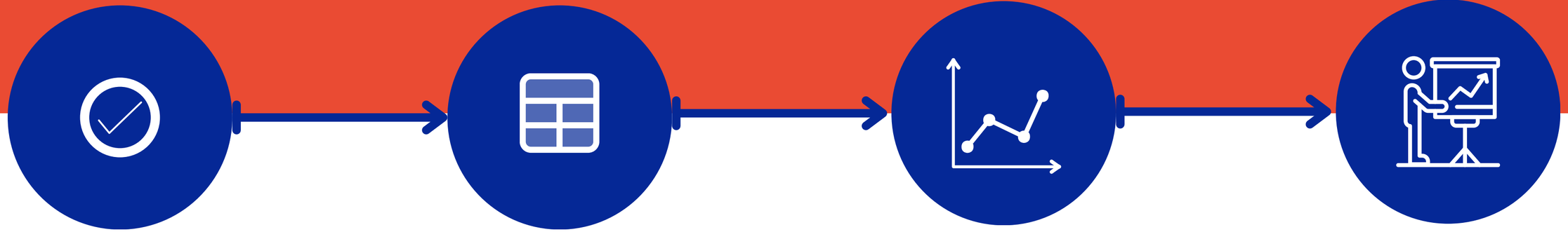
GROUP 9 REPORT

Credit Score Analysis

Giacomo Cirò, Luca Colaci, Costanza D'Ercole,
Alessandro Morosini, Davide Romano, Francesco Vacca

OVERVIEW



GOAL OF THE ANALYSIS AND WORKFLOW

DATASET DESCRIPTION AND PREPARATION

DATA MODELING

MANAGERIAL IMPLICATIONS

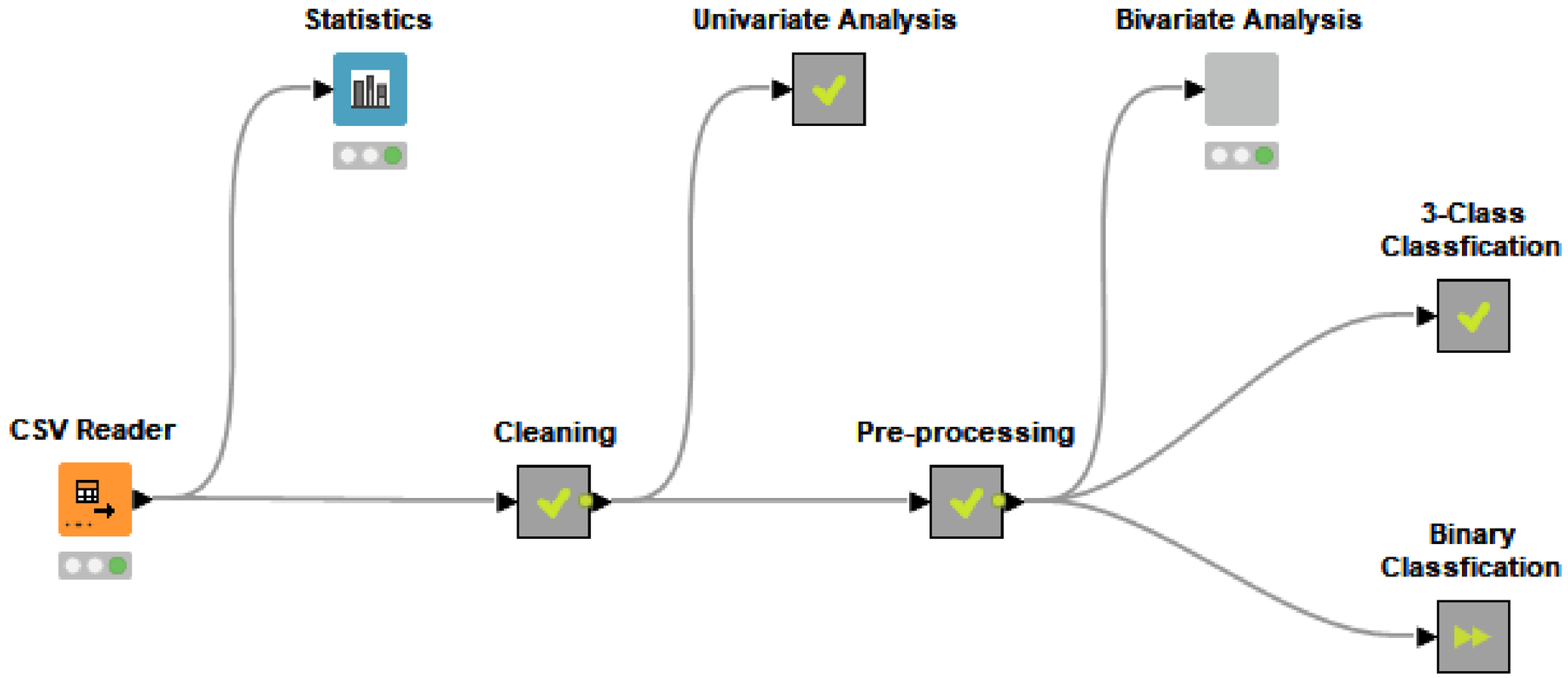
GOAL OF THE ANALYSIS

The primary objective of our analysis is to predict a **customer's credit score** given different attributes.

This score represents the anticipated conduct observed when an individual seeks credit, delineating their propensity to make timely payments and, more broadly, their inclination to honor financial obligations without default.

We first conduct a comprehensive data analysis and visualization to explore the variable at hands. The insights gained from this will serve as the foundational basis for constructing advanced machine learning models. These models' final goal is to offer banks and businesses in general an effective methodology to forecast customer credit risk with precision and efficiency.

WORKFLOW


Statistics
Univariate Analysis
Bivariate Analysis
3-Class
Classfication
CSV Reader
Cleaning
Pre-processing
Binary
Classfication

DATA DESCRIPTION AND PREPARATION

Data Cleaning

Univariate Analysis

Data Pre-processing

Bivariate Analysis

DATASET OVERVIEW

The Dataset

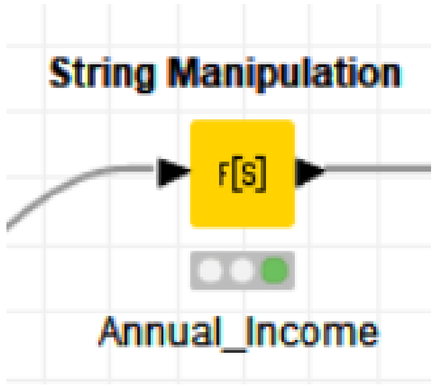
The dataset comprises a compilation of 12,500 entries, each corresponding to a unique customer and encompassing a total of 27 variables.

These measure demographic factors (e.g. age, occupation etc.) and financial attributes (e.g. spending behavior, number of loans etc.) .

DATA DESCRIPTION AND PREPARATION

WRONG DATATYPE

First, we corrected all the data that was saved with a wrong data type, due to minor errors in the record. To do so, we used the node "**String Manipulation**".



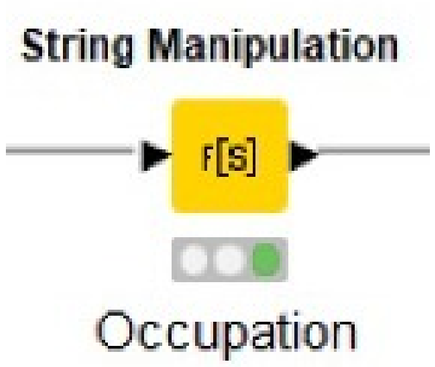
For example, in the variable **Age** some numbers were followed by a underscore, so they were saved as a string.
To solve the issue we removed the underscore and saved correctly the data as an integer value.

S Age
24
35
40_
50
32
54
23
22
23
26
29
36
28
52
20
41
42
35_
19

PLACEHOLDER TO NULL

Data Cleaning

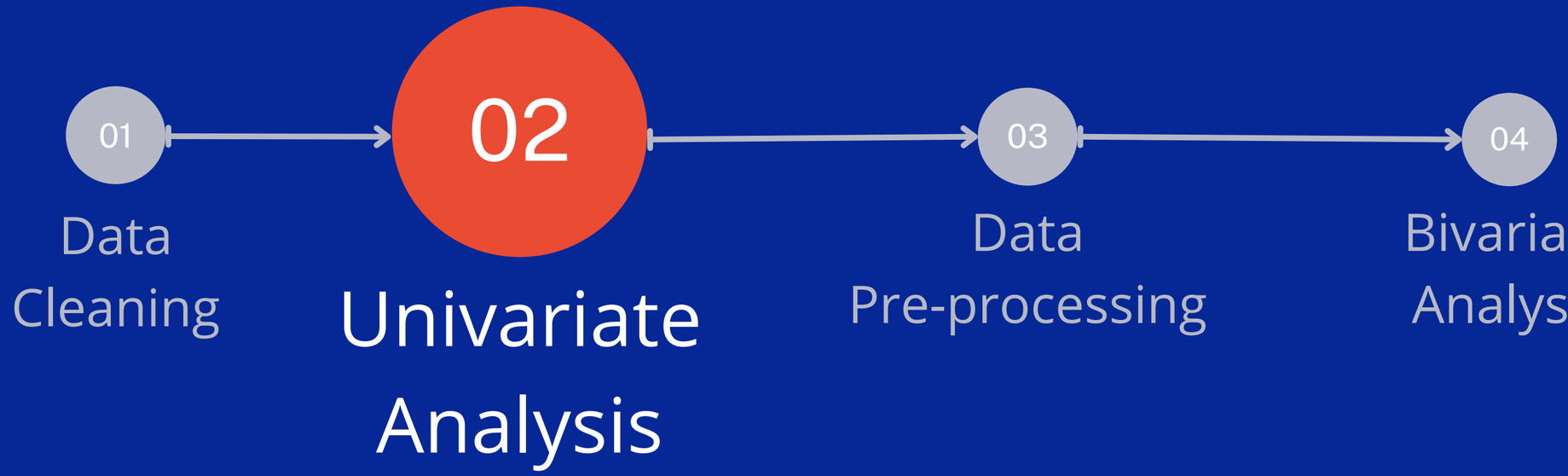
In other cases, we decided to substitute placeholders for unknown values with null values, to deal with them later.



Take as an example the variable **Occupation**, where some values were registered as "_____"; those were removed and substituted with nulls.

S Occupa...
Scientist
_____
Journalist
Accountant
Teacher
Musician
Scientist
Musician
Developer
Lawyer
Media_Mana...
_____
Media_Mana...
Entrepreneur
Writer
Writer
Lawyer
Mechanic
Teacher

DATA DESCRIPTION AND PREPARATION



Age

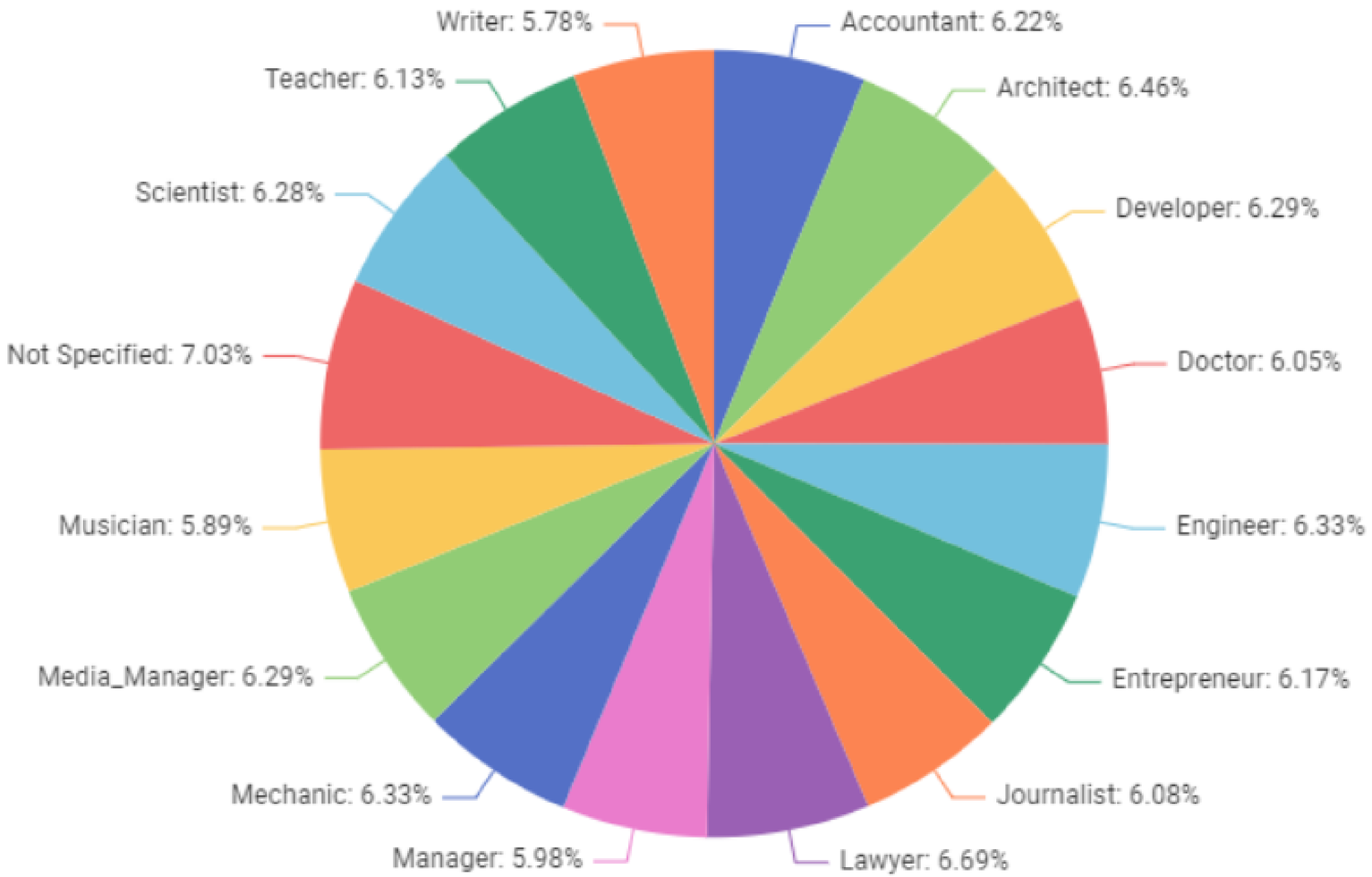
Univariate Analysis

- **Nature:** Numerical, continuous
- **Description:** Customer‘s age
- **Range:** -500 to 8698
- **Insights:** The graph shows absurd values (negative age, or age above 8000 years), which are clearly record errors. We will handle these values in the following steps.

Occupation

Univariate Analysis

- **Nature:** Categorical, nominal
- **Description:** Customer‘s job
- **#Categories:** 16
- **Insights:** Customers are almost equally distributed among the different jobs, with 7.03% not specified (nulls) records.



Annual Income

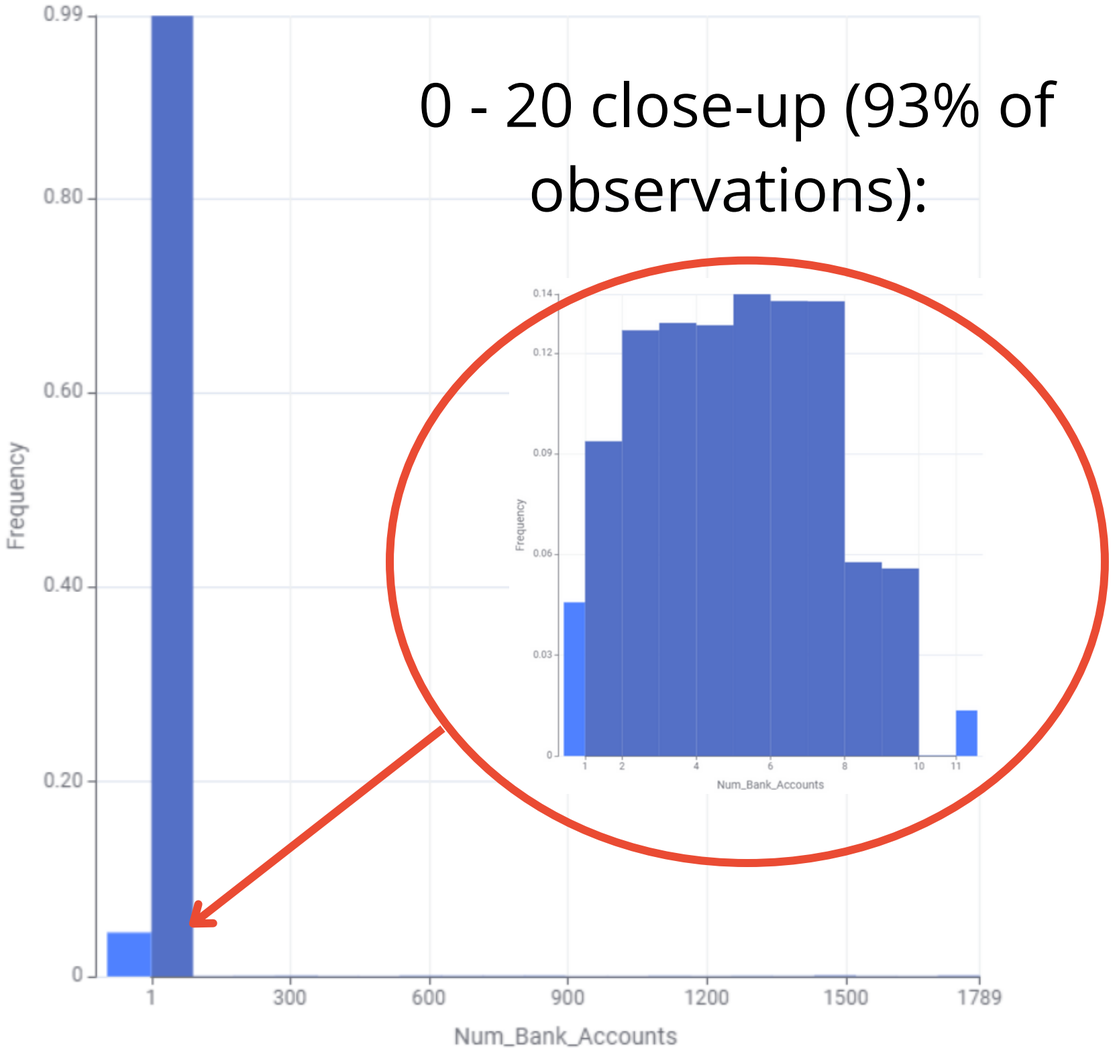
Univariate Analysis

- **Nature:** Numerical, continuous
- **Description:** Customer‘s annual income
- **Range:** $7k to $24.000k
- **Insights:** Customers’ incomes are concentrated around the median value ($37,572), with most of the incomes between 0 and $100.000, and few upper outliers.

Num_bank_accounts

Univariate Analysis

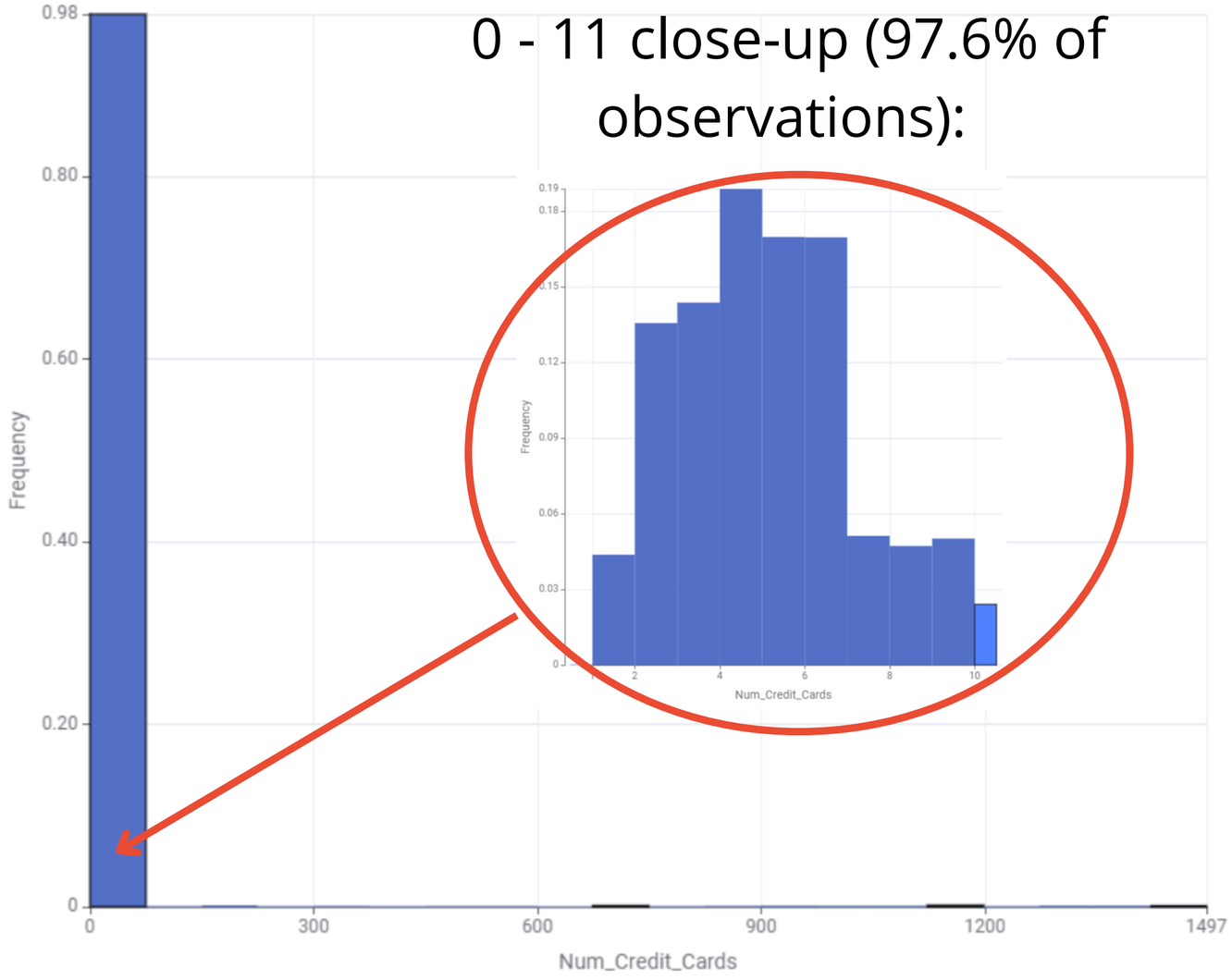
- **Nature:** Numerical, discrete
- **Description:** Number of bank accounts for each customer
- **Range:** 0 to 1789
- **Insights:** The majority of customers have between 0 and 20 bank accounts, with a mean value of 6. However, there are a few outliers on both ends.



Num_Credit_Card

Univariate Analysis

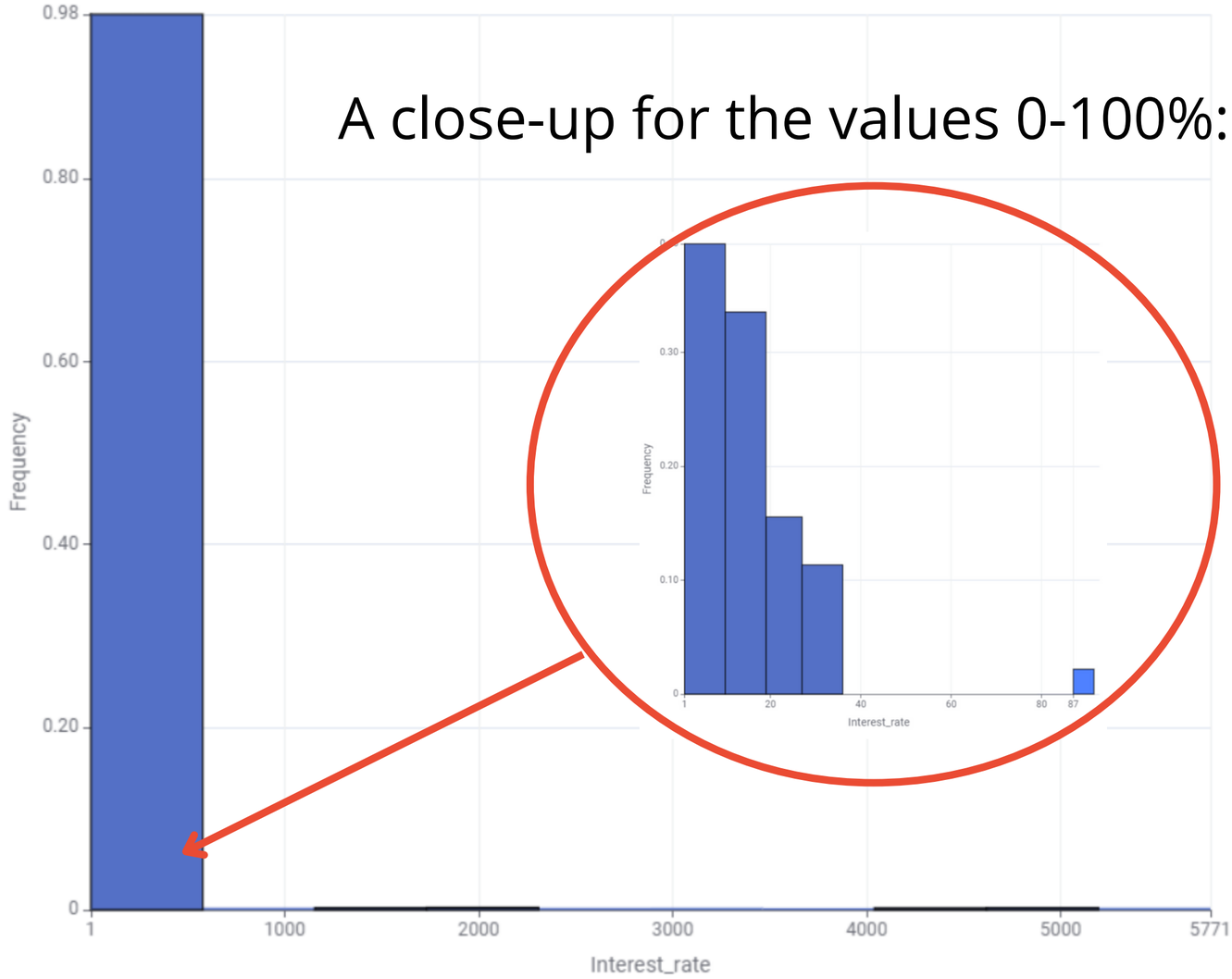
- **Nature:** Numerical, discrete
- **Description:** Number of credit cards held by the customers
- **Range:** 0 to 1497
- **Insights:** Like for other variables, most data is concentrated in lower values (the median is 5). Still, there are a few outliers, with the maximum value being 1497.



Interest_Rate

Univariate Analysis

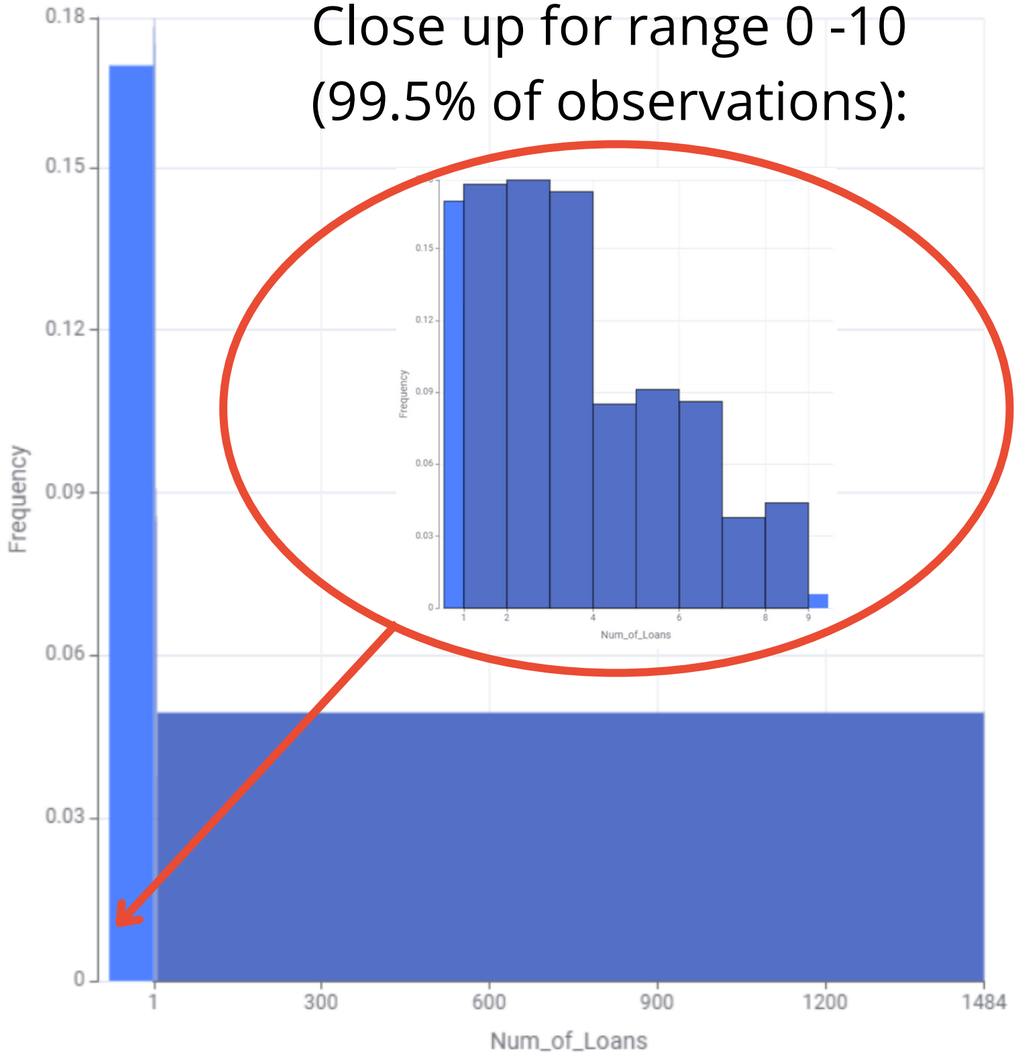
- **Nature:** Numerical, continuous
- **Description:** Interest rate on credit card
- **Range:** 0% to 5771%
- **Insights:** The mean value is 13%, with a few upper outliers over 5000%, probably because of some conversion and/or inputting issues during the data collection process.



Num_of_loan

Univariate Analysis

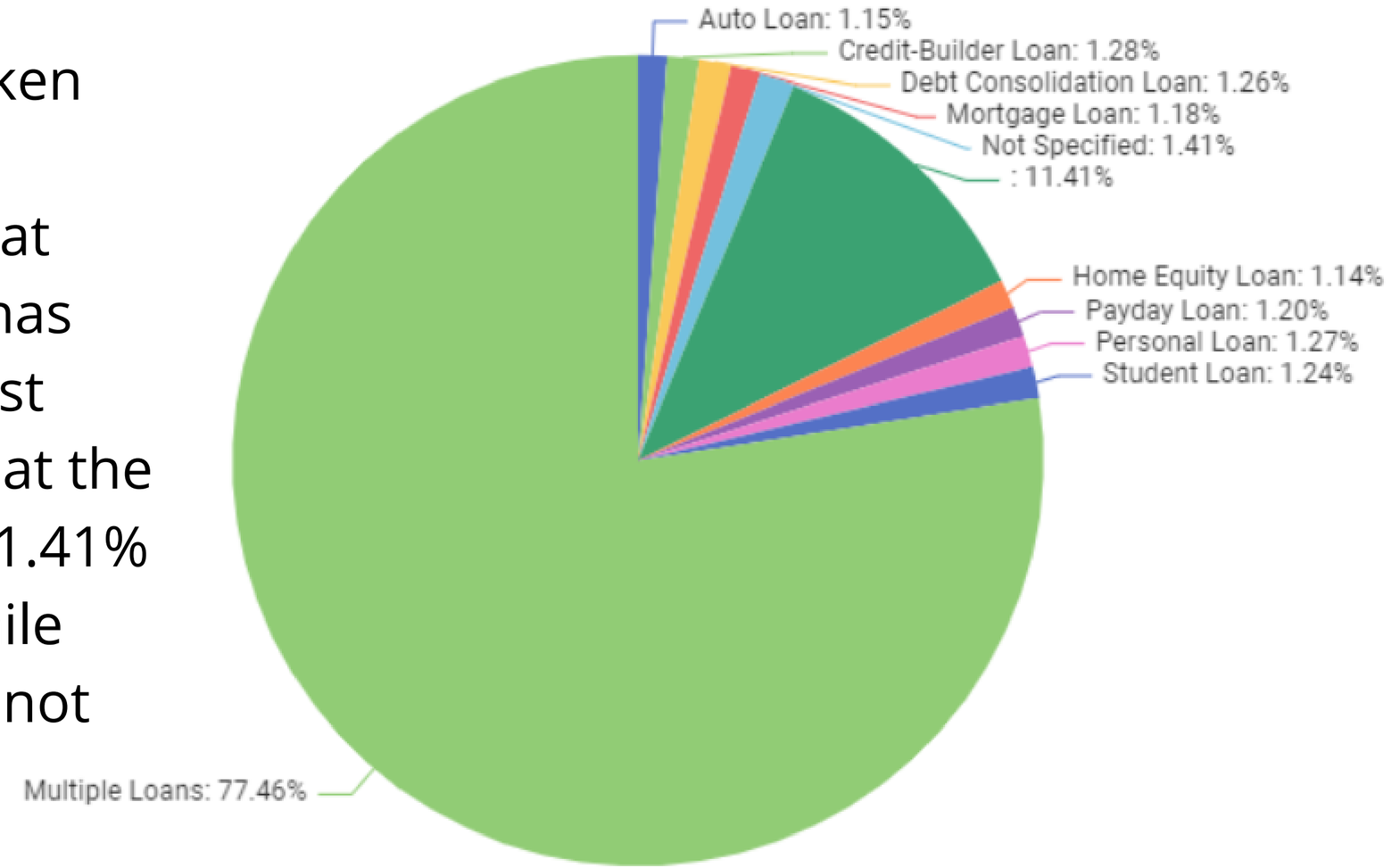
- **Nature:** Numerical, discrete
- **Description:** Number of loans taken by the customer
- **Range:** 0 to 1484
- **Insights:** The average value is 3, but we can find upper outliers here as well.



Type_of_Loan

Univariate Analysis

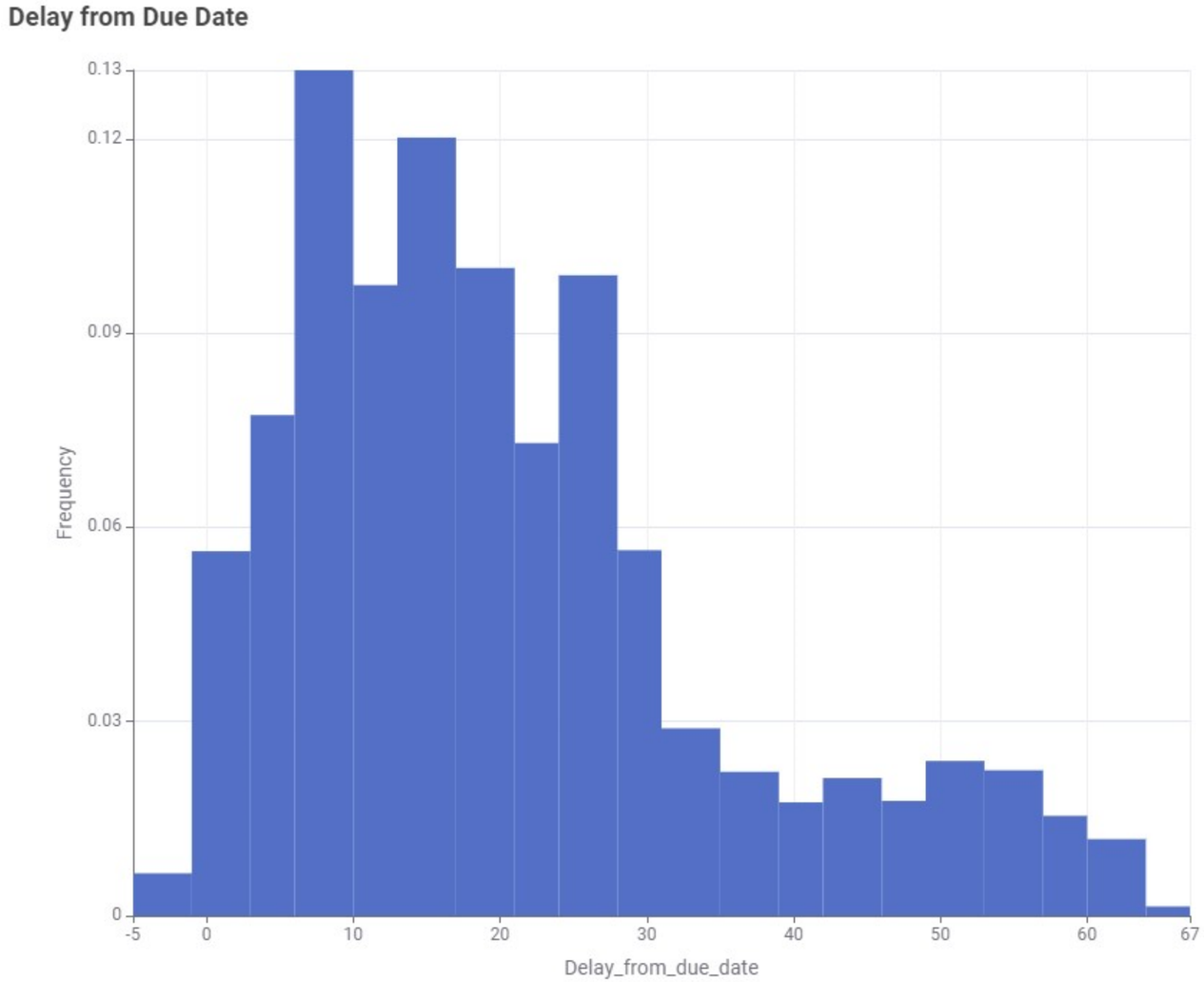
- **Nature:** Categorical, nominal
- **Description:** Type of loan taken by the customer
- **Insights:** The graph shows that just a minority of customers has one type of loan, while the vast majority takes multiple loans at the same time. Notice also that 11.41% does not have loans at all, while ~1.5% of the customers have not specified ones.



Delay_from_due_date

Univariate Analysis

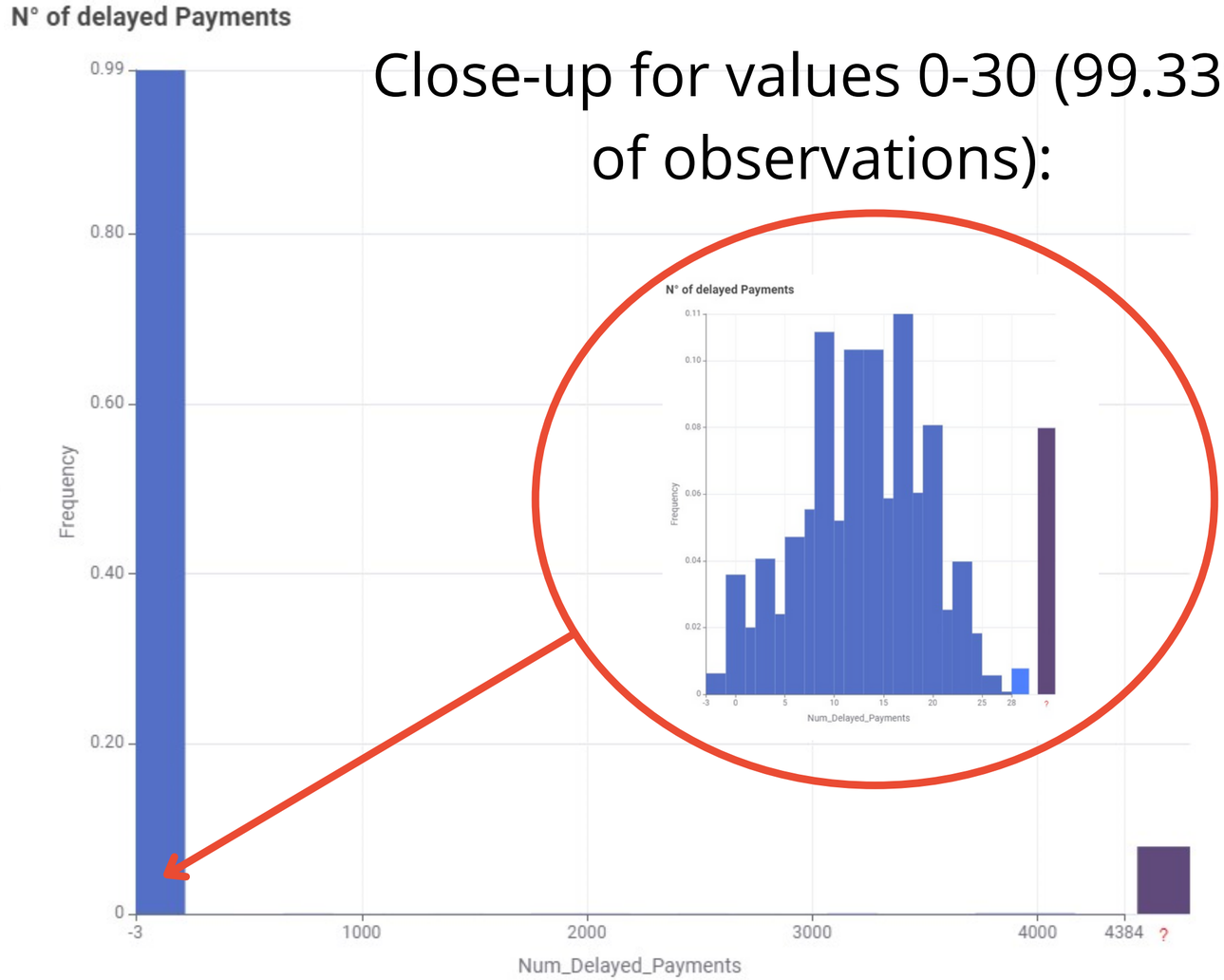
- **Nature:** Numerical, discrete
- **Description:** Avg. number of days delayed from the payment date
- **Range:** -5 to 67
- **Insights:** This variable does not show particular outliers, however there are negative values, that could be interpreted as customers usually paying their installments before due date.



Num_of_delayed_payments

Univariate Analysis

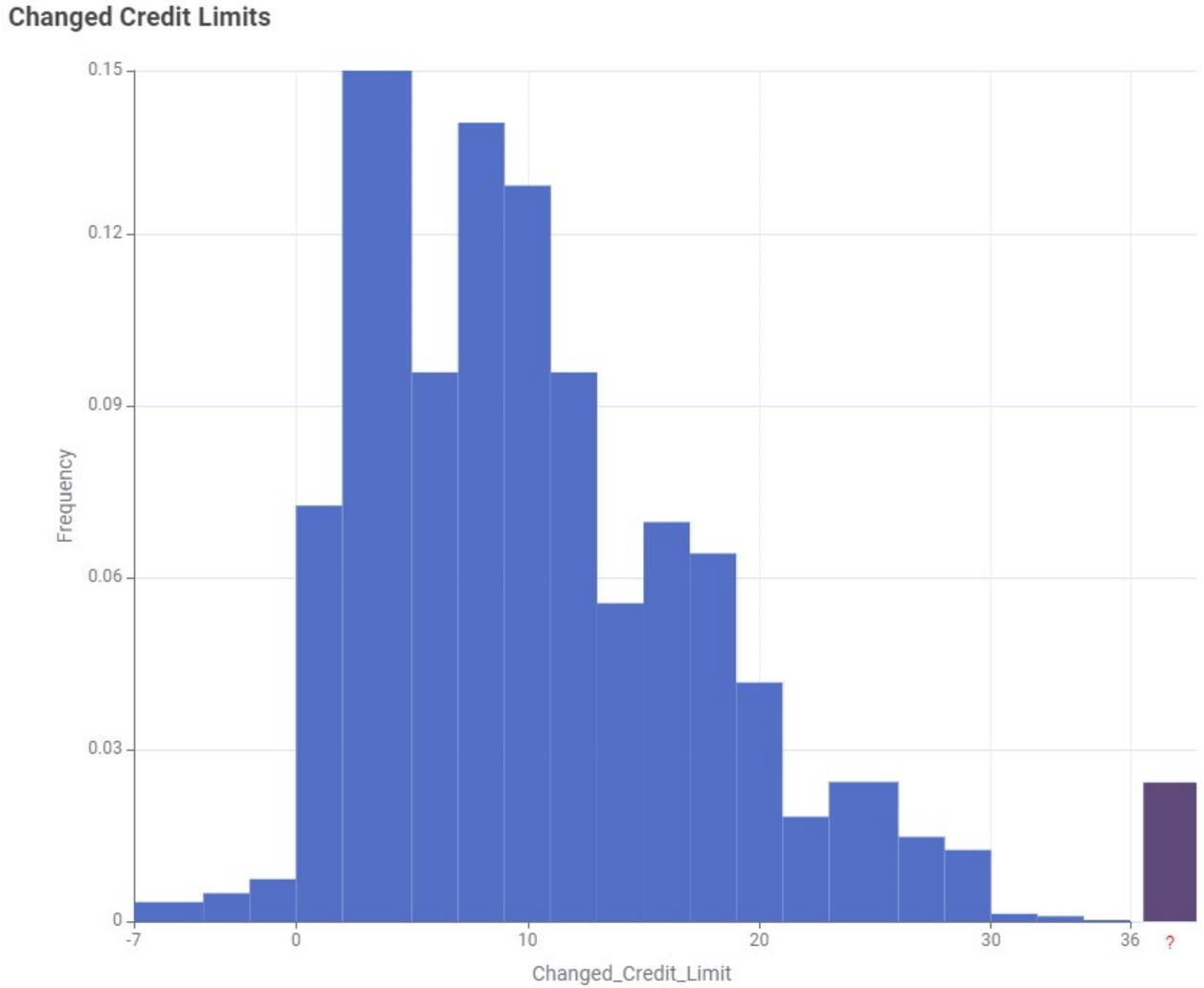
- **Nature:** Numerical, discrete
- **Description:** Average number of payments delayed by a customer
- **Range:** -3 to 4384
- **Insights:** Most customers miss some payments, however there are some negative values, interpreted as paying before due date. 8% of the values is missing, we'll manage them in the pre-processing phase.



Changed_Credit_Limit

Univariate Analysis

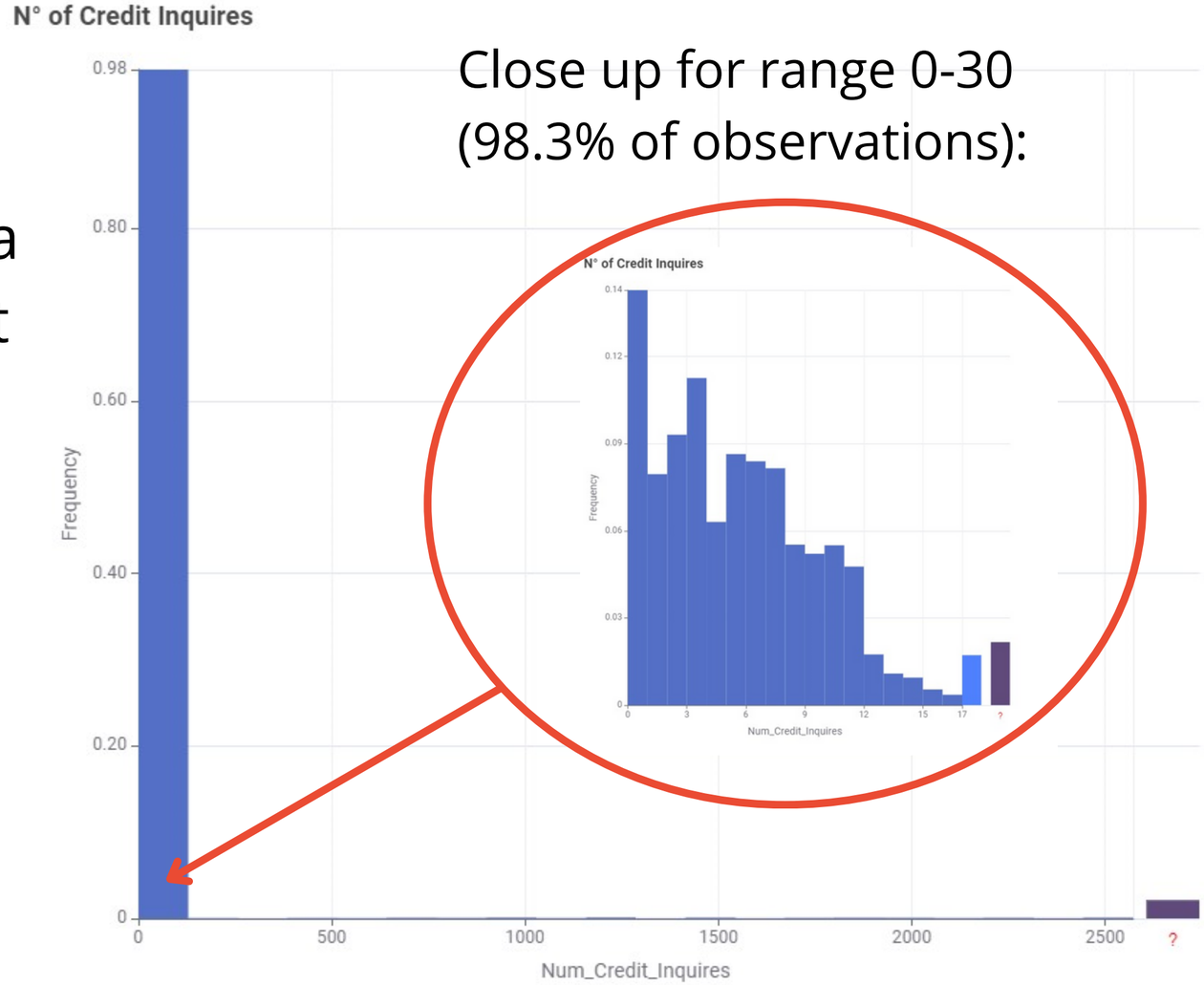
- **Nature:** Numerical, discrete
- **Description:** Percentage change in credit card limit
- **Range:** -7% to 36%
- **Insights:** Most of the values are positive. We interpreted the negative values as customers that decided to lower the credit card limit. There are 2.4 % of missing values.



Num_Credit_Inquiries

Univariate Analysis

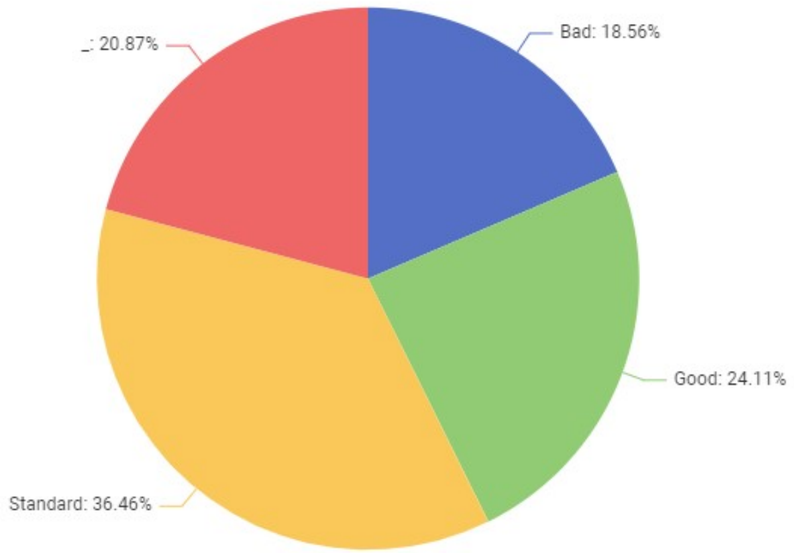
- **Nature:** Numerical, discrete
- **Description:** Number of credit inquiries, i.e., the number of times a financial institution asked the credit bureau for a check on a customer's financial situation to approve them for credit requests.
- **Range:** 0 to 2500
- **Insights:** Most values are in the range 0-30. There are 2.2% of missing values.



Credit_Mix

Univariate Analysis

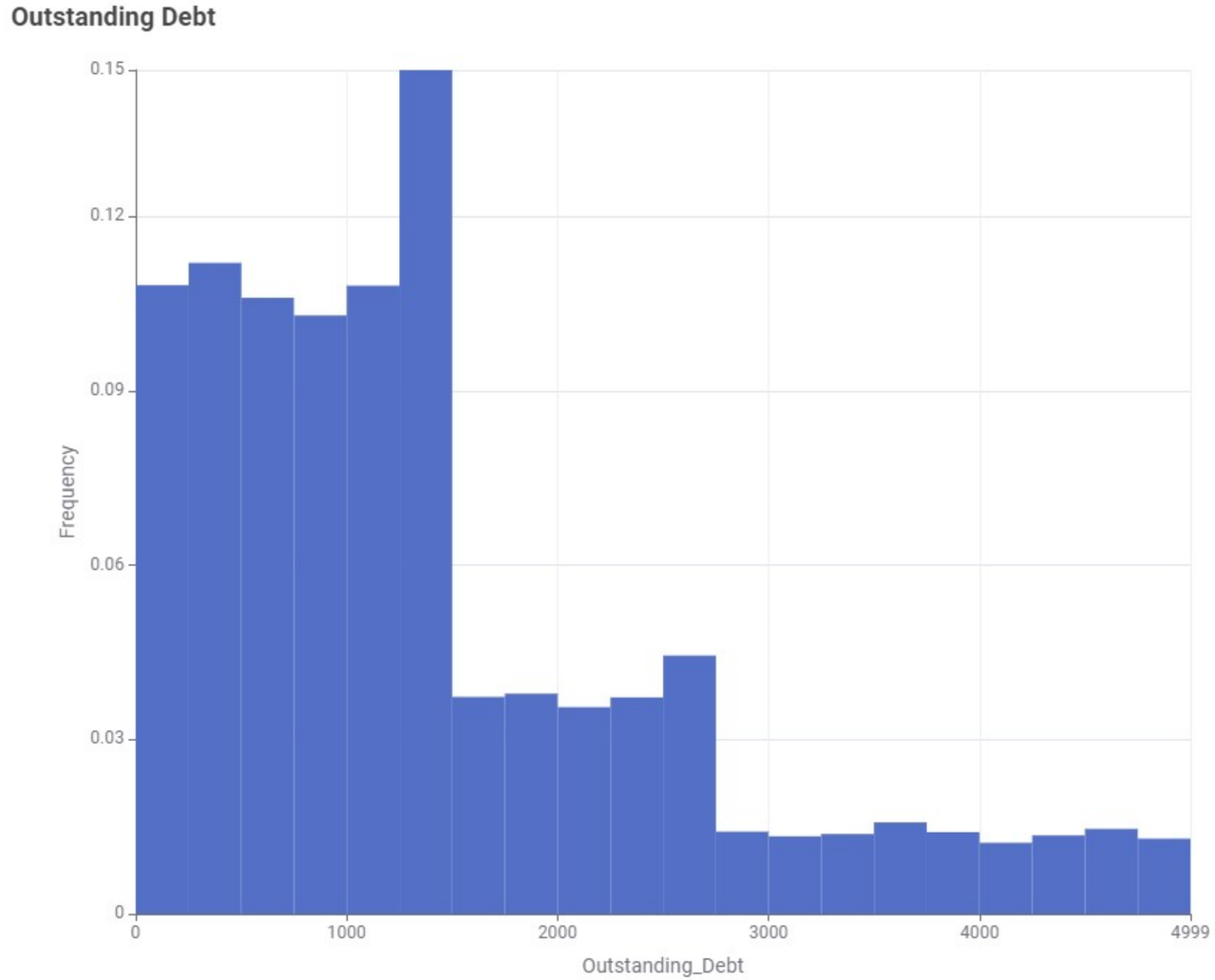
- **Nature:** Categorical, nominal
- **Description:** Classification of the mix of credit held by the customer
- **Categories:** [‘Standard‘,‘Good‘,‘Bad‘,‘_‘]
- **Insights:** We don’t know how the credit mix was evaluated by the financial institution in the first place, but we can notice that it splits quite evenly between the three categories. However, there are 20.87% of missing values, indicated by the placeholder ‘_‘.



Outstanding_Debt

Univariate Analysis

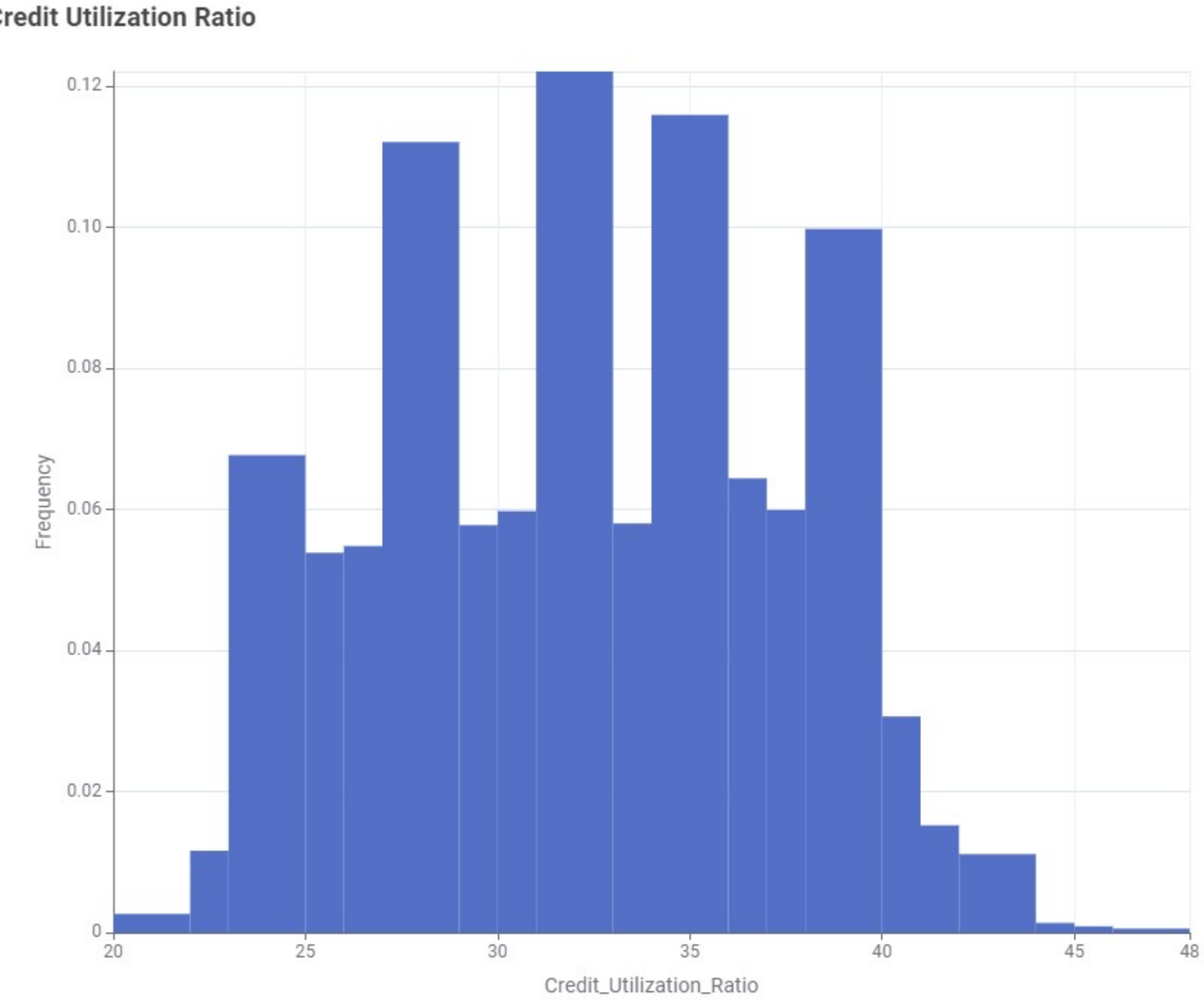
- **Nature:** Numerical, continuous
- **Description:** The part of debt remained to be paid
- **Range:** 0 to 4999
- **Insights:** This variable does not show missing values or outliers in its distribution, and we can see that the customers reported in the dataset are mostly done with their debit payments.



Credit_Utilization_Ratio

Univariate Analysis

- **Nature:** Numerical, continuous
- **Description:** Utilization ratio of credit card, showing the average percentage of utilization of the credit card maximum amount by the customer.
- **Range:** 0 to 4999
- **Insights:** This is one of the most fundamental aspects bank consider for canonical credit score evaluation.



Credit_History_Age

Univariate Analysis

- **Nature:** Categorical, nominal
- **Description:** Age of the accounts that appear in each customer's credit reports. It indicates the number of years and months.
- **Insights:** We will manage this variable in the pre-processing section, as it was encoded in string values, really hard to work on given the type of information. We'll therefore process them in order to cast them in a more appropriate data type.

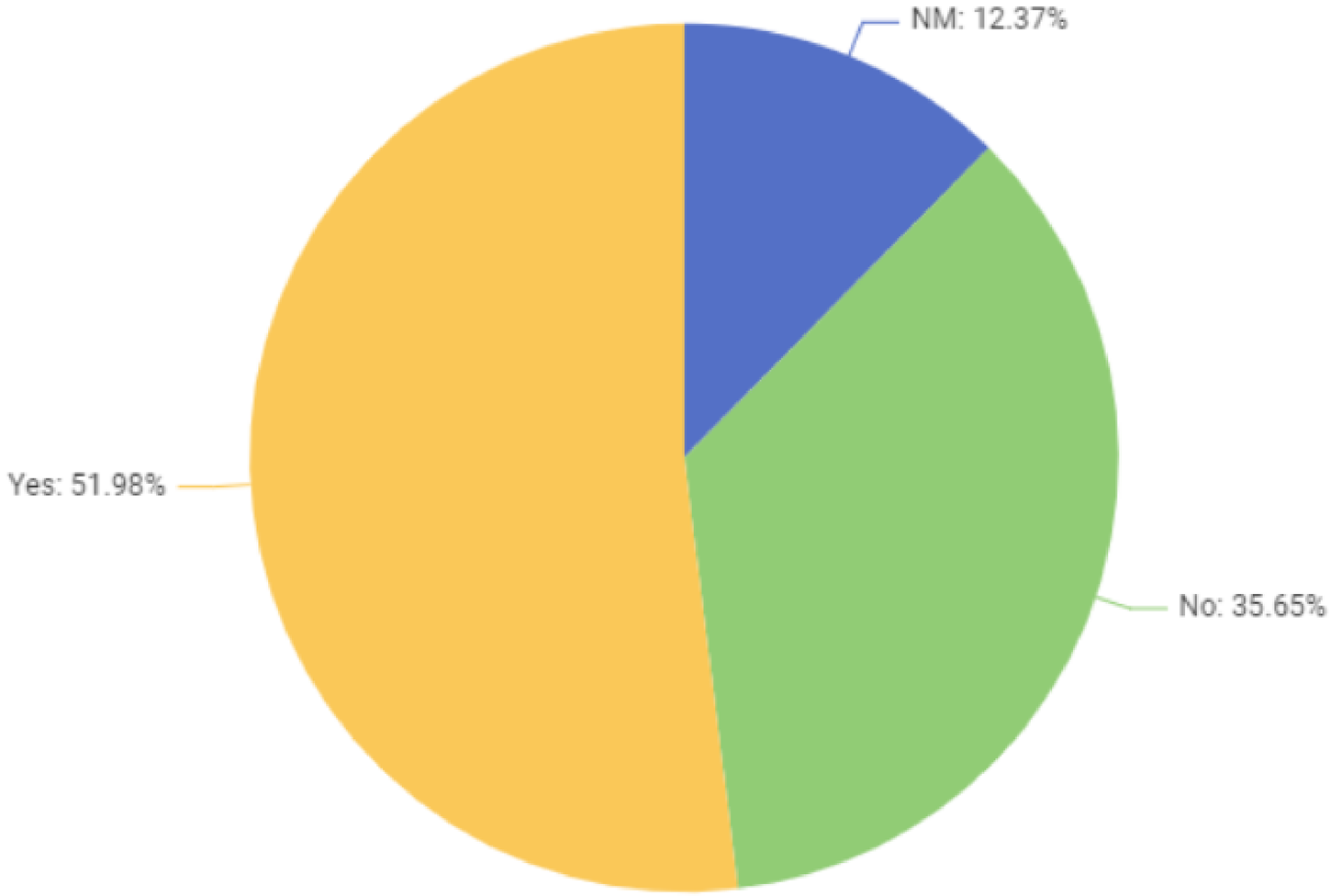
Example of some records:

22 Years and 10 Months
NA
23 Years and 0 Months
27 Years and 3 Months
27 Years and 4 Months

Payment of Min Amount

Univariate Analysis

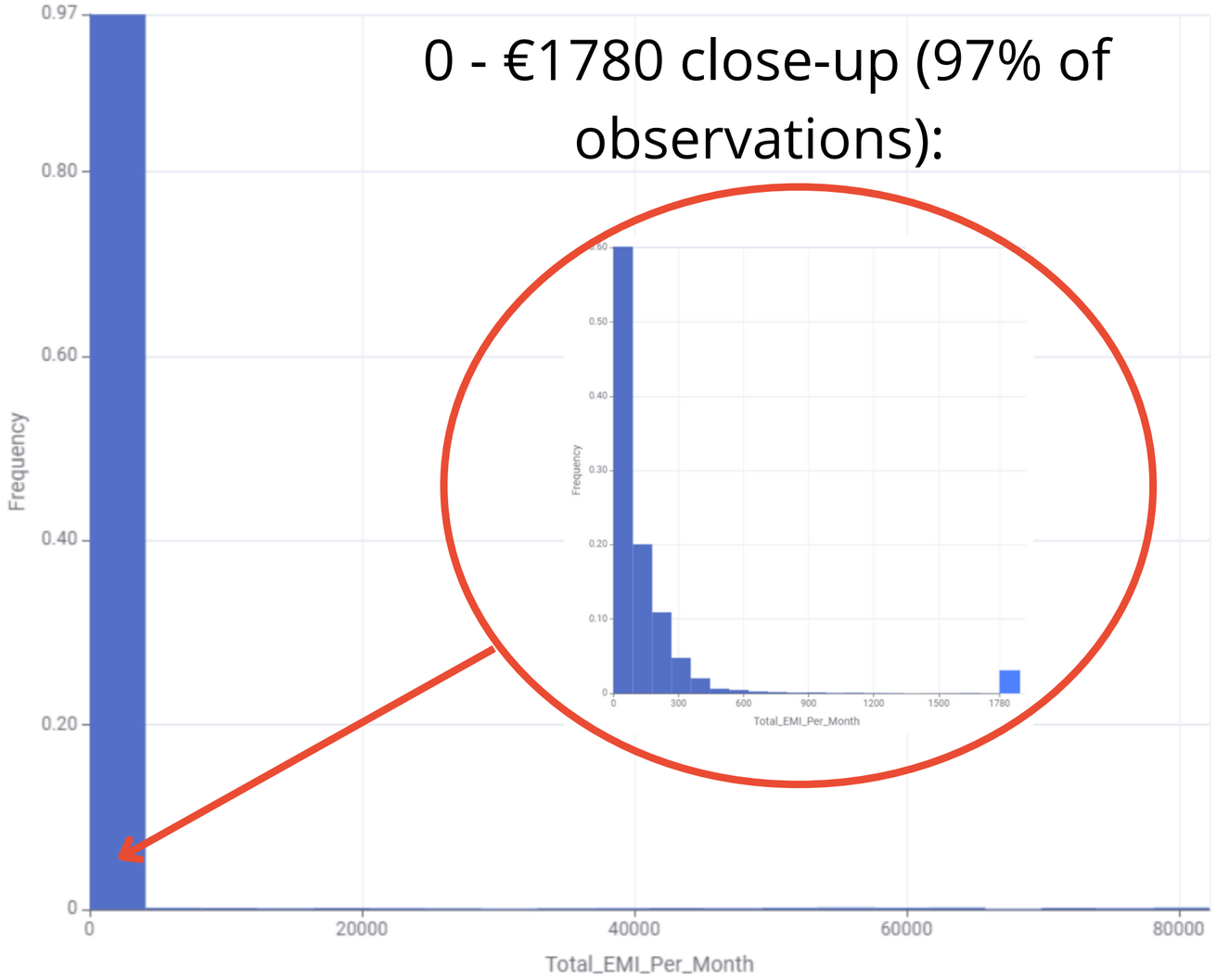
- **Nature:** Categorical, nominal
- **Description:** Indicates whether only the minimum amount was paid by the person.
- **Categories:** [‘Yes‘,‘No‘,‘NM‘]
- **Insights:** Most values are ‘Yes‘.‘NM‘ indicates unknown information. This variable will be better addressed during data pre-processing.



Total_EMI

Univariate Analysis

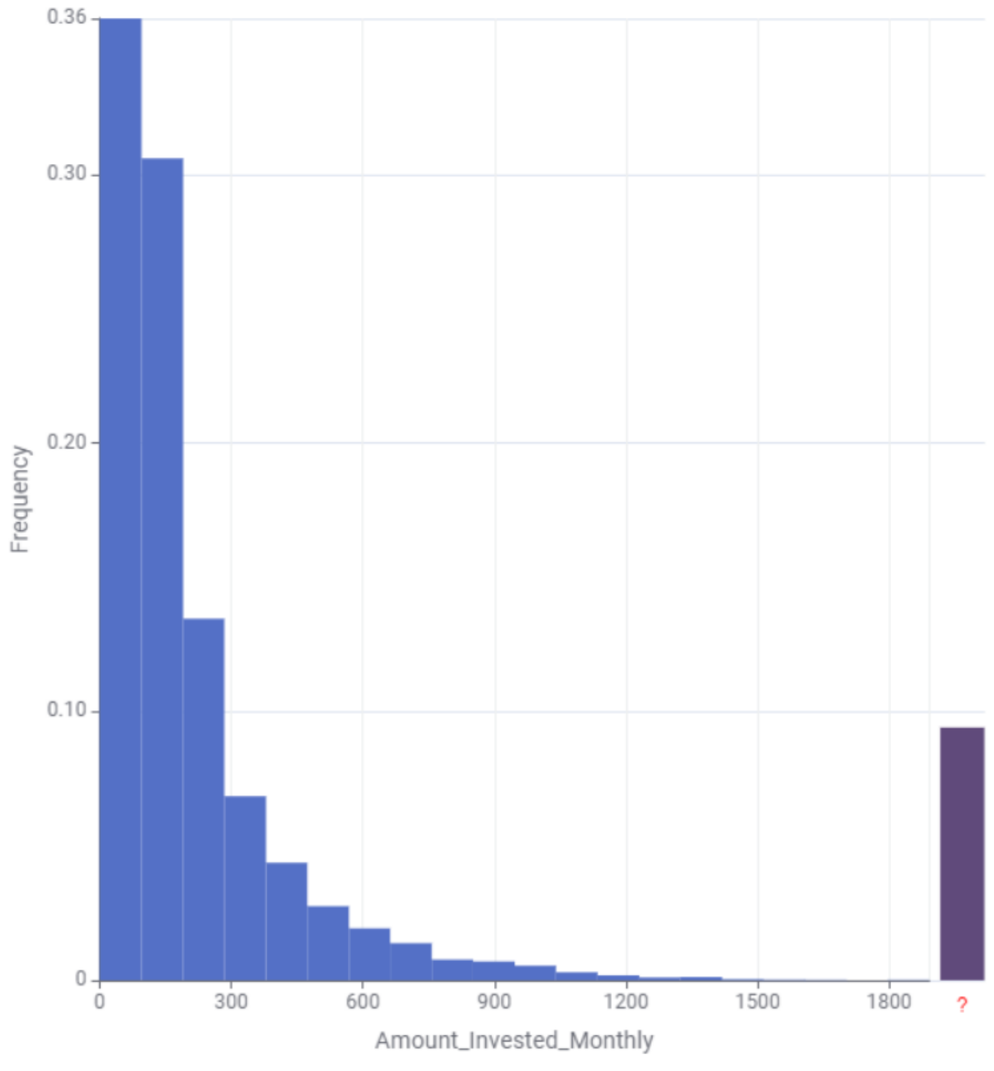
- **Nature:** Numerical, continuous
- **Description:** It indicates the monthly EMI (Equated Monthly Installment) payments
- **Range:** 0 to 80k
- **Insights:** The mean value is 1,397, but there is a number of outliers on the far right side, with customers that would theoretically have to pay up to $80k a month.



Amount_Invested

Univariate Analysis

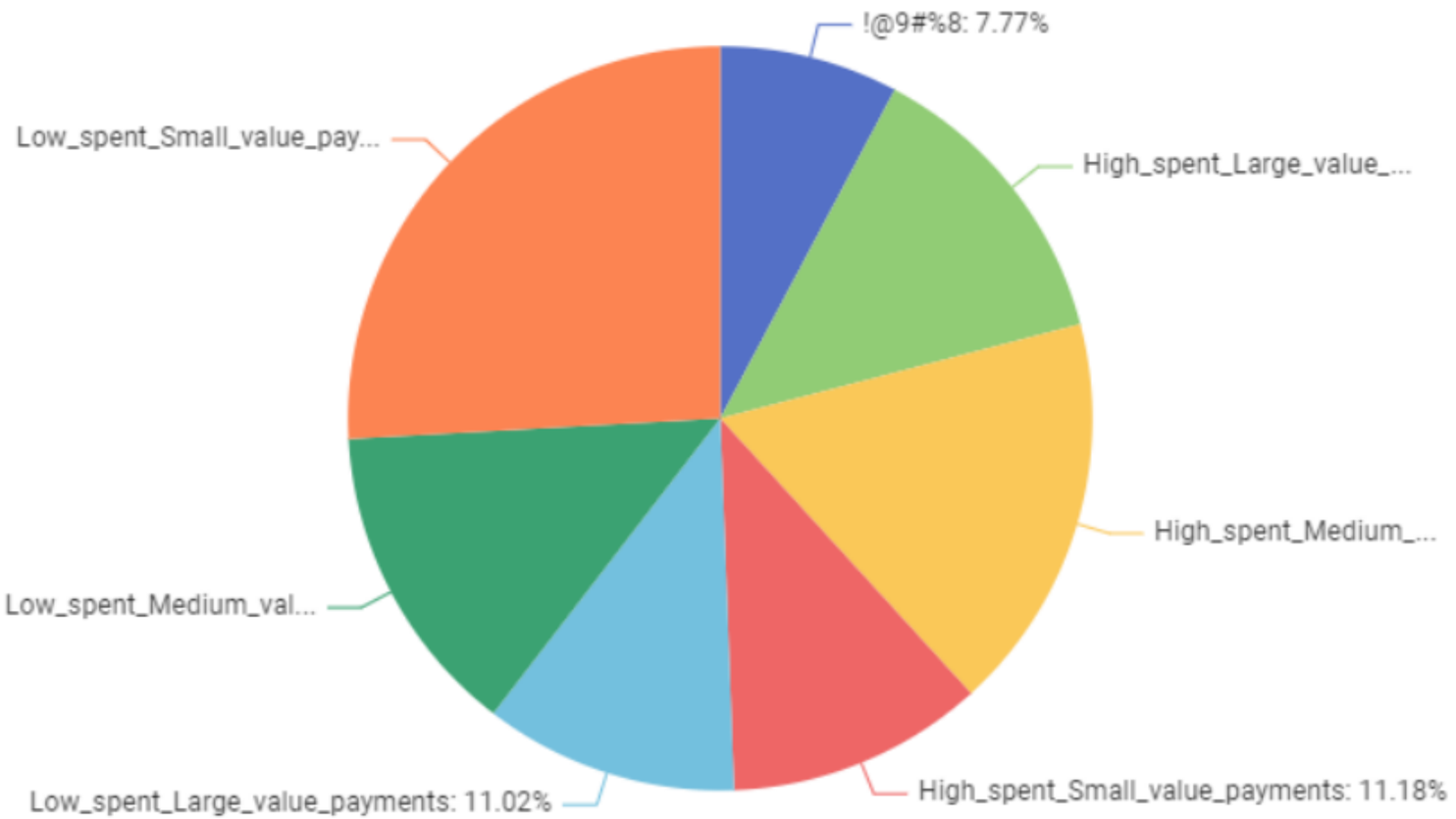
- **Nature:** Numerical, continuous
- **Description:** Monthly amount invested by the customer
- **Range:** 0 to 2000
- **Insights:** The mean value is €198, and no particular outlier was found during our preliminary data analysis. There are 9% of missing values, which will be handled in later stages of our analysis.



Payment_Behaviour

Univariate Analysis

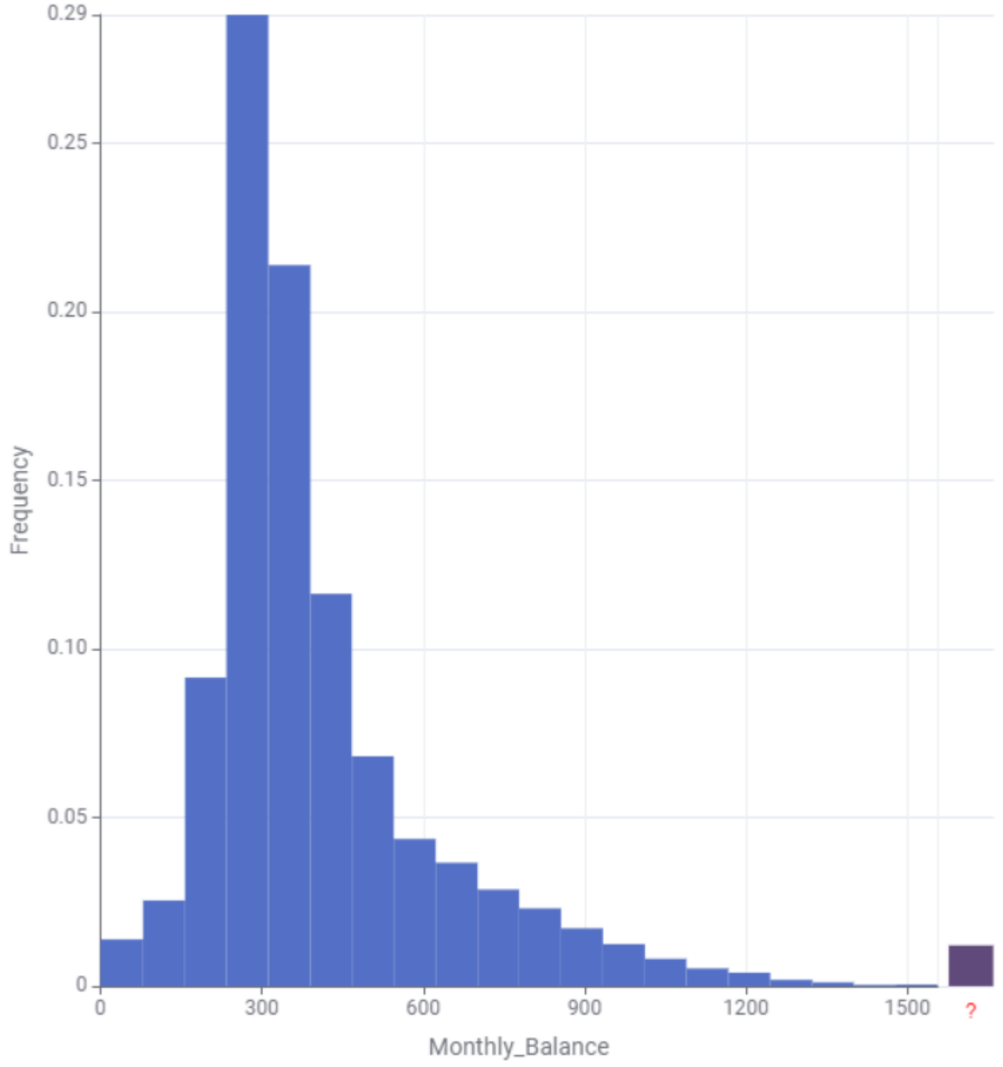
- **Nature:** Categorical, nominal
- **Description:** Payment behaviour of the customer
- **# Categories:** 7
- **Insights:** Describes how a customer usually behaves in its spendings by chaining both the overall amount spent (either Low or High) and the average amount of transactions (Small, Medium or Large).



Balance

Univariate Analysis

- **Nature:** Numerical, continuous
- **Description:** balance of the customer on the date of record.
- **Range:** 0 to 1500
- **Insights:** There is a small percentage of missing values, probably due to errors in the data collection process. Other than that, data seems to be nicely distributed, with no particular outliers and an average balance of 300$.

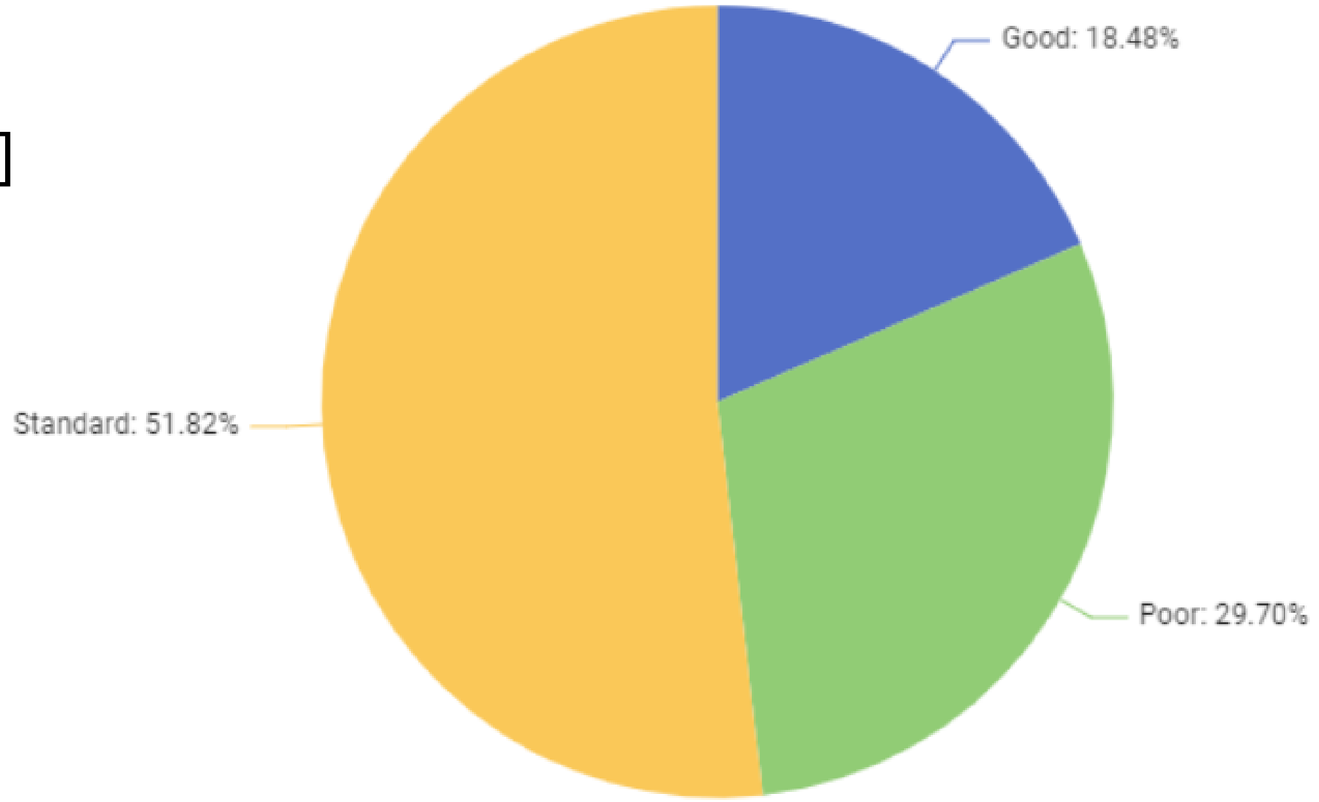


Credit_Score

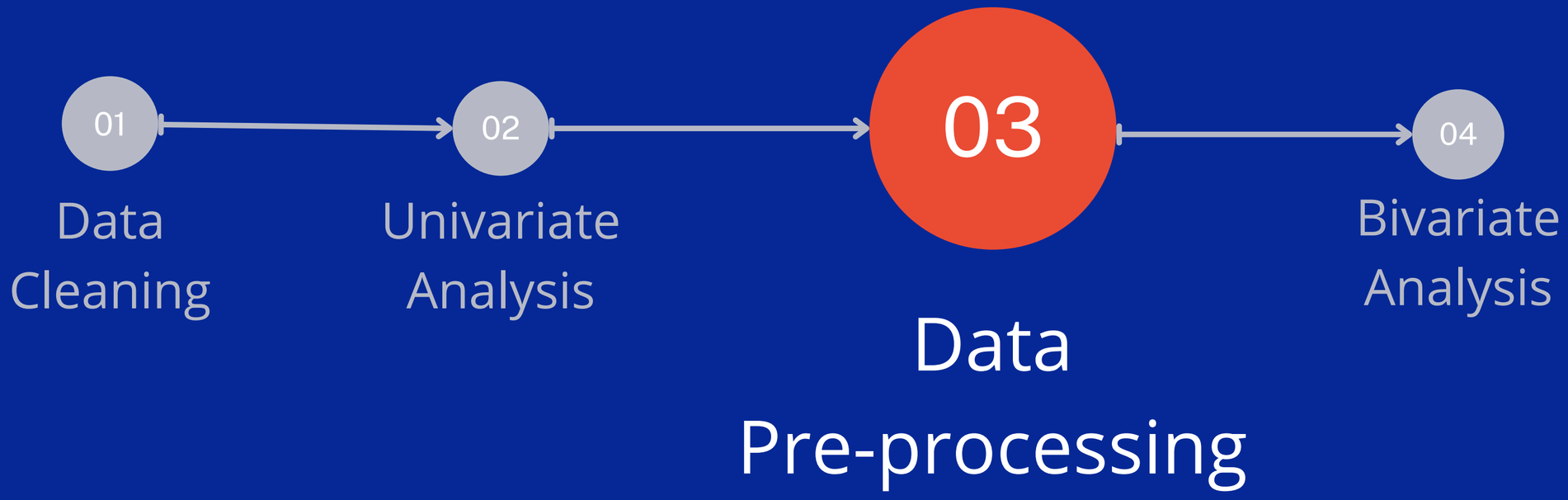
Target variable

Univariate Analysis

- **Nature:** Categorical, nominal
- **Description:** Credit score of each customer
- **Categories:** [‘Standard‘,‘Good‘,‘Poor‘]
- **Insights:** Around one third of the customers are classified as ‘Poor‘. These are the customers which is most important to correctly identify, as they can be most critical for the business. No information was given on the prior classification process.



DATA DESCRIPTION AND PREPARATION

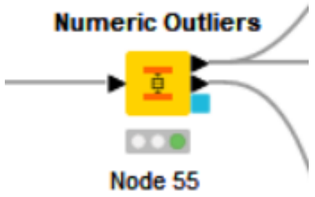


HANDLING OUTLIERS

- For the **Age** variable we casted to null values out of the the range 0-90, following logic and looking at distributions. We used a Rule Engine node.

```
$Age$ < 90 AND $Age$ > 0 => $Age$
```

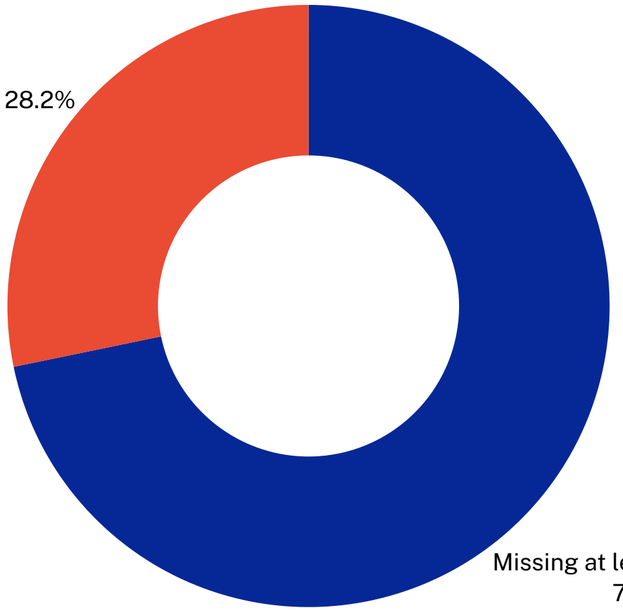
- For **other variables** we used the Numeric Outliers node, with an interquartile range multiplier **k = 1.5**, casting to null the detected outliers.



Annual_Income	348
Num_Bank_Accounts	160
Num_Credit_Card	288
Interest_Rate	268
Num_of_Loan	501
Delay_from_due_date	501
Num_of_Delayed_Payment	89
Changed_Credit_Limit	73
Num_Credit_Inquiries	207
Outstanding_Debt	659
Credit_Utilization_Ratio	0
Total_EMI	850
Amount_invested	953
Balance	970

HANDLING OUTLIERS – ALTERNATIVES

- **Dropping:** one viable alternative for the treatment of outliers is to directly drop the row. However, given the high number of outliers, we decided to not pursue this approach to conserve a good amount of observations.
- **Binning:** alternatively, we could have binned the numerical features, in order to reduce the impact of outliers transforming them into categorical. However, given the “extremely wrong” values, we believe the most logical approach was to cast them as null rather than consider them as valuable information.



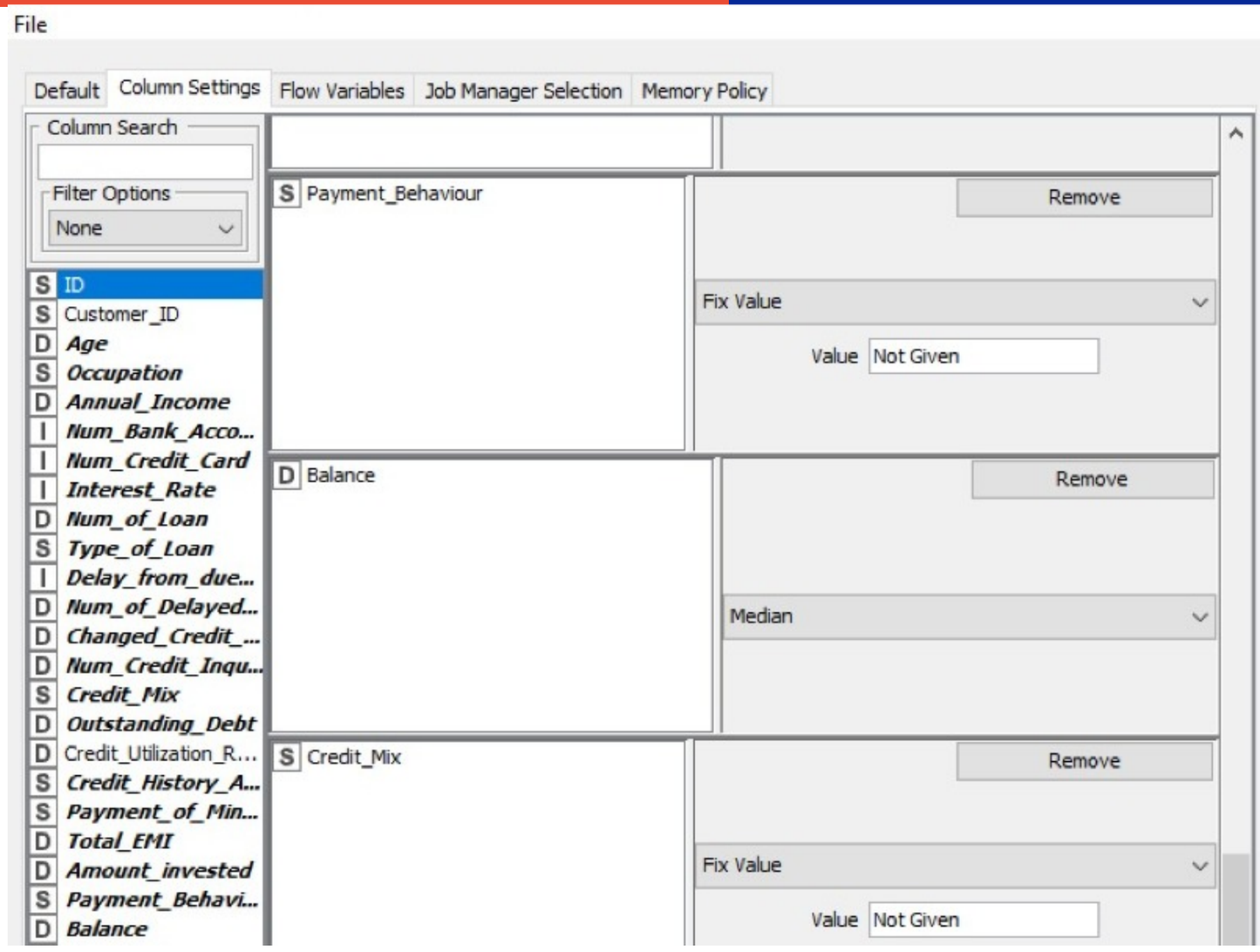
Percentage of records with at least one missing feature.

HANDLING THE NULLS

Numerical variables: we imputed the majority of variables with the median, as it is a more robust statistic than the mean. Few variables, like "Num_bank_accounts" and "Num_Credit_Card". "Total EMI" was imputed with 0, as NaNs were associated with customers without loans.

Categorial variables: we replaced the null values with a flag "Not Given", treating them as a category.

The next two slides show summary tables for all the variables and null details.



HANDLING THE NULLS

Data Pre-Processing

Name	Amount invested	Payment of Min Amount	Type of Loan	Credit History Age	Balance	Num of delayed payments	Payment behaviour	Occupation	Total EMI	Age
Type	Numerical Variable	Categorical variable	Categorical variable	Categorical Variable	Numerical Variable	Numerical Variable	Categorical variable	Categorical variable	Numerical Variable	Numerical Variable
NULL Percentage	16.2%	12.3%	11.4%	9%	8.9%	8%	7.7%	7%	6.8%	5.7%
Substituted with	Median	Most Frequent Value	Value Fixed as “NOLOAN”	Most Frequent Values	Median	Median	Value Fixed as “Not Given”	Value Fixed as “Not Given”	Value Fixed at 0.0	Median

HANDLING THE NULLS

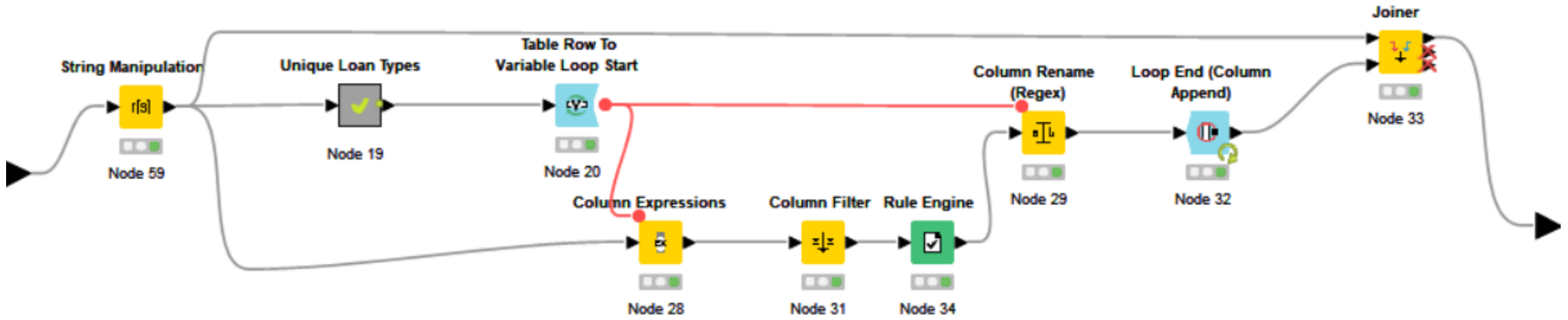
Data Pre-Processing

Name	Outstanding Debt	Delay From Due Date	Credit Mix	Number Of Loan	Number of Credit Inquires	Changed Credit Limit	Annual Income	Number of Credit Card	Interest Rate	Number of Bank Accounts
Type	Numerical Variable	Numerical Variable	Categorical Variable	Categorical variable	Numerical Variable	Numerical Variable	Numerical Variable	Numerical Variable	Numerical Variable	Numerical Variable
NULL Percentage	5.2%	4%	20.9%	4%	3.7%	2.9%	2.7%	2.3%	2.1%	1.2%
Substituted with	Median	Median	Value Fixed as “Not Given”	Value Fixed as “Not Given”	Median	Median	Median	Most Frequent Value	Median	Most Frequent Value

FEATURE ENGINEERING

Data Pre-Processing

- The **Type_of_loan** variable was a string reporting which types of loan were taken on by each customer, resulting in a set of multiple categories.
- We identified a list of unique loan types appearing in the dataset and created a column for each of these, unique loan type and inserted a flag indicating whether or not each customer had that specific loan type. We essentially combined string manipulation and one hot encoding.
- We noticed that null values of this feature were associated with a 0 in “Num_Loan”, so we decided to drop the entire binary column encoding the nulls as the information was already captured.



Type_of_Loan	Credit-...	Auto Loan	Mortga...	Debt C...	Payday...	Person...	Studen...	Home E...	Not Sp...
Credit-Builder Loan, and Mortgage Loan	1	0	1	0	0	0	0	0	0

FEATURE ENGINEERING

Data Pre-Processing

- The variable **Credit_History_Age** indicated for each customer the number of years and month of their credit history in a string format.

- We processed this feature variable transforming it into the total amount of months and casting it as an integer.

Before:

S Credit_History_Age
22 Years and 5 Months
26 Years and 11 Months
18 Years and 1 Months
17 Years and 7 Months
31 Years and 0 Months
32 Years and 3 Months

After:

D Credit_History_Age (months)
269
323
217
211
372
387

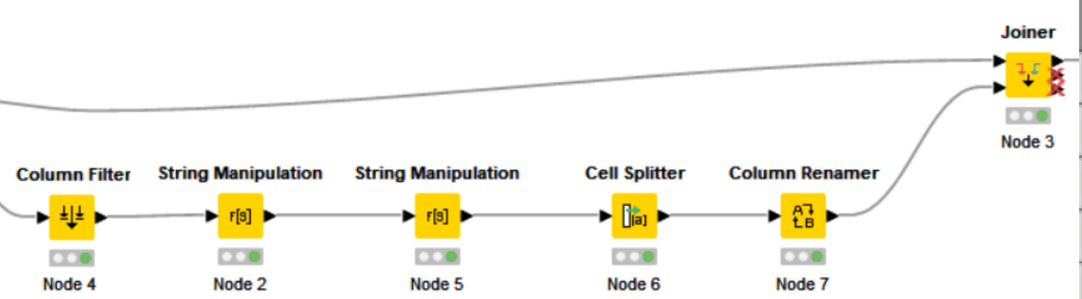
FEATURE ENGINEERING

Data Pre-Processing

- A similar issue was presented with the variable **Payment_Behaviour**, in which for each customer was indicated whether he/she had High/Low expenses and Small/Medium/Large value of payments.

- We created 2 new columns: one for the **amount spent**, and one for the **payments value**.

Payment_Behaviour
High_spent_Medium_value_payments
Low_spent_Small_value_payments
Low_spent_Large_value_payments
Low_spent_Large_value_payments
High_spent_Small_value_payments
Low_spent_Large_value_payments
High_spent_Large_value_payments



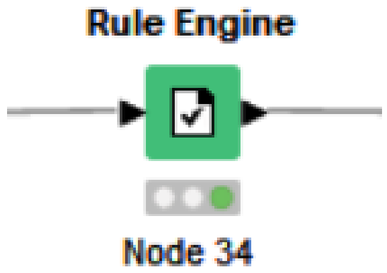
Amount_Spent	Payments_Value
High	Medium
Low	Small
Low	Large
Low	Large
High	Small
Low	Large

FEATURE ENGINEERING

Data Pre-Processing

- The variable **Payment_of_min_amount** contained “Yes“ and “No“ values as strings.

- We simply tranformed these into binary values, 1 representing yes and 0 representing no.




```
$Payment_of_Min_Amount$ = "Yes" => 1
$Payment_of_Min_Amount$ = "No" => 0
```

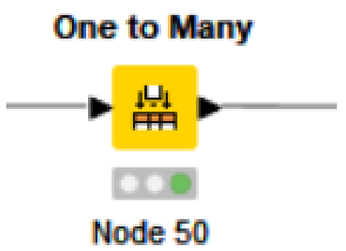
FEATURE ENGINEERING

We encoded categorical variables using the “**One to Many**” node.
For each possible value of each categorical variables, it creates a new column with either 1 or 0, i.e. it creates a one hot enconding. We used this procedure for the features:

- Occupation
- Credit_Mix
- Amount_Spent
- Payment_Value

Example for **Occupation**:

S Occupation
Scientist
Teacher
Lawyer
Entrepreneur
Developer



I Developer	I Lawyer
0	0
0	0
0	1
0	0
1	0

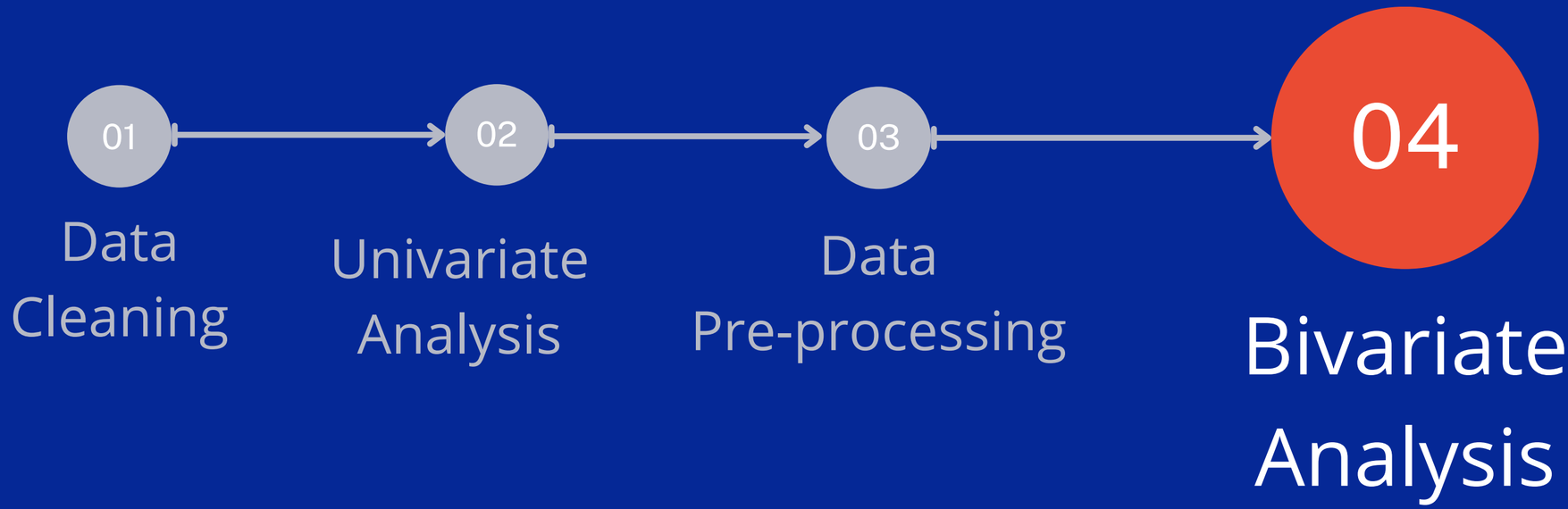
FEATURE ENGINEERING

Data Pre-Processing

We expanded the feature space to enhance the model predictive power by introducing interaction terms. Examples of features generated with this proceudre are:

- **income_per_card** (annual income / number of credit cards): the rationale behind this feature is that having a high income relative to the number of cards might be evidence of better wealth management skills.
- **income_per_age** (annual income / age): the rational is that younger people with more money could be really risk lover which could incur in their ability to pay pack loans.
- **debt_to_income** (outstanding debt / annual income): this metric indirectly reflects an individual's debt-to-income ratio, which is also a common ratio in accounting.

DATA DESCRIPTION AND PREPARATION



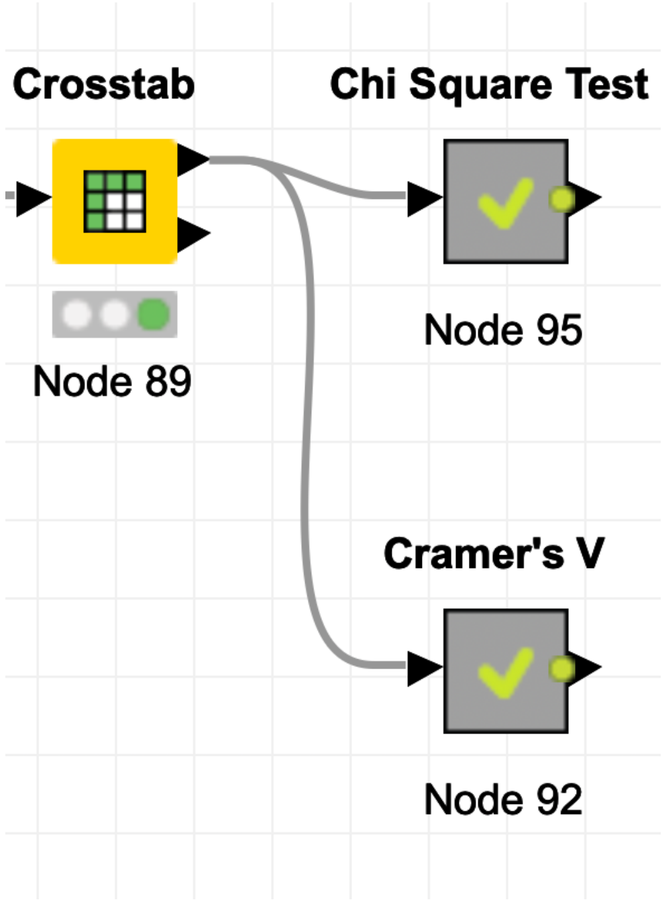
CRAMER'S V & CHI-SQUARE TEST

Data Pre-Processing

To perform the bivariate analysis of our initial categorical features against our target feature we're going to implement two main tests:

- **Cramer's V** is a number between 0 and 1 that measures how strongly two categorical fields are associated. It is computed by looking at how records splits between the combinations of two categorical features;
- **Chi-Square test** is used to state the significance of Cramer's V.

Together, these two tests give us a complete overview of how categorical variables in our dataset relates to the target.



CREDIT SCORE – OCCUPATION (1)

Bivariate Analysis

Cramer's V	Chi square test							
0.033 - Low	Frequency Expected Row Percent	Accountant	Architect	Developer	Doctor	Engineer	Entrepreneur	Journalist
	Good	151.0 143.77 6.54%	149.0 149.32 6.45%	156.0 145.25 6.75%	150.0 139.71 6.49%	146.0 146.18 6.32%	145.0 142.48 6.28%	146.0 140.45 6.32%
	Poor	247.0 231.03 6.65%	234.0 239.94 6.3%	232.0 233.41 6.25%	204.0 224.5 5.5%	249.0 234.9 6.71%	232.0 228.96 6.25%	219.0 225.69 5.9%
	Standard	380.0 403.19 5.87%	425.0 418.74 6.56%	398.0 407.34 6.14%	402.0 391.79 6.21%	396.0 409.93 6.11%	394.0 399.56 6.08%	395.0 393.86 6.1%

CREDIT SCORE – OCCUPATION (2)

Bivariate Analysis

Cramer's V	Chi square test									
0.033 - Low	Frequency Expected Row Percent	Lawyer	Manager	Mechanic	Media Manager	Musician	Scientist	Teacher	Writer	Not Given
	Good	145.0 154.49 6.28%	137.0 138.05 5.93%	131.0 146.18 5.67%	150.0 145.25 6.49%	147.0 136.2 6.36%	151.0 145.07 6.54%	147.0 141.56 6.36%	102.0 133.61 4.42%	157.0 162.44 6.8%
	Poor	254.0 248.26 6.84%	219.0 221.83 5.9%	240.0 234.9 6.47%	214.0 233.41 5.77%	205.0 218.86 5.52%	243.0 233.11 6.55%	235.0 227.47 6.33%	220.0 214.7 5.93%	265.0 261.03 7.14%
	Standard	437.0 433.25 6.75%	391.0 387.13 6.04%	420.0 409.93 6.48%	422.0 407.34 6.51%	385.0 381.94 5.94%	391.0 406.82 6.04%	384.0 396.97 5.93%	401.0 374.69 6.19%	457.0 455.53 7.05%

The two tests show no significant correlation between the target variable and the selected feature, forecasting low predictive power for it.

CREDIT SCORE – PAYMENT BEHAVIOUR

Bivariate Analysis

Cramer's V	Chi square test			
	Frequency Expected Row Percent	High Spending	Low Spending	Not Given
0.075 - Low	Good	1148.0 961.88 49.7%	995.0 1168.68 43.07%	167.0 179.44 7.23%
	Poor	1297.0 1545.68 34.94%	2126.0 1877.98 57.27%	289.0 288.35 7.79%
	Standard	2760.0 2697.44 42.61%	3203.0 3277.35 49.44%	515.0 503.21 7.95%

Also in this case we find low correlation between the selected variables.

CREDIT SCORE – PAYMENT VALUES

Bivariate Analysis

Cramer's V	Chi square test				
	Frequency Expected Row Percent	High Payments	Medium Payments	Small Payments	Not Given
0.064 - Low	Good	672.0 558.47 29.09%	756.0 716.1 32.73%	715.0 855.99 30.95%	167.0 179.44 7.23%
	Poor	774.0 897.41 20.85%	1078.0 1150.72 29.04%	1571.0 1375.52 42.32%	289.0 288.35 7.79%
	Standard	1576.0 1566.12 24.33%	2041.0 2008.18 31.51%	2346.0 2400.49 36.21%	515.0 503.21 7.95%

Also in this case we find low correlation between the selected variables.

CREDIT SCORE – PAYMENT OF MIN AMOUNT

Bivariate Analysis

Cramer's V	Chi square test		
0.408 - Medium	Frequency Expected Row Percent	Yes	No
	Good	593.0 1486.35 25.67%	1717.0 823.65 74.33%
	Poor	3068.0 2388.45 82.65%	644.0 1323.55 17.35%
	Standard	4382.0 4168.2 67.64%	2096.0 2309.8 32.36%

The correlation is expected, as paying the minimum amount is a good indicator for bad credit management.

CREDIT SCORE – TYPE OF LOAN (1)

Bivariate Analysis

Cramer's V	Type of Loan	Frequency Expected Row Percent	Good	Poor	Standard
0.14 - Low	Credit-Builder Loan	No	1791.0 1577.08 20.99%	2196.0 2534.26 25.73%	4547.0 4422.66 53.28%
		Yes	519.0 732.92 13.09%	1516.0 1177.74 38.22%	1931.0 2055.34 48.69%
0.139 - Low	Auto Loan	No	1830.0 1604.06 21.08%	2207.0 2577.61 25.43%	4643.0 4498.32 53.49%
		Yes	480.0 705.94 12.57%	1505.0 1134.39 39.4%	1835.0 1979.68 48.04%
0.153 - Low	Mortgage Loan	No	1792.0 1585.58 20.89%	2211.0 2547.92 25.77%	4577.0 4446.5 53.34%
		Yes	518.0 724.42 13.21%	1501.0 1164.08 38.29%	1901.0 2031.5 48.49%

CREDIT SCORE – TYPE OF LOAN (2)

Bivariate Analysis

Cramer's V	Type of Loan	Frequency Expected Row Percent	Good	Poor	Standard
0.135 - Low	Debt Consolidation Loan	No	1817.0 1592.98 21.08%	2250.0 2559.8 26.1%	4553.0 4467.23 52.82%
		Yes	493.0 717.02 12.71%	1462.0 1152.2 37.68%	1925.0 2010.77 49.61%
0.14 - Low	Payday Loan	No	1788.0 1572.09 21.02%	2189.0 2526.24 25.73%	4530.0 4408.67 53.25%
		Yes	522.0 737.91 13.07%	1523.0 1185.76 38.14%	1948.0 2069.33 48.79%
0.141 - Low	Personal Loan	No	1842.0 1591.5 21.39%	2251.0 2557.42 26.14%	4519.0 4463.08 52.47%
		Yes	468.0 718.5 12.04%	1461.0 1154.58 37.58%	1959.0 2014.92 50.39%

CREDIT SCORE – TYPE OF LOAN (3)

Bivariate Analysis

Cramer's V	Type of Loan	Frequency Expected Row Percent	Good	Poor	Standard
0.14 - Low	Student Loan	No	1792.0 1592.98 20.79%	2214.0 2559.8 25.68%	4614.0 4467.23 53.53%
		Yes	518.0 717.02 13.35%	1498.0 1152.2 38.61%	1864.0 2010.77 48.04%
0.126 - Low	Home-Equity Loan	No	1797.0 1584.66 20.96%	2260.0 2546.43 26.36%	4518.0 4443.91 52.69%
		Yes	513.0 725.34 13.07%	1452.0 1165.57 36.99%	1960.0 2034.09 49.94%
0.129 - Low	Not Specified	No	1772.0 1578.19 20.75%	2221.0 2536.04 26.01%	4547.0 4425.77 53.24%
		Yes	538.0 731.81 13.59%	1491.0 1175.96 37.65%	1931.0 2052.23 48.76%

CREDIT SCORE – CREDIT MIX

Bivariate Analysis

Cramer's V	Chi square test				
	Frequency Expected Row Percent	Bad Credit Mix	Standard Credit Mix	Good Credit Mix	Not Given
0.373 - Medium	Good	47.0 428.74 2.03%	347.0 842.13 7.61%	1438.0 556.99 47.71%	478.0 482.14 18.32%
	Poor	1386.0 688.95 59.74%	1044.0 1353.25 22.91%	489.0 895.04 16.22%	793.0 774.77 30.39%
	Standard	887.0 1202.32 38.23%	3166.0 2361.62 69.48%	1087.0 1561.98 36.07%	1338.0 1352.09 51.28%

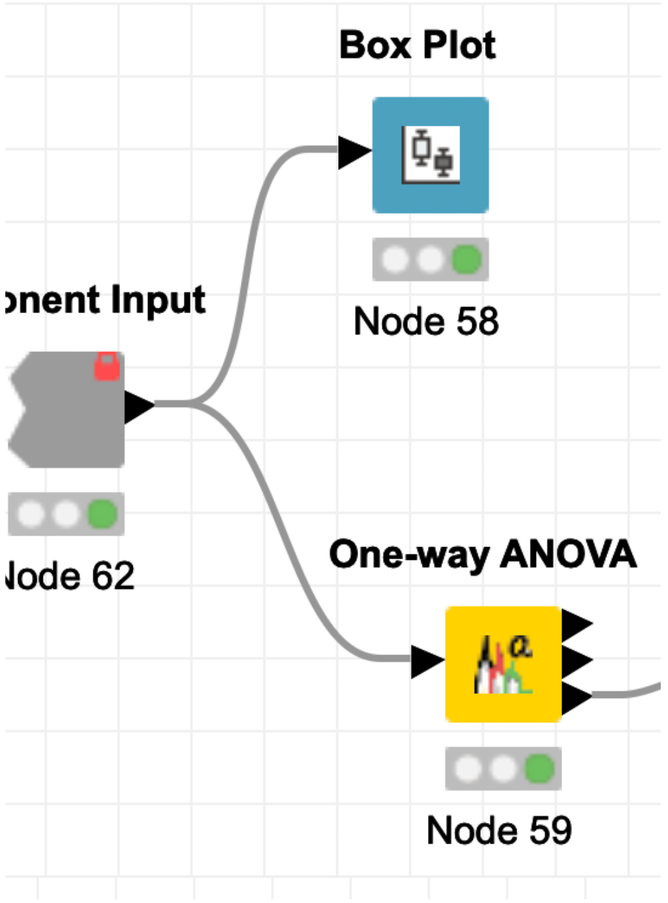
Again, the correlation is logical: holders of bad credit mixes will be more likely to be classified as poor credit holders, and the same goes for the other categories.

ONE–WAY ANOVA & BOX PLOT

Data Pre-Processing

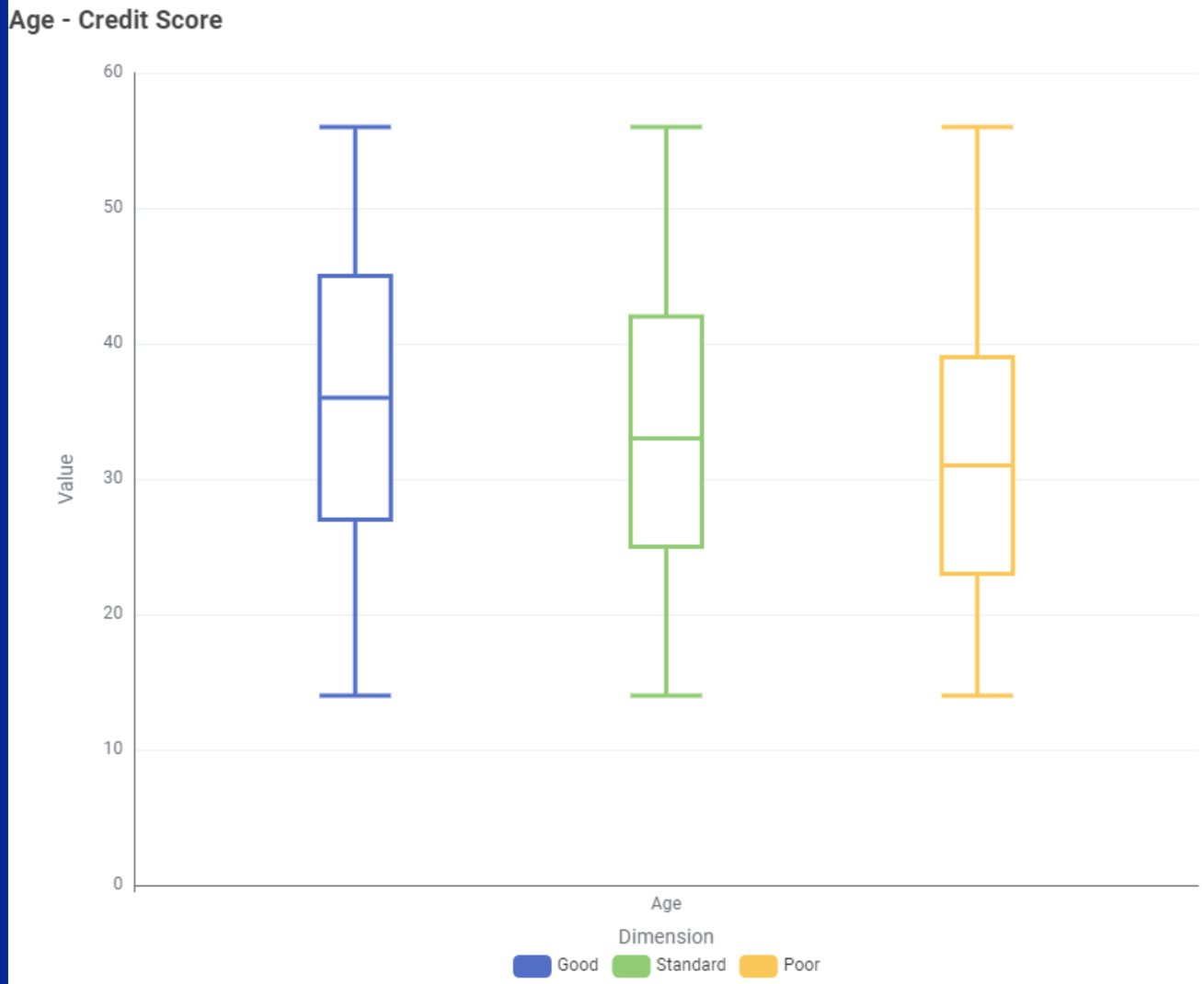
On the other hand, doing bivariate analysis between our target variable and the continuous variables present in our dataset will need the implementation of these tests:

- **One-Way ANOVA** is a test to understand if there's a significant difference in means of continuous variables when divided by the value of the target variable;
- **Boxplots** are used in order to confirm the result of the One-Way AnOVa test in a visual way, and to provide stakeholders more easily interpretable metrics.



CREDIT SCORE – AGE

Bivariate Analysis

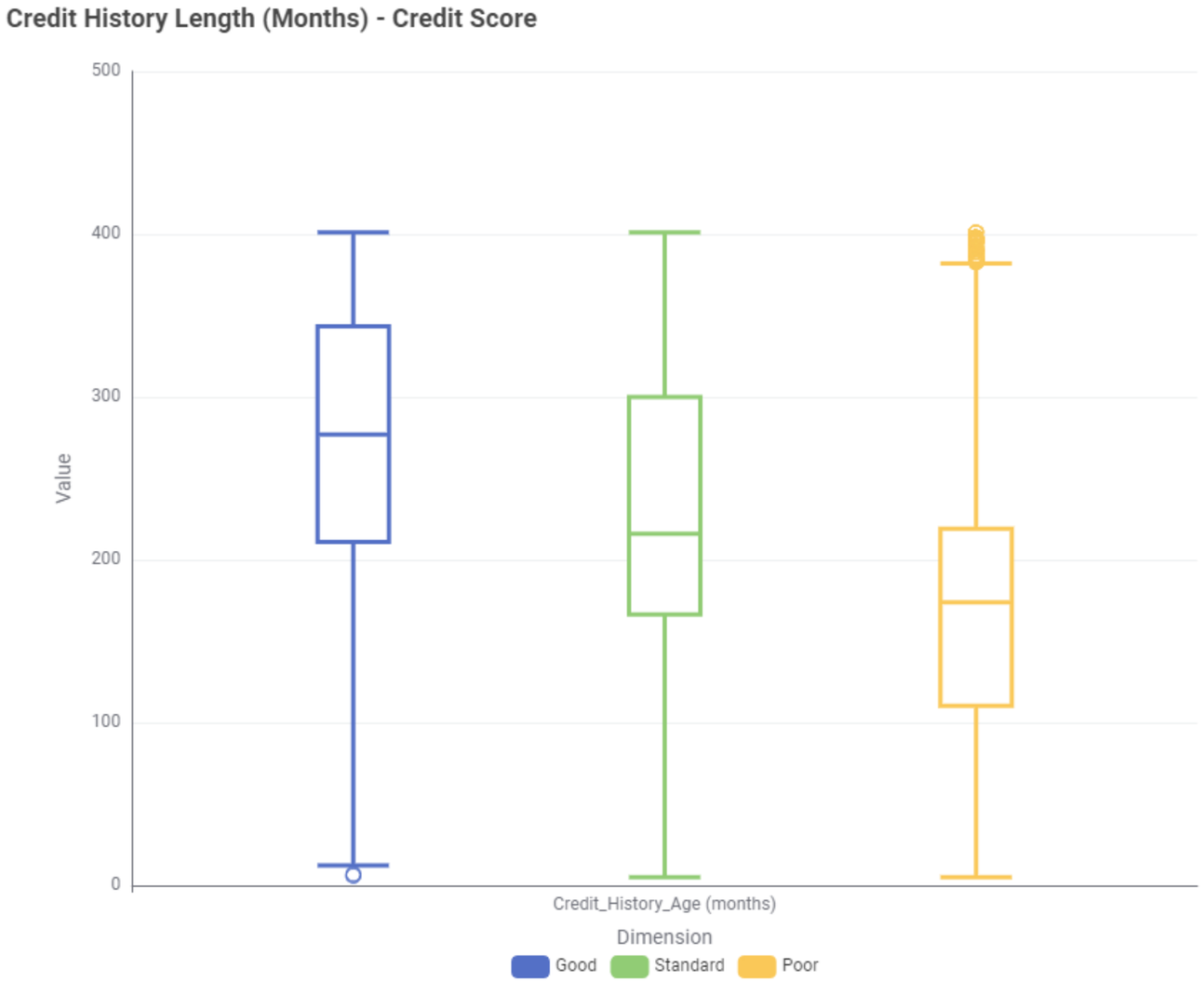


	Group	N° of records	Mean	Standard Deviation
Credit Score	Good	2310	36.12	11.01
	Standard	6478	33.46	10.63
	Poor	3712	31.39	9.91
	Total	12500	33.34	10.62

As we can see from the boxplot, the average age of people with a **poor** credit score is the same as the one of people with a **standard** or an **high** credit score.

CREDIT SCORE – CREDIT HISTORY LENGTH

Bivariate Analysis

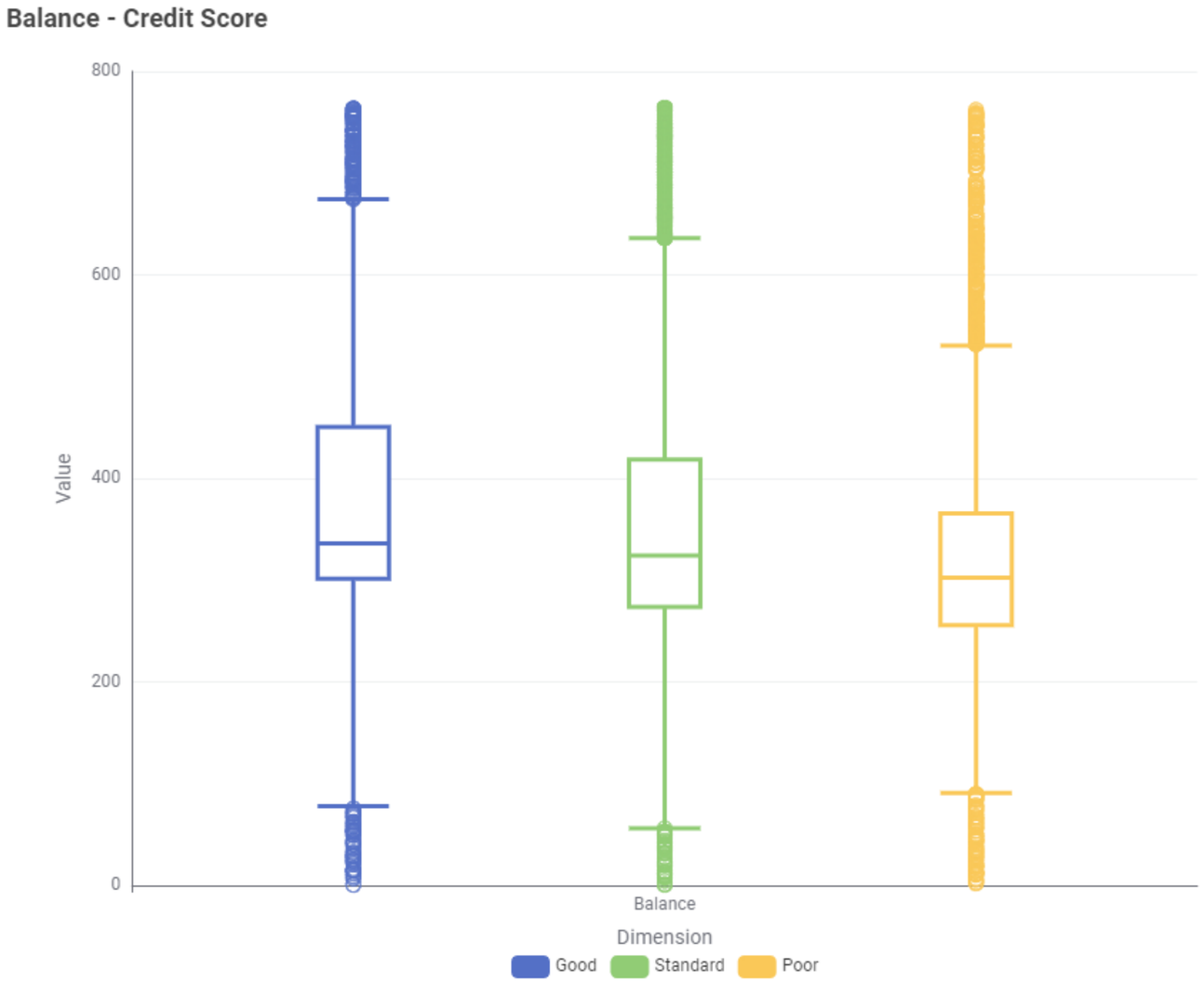


	Group	N° of records	Mean	Standard Deviation
Credit Score	Good	2310	274.32	77.78
	Standard	6478	225.21	95.12
	Poor	3712	174.21	84.82
	Total	12500	219.14	95.52

Here the difference in the mean for the 3 categories of Credit Score is even more significant. The longer the Credit History is, the more likely is the Credit Score to be higher.

CREDIT SCORE – BALANCE

Bivariate Analysis

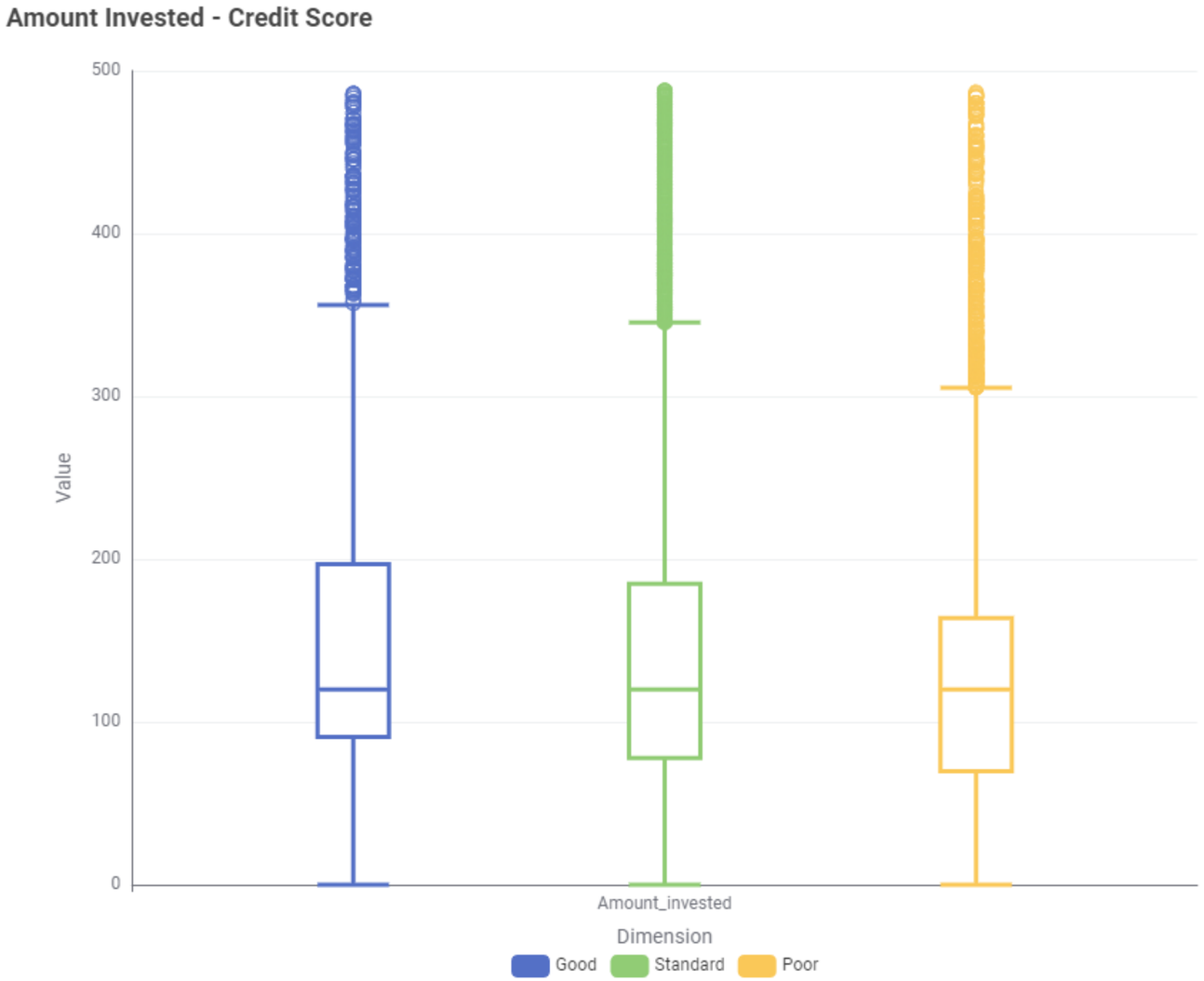


	Group	N° of records	Mean	Standard Deviation
Credit Score	Good	2310	378.48	142.66
	Standard	6478	358.63	136.58
	Poor	3712	323.39	120.99
	Total	12500	351.83	134.79

In this case, the mean is almost the same in each category of Credit Score. However, customers with a Poor score have a balance that is more concentrated around the mean.

CREDIT SCORE – AMOUNT INVESTED

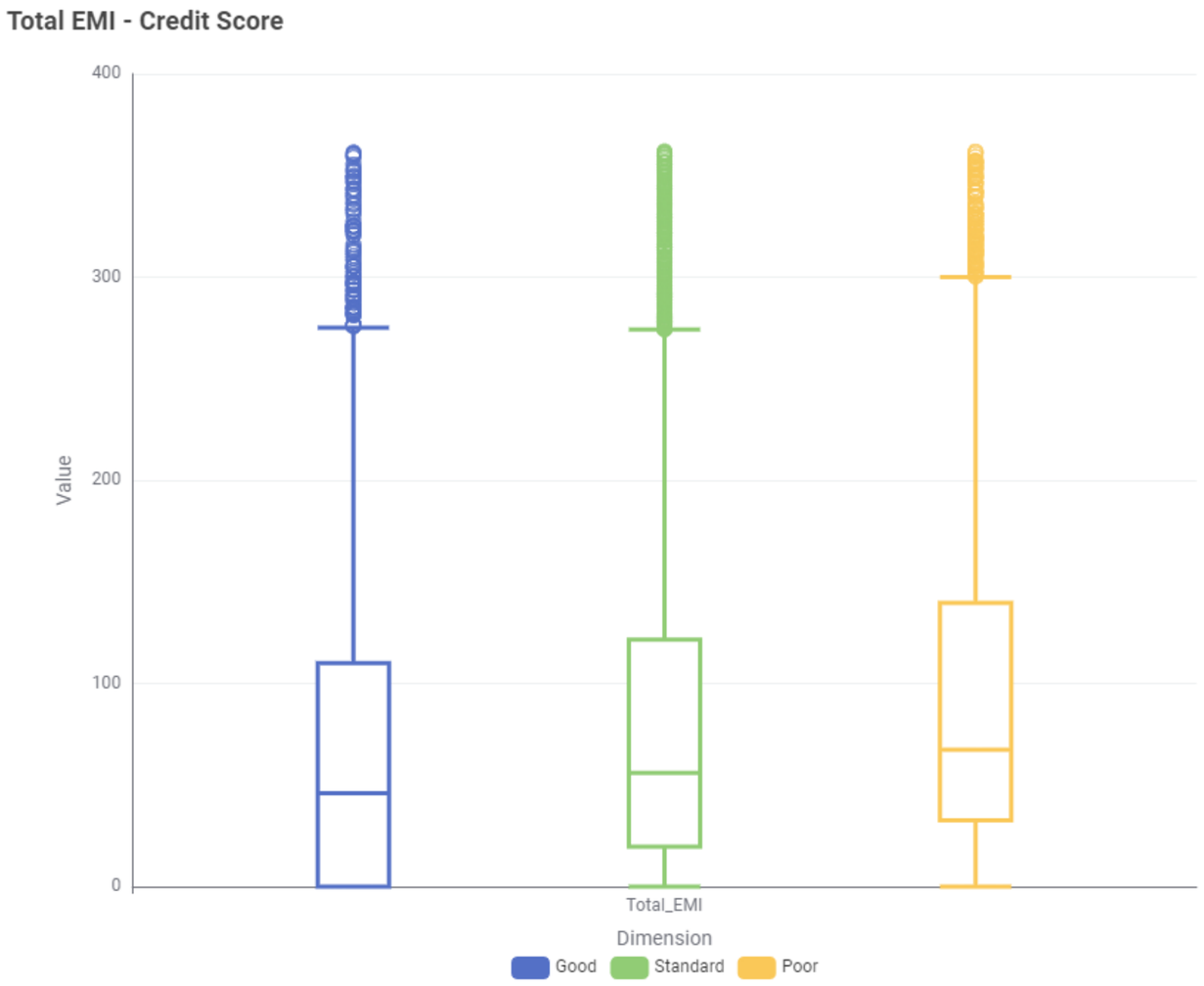
Bivariate Analysis



	Group	N° of records	Mean	Standard Deviation
Credit Score	Good	2310	154.16	100.89
	Standard	6478	147.25	100.57
	Poor	3712	134.53	92.99
	Total	12500	144.75	98.69

CREDIT SCORE – TOTAL EMI

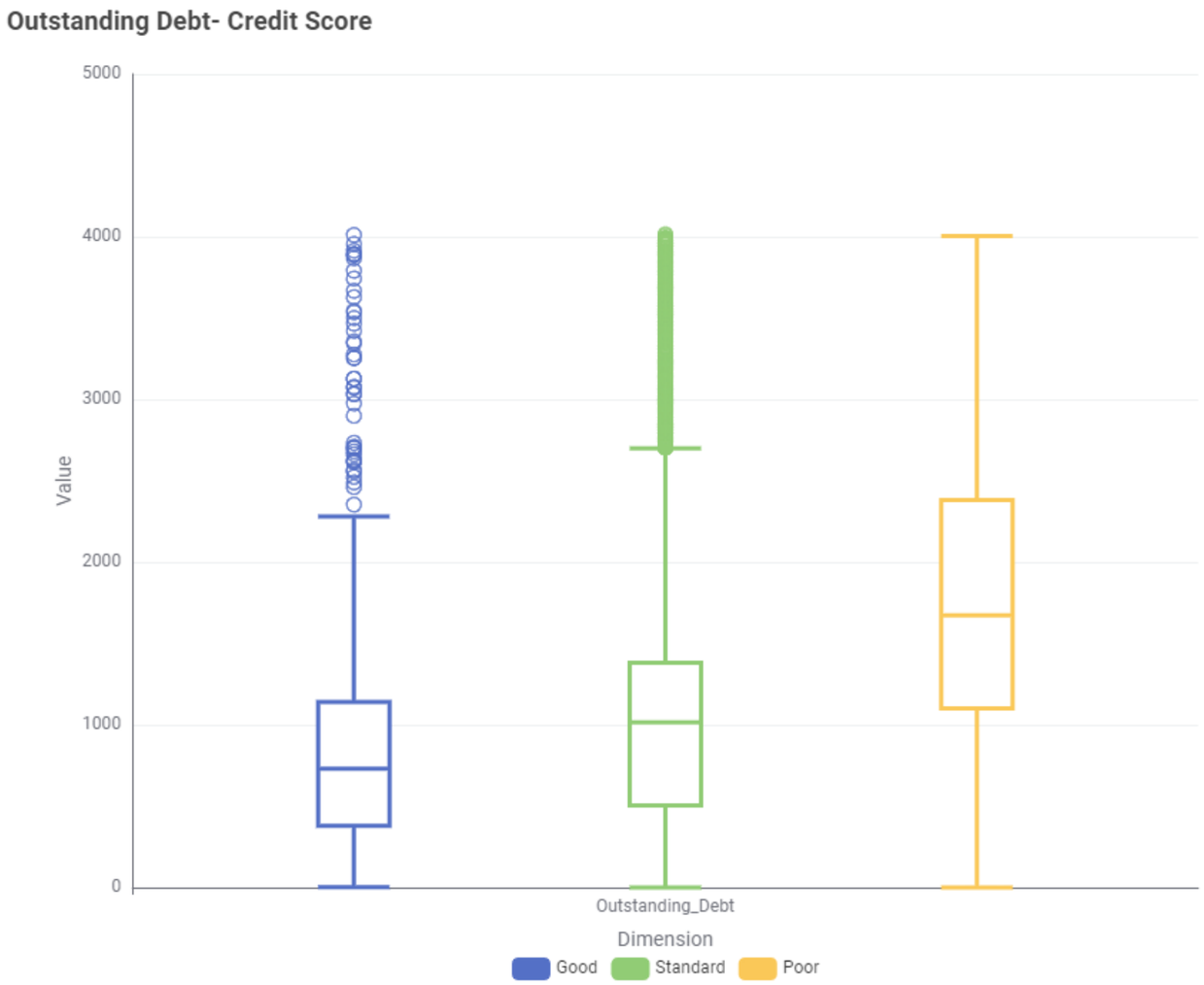
Bivariate Analysis



	Group	N° of records	Mean	Standard Deviation
Credit Score	Good	2310	75.05	85.46
	Standard	6478	81.99	82.55
	Poor	3712	94.84	84.42
	Total	12500	84.53	83.95

CREDIT SCORE – OUTSTANDING DEBT

Bivariate Analysis

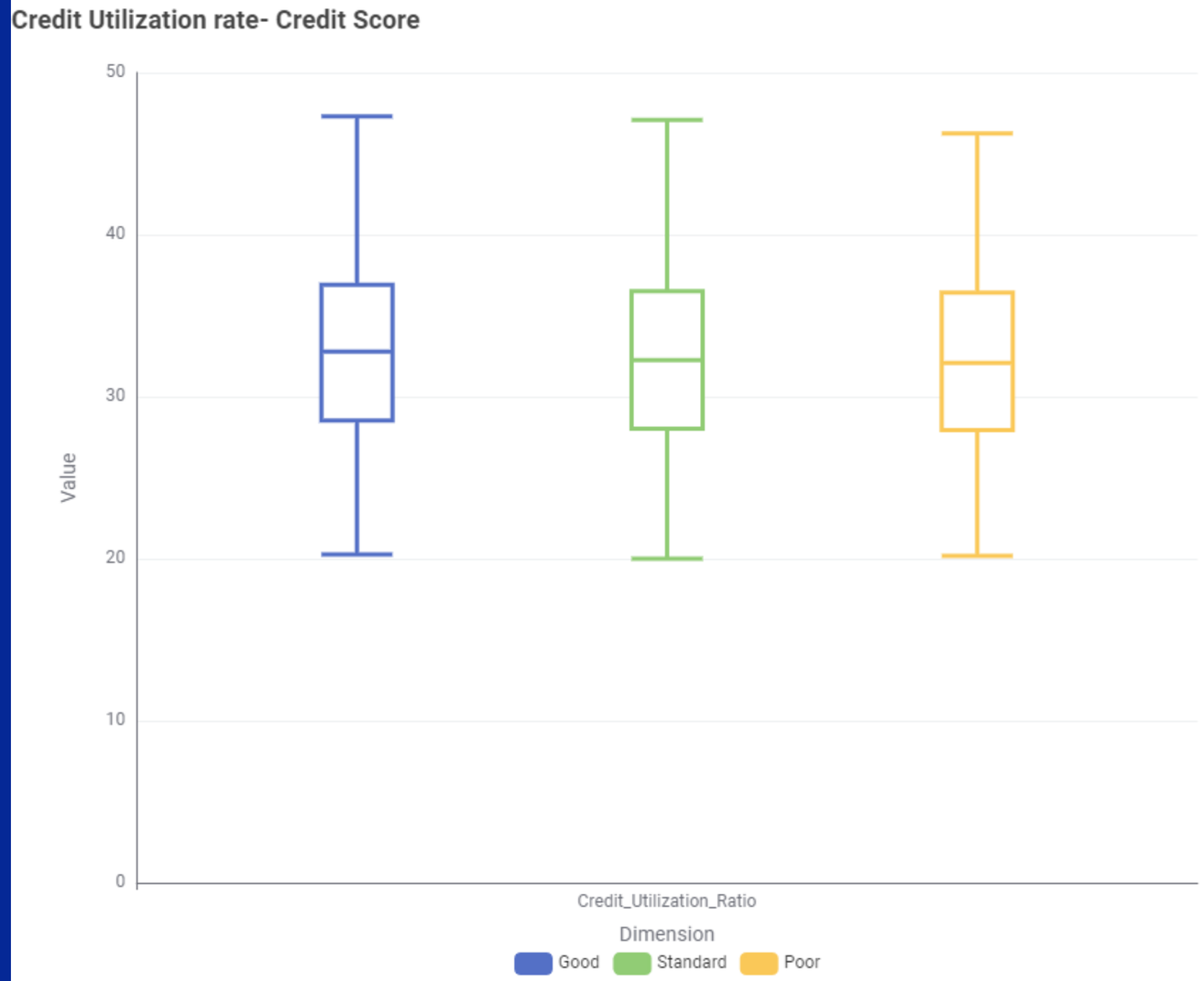


	Group	N° of records	Mean	Standard Deviation
Credit Score	Good	2310	785.45	555.22
	Standard	6478	1122.90	857.78
	Poor	3712	1748.02	899.20
	Total	12500	1246.17	894.54

Clearly, the higher the outstanding debt is, the worse is, on average, the credit score.

CREDIT SCORE – CREDIT UTILIZATION RATE

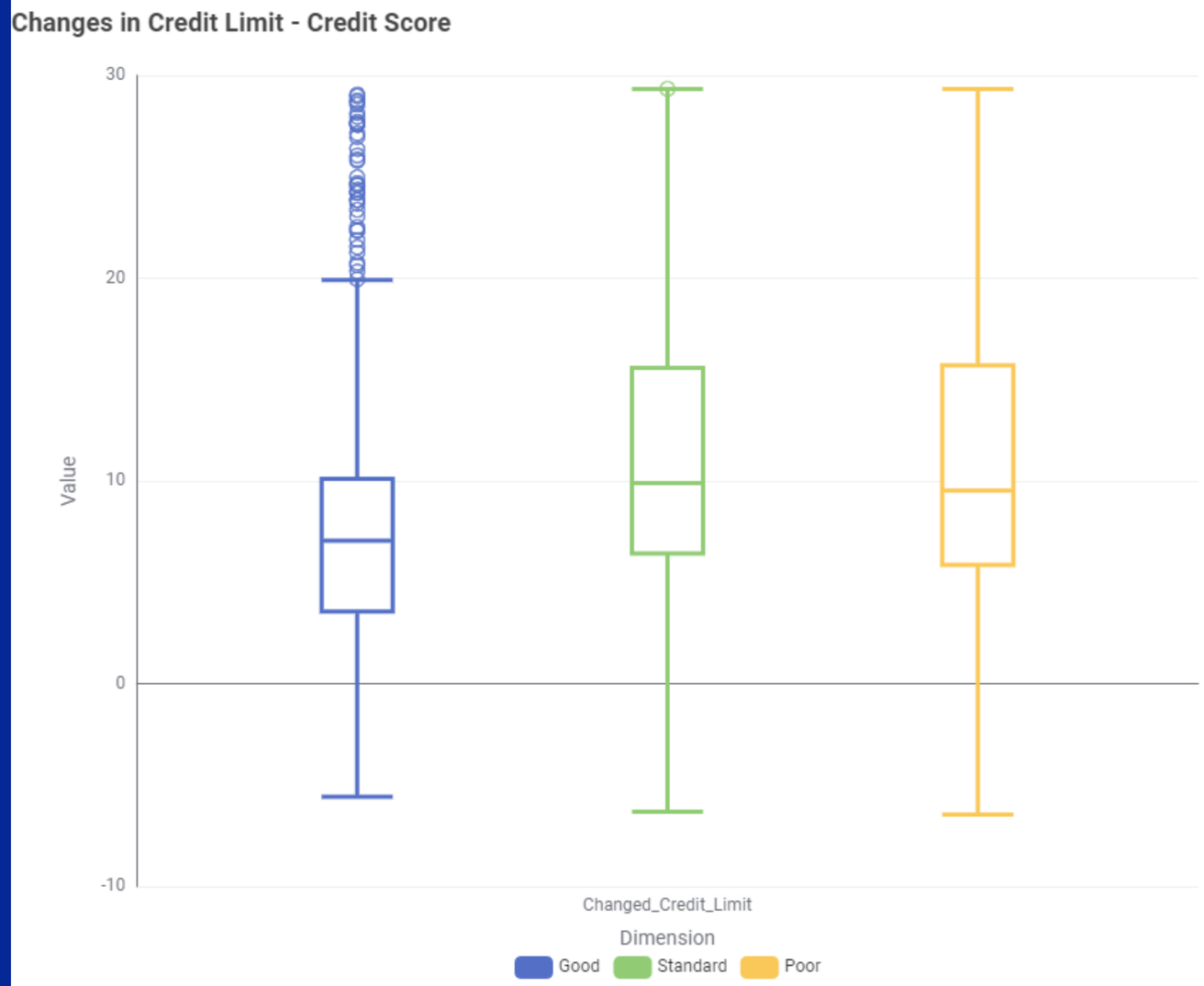
Bivariate Analysis



	Group	N° of records	Mean	Standard Deviation
Credit Score	Good	2310	32.76	5.15
	Standard	6478	32.26	5.12
	Poor	3712	32.11	5.09
	Total	12500	32.31	5.12

CREDIT SCORE – CHANGES IN CREDIT LIMIT

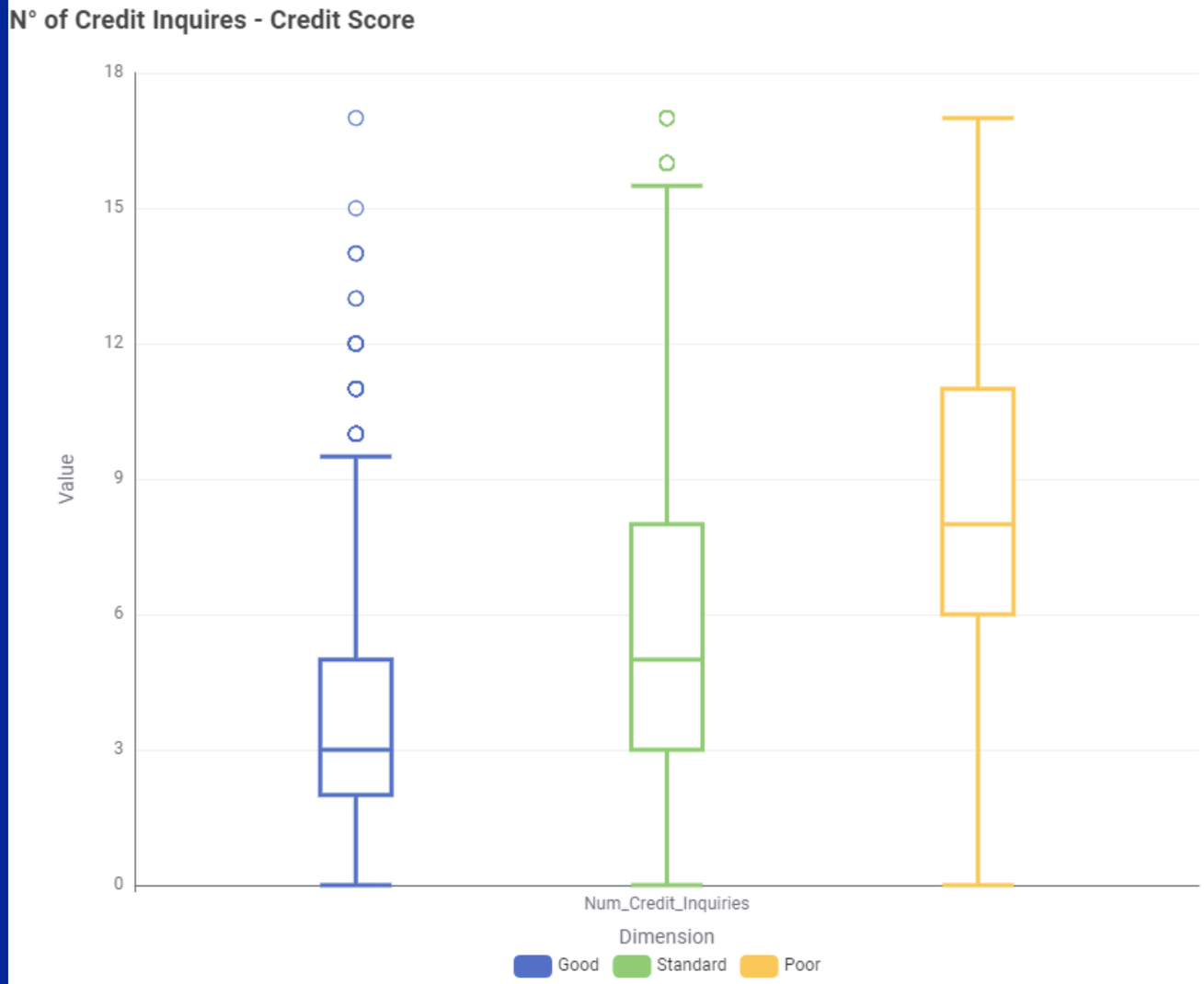
Bivariate Analysis



	Group	N° of records	Mean	Standard Deviation
Credit Score	Good	2310	7.32	4.98
	Standard	6478	10.91	6.52
	Poor	3712	10.89	6.88
	Total	12500	10.24	6.52

CREDIT SCORE – # CREDIT INQUIRES

Bivariate Analysis



	Group	N° of records	Mean	Standard Deviation
Credit Score	Good	2310	3.60	2.60
	Standard	6478	5.51	3.58
	Poor	3712	8.03	3.75
	Total	12500	5.91	3.81

As we can see from the boxplot, customers with a good credit score have averagely less credit inquires than customers with a poor credit score.

CREDIT SCORE – DELAY FROM DUE DATE

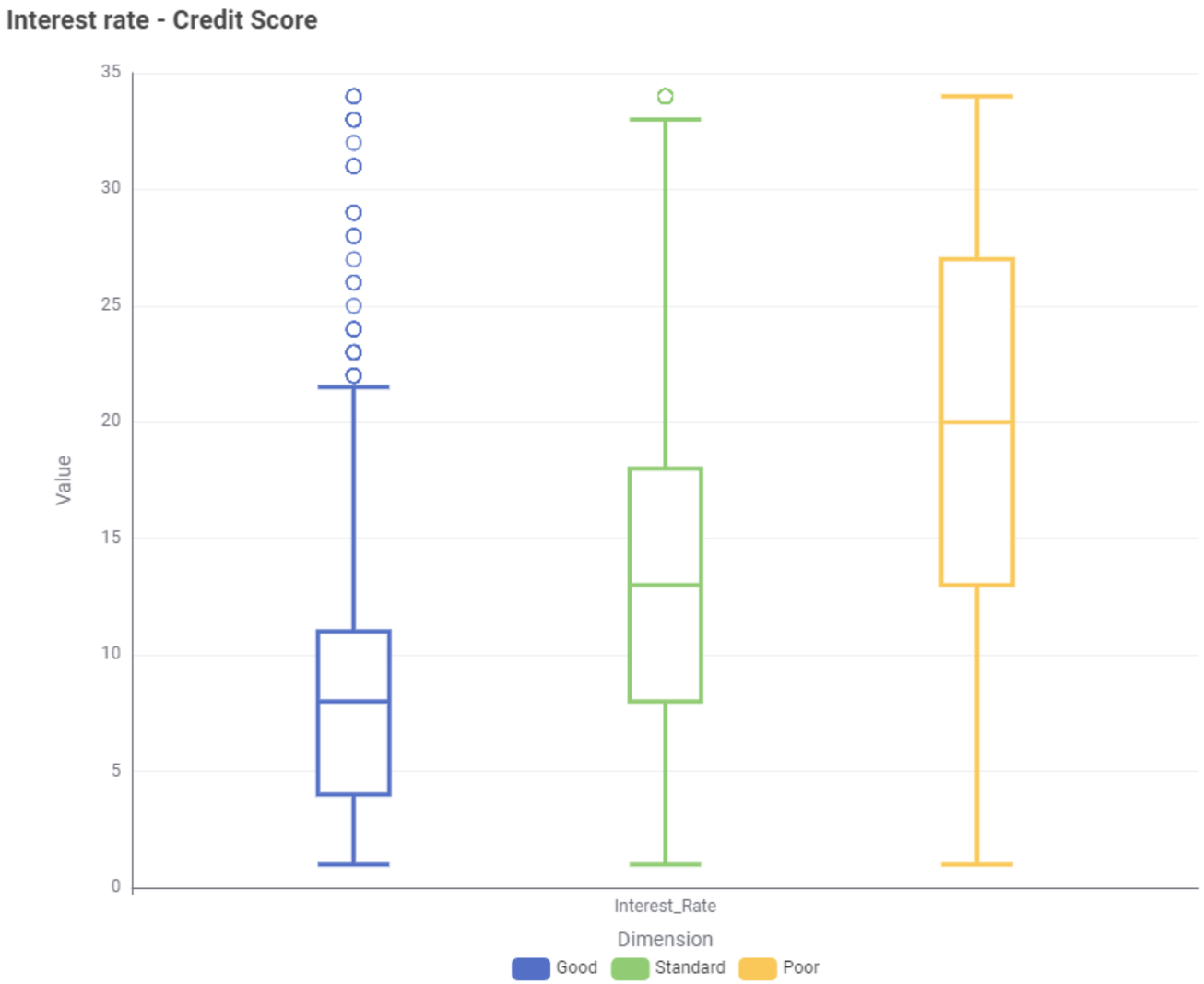
Bivariate Analysis

	Group	N° of records	Mean	Standard Deviation
Credit Score	Good	2310	3.60	2.60
	Standard	6478	5.51	3.58
	Poor	3712	8.03	3.75
	Total	12500	5.91	3.81

Credit score is negatively correlated with the delay from due date. The more the delay, the lower the credit score.

CREDIT SCORE – INTEREST RATE

Bivariate Analysis

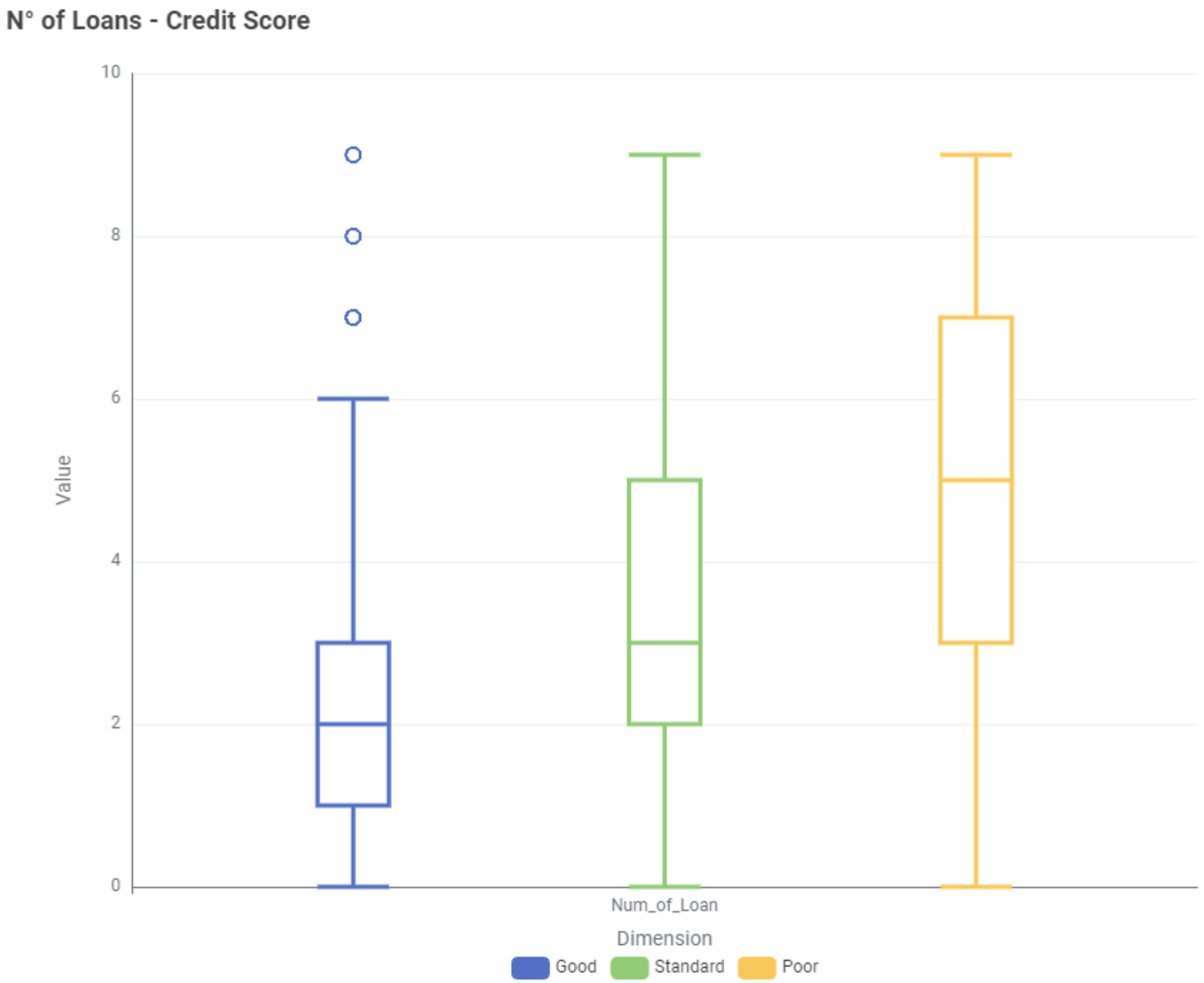


	Group	N° of records	Mean	Standard Deviation
Credit Score	Good	2310	8.17	5.19
	Standard	6478	13.78	7.72
	Poor	3712	19.69	8.87
	Total	12500	14.49	8.65

The boxplot shows that customers with a poor credit score will likely face an higher interest rate.

CREDIT SCORE – # LOANS

Bivariate Analysis

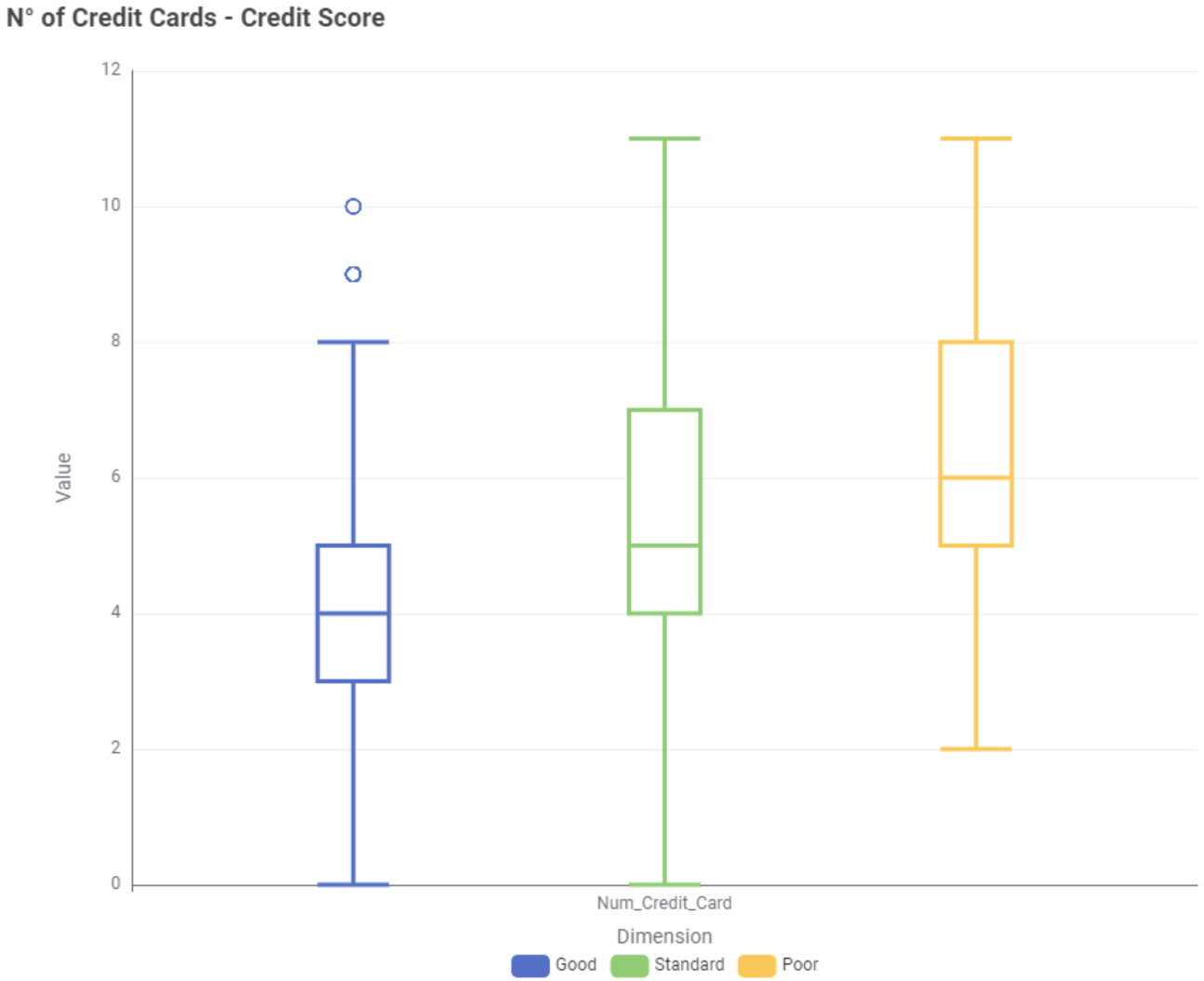


	Group	N° of records	Mean	Standard Deviation
Credit Score	Good	2310	2.30	1.69
	Standard	6478	3.29	2.33
	Poor	3712	4.64	2.43
	Total	12500	3.51	2.40

The better the credit score, the higher the number of loan for each customer usually is.

CREDIT SCORE – # CREDIT CARDS

Bivariate Analysis

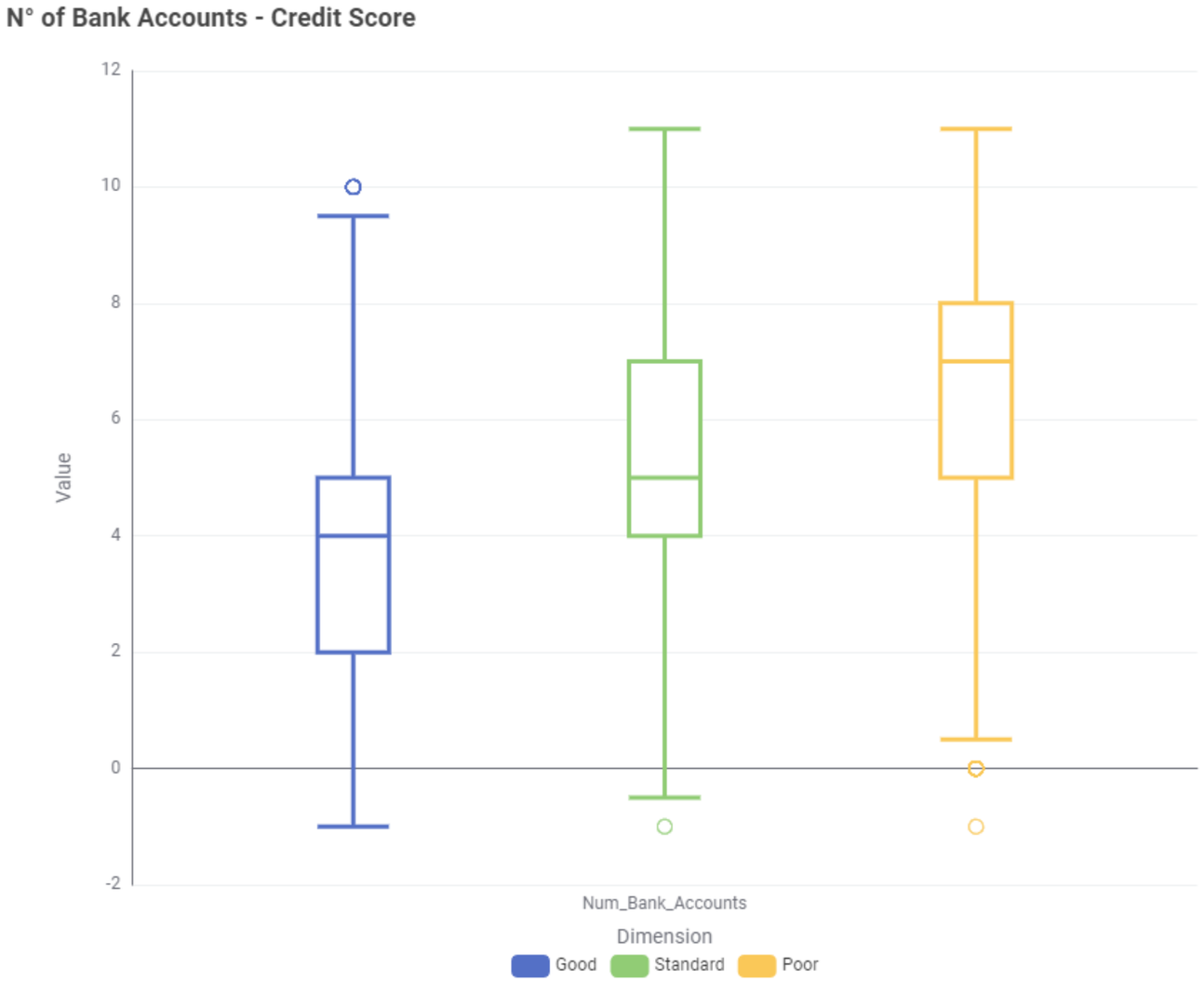


	Group	N° of records	Mean	Standard Deviation
Credit Score	Good	2310	4.25	1.78
	Standard	6478	5.41	1.88
	Poor	3712	6.52	1.98
	Total	12500	5.52	2.05

Customers with a poor credit score have, on average, an higher number of credit cards.

CREDIT SCORE – # BANK ACCOUNTS

Bivariate Analysis

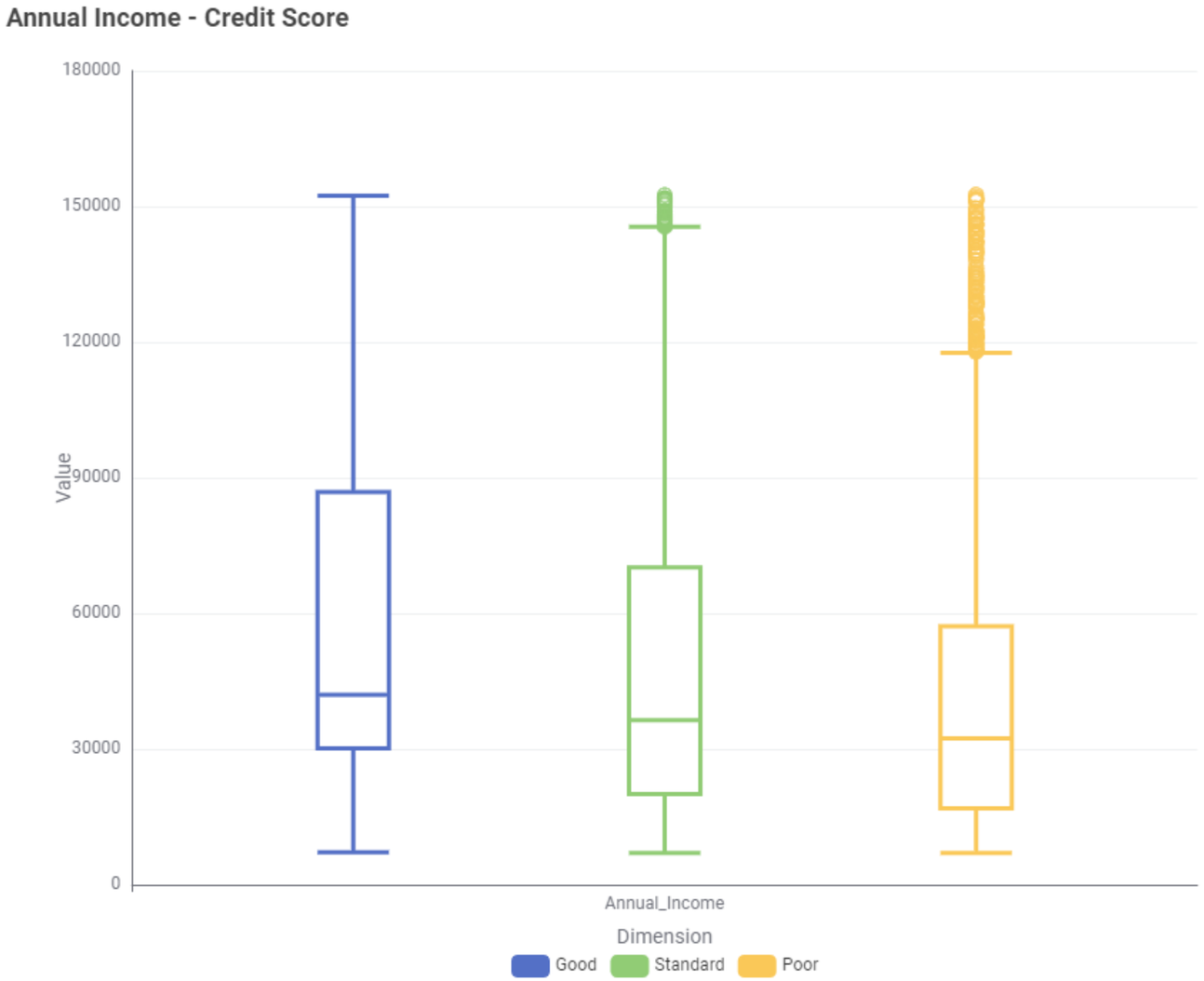


	Group	N° of records	Mean	Standard Deviation
Credit Score	Good	2310	4.18	1.99
	Standard	6478	5.59	2.16
	Poor	3712	6.65	2.25
	Total	12500	5.64	2.31

Customers with a poor credit score usually have a higher number of bank accounts.

CREDIT SCORE – ANNUAL INCOME

Bivariate Analysis



	Group	N° of records	Mean	Standard Deviation
Credit Score	Good	2310	4.25	1.78
	Standard	6478	5.41	1.88
	Poor	3712	6.52	1.98
	Total	12500	5.52	2.05

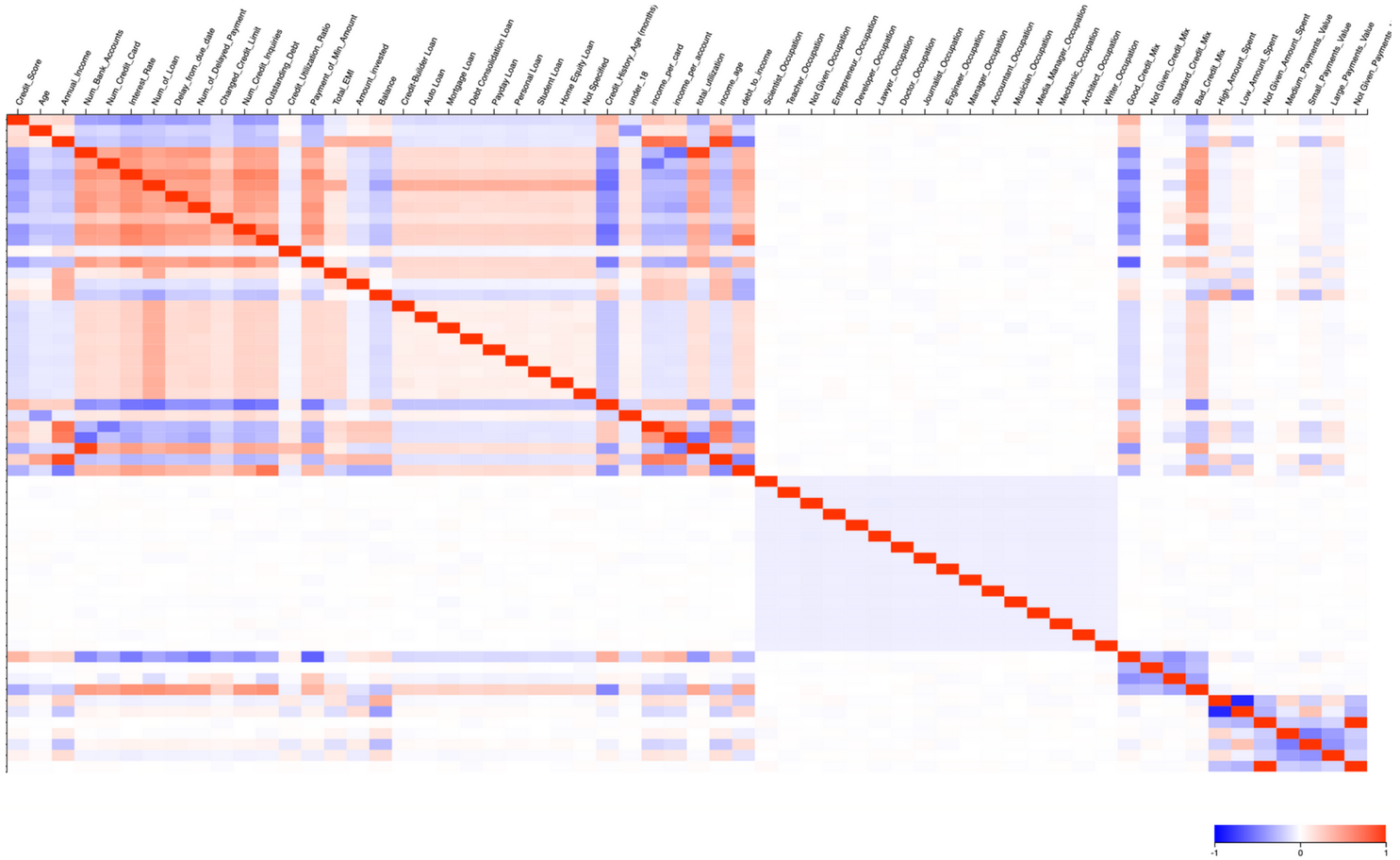
The better the credit score, the higher the annual income of the customer usually is.

CORRELATION HEATMAP OF FEATURES

Bivariate Analysis

The table on the right is used to summarize **linear correlations** between features.

The first row indicates correlations with our target feature, a useful indication on the informative power of the other characteristics recorded; all the other squares, on the other hand, are helpful to check for multicollinearity between predictors.



CORRELATION HEATMAP OF FEATURES

Bivariate Analysis

Negative correlations

- **interest rate**: the higher the credit scoring, the lower the default that will be required by the bank.
- **delay from due date**: as a delay from due date increases, there could be a higher risk of late payments negatively impacting credit scores.

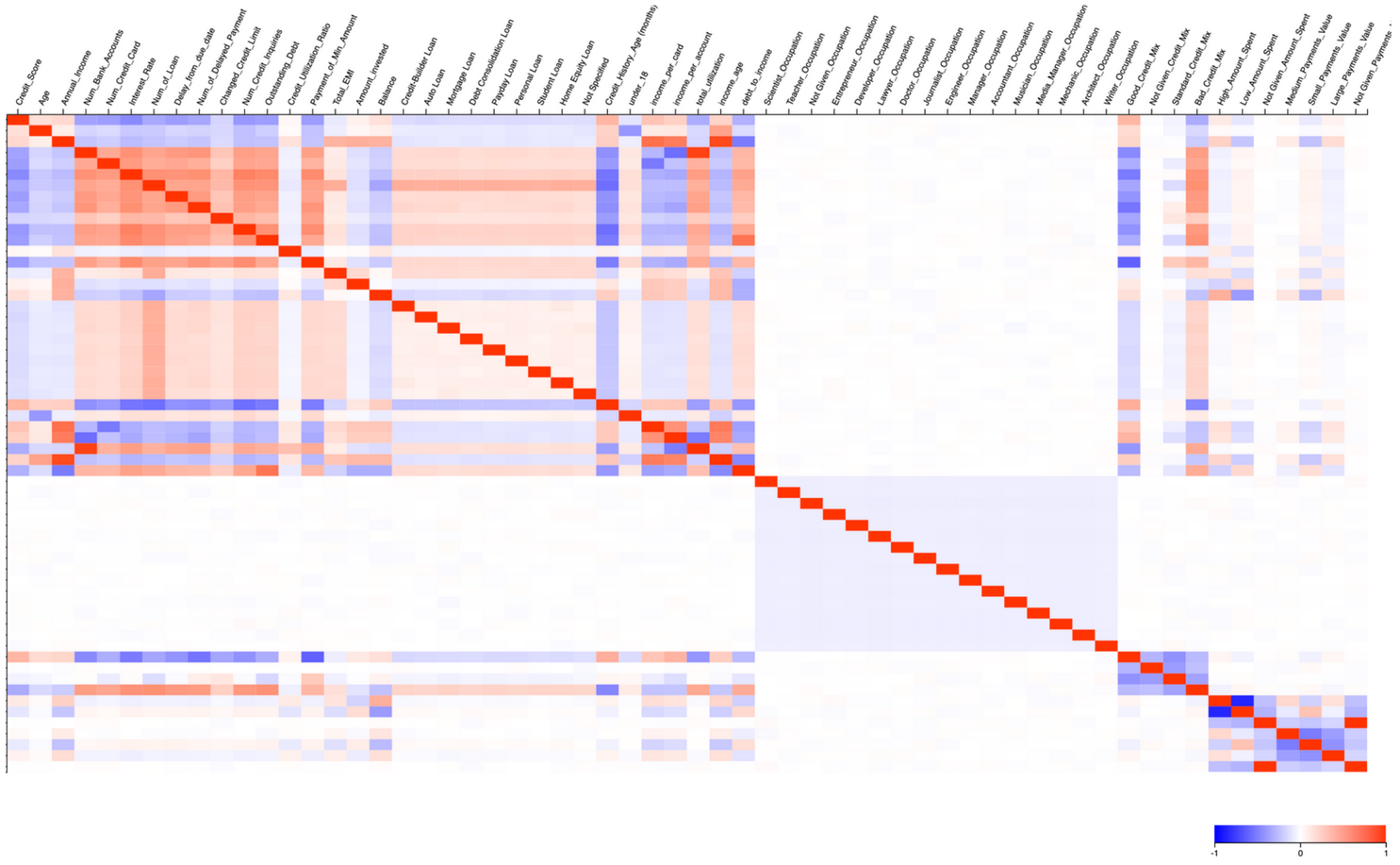
Positive correlations

- **credit history age:** a longer period of proven financial responsibility and reliability in managing credit is associated to higher scores.
- **good credit mix**: a high quality of credit reflects high diversity of credit accounts, including credit cards, as mortgage loans, reflected by high credit scores.

CORRELATION HEATMAP OF FEATURES

Bivariate Analysis

As a final remark, we note that some features present correlation among each other, while some other features present no correlation with the target variable, which might hint to low predictive power. We decided not to drop any predictor as this heatmap is indicative of linear correlation only. We might have some higher order correlation which does not appear here.

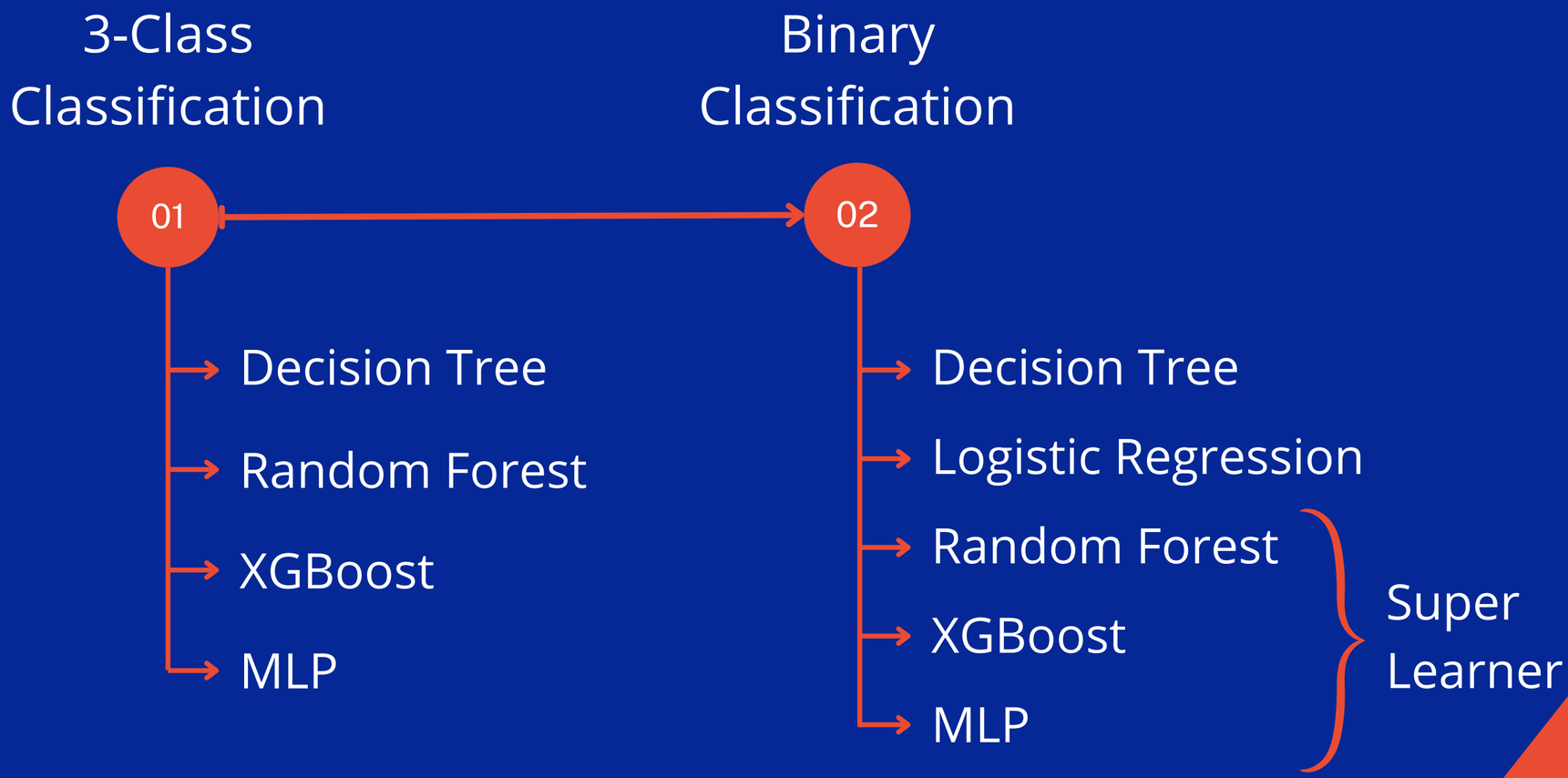


RECAP

- **Data cleaning and validation:** we successfully imported and cleaned the dataset, ensured accuracy in data types, aligning each with its corresponding variable.
- **Outliers:** we conducted a thorough analysis to identify outliers, and applied appropriate techniques to manage and rectify them.
- **Null values:** we identified and addressed missing values across the dataset, employing various strategies and achieving a dataset with no missing entries.
- **Feature Engineering:** we processed existing columns and created interaction terms, expanding the dataset feature space.

The dataset now contains 59 predictors and the credit-score target variable. It is ready to be modeled.

DATA MODELING



OVERVIEW

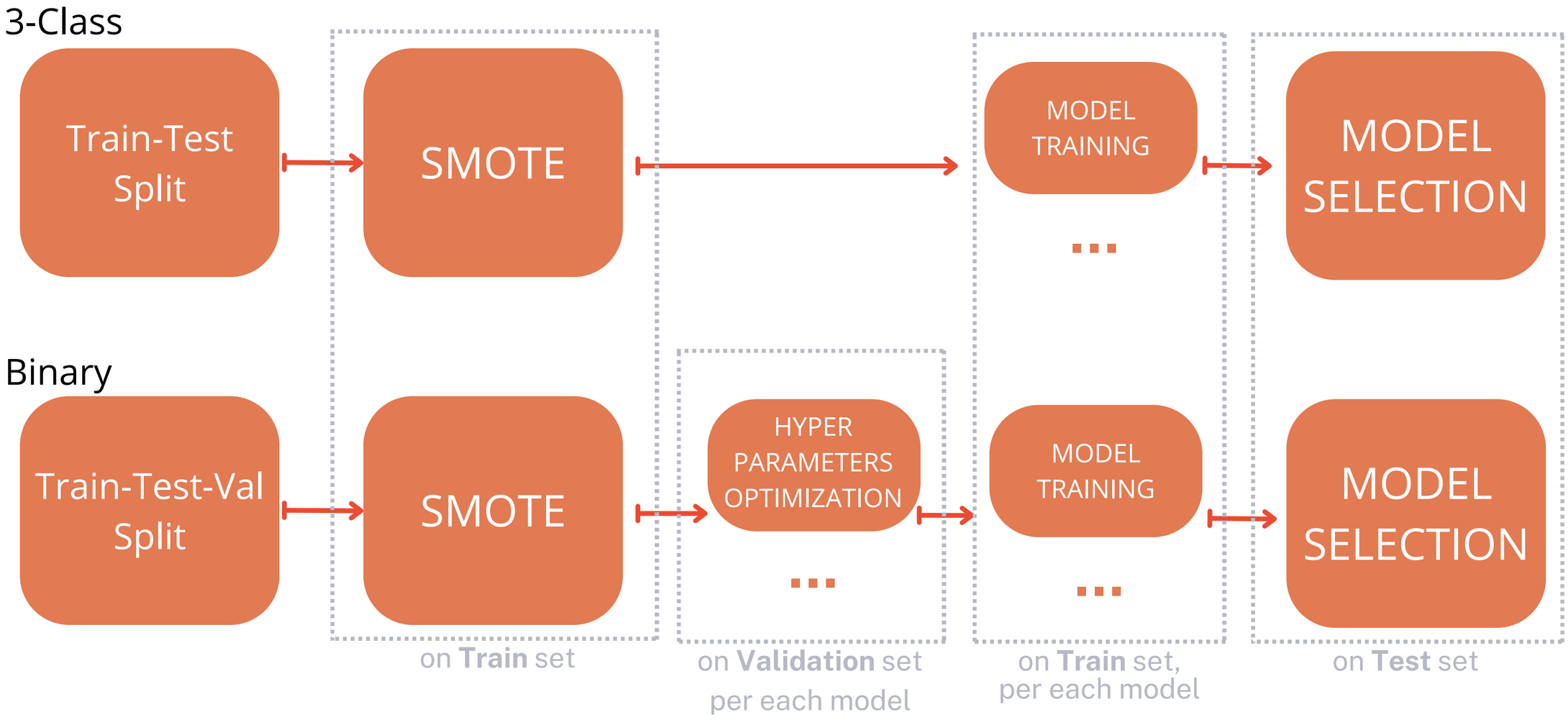
Our analysis aims at predicting the variable **Credit_Score**, that indicates whether each customer has a **Good**, **Standard** or **Poor** credit score. To do so, we first tested different *Multi-Class models*, that predicted all 3 possible values for the variable.

We then casted this multi-class classification to a binary classification, as we realized that for a bank it may be less useful to discriminate between a Good and Standard customer. Rather, it is more crucial to effectively identify the customers with Poor credit score, as these are the riskiest for the institution. Therefore, we decided to implement *Binary models* performing this classification:

- Poor ==> 1
- Good/Standard ==> 0

Moreover, we deem the tradeoff between costs and results to be preferable for the second task, and thus focused our analysis on this binary classification problem, performing different optimization loops to maximize our accuracy.

OVERVIEW


3-Class
Train-Test Split
SMOTE
MODEL TRAINING
...
MODEL SELECTION
Binary
Train-Test-Val Split
SMOTE
HYPER PARAMETERS OPTIMIZATION
...
MODEL TRAINING
...
MODEL SELECTION
on Train set
on Validation set per each model
on Train set, per each model
on Test set

DATA MODELING

3-Class Classification

01

- **Decision Tree**
- **Random Forest**
- **XGBoost**
- **MLP**

Binary Classification

02

- Decision Tree
- Logistic Regression
- Random Forest
- XGBoost
- MLP

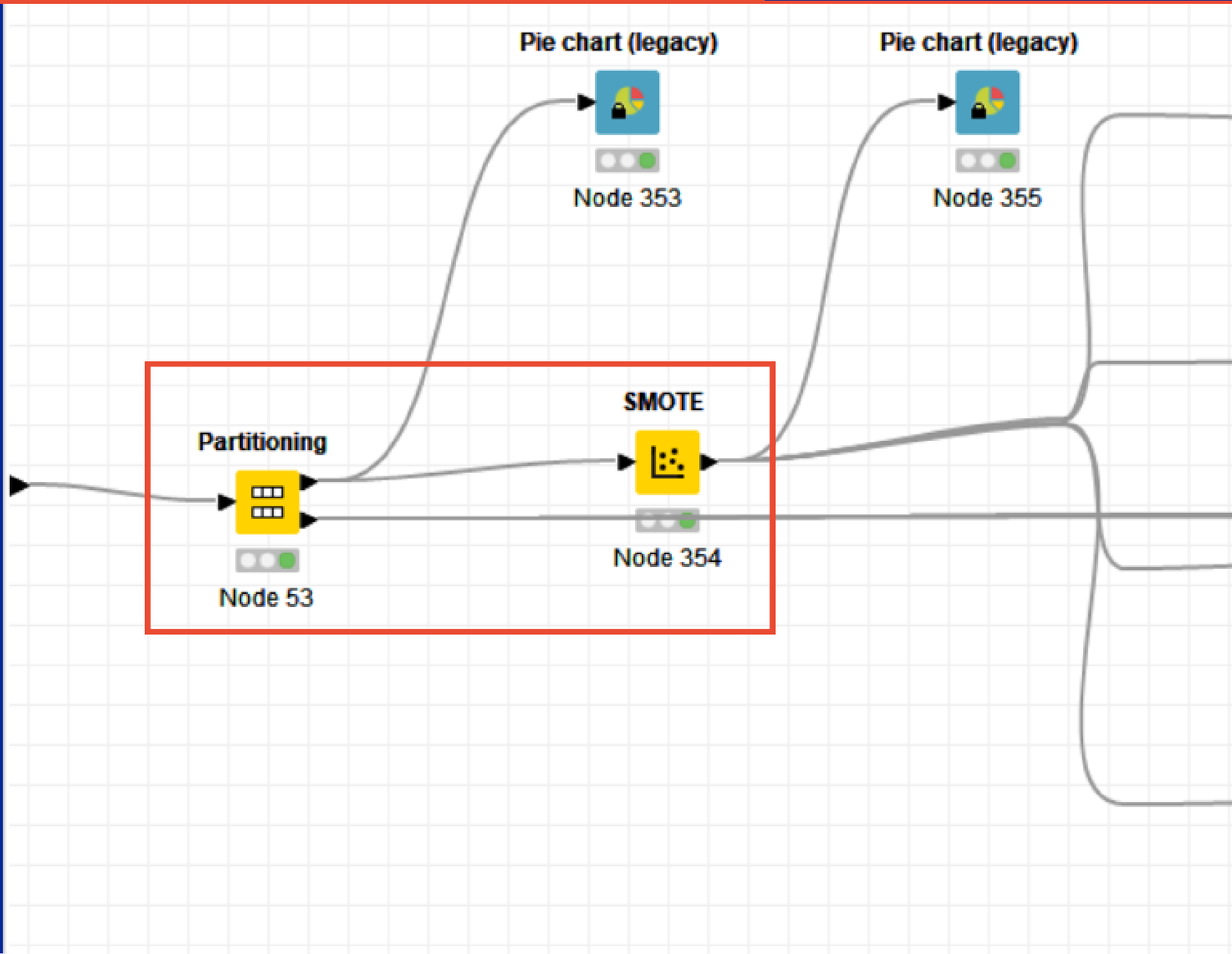
Super Learner

TRAIN–TEST SPLIT & SMOTE

3-Class Classification

- First, we used the node **Partitioning** to randomly split the data between train (70%) and test (30%).

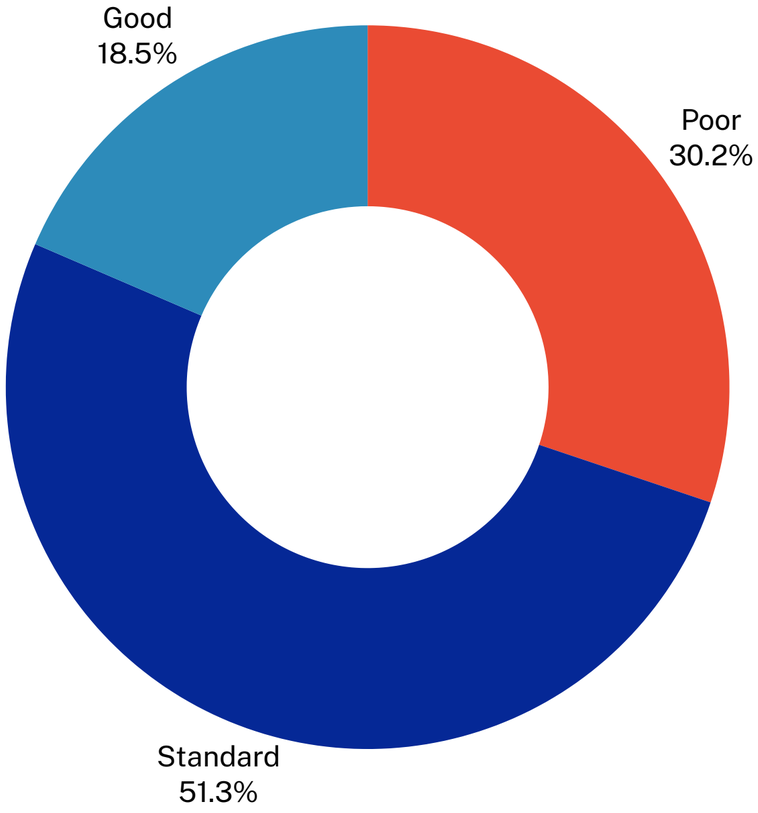
- Then, with the node **SMOTE**, we enlarged the train dataset, by creating new artificial rows. SMOTE works so that the output training dataset is balanced among target classes



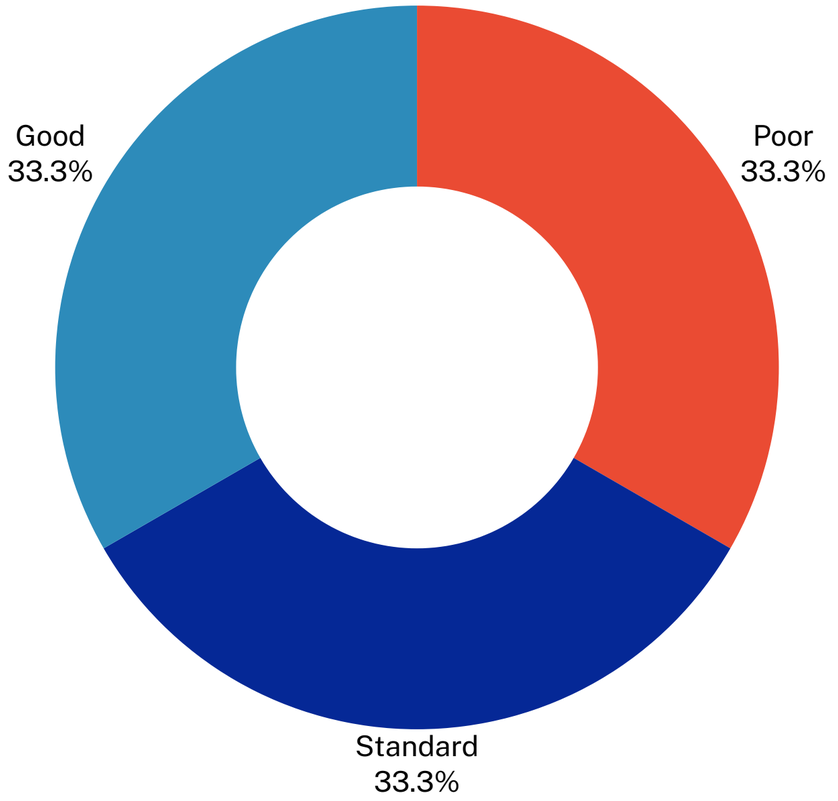
SMOTE

3-Class Classification

Credit_Score before **SMOTE**:



Credit_Score after **SMOTE**:



DECISION TREE

3-Class Classification

Model:

A decision tree is a model that categorizes data into distinct classes using a series of simple, hierarchical decision rules based on the features. Each branch represents a decision, and the leaf nodes signify the final class labels.

Parameters:

- **Quality measure**: Gain Ratio
- **Pruning method**: None
- **Min records per node**: 2

Credit Score \ Predicted Credit Score	Good	Standard	Poor
Good	297	307	85
Standard	367	1182	439
Poor	93	425	555

Model Accuracy:

55.24%

RANDOM FOREST

3-Class Classification

Model:

The Random Forest model is an ensemble learning method that combines multiple decision trees. Each tree in the forest is built from a random subset of the data and makes individual predictions, which are then voted upon for classification.

Parameters:

- **Min node size**: 1
- **Max tree depth**: None
- **Number of models**: 100

Credit Score \ Predicted Credit Score	Good	Standard	Poor
Good	433	235	21
Standard	291	1334	363
Poor	76	257	740

Model Accuracy:

66.86%

XGBOOST

3-Class Classification

Model:

XGBoost is a gradient boosting method that uses an esemble of decision tree. The algorithm sequentially corrects the mistakes of previous trees, combining weak learners to form a strong predictive model.

Parameters:

- **Max tree depth**: 4
- **Number of trees**: 100
- **Learning rate**: 0.1

Predicted Credit Score / Credit Score	Good	Standard	Poor
Good	412	249	28
Standard	270	1406	312
Poor	64	301	708

Model Accuracy:

67.36%

MULTI–LAYER PERCEPTRON

3-Class Classification

Model:

A Multilayer Perceptron (MLP) is a type of neural network with deep, fully connected layers of neurons, where each layer's output is the input for the next. It utilizes backpropagation, adjusting weights based on errors made in predictions.

Before running the model, we normalize the values with a Z-Score normalization.

Parameters:

- **Hidden Layers**: 2
- **Hidden Neurons**: 10
- **Epochs**: 100

Credit Score \ Predicted Credit Score	Good	Standard	Poor
Good	493	141	55
Standard	424	1038	526
Poor	109	161	803

Model Accuracy:

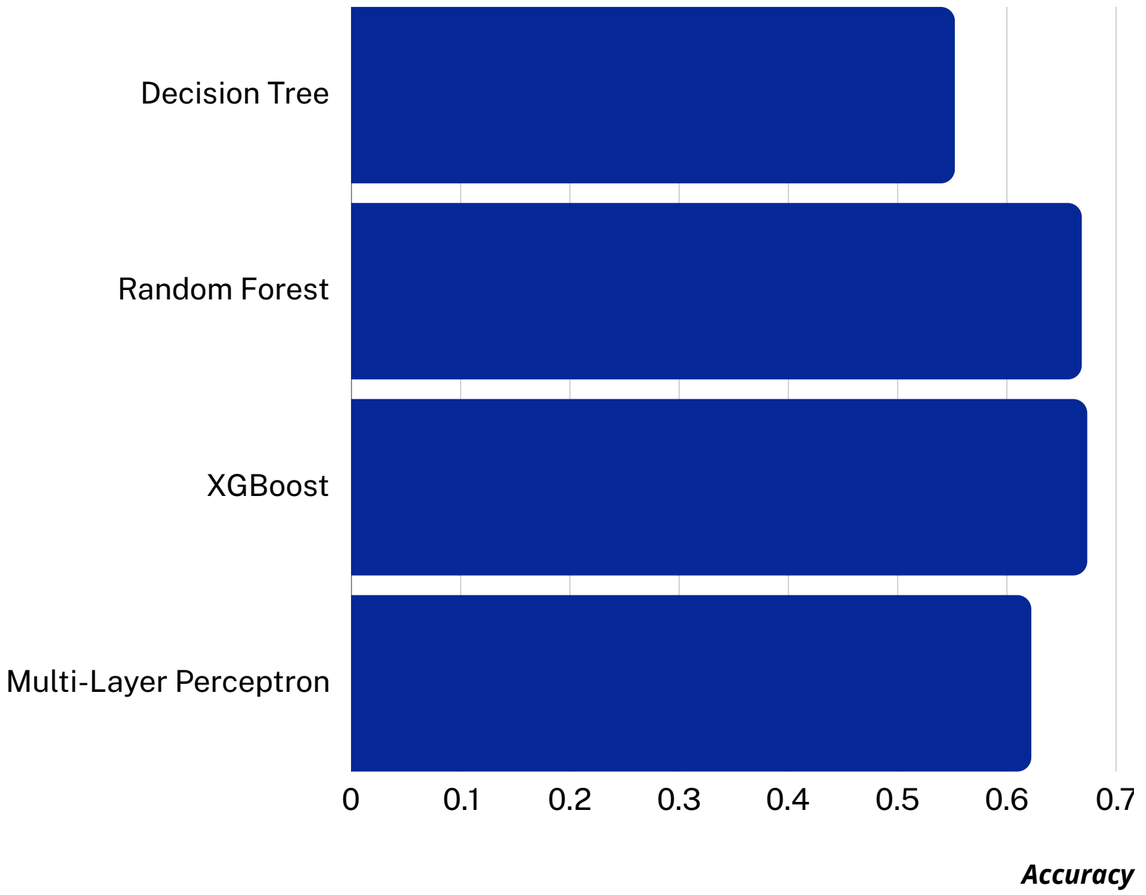
62.24%

MODEL COMPARISON

3-Class Classification

We come to the conclusion that the tree ensembles have the best accuracy within the trained models, with the XGBoost model being slightly better than the Random Forest.

However, we decided not to dive deeper into the analysis as this multi-class classification was not our main goal. We limited ourselves to analyze accuracy and AUC, while for the Binary Models we will focus on other metrics as well and perform some more thorough hyperparameter tuning, in order to have better results to advise managers and institutions.



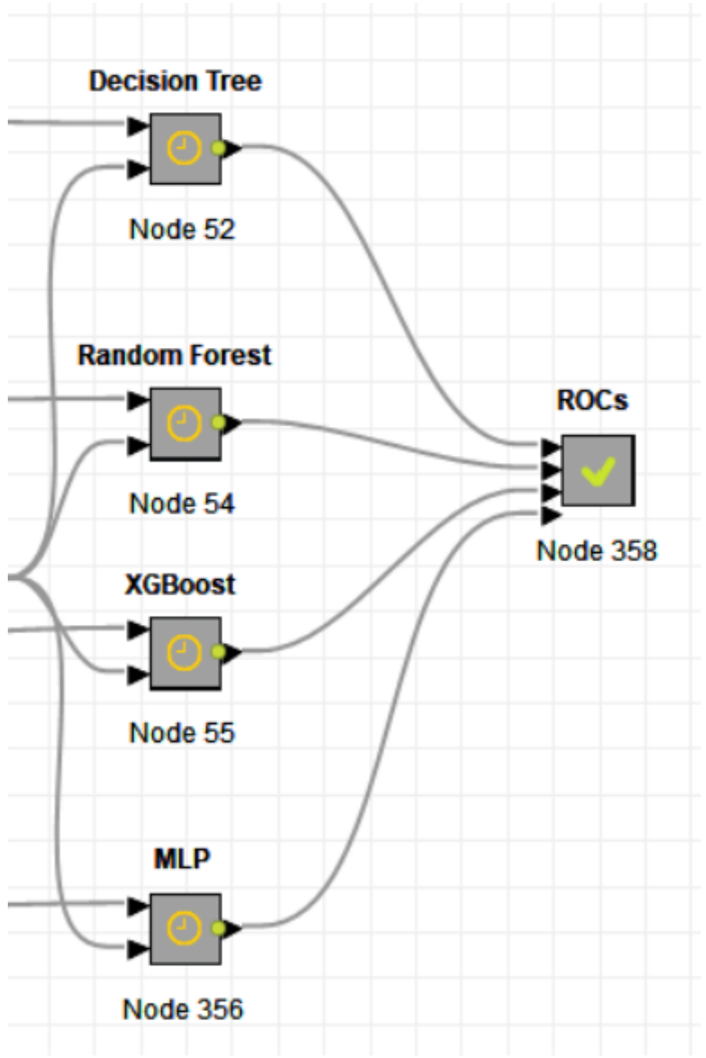
OvR ROCs

3-Class Classification

The One-vs-the-Rest (OvR) is a strategy to evaluate a multiclass classification model performance.
Also known as one-vs-all, consists in computing a ROC curve per each of the n_classes.

At each step, a given class is regarded as the positive class and the remaining classes are regarded as the negative class in bulk.

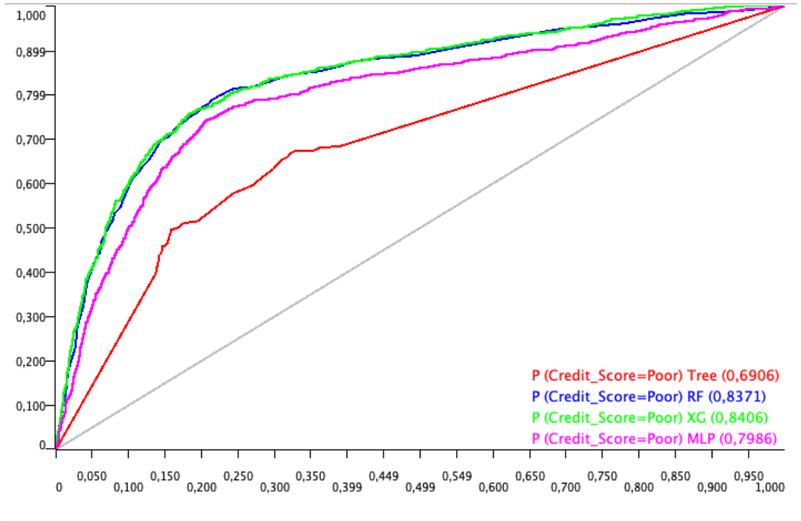
To do so, we collected all the predicted probabilities for each final trained model and we plot the ROC curves.



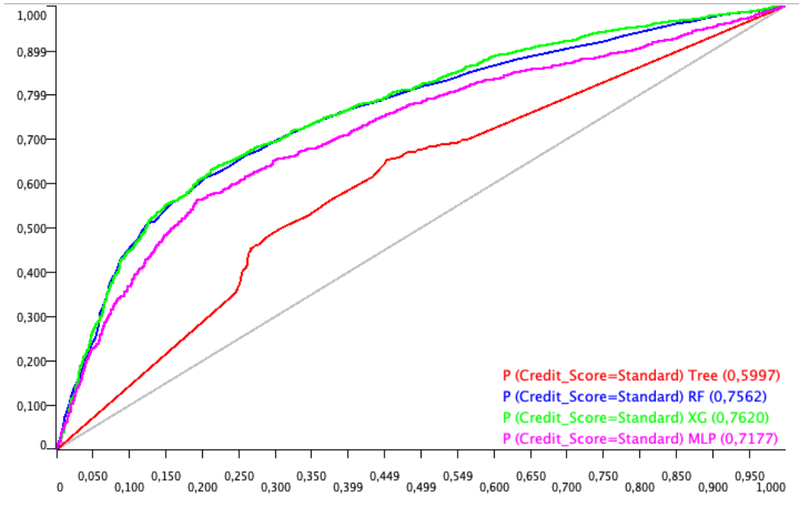
OvR ROCs

3-Class Classification

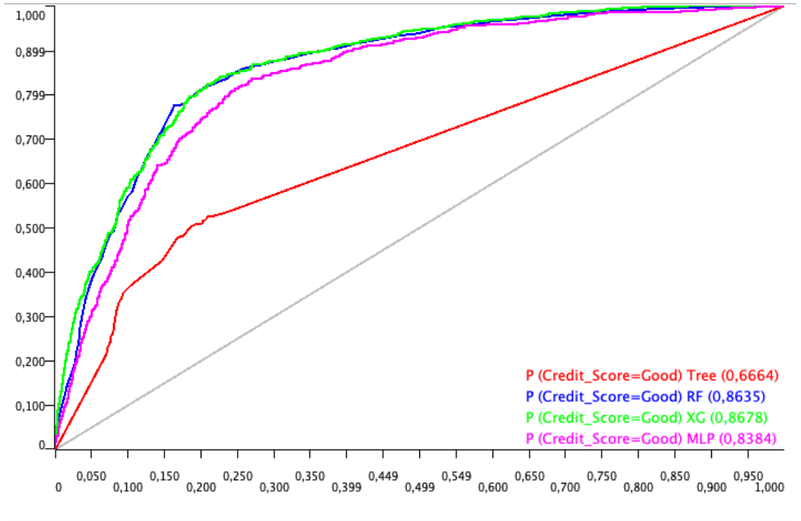
Poor vs the Rest



Standard vs the Rest

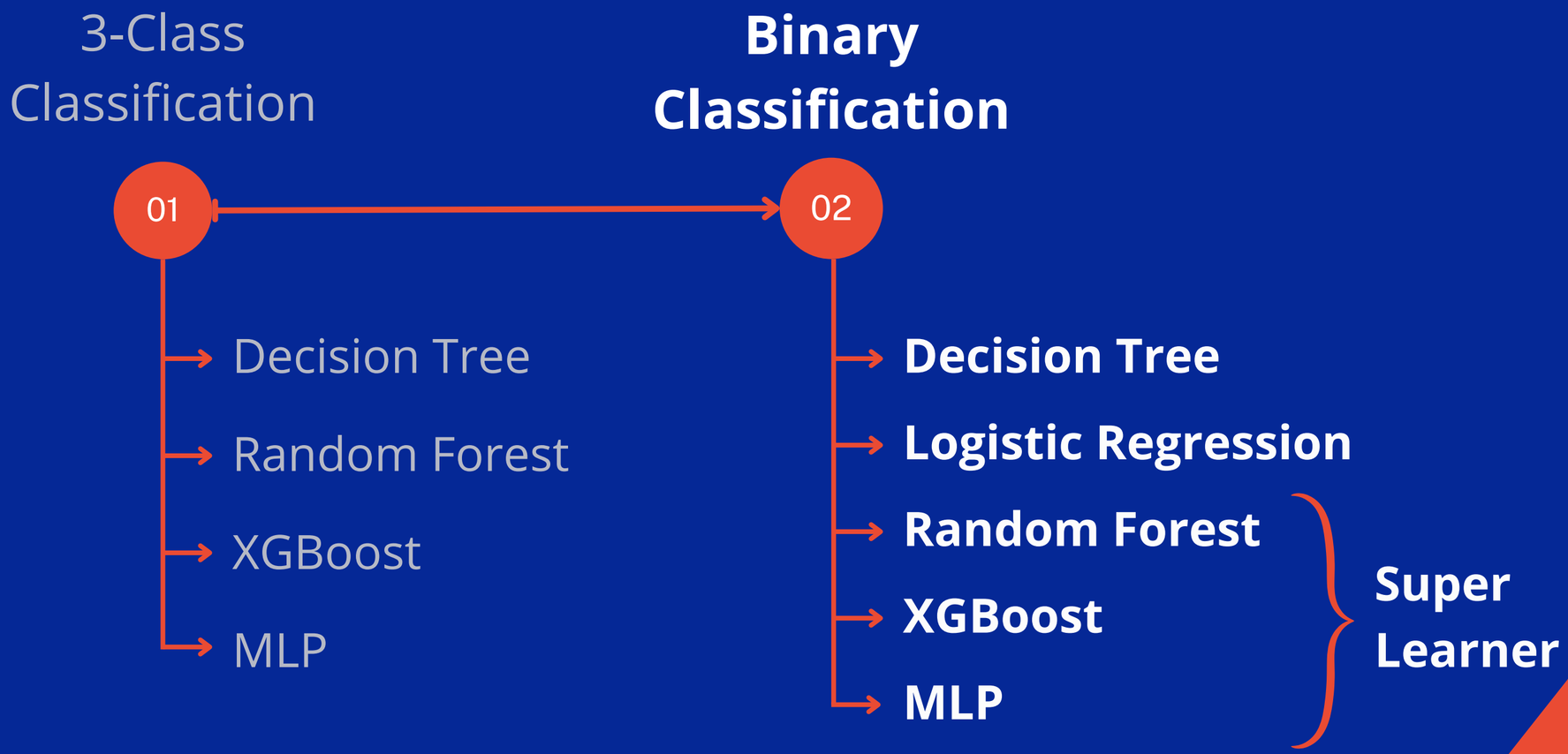


Good vs the Rest



Overall, XGBoost is the most predictive models for each class, with higher Area Under the Curve in every scenario. Random Forest performs similarly to XGB, while Decision Tree has the lower AUCs in each case. MLP is always the second to last, this suggest there might be room for improvement as it is widely known ANNs require lots of tuning in order to be optimized. However, our results are in line with literature, suggesting tree-based models are in general the state-of-the-art for tabular data, with ensembles clearly better than base model.

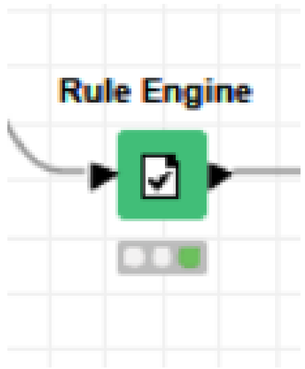
DATA MODELING



3–Class to Binary

Binary Classification

We used a **Rule Engine** node to cast the multi-class into a binary classification problem, combining “Good” and “Standard” into a unique “Good or Standard” class.



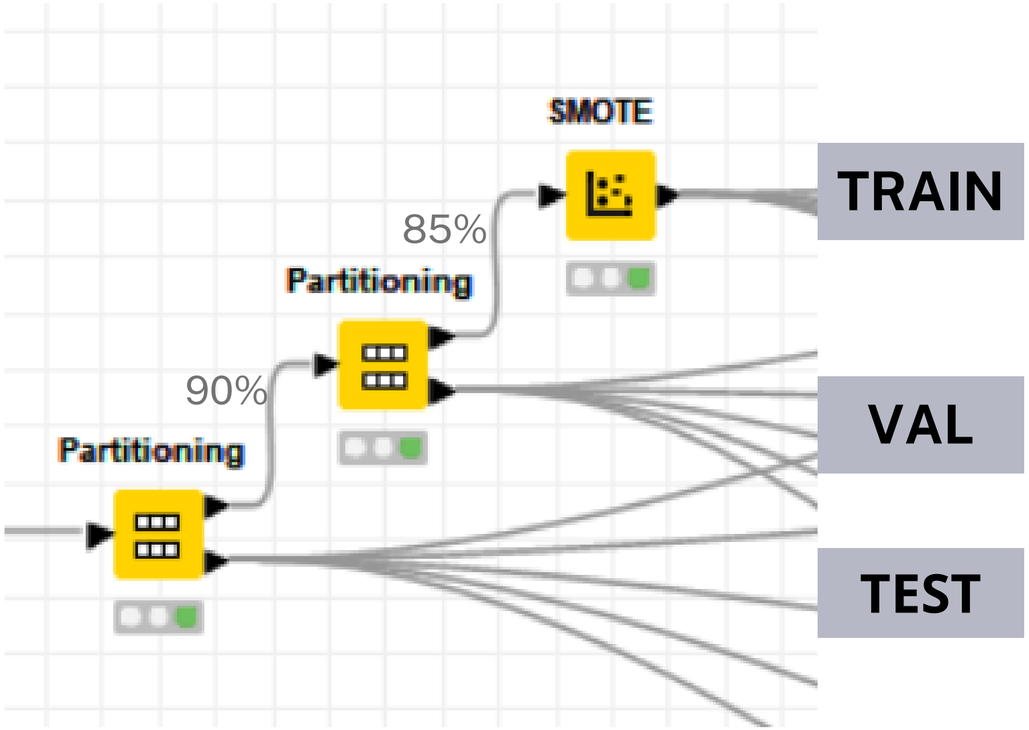

```
$Credit_Score$ = "Standard" => "Good or Standard"
$Credit_Score$ = "Good" => "Good or Standard"
$Credit_Score$ = "Poor" => "Poor"
```

TRAIN–TEST–VAL SPLIT & SMOTE

Binary Classification

- We used the node **Partitioning** to randomly split the data between
  - Train (76.5%)
  - Validation (13.5%)
  - Test (10%).

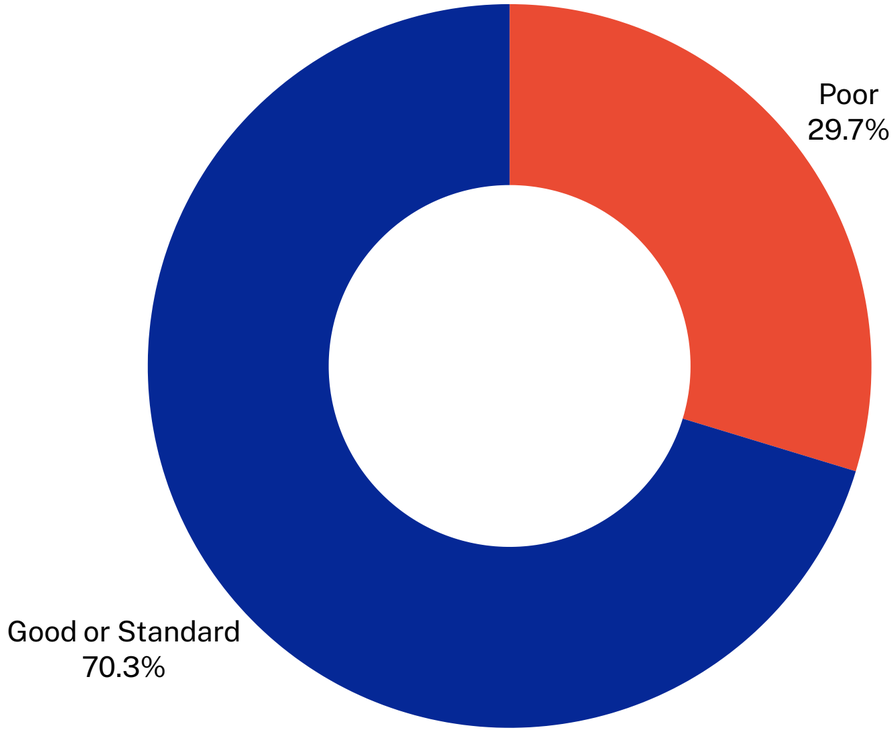
- We balanced the train dataset using **SMOTE**



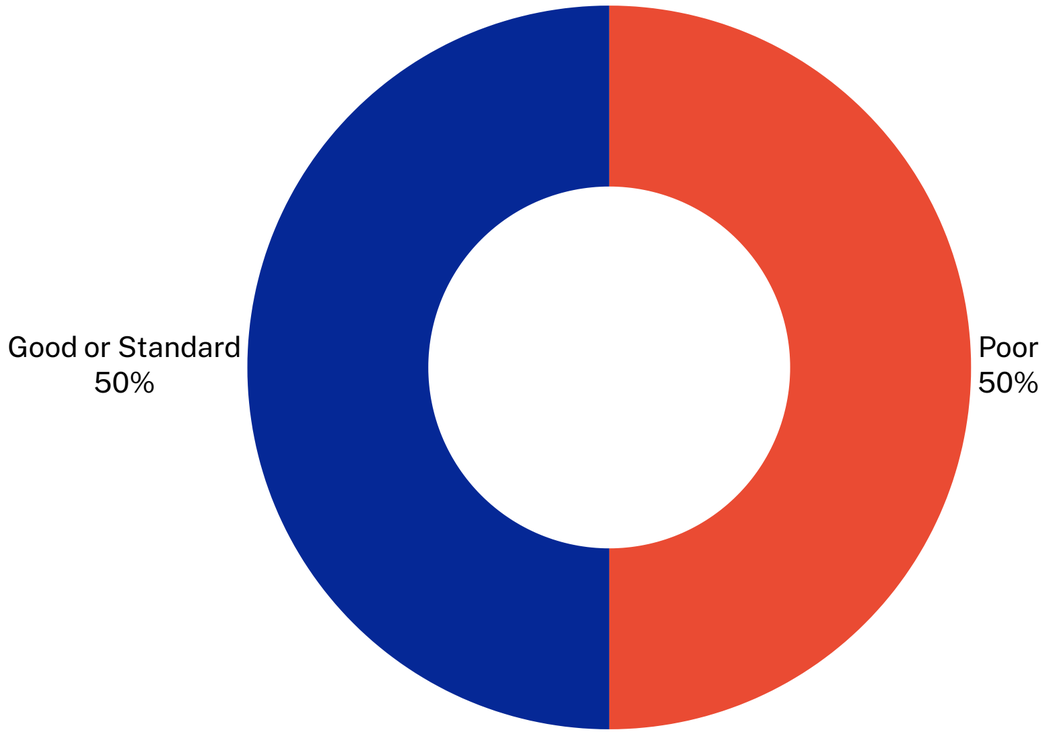
SMOTE

Binary Classification

Credit_Score before **SMOTE**:



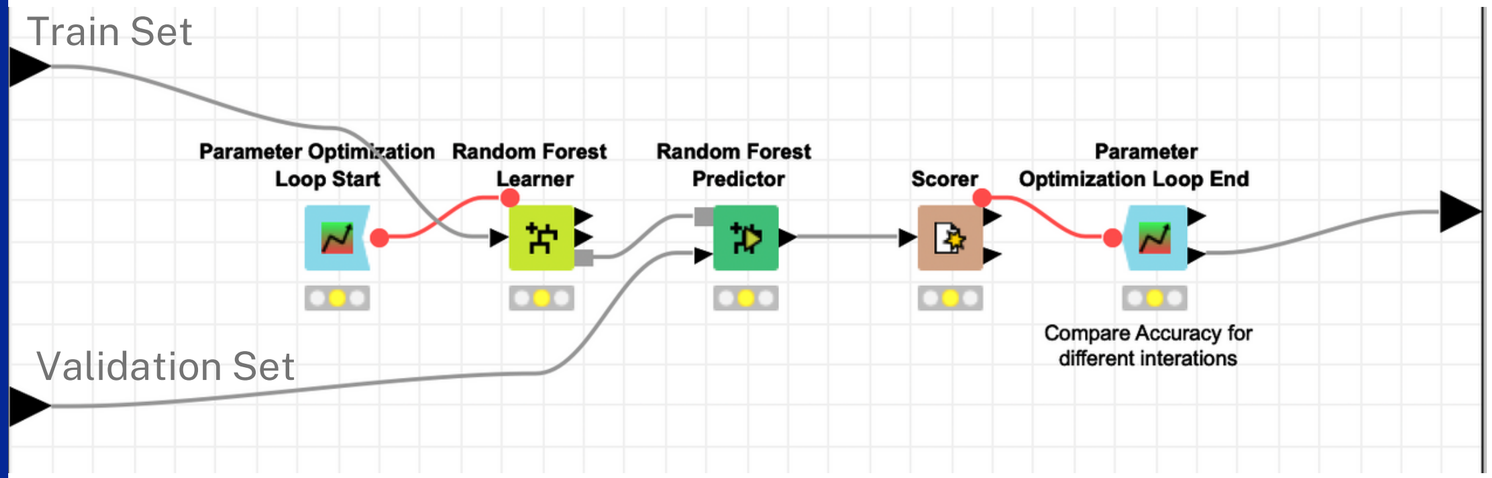
Credit_Score after **SMOTE**:



HYPER–PARAMETER OPTIMIZATION

Binary Classification

- In order to find the best hyper-parameters, we performed a parameter optimization loop.
- For each model, we defined a grid of hyper-parameters to be tested.
- Then, for each combination of parameters, we compare the performances of the trained model.
- Finally, the model is trained with the best hyper-parameters.



Exaple of optimization loop for Random Forest

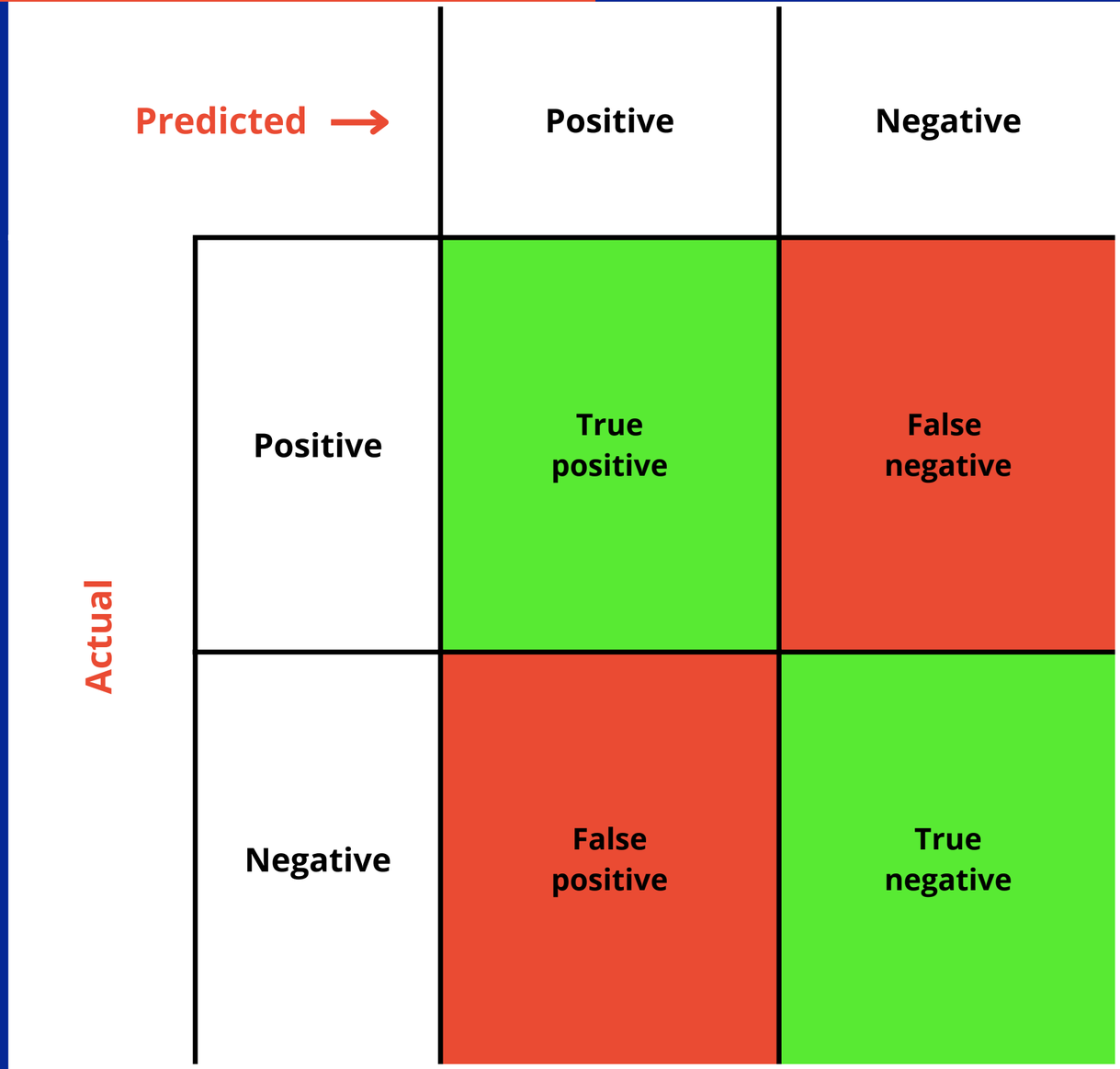
OBJECTIVE METRICS – *Sensitivity*

Binary Classification

In our problem, positive (1) means **poor credit score**. As in all kind of anomaly detection problems, our goal is to find as much positives as possible out of all positives. Athe number of customer predicted as good credit score but who actually have a poor credit score. We shall maximize sensitivity:

Sensitivity = TP / (TP + FN)

In this way, we maximize the number of poor credit scores detected and minimize the number of poor credit scores missed.



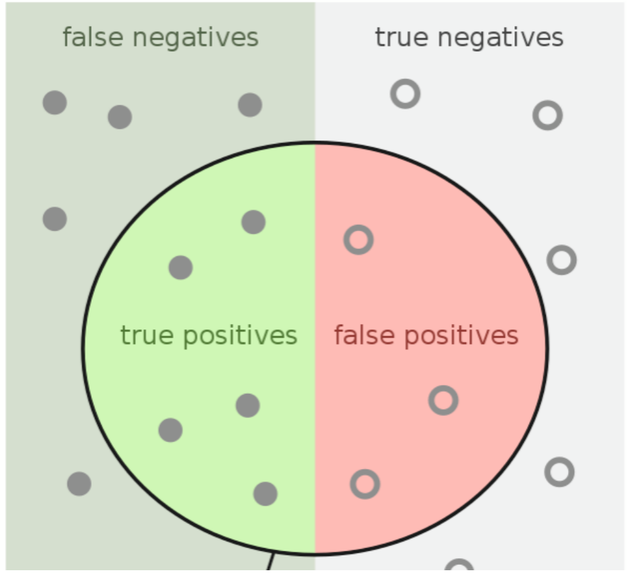
OBJECTIVE METRICS – *Specificity*

Binary Classification

Another important metric we should keep into consideration is *Specificity*, that represents the goodness of our model in classifying negative labelled records.

As we said, bank mostly care about correctly labelling poor credit seeker, as these are the riskier in terms of losses.

However, correctly labelling good creditors as a return regarding profits as well, as a correct evaluation gives the bank the possibility to give lower interest rates, therefore making offers from the institution more attractive for customers.



$$\text{Sensitivity} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

$$\text{Specificity} = \frac{\text{true negatives}}{\text{true negatives} + \text{false positives}}$$

OBJECTIVE METRICS – *Youden Index*

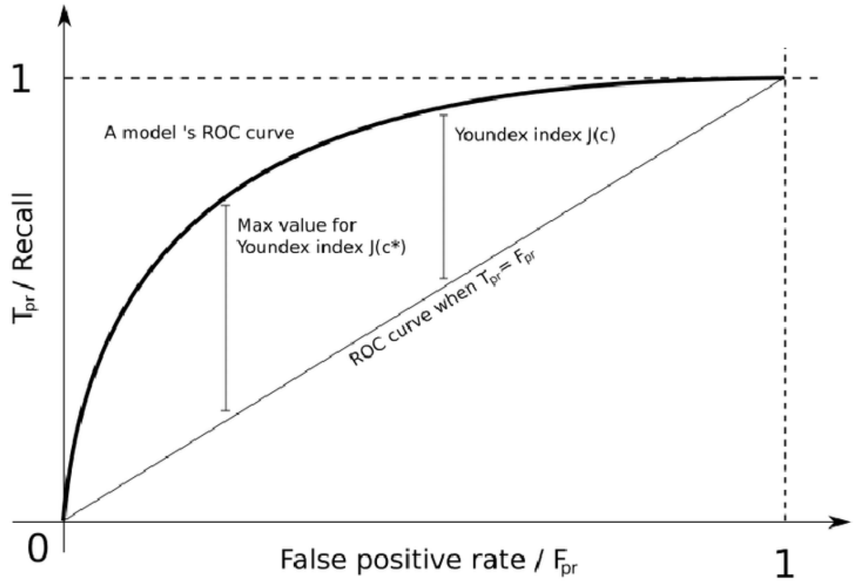
Binary Classification

Also called Youden J, this metric acts as a summary for the goodness of a model, and it helps when it is necessary to quickly understand the predictive power of a process. It incorporates both specificity and sensitivity and it goes from 0 to 1, with the maximum being reached when a model correctly classify all records proposed in a test.

This is obviously an utopic scenario, but keeping this metrics around is useful for explanatory purposes, as it is easily interpretable also for non-technical stakeholders.

$$J = sensitivity + specificity - 1$$

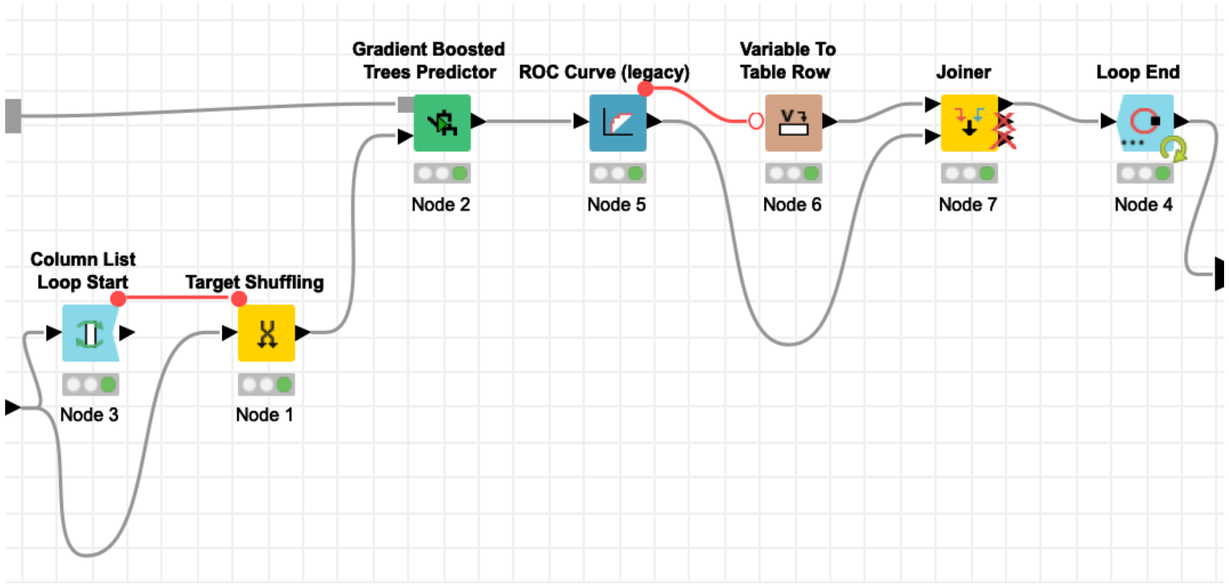
$$= \frac{TP}{TP + FN} + \frac{TN}{TN + FP} - 1$$



FEATURE IMPORTANCE

Binary Classification

In order to understand which features are considered more important by our model we implemented a node to compute **Permutation Feature Importance** (**PFI**).



This approach was implemented by following the findings of Breiman, L. Random Forests. Machine Learning 45, 5–32 (2001) and Fisher, Aaron, Cynthia Rudin, and Francesca Dominici. "All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously." J. Mach. Learn. Res. 20.177 (2019): 1-81.

PFI calculates the impact of a certain feature on overall predictive power of a model by comparing a performance metric of choice (in our case AUC) of the model on the original dataset and on a dataset where one feature has been shuffled. This processes is repeated for all the features. The higher the difference between the two, the more important the shuffled feature is.

$$PFI_i = AUC - AUC_i$$

Where *i* indicates the shuffled feature

DECISION TREE

Binary Classification

Best performing parameters:

Quality Measure: Gain Ratio
Pruning method: No pruning
Minimum n° of records per leaf: 5

We decided to only use the minimum number of samples per leaf as stopping criterion, without setting the maximum number of levels.

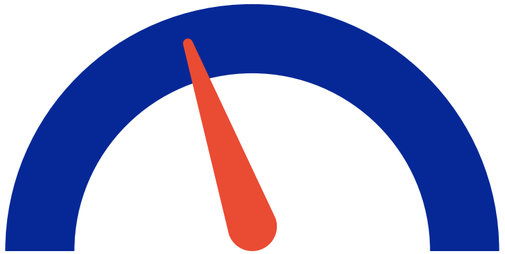
We set the classification threshold to P (poor) = **0.190** in order to maximize Youden's Index.

Prediction / Actual	Good or Standard	Poor
Good or Standard	593	306
Poor	90	261

Model Sensitivity: 74.35%

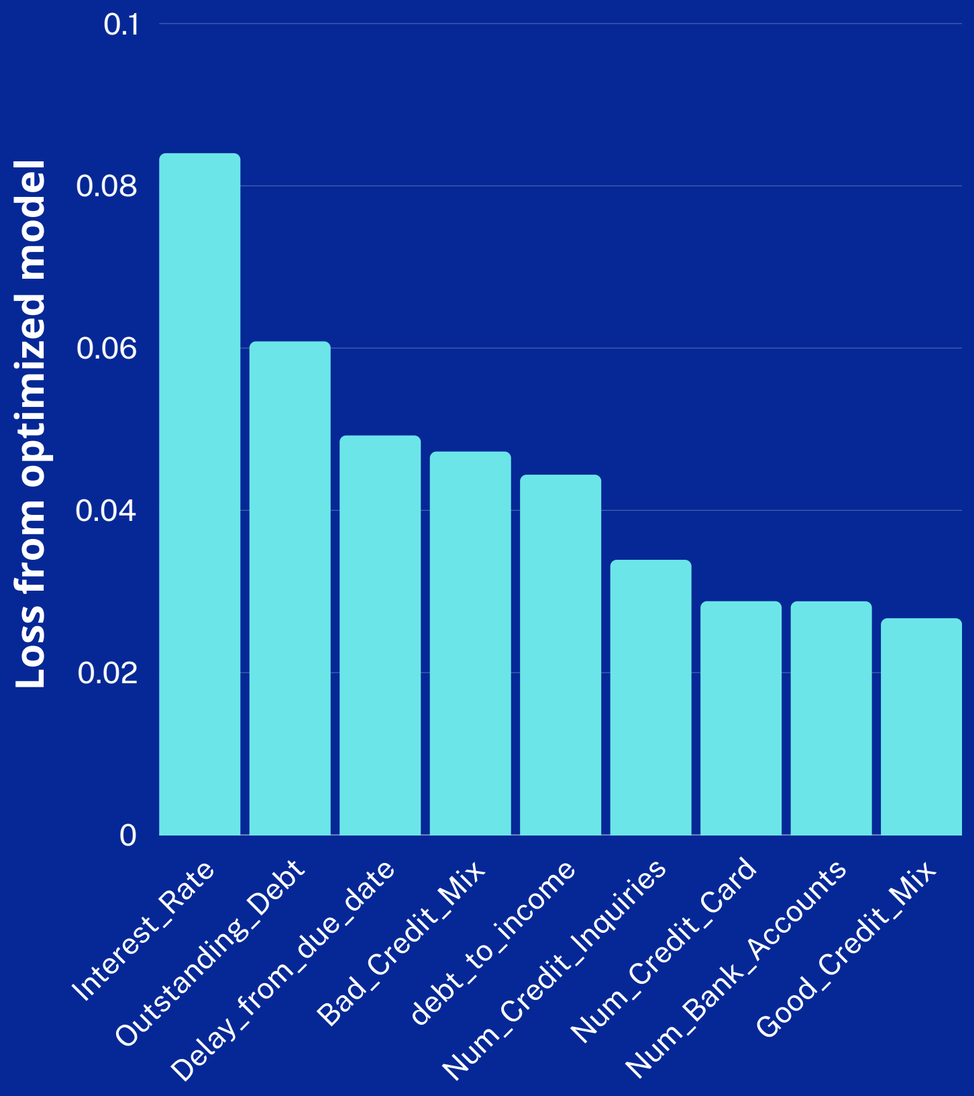
Model Specificity: 65.96%

Youden's Index: 0.403



DECISION TREE – PFI

Binary Classification



Interest rate: individuals with higher credit scores are offered lower interest rates because they are considered less risky to lenders. Indeed, this is the most (negatively) correlated feature with the credit score.

Outstanding_debt: it will have a negative influence on the credit score, particularly if the individual has a large amount of credit already.

Delay from due date: as a delay from due date increases, there could be a higher risk of late payments negatively impacting credit scores.

LOGISTIC REGRESSION

Binary Classification

First of all, we normalized the input through **Z-score normalization**, which is good practice when using this kind of algorithm.

Then, given the high number of features in our dataset, we applied **L1** (Lasso) **regularization** to discard the most irrelevant features.

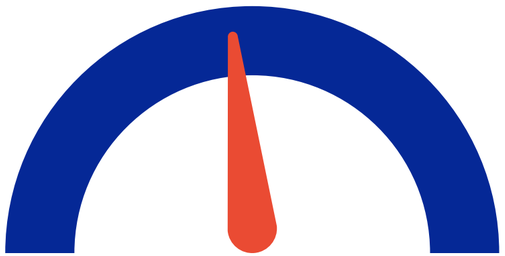
Finally, we set the threshold to P (poor) = **0.531** in order to maximize Youden's Index.

Prediction / Actual	Good or Standard	Poor
Good or Standard	713	186
Poor	113	238

Model Sensitivity: 67.81%

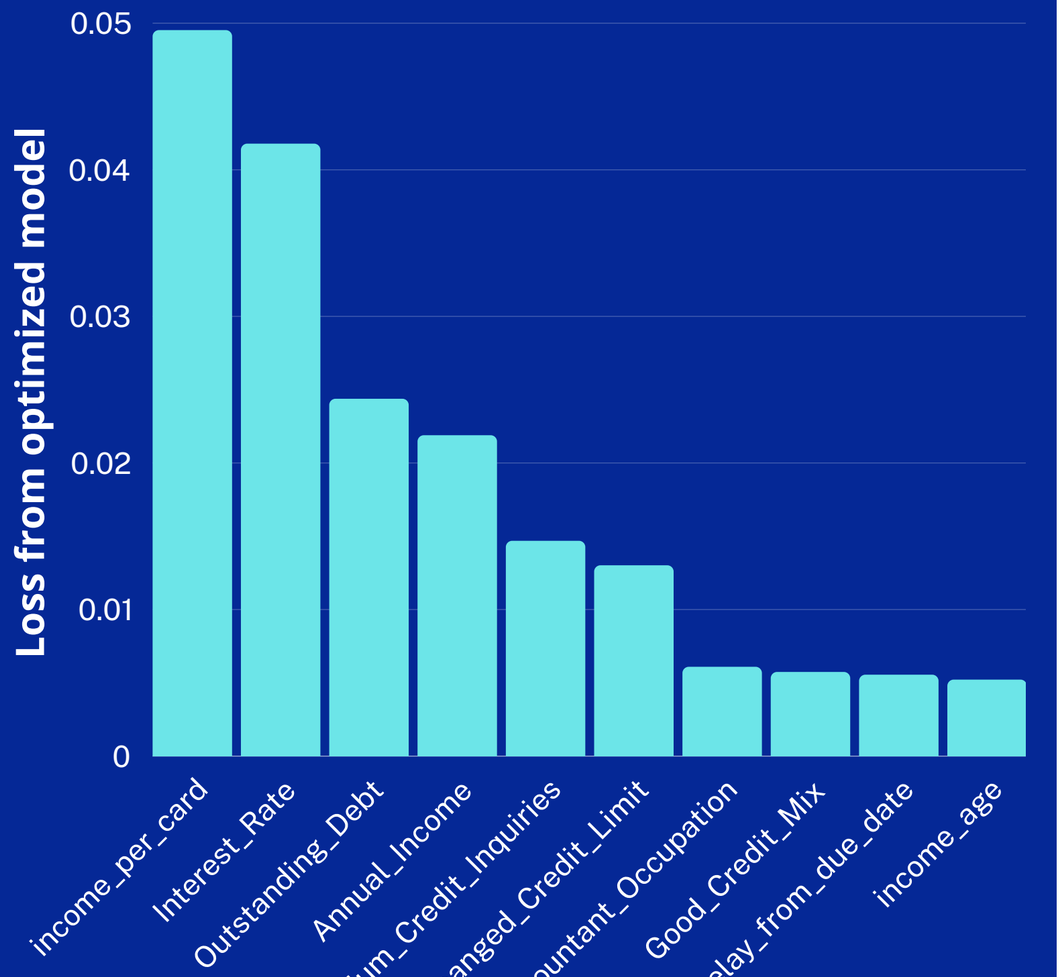
Model Specificity: 79.31%

Youden's Index: 0.471



LOGISTIC REGRESSION – PFI

Binary Classification



Income per card: this suggests that the income associated with each credit card account is a strong indicator of credit score. We expect this feature to impact positively the target, in line with previous findings from linear correlations.

Interest rate: individuals with higher credit scores are offered lower interest rates because they are considered less risky to lenders. Indeed, this is the most (negatively) correlated feature with the credit score.

Outstanding_debt: it will have a negative influence on the credit score, particularly if the individual has a large amount of credit already. These results are in line with the linear correlation evidenced by the heatmap in the bivariate analysis.

RANDOM FOREST

Binary Classification

Best performing parameters:

Splitting Criterion: Gain Ratio
Tree Depth: None
Minimal node size: 1
Number of trees: 100

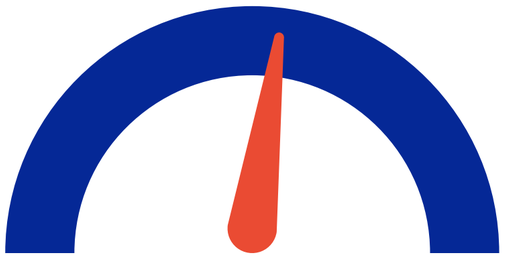
The tree depth is not limited, while the number of trees, which represents the number of base learners, is 100. We could increase this number without incurring in overfitting, but it would be computationally expensive.

Prediction / Actual	Good or Standard	Poor
Good or Standard	697	202
Poor	84	267

Model Sensitivity: 76.06%

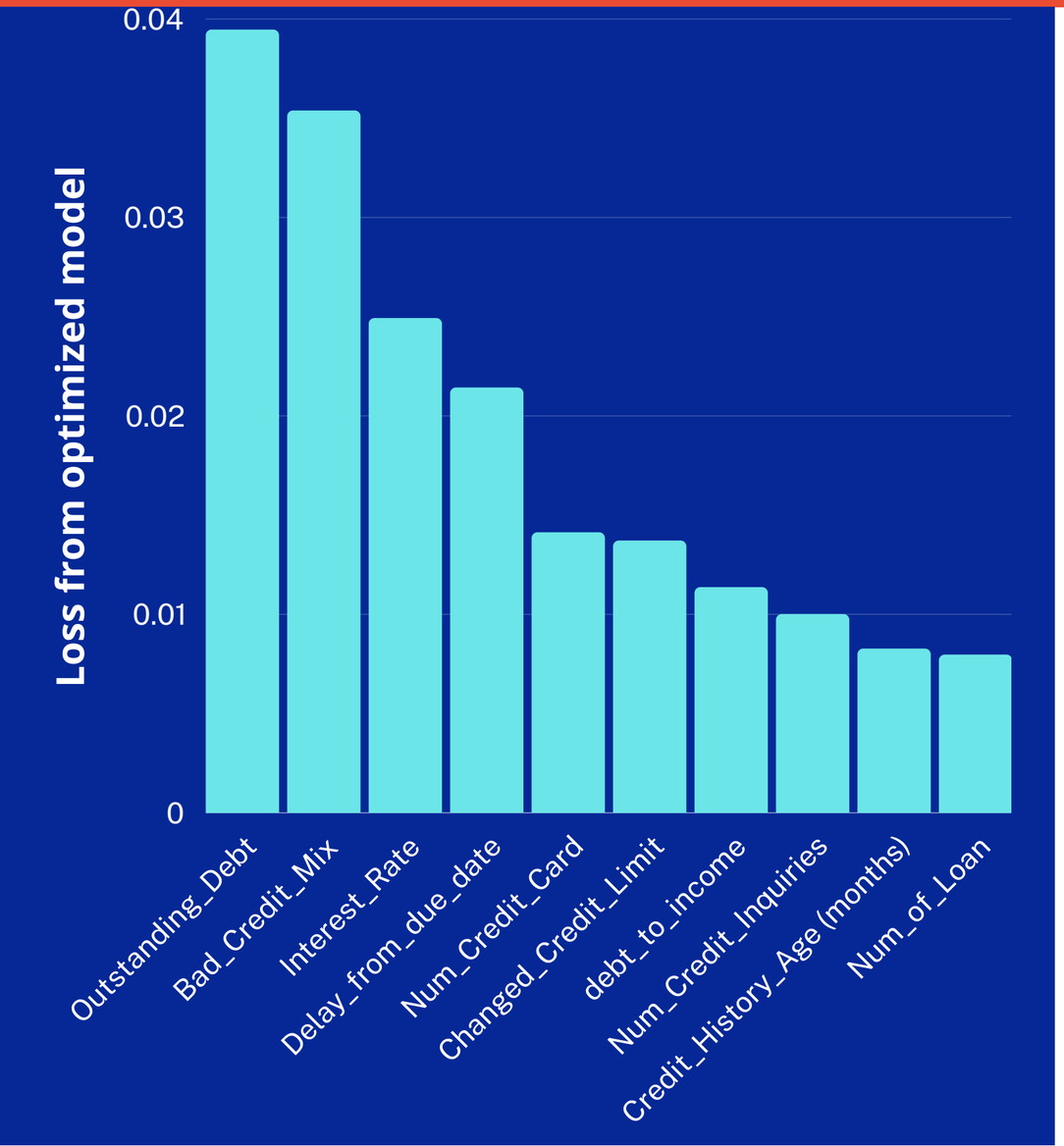
Model Specificity: 77.53%

Youden's Index: 0.535



RANDOM FOREST – PFI

Binary Classification



Outstanding Debt is the most relevant feature according to PFI, as it was relevant for previous models.

Bad Credit Mix: credit mix reflects the diversity of credit accounts, including credit cards, mortgage loans and other characteristics. A bad credit is negatively correlated with the credit score, and results in one of the most predictive features.

Interest rate: as explained before, individuals with higher credit scores are offered lower interest rates because they are considered less risky to lenders.

XGBOOST

Binary Classification

Best performing parameters:

Max Tree Depth: 7
Minimal node size: 1
Number of trees: 100
Learning rate: 0.1

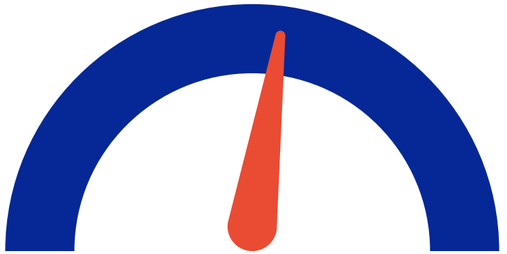
We set the threshold to P (poor) = **0.283** in order to maximize Youden's Index.

Prediction / Actual	Good or Standard	Poor
Good or Standard	685	214
Poor	77	274

Model Sensitivity: 78.06%

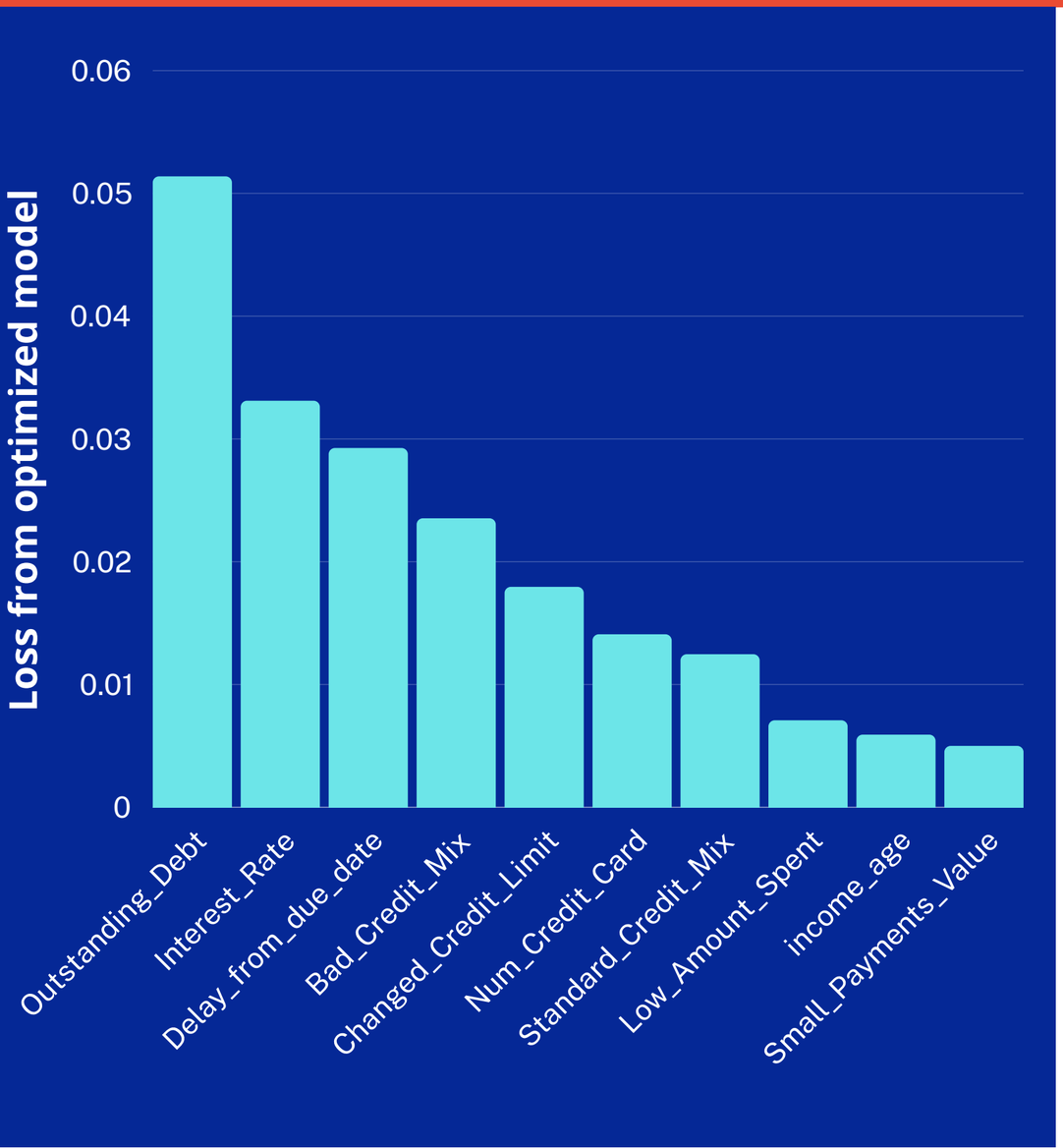
Model Specificity: 76.20%

Youden's Index: 0.542



XGBOOST – PFI

Binary Classification



Just like in previous models, **outstanding debt** is present as the most informative feature on the goodness of a credit seeker. Also the two lower steps of the podium, **interest rate** and **delay from due date** were already present in top positions for previous models.

A new presence is **Changed credit limit**, indicating whether the customer raised his credit card limits: this is a behavior linked tightly with poor management of credit.

MULTI–LAYER PERCEPTRON

Binary Classification

Best performing parameters:

Number of Epochs: 100
Hidden Layers: 10
Hidden Neurons: 10

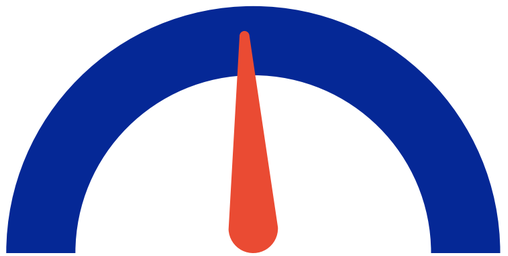
We set the threshold to P (poor) = **0.249** in order to maximize Youden's Index.

Prediction / Actual	Good or Standard	Poor
Good or Standard	673	226
Poor	97	254

Model Sensitivity: 72.36%

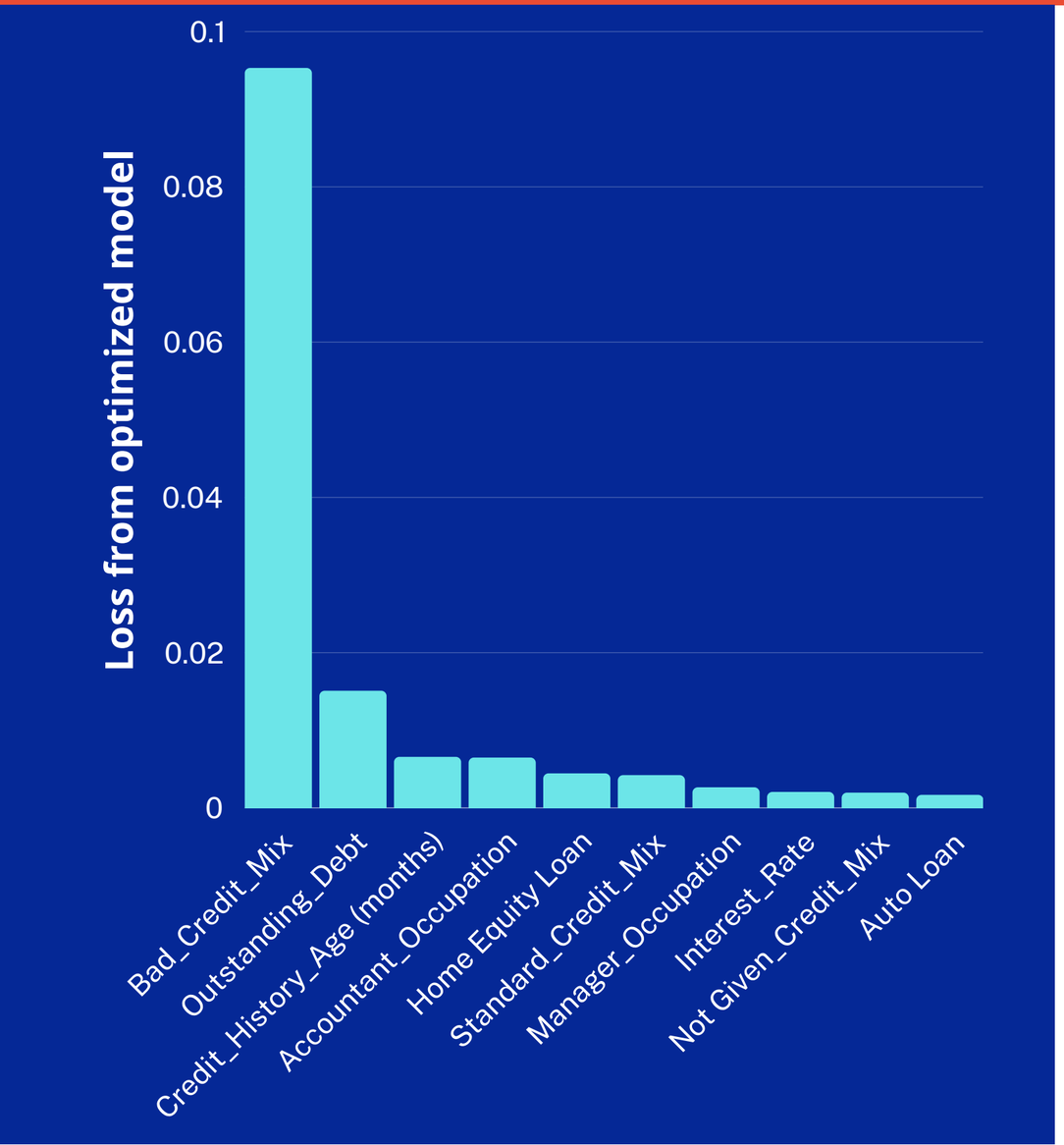
Model Specificity: 74.86%

Youden's Index: 0,472



MULTI–LAYER PERCEPTRON – PFI

Binary Classification



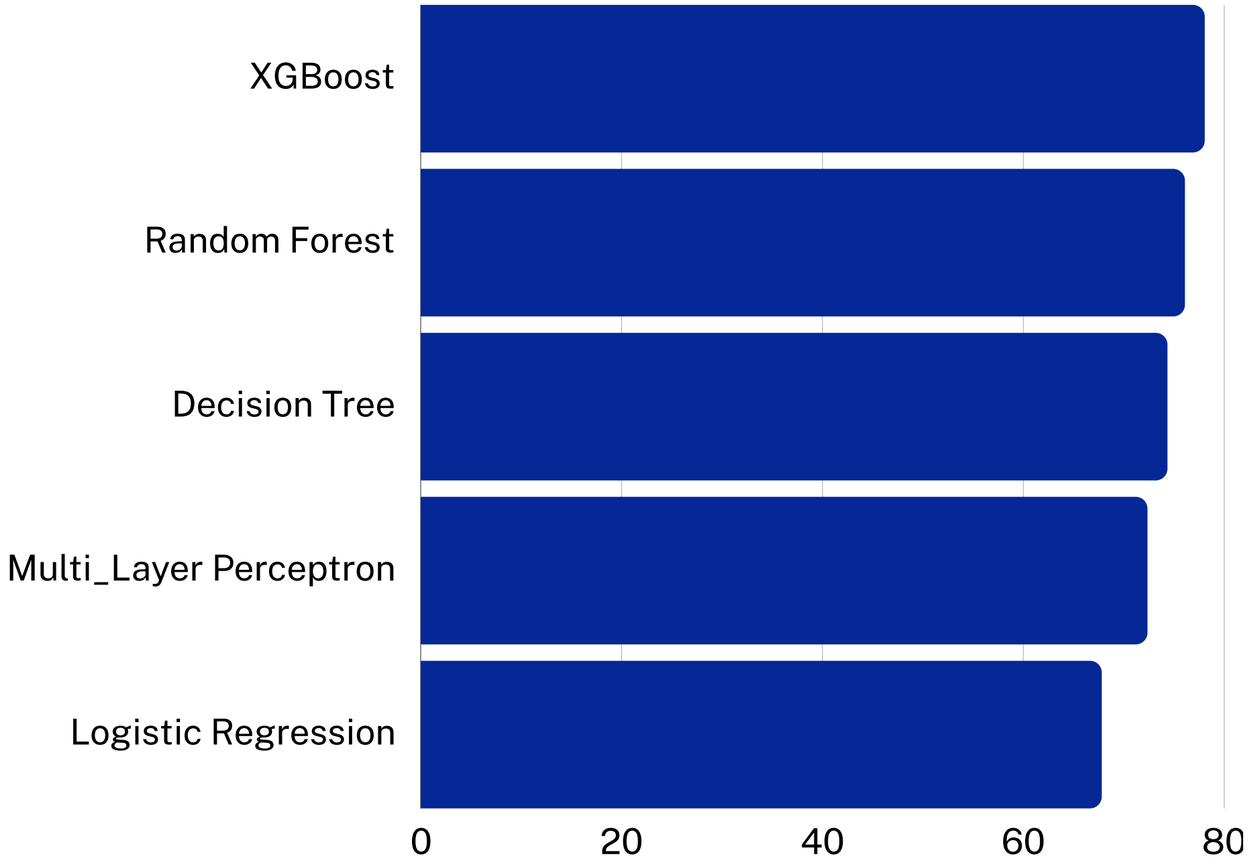
In the MLP model, according to the PFI, the most relevant feature is the classification of the customer as a **Bad credit mix** holder. Notice how strongly the model relies on this feature, producing an error nearly 5x bigger than the second biggest.

After that, we find some common names in these rankings, such as **Outstanding debt** and **Credit history length**, as well as other categories of **Credit Mix**.

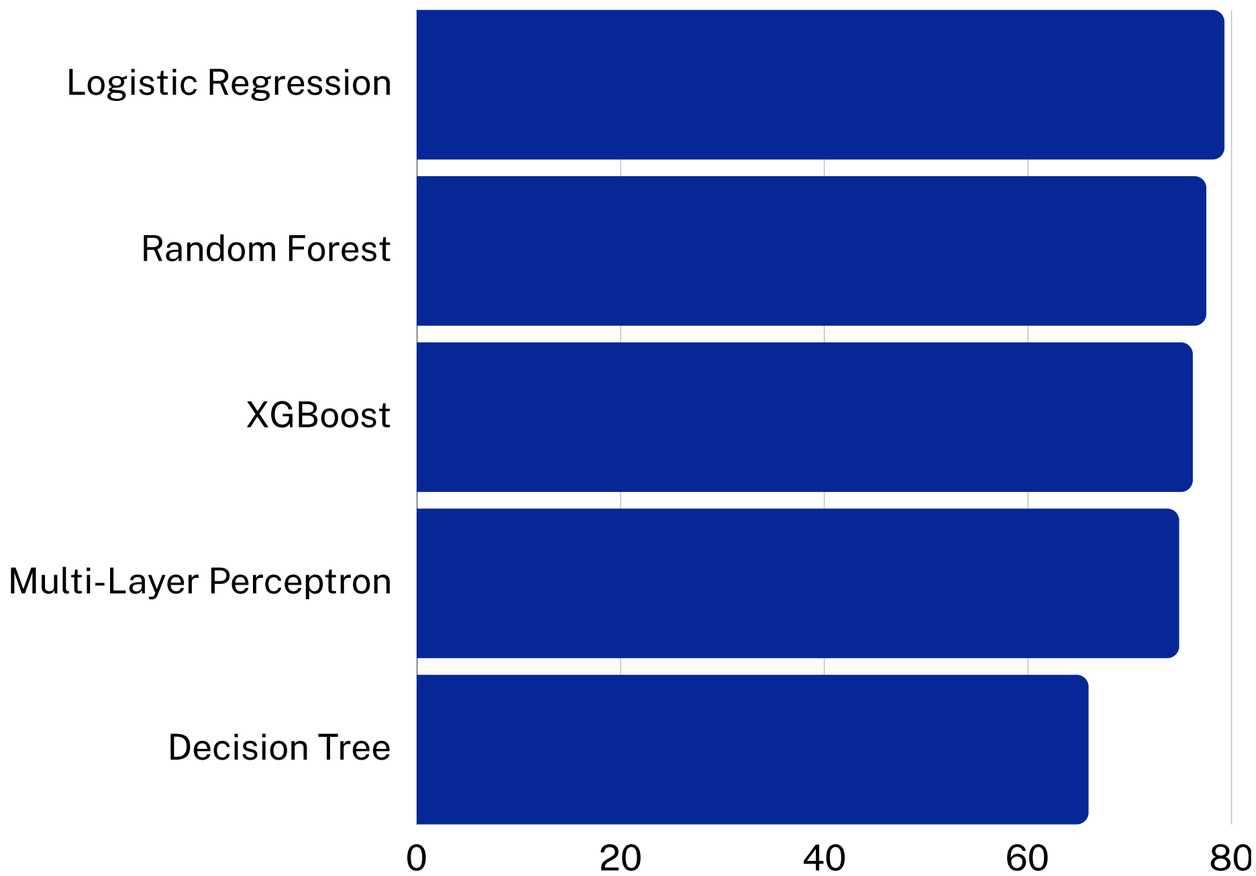
MODELS COMPARISON

Binary Classification

Sensitivity



Specificity

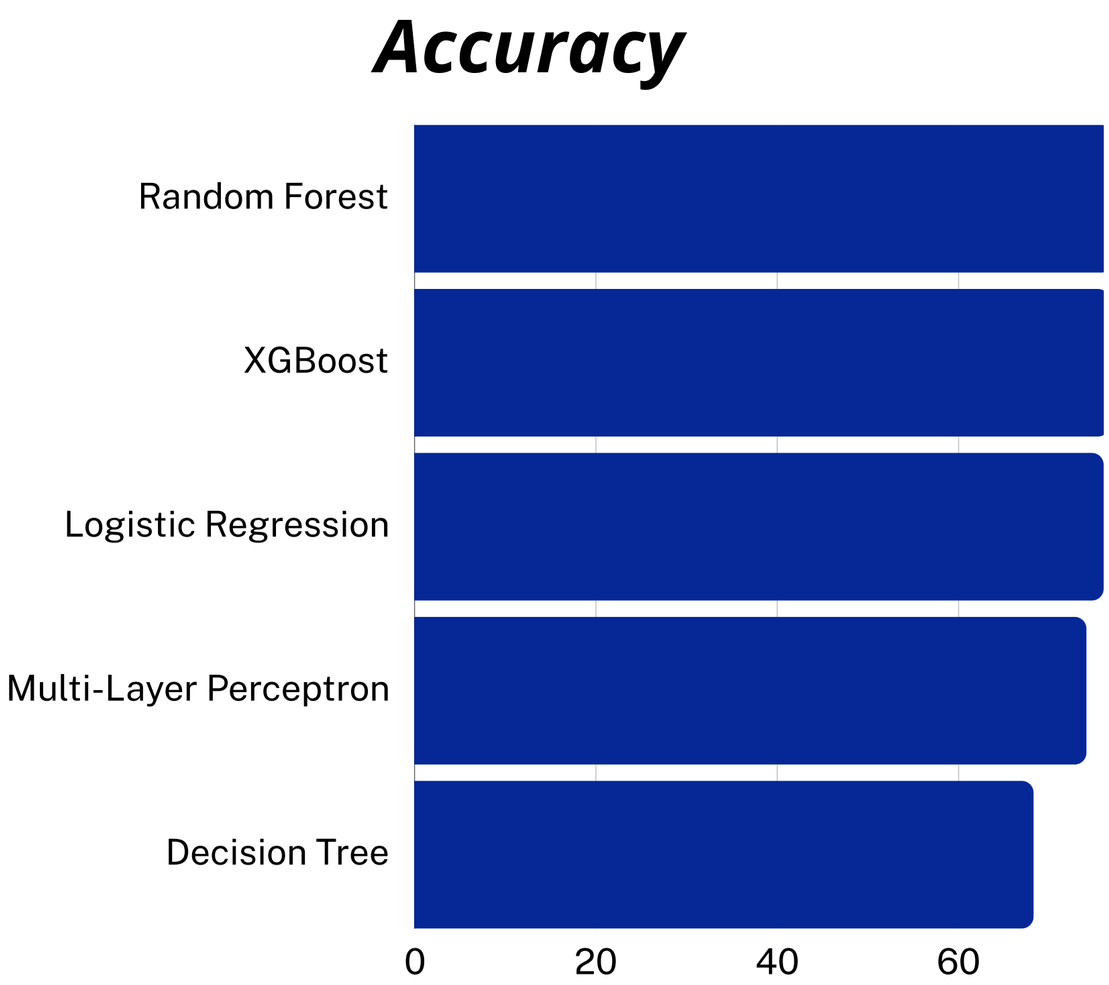


MODELS COMPARISON

Binary Classification

From these 3 metrics we see how the ensemble models are in general the best performing ones. In fact, we can notice how Random Forest and XGBoost are in the top part of the ranking for all metrics.

These models aggregate predictions from multiple models, reducing variance and bias, and hence improving accuracy. They effectively capture complex relationships in the data by combining the strengths of individual models, leading to robustness against overfitting and better generalization on unseen data.



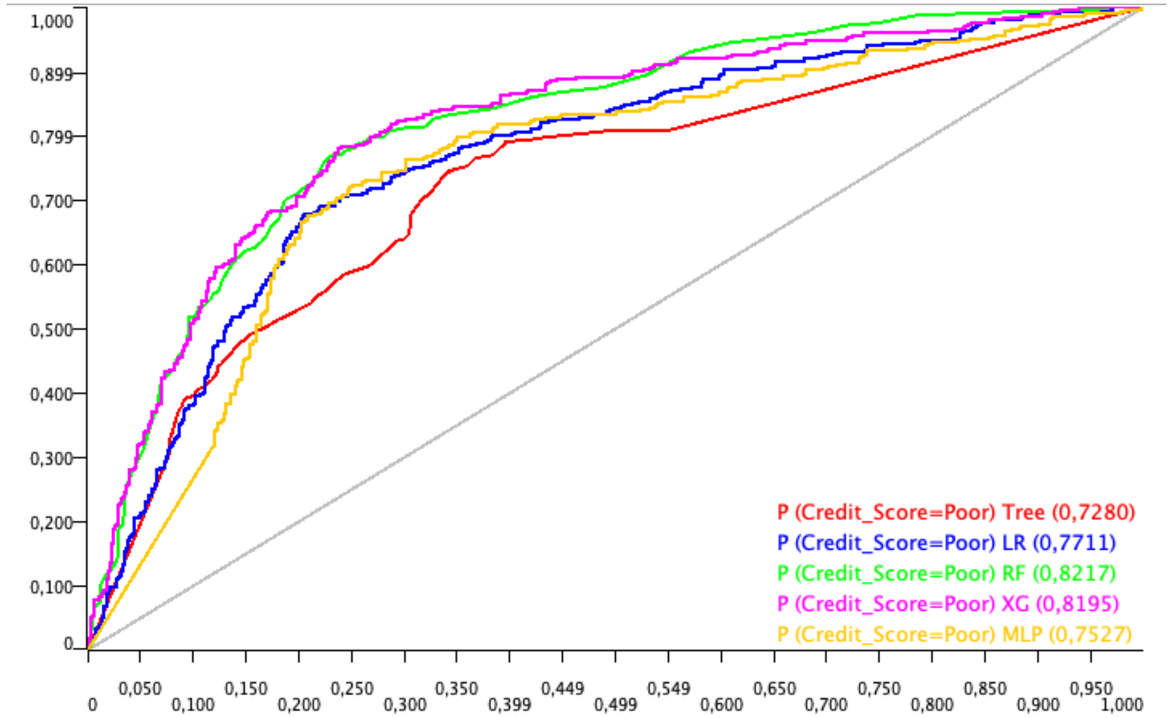
MODELS COMPARISON

Binary Classification

According to AUC, the models rank as follows:

1. **Random Forest (0.822)**
2. XGBoost (0.820)
3. Logistic Regression (0.771)
4. MLP (0.753)
5. Tree (0.728)

The ensemble models are the best ones, with Random Forest outperforming Gradient Boosting. The Logistic Regression is slightly behind followed by the ANN. The Naive Decision Tree is the worse model with AUC very distant from the top performers.



BEST MODEL

Binary Classification

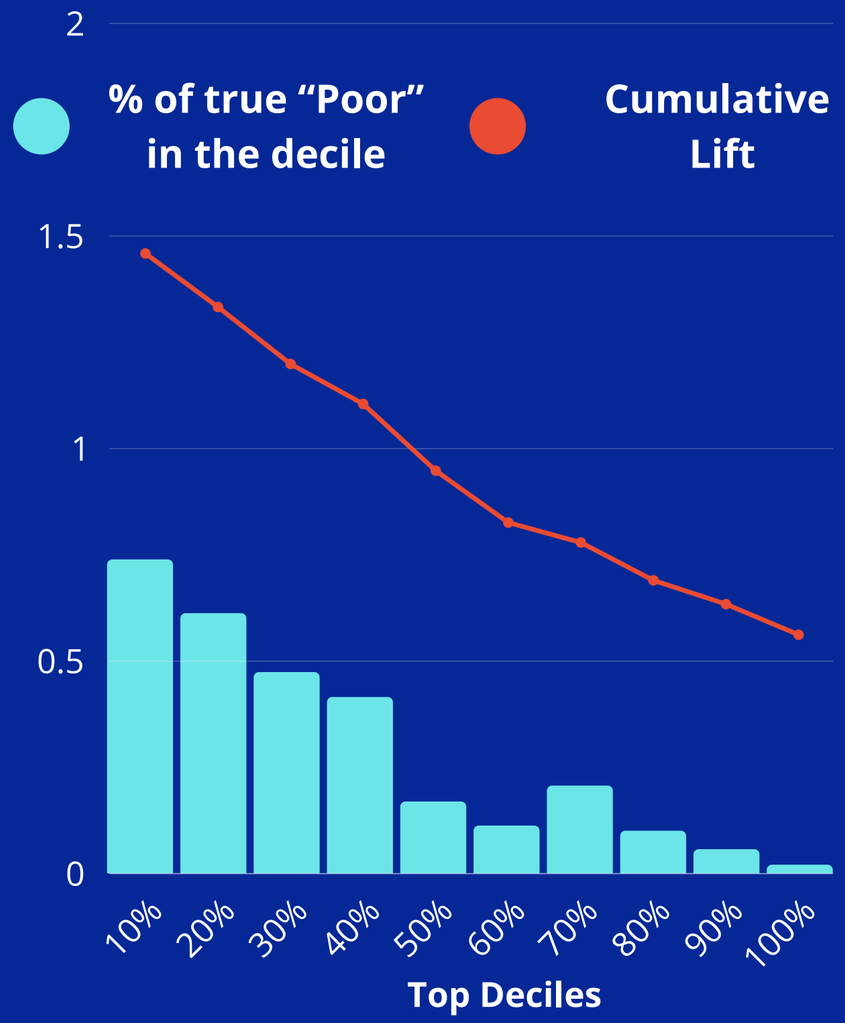
As we discussed, we should choose the model with a sensitivity as high as possible, in order to catch as many poor credit scores as possible.
Even though Random Forest has a sensitivity of 0.76 (slightly lower than the 0.78 XGB has), it should be the one chosen to carry out the task, due to a higher specificity, accuracy and AUC (the most significant measure when evaluating a classification task in general). This model always outperforms Decision Tree and MLP.
Logistic Regression is clearly the top model by specificity, but poorly performs when looking at the other metrics.

RANDOM FOREST

AUC = 0.8217
SENSITIVY = 76.06%
SPECIFICITY = 77.53%
ACCURACY = 77.05%
THRESHOLD = 0.360

CUMULATIVE LIFT

Binary Classification



For the purpose of our analysis, it is useful to look at the Lift chart. This measure shows how our model performs compared to a baseline model (random) as a function of the percentage of population considered, sorted by predicted probability.

E.g., among the top 10% observations with highest predicted probability of being “Poor”, about 70% are true “Poor”. A baseline model would have classified only 50% of these as poor. Thus our Random Forest has a lift of 0.7/0.5 = 1.4

This suggests an alternative approach to the Max Youden J we used for previous classification. That is, in order to maximize the benefits/costs ratio, a bank or credit institution should intervene on the top 40% observations with highest predicted probabilities, where the cumulative lift is above 1 and thus better than drawing at random.

SUPER LEARNER

Binary Classification

To conclude, we decided to combine the top performing models (Random Forest and Gradient Boosting) in order to try and improve the prediction even further.

To do so, we split the test set (now containing the predicted probabilities of the top models) and train-test a Logistic Regression.

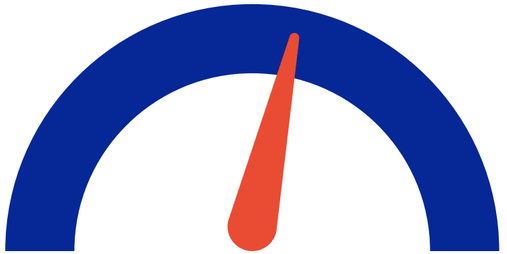
In this case, we didn't normalize the features as they already were probabilities [0,1], and didn't use regularization as we only had 2 features.

Prediction / Actual	Good or Standard	Poor
Good or Standard	206	66
Poor	20	83

Model Sensitivity: 80.58%

Model Specificity: 75.73%

Youden's Index: 0,563



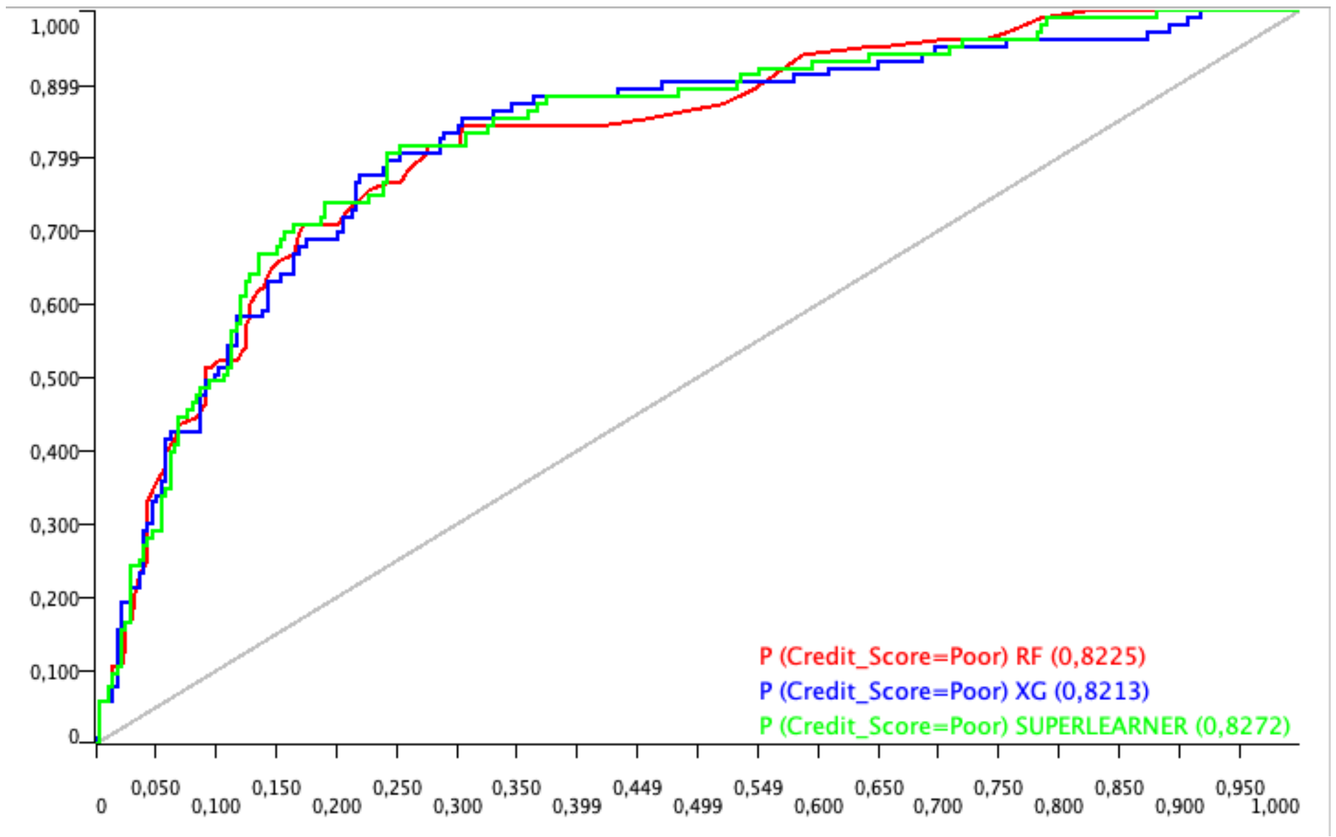
SUPER LEARNER

Binary Classification

The obtained model slightly outperforms the others on this chunk of test set, with a higher AUC:

- **SuperLearner (0.827)**
- RandomForest (0.822)
- XGBoost (0.820)

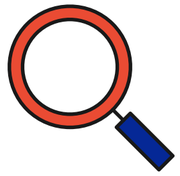
However, we decide to stick with the Random Forest classifier, as the improvement is not worth the additional training cost.



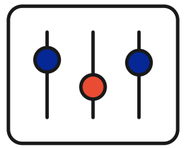
MANAGERIAL IMPLICATIONS

OVERVIEW

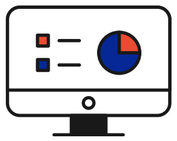
What can a bank gain from our analysis?



Most important characteristics to classify bad creditors



General decision rules for credit score classification



Machine Learning models to automatize credit classification

IMPORTANT FEATURES

Credit utilization patterns

The most predictive variables, according to PFI in the top performing model, are those related to credit. In particular, as it could intuitively be expected, "**Interest_rate**" is one of the most important features. Indeed, an higher interest rate is usually associated with worse credit behaviour.

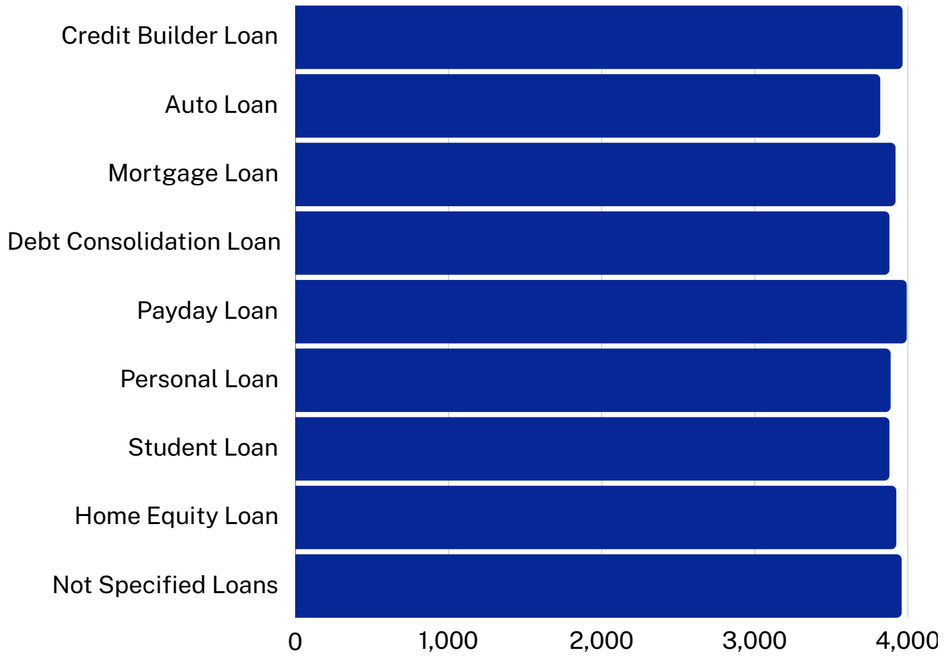
Credit Score	Avg. Interest Rate
720–850	10.73 - 12.50%
690–719	13.50 - 15.50%
630–689	17.80 - 19.90%
300–629	28.50 - 32.00%

Average Interest Rate for Personal Loans in 2019. Source: FED

IMPORTANT FEATURES

Already-owned credit characteristics

Even if it seems a bit tautologic, characteristics of already-owned credit by a customer are great indicators of his goodness as a debitor. This was also picked up by the models we trained, as we can notice that “**Delay_from_due_date**“ and the various categories of **Credit_Mix** are often in top positions in the PFI ranking of our chosen model. As they are an observable effect to a credit holder’s real goodness, they act as optimal predictor in ML analysis.



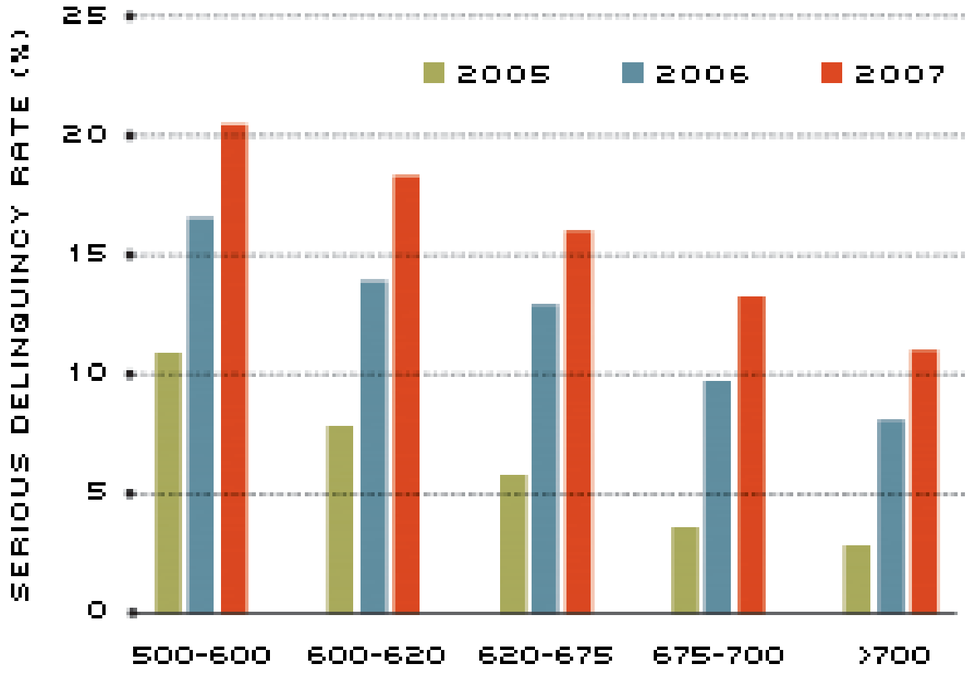
Type of Loan Occurences in our dataset

IMPORTANT FEATURES

The importance of the financial landscape

Notice that features studied in this report are not sufficient for a comprehensive overview of credit score.

It's also crucial to consider the ongoing financial landscape changes, including regulatory shifts and market trends, which are fundamental in shaping the banking sector's approach to credit scoring. These modifications of the market can in fact shift sensibly both the condition of the customer and the bank aversiveness towards risk.



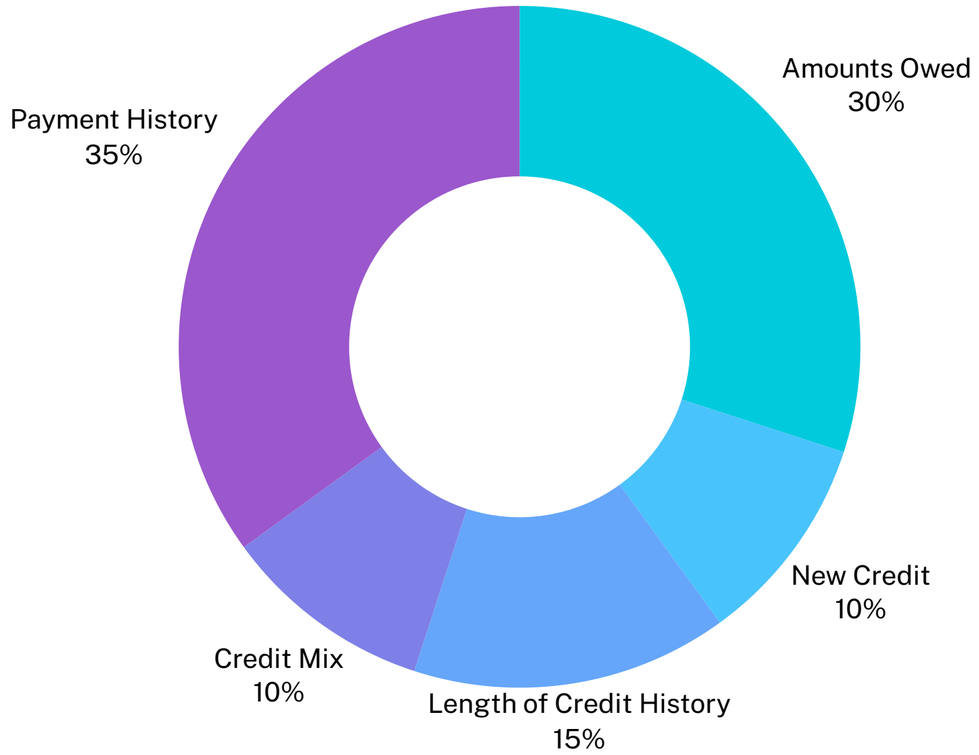
Customer default rate during the 2006 crisis for mortgage loans by credit score.
Source: St. Luis FED

VALIDATION OF FINDINGS

How is FICO score calculated

In the US, one of the most used metrics for credit risk is the **FICO Score**. It's widely used across the nation to have an easy understanding of a credit seeker's goodness as far as financial liabilities, and the calculation is freely accessible through the web.

It is possible to notice that the variables used in its calculation are present in our dataset and mostly significant across models, partially validating our process.

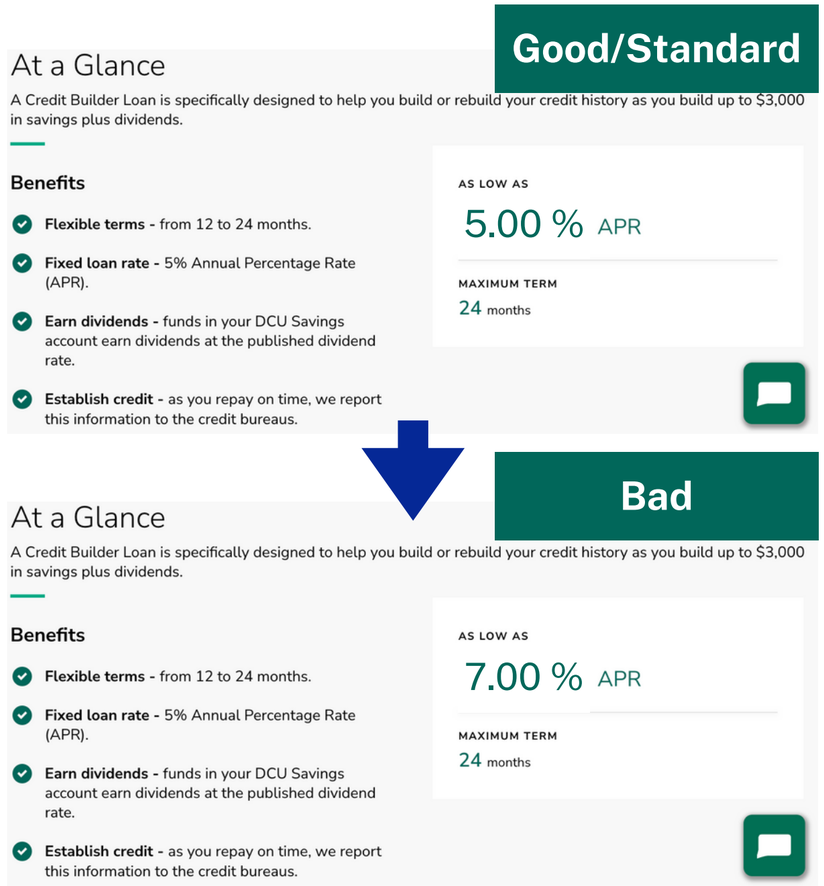


FICO Score impact of various credit-related data categories. Source: myFICO.com

POSSIBLE APPLICATIONS

Targeted advertisement

Our models could be implemented for fast customer segmentation when advertising loan rates to current customers: when looking for loan's interest rates this algorithm runs on their account and, depending on the rating, adjusts the maximum and/or minimum interest rates depending on the customer classification by using some pre-defined values given by marketing and credit risk departments. The effect of this type of treatment could be higher subscription rate from good credit score customers.

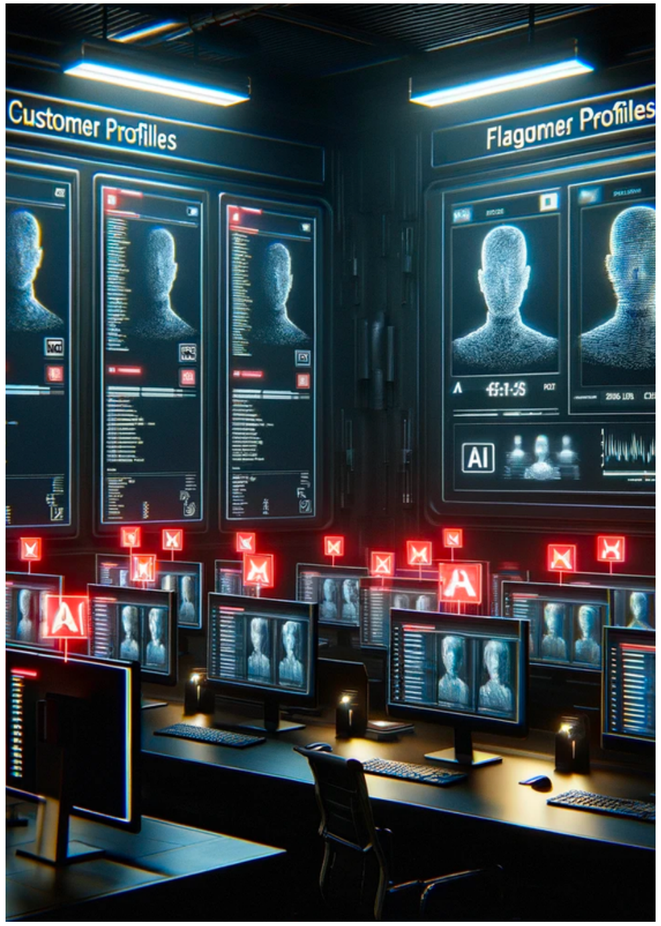


Simulation of adjusted interest rates in loan advertisement.

POSSIBLE APPLICATIONS

Flagging changes in customer behaviors

A second possible application for our model could be a lightweight and cost efficient detection of changes in customer credit behavior. As credit scores needs documents from bureaus to be recalculated, this model could be implemented to periodically recalculate the customer's credit score bracket: if changes are detected, the customer is flagged and a more accurate analysis by a human operator is requested.
This could be an improvement in current workflows of banks, granting more accurate credit segmentation over time.



THANK YOU FOR YOUR TIME!

GROUP 9

Francesco Vacca

Giacomo Cirò

Luca Colaci

Davide Romano

Costanza D’Ercole

Alessandro Morosini