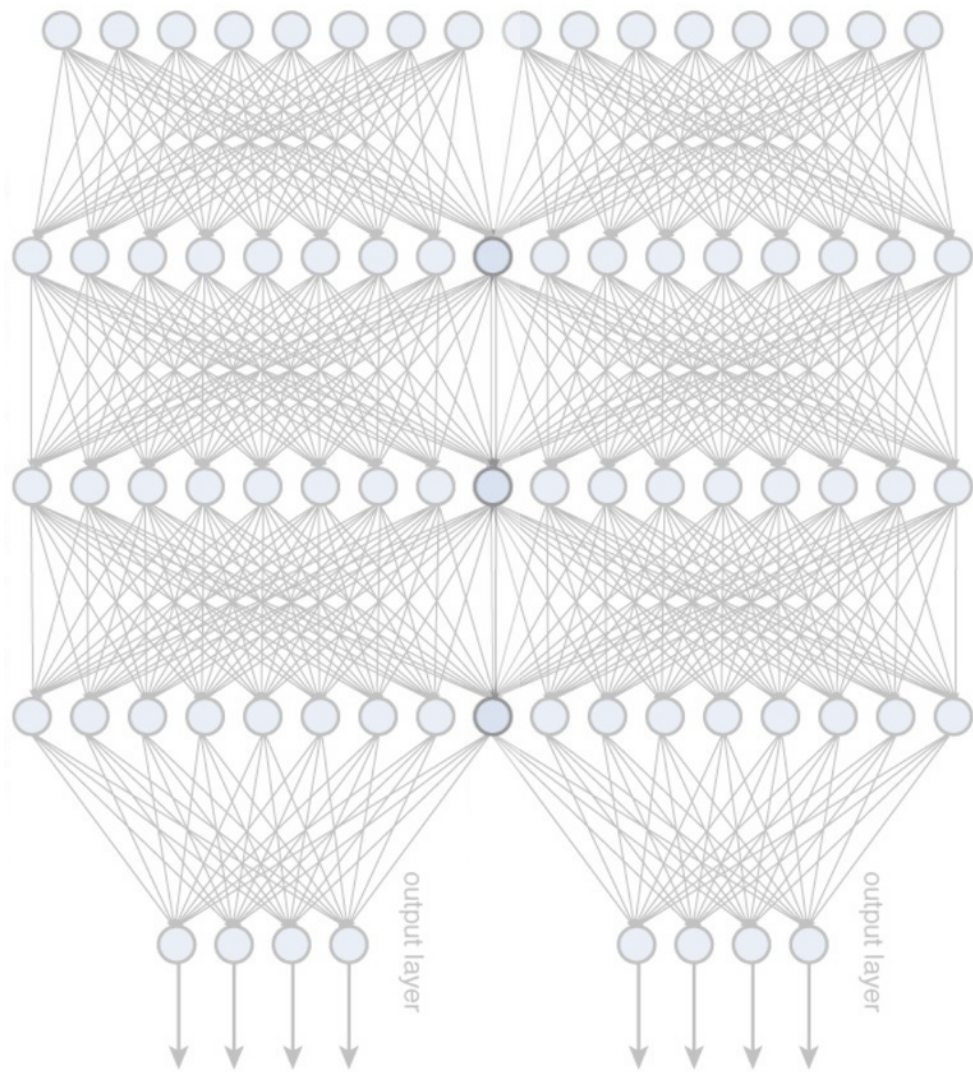




NEUROTECH

**Processo seletivo**

# **Modelo de Concessão de Crédito**



# Índice

- Projeto
- Entendimento dos dados
- Análise exploratória
  - Análise descritiva
  - Análise distribuição geográfica
- Pré processamento
- Treinamento e otimização
- Análise técnica
- Análise financeira
- Escoragem base out-of-time
- Melhorias futuras

# Cientista de dados



**Antônio Ramos**

- Engenheiro de Produção
- Mestrando em Computação Inteligente
- Cientista de Dados
- Atuação em projeto envolvendo IA na área da saúde, indústria, moda, tecnologia, e outros.

# Projeto

# Projeto

**Objetivo:** Desenvolver um modelo de concessão de crédito.

## **Etapas:**

1. Entendimento da base e análise exploratória dos dados.
2. Pré-processamento das variáveis.
3. Treinamento de um modelo de classificação binária.
4. Análise técnica da performance do modelo, medida sobre a base de Teste.
5. Análise financeira do modelo. Para este ponto, observe a subseção `Análise financeira`.
6. Escoragem da base Out-of-time, para posterior avaliação da performance nessa base com o alvo detido pelo Prophet/Neurolake.

## **Entregáveis:**

1. Um Jupyter Notebook legível, contendo as etapas do projeto;
2. Apresentação do projeto;
3. Base Out-of-time com previsões.



# Entendimento dos dados

# Entendimento dos dados

## Dados de treinamento

- **Tamanho da base (linhas,colunas):** (120.750, 151)
- **Início da base:** 02/01/2017
- **Fim da base:** 31/08/2017
- **Cientes únicos:** 120.750
- **Cientes duplicados:** 0
- **Colunas com mais de 65% de valores nulos:** 29
- **Balanceamento das classes:**
  - **0 (Bom pagador):** 91.163 / 75%
  - **1 (Mau pagador):** 29.587 / 25%



# Análise exploratória



# **Análise descritiva**

# Análise descritiva

## Mapeando variáveis potencialmente mais informativas

### 1) Teste t de Student para Variáveis Numéricas:

Apliquei o teste t de Student (independente) para comparar as médias de cada variável numérica entre os grupos de Bom Pagador e Mau Pagador. Essas variáveis são potencialmente mais informativas para o modelo.

### 2) Teste Qui-quadrado para Variáveis Categóricas:

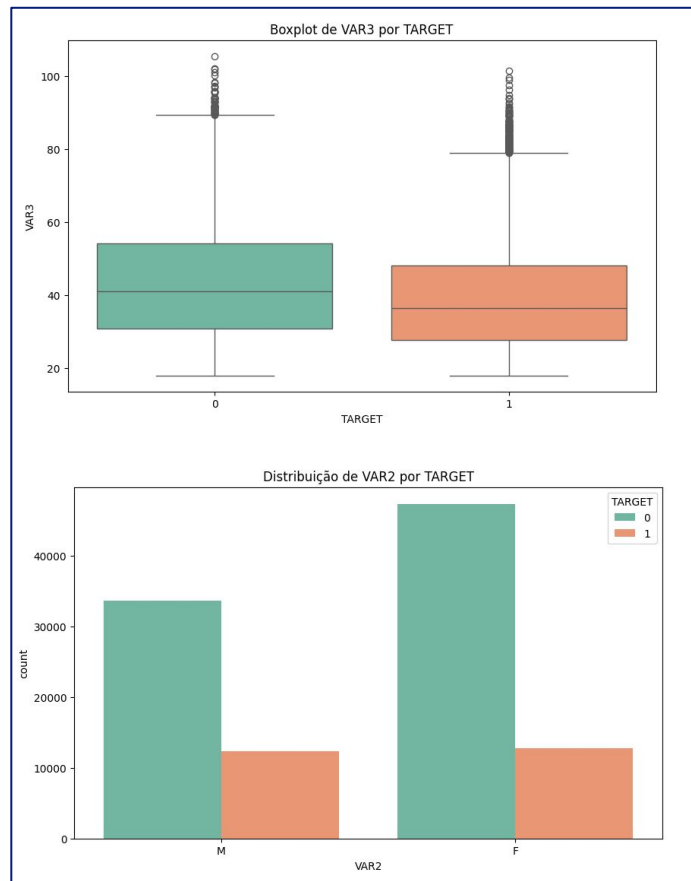
Para as variáveis categóricas, utilizei o teste Qui-quadrado de independência. Este teste é apropriado para verificar se existe uma associação significativa entre a variável categórica e a variável alvo.

### 3) Seleção de Variáveis:

Criei listas de variáveis numéricas e categóricas que demonstraram diferenças ou associações significativas com a variável alvo.

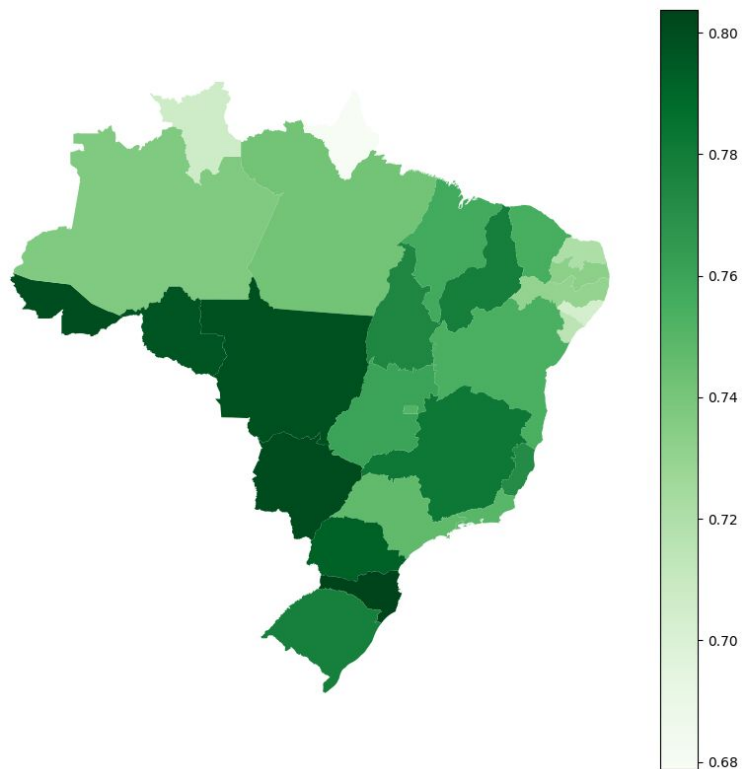
Esta abordagem foca o modelo nas variáveis mais informativas.

## Exemplo de variável numérica e categórica que passaram na análise



# **Análise distribuição geográfica**

# Análise distribuição geográfica



**A análise teve como foco entender como a propensão de ser um bom pagador varia entre diferentes regiões geográficas.**

É possível perceber que, geograficamente, os bons pagadores estão bem disseminados, o que é um ótimo sinal pois não corremos o risco de ter uma viés por parte da localização geográfica.

Porém vale ressaltar que temos alguns estados com maior concentração.

# Pré processamento

# Pré processamento

## **Balanceamento de Classes (Downsampling)**

Separei as observações da classe majoritária (0: Bom Pagador) e minoritária (1: Mau Pagador) e reduzi aleatoriamente a classe majoritária para ter o mesmo número de observações que a classe minoritária.

## **Transformação dos Dados (SimpleImputer, OneHotEncoder, StandardScaler)**

Construí pipelines separados para tratar as variáveis numéricas e categóricas. Para as numéricas, apliquei imputação de média e normalização. Para as categóricas, apliquei imputação constante e codificação one-hot.

## **Seleção de Recursos (SelectFromModel)**

Usei um modelo XGBClassifier com o método SelectFromModel para selecionar as características mais importantes.

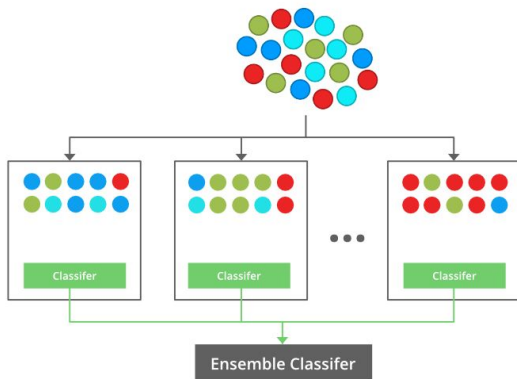


# Treinamento e otimização

# Treinamento e otimização

## XGBClassifier

Escolhi o XGBoost pois ele atua com o bootstrapping, que é uma técnica de reamostragem com substituição. Isso permite que o modelo não seja apenas robusto contra overfitting, mas também maneje a variância de uma maneira que melhore a generalização.



## Otimização de hiperparâmetros

A otimização utilizou uma pesquisa de grade. Defini um conjunto de parâmetros a serem testados, incluindo o número de estimadores, a taxa de aprendizado e a profundidade máxima das árvores.

```
param_grid = {  
    'n_estimators': [100, 200, 300, 500, 1000],  
    'learning_rate': [0.001, 0.01, 0.05, 0.1, 0.3],  
    'max_depth': [3, 4, 5, 8, 10],  
}
```

### Melhores parâmetros:

- learning\_rate: 0.05
- max\_depth: 5
- n\_estimators: 500

### Melhor AUC-ROC no conjunto de teste:

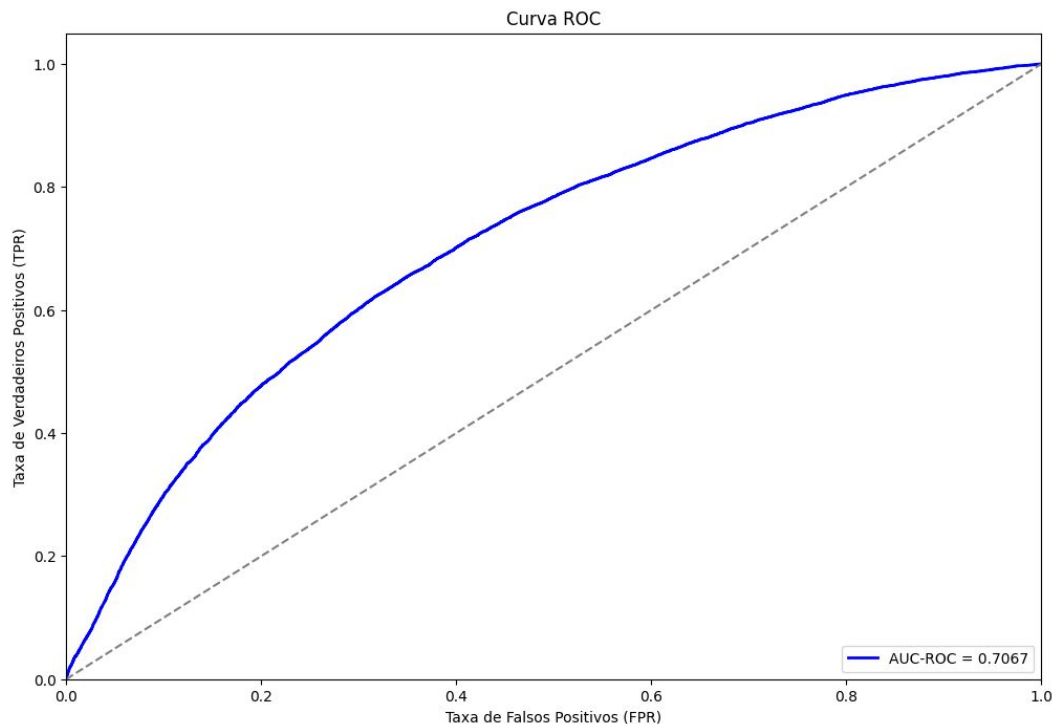
0.7067





# **Análise técnica**

# Análise técnica



A área sob a curva (**AUC-ROC**) de aproximadamente **0.7067** é um indicador de que o modelo tem uma **capacidade razoável de discriminar entre as classes de bom e mau pagador**

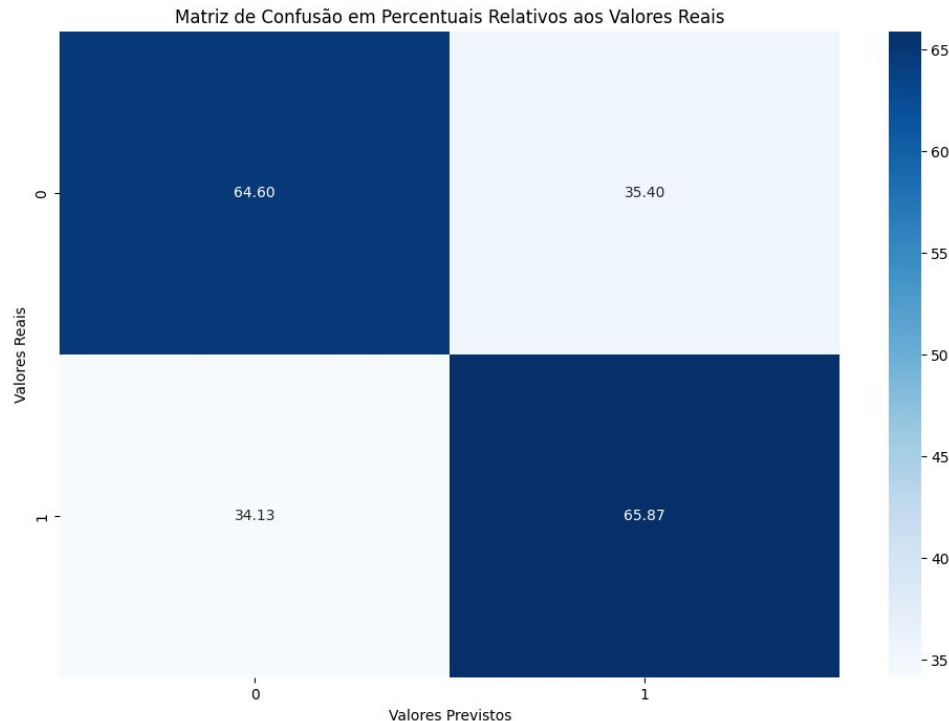
Este valor é superior ao que seria obtido por um modelo que faz escolhas ao acaso (AUC-ROC = 0.5, indicado pela linha pontilhada), **mas ainda há espaço para melhoria.**



# Análise técnica

O modelo acerta **64.60%** dos verdadeiros negativos (Bom pagador) e **65.87%** dos verdadeiros positivos (Mau pagador). Isto indicar que não há um viés forte em favor de uma classe sobre a outra.

No entanto, a taxa de erro também é notável, com **35.40%** dos negativos verdadeiros previstos incorretamente e **34.13%** dos positivos verdadeiros.



# **Análise financeira**

# Análise financeira

## Política AS-IS

O ponto de corte são **clientes com 28 anos ou menos.**

Carteira de crédito aprovado para o mês de agosto de 2017 foi de **R\$5.432.000.**

## Política TO-BE

O ponto de corte é o cliente ter uma **probabilidade igual ou maior a 59,28% (Mediana da probabilidade) de ser bom pagador.**

Carteira de crédito aprovado para o mês de agosto de 2017 foi de **R\$3.653.000.**

A política TO-BE pode estar contribuindo para uma carteira de maior qualidade ao evitar empréstimos a clientes com maior probabilidade de inadimplência. Porém, ainda é preciso melhorar a qualidade do modelo para tornar a regra mais robusta.



# **Escoragem base out-of-time**

# Escoragem base out-of-time

```
y_out_pred = xgb_best.predict(X_out)
y_out_proba = xgb_best.predict_proba(X_out)

[ ] df_out['TARGET'] = y_out_pred
df_out['proba_bom_pagador_label_0'] = y_out_proba[:,0]
df_out['proba_mau_pagador_label_1'] = y_out_proba[:,1]

[ ] df_out = df_out[['ID', 'TARGET', 'proba_bom_pagador_label_0', 'proba_mau_pagador_label_1']]
df_out.head()
```

	ID	TARGET	proba_bom_pagador_label_0	proba_mau_pagador_label_1
0	61900	1	0.454969	0.545031
1	300199	0	0.732929	0.267071
2	45656	0	0.804951	0.195049
3	153386	1	0.477663	0.522337
4	321676	0	0.797648	0.202352

```
[ ] df_out.TARGET.value_counts()

0    61700
1    30265
Name: TARGET, dtype: int64

[ ] df_out.to_csv('/content/drive/MyDrive/Neurotech - Cientista de Dados III/challenge-data-scientist/datasets/credit_01/df_out_with_target.csv', index=False)
```

**61.700**  
**Bons pagadores**

**30.265**  
**Maus pagadores**



# Melhorias futuras



# Melhorias futuras

Em virtude do tempo para entrega do desafio, faltaram o teste de algumas técnicas que podemos pontuar como melhorias futuras para o modelo.

## 1. Rebalanceamento de Classes:

- Atualmente, o modelo pode estar sofrendo de um desequilíbrio de classes, o que pode ser abordado por técnicas de rebalanceamento mais avançadas, como **SMOTE** ou **CTGAN**, para garantir que o modelo não seja enviesado em favor da classe majoritária.

## 2. Avaliação de Hiperparâmetros:

- Podemos também utilizar técnicas mais sofisticadas de otimização de hiperparâmetros, como **pesquisa aleatória** ou **otimização bayesiana**, que podem ser mais eficientes que a pesquisa em grid e encontrar um conjunto de hiperparâmetros mais ótimo.

## 3. Validação Cruzada:

- Não conseguimos implementar uma estratégia de **validação cruzada k-fold** para uma avaliação mais robusta do desempenho do modelo, o que pode ajudar a identificar se a variação no desempenho é devida a peculiaridades do conjunto de teste ou a uma característica do modelo.



# Melhorias futuras

## 4. Testar outros modelos:

- Rodar o **pycaret** para identificar o desempenho dos dados em outros modelos, como: **Gradient Boosting Machines (GBM)**, **LightGBM** e **CatBoost**, **regressões**, **Decision tree** e outros.

## 5. Interpretabilidade do Modelo:

- Aplicação das técnicas como **SHAP** (SHapley Additive exPlanations) ou **LIME** (Local Interpretable Model-Agnostic Explanations) para entender melhor como as variáveis estão afetando as previsões do modelo, o que poderia ajudar na identificação de possíveis vieses ou na melhoria da interpretabilidade do modelo.

## 6. Estratégias de Ensemble:

- Investigação da possibilidade da utilização de técnicas de ensemble, como **stacking** ou **blending**, para combinar o **XGBoost** com outros modelos e verificar se há ganhos de desempenho.





NEUROTECH