

Estudos de regressão logística

A Regressão Logística é um método estatístico utilizado quando a variável dependente é categórica, ou seja, quando você está interessado em prever a probabilidade de um evento ocorrer ou não. É frequentemente usada em problemas de classificação, como prever se um e-mail é spam ou não, se um paciente tem uma doença específica ou não, entre outros. A Regressão Logística é uma extensão da regressão linear que modela a relação entre uma variável categórica binária (0 ou 1) e uma ou mais variáveis independentes.

	Linear	Logística
Reta	Reta	Curva - S
Variável Dependente	Continua	Categórica
Interpretação	\hat{y}	$\ln\left(\frac{p}{1+p}\right)$

Ao estudar a Regressão Logística, você aprenderá a ajustar o modelo, interpretar os coeficientes, calcular probabilidades e usar métricas de avaliação para medir o desempenho do modelo de classificação. Lembre-se de que em um cenário real, você teria um conjunto de dados muito maior e mais complexo, mas esta tabela de exemplo pode ser útil para entender os conceitos iniciais.

Aqui temos uma tabela com duas variáveis independentes (binária), “Horas de Estudo” (X1) e “Horas de Sono” (X2).

Horas de Estudo (X1)	Horas de Sono (X2)	Resultado
(Passou/Reprovou)		
3 horas	8 horas	Reprovou
6 horas	7 horas	Passou
4 horas	6 horas	Reprovou
7 horas	7 horas	Passou
5 horas	5 horas	Reprovou
8 horas	8 horas	Passou
7 horas	6 horas	Passou
9 horas	7 horas	Passou
2 horas	6 horas	Reprovou
5 horas	7 horas	Reprovou

Bora resolver?

```
import pandas as pd

data = {
    'Horas de Estudo': [3, 6, 4, 7, 5, 8, 7, 9, 2, 5],
    'Horas de Sono': [8, 7, 6, 7, 5, 8, 6, 7, 6, 7],
    'Resultado': ['Reprovou', 'Passou', 'Reprovou', 'Passou',
'Reprovou', 'Passou', 'Passou', 'Passou', 'Reprovou', 'Reprovou']
}

df = pd.DataFrame(data)
from sklearn.linear_model import LogisticRegression

# Separar as variáveis independentes (X) e a variável dependente (Y)
X = df[['Horas de Estudo', 'Horas de Sono']]
Y = df['Resultado']

# Criar e ajustar o modelo de Regressão Logística
model = LogisticRegression()
model.fit(X, Y)

# Fazer previsões com o modelo
previsoes = model.predict(X)

# Adicionar as previsões ao DataFrame
df['Previsão'] = previsoes

# Exibir o DataFrame com as previsões
print(df)
```

Além disso, você pode usar métricas de avaliação, como precisão, recall e F1-score, para avaliar a qualidade do modelo de Regressão Logística.

Sem variáveis numéricas

Compras Anteriores (X1) (Sim/Não)		Tempo Gasto no Site (X2)		Compra Online
Baixo		Baixo		Não
Alto		Alto		Sim
Médio		Médio		Sim
Baixo		Alto		Não
Alto		Baixo		Não
Médio		Alto		Sim
Baixo		Médio		Não

Como fica

```
pip install scikit-learn
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import LabelEncoder

# Criar o DataFrame com os dados
data = {
    'Compras Anteriores': ['Baixo', 'Alto', 'Médio', 'Baixo', 'Alto',
'Médio', 'Baixo'],
```

```

        'Tempo Gasto no Site': ['Baixo', 'Alto', 'Médio', 'Alto', 'Baixo',
        'Alto', 'Médio'],
        'Compra Online': ['Não', 'Sim', 'Sim', 'Não', 'Não', 'Sim', 'Não']
    }

df = pd.DataFrame(data)

# Converter variáveis categóricas em numéricas usando LabelEncoder
le = LabelEncoder()
df['Compras Anteriores'] = le.fit_transform(df['Compras Anteriores'])
df['Tempo Gasto no Site'] = le.fit_transform(df['Tempo Gasto no
Site'])

# Separar as variáveis independentes (X) e a variável dependente (Y)
X = df[['Compras Anteriores', 'Tempo Gasto no Site']]
Y = df['Compra Online']

# Criar e ajustar o modelo de Regressão Logística
model = LogisticRegression()
model.fit(X, Y)

# Fazer previsões com o modelo
previsoes = model.predict(X)

# Adicionar as previsões ao DataFrame
df['Previsão'] = previsoes

# Exibir o DataFrame com as previsões
print(df)

```