

```
In [5]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [6]: df = pd.read_csv("Downloads/Tour prático de Machine Learning com Scikit-Learn/he
df.head()
```

```
Out[6]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2

EVALUATION

Se conseguimos atingir 95% de precisão em prever se um paciente tem ou não uma doença cardíaca durante o desenvolvimento.

✓ Atributos do Conjunto de Dados (13 variáveis + rótulo)

1. **age**: idade (em anos)
2. **sex**: sexo (1 = masculino; 0 = feminino)
3. **cp**: tipo de dor no peito
 - Valor 0: angina típica
 - Valor 1: angina atípica
 - Valor 2: dor não anginosa
 - Valor 3: assintomático
4. **trestbps**: pressão arterial em repouso (em mm Hg na admissão hospitalar)
5. **chol**: colesterol sérico (em mg/dl)
6. **fbs**: glicemia em jejum > 120 mg/dl (1 = verdadeiro; 0 = falso)
7. **restecg**: resultados do eletrocardiograma em repouso
 - Valor 0: normal
 - Valor 1: anormalidade da onda ST-T (inversões da onda T e/ou elevação ou depressão do segmento ST > 0,05 mV)
 - Valor 2: hipertrofia ventricular esquerda provável ou definitiva (critérios de Estes)
8. **thalach**: frequência cardíaca máxima atingida

9. **exang**: angina induzida por exercício (1 = sim; 0 = não)
10. **oldpeak**: depressão do segmento ST induzida por exercício em relação ao repouso
11. **slope**: inclinação do segmento ST no pico do exercício
- Valor 0: ascendente
 - Valor 1: plano
 - Valor 2: descendente
12. **ca**: número de vasos principais (0-3) visualizados por fluoroscopia
13. **thal**: condição da talassemia
- Valor 0: normal
 - Valor 1: defeito fixo
 - Valor 2: defeito reversível
-

Rótulo (variável-alvo):

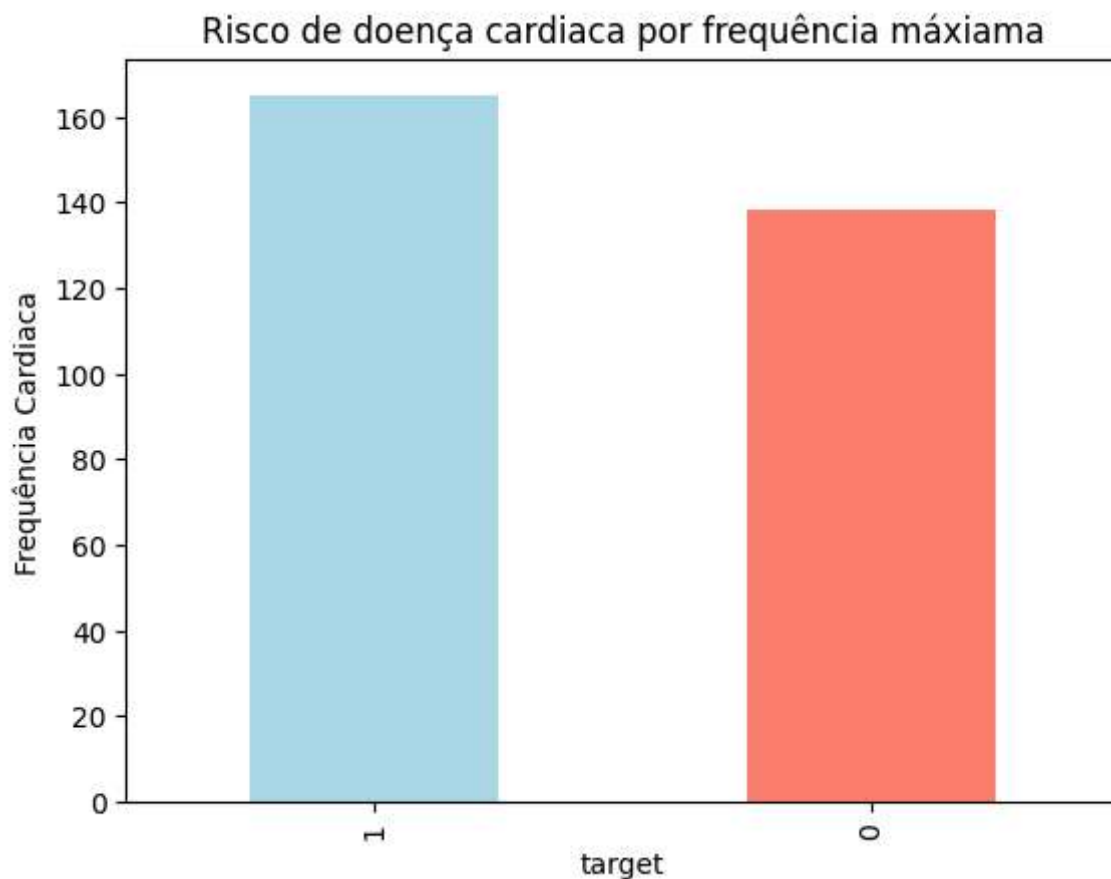
- **condition**: presença de doença cardíaca
 - 0 = não tem doença
 - 1 = tem doença
-

Se quiser que eu gere um dicionário, dataframe, gráfico, ou explicações mais detalhadas, é só avisar!

EDA (Análise exploratória)

```
In [14]: df["target"].value_counts().plot(kind="bar", color=["lightblue", "salmon"]);  
plt.title("Risco de doença cardíaca por frequência máxiama")  
plt.ylabel("Frequência Cardíaca")
```

```
Out[14]: Text(0, 0.5, 'Frequência Cardíaca')
```



```
In [12]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 303 entries, 0 to 302  
Data columns (total 14 columns):  
#   Column      Non-Null Count  Dtype  
---  ---  
0   age         303 non-null   int64  
1   sex         303 non-null   int64  
2   cp          303 non-null   int64  
3   trestbps    303 non-null   int64  
4   chol        303 non-null   int64  
5   fbs         303 non-null   int64  
6   restecg     303 non-null   int64  
7   thalach     303 non-null   int64  
8   exang       303 non-null   int64  
9   oldpeak     303 non-null   float64  
10  slope       303 non-null   int64  
11  ca          303 non-null   int64  
12  thal        303 non-null   int64  
13  target      303 non-null   int64  
dtypes: float64(1), int64(13)  
memory usage: 33.3 KB
```

```
In [9]: pd.crosstab(df["target"],df["sex"])
```

Out[9]: **sex** 0 1

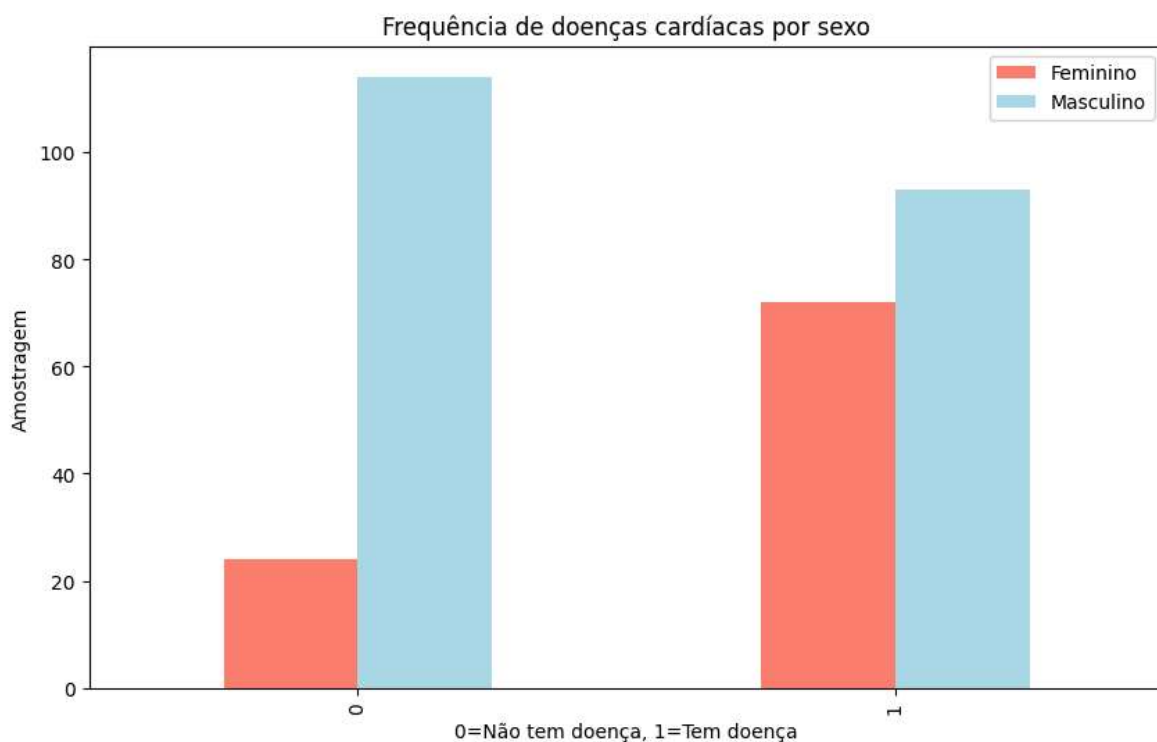
target

0 24 114

1 72 93

```
In [10]: pd.crosstab(df["target"], df["sex"]).plot(
    kind="bar",
    figsize=(10, 6),
    color=["salmon", "lightblue"]
)

plt.title("Frequência de doenças cardíacas por sexo")
plt.xlabel("0=Não tem doença, 1=Tem doença")
plt.ylabel("Amostragem")
plt.legend(["Feminino", "Masculino"])
plt.show()
```



In [16]: *#Valores positivos*

```
plt.figure(figsize=(10,6))
plt.scatter(df["age"][df["target"]==1],
            df["thalach"][df["target"]==1],
            c="salmon")
```

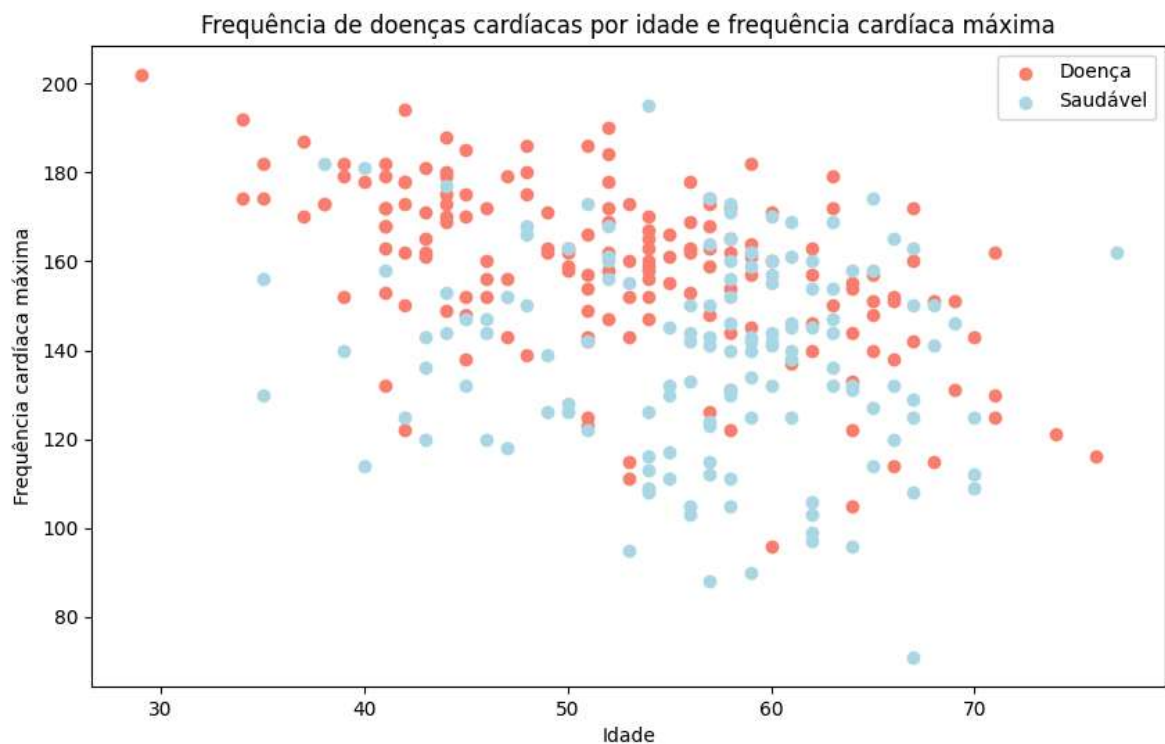
#Valores negativos

```
plt.scatter(df["age"][df["target"]==0],
            df["thalach"][df["target"]==0],
            c="lightblue")
```

```
plt.title("Frequência de doenças cardíacas por idade e frequência cardíaca máxima")
plt.xlabel("Idade")
```

```
plt.legend(["Doença", "Saudável"])
plt.ylabel("Frequência cardíaca máxima")
```

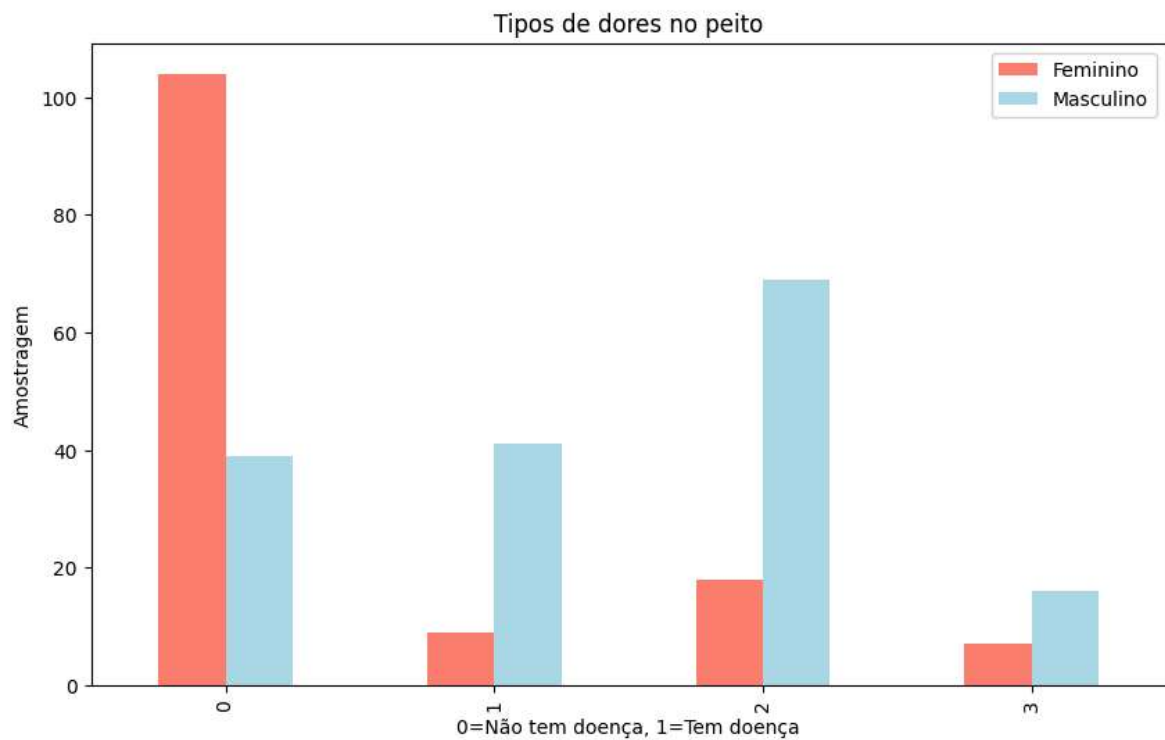
Out[16]: Text(0, 0.5, 'Frequência cardíaca máxima')



```
In [20]: pd.crosstab(df["cp"], df["target"]).plot(
    kind="bar",
    figsize=(10, 6),
    color=["salmon", "lightblue"]
)

plt.title("Tipos de dores no peito")
plt.xlabel("0=Não tem doença, 1=Tem doença")
plt.ylabel("Amostragem")
plt.legend(["Feminino", "Masculino"])
```

Out[20]: <matplotlib.legend.Legend at 0x2bfb9ad1090>



```
In [ ]: cp: tipo de dor no peito

Valor 0: angina típica
Valor 1: angina atípica
Valor 2: dor não anginosa
Valor 3: assintomático
```

MODELAGEM DOS DADOS

```
In [ ]:
```