**Chapter 5**

**IMPLEMENTATION AND TESTING**

**// Import the libraries**

import math

import numpy as np

import pandas as pd

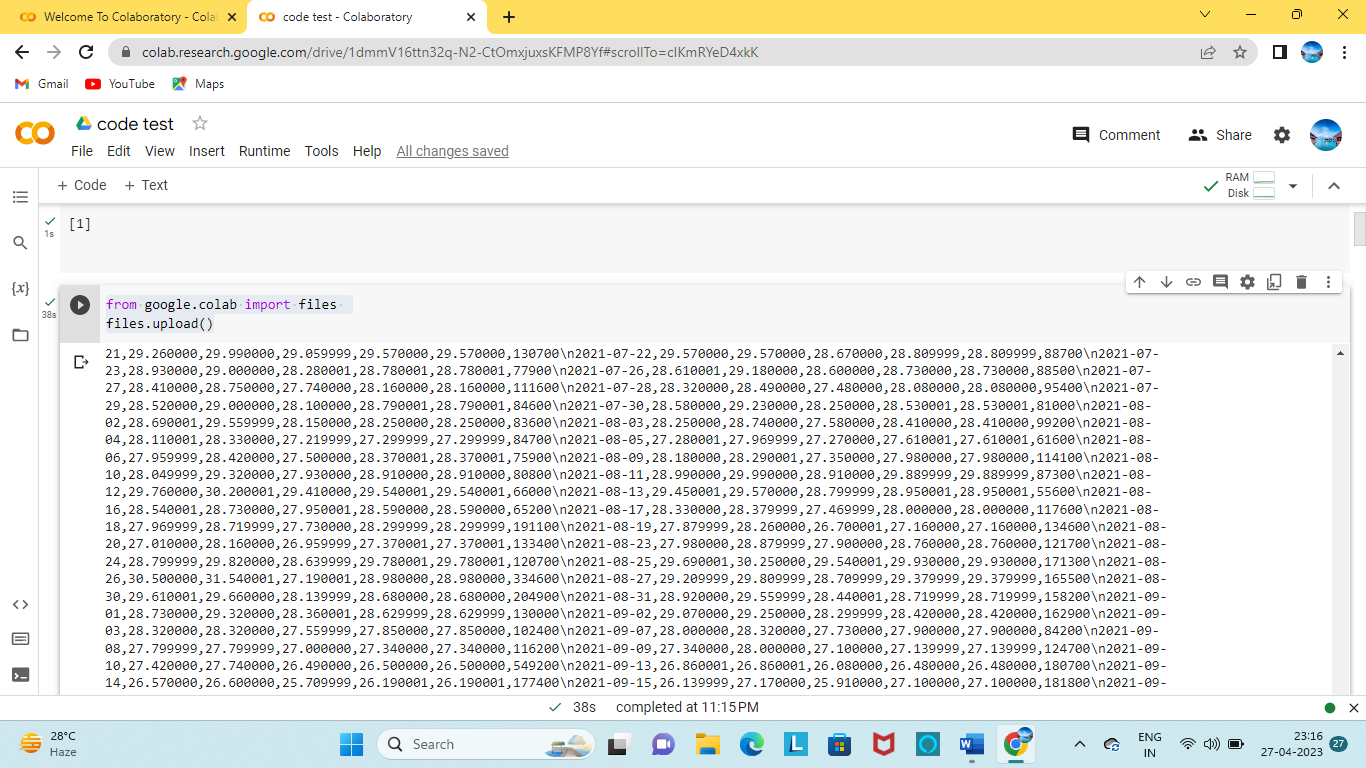
from sklearn.preprocessing import MinMaxScaler

import matplotlib.pyplot as plt

**// Get the stock data**

from google.colab import files

files.upload()



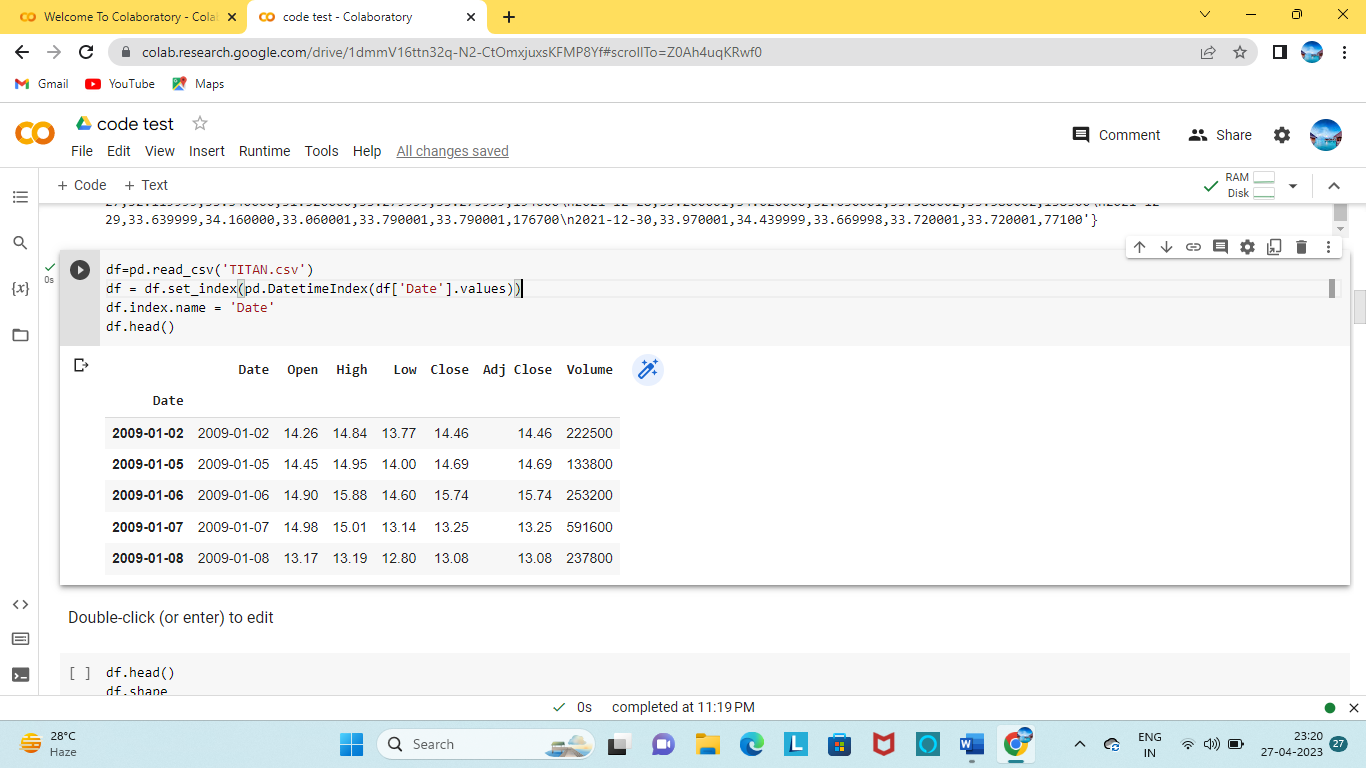
**// Show the data**

df=pd.read\_csv('TITAN.csv')

df = df.set\_index(pd.DatetimeIndex(df['Date'].values))

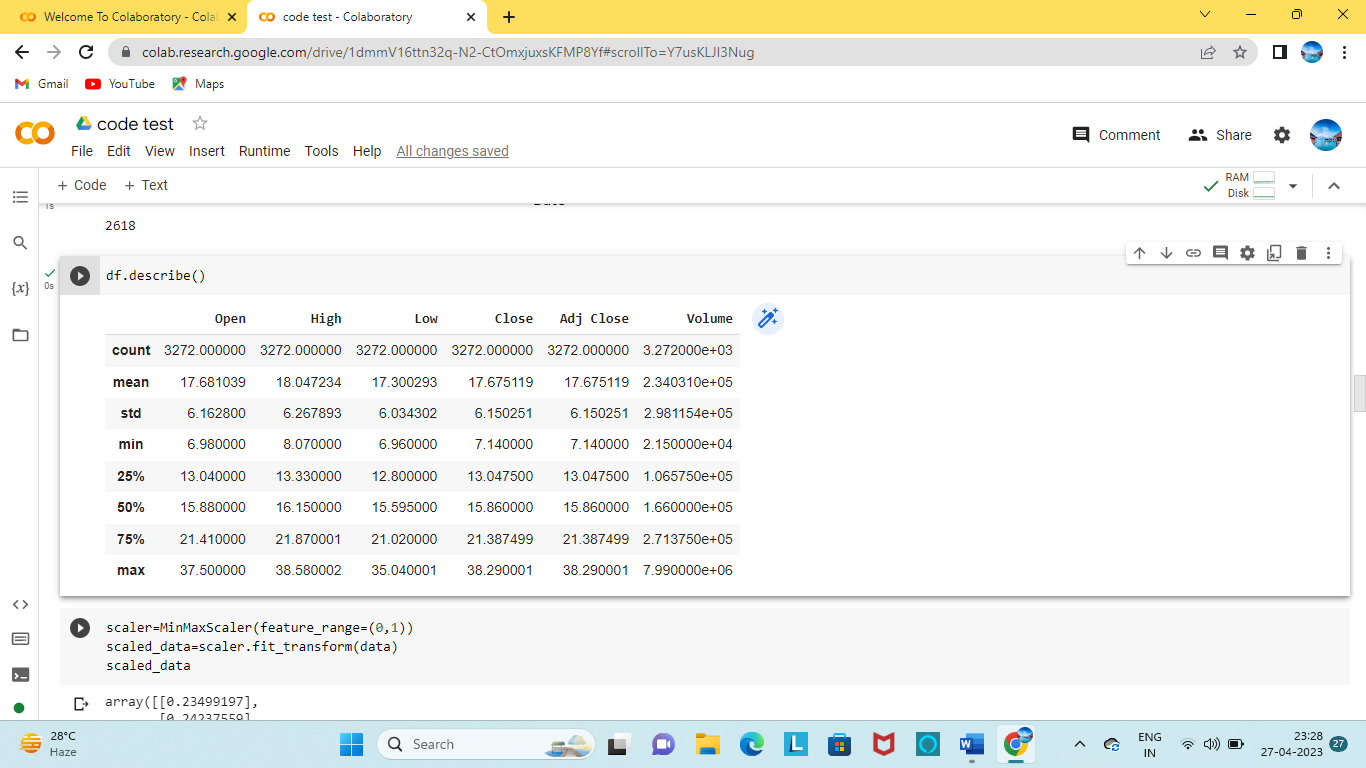
df.index.name = 'Date'

df.head()

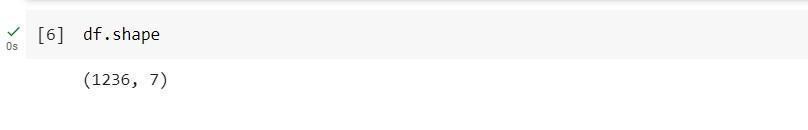


**// Describe the data**

df.describe( )



**// Get the number of rows and columns in data set**



**// Visualize the closing price history**

df.head()

df.shape

plt.figure(figsize=(10,5))

plt.title('Close price history')

plt.plot(df['Close'])

plt.xlabel('Date')

plt.ylabel('Close price RS ₹ ')

plt.show()

data=df.filter(['Close'])

dataset=data.values

training\_data\_len=math.ceil(len(dataset) \* .8)

training\_data\_len

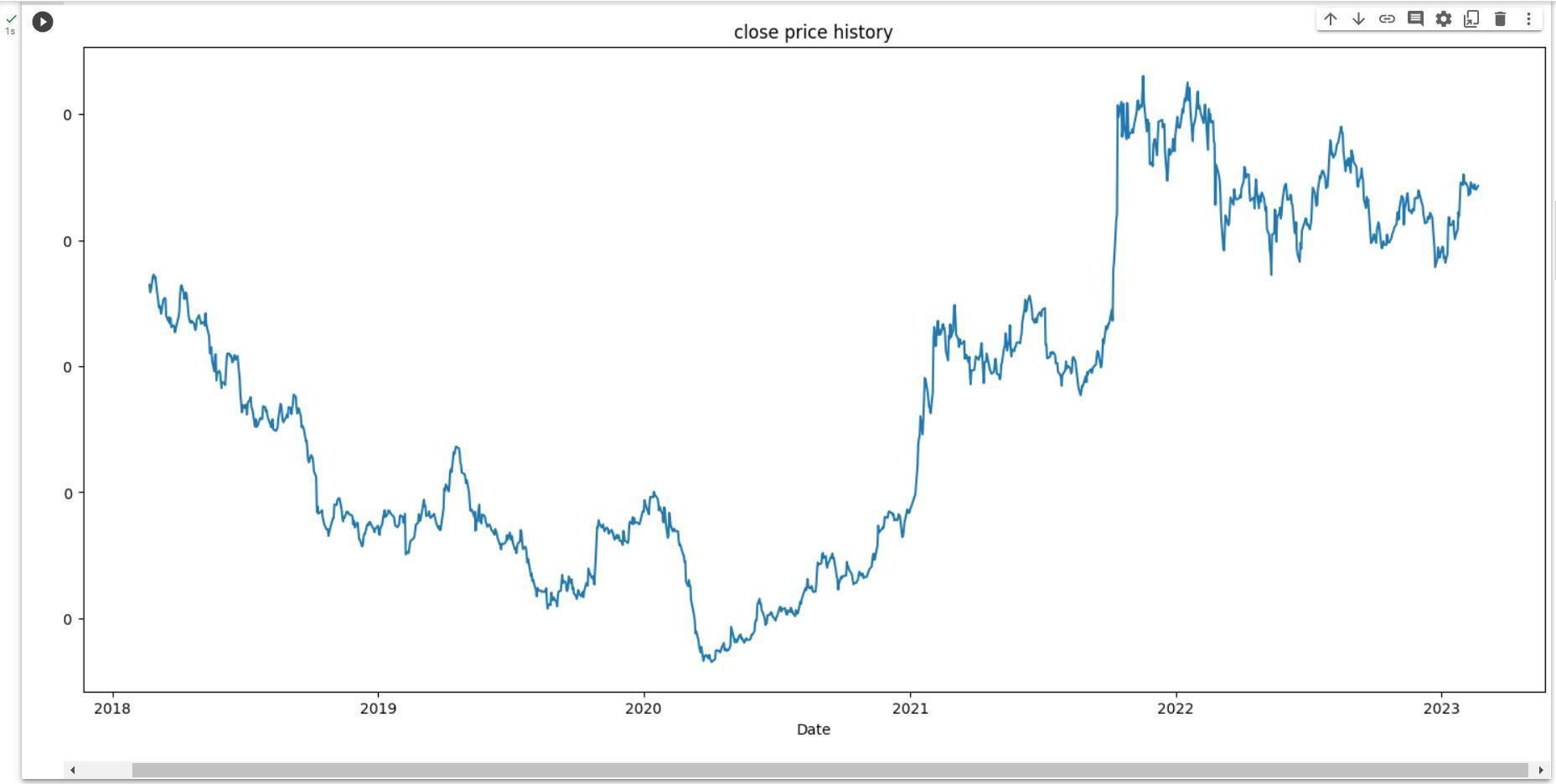


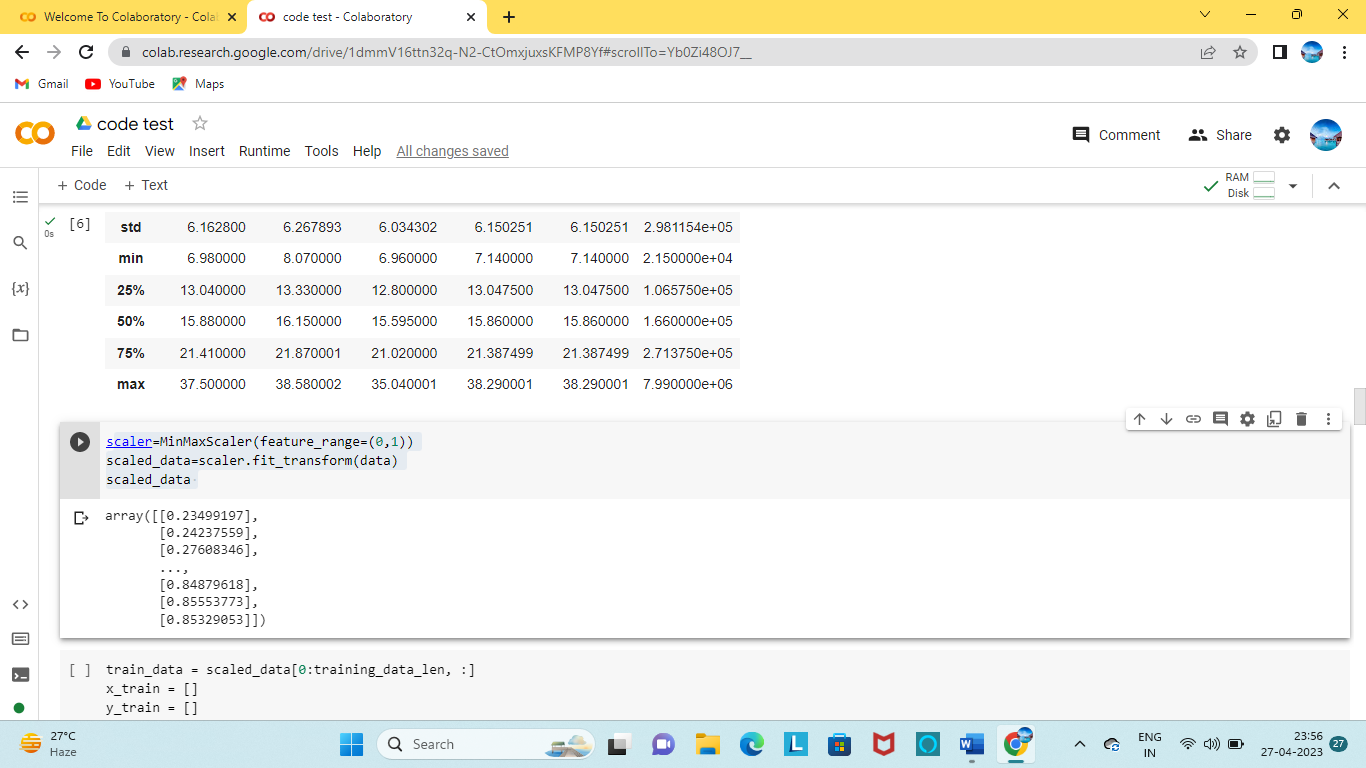
Fig 5.1 Closing Price Graph

**// Scale the data**

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

scaled\_data=scaler.fit\_transform(data)

scaled\_data



**// Create the training dataset**

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])

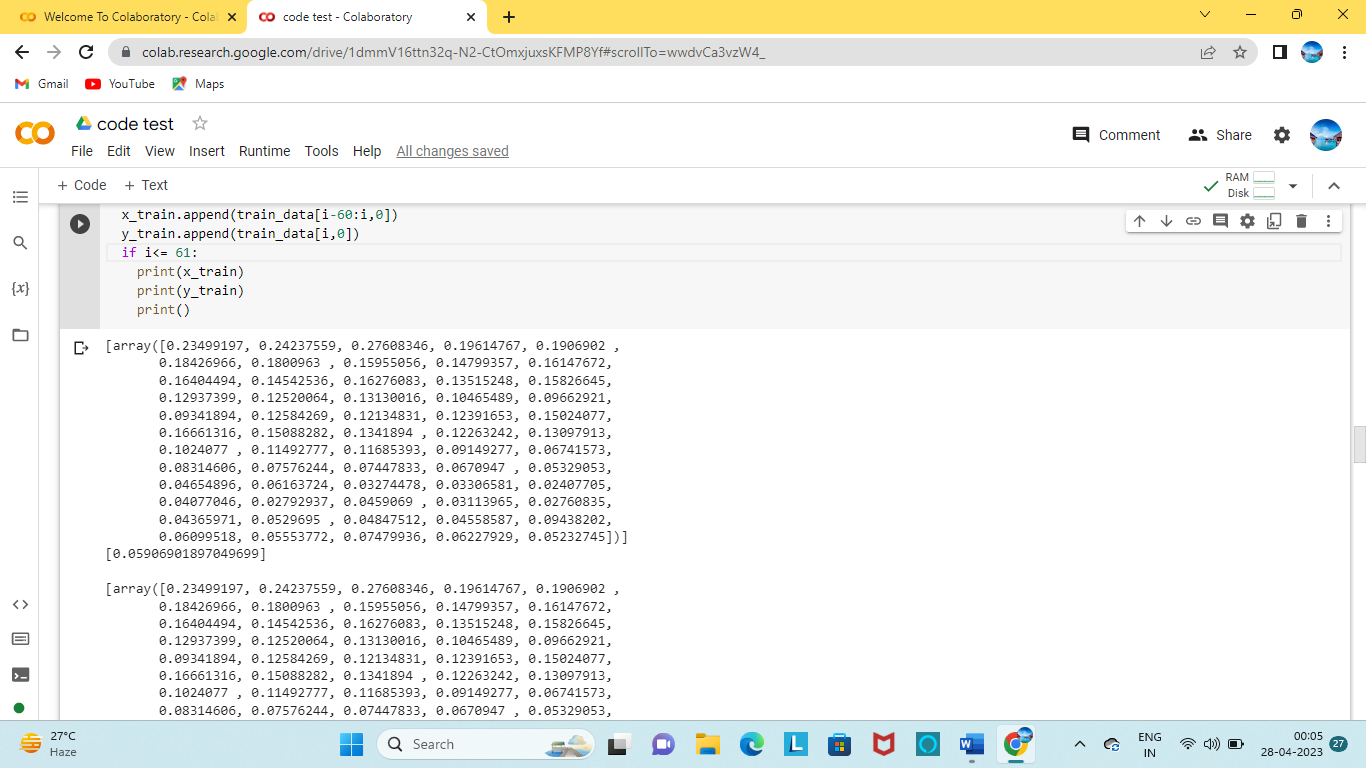
y\_train.append(train\_data[i,0])

if i<= 61:

print(x\_train)

print(y\_train)

print()



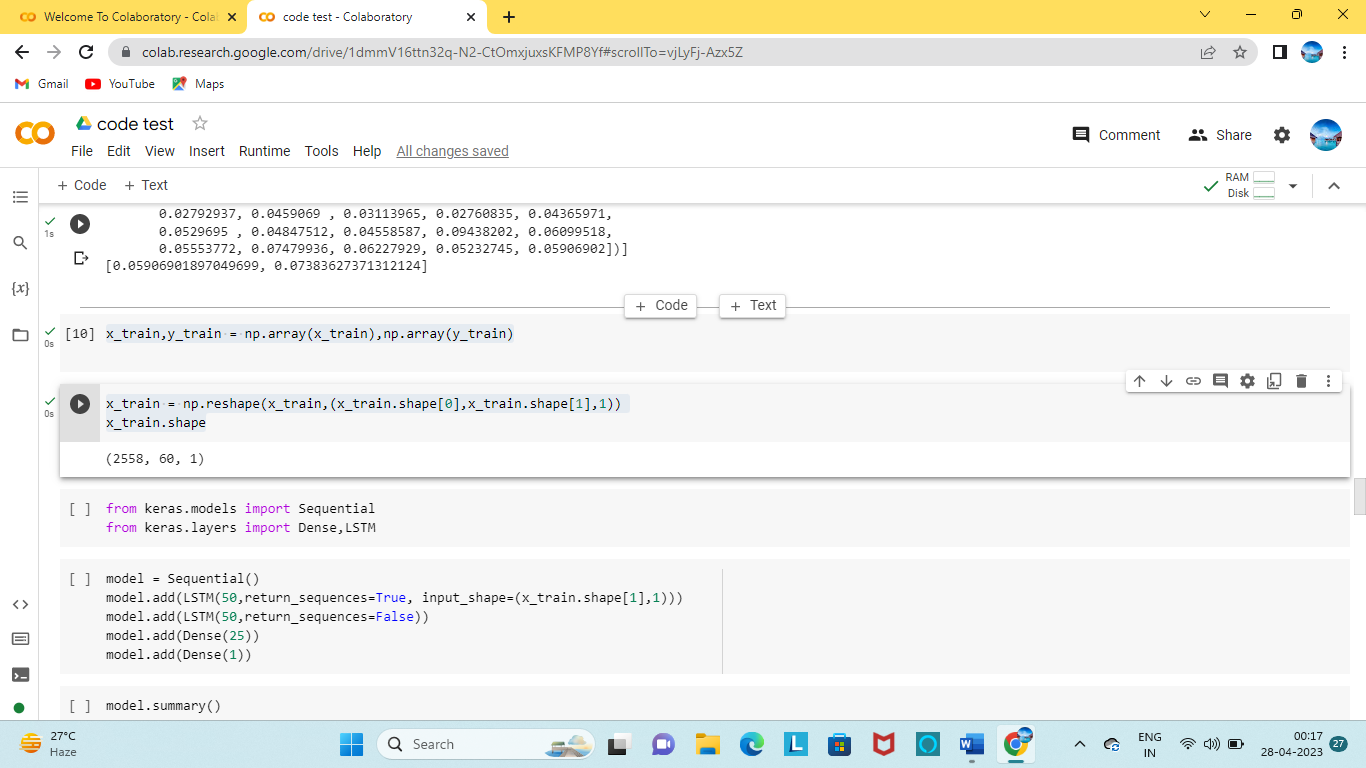
**// Convert the x\_train and y\_train to numpy arrays**

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

**// Reshape the data**

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

x\_train.shape



**// Import the libraries required for LSTM model**

from keras.models import Sequential

from keras.layers import Dense,LSTM

**// Build the LSTM model**

model = Sequential()

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

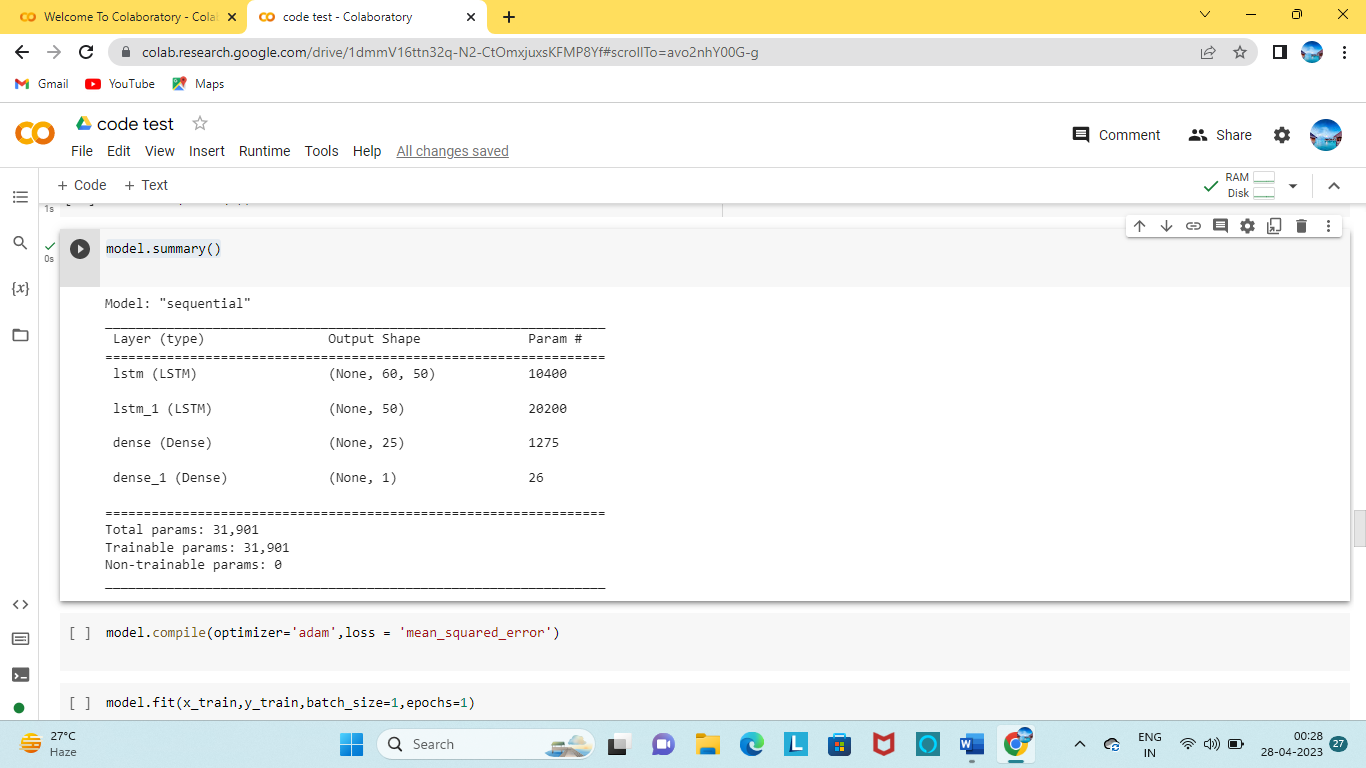
model.add(LSTM(50,return\_sequences=False))

model.add(Dense(25))

model.add(Dense(1))

**// View the summary of LSTM model**

model.summary()

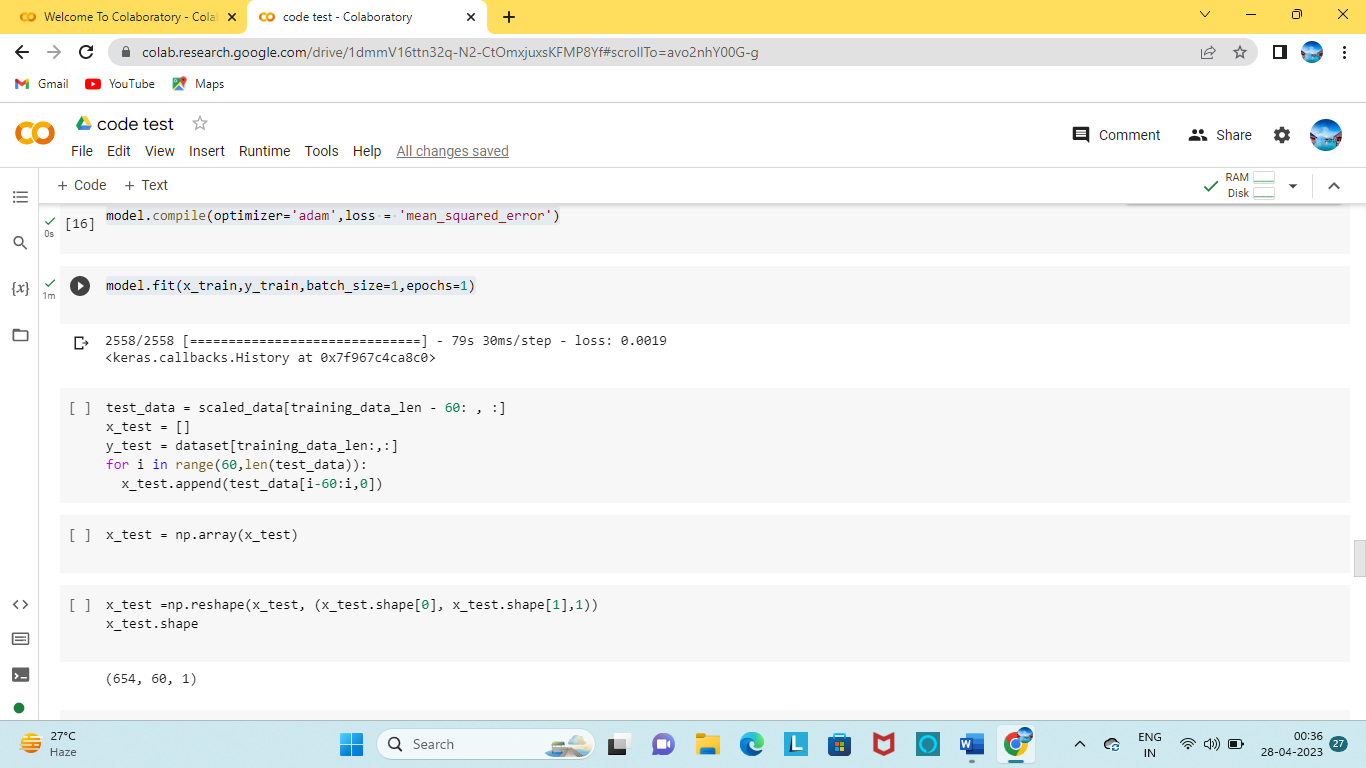


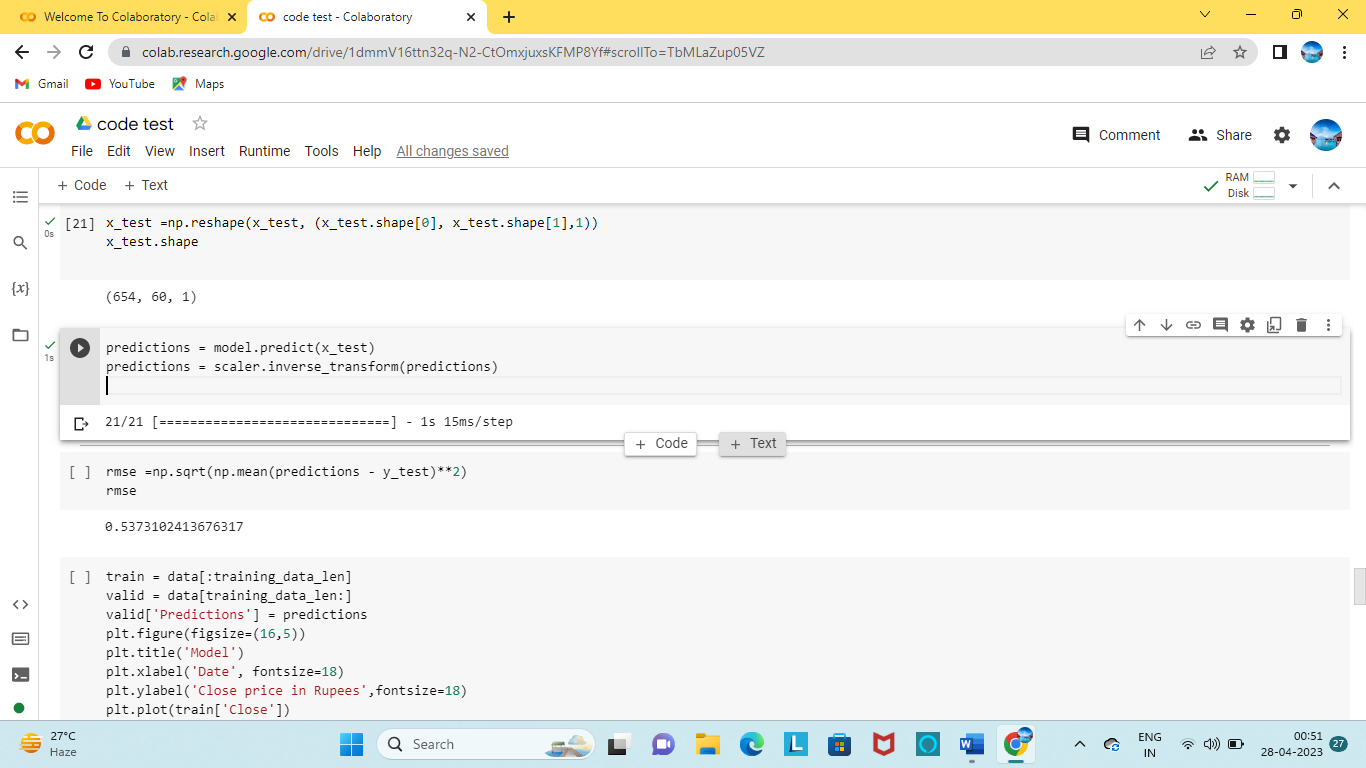
**// Compile the model**

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

**// Train the model**

model.fit(x\_train,y\_train,batch\_size=1,epochs=1)



**// Create the testing data set**

test\_data = scaled\_data[training\_data\_len - 60: , :]

x\_test = []

y\_test = dataset[training\_data\_len:,:]

for i in range(60,len(test\_data)):

x\_test.append(test\_data[i-60:i,0])

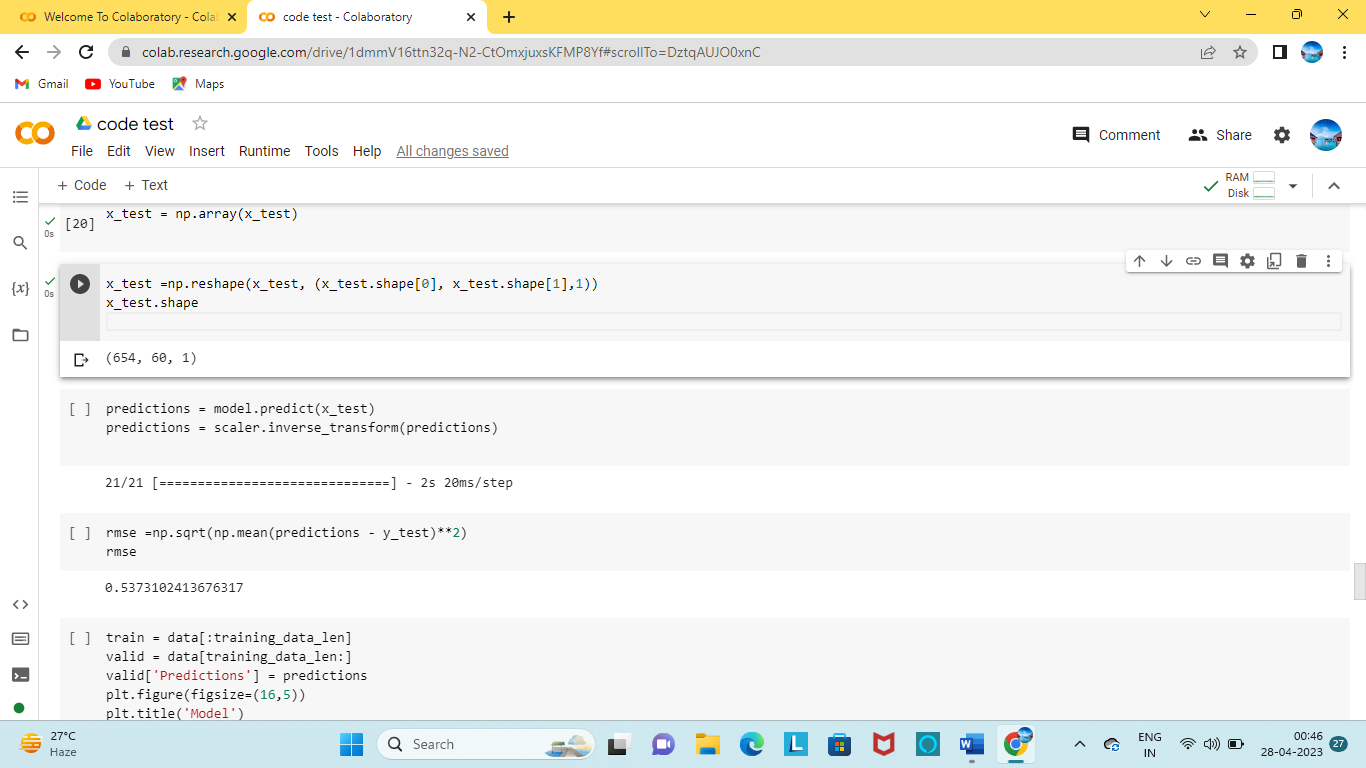
**// Convert the data to a numpy** **array**

x\_test = np.array(x\_test)

**// Reshape the data**

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

x\_test.shape



**// Get the model predicted price values**

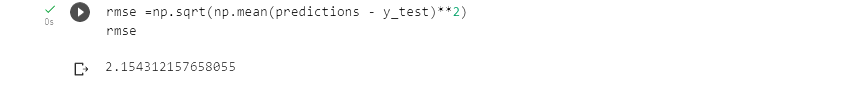
predictions = model.predict(x\_test)

predictions = scaler.inverse\_transform(predictions)

**// Get the root mean squared error (RMSE)**

rmse =np.sqrt(np.mean(predictions - y\_test)\*\*2)

rmse



**// Plot the data**

train = data[:training\_data\_len]

valid = data[training\_data\_len:]valid['Predictions'] = predictions

plt.figure(figsize=(16,5))

plt.title('Model')

plt.xlabel('Date', fontsize=18)

plt.ylabel('Close price in Rupees',fontsize=18)

plt.plot(train['Close'])

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

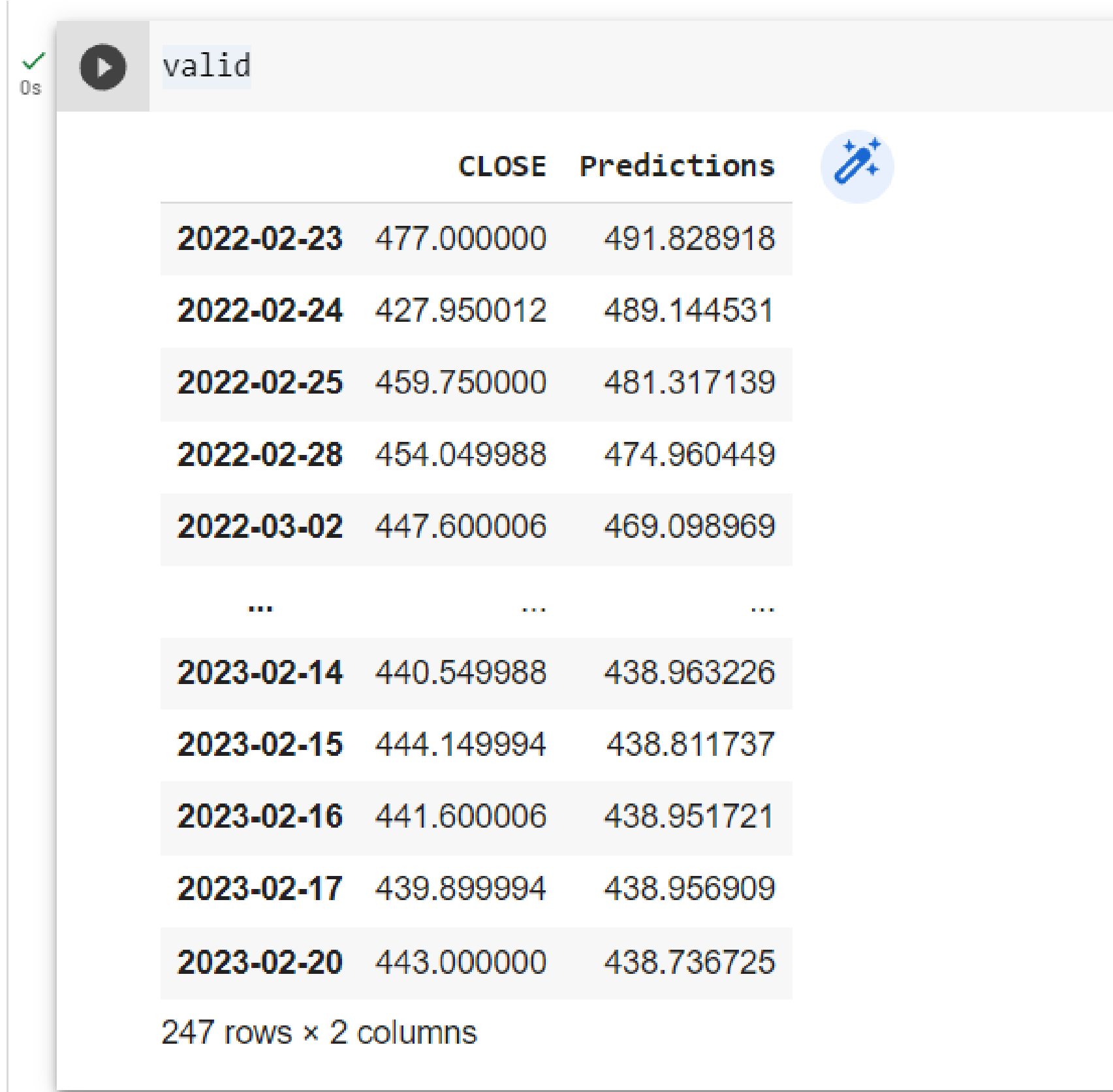
plt.legend(['Train','Val','Predictions'],loc='lower right')

plt.show()



Fig 5.2 Data Analysis Graph

**// Show the valid and predicted prices**



**Chapter 6**

**RESULTS, DISCUSSION AND COMPARISON**

# Eventually, after enough training epochs, it will produce better and better results over time. This is how you would tackle a sequence prediction problem with LSTM. LSTMs are a viable answer for problems involving sequences and time series.

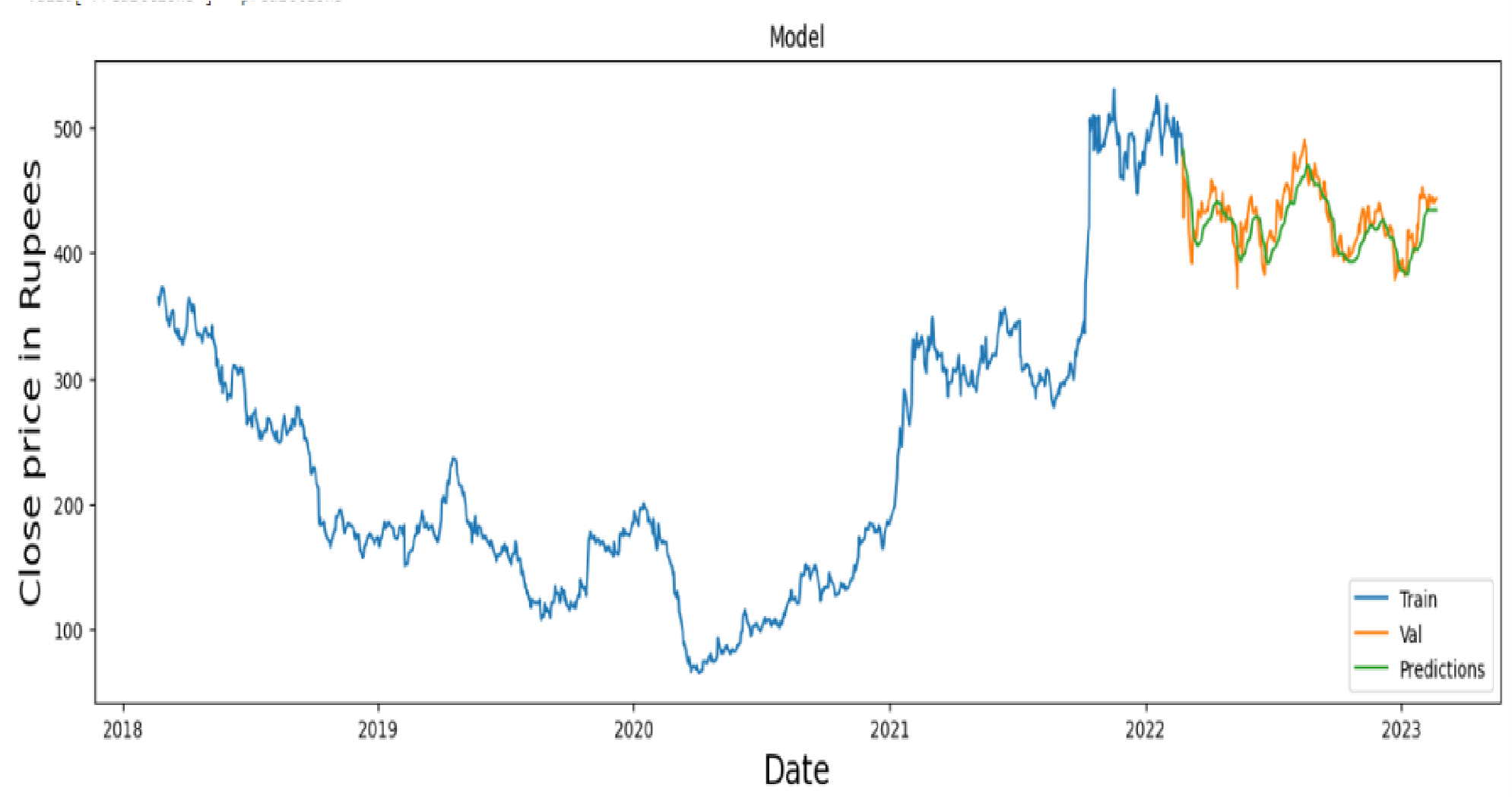
# 6.1 RELIANCE DATA ANALYSIS (2000-2020)

Fig 6.1 Reliance Data Analysis Graph

|  |  |
| --- | --- |
| **EPOCHS** | **RMSE (ROOT MEAN SQUARE ERROR)** |
| **1** | 11.417019872806344 |
| **5** | 11.417019872860344 |
| **10** | 21.590356601658712 |

Table 6.1 RMSE calculations of Reliance data analysis graph

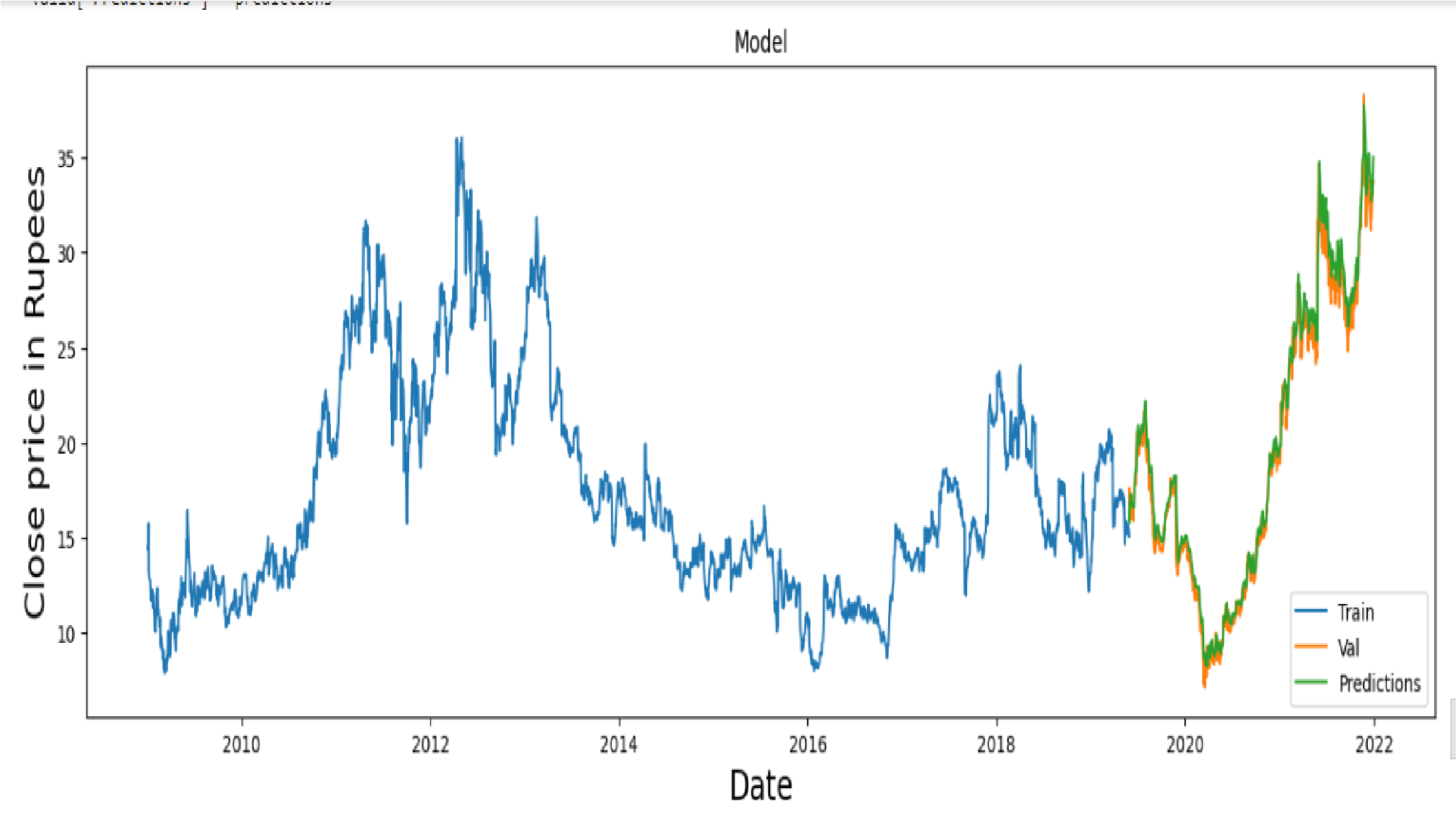
### 6.2 TATA MOTORS DATA ANALYSIS (2018-2023)

Fig 6.2 Tata Motors Data Analysis Graph

|  |  |
| --- | --- |
| **EPOCHS** | **RMSE (ROOT MEAN SQUARE ERROR)** |
| 1 | 15.406698260153087 |
| 5 | 15.30870604957142 |
| 10 | 10.795229264676115 |

Table 6.2 RMSE calculations of TataMotors data analysis graph

### 6.3 TITAN DATA ANALYSIS (2010-2022)

Fig 6.3Titan Data Analysis Graph

|  |  |
| --- | --- |
| **EPOCHS** | **RMSE (ROOT MEAN SQUARE ERROR)** |
| 1 | 0.21497065376636376 |
| 5 | 0.013167143788748906 |
| 10 | 0.5714419731942273 |

Table 6.3 RMSE calculations of Titan data analysis graph

**6.4 ITC DATA ANALYSIS (2016-2022)**

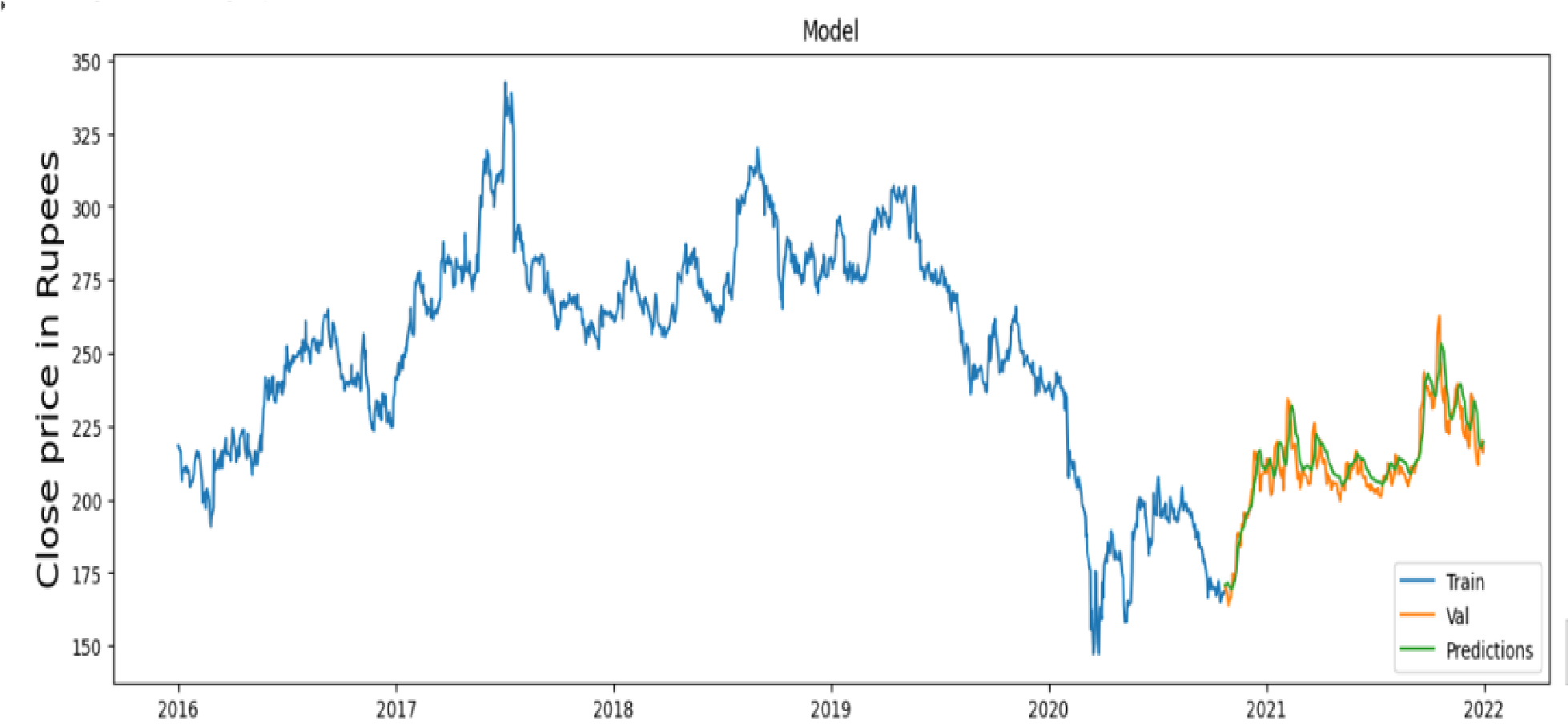


Fig 6.4 ITC Data Analysis Graph

|  |  |
| --- | --- |
| **EPOCHS** | **RMSE (ROOT MEAN SQUARE ERROR)** |
| 1 | 2.468700977209143 |
| 5 | 1.893248891895191 |
| 10 | 4.1621238102136555 |

Table 6.4 RMSE calculations of ITC data analysis graph

**Chapter 7**

**CONCLUSION AND FUTURE ENHANCEMENT**

There are various methods for predicting the stock market at the moment, however, they are less accurate. We suggested a more accurate model that employs RNN and LSTM to forecast the trend in stock prices. In the buried layer of the network, LSTM introduces the memory cell, a computational unit that substitutes typical artificial neurons. The accuracy of prediction is improved in this work by increasing the Epochs and batch size. In the suggested method, we use test data to forecast, which results in more accurate outcomes using the test data. The proposed method is capable of stock market trading andprediction, and the forecast will yield higher and more accurate outcomes. In our above model, we are getting accurate results that will be more useful to stock analysts,Business analysts, and Stock Market Investors. There is a possible scope for further improvement of this model.

**Chapter 8**

**REFERENCES**

1. Stock Price Prediction Using LSTM on the Indian Share Market by Achyut Ghosh,Soumik Bose1, Giridhar Maji, Narayan C. Debnath, Soumya Sen.
2. S. Selvin, R. Vinayakumar, E. A. Gopalkrishnan, V. K. Menon and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in International Conference on Advances in Computing, Communications and Informatics, 2017.
3. Murtaza Roondiwala, Harshal Patel, Shraddha Varma, “Predicting Stock Prices Using LSTM” in Undergraduate Engineering Students, Department of Information Technology, Mumbai University, 2015.
4. Xiongwen Pang, Yanqiang Zhou, Pan Wang, Weiwei Lin, “An innovative neural

network approach for stock market prediction”, 2018.

1. Ishita Parmar, Navanshu Agarwal, Sheirsh Saxena, Ridam Arora, Shikhin Gupta, Himanshu Dhiman, Lokesh Chouhan Department of Computer Science and Engineering National Institute of Technology, Hamirpur – 177005, INDIA - Stock Market Prediction Using Machine Learning.
2. Durmagambetov currently works at CNTFI. He does research in the Theory of Computation and Computing in Mathematics, Natural Science, Engineering and Medicine. Their current project is 'The Riemann Hypothesis-Millennium Prize Problems - stock market predictions
3. Mariam Moukalled Wassim El-Hajj Mohamad Jaber Computer Science Department American University of Beirut - Automated Stock Price Prediction Using Machine Learning.
4. Dharmaraja Selvamuthu, Vineet Kumar and Abhishek Mishra, Department of Mathematics, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016,

India. Indian stock market prediction using artificial neural networks on tick data.

1. <https://builtin.com/data-science/recurrent-neural-networks-and-lstm>
2. https://colah.github.io/posts/2015-08-Understanding-LSTMs/