

netflix

November 15, 2023

Investimentos - Netflix

Brenno de Oliveira da Rosa - 2021029935 Eduardo Alves Carvalho - 2021017550 Lucas Luan Belarmino Barbosa - 2021017872

Neste bloco, trazemos as importações necessárias e importamos o histórico de valores da Netflix Chamamos os primeiros dados para verificar se a biblioteca havia sido importada:

```
[47]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import cross_val_score, train_test_split
from sklearn.linear_model import LogisticRegression

df=pd.read_csv('archive/Netflix_stock_history.csv')
df.head()
```

```
[47]:
```

	Date	Open	High	Low	Close	Volume	Dividends	\
0	2002-05-23	1.156429	1.242857	1.145714	1.196429	104790000	0	
1	2002-05-24	1.214286	1.225000	1.197143	1.210000	11104800	0	
2	2002-05-28	1.213571	1.232143	1.157143	1.157143	6609400	0	
3	2002-05-29	1.164286	1.164286	1.085714	1.103571	6757800	0	
4	2002-05-30	1.107857	1.107857	1.071429	1.071429	10154200	0	

```
Stock Splits
0      0.0
1      0.0
2      0.0
3      0.0
4      0.0
```

Algumas informações do dataset:

```
[44]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4870 entries, 0 to 4869
Data columns (total 8 columns):
 #   Column          Non-Null Count  Dtype
---  -
```

```
0   Date          4870 non-null   object
1   Open          4870 non-null   float64
2   High          4870 non-null   float64
3   Low           4870 non-null   float64
4   Close         4870 non-null   float64
5   Volume        4870 non-null   int64
6   Dividends     4870 non-null   int64
7   Stock Splits  4870 non-null   float64
dtypes: float64(5), int64(2), object(1)
memory usage: 304.5+ KB
```

Aqui verificamos se existe algum dado da biblioteca que é nulo, pois se houvesse, deveríamos remove-lo dos nossos testes:

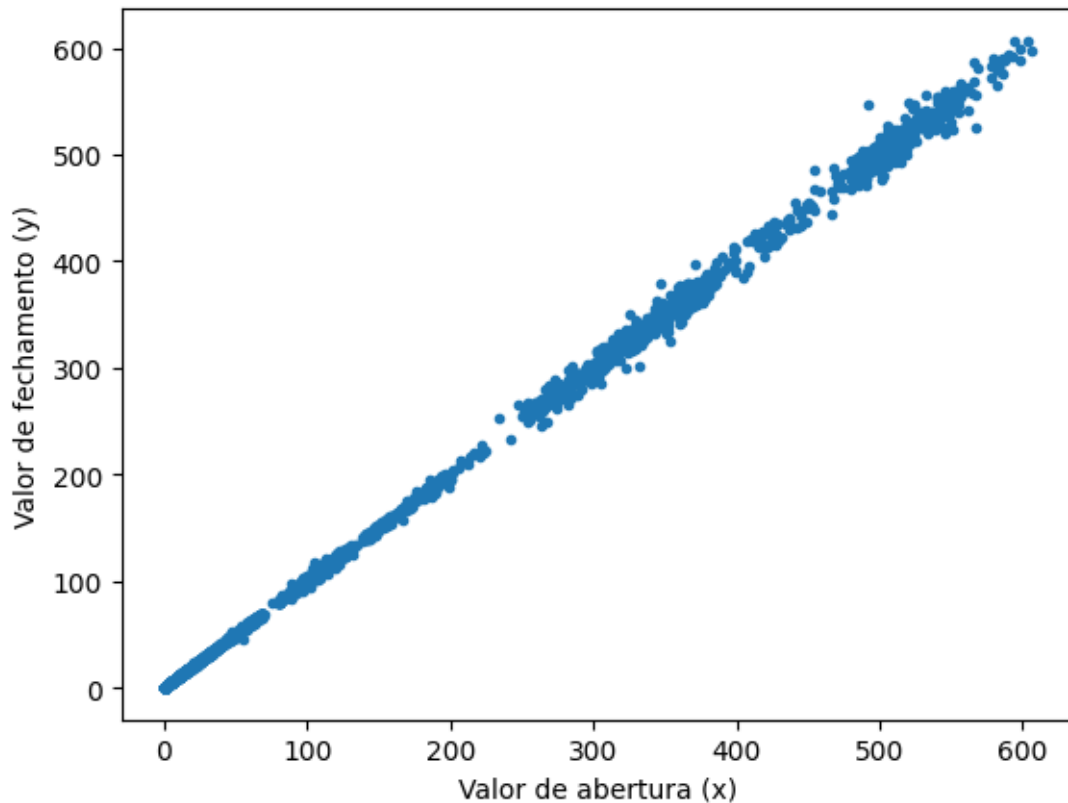
```
[16]: df.isnull().sum()
```

```
[16]: Date          0
      Open          0
      High          0
      Low           0
      Close         0
      Volume        0
      Dividends     0
      Stock Splits  0
      dtype: int64
```

Com todos os dados possuindo algum conteúdo, verificamos então qual a situação entre os valores de abertura do dia e do fechamento do dia na bolsa de valores:

```
[53]: plt.scatter(df['Open'], df['Close'], marker='.')
      plt.xlabel('Valor de abertura (x)')
      plt.ylabel('Valor de fechamento (y)')
```

```
[53]: Text(0, 0.5, 'Valor de fechamento (y)')
```



Com o gráfico, temos os valores previsto de abertura e fechamento, podemos ver que nem sempre o dia acaba com o mesmo valor de abertura, considerando que isto é algo comum na área de investimentos, podendo valorizar ou desvalorizar de um dia para o outro.

Com estes valores, gostaríamos de obter uma previsão do valor de fechamento das ações da Netflix dado o valor de sua abertura, para se ter uma noção da variação dos valores em um dia, podendo esse ser um fator crucial na realização de investimentos.

```
[55]: from sklearn.model_selection import train_test_split
x_val = df['Open'].values.reshape(-1,1)
y_val = df['Close']
x_train, x_test, y_train, y_test = train_test_split(x_val, y_val, test_size=0.
↳3, random_state=42)

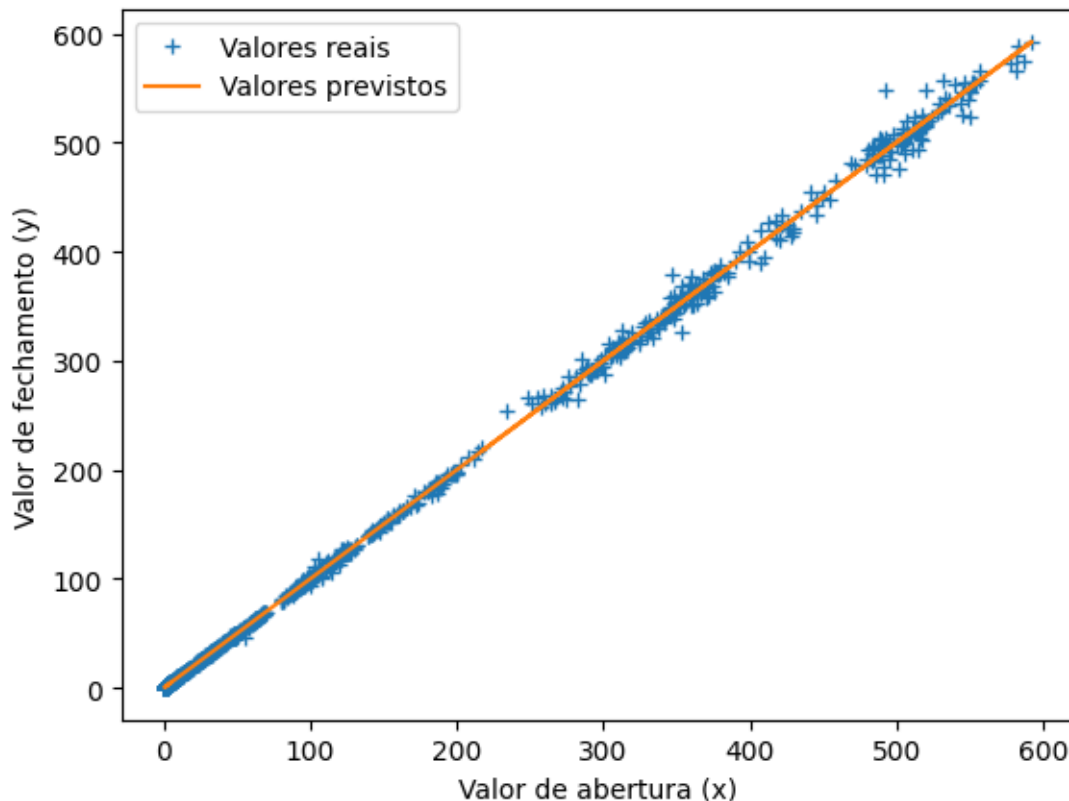
from sklearn.linear_model import LinearRegression
# regressão (treinamento)
my_model = LinearRegression()
my_model.fit(x_train, y_train)
print("R_squared =", my_model.score(x_test, y_test)) # r2 squared
#regressão (teste)
test_pred = my_model.predict(x_test)
```

```
R_squared = 0.9993163934178189
```

Com os cálculos, obtemos um valor de R^2 que indica que o valor de abertura incide 99.93% sobre o valor de fechamento.

Imprimindo um novo gráfico com o valor da previsão:

```
[56]: plt.plot(x_test, y_test, '+', label='Valores reais')
plt.plot(x_test, test_pred, '-', label='Valores previstos')
plt.xlabel('Valor de abertura (x)')
plt.ylabel('Valor de fechamento (y)')
plt.legend()
plt.show()
```



De acordo com o esperado, obtemos uma reta que se situa entre os pontos, acompanhando assim o seu crescimento linearmente. O crescimento indica que o valor das ações da Netflix está crescendo, o que indica que os valores de fechamento se mantêm próximos aos valores de abertura. Como visto abaixo, o valor informado para o coeficiente da função obtida é de 0.99979742, o que indica uma certa perda do valor de fechamento em relação ao valor de abertura.

```
[58]: print(my_model.intercept_, my_model.coef_, my_model.score(x_test, y_test))
```

```
0.056357304746072145 [0.99979742] 0.9993163934178189
```

Com isso a equação obtida é $y = 0.99979742x + 0.056357304746072145$