

Algebra Linear, Vetores e Geometria Analitica

A Regressão Linear é uma das técnicas mais fundamentais de aprendizado supervisionado. Seu objetivo é encontrar uma relação linear entre uma variável independente (X) e uma variável dependente (Y), ajustando uma reta que minimize o erro quadrático entre as previsões e os valores reais.

Após a implementação das bibliotecas e leituras dos dados pelo arquivo .csv, definimos os dados que serão utilizados como parâmetros x e y . Sendo x o preço total do pedido (totalAmount) e y a quantidade de itens no pedido (quantidade).

```
x = df['totalAmount'].values
y = df['quantidade'].values
```

Para realizar o cálculo do modelo na equação da reta $y = b_0 + b_1x$, adicionamos uma coluna de 1 à reta x , transformando ela (mX) em uma matriz de duas colunas.

```
mX = np.column_stack((np.ones(x.shape[0]), x))
```

Com a matriz feita, podemos realizar o método dos mínimos quadrados ($\beta = (mX^T * mX)^{-1} * mX^T * y$), com β sendo tanto b_0 quanto b_1 , mX sendo a matriz das variáveis independentes e y o vetor das variáveis dependentes

```
beta = np.linalg.inv(mX.T.dot(mX)).dot(mX.T).dot(y)

b0, b1 = beta
print(f"Intercepto (b0): {b0:.4f}")
print(f"Coeficiente (b1): {b1:.4f}")

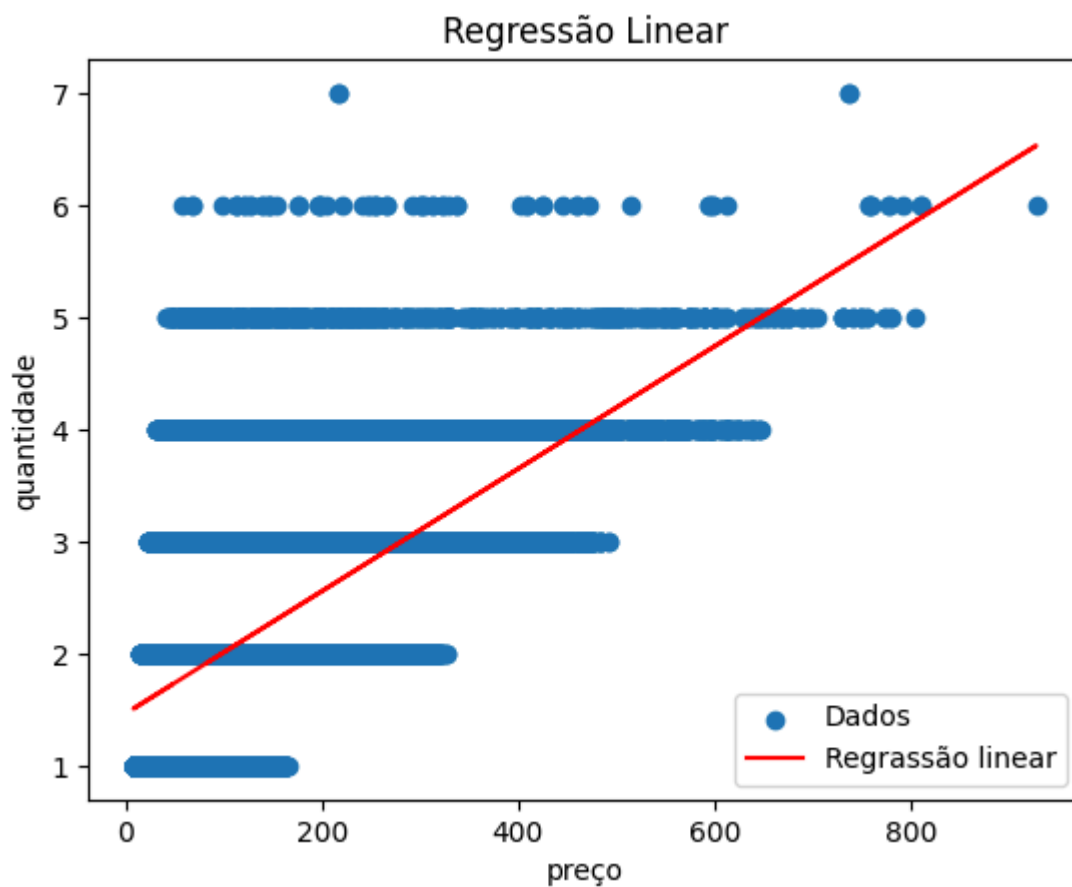
Intercepto (b0): 1.4673
Coeficiente (b1): 0.0055
```

Os valores calculados de b_0 (intercepto) e b_1 (inclinação) definem a equação da reta ajustada: $y^{\wedge} = b_0 + b_1x$

Com o cálculo da regressão linear feito, podemos realizar a geração dos gráficos.

```
y_pred = mX.dot(beta)

plt.scatter(x, y, label='Dados')
plt.plot(x, y_pred, color='red', label='Regração linear')
plt.xlabel('preço')
plt.ylabel('quantidade')
plt.legend()
plt.title('Regressão Linear')
plt.show()
```



A partir dos coeficientes estimados, é possível prever valores de quantidade para novos valores de totalAmount, dentro do intervalo de dados analisados.