

PROJETO

Aprendizado de Máquina

Integrantes

Equipe 9

- Lucas Araujo Bourguignon
- Lucas Nascimento Brandão
- Luiz Eduardo de Freitas Von Schmalz
- Vinicius Seabra Lago Lima

Entendimento do negócio

O **problema** central do nosso projeto está relacionado à análise de pedidos de empréstimo bancário. Muitas instituições financeiras enfrentam dificuldades para identificar, de forma rápida e precisa, quais clientes têm maior risco de inadimplência. Isso pode resultar em prejuízos financeiros, demora no processo de aprovação e concessão de crédito a perfis inadequados.

Nosso **objetivo** é desenvolver um modelo preditivo que auxilie na tomada de decisão, reduzindo esses riscos e otimizando a concessão de crédito de forma mais segura e eficiente.



Entendimento do negócio

Motivação:

- Alta relevância para o setor financeiro.
- Necessidade de reduzir inadimplência e fraudes.
- Otimização dos processos de análise de crédito.

Metas:

- Criar um modelo preditivo eficiente.
- Reduzir riscos na concessão de crédito.
- Automatizar e agilizar a análise de pedidos.



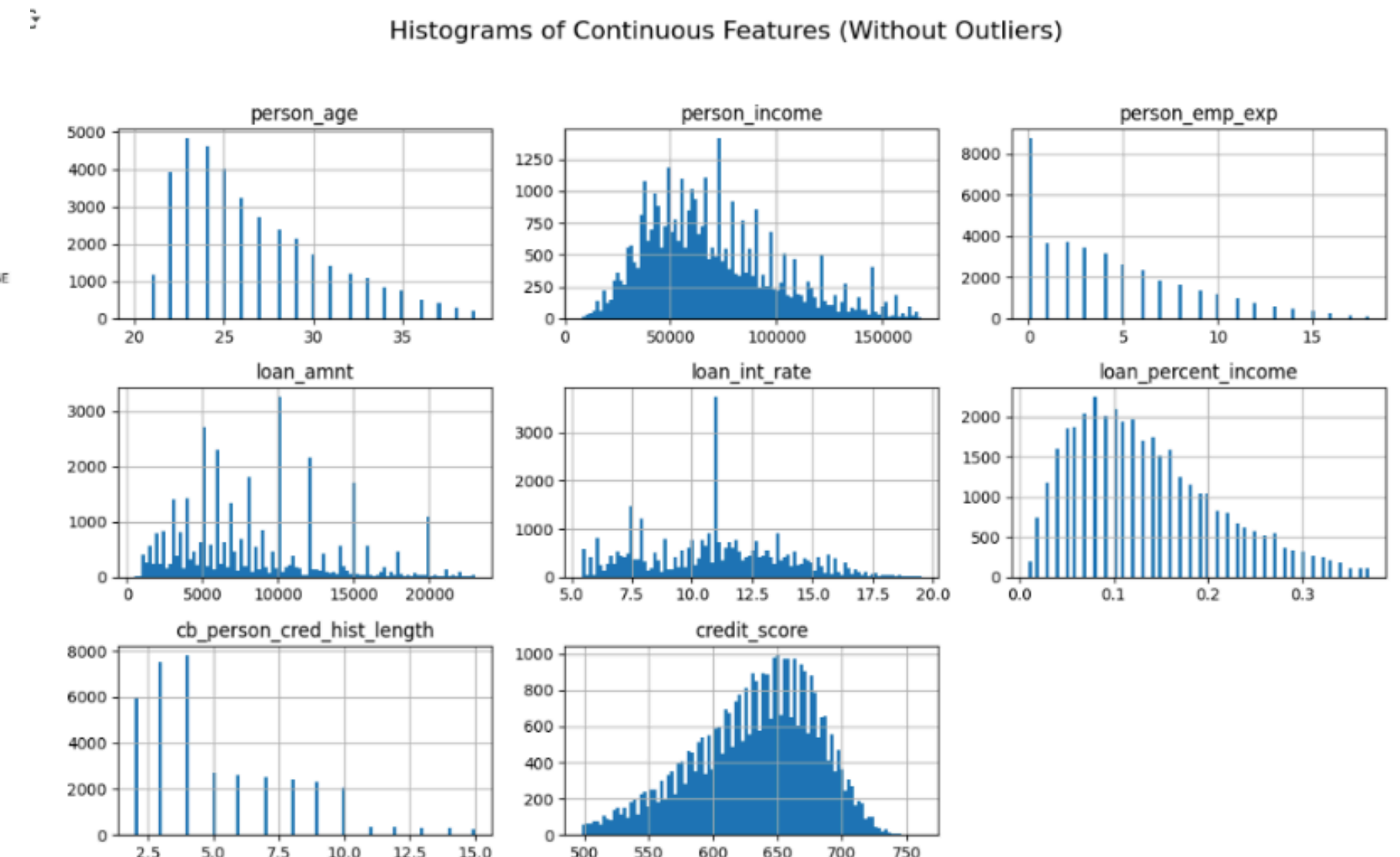
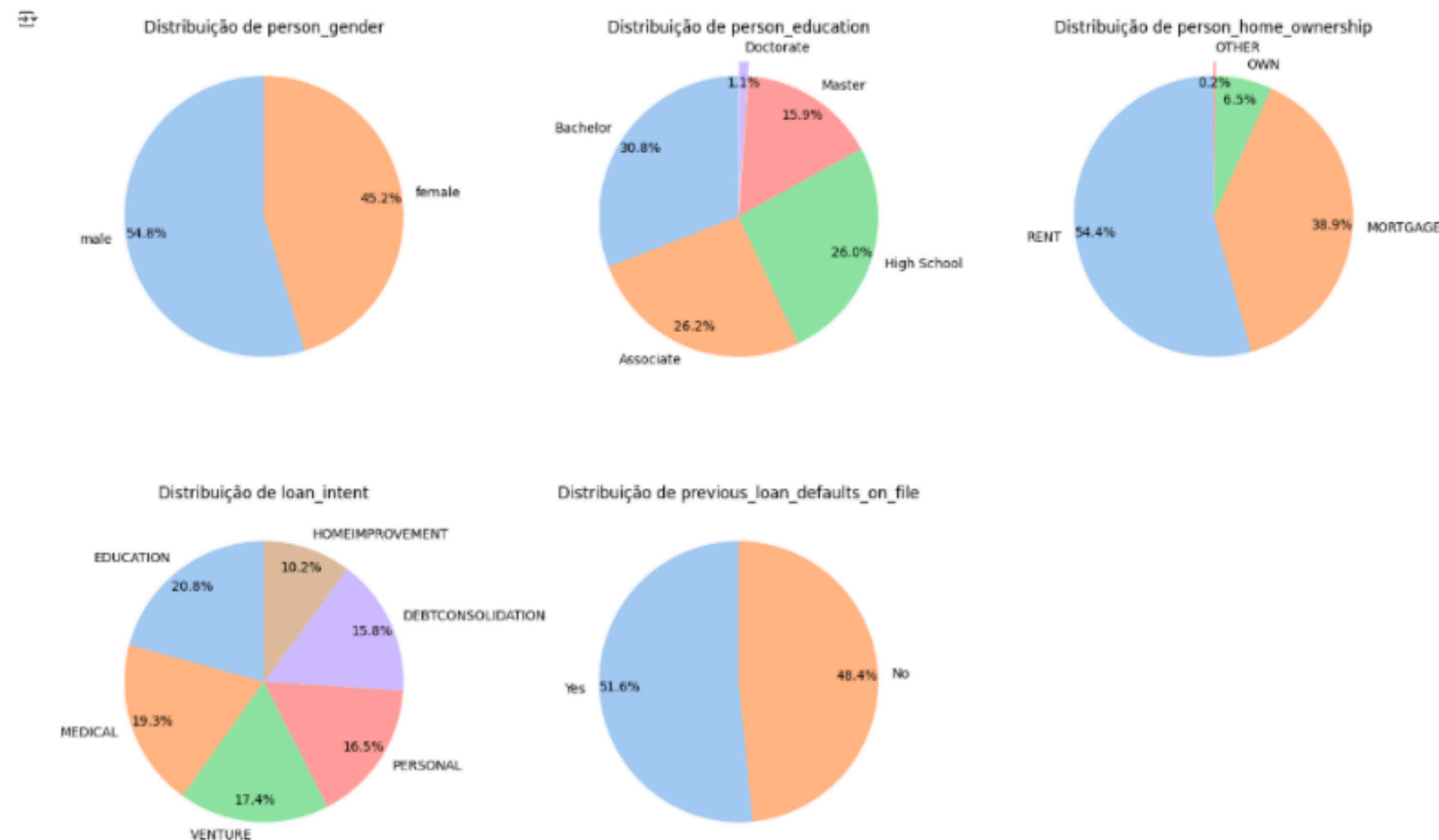
Entendimento dos dados

Descrição da base de dados

- 13 features
 - 8 features numéricas
 - 5 features categóricas
- 45000 registros

Representação gráfica das features categóricas:

Representação gráfica das features numéricas:



Entendimento dos dados

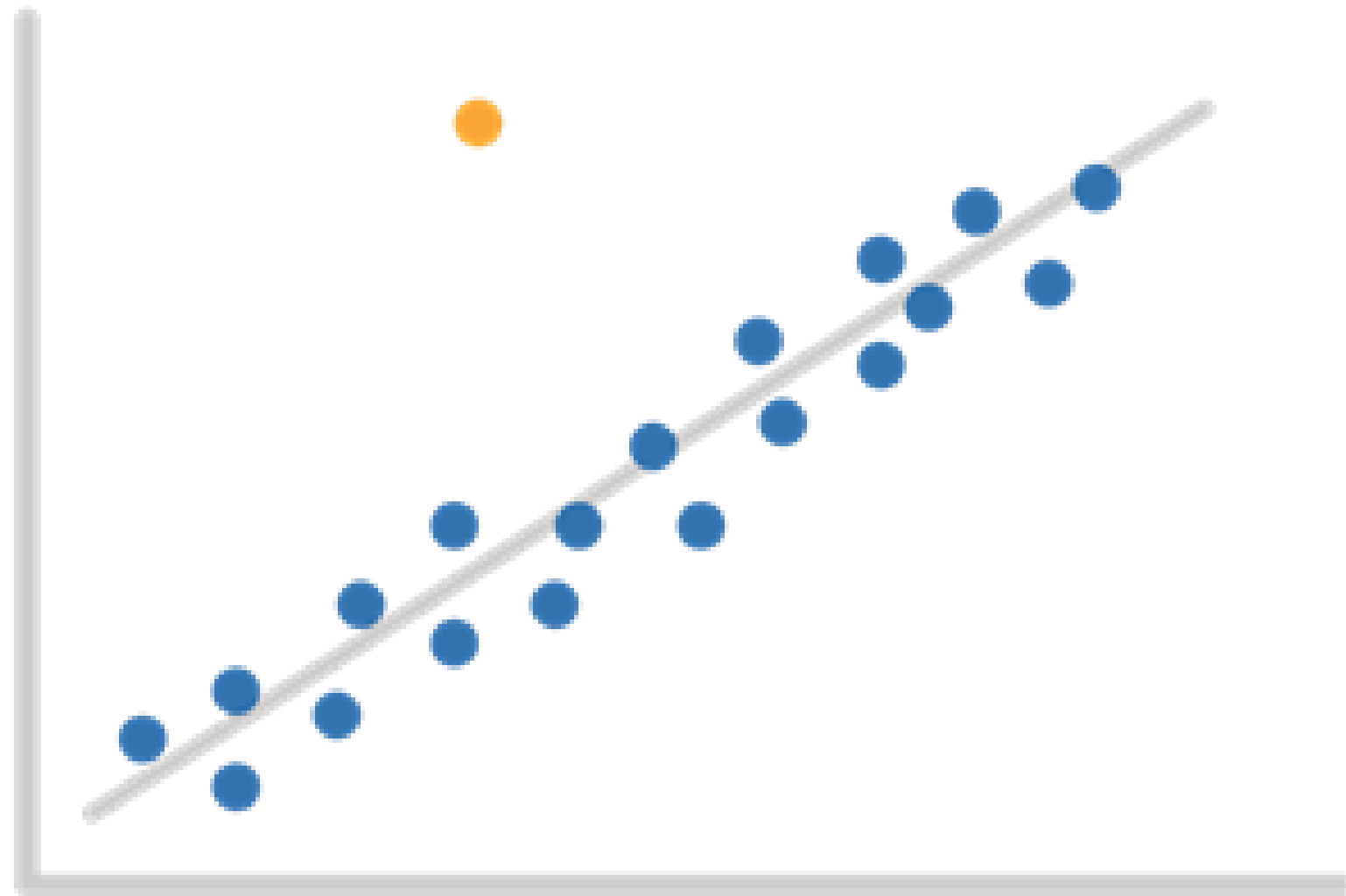


Limpeza da base de dados

Dados ausentes	Não
Presença de outliers	Sim
Ocorrência de duplicatas	Não
Desbalanceamento das classes	Sim

- Base de dados consideravelmente “limpa”
- Presença de uma porcentagem relevantes de outliers entre as colunas, além do desbalanceamento das classes, que serão tratados na próxima etapa

Tratamento de dados



- O primeiro passo no tratamento de dados foi a substituição dos outliers pela média das colunas, o resultado dessa limpeza foi:

Outliers por coluna (método IQR):

person_age: 2188 outliers
person_income: 2218 outliers
person_emp_exp: 1724 outliers
loan_amnt: 2348 outliers
loan_int_rate: 124 outliers
loan_percent_income: 744 outliers
cb_person_cred_hist_length: 1366 outliers
credit_score: 467 outliers

Tratamento de dados

Balanceamento do dataset

- Fizemos uma verificação de quantas classes existiam dentro do dataset e esse foi o resultado:

Distribuição das classes (%):

loan_status

0 77.777778

1 22.222222

Name: proportion, dtype: float64

- Presença de grande desbalanceamento dos dados
- Foi aplicado o método *random oversampling*, que balanceia as classes de forma que o treinamento não será afetado pela quantidade de ocorrências de cada classe:

Nova distribuição das classes após Random

Oversampling:

Classe 1: 50.00%

Classe 0: 50.00%

Tratamento de dados

Categorizando colunas

- O próximo passo no tratamento foi categorizar certas colunas como idade, educação, tipo de moradia, crédito antigo etc.
- O objetivo desse passo é dividir colunas como idade em 'categorias' como jovem, adulto e idoso. Para isso consideramos um range de idades e trocamos os valores de idade por valores numéricos representando as categorias.

```
bins = [0, 20, 30, 100]
labels = [0, 1, 2]
df_tratado['age_category'] =
pd.cut(df_tratado['person_age'], bins=bins,
labels=labels, right=True)
```

age_category

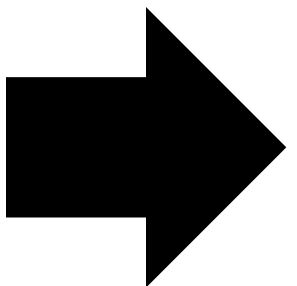
0	1
1	1
2	1
3	1
4	1

Tratamento de dados

Representação de Categorias Não Ordenadas

- Colunas sem ordem natural (ex: tipo de moradia, faixas etárias) não devem ser tratadas como sequenciais.
- Utiliza-se codificação one-hot para representar essas variáveis.
- Cada categoria se torna uma nova coluna.
- Cada linha recebe:
 - 1 na coluna da sua categoria.
 - 0 nas demais colunas.
- Exemplo: "tipo de moradia" vira "Aluguel", "Proprietário", "Hipoteca" e "Outros".

INDEX	PERSONAL EDUCATION
0	PERSONAL EDUCATION
1	MEDICAL
2	VENTURE
3	VENTURE
4	VENTURE



Tratamento de dados

Exemplo pós codificação one-hot

INDEX\LOAN_INTENT	PERSONAL	EDUCATION	MEDICAL	VENTURE	HOME IMPROVEMENT	DEBT CONSOLIDATI ON
0	0	1	0	0	0	0
1	0	0	1	0	0	0
2	0	0	0	1	0	0
3	0	0	0	1	0	0
4	0	0	0	1	0	0

Tratamento de dados

Normalização dos Valores Numéricos

- Depois de passar por todos esses passos, nos resta normalizar o dataset para garantir que os valores numéricos tenham o mesmo peso durante o treinamento e assim acabamos nosso tratamento de dados. Para a normalização, utilizamos o método MIN-MAX







Tratamento de dados

Exemplo pós normalização

INDEX\NORMALIZED COLUMNS	loan_amnt	loan_int_rate	loan_percent_inco me	cb_person_cred_hist_len gth
0	0.361767	0.748059	0.364746	0.076923
1	0.022173	0.403670	0.216216	0.000000
2	0.221729	0.525759	0.364746	0.076923
3	0.361767	0.692308	0.364746	0.000000
4	0.361767	0.624559	0.364746	0.153846

Tratamento de dados

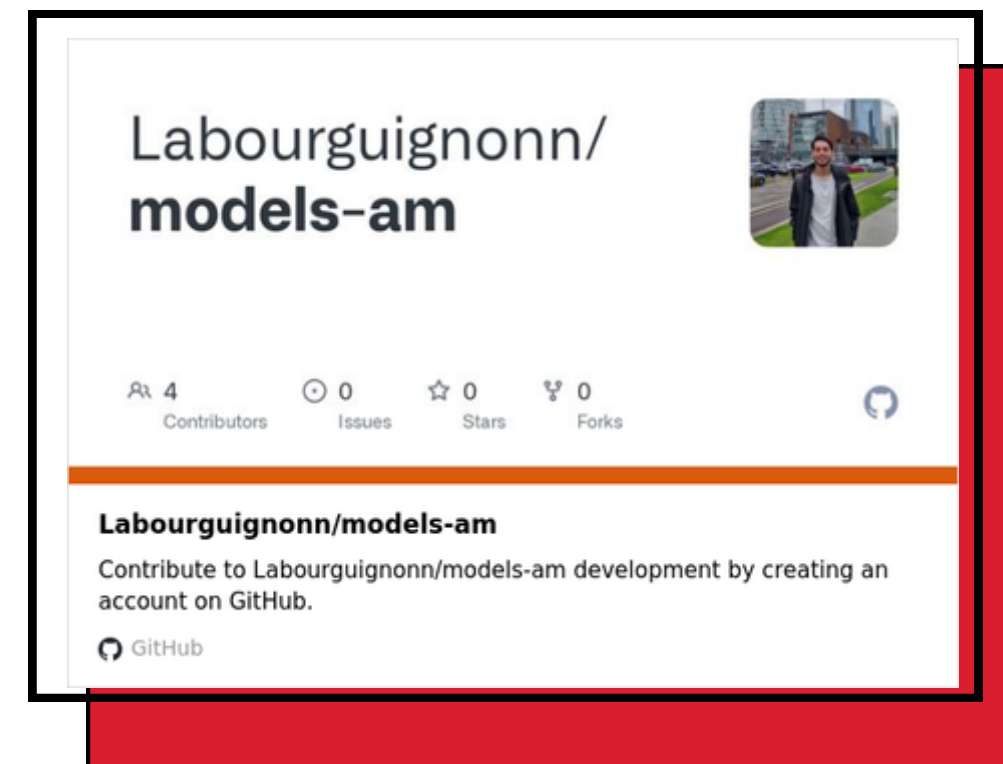
- Por fim, separamos o dataset em treino, validação e teste, para garantir que durante os experimentos os mesmos dados fossem utilizados.

 Labourguignonn updated-lucas-files	
 modelos	updated-lucas-files
 README.md	Initial commit
 test_normalized.csv	atualização dos datasets
 train_normalized.csv	atualização dos datasets
 val_normalized.csv	atualização dos datasets

Modelagem

Modelos escolhidos
LVQ
MLP
SVM
Decision tree
Heterogeneous ensemble
KNN
LightGBM
nn comitee
Randon florest
Xgboost

Vamos analisar os modelos com melhor desempenho, são eles: **Árvore de decisão, SVM, XGBoost, LightGBM.**



 Repositório do projeto

Árvore de Decisão

Árvore de Decisão é um algoritmo supervisionado de aprendizado de máquina utilizado para classificação e regressão.

Espaço de busca

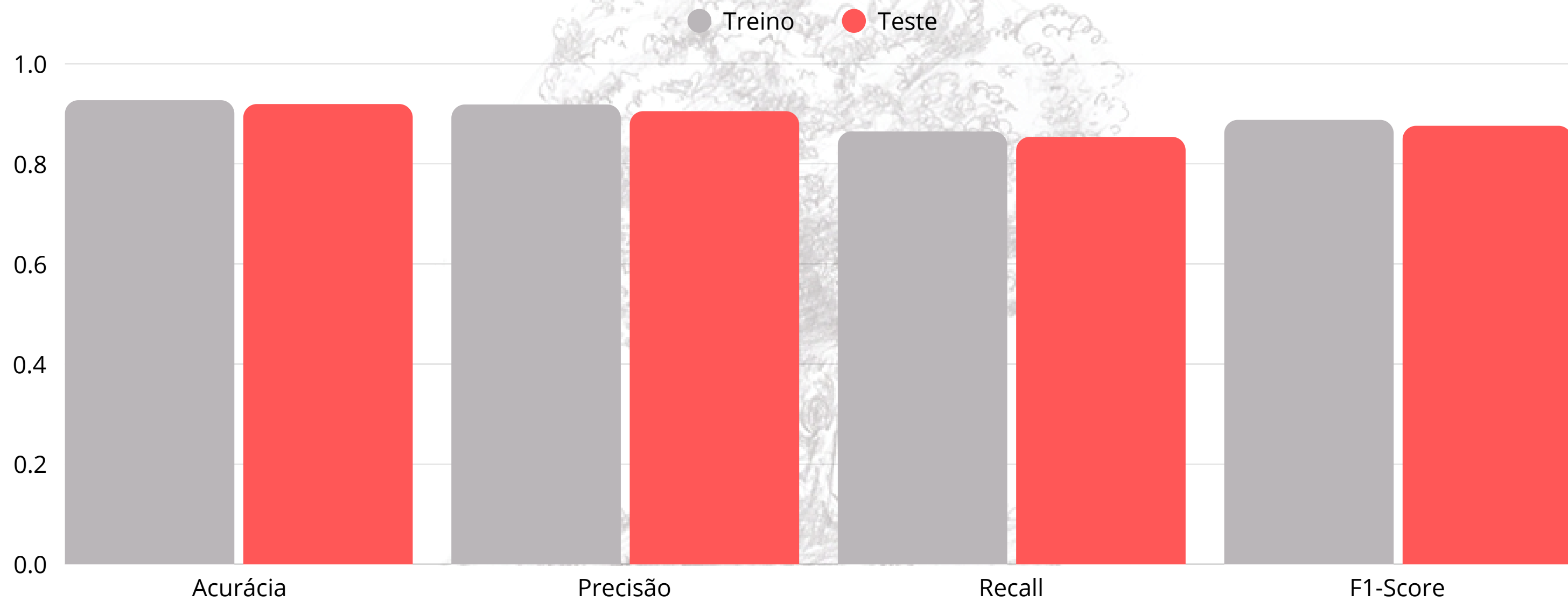
`max_depth: randint(1, 21),` `min_samples_split: randint(2, 21),`
`min_samples_leaf: randint(1, 21),` `criterion: ['gini', 'entropy'],`

Melhores hiperparâmetros encontrados

`max_depth: randint(1, 21),` `min_samples_split: randint(2, 21),`
`min_samples_leaf: randint(1, 21),` `criterion: ['gini', 'entropy'],`

Árvore de Decisão

Melhor média de acurácia nos folds: 0.9163



SVM

O modelo SVM (Máquinas de Vetores de Suporte) é um algoritmo de que busca encontrar o hiperplano que melhor separa as classes de dados, maximizando a margem entre os pontos de classes diferentes.

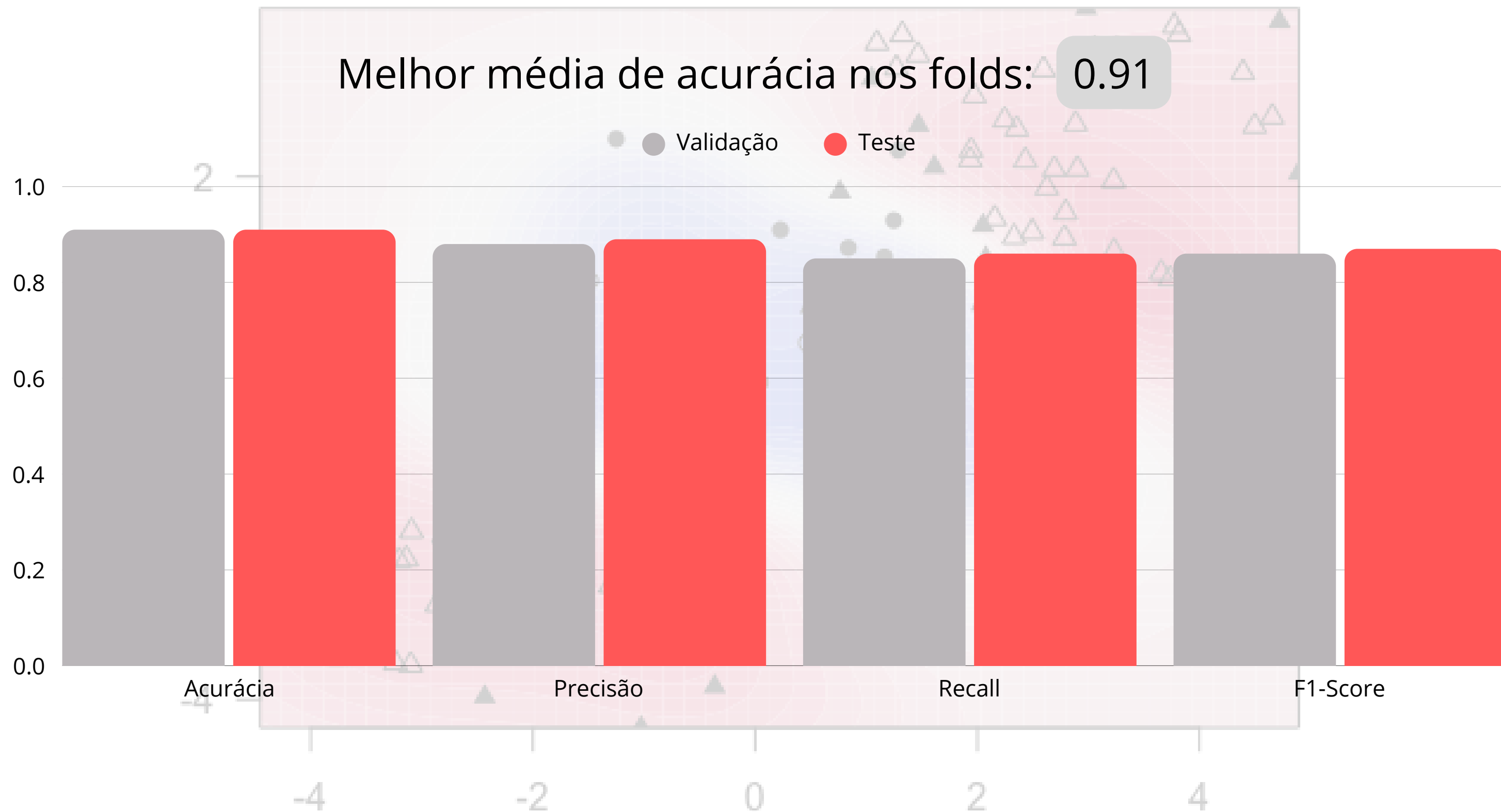
Espaço de busca

C: (1e-3, 1e3), **gamma:** (1e-3, 1e1),
kernel: ['linear', 'rbf']

Melhores hiperparâmetros encontrados

C: 9.4435, **gamma:** 0.01765,
kernel: 'rbf'

SVM



XGBoost

XGBoost (Extreme Gradient Boosting) é baseado em árvores de decisão que utiliza a técnica de boosting para combinar vários modelos fracos, corrigindo erros iterativamente e otimizando o desempenho por meio de regularização e paralelização.

Espaço de busca

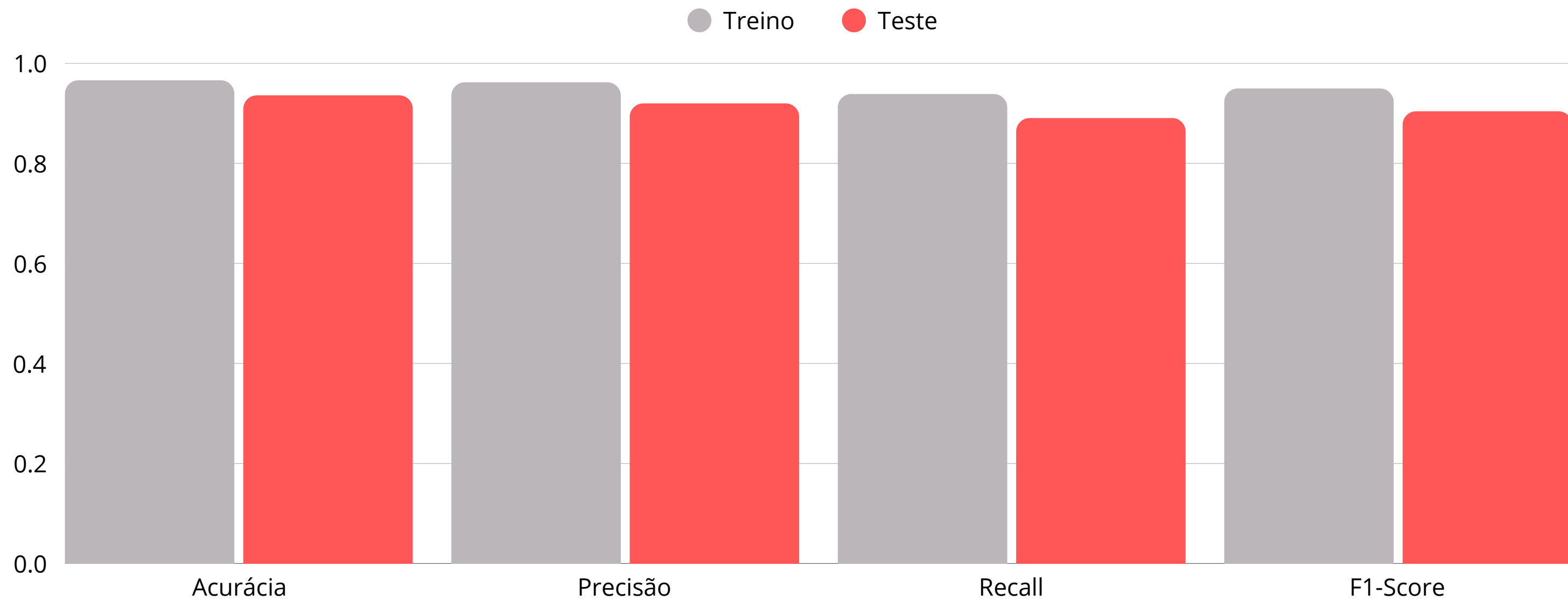
n_estimators: (50, 300), **max_depth:** (3, 10), **learning_rate:** (0.01, 0.3),
subsample: (0.5, 0.5), **colsample_bytree:** (0.5, 0.5), **gamma:** (0, 0.5)

Melhores hiperparâmetros encontrados

n_estimators: 235, **max_depth:** 7, **learning_rate:** 0.14152,
subsample: 0.76924, **colsample_bytree:** 0.5381, **gamma:** 0.3899

XGBoost

Melhor média de acurácia nos folds: 0.9303



LightGBM

LightGBM (Light Gradient Boosting Machine) é baseado em árvores desenvolvido para ser rápido e eficiente, que usa técnicas como histogramas e crescimento de árvore por folhas (leaf-wise) para melhorar a velocidade e a acurácia em grandes volumes de dados.

Espaço de busca

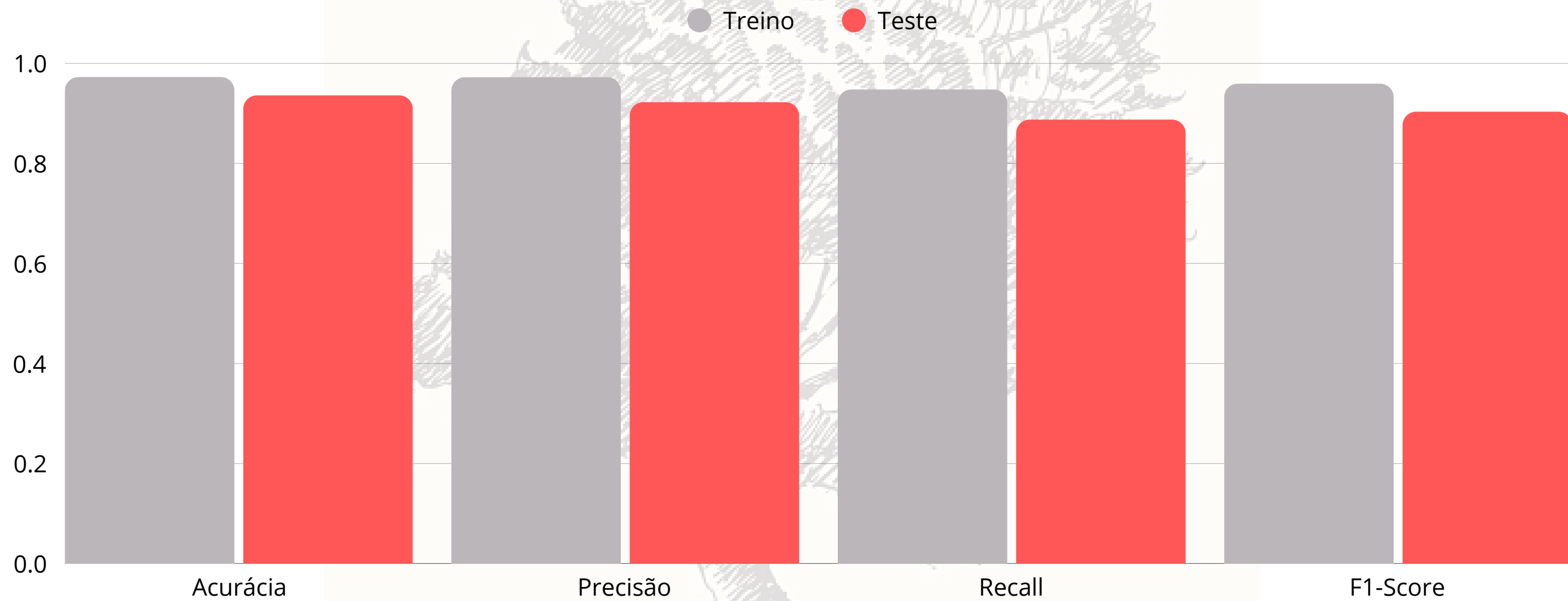
`num_leaves`: (20, 150), `max_depth`: (1, 21), `learning_rate`: [0.01, 0.05, 0.1, 0.2],
`n_estimators`: (50, 200), `subsample`: (0.6, 1.0), `colsample_bytree`: (0.6, 1.0)

Melhores hiperparâmetros encontrados

`num_leaves`: 139, `max_depth`: 13, `learning_rate`: 0.05,
`n_estimators`: 183, `subsample`: 0.6, `colsample_bytree`: 0.6

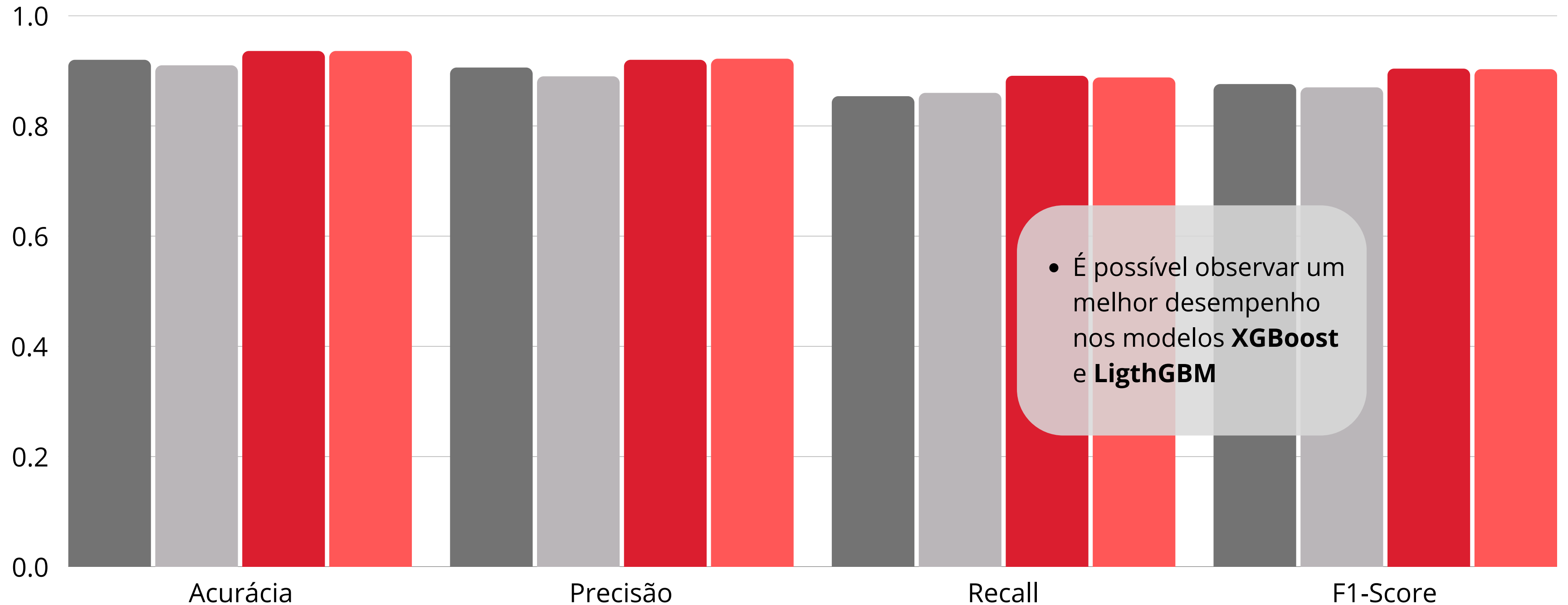
LightGBM

Melhor média de acurácia nos folds: 0.9316



Avaliação

● DT ● SVM ● XGBoost ● LightGBM



- É possível observar um melhor desempenho nos modelos **XGBoost** e **LightGBM**

Conclusão

Concluimos que, por meio de um cuidadoso processo de tratamento de dados e avaliação de diversos algoritmos de machine learning, foi possível desenvolver modelos preditivos robustos e eficientes para apoiar a análise de pedidos de empréstimo bancário.

Dentre os modelos testados, o **XGBoost** e o **LightGBM** se destacaram, alcançando as melhores métricas de desempenho.



Lucas Araujo Bourguignon
lab9@cin.ufpe.br



Lucas Nascimento Brandão
lnb@cin.ufpe.br



Luiz Eduardo Schmalz
lefvs@cin.ufpe.br



Vinícius Seabra Lago Lima
vsll@cin.ufpe.br