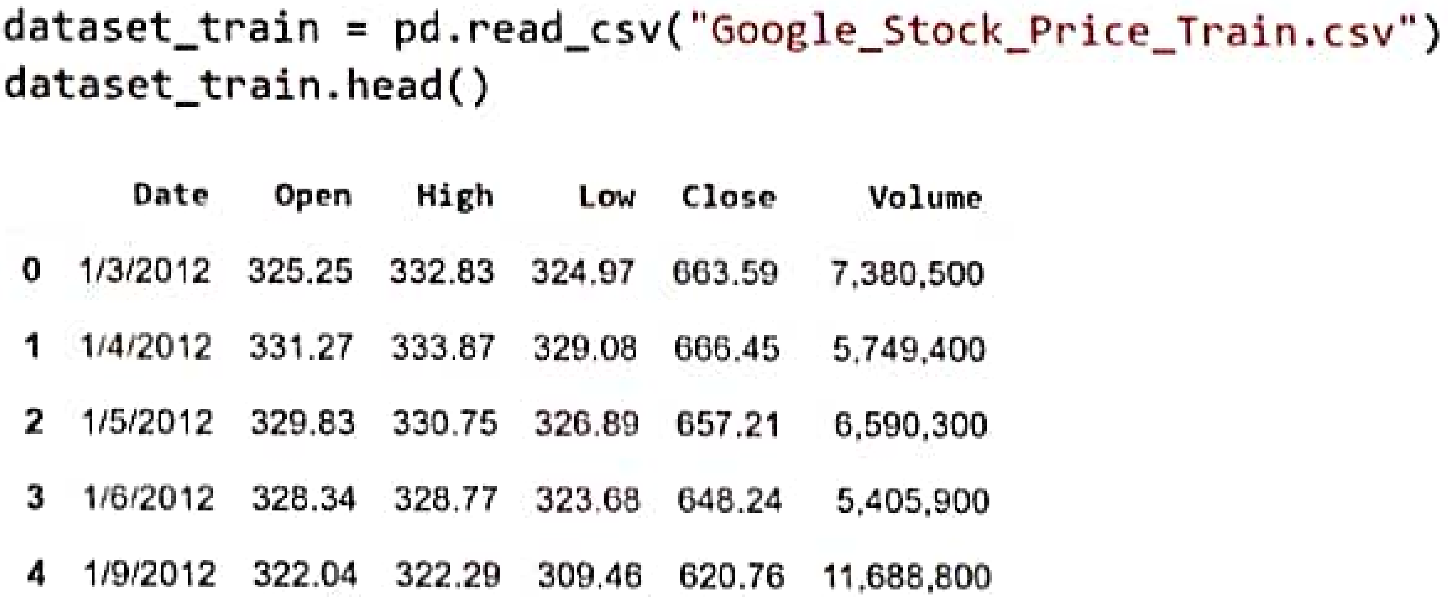


```python
dataset_train = pd.read_csv("Google_Stock_Price_Train.csv")
dataset_train.head()
```

```
       Date    Open    High     Low   Close      Volume
0  1/3/2012  325.25  332.83  324.97  663.59   7,380,500
1  1/4/2012  331.27  333.87  329.08  666.45   5,749,400
2  1/5/2012  329.83  330.75  326.89  657.21   6,590,300
3  1/6/2012  328.34  328.77  323.68  648.24   5,405,900
4  1/9/2012  322.04  322.29  309.46  620.76  11,688,800
```

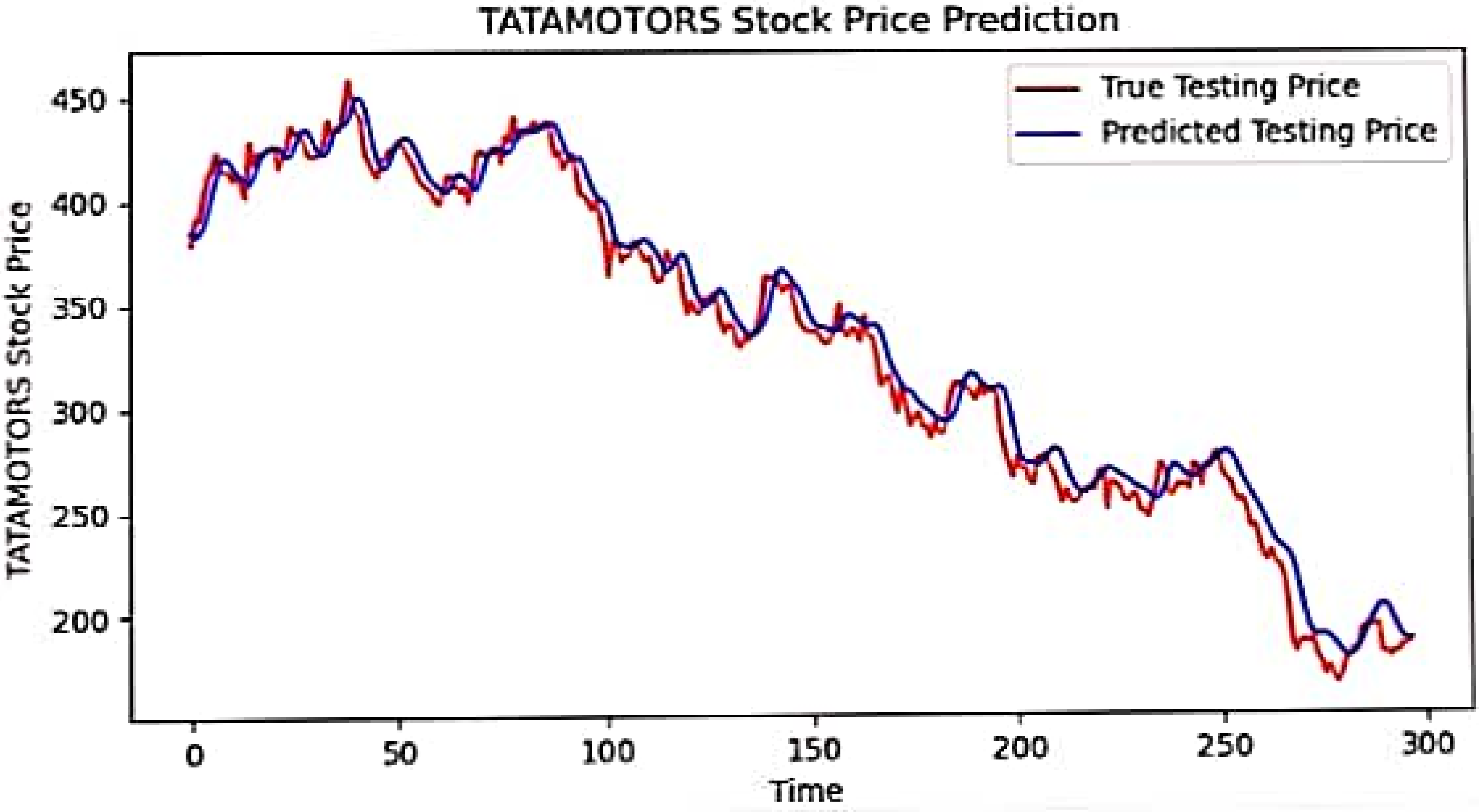
Understanding Long Short Term Memory Network

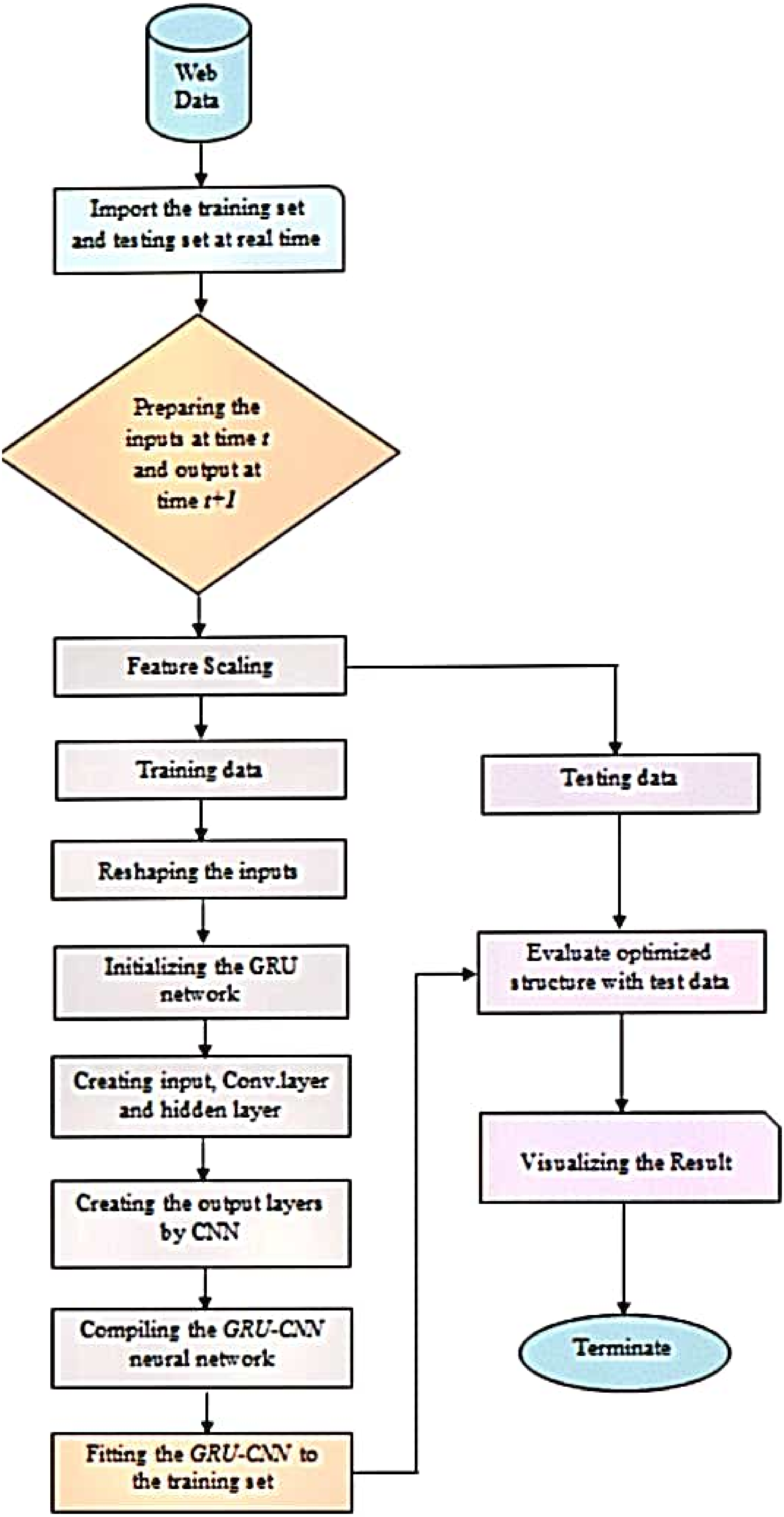
Here, you will use a Long Short Term Memory Network (LSTM) for building your model to predict the stock prices of Google.

LTSMs are a type of Recurrent Neural Network for learning long-term dependencies. It is commonly used for processing and predicting time-series data.

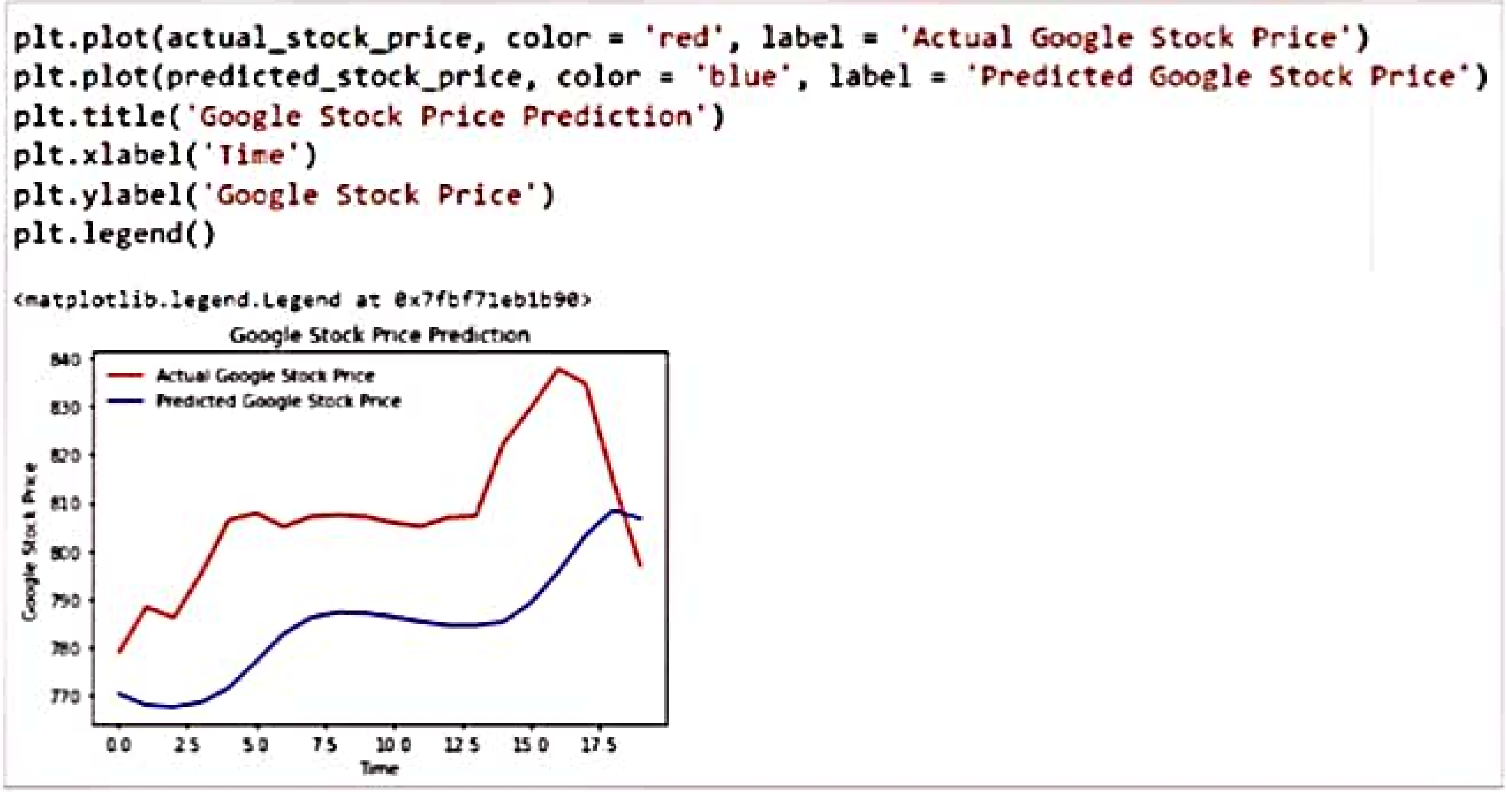
Importance of Stock Market

- Stock markets help companies to raise capital.
- It helps generate personal wealth.
- Stock markets serve as an indicator of the state of the economy.
- It is a widely used source for people to invest money in companies with high growth potential.


TATAMOTORS Stock Price Prediction
True Testing Price
Predicted Testing Price
TATAMOTORS Stock Price
450
400
350
300
250
200
0
50
100
150
200
250
300
Time


Web Data
Import the training set and testing set at real time
Preparing the inputs at time t and output at time t+1
Feature Scaling
Training data
Testing data
Reshaping the inputs
Initializing the GRU network
Evaluate optimized structure with test data
Creating input, Conv.layer and hidden layer
Visualizing the Result
Creating the output layers by CNN
Compiling the GRU-CNN neural network
Terminate
Fitting the GRU-CNN to the training set

12. Plotting the Actual and Predicted Prices for Google Stocks.



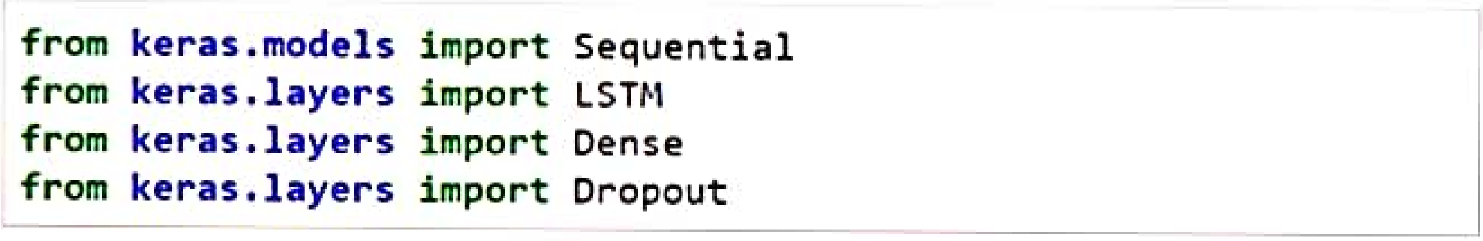
```
plt.plot(actual_stock_price, color = 'red', label = 'Actual Google Stock Price')
plt.plot(predicted_stock_price, color = 'blue', label = 'Predicted Google Stock Price')
plt.title('Google Stock Price Prediction')
plt.xlabel('Time')
plt.ylabel('Google Stock Price')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7fbf71eb1b90>
```

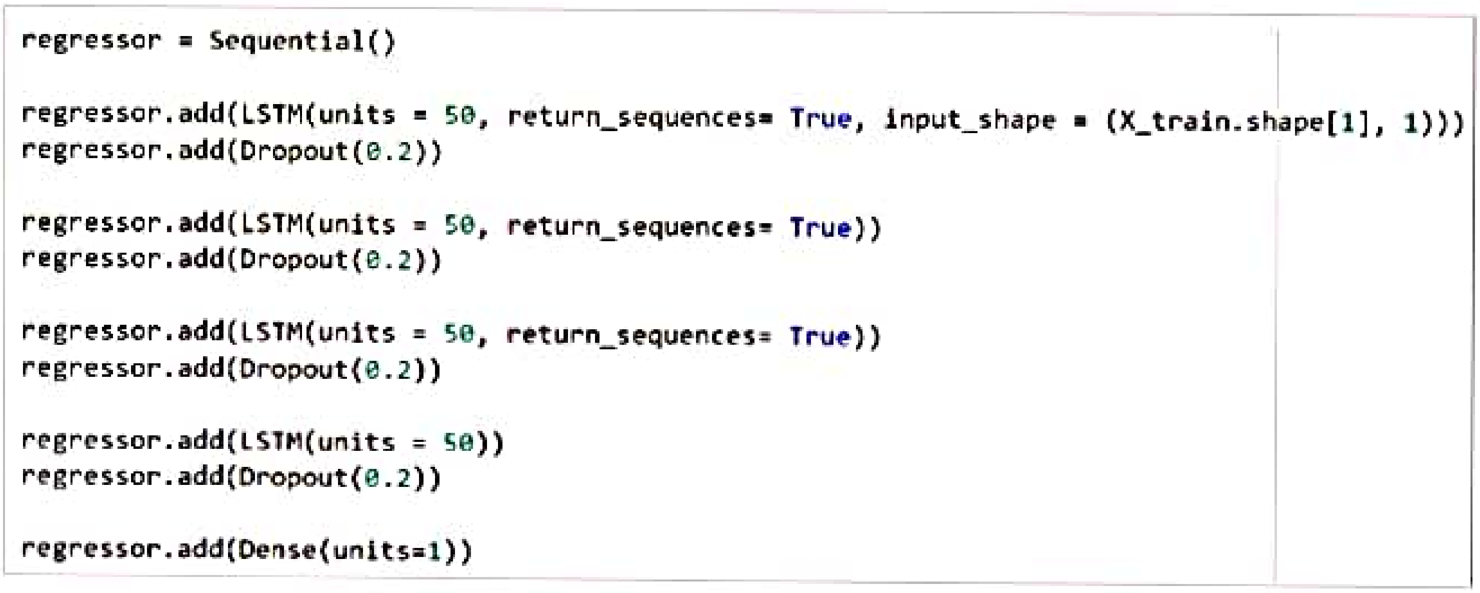
Google Stock Price Prediction
Actual Google Stock Price
Predicted Google Stock Price
Google Stock Price
840
830
820
810
800
790
780
770
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Time

As you can see above, the model can predict the trend of the actual stock prices very closely. The accuracy of the model can be enhanced by training with more data and increasing the LSTM layers.

7. Building the Model by Importing the Crucial Libraries and Adding Different Layers to LSTM.

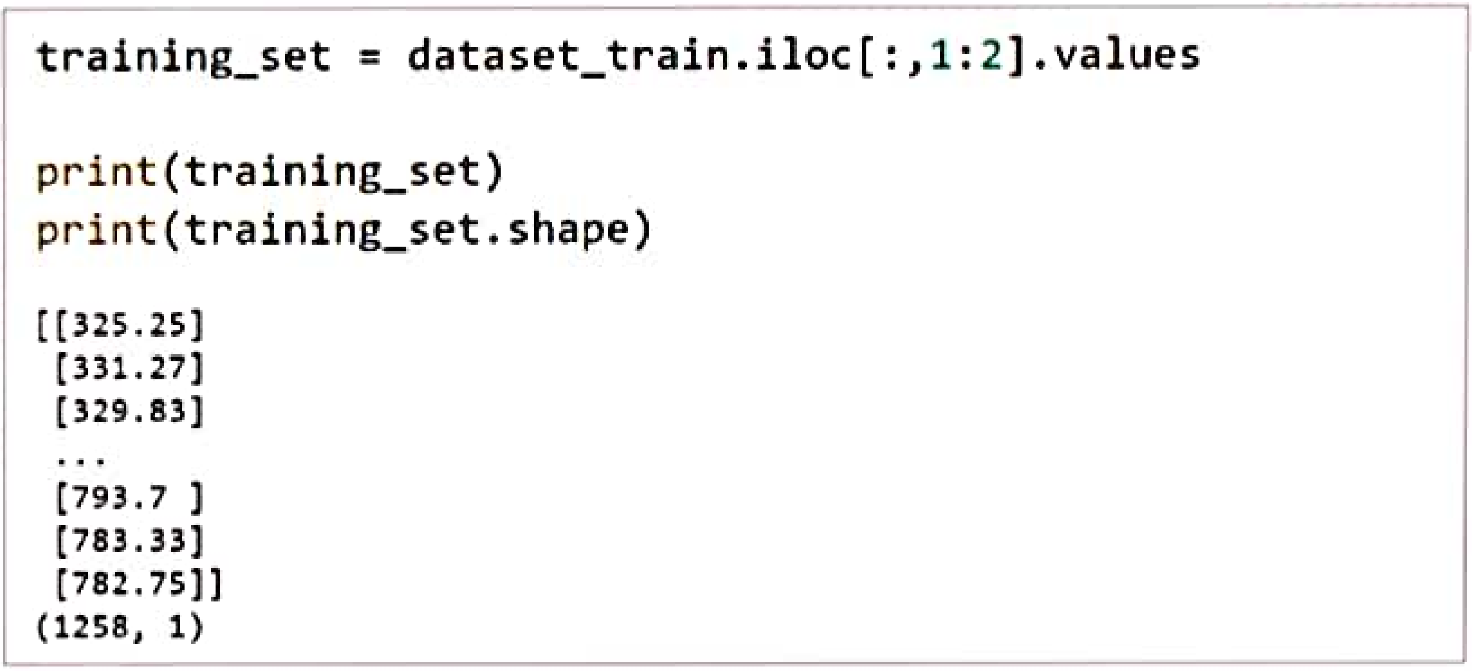


```
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers import Dense
from keras.layers import Dropout
```



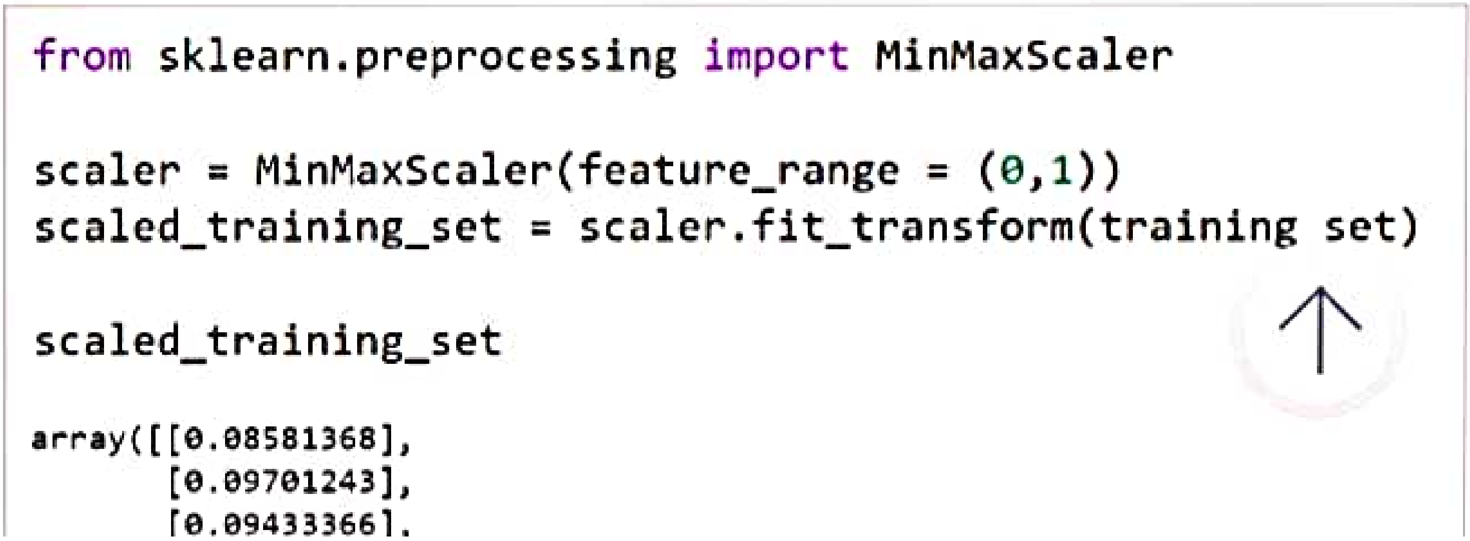
```
regressor = Sequential()

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))
```

3. Use the Open Stock Price Column to Train Your Model.



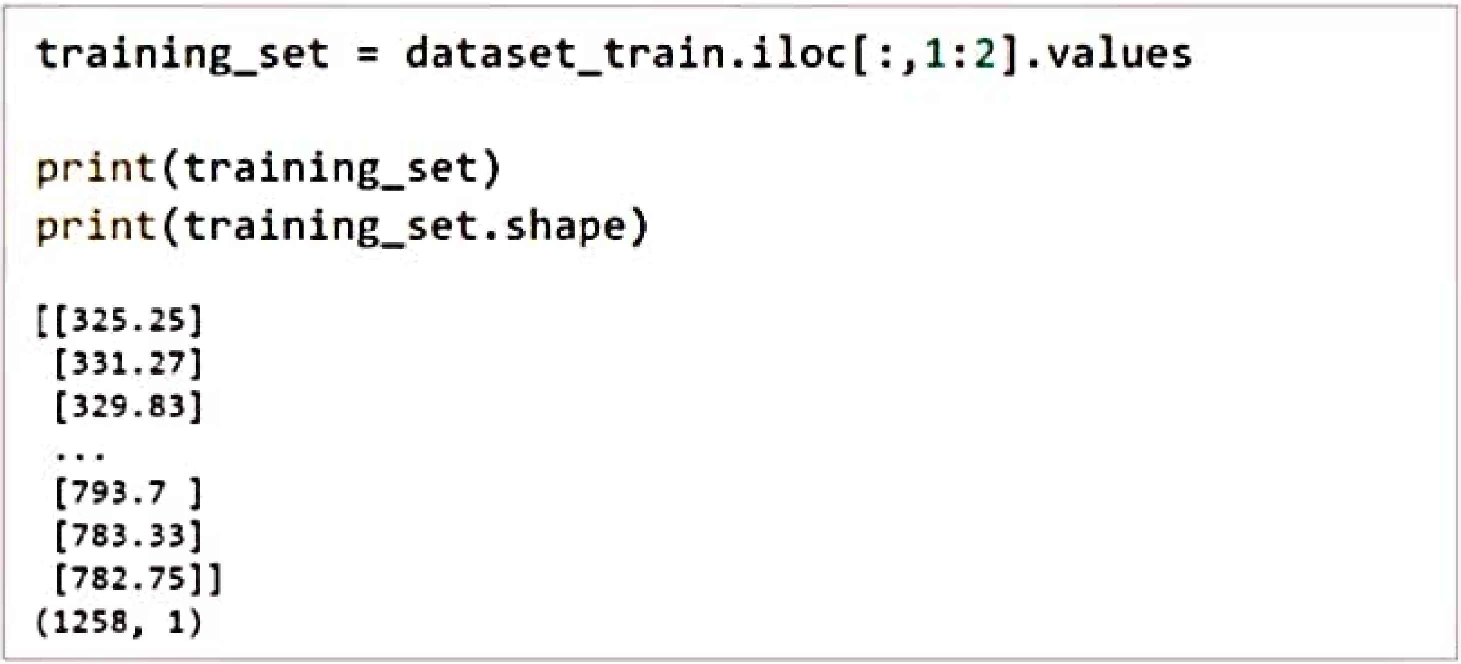
```
training_set = dataset_train.iloc[:,1:2].values

print(training_set)
print(training_set.shape)

[[325.25]
 [331.27]
 [329.83]
 ...
 [793.7 ]
 [783.33]
 [782.75]]
(1258, 1)
```

4. Normalizing the Dataset.



```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler(feature_range = (0,1))
scaled_training_set = scaler.fit_transform(training set)

scaled_training_set

array([[0.08581368],
       [0.09701243],
       [0.09433366],
```

3. Use the Open Stock Price Column to Train Your Model.



```
training_set = dataset_train.iloc[:,1:2].values

print(training_set)
print(training_set.shape)

[[325.25]
 [331.27]
 [329.83]
 ...
 [793.7 ]
 [783.33]
 [782.75]]
(1258, 1)
```