import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

from matplotlib.pylab import rcParams

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

from sklearn.preprocessing import MinMaxScaler

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

df=pd.read\_csv("NSE-TATA.csv")

df.head()

df.index=df['Date']

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

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

from keras.models import Sequential

from keras.layers import LSTM,Dropout,Dense

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

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

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

new\_dataset["Date"][i]=data['Date'][i]

new\_dataset["Close"][i]=data["Close"][i]

new\_dataset.index=new\_dataset.Date

new\_dataset.drop("Date",axis=1,inplace=True)

final\_dataset=new\_dataset.values

train\_data=final\_dataset[0:987,:]

valid\_data=final\_dataset[987:,:]

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

scaled\_data=scaler.fit\_transform(final\_dataset)

x\_train\_data,y\_train\_data=[],[]

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

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

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

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

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

lstm\_model=Sequential()

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

lstm\_model.add(LSTM(units=50))

lstm\_model.add(Dense(1))

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

lstm\_model.fit(x\_train\_data,y\_train\_data,epochs=1,batch\_size=1,verbose=2)

inputs\_data=new\_dataset[len(new\_dataset)-len(valid\_data)-60:].values

inputs\_data=inputs\_data.reshape(-1,1)

inputs\_data=scaler.transform(inputs\_data)

X\_test=[]

for i in range(60,inputs\_data.shape[0]):

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

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

lstm\_model.save("saved\_model.h5")

train\_data=new\_dataset[:987]

valid\_data=new\_dataset[987:]

valid\_data['Predictions']=predicted\_closing\_price

plt.plot(train\_data["Close"])

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