



```
from google.colab import drive
drive.mount('/content/drive')
```



 Mounted at /content/drive

```
#abalone_df = pd.read_csv('caminho/para/seus/dados.csv') # Exemplo de carregamento de dados
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
dados_abalone = pd.read_csv("/content/drive/MyDrive/2024.2/APLICAÇÕES DE MACHINE LEARNING/Módulo II/abalone/abalone.data", header=None)
dados_abalone.head()
```



	0	1	2	3	4	5	6	7	8	
0	M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15	
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7	
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9	
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10	
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	7	

Passos seguintes:

[Gerar código com dados_abalone](#)



[Ver gráficos recomendados](#)

[New interactive sheet](#)

```
#Length: Comprimento
#Diam: Diâmetro
#Height: Altura
#Whole: Peso total (inteiro)
#Shucked: Peso da carne (sem casca)
#Viscera: Peso das vísceras
#Shell: Peso da casca
#Rings: Anéis (idade)
cabecalho_abalone = ['Sex', 'Length', 'Diam', 'Height', 'Whole', 'Shucked', 'Viscera', 'Shell', 'Rings']
```

```
# Criar um novo DataFrame com os nomes das colunas
abalone_df = pd.DataFrame(dados_abalone.values, columns=cabecalho_abalone)
abalone_df
```



	Sex	Length	Diam	Height	Whole	Shucked	Viscera	Shell	Rings	
0	M	0.455	0.365	0.095	0.514	0.2245	0.101	0.15	15	
1	M	0.35	0.265	0.09	0.2255	0.0995	0.0485	0.07	7	
2	F	0.53	0.42	0.135	0.677	0.2565	0.1415	0.21	9	
3	M	0.44	0.365	0.125	0.516	0.2155	0.114	0.155	10	
4	I	0.33	0.255	0.08	0.205	0.0895	0.0395	0.055	7	
...	
4172	F	0.565	0.45	0.165	0.887	0.37	0.239	0.249	11	
4173	M	0.59	0.44	0.135	0.966	0.439	0.2145	0.2605	10	
4174	M	0.6	0.475	0.205	1.176	0.5255	0.2875	0.308	9	
4175	F	0.625	0.485	0.15	1.0945	0.531	0.261	0.296	10	
4176	M	0.71	0.555	0.195	1.9485	0.9455	0.3765	0.495	12	

4177 rows x 9 columns

Passos seguintes:

[Gerar código com abalone_df](#)

[Ver gráficos recomendados](#)

[New interactive sheet](#)

> **Análise Exploratória**

[] ↳ 17 células ocultas

✓ Regressão Linear Multipla

```
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import OneHotEncoder
from sklearn.pipeline import make_pipeline

abalone_df.columns

Index(['Sex', 'Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weight',
       'Viscera weight', 'Shell weight', 'Rings'],
      dtype='object')

# Definir as features (variáveis independentes) e o target (variável dependente)
features = ['Length', 'Diameter', 'Height', 'Whole weight', 'Shucked weight',
            'Viscera weight', 'Shell weight', 'Rings']
target = 'Sex'

# Converter a coluna 'Sex' para valores numéricos
abalone_df['Sex'] = abalone_df['Sex'].map({'M': 0, 'F': 1, 'I': 2})

# Separar os dados em conjuntos de treino e teste
X_train, X_test, y_train, y_test = train_test_split(abalone_df[features], abalone_df[target], test_size=0.2, random_state=42)

# Criar e treinar um modelo de regressão linear
model = LinearRegression()
model.fit(X_train, y_train)
```

LinearRegression ⓘ ?

LinearRegression()

```
# Fazer previsões no conjunto de teste
y_pred = model.predict(X_test)

# Avaliar o modelo
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Erro Quadrático Médio (MSE): {mse}")
print(f"R-quadrado (R2): {r2}")

Erro Quadrático Médio (MSE): 0.49990506415752517
R-quadrado (R2): 0.2694795139934476
```

Discussão do Resultado e Limitação do Modelo

O modelo busca prever o sexo de um Abalone(molusco) com base em suas características físicas (features). Os resultados indicaram limitações importantes:

Baixo poder preditivo:

- R-quadrado (R2): O valor de R2 de 0.269 indica que o modelo explica apenas 26.9% da variância na variável 'Sex'. Ou seja, a maior parte da variação no sexo não é capturada pelas features utilizadas no modelo.
- Erro Quadrático Médio (MSE) alto: O MSE de 0.4999 representa o erro médio do modelo ao prever o sexo. Esse valor relativamente alto indica que as previsões do modelo estão distantes dos valores reais.