**Linear regression**

X = data[‘High’]

Y = data[‘Close’]

The score remains pretty much the same every time I run the code. However, the Mean Square Error fluctuates by big margins, from 0.8 to 12.9.

The last time I run the code, I got these values for score and mean square error:

**Score: 0.997195089335449**

**Mean Square Error: 1.3136745483580077**

**The Python Code**

import pandas as pd

import os

import matplotlib.pyplot as plt

import numpy as np

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import mean\_squared\_error

dirname = os.path.dirname(\_\_file\_\_)

stocks\_file = os.path.join(dirname, 'stocks.csv')

data = pd.read\_csv(stocks\_file)

*# Handle the NaN values*

data.dropna()

X = data['High']

X = X.fillna(X.mean()) *# Used fillna() because otherwise it would be inconsistent # of values between X and Y*

X = X.values.reshape(-1,1) *# reshaped because array was 1D instead of expected 2D*

y = data['Close']

y = y.fillna(y.mean())

y = y.values.reshape(-1,1)

plt.scatter(X, y)

plt.show()

*# Split*

*# Use scikit-learn’s train\_test\_split() method to split x into 80% training set and 20% testing set*

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, train\_size=0.8, test\_size=0.2)

plt.scatter(X\_train,y\_train,s=5)

plt.xlabel('X')

plt.ylabel('y')

plt.show()

lr\_model = LinearRegression()

lr\_model.fit(X\_train, y\_train)

score = lr\_model.score(X\_train, y\_train)

print("Score: ", score)

prediction = lr\_model.predict(X\_test)

plt.scatter(X\_test, y\_test, s=5)

plt.plot(X\_test, prediction, color='red')

plt.xlabel('X Test')

plt.ylabel('y Predictions')

plt.show()

sq\_err = mean\_squared\_error(y\_test, prediction)

print("Mean Square Error: ", sq\_err)

**Pyplot**

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generated