# Análise de Credit Score

## Trabalho Final - Digital House Coding School

Modelos de classificação - Credit Score

Grupo 1 - Augusto Ribeiro, Eduardo Henrique,
Felipe Ferraz, Lucas Ribeiro, Thiago Ferro, Inês Dantas

# Sumário

# Avaliação de risco e Inadimplência

- Serasa: nº inadimplentes cresce desde 2016. 66,6mi nomes negativados por dívidas;
- 28,2% das dívidas são com **bancos e cartões**, seguido de 22,7% com contas essenciais;
- Como os bancos definem qual limite de crédito devem liberar para os clientes? E em relação aos empréstimos, quantos, a quantia e **SE** os concedem?
- Avaliação de Riscos: capacita a identificação de onde investir/emprestar → evitar inadimplência, ou seja, perder o dinheiro


# O que é o credit score?

Credit Score é uma pontuação resultante dos hábitos de pagamento e relacionamento do consumidor com o mercado de crédito. Fatores como pagar contas em dia, histórico de dívidas negativadas, saldo devedor etc, representam o risco de inadimplência e saúde financeira geral de uma pessoa.

O score vai de 0 a 1.000 pontos. Esses pontos foram divididos em nosso dataset dentre três classes (classificação multiclasse):

- Poor (ruim) - alto risco de inadimplência;
- Standard (padrão) - médio risco;
- Good (bom) - baixo risco.

# Dados

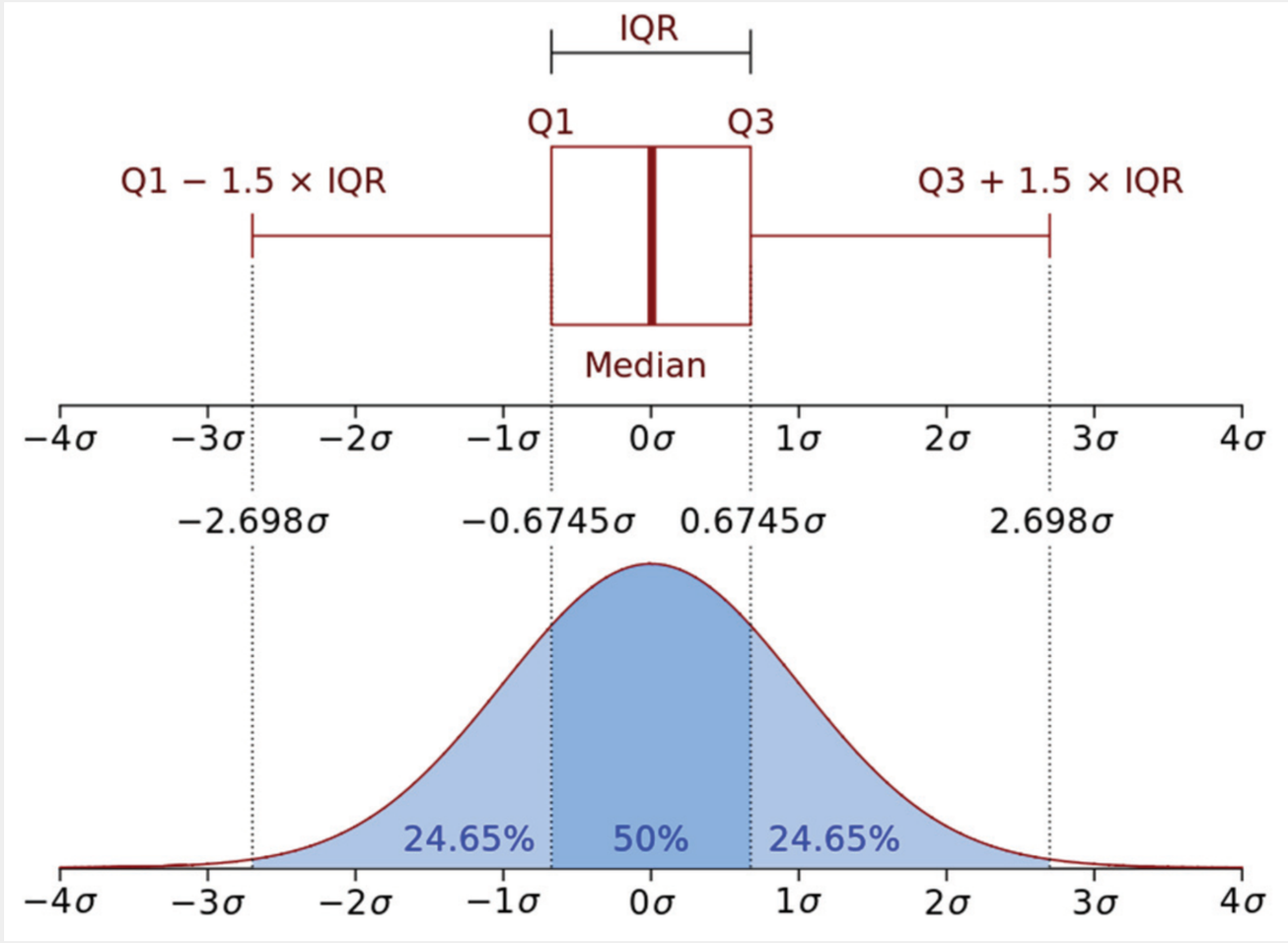
Os dados escolhidos foram obtidos no Kaggle e são fictícios. O objetivo é construir um sistema inteligente para segregar as pessoas em faixas de pontuação de crédito para reduzir os esforços manuais.

- ID - Identificador único de entrada (observação)
- Customer_ID - ID único de cliente
- Month - Mês do ano
- Name - nome do cliente
- Age - Idade do cliente
- SSN - Social Security Number (CPF no Brasil)
- Occupation - Ocupação do cliente
- Monthly_Inhand_Salary - Salário mensal do cliente
- Num_Bank_Accounts - Quantidade de contas em bancos
- Num_Credit_Card - Quantidade de cartões de crédito
- Interest_Rate - Taxa de juros do cartão de crédito
- Num_of_Loan - Quantidade de empréstimos feitos no banco
- Type_of_Loan - Tipo de empréstimo feito pelo cliente
- Num_Credit_Inquiries - Quantidade de "cobranças" no cartão
- Total_EMI_per_month - Parcela mensal de empréstimo
- Outstanding_Debt - Restante da dívida a ser paga
- Delay_from_due_date - Qtd. de dias de atraso no pagamento do cartão
- Changed_Credit_Limit - Variação percentual de limite do cartão de crédito
- Credit_History_Age - Tempo de histórico de crédito do cliente
- Monthly_Inhand_Salary - Salário mensal do cliente
- Credit_Mix - Qualidade da variedade de crédito do cliente (crédito rotativo e parcelado)
- Num_of_Delayed_Payment - Média de pagamentos atrasados pelo cliente
- Credit_Utilization_Ratio - Taxa de utlização do cartão de crédito
- Payment_of_Min_Amount - Se o cliente pagou ou não o mínimo da dívida
- Monthly_Balance - Saldo mensal do cliente
- Payment_Behaviour - Comportamento de pagamento do cliente
- **Credit Score - Pontuação da credibilidade do cliente**

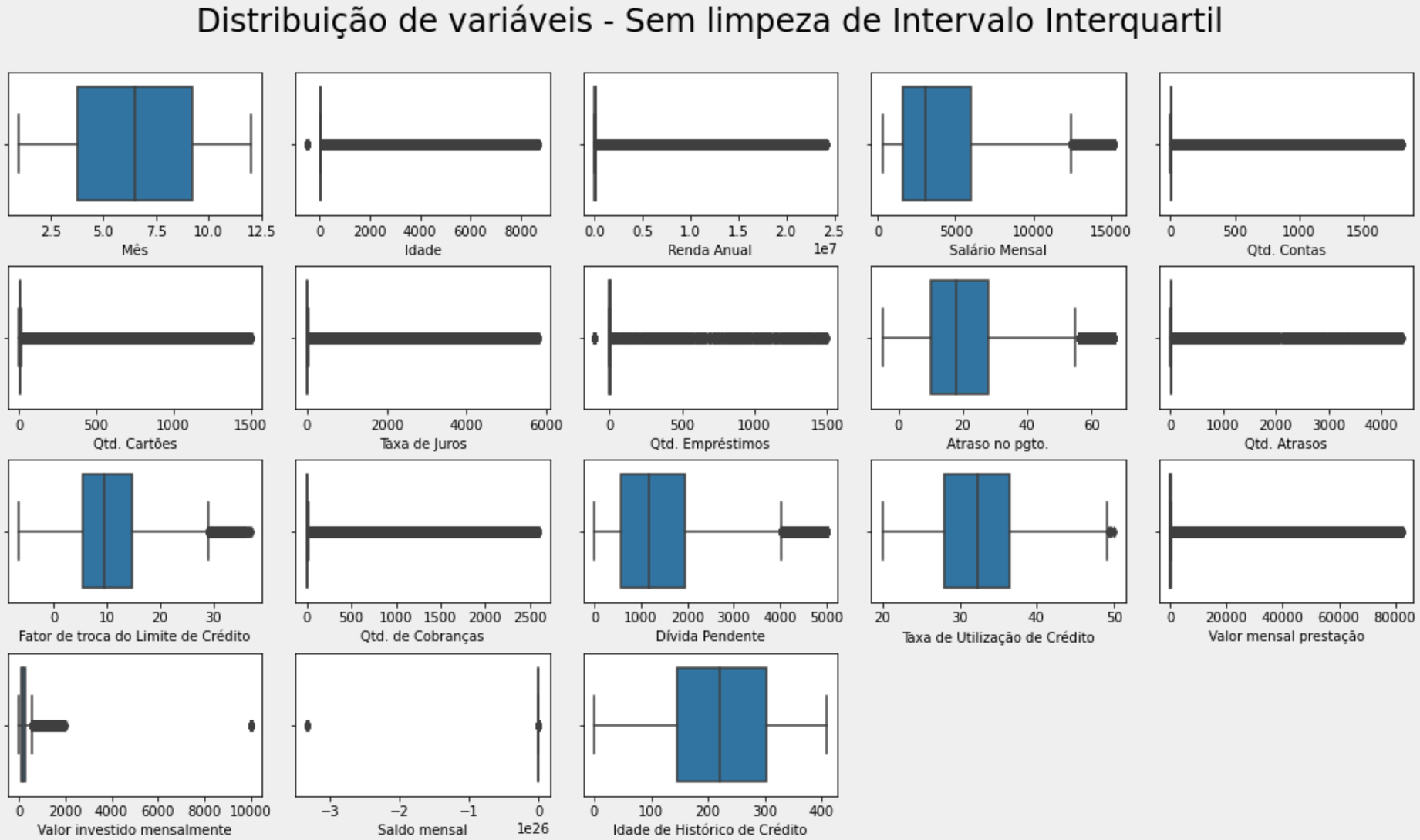
# EDA e Data Cleaning

# EDA e Data Cleaning

- Drop de todos os NaN
- Idade de 18 a 100 anos;
- Drop de Outliers com intervalo interquartil;

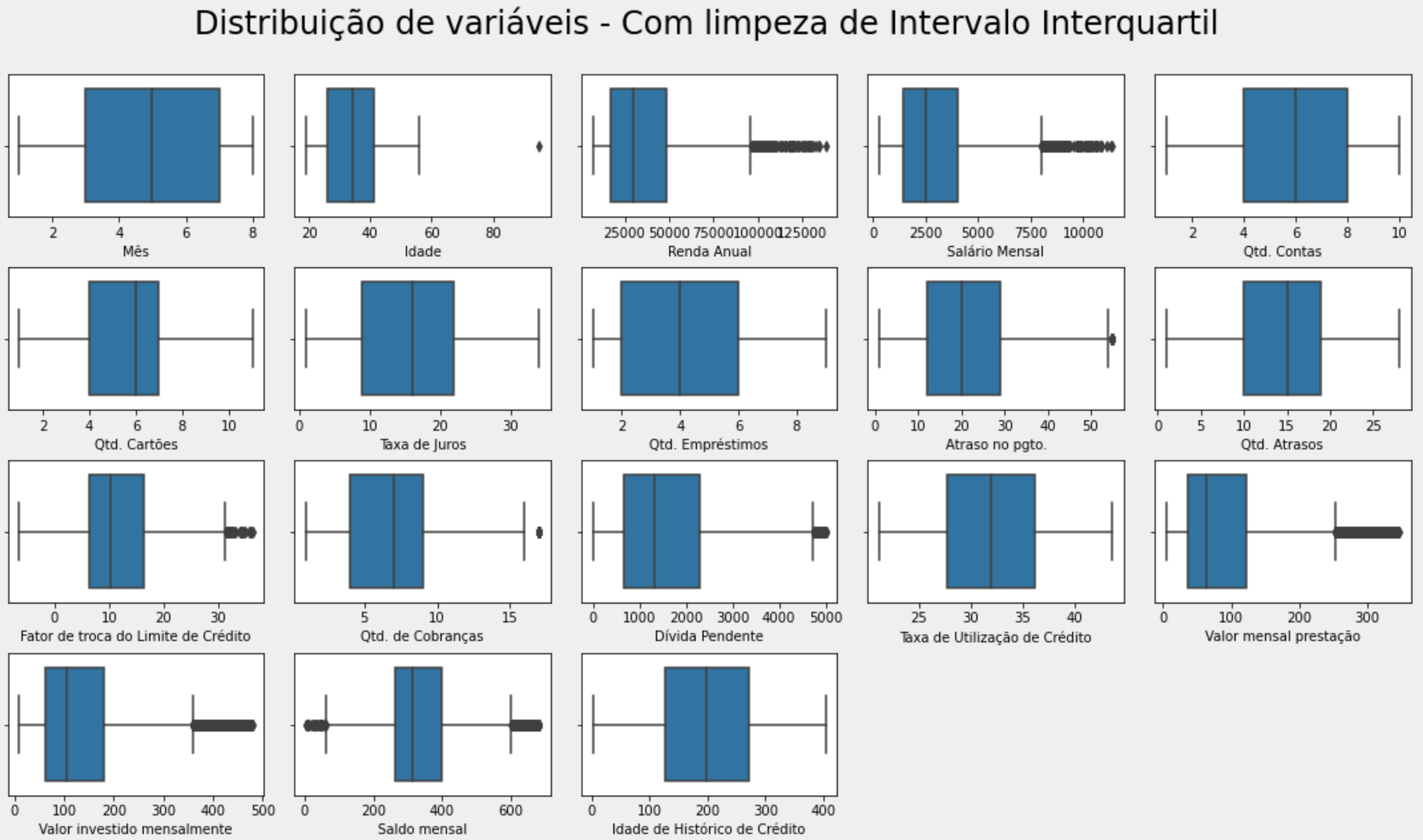


# EDA e Data Cleaning



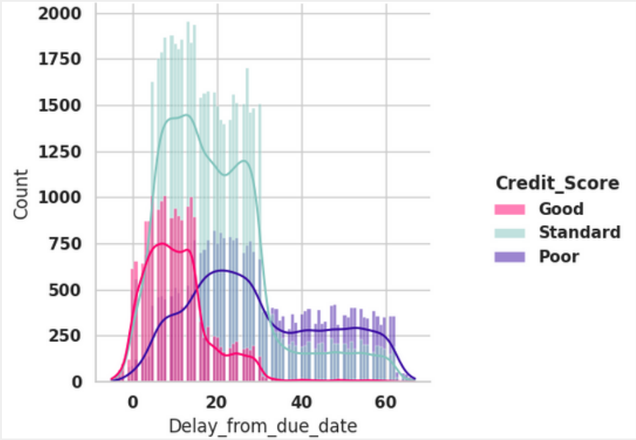
# EDA e Data Cleaning

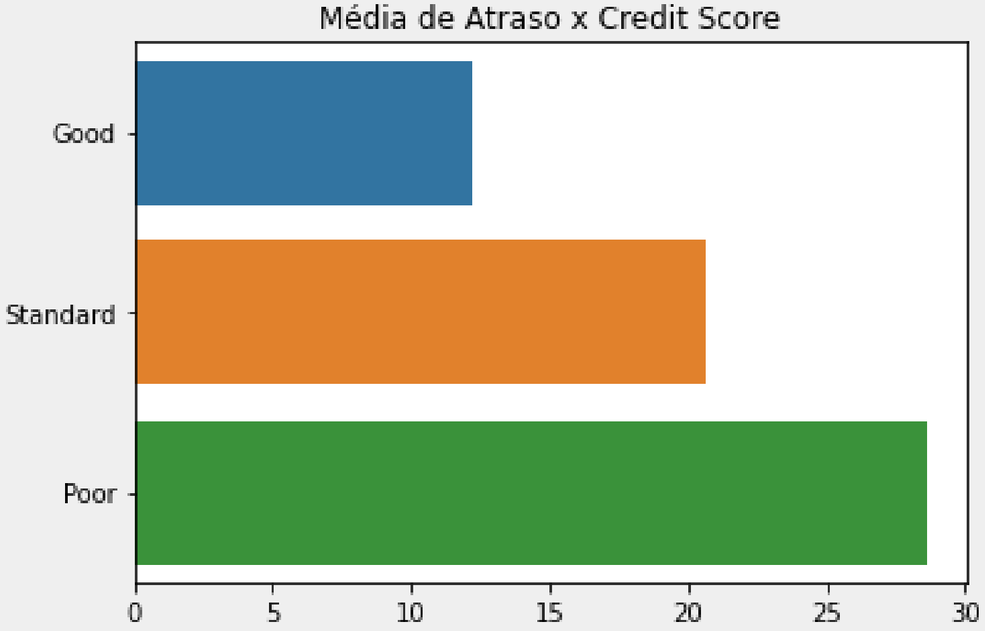
- 50% das idades estão na faixa de [26; 41];
- 75% das pessoas ganham menos de 4 mil;
- Em média, as pessoas tem 6 cartões e 4 empréstimos feitos;
- Em média, as pessoas tem saldo devedor de 1600;
- 50% das pessoas investem mensalmente na faixa de [62; 181].

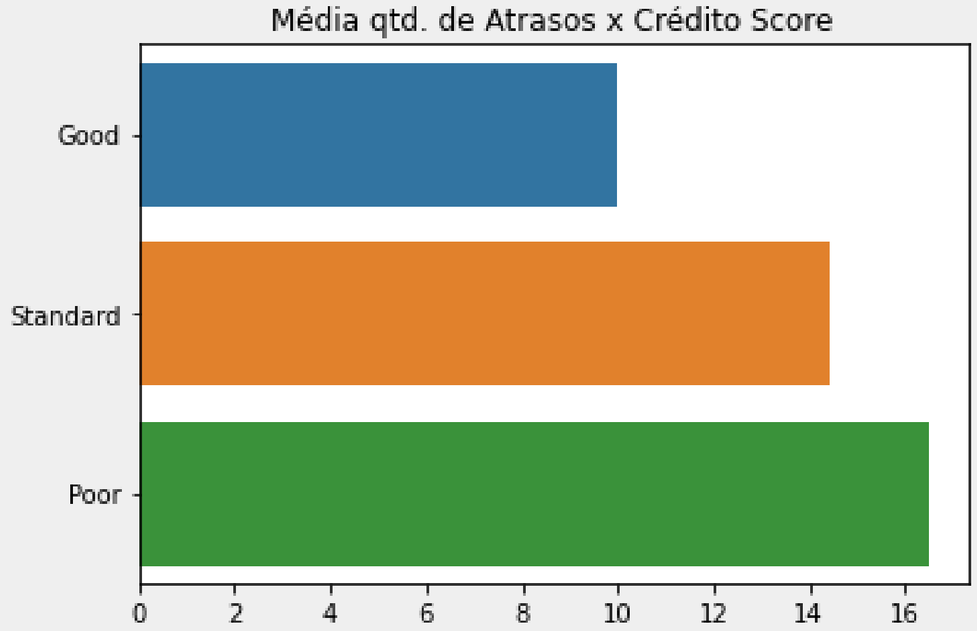


# EDA e Data Cleaning

- Pessoas com Credit Score mais baixo tem uma média de atraso no pagamento das parcelas maior. Assim como uma maior quantia de atrasos.



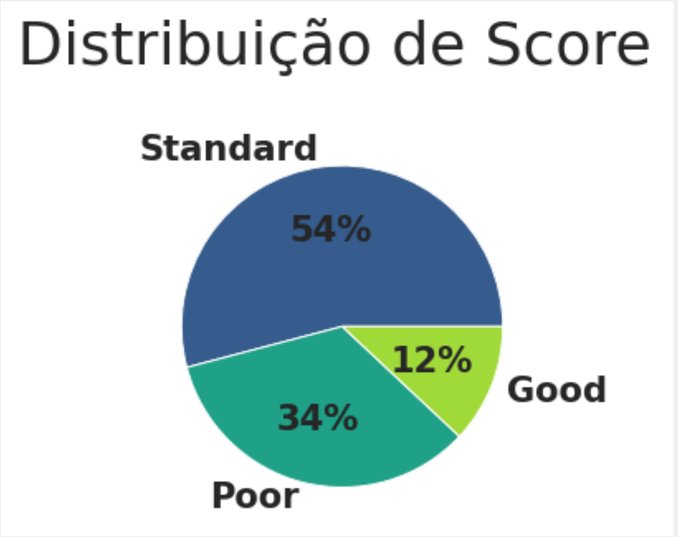



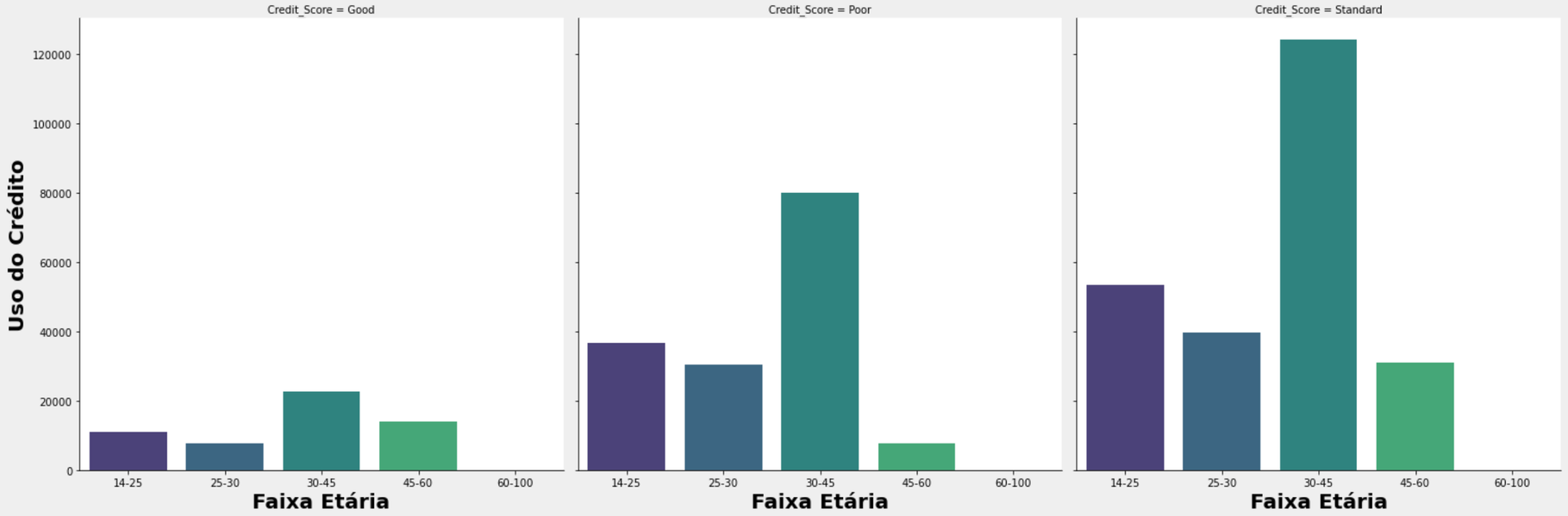



# EDA e Data Cleaning

**Credit Score e Uso do Crédito**

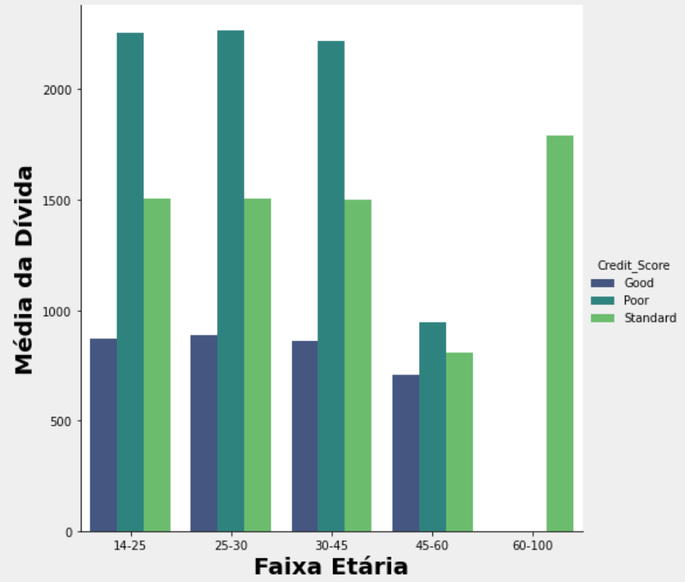
- Maioria do Credit Score é Standard (54%), **target desbalanceada**;
- Idades entre 35 e 45 anos usam mais o Cartão de Crédito;

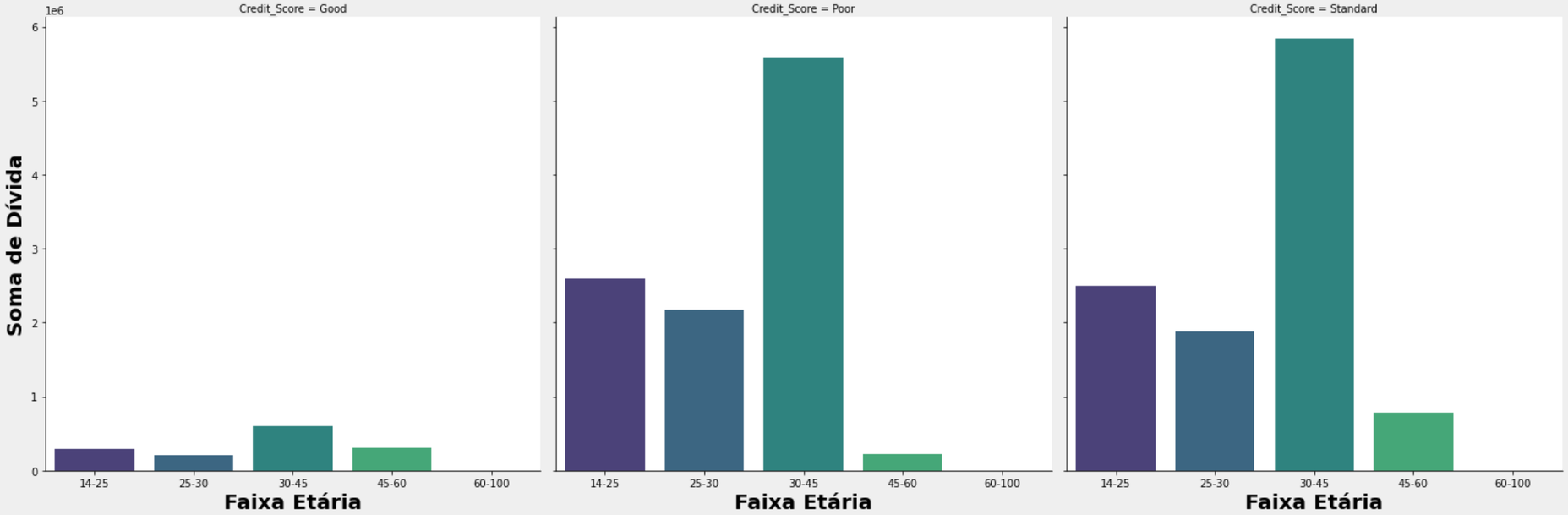


# EDA e Data Cleaning

- Pessoas com alta dívida tendem a ter pior Credit Score;
- Idades entre 25 e 45 anos possuem mais dívida, provavelmente pelo maior uso do crédito.

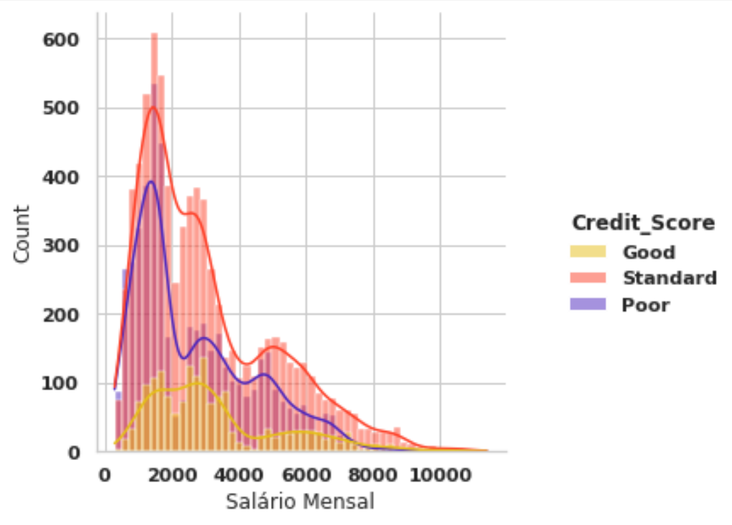


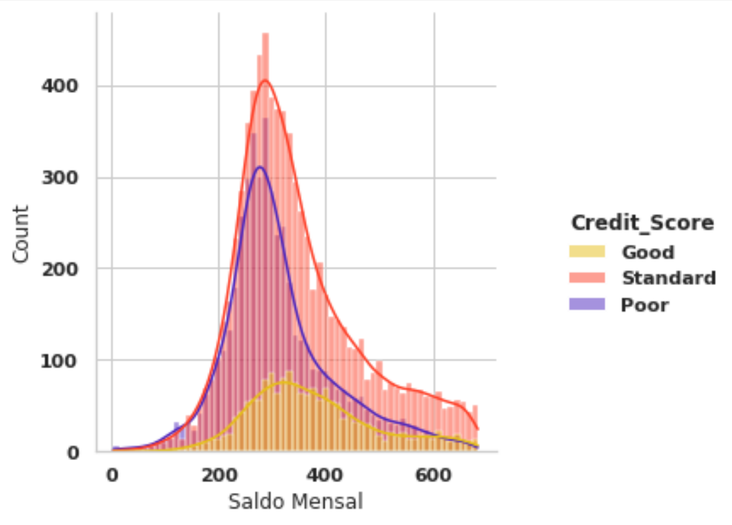



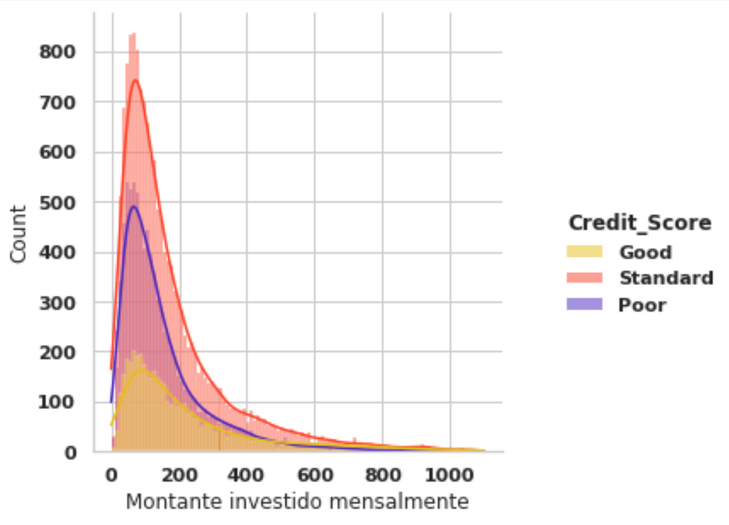
# EDA e Data Cleaning

- O quanto a pessoa ganha não implica diretamente no Credit Score, mas sim o quão bem ela gerencia suas finanças;

- Pessoas com bom Credit Score costumam guardar mais dinheiro ao final do mês.



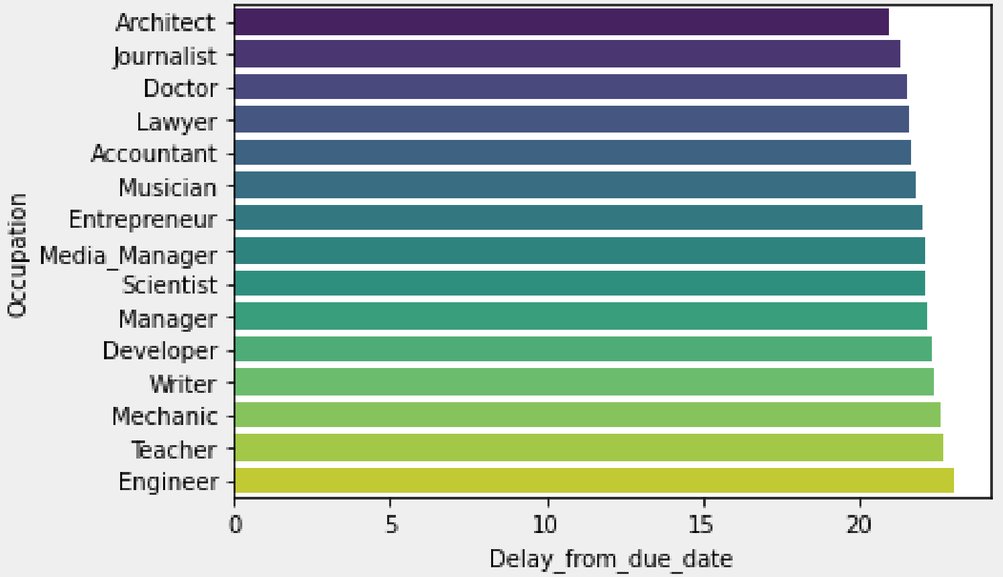



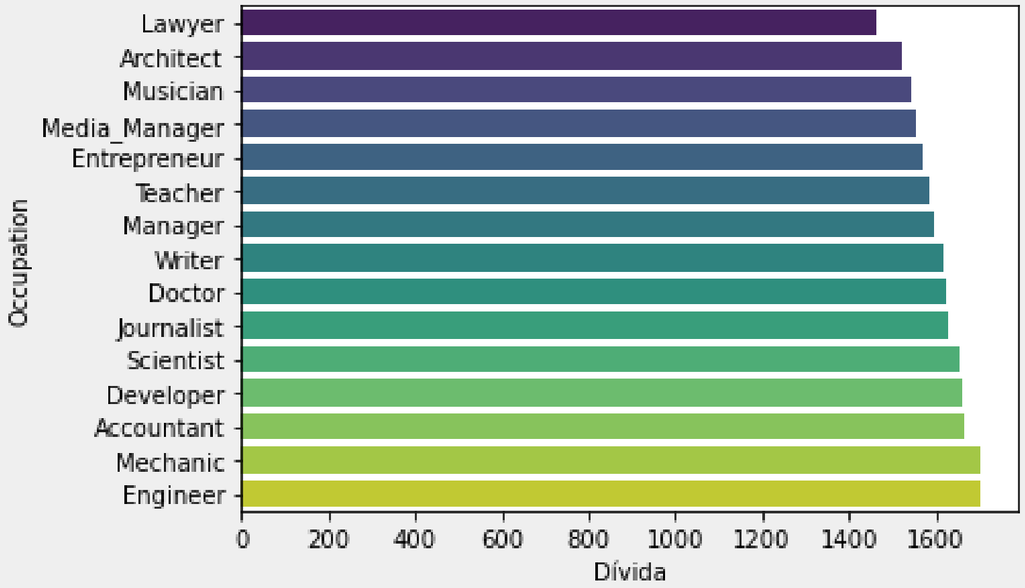



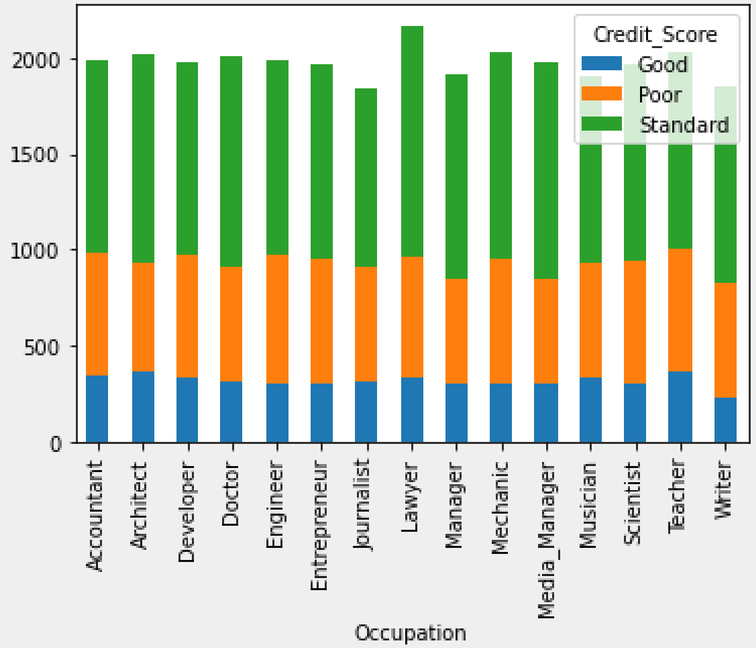
# EDA e Data Cleaning

**Profissão**

- Apresenta distribuição uniforme em atrasos e Dívida;
- Acompanham a distribuição desbalanceada de Credit Score, ou seja, não influencia na target.



# Feature Engineering

Data Wrangling, Feature Importance e
Feature Selection

# Encoding e Normalização

**Variáveis Dropadas**

- Customer_ID, Name, SSN, Type_of_Loan, Type_of_Loan_ajustado, Credit_History_Age, Credit_Score, Month, Monthly_Balance, Occupation, Payment_Behaviour

**Categóricas (pd.DataFrame.factorize())**

- Credit_Mix (mix de crédito) e Payment_of_Min_Amount (pagamento do mínimo da dívida)

**Numéricas (Standard Scaler)**

- Utilizamos o Standard Scaler nas variáveis numéricas restantes (média 0 e desvio padrão 1)

**Target multiclasse → 0 = Good, 1 = Poor, 2 = Standard**

## Factorize de Credit Mix

```
Standard    10882
Good         5781
Bad          5255
Name: Credit_Mix, dtype: int64


1    10882
0     5781
2     5255
Name: Credit_Mix, dtype: int64
```

# Feature Selection

**Tabela StatsModels Summary**

Passam no teste e tem coeficiente relevante:

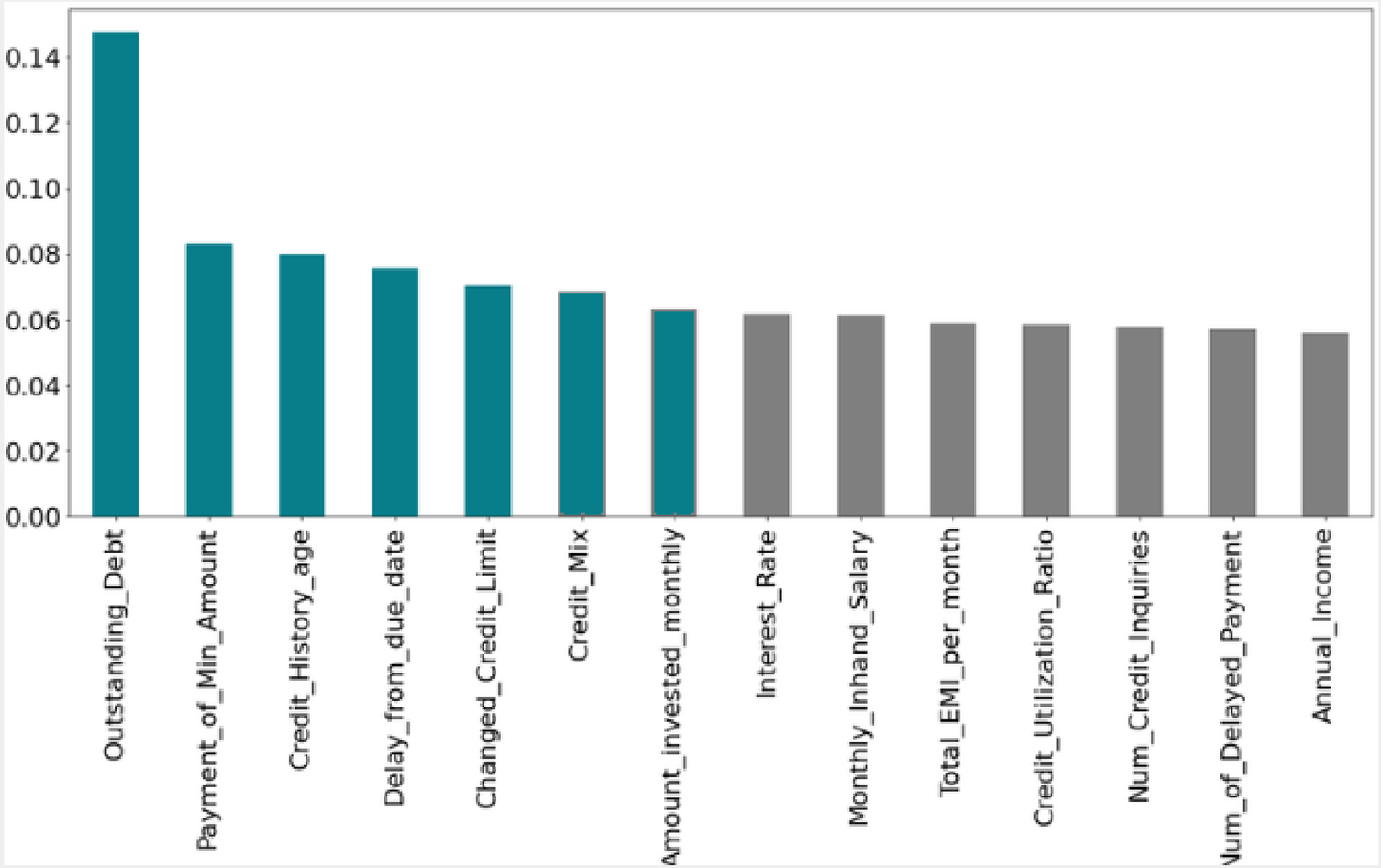
- Salário Mensal (Monthly_Inhand_Salary);
- Dias de atraso de pagamento do cartão (Delay_from_due_date);
- Fator de troca de limite do cartão (Changed_Credit_Limit);
- Saldo devedor (Outstanding_Debt);
- Mix de Crédito (Credit_Mix);
- Tempo de histórico com cartão (Credit_History_Age); e
- Pagamento do montante mínimo (Payment_of_Min_Amount).

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| Annual_Income | -0.0015 | 0.006 | -0.255 | 0.799 | -0.013 | 0.010 |
| Monthly_Inhand_Salary | 0.0630 | 0.006 | 10.299 | 0.000 | 0.051 | 0.075 |
| Interest_Rate | 0.0009 | 0.006 | 0.156 | 0.876 | -0.010 | 0.012 |
| Delay_from_due_date | -0.2421 | 0.007 | -33.600 | 0.000 | -0.256 | -0.228 |
| Num_of_Delayed_Payment | -0.0053 | 0.006 | -0.932 | 0.351 | -0.017 | 0.006 |
| Changed_Credit_Limit | 0.0516 | 0.007 | 7.516 | 0.000 | 0.038 | 0.065 |
| Num_Credit_Inquiries | -0.0014 | 0.006 | -0.244 | 0.807 | -0.012 | 0.010 |
| Outstanding_Debt | -0.3321 | 0.009 | -38.266 | 0.000 | -0.349 | -0.315 |
| Credit_Utilization_Ratio | 0.0104 | 0.006 | 1.792 | 0.073 | -0.001 | 0.022 |
| Total_EMI_per_month | -0.0064 | 0.006 | -1.121 | 0.262 | -0.018 | 0.005 |
| Amount_invested_monthly | 0.0032 | 0.006 | 0.554 | 0.580 | -0.008 | 0.014 |
| Credit_History_age | 0.1942 | 0.008 | 24.861 | 0.000 | 0.179 | 0.210 |
| Credit_Mix | 1.0444 | 0.013 | 82.097 | 0.000 | 1.019 | 1.069 |
| Payment_of_Min_Amount | 0.3845 | 0.019 | 20.404 | 0.000 | 0.348 | 0.421 |

Testes de Hipóteses das Features

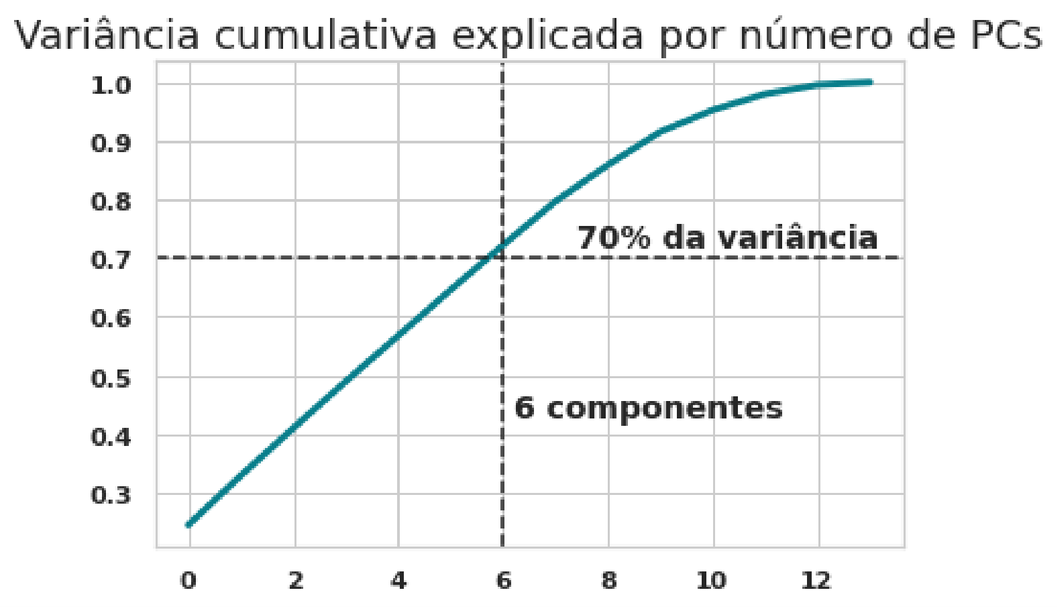
# Feature Selection

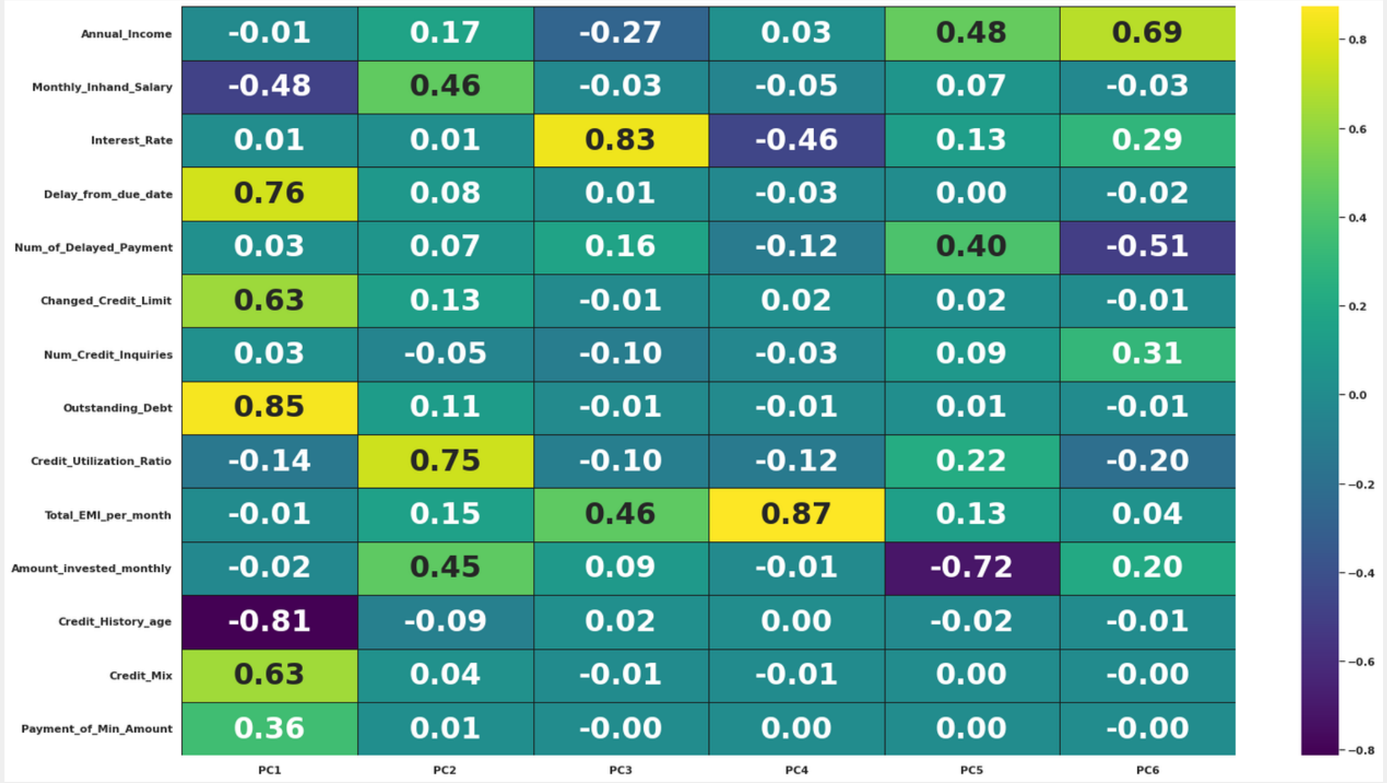
**ExtraTreesClassifier.feat_importance_ (Coef. Gini)**



# PCA

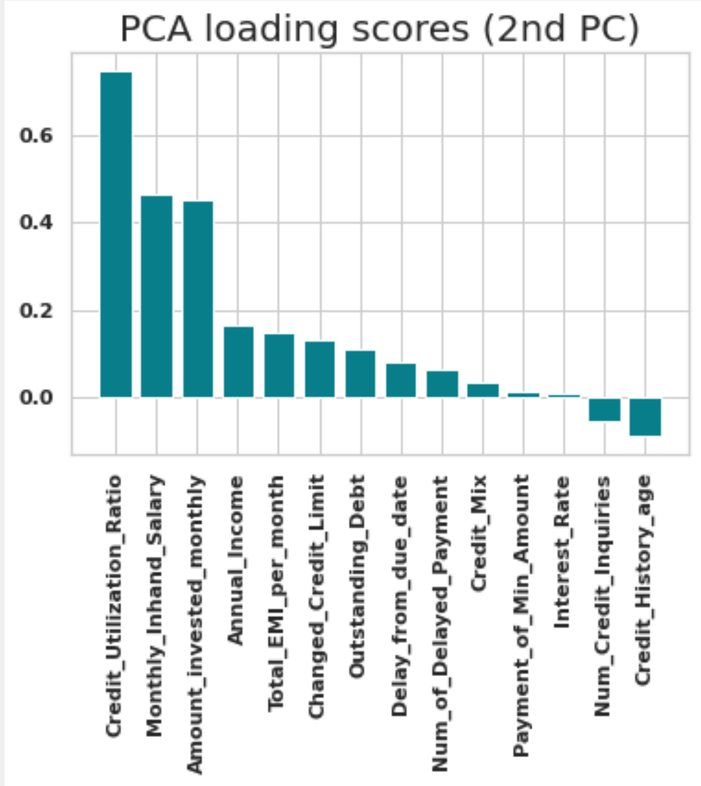
- Análise de componentes principais nos permite visualizar quais das variáveis tem maior influência no comportamento dos dados com a target;
- No caso, pelo menos 70% da variância é explicada pelas 6 primeiras PCs.

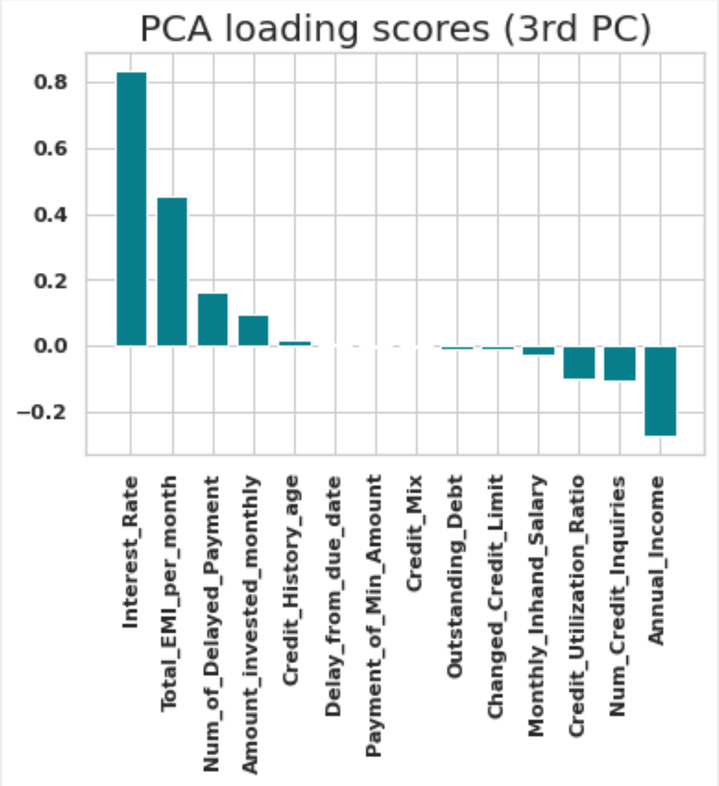


# PCA

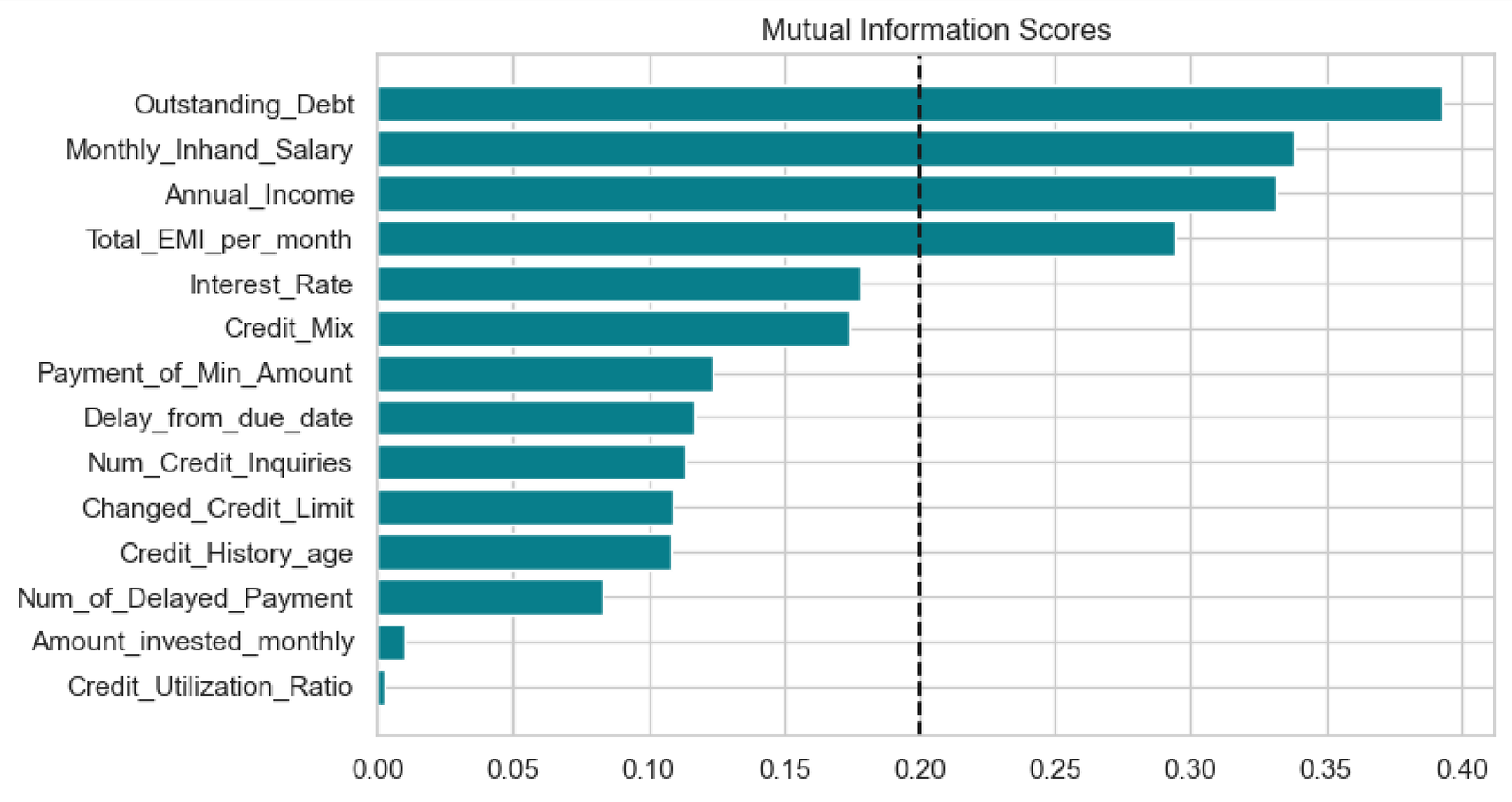
- Correlação das 3 primeiras componentes principais com as features:

# Mutual Information Score

A pontuação de Informação Mútua expressa até que ponto a frequência observada de coocorrência entre duas variáveis difere do que seria de esperar, ou seja, o quanto elas se explicam mutuamente.



# Melhores Features

Dos métodos, as melhores features são:

- Saldo devedor (Outstanding_Debt);
- Salário Mensal (Monthly_Inhand_Salary) e Renda Anual (Annual_Income);
- Mix de Crédito (Credit_Mix);
- Razão de uso do Cartão de Crédito (Credit_Utilization_Ratio);
- Dias de atraso de pagamento do cartão (Delay_from_due_date);
- Fator de troca de limite do cartão (Changed_Credit_Limit);
- Montante investido mensalmente (Amount_invested_monthly);
- Tempo de histórico com cartão (Credit_History_Age); e
- Pagamento do montante mínimo (Payment_of_Min_Amount).

- 3 Métodos de Feature Selection que conversam entre si, o que trás maior confiança na seleção

# Modelos

Extra Trees

Random Forest

XGBoost

Multi-layer Perceptron

Stacking: XGBoost e SVC

# Características Gerais

- Target multi-classe (Poor, Standard, Good);
- Splits com Stratify = y, lida com assimetria da target;
- Todos os modelos foram treinados com features selecionadas;
- Otimização de hiperparâmetros pelo método Bayesiano e com foco em Acurácia e F1-Score (média harmônica entre precision e recall).


$$F_1\text{–score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$$

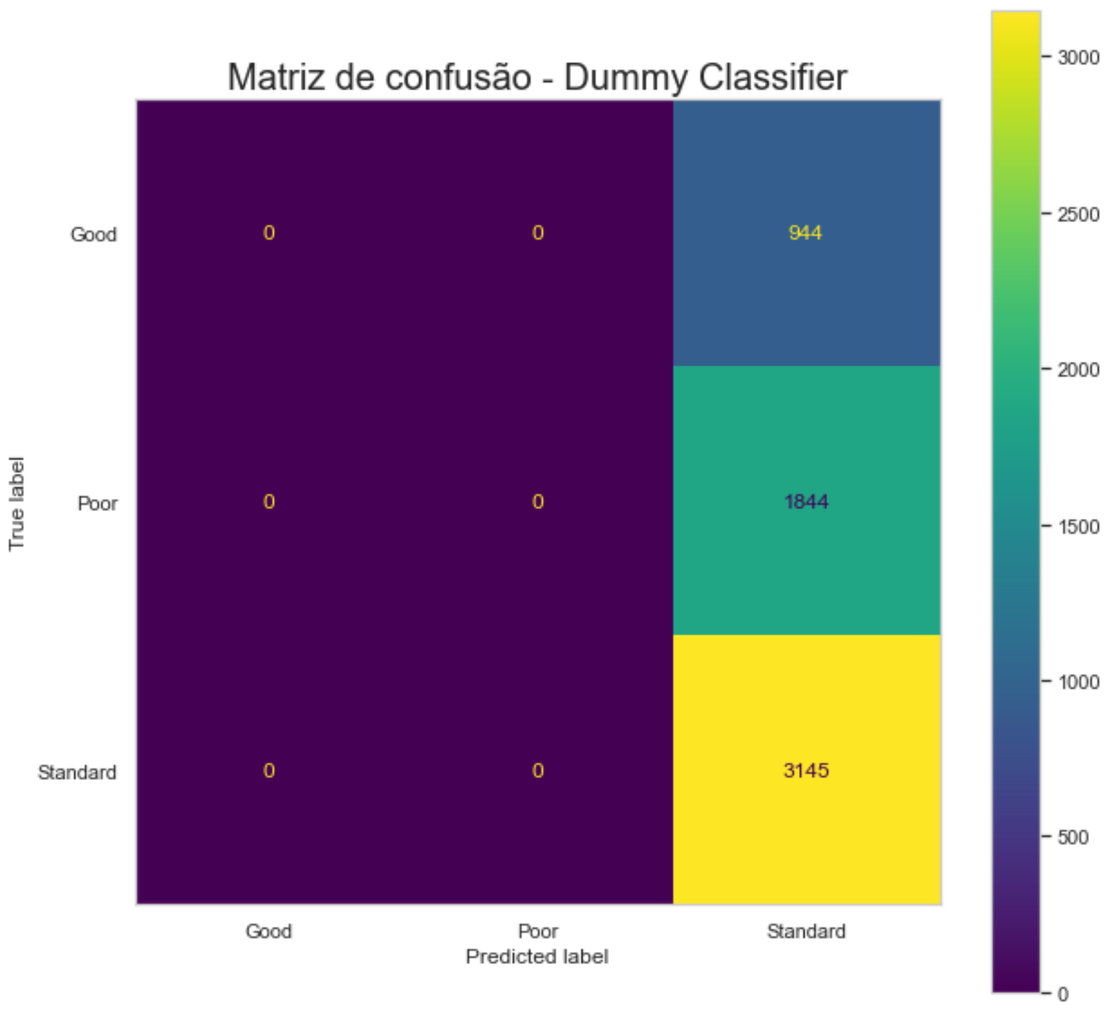
# Comparativo para seleção de Modelos (SKLearn)

- Através do método Cross Val Score do SKLearn, testamos 7 modelos para escolher os modelos a treinar realmente.

```
* Comparação Cross-Validation-Score entre vários modelos:
```

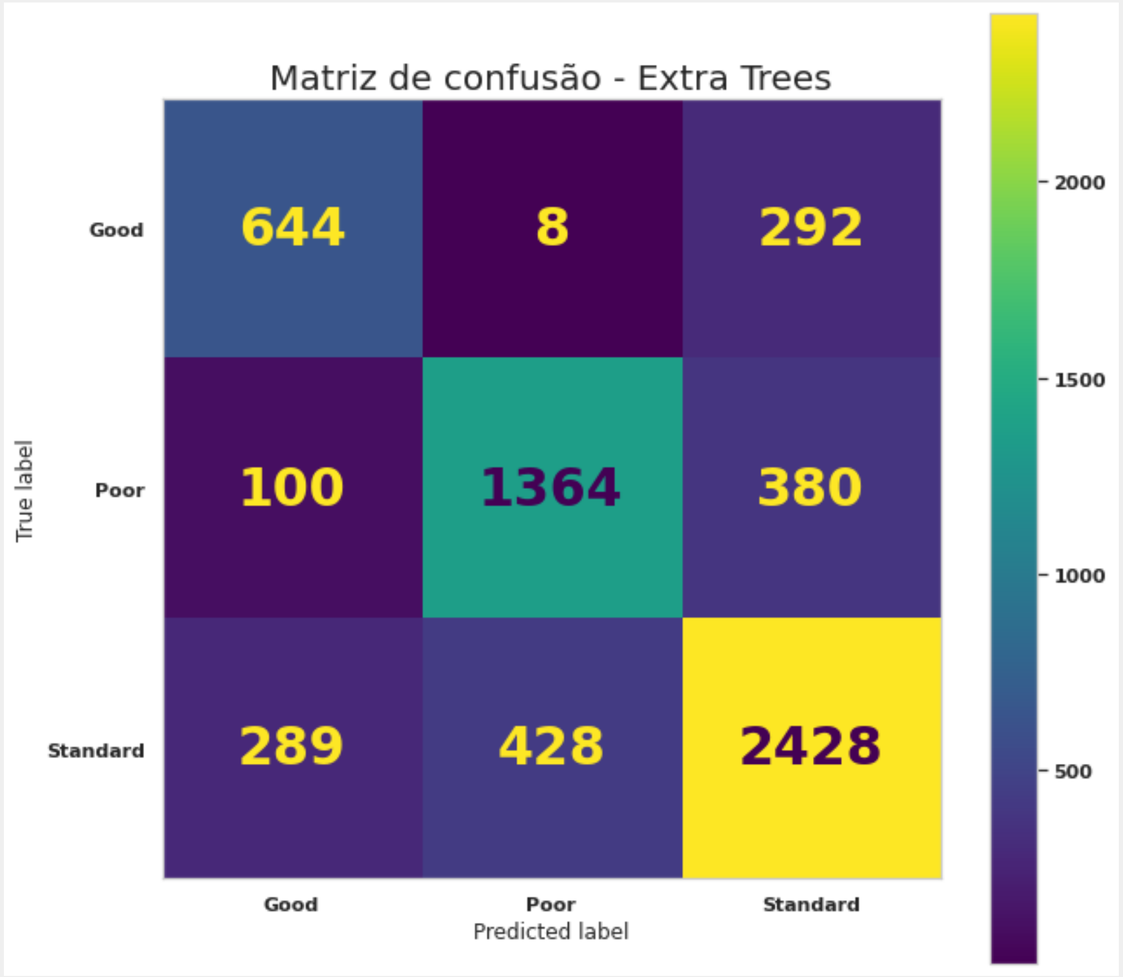
| | Modelo | Score |
|---|---|---|
| 0 | LogisticRegression | 0.6108 |
| 1 | KNeighborsClassifier | 0.6647 |
| 2 | DecisionTreeClassifier | 0.6529 |
| 3 | RandomForestClassifier | 0.7470 |
| 4 | XGBClassifier | 0.7248 |
| 5 | ExtraTreeClassifier | 0.6248 |
| 6 | MLPClassifier | 0.6749 |

# Modelo Baseline: Dummy Classifier



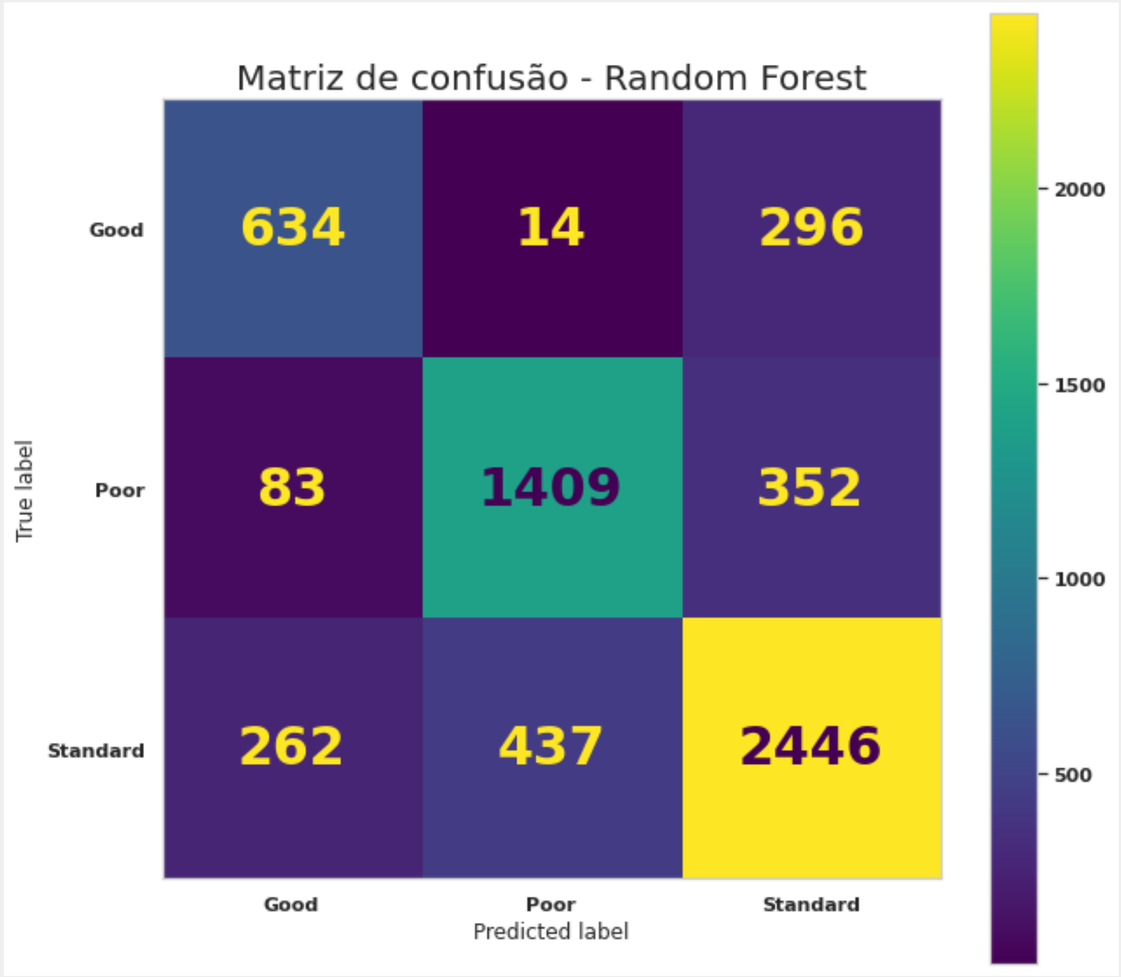

- Modelo que chuta todas as predições como Standard e serve de benchmark na performance dos outros modelos;
- Acurácia: 0.5301;
- OBS.: Não é possível calcular seu Recall e F1-Score, já que teríamos divisão por zero.

# Extra Trees



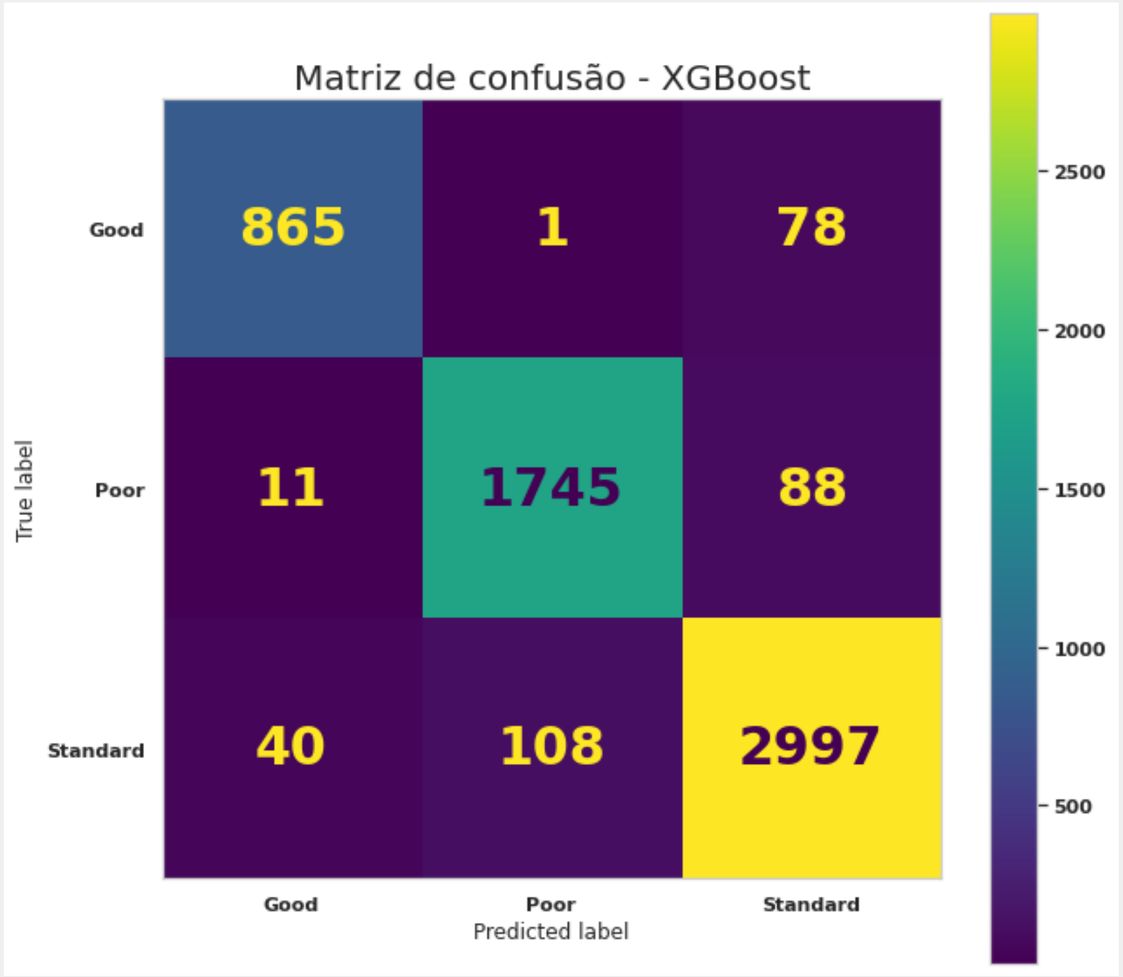

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.62 | 0.68 | 0.65 | 944 |
| 1 | 0.76 | 0.74 | 0.75 | 1844 |
| 2 | 0.78 | 0.77 | 0.78 | 3145 |
| accuracy | | | 0.75 | 5933 |
| macro avg | 0.72 | 0.73 | 0.73 | 5933 |
| weighted avg | 0.75 | 0.75 | 0.75 | 5933 |

# Random Forest



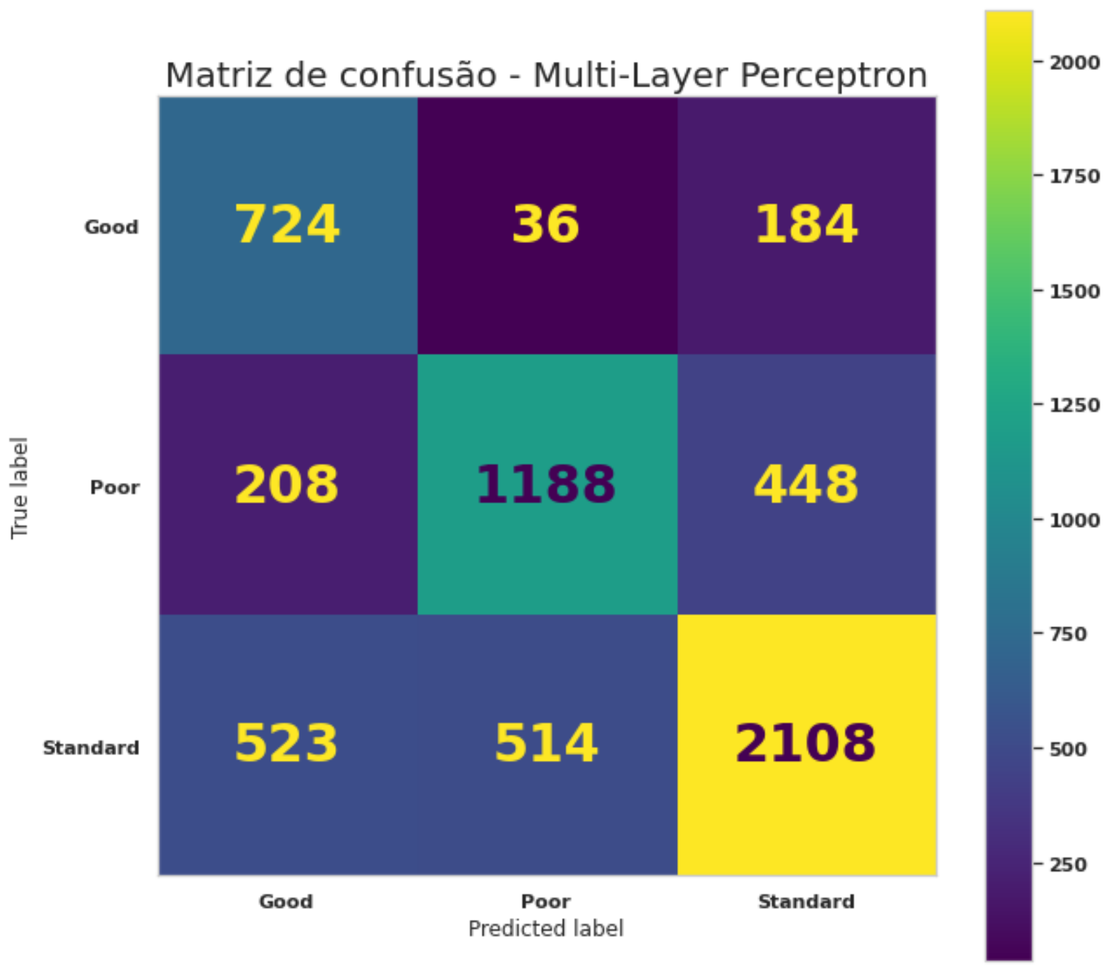

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.65 | 0.67 | 0.66 | 944 |
| 1 | 0.76 | 0.76 | 0.76 | 1844 |
| 2 | 0.79 | 0.78 | 0.78 | 3145 |
| accuracy | | | 0.76 | 5933 |
| macro avg | 0.73 | 0.74 | 0.73 | 5933 |
| weighted avg | 0.76 | 0.76 | 0.76 | 5933 |

# XGBoost



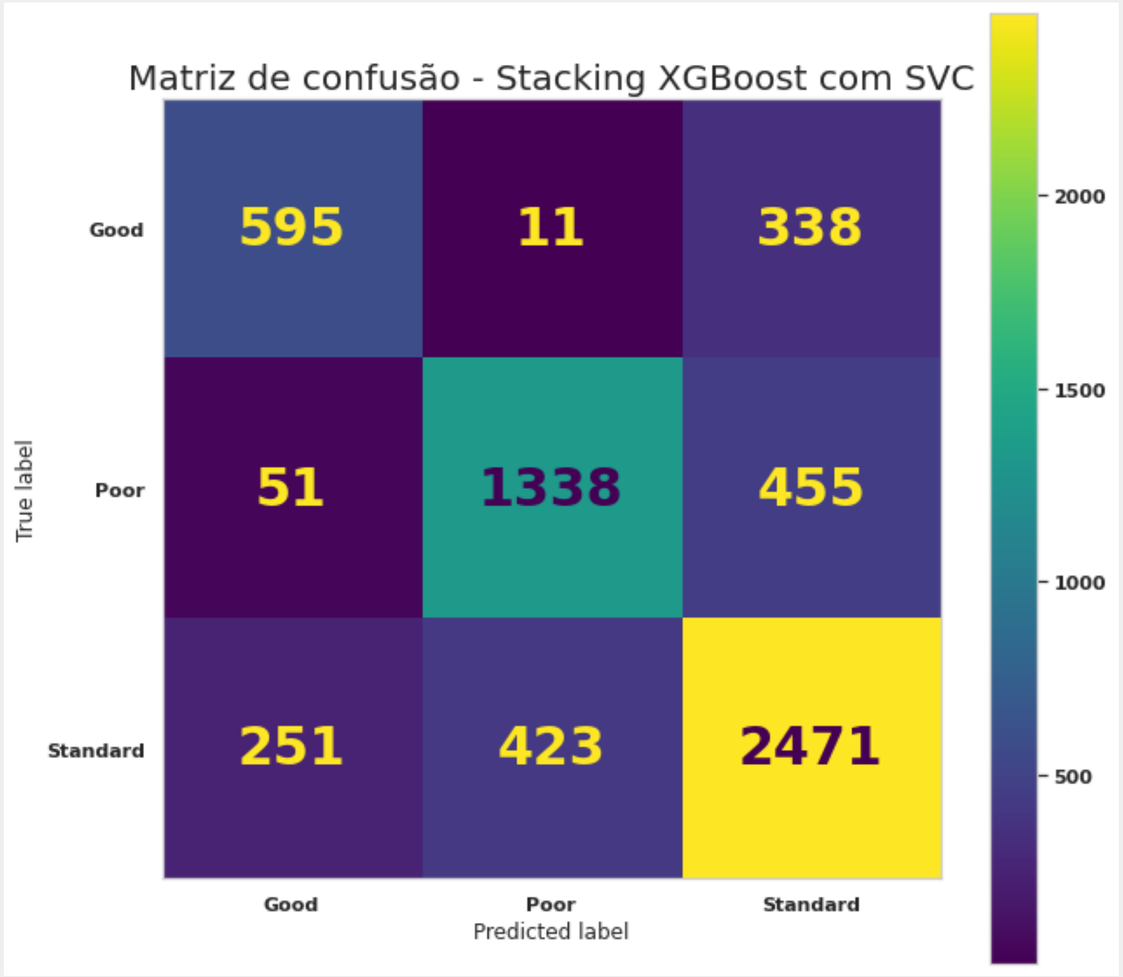

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.94 | 0.92 | 0.93 | 944 |
| 1 | 0.94 | 0.95 | 0.94 | 1844 |
| 2 | 0.95 | 0.95 | 0.95 | 3145 |
| accuracy | | | 0.95 | 5933 |
| macro avg | 0.94 | 0.94 | 0.94 | 5933 |
| weighted avg | 0.95 | 0.95 | 0.95 | 5933 |

# Multi-Layer Perceptron (SKLearn)




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.50 | 0.77 | 0.60 | 944 |
| 1 | 0.68 | 0.64 | 0.66 | 1844 |
| 2 | 0.77 | 0.67 | 0.72 | 3145 |
| accuracy | | | 0.68 | 5933 |
| macro avg | 0.65 | 0.69 | 0.66 | 5933 |
| weighted avg | 0.70 | 0.68 | 0.68 | 5933 |

# Stacking de XGBoost com SVC




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.66 | 0.63 | 0.65 | 944 |
| 1 | 0.76 | 0.73 | 0.74 | 1844 |
| 2 | 0.76 | 0.79 | 0.77 | 3145 |
| accuracy | | | 0.74 | 5933 |
| macro avg | 0.73 | 0.71 | 0.72 | 5933 |
| weighted avg | 0.74 | 0.74 | 0.74 | 5933 |

# Performance

- XGBoost foi o melhor modelo, disparado (>AUC, Acc, F1; <FN);
- RF, ExtraTrees e Stacking tiveram performance similar, com maior AUC de RF;
- Dummy Classifier serve de referência para os demais, principalmente durante a otimização de hiperparâmetros.

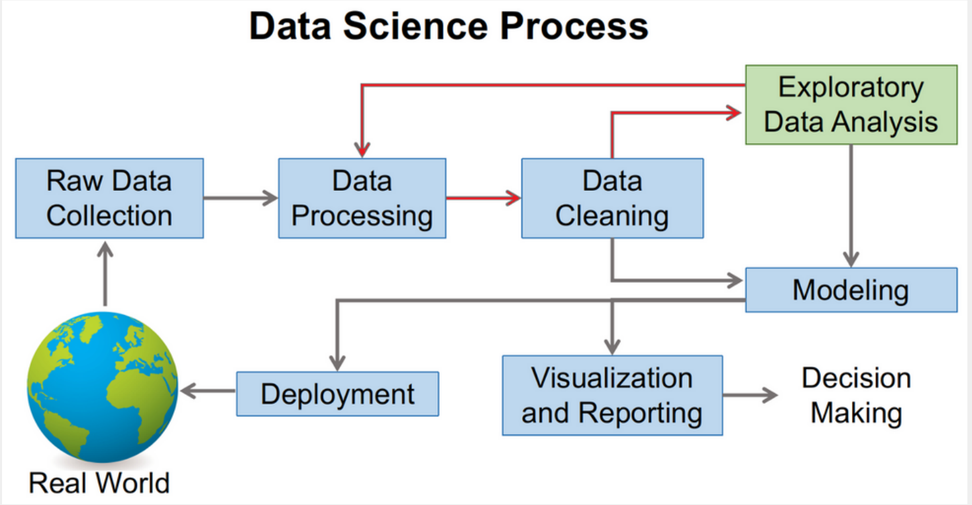
| | Acurácia | Recall | FN | F1-Score | AUROC |
|---|---|---|---|---|---|
| XGBoost | 0.9451 | 0.9385 | 148.0 | 0.9414 | 0.9934 |
| Random Forest | 0.7566 | 0.7378 | 699.0 | 0.7348 | 0.8949 |
| Extra Trees | 0.7477 | 0.7313 | 717.0 | 0.7259 | 0.8910 |
| Stacking | 0.7423 | 0.7139 | 674.0 | 0.7192 | 0.8833 |
| Multi Layer Perceptron | 0.6776 | 0.6938 | 1037.0 | 0.6611 | 0.8949 |
| Dummy | 0.5301 | 0.3333 | 0.0 | 0.2310 | 0.5000 |

- OBS.: Curvas ROC plotadas no esquema One vs Rest (multi-classe). A variável referência escolhida foi a mais frequente, Standard.

# Conclusões

- A classificação Multi-classe se mostrou uma tarefa à parte já que muitos dos métodos e funções de validação/métricas tiveram que ser substituídos ou reescritos. Não somente, a codificação da target seguiu outro método do que já é costumeiro (binário, OneHot etc);

- Dataset com qualidade baixa de dados, porém com quantia grande de observações. EDA, Data Wrangling e Limpeza de Dados foram pontos de foco (vide ciclo iterativo destacado em vermelho).

# Conclusões

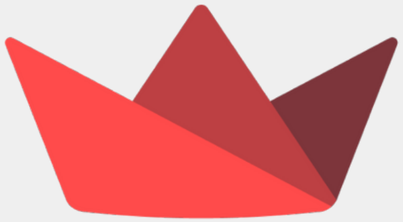
- XGBoost com dados processados, normalizados etc teve melhores resultados do que para com dados crus (fora grande tendência de overfitting neste último);

- MLP foi um modelo que trouxe o pior resultado, devido primeiramente ao desconhecimento do mesmo e seus hiperparâmetros por parte dos autores. Utilização das bibliotecas Keras+Tensorflow poderiam trazer o potencial enorme que as redes neurais apresentam;

- Stacking de XGBoost com SVC (e mesmo quando de XGBoost com MLP), também não rendeu bons resultados, já que não foi possível encontrar modelos de soluções complementares. Talvez com uma MLP melhor otimizada esse seria o melhor modelo disparado.

# StreamLit

- Dashboard interativo como exemplo do funcionamento do modelo. (https://creditscore-projetofinal.streamlitapp.com/)



Streamlit

# Fontes

## O que é Score de Crédito?

https://www.serasa.com.br/score/blog/o-que-e-score-de-credito/

## Como definir o risco de inadimplência de clientes?

https://www.cobrefacil.com.br/blog/risco-de-inadimplencia

## Número de inadimplentes no Brasil atinge recorde, aponta Serasa

https://www.cnnbrasil.com.br/business/numero-de-inadimplentes-no-brasil-atinge-recorde-da-serie-historica-aponta-serasa/

## Dados

https://www.kaggle.com/datasets/parisrohan/credit-score-classification