Analytics Skills Accelerator

Estudo de Caso: Placas de Aço com Defeito

Luiza Batista Laquini

Cenário

Base de dados: Placas produzidas no lingotamento contínuo em que ocorreram defeitos:

- tipo 0
- tipo 1

Queremos predizer o tipo do defeito.
Problema de classificação! - Aprendizado Supervisionado.

Dicionário de Dados

Caracterização:

- type_of_steel – Identifica a classe do aço: A300 ou A400 -> object
- defect_type – Tipo de defeito da classe. Pode ser do tipo 0 ou do tipo 1 -> int (coluna alvo)

Coordenadas do defeito:

- min_x_defect – Coordenada x inicial do defeito -> float
- max_x_ defect – Coordenada x final do defeito -> float
- min_y_ defect – Coordenada y inicial do defeito -> float
- max_y_ defect – Coordenada y final do defeito -> float

Dicionário de Dados

Medidas das Placas:

- slab_width – Largura da placa (eixo X) -> float
- slab_length – Comprimento da placa (eixo Y) -> float
- slab_thickness - Espessura da placa (eixo Z) -> float

Referente aos Pixels:

- area_pixels – Total de pixels presentes na placa -> float
- sum_pixel_luminosity – Soma da luminosidade dos pixels -> float
- min_pixel_luminosity – Mínima luminosidade dos pixels -> float
- max_pixel_luminosity – Máxima luminosidade dos pixels -> float

Dicionário de Dados

Medidas das Esteiras:

• conveyer_width – Largura da esteira (correia) transportadora (eixo X)

-> float

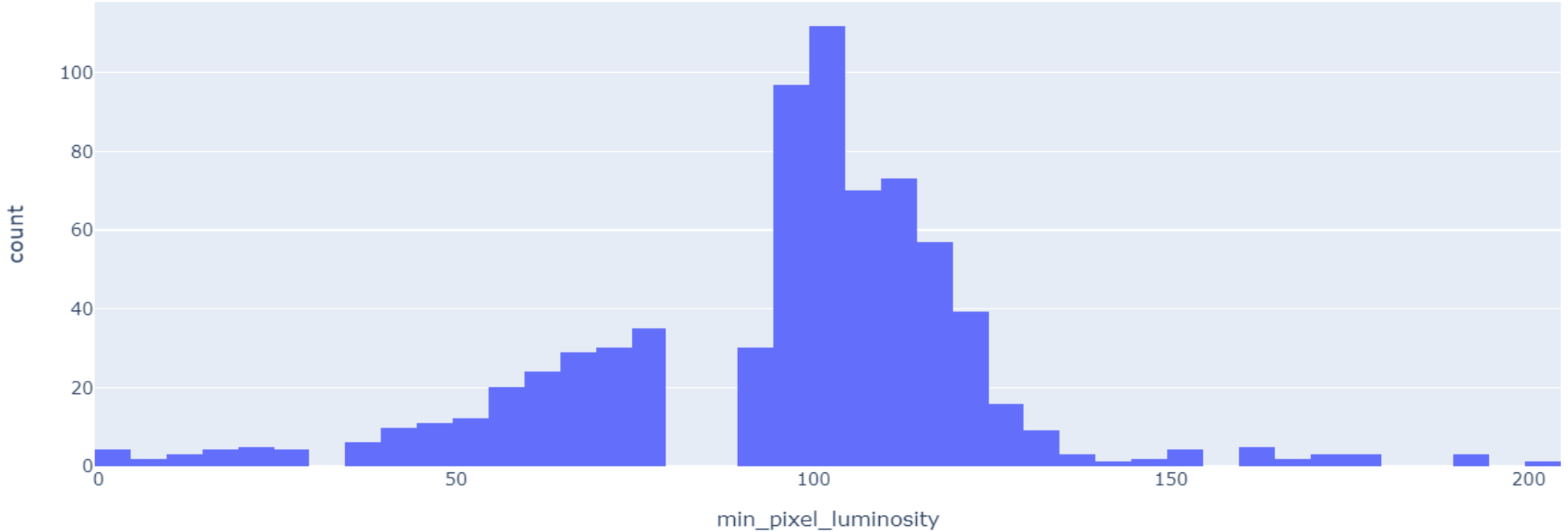
Esses dados formam uma tabela de **967 linhas x 14 colunas**

Integridade dos Dados

- **Nulos**

Apenas uma coluna com nulos: min_pixel_luminosity

- 238 valores nulos (aprox. 24,61% dos dados)



Integridade dos Dados

- **Nulos**

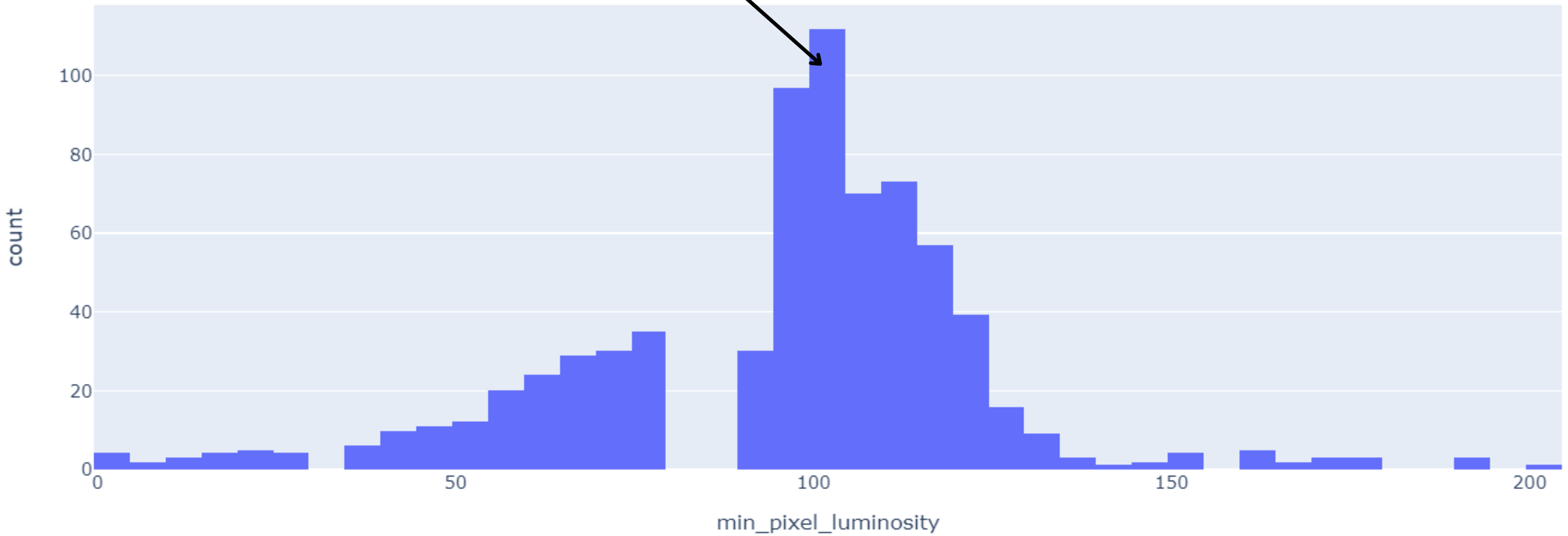
Alternativas:
- Deletar as linhas com os valores nulos ❌
- Preencher com zeros ❌
- bfill ou ffill ❌
- Preencher com a média
- Preencher com a mediana

} ?

Integridade dos Dados

- **Nulos**

Média = 95.78 x Mediana = 101



Integridade dos Dados

- ***Outliers***

Abordagem IQR = muitos *outliers*

Observei que os valores mais extremos eram casos particulares possíveis.

Na ausência de informações (unidades de medida) para julgar o que era erro de medição, trato todas as medições como corretas.

Busca por *outliers* multivariados.

Integridade dos Dados

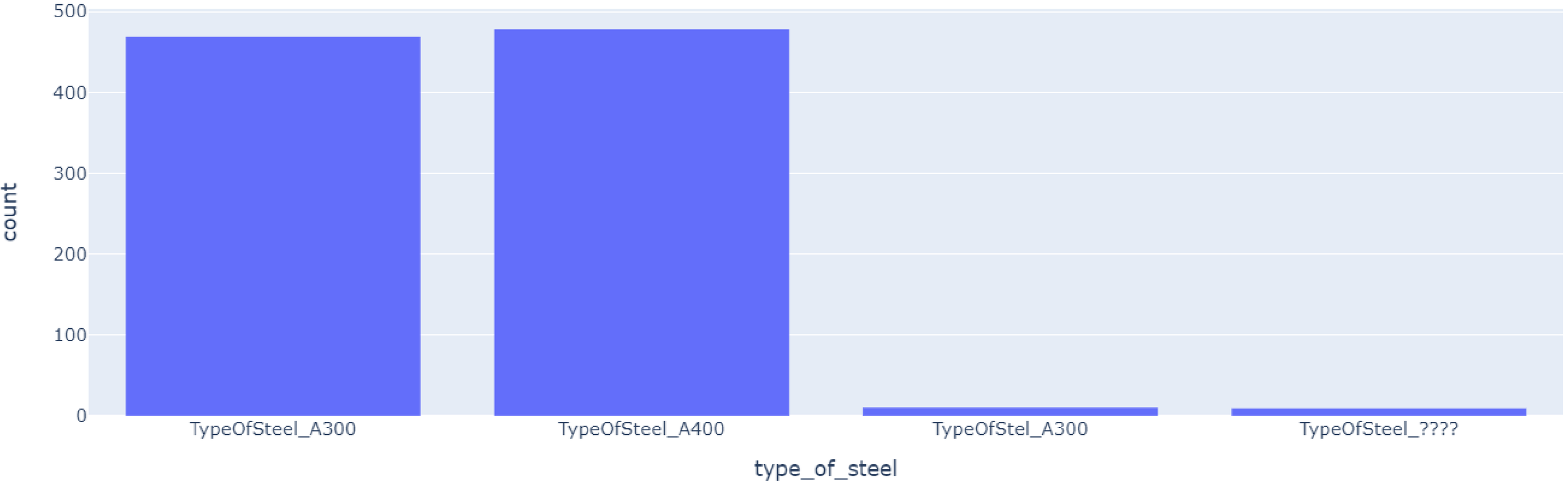
- ***Outliers***

Busca por *outliers* multivariados (exemplos):

- Não pode haver uma placa maior que a esteira que a transporta
- Não pode haver uma placa com luminosidade mínima maior que a luminosidade máxima
- Não pode haver uma coordenada de defeito em um ponto que esteja fora da placa

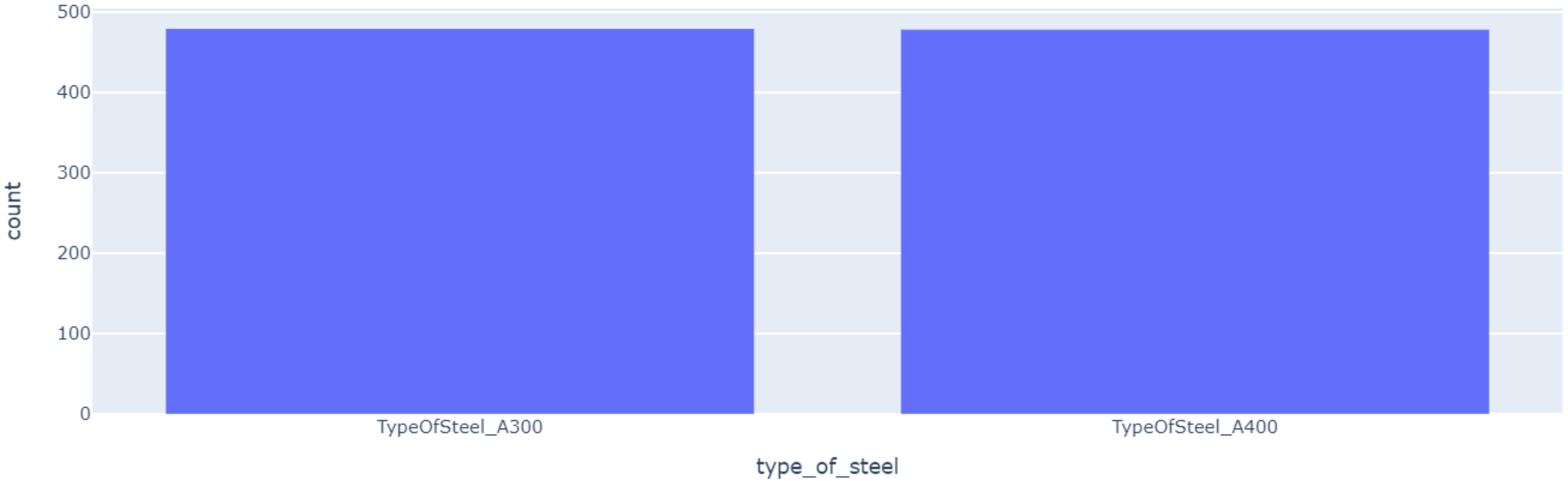
Análise Exploratória

- Qual a distribuição de aço entre as categorias existentes?



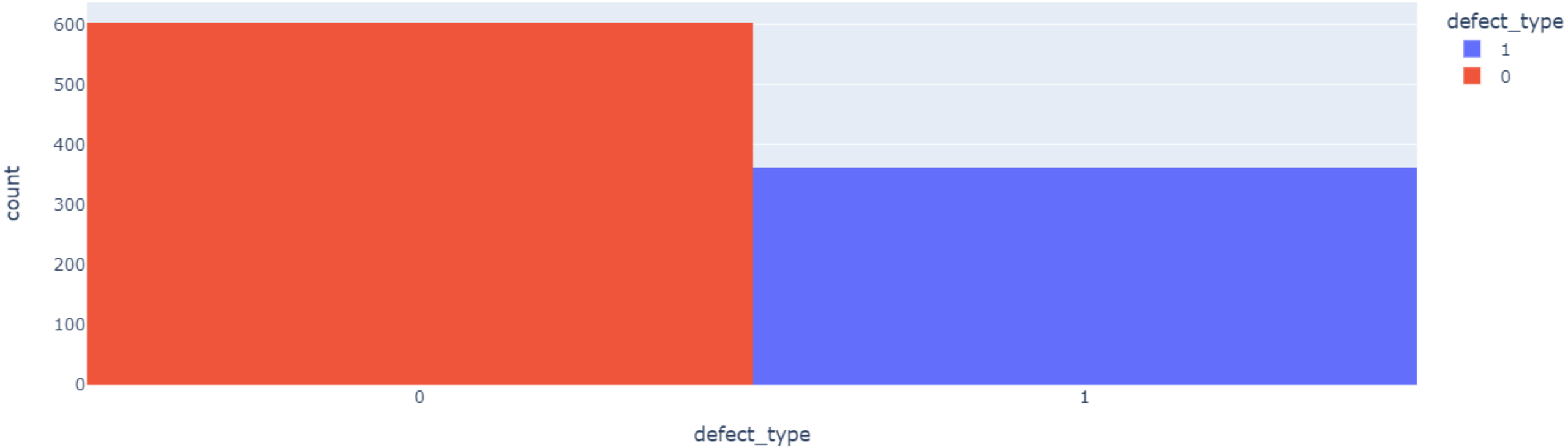
Análise Exploratória

- Qual a distribuição de aço entre as categorias existentes?



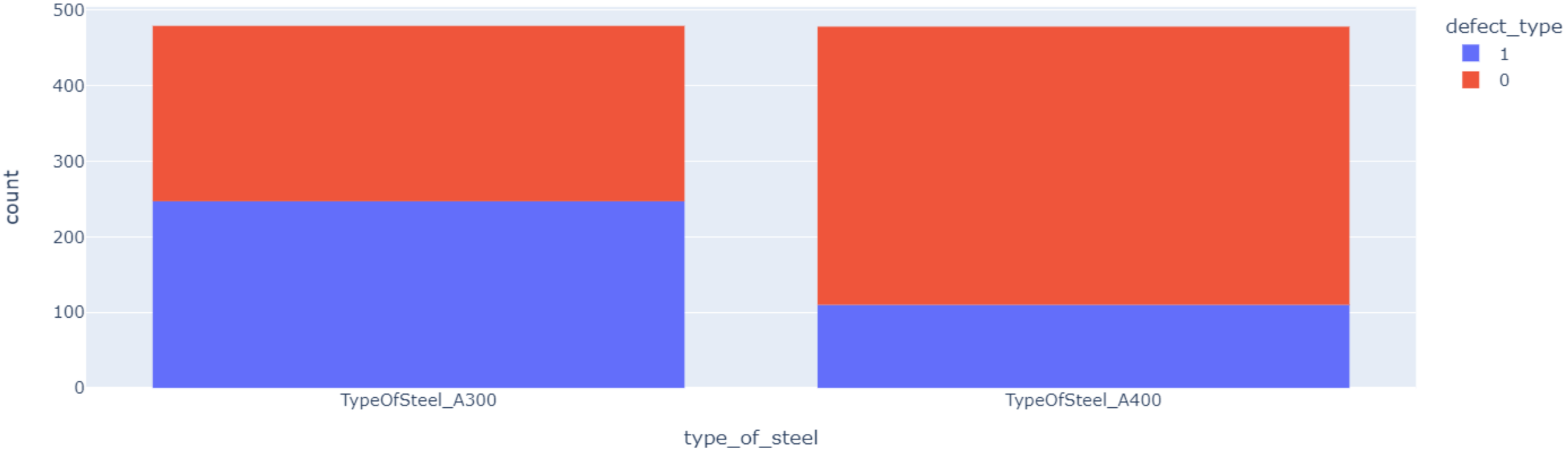
Análise Exploratória

- Como está a divisão dos dados entre os defeitos 0 e 1?



Análise Exploratória

- Como está a distribuição dos defeitos dentro das classes existentes?



Análise Exploratória

Outras perguntas feitas:

- Como está a distribuição dos defeitos nas outras variáveis?
- Como está a distribuição das classes nas outras variáveis?

Muitos gráficos, muitas observações.

Observações Diversas

- Não há muito padrão de comprimento x largura da placa, mas há padrão de espessura: 40, 50, 60, ...
- Por algum motivo, não existe luminosidade mínima dos pixels entre 77 e 92.

Modelo Escolhido

Extreme Gradient Boosting (XGBOOST):

- regularização embutida para evitar o *overfiting*
- algoritmo de árvore (não precisa normalizar) conhecido por seu desempenho superior
- lida bem com dados desbalanceados
- robusto a *outliers*

Separação em treino e teste usando validação cruzada onde N_folds = 5

Otimização por Hiperparâmetros

Otimização com GridSearchCV : obtenção de melhores métricas sem se preocupar muito com o *overfiting*.

Otimização manual:

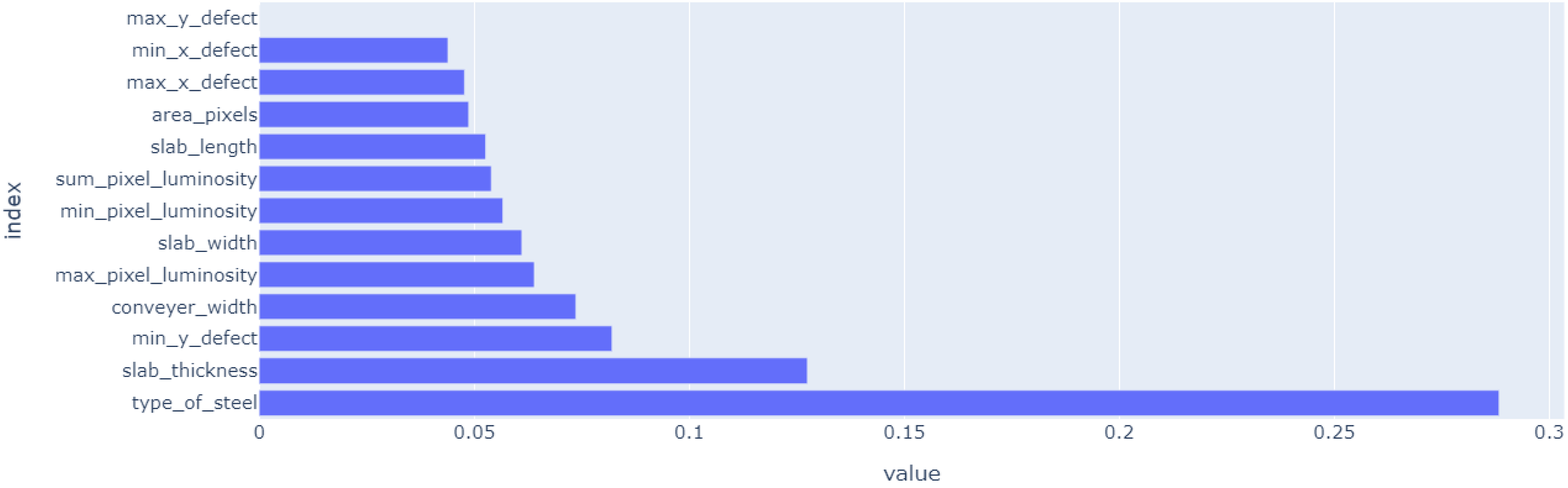
```
hyper_dict = {
   'objective':'binary:hinge',
   'learning_rate': 0.05
   'min_child_weight': 60,
   'max_depth': 5,
   'subsample': 0.85,
}
```

Métricas de Treino e de Teste (médias)

TREINO:
accuracy = 0.8079
recall = 0.8079
precision = 0.8068
f1 = 0.8072

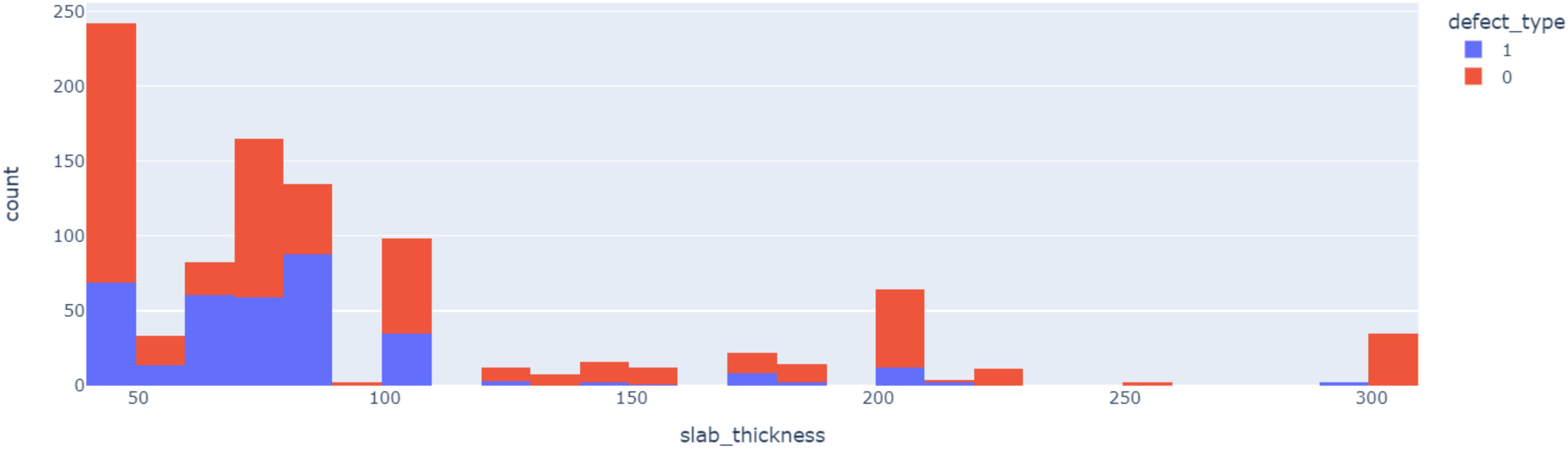
TESTE:
accuracy = 0.7512
recall = 0.7512
precision = 0.7519
f1 = 0.7503

Feature Importance

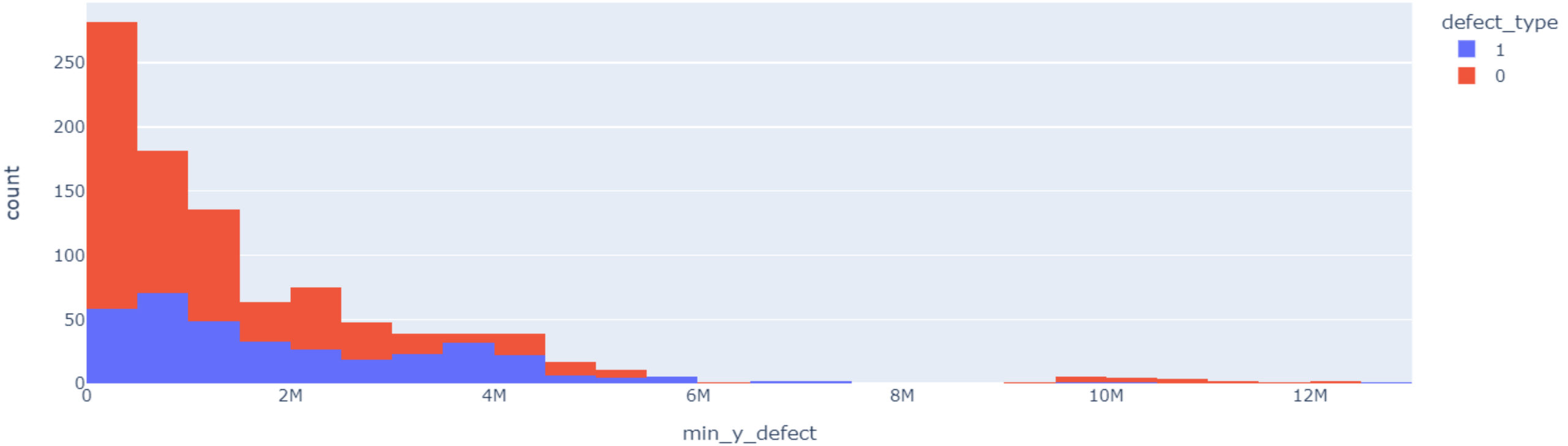


Lembrando que essa importância não necessariamente resume a realidade e sim o cenário dos dados fornecidos ao modelo.

Feature Importance



Feature Importance



Economia do Modelo

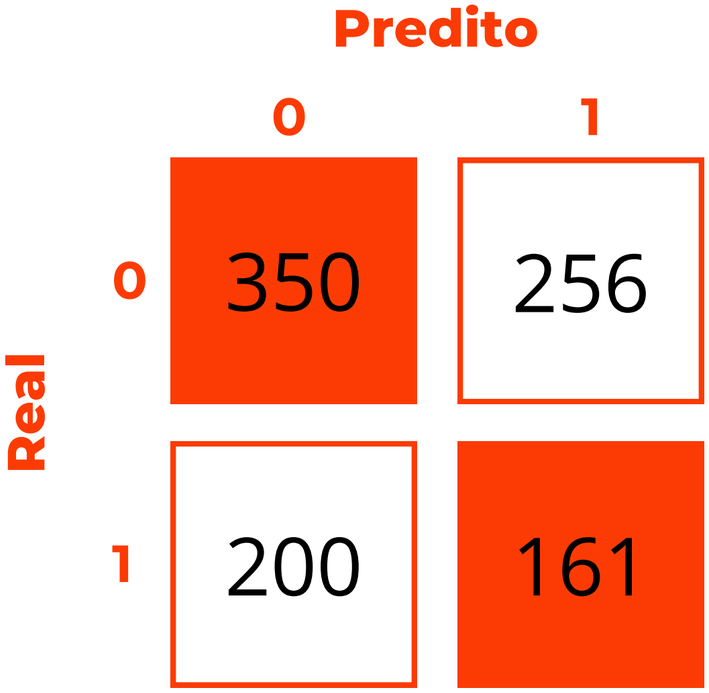
Custos:

- Verdadeiro zero e verdadeiro um: R$500,00
- Falso um: R$500,00 + R$3.500,00 = R$4.000,00
- Falso zero: R$500,00 + R$6.213,00 = R$ 6.713,00

Economia = Custo do especialista - Custo do modelo

Economia do Modelo

Matriz de confusão do especialista:



Para os custos fornecidos, o custo da matriz de confusão do especialista é de:
R$2.622.100,00

Economia do Modelo

Real \ Predito	0	1
0	350	256
1	200	161

Total = 967

×

Real \ Predito	0	1
0	484	116
1	122	235

Total = 957

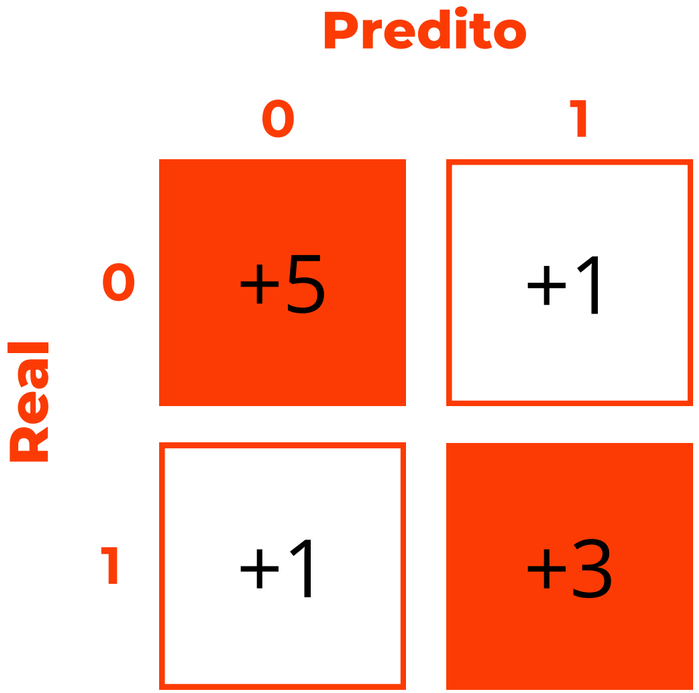
Necessidade de igualar o total para uma comparação mais justa

Economia do Modelo

Rebalanceando:

Custo do modelo:
R$1.642.486,00 + R$14.713,00
= R$1.657.199,00

Economia do modelo:
R$2.622.100,00 - R$1.657.199,00
= **R$964.901,00**



Distribuição respeitando
as métricas calculadas

Considerações Finais

- A economia obtida com o modelo é significativa.
- Algumas limitações presentes se deram a informações faltantes como, por exemplo, as unidades de medida de cada variável fornecida.
- Levamos em consideração a métrica *f1-score* como principal, entretanto, sabendo que um defeito 0 identificado como 1 gera um custo maior, talvez uma melhoria futura pudesse ser dar mais relevância à métrica *recall*.

Considerações Finais

- Outra melhoria futura seria analisar a correlação linear entre todas as variáveis, visando não só fazer modificações que ajudem o modelo, mas também tirar observações valiosas.
- Também é possível aprimorar o processo de busca por *outliers* multivariados.
- Por fim, também seria interessante obter mais dados do defeito tipo 1 para se ter um equilíbrio/balanceamento melhor da base de dados.

Obrigada!