import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, LSTM, Dropout

# load the dataset

df = pd.read\_csv('GOOG.csv', index\_col='Date', parse\_dates=['Date'])

df = df[['Close']]

df.head()

# visualize the dataset

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

plt.title('Google Stock Prices')

plt.plot(df['Close'])

plt.xlabel('Date', fontsize=18)

plt.ylabel('Close Price USD ($)', fontsize=18)

plt.show()

# preprocess the data

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

scaled\_data = scaler.fit\_transform(df['Close'].values.reshape(-1,1))

# split the data into training and testing sets

training\_data\_len = int(np.ceil(len(scaled\_data) \* 0.8))

train\_data = scaled\_data[0:training\_data\_len, :]

x\_train = []

y\_train = []

for i in range(60, len(train\_data)):

    x\_train.append(train\_data[i-60:i, 0])

    y\_train.append(train\_data[i, 0])

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

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

# define the RNN architecture

model = Sequential()

model.add(LSTM(50, return\_sequences=True, input\_shape=(x\_train.shape[1], 1)))

model.add(Dropout(0.2))

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

model.add(Dropout(0.2))

model.add(LSTM(50))

model.add(Dropout(0.2))

model.add(Dense(1))

# compile the model

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

# train the model

model.fit(x\_train, y\_train, epochs=25, batch\_size=32)

# visualize the predictions

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

plt.title('Google Stock Prices - Predicted vs Actual')

plt.plot(df.iloc[training\_data\_len:, :].index, y\_test, label='Actual')

plt.plot(df.iloc[training\_data\_len:, :].index, predictions, label='Predicted')

plt.xlabel('Date', fontsize=18)

plt.ylabel('Close Price USD ($)', fontsize=18)

plt.legend()

plt.show()