#Import libraries

import os

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

!pip install matplotlib

%matplotlib inline

dataset\_train=pd.read\_csv(r"C:\Users\Dell\Desktop\Internships\WiDs\Google\_test\_data.csv")

dataset\_train

dataset\_train.head()

training\_set= dataset\_train.iloc[:,1:2].values

print(training\_set)

print(training\_set.shape)

from sklearn.preprocessing import MinMaxScaler

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

scaled\_training\_set = scaler.fit\_transform(training\_set)

scaled\_training\_set

X\_train = []

y\_train = []

for i in range(60,252):

X\_train.append(scaled\_training\_set[i-60:i,0])

y\_train.append(scaled\_training\_set[i,0])

X\_train = np.array(X\_train)

y\_train = np.array(y\_train)

print(X\_train.shape)

print(y\_train.shape)

X\_train = np.reshape(X\_train,(X\_train.shape[0], X\_train.shape[1], 1))

X\_train.shape

!pip install jupyterlab

!pip install keras

get\_ipython().system('pip install keras')

from keras.models import Sequential

from keras.layers import LSTM

from keras.layers import Dense

from keras.layers import Dropout

!pip install regressor

regressor = Sequential()

regressor.add(LSTM(units = 50, return\_sequences = True, input\_shape = (X\_train.shape[1],1) ))

regressor.add(Dropout(0.2))

regressor.add(LSTM(units = 50, return\_sequences = True))

regressor.add(Dropout(0.2))

regressor.add(LSTM(units = 50, return\_sequences = True))

regressor.add(Dropout(0.2))

regressor.add(LSTM(units = 50))

regressor.add(Dropout(0.2))

regressor.add(Dense(units=1))

!pip install regressor

regressor.compile(optimizer = 'adam', loss = 'mean\_squared\_error')

regressor.fit(X\_train, y\_train, epochs=100,batch\_size=32)

dataset\_test = pd.read\_csv(r"C:\Users\Dell\Desktop\Internships\WiDs\Google\_test\_data.csv")

actual\_stock\_price = dataset\_test.iloc[:,1:2].values

dataset\_total = pd.concat((dataset\_train['Open'],dataset\_test['Open']), axis=0)

inputs = dataset\_total[len(dataset\_total)-len(dataset\_test)-60:].values

inputs = inputs.reshape(-1,1)

inputs = scaler.transform(inputs)

X\_test = []

for i in range (60,80):

X\_test.append(inputs[i-60:i,0])

X\_test = np.array(X\_test)

X\_test = np.reshape(X\_test,(X\_test.shape[0],X\_test.shape[1],1))

predicted\_stock\_price = regressor.predict(X\_test)

predicted\_stock\_price = scaler.inverse\_transform(predicted\_stock\_price)

plt.plot(actual\_stock\_price, color = 'red', label= 'Actual Google Stock Price')

plt.plot(predicted\_stock\_price,color='blue', label='Predicted Google Stock Price')

plt.title('Google Stock Price Prediction')

plt.xlabel('Time')

plt.ylabel('Google Stock Price')

plt.legend()