**LINEAR REGRESSION**

import csv

import numpy as np

from sklearn.svm import SVR

import matplotlib.pyplot as plt

dates = []

prices = []

def get\_data(filename):

with open(filename, 'r') as csvfile:

csvFileReader = csv.reader(csvfile)

next(csvFileReader) # skipping column names

for row in csvFileReader:

dates.append(int(row[0].split('-')[0]))

prices.append(float(row[1]))

return

def predict\_price(dates, prices, x):

dates = np.reshape(dates,(len(dates), 1)) # converting to matrix of n X 1

svr\_rbf = SVR(kernel= 'rbf', C= 1e3, gamma= 0.1) # defining the support vector regression models

svr\_lin = SVR(kernel= 'linear', C= 1e3, gamma = 'auto')

svr\_poly = SVR(kernel= 'poly', C= 1e3, degree= 2, gamma = 'auto')

svr\_rbf.fit(dates, prices) # fitting the data points in the models

svr\_lin.fit(dates, prices)

svr\_poly.fit(dates, prices)

plt.scatter(dates, prices, color= 'black', label= 'Data') # plotting the initial datapoints

plt.plot(dates, svr\_rbf.predict(dates), color= 'red', label= 'RBF model') # plotting the line made by the RBF kernel

plt.plot(dates,svr\_lin.predict(dates), color= 'green', label= 'Linear model') # plotting the line made by linear kernel

plt.plot(dates,svr\_poly.predict(dates), color= 'blue', label= 'Polynomial model') # plotting the line made by polynomial kernel

plt.xlabel('Date')

plt.ylabel('Price')

plt.title('Support Vector Regression')

plt.legend()

plt.show()

return svr\_rbf.predict(np.array(x).reshape(-1,1))[0], svr\_lin.predict(np.array(x).reshape(-1,1))[0], svr\_poly.predict(np.array(x).reshape(-1,1))[0]

get\_data('goog.csv') # calling get\_data method by passing the csv file to it

print("Dates- ", dates)

print("Prices- ", prices)

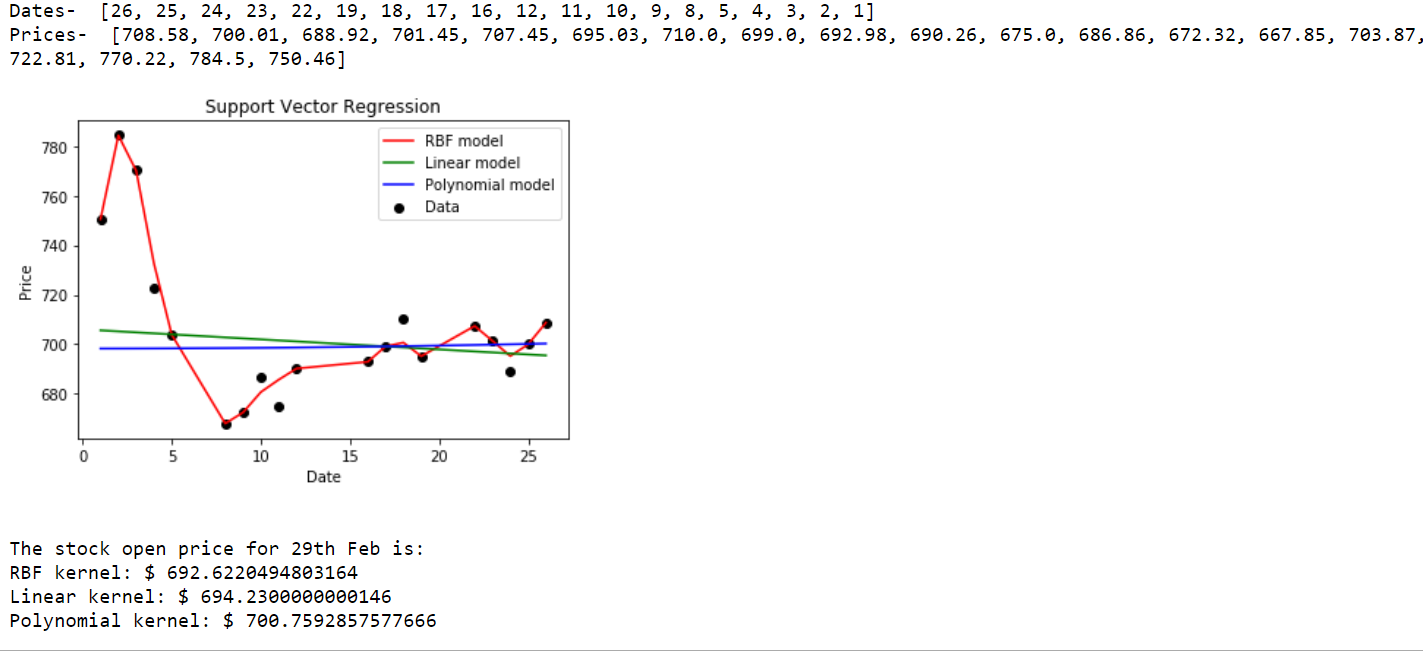
predicted\_price = predict\_price(dates, prices, 29)

print("\nThe stock open price for 29th Feb is:")

print("RBF kernel: $", str(predicted\_price[0]))

print("Linear kernel: $", str(predicted\_price[1]))

print("Polynomial kernel: $", str(predicted\_price[2]))

****

**Moving Average**

#import packages

import pandas as pd

import numpy as np

#to plot within notebook

import matplotlib.pyplot as plt

%matplotlib inline

#setting figure size

from matplotlib.pylab import rcParams

rcParams['figure.figsize'] = 20,10

#for normalizing data

from sklearn.preprocessing import MinMaxScaler

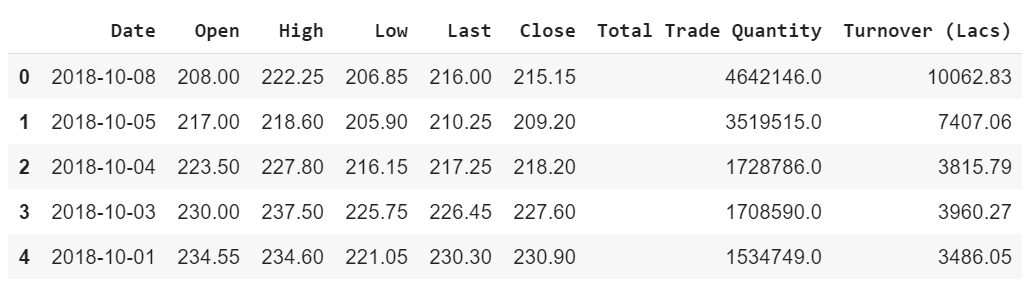
scaler = MinMaxScaler(feature\_range=(0, 1))

#read the file

df = pd.read\_csv('https://raw.githubusercontent.com/yishnu96/Python\_training/master/Project/NSE-TATAGLOBAL11.csv')

#print the head

df.head()



#setting index as date

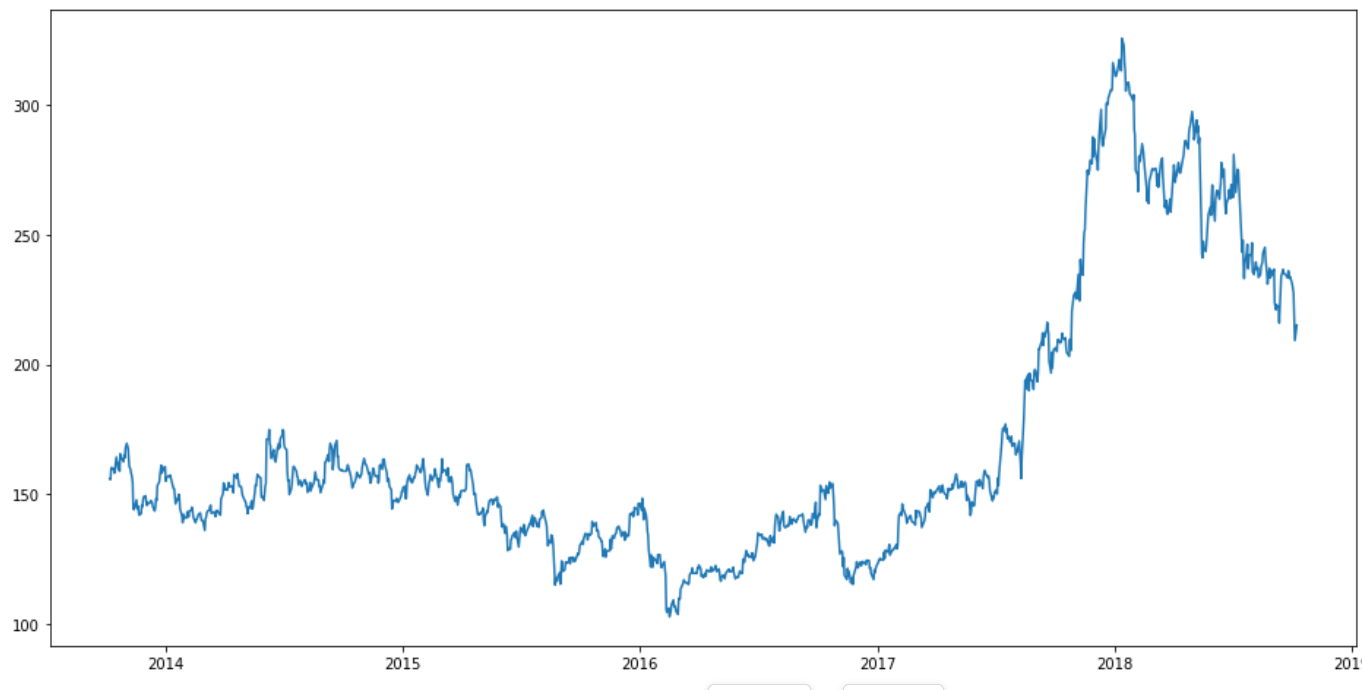
df['Date'] = pd.to\_datetime(df.Date,format='%Y-%m-%d')

df.index = df['Date']

#plot

plt.figure(figsize=(16,8))

plt.plot(df['Close'], label='Close Price history')



#creating dataframe with date and the target variable

data = df.sort\_index(ascending=True, axis=0)

new\_data = pd.DataFrame(index=range(0,len(df)),columns=['Date', 'Close'])

for i in range(0,len(data)):

new\_data['Date'][i] = data['Date'][i]

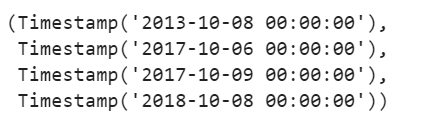
new\_data['Close'][i] = data['Close'][i]

#splitting into train and validation

train = new\_data[:987]

valid = new\_data[987:]

train['Date'].min(), train['Date'].max(), valid['Date'].min(), valid['Date'].max()



#make predictions

preds = []

for i in range(0,248):

a = train['Close'][len(train)-248+i:].sum() + sum(preds)

b = a/248

preds.append(b)

#calculate rmse

k = np.array(valid['Close'])

l=np.power((k-preds),2)

kl=np.mean(l)

rms=np.sqrt(kl)

#rms=np.sqrt(np.mean(np.power((np.array(valid['Close'])-preds),2)))

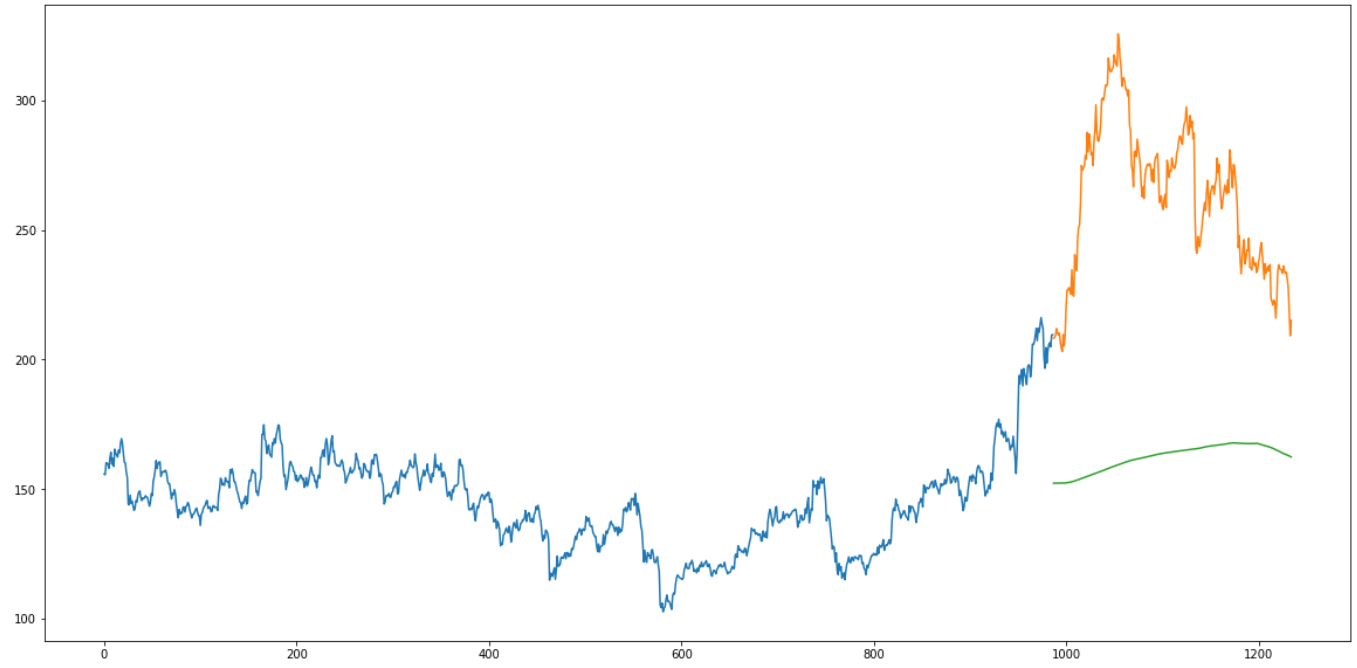
#plot

valid['Predictions'] = 0

valid['Predictions'] = preds

plt.plot(train['Close'])

plt.plot(valid[['Close', 'Predictions']])

****