import matplotlib.pyplot as plt

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

from sklearn import datasets, linear\_model

from sklearn.metrics import mean\_squared\_error, r2\_score

from sklearn.linear\_model import LinearRegression

# Load the dataset

data = pd.read\_csv("C:/Users/x/Desktop/Desktop/ML/AAPL.csv")

# Dependent variable Y

close\_prices = data['Close(t)']

#Open + Volume = independent variables X

data2 = np.genfromtxt("C:/Users/x/Desktop/Desktop/ML/AAPL.csv", dtype='float', delimiter=',', skip\_header=1, usecols=(1,2,3,6))

X = np.array(data2)

print(X)

# Split the data into training/testing sets

X\_train = np.array(X[:450]).reshape((-1, 4)) #we take the first 450 elements

X\_test = np.array(X[450:]).reshape((-1, 4)) #we take elements after 450 index

# Split the targets into training/testing sets

Y\_train = close\_prices[:450] #we take the first 450 elements

Y\_test = close\_prices[450:] #we take elements after 450 index

regr = linear\_model.LinearRegression()

# Train the model using the training sets

regr.fit(X\_train, Y\_train)

# Make predictions using the testing set

Y\_pred = regr.predict(X\_test)

# The coefficients

print("Coefficients: \n", regr.coef\_)

# The mean squared error

print("Mean squared error: %.2f" % mean\_squared\_error(Y\_test, Y\_pred))

# The coefficient of determination: 1 is perfect prediction

print("Coefficient of determination: %.2f" % r2\_score(Y\_test, Y\_pred))

# x axis values

x = data['Close(t)']

# corresponding y axis values

y = data['Year']

# plotting the points

plt.plot(x, y)

# naming the x axis

plt.xlabel('Price')

# naming the y axis

plt.ylabel('Date')

# giving a title to my graph

plt.title('Closing price')

# function to show the plot

plt.show()

# x axis values

x = Y\_pred

# corresponding y axis values

y = data['Year']

# plotting the points

plt.plot(x, y)

# naming the x axis

plt.xlabel('pred.Price')

# naming the y axis

plt.ylabel('Date')

# giving a title to my graph

plt.title('Closing price pr')

# function to show the plot

plt.show()