Team No: 12

Team Name: Mind Optimizers

MAKE SKILLED

EARN THE SKILL WITH MAKING

Stock Price Prediction System

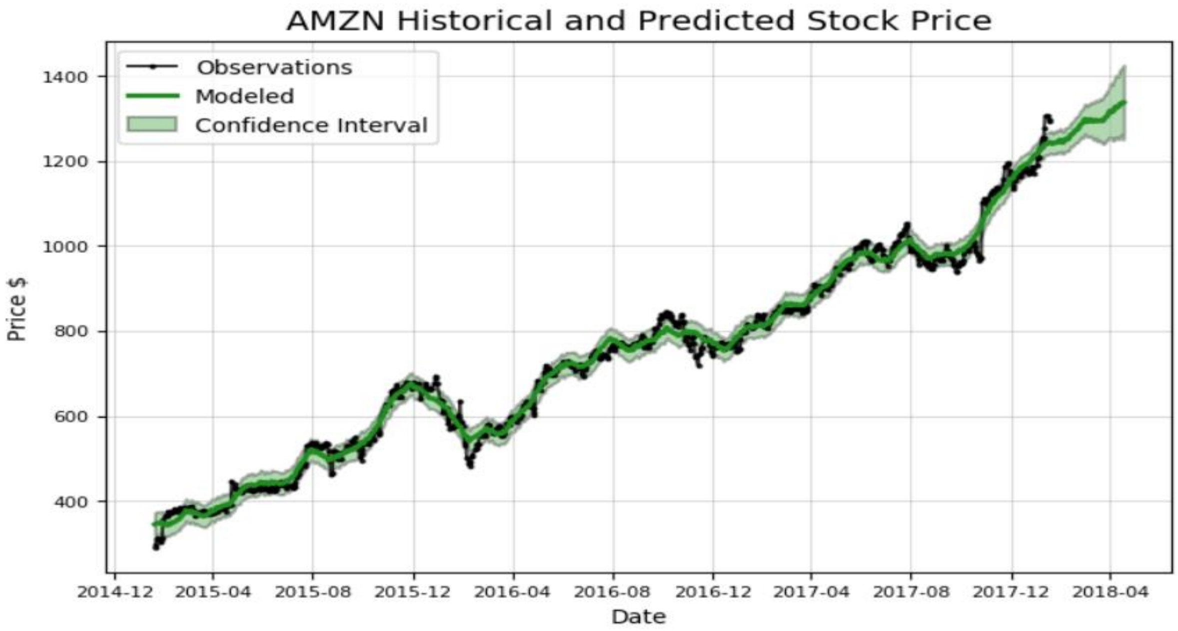
Presented By

D.Surekha	-	20KE1A0520
D.Niharika	-	20KE1A1212
K.Priyanka	-	20KE1A0544
G.Pavani	-	20KE1A0526
P.Madhavi latha	-	20KE1A1251

Malineni
REVIEW
2022

STORY

- The aim is to predict the future value of the financial stocks of a company.
- The price to earnings ratio is likely the ratio most commonly used by investors to predict stock price



PROJECT

- Stock price forecasting is a popular and important topic in financial and academic studies.

- Predicting of Stock Market Prices using Deep Learning LSTM Model

- The process of predicting the future value of a stock trade or stock exchange for reaping profits.

EXISTING SYSTEM

- The prediction of future stock price by SlidingWindowAlogorithm is less efficient because of processing unwanted data

- The SlidingWindowAlogorithm which is used in existing system is not much effective in handling non linear data

- So in our project the future stock price prediction is done using LSTM

- The LSTM means Long Short Term Memory which is more efficient than SlidingWindowAlogorithm

TOOL

- The tool that is used for project is ‘vscode’

Packages & Why?

□ streamlit:

- It helps us create web apps for data science and machine learning in a short time.

□ numpy:

- It provides a high-performance multidimensional array object, and tools for working with these arrays.

❑pandas:

- ➢Pandas is mainly used for data analysis and associated manipulation of tabular data in dataframes

❑nsepy:

- ➢nsepy is a library to extract historical and realtime data from NSE’s website

❑ sklearn:

➢ Scikit-learn is a key library for the Python programming language that is typically used in machine learning projects

❑ tensorflow:

➢ tensorflow allows developers to create dataflow graphs-structures that describe how data moves through a graph

Programming Languages

- Python is the language used for these project
- Python is used to forecast the stock market prediction

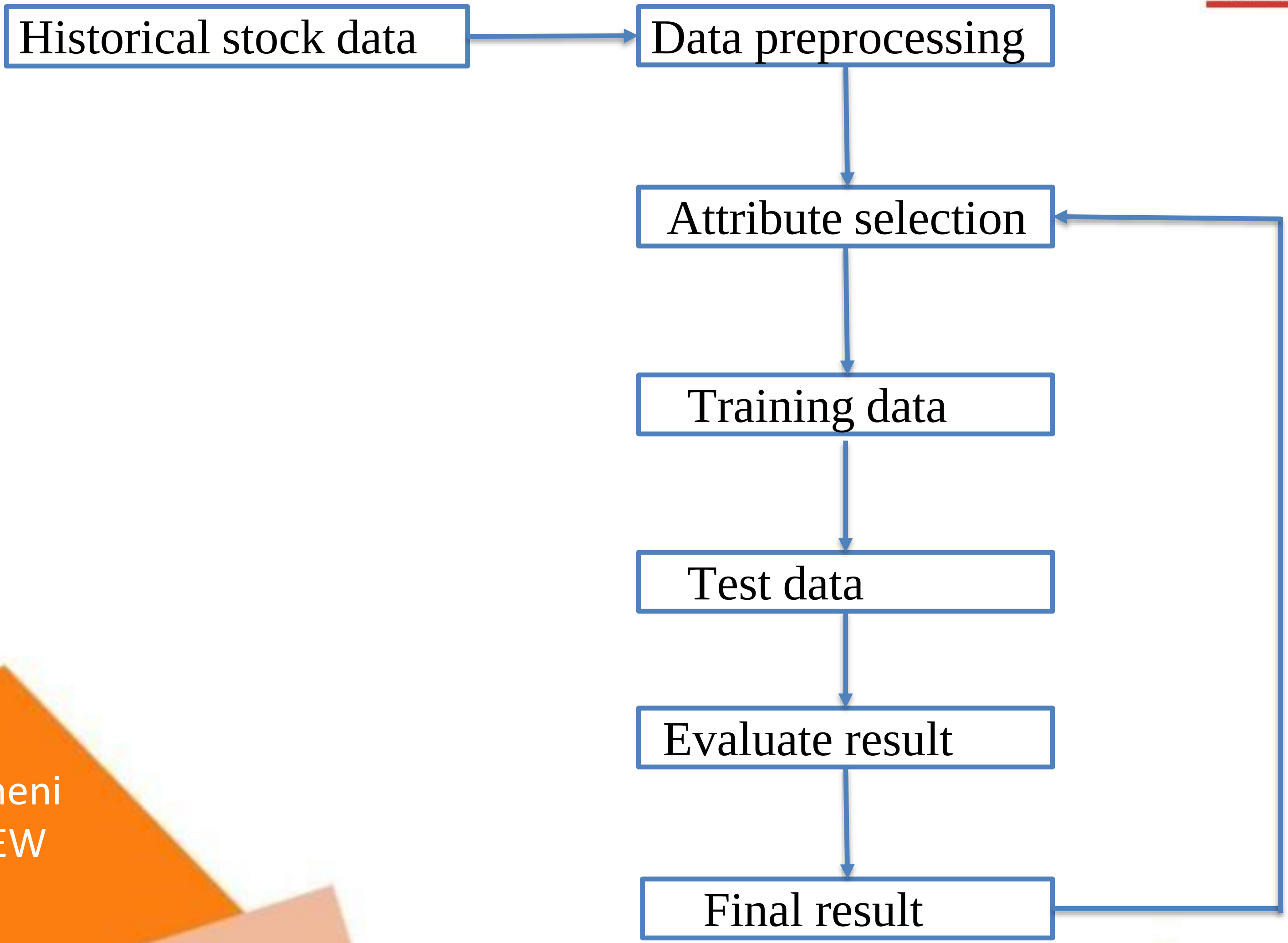
Pros of the project

- Grow with economy
- Stay ahead of inflation
- Easy to Buy
- Don’t need a lot of money to start investing
- Income from price appreciation and dividends
- Liquidity

cons of the project

- Risk
- Stockholders of broke companies get paid last
- Takes time to research
- Taxes on profitable stock sales
- Emotional ups and downs
- Competing with institutional and professional investors

Block Diagram



Historical stock data

Data preprocessing

Attribute selection

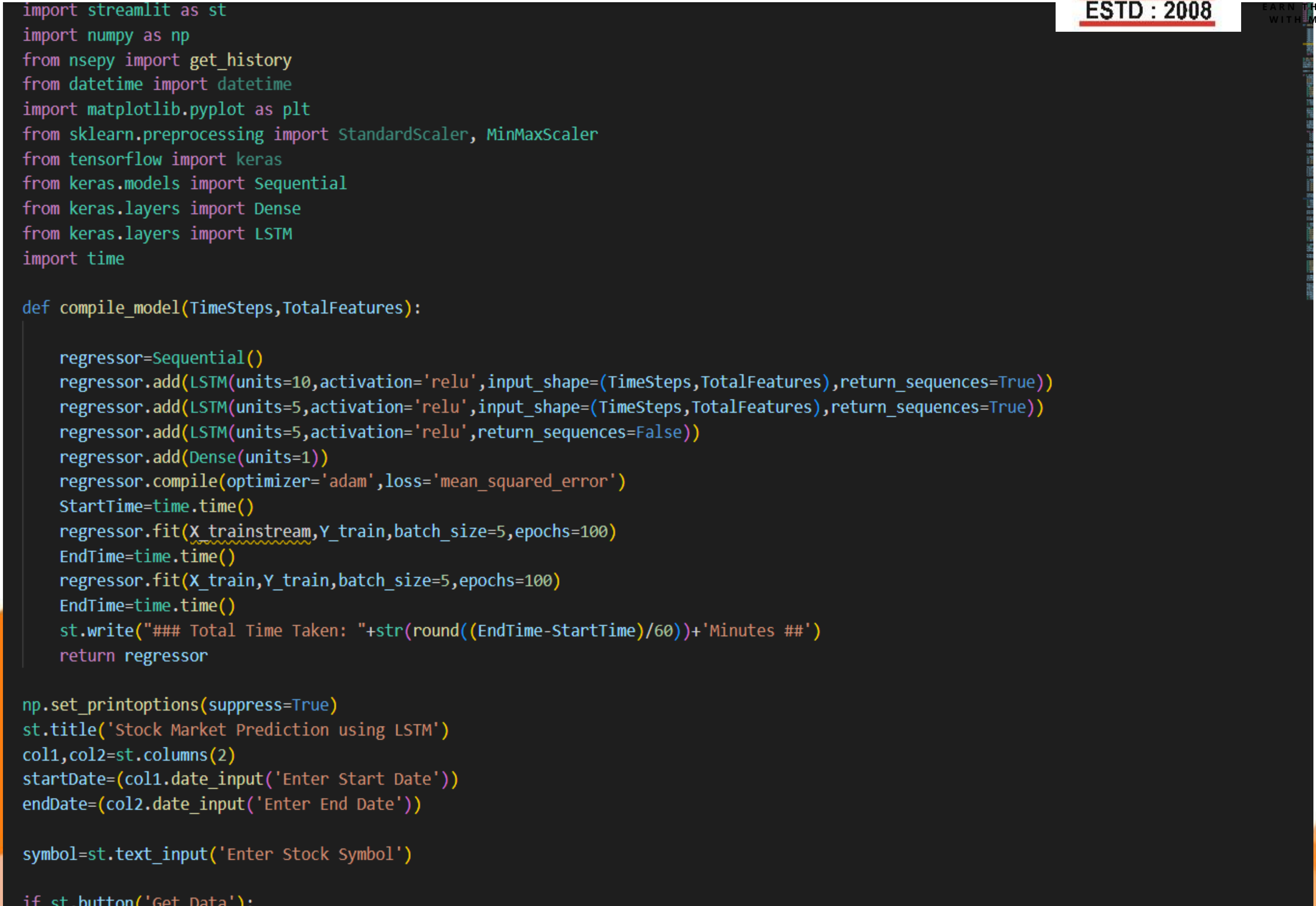
Training data

Test data

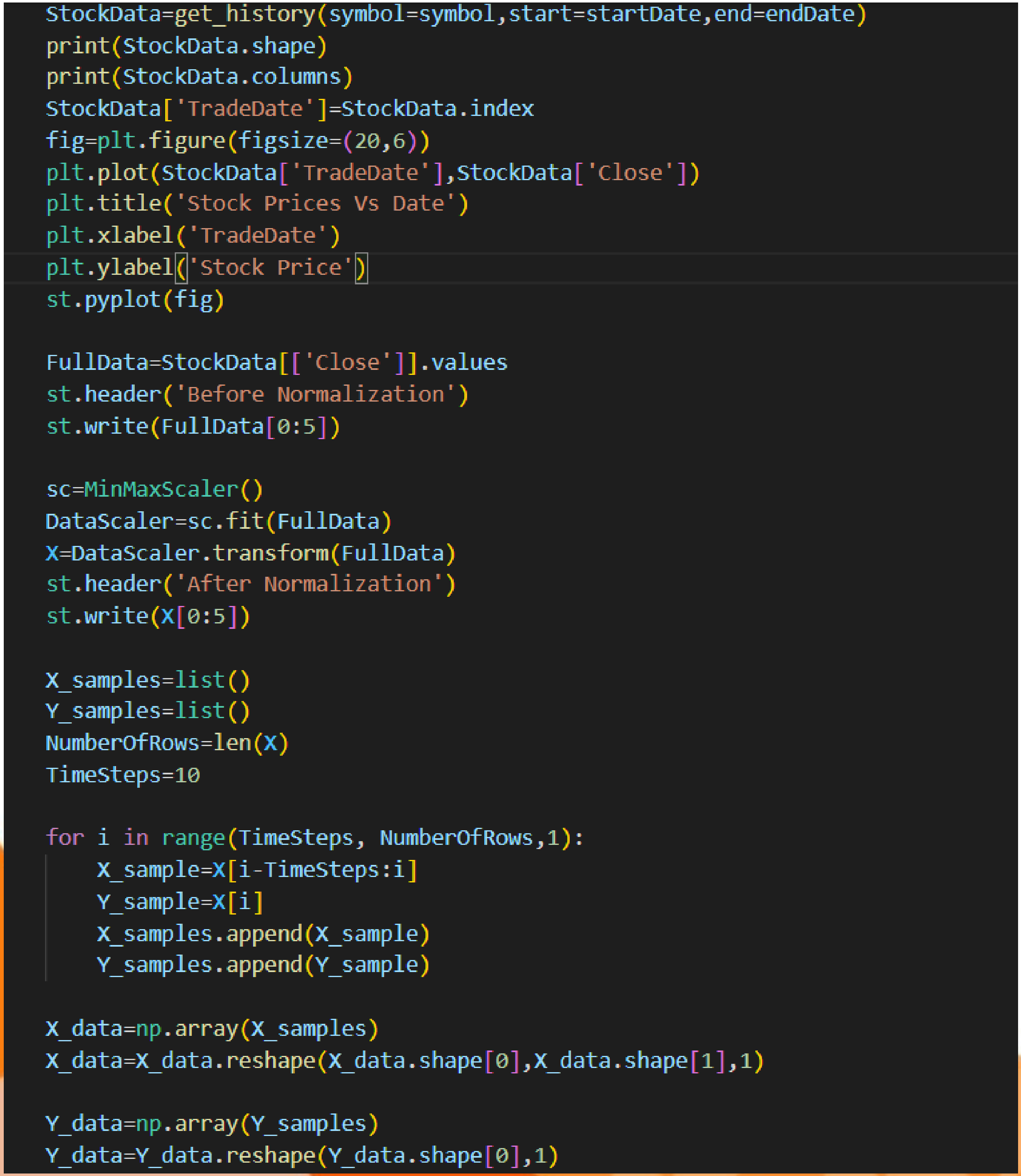
Evaluate result

Final result

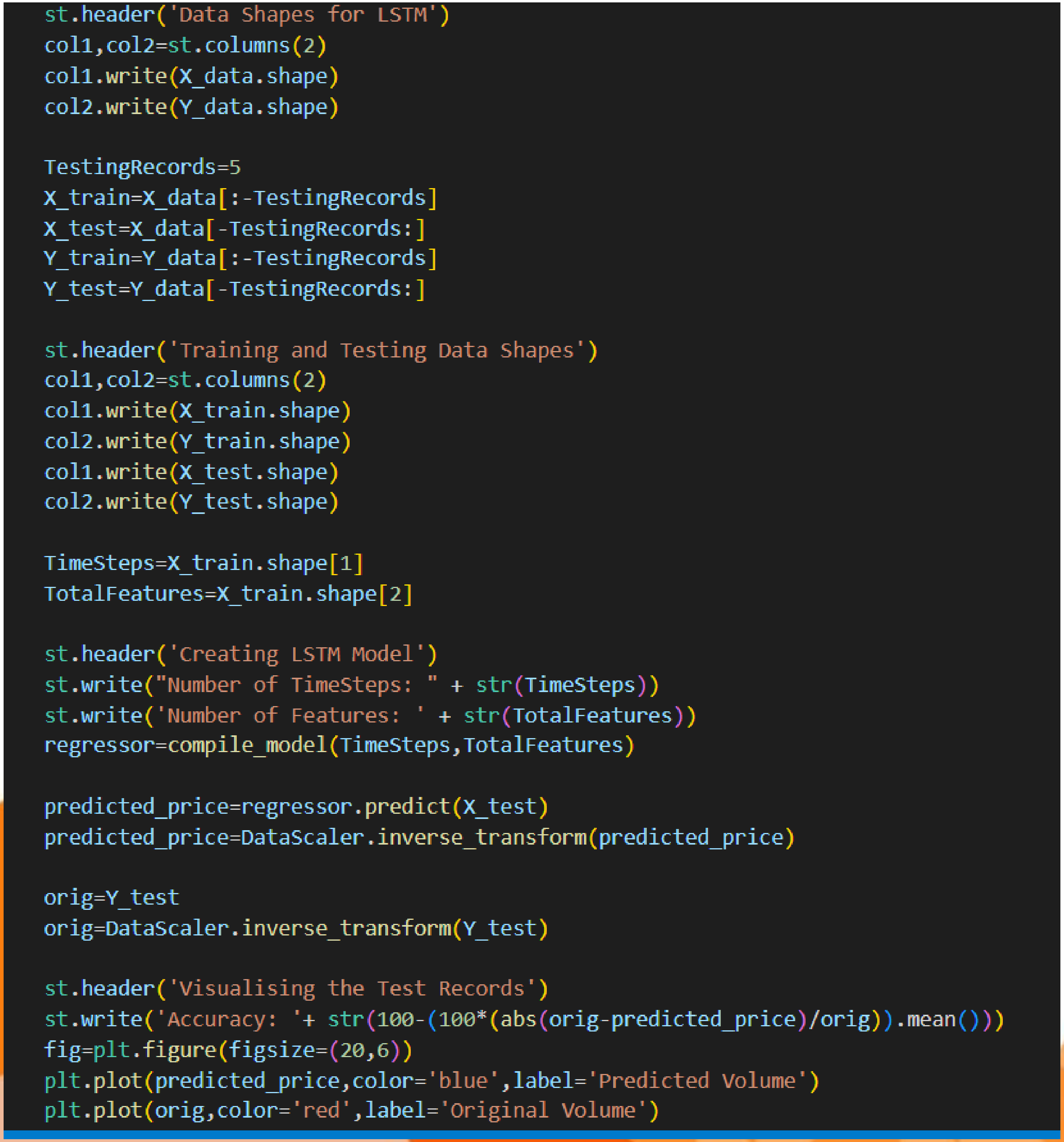
Working Procedure



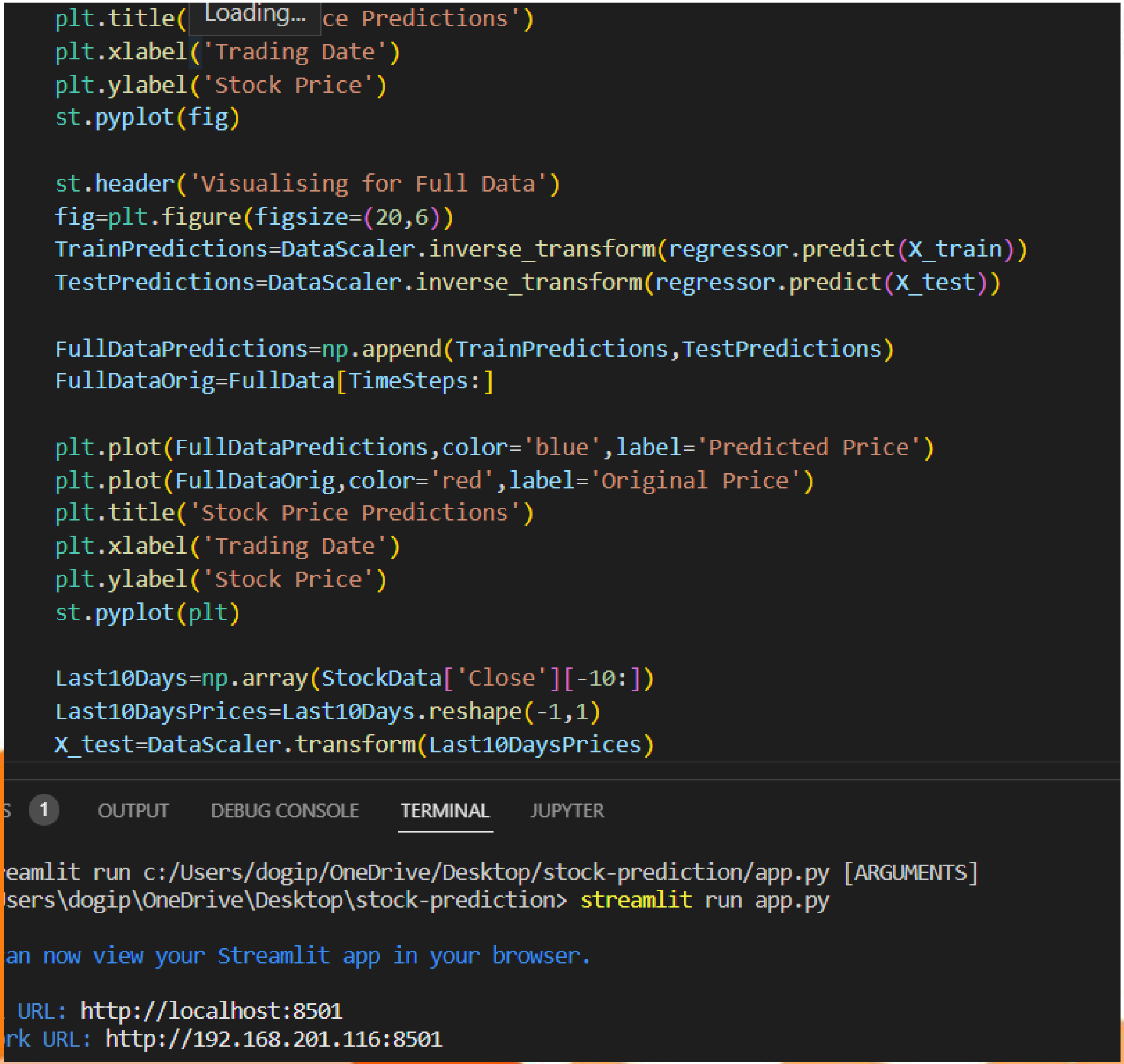
```
import streamlit as st
import numpy as np
from nsepy import get_history
from datetime import datetime
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from tensorflow import keras
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
import time

def compile_model(TimeSteps,TotalFeatures):

    regressor=Sequential()
    regressor.add(LSTM(units=10,activation='relu',input_shape=(TimeSteps,TotalFeatures),return_sequences=True))
    regressor.add(LSTM(units=5,activation='relu',input_shape=(TimeSteps,TotalFeatures),return_sequences=True))
    regressor.add(LSTM(units=5,activation='relu',return_sequences=False))
    regressor.add(Dense(units=1))
    regressor.compile(optimizer='adam',loss='mean_squared_error')
    StartTime=time.time()
    regressor.fit(X_trainstream,Y_train,batch_size=5,epochs=100)
    EndTime=time.time()
    regressor.fit(X_train,Y_train,batch_size=5,epochs=100)
    EndTime=time.time()
    st.write("### Total Time Taken: "+str(round((EndTime-StartTime)/60))+'Minutes ##')
    return regressor

np.set_printoptions(suppress=True)
st.title('Stock Market Prediction using LSTM')
col1,col2=st.columns(2)
startDate=(col1.date_input('Enter Start Date'))
endDate=(col2.date_input('Enter End Date'))

symbol=st.text_input('Enter Stock Symbol')

if st.button('Get Data'):
```



```
StockData=get_history(symbol=symbol,start=startDate,end=endDate)
print(StockData.shape)
print(StockData.columns)
StockData['TradeDate']=StockData.index
fig=plt.figure(figsize=(20,6))
plt.plot(StockData['TradeDate'],StockData['Close'])
plt.title('Stock Prices Vs Date')
plt.xlabel('TradeDate')
plt.ylabel('Stock Price')
st.pyplot(fig)

FullData=StockData[['Close']].values
st.header('Before Normalization')
st.write(FullData[0:5])

sc=MinMaxScaler()
DataScaler=sc.fit(FullData)
X=DataScaler.transform(FullData)
st.header('After Normalization')
st.write(X[0:5])

X_samples=list()
Y_samples=list()
NumberOfRows=len(X)
TimeSteps=10

for i in range(TimeSteps, NumberOfRows,1):
    X_sample=X[i-TimeSteps:i]
    Y_sample=X[i]
    X_samples.append(X_sample)
    Y_samples.append(Y_sample)

X_data=np.array(X_samples)
X_data=X_data.reshape(X_data.shape[0],X_data.shape[1],1)

Y_data=np.array(Y_samples)
Y_data=Y_data.reshape(Y_data.shape[0],1)
```



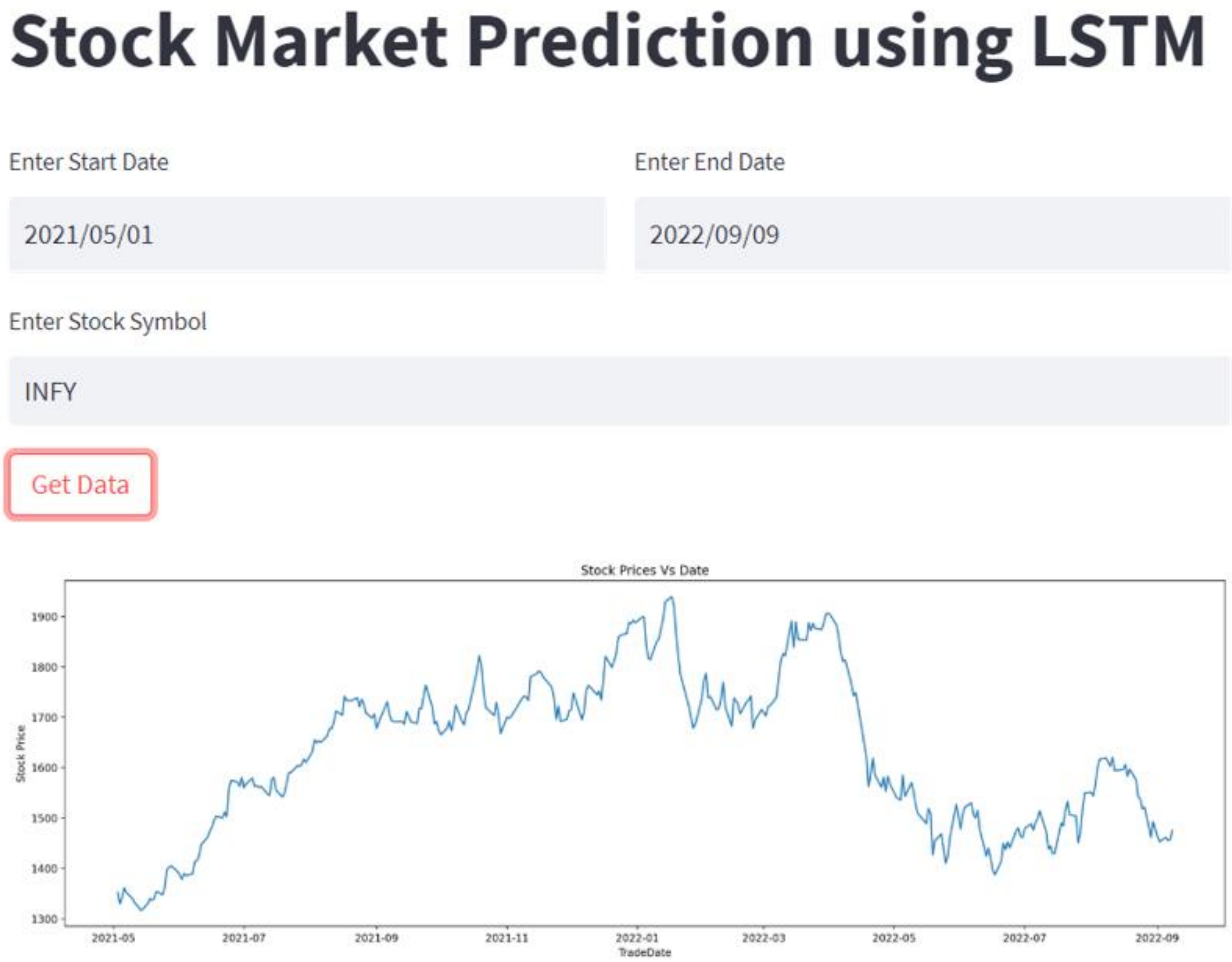
```
    st.header('Data Shapes for LSTM')
    col1,col2=st.columns(2)
    col1.write(X_data.shape)
    col2.write(Y_data.shape)

    TestingRecords=5
    X_train=X_data[:-TestingRecords]
    X_test=X_data[-TestingRecords:]
    Y_train=Y_data[:-TestingRecords]
    Y_test=Y_data[-TestingRecords:]

    st.header('Training and Testing Data Shapes')
    col1,col2=st.columns(2)
    col1.write(X_train.shape)
    col2.write(Y_train.shape)
    col1.write(X_test.shape)
    col2.write(Y_test.shape)

    TimeSteps=X_train.shape[1]
    TotalFeatures=X_train.shape[2]

    st.header('Creating LSTM Model')
    st.write("Number of TimeSteps: " + str(TimeSteps))
    st.write('Number of Features: ' + str(TotalFeatures))
    regressor=compile_model(TimeSteps,TotalFeatures)

    predicted_price=regressor.predict(X_test)
    predicted_price=DataScaler.inverse_transform(predicted_price)

    orig=Y_test
    orig=DataScaler.inverse_transform(Y_test)

    st.header('Visualising the Test Records')
    st.write('Accuracy: '+ str(100-(100*(abs(orig-predicted_price)/orig)).mean()))
    fig=plt.figure(figsize=(20,6))
    plt.plot(predicted_price,color='blue',label='Predicted Volume')
    plt.plot(orig,color='red',label='Original Volume')
```



```
plt.title( Loading... ce Predictions')
plt.xlabel('Trading Date')
plt.ylabel('Stock Price')
st.pyplot(fig)

st.header('Visualising for Full Data')
fig=plt.figure(figsize=(20,6))
TrainPredictions=DataScaler.inverse_transform(regressor.predict(X_train))
TestPredictions=DataScaler.inverse_transform(regressor.predict(X_test))

FullDataPredictions=np.append(TrainPredictions,TestPredictions)
FullDataOrig=FullData[TimeSteps:]

plt.plot(FullDataPredictions,color='blue',label='Predicted Price')
plt.plot(FullDataOrig,color='red',label='Original Price')
plt.title('Stock Price Predictions')
plt.xlabel('Trading Date')
plt.ylabel('Stock Price')
st.pyplot(plt)

Last10Days=np.array(StockData['Close'][-10:])
Last10DaysPrices=Last10Days.reshape(-1,1)
X_test=DataScaler.transform(Last10DaysPrices)
```

```
eamlit run c:/Users/dogip/OneDrive/Desktop/stock-prediction/app.py [ARGUMENTS]
Jsers\dogip\OneDrive\Desktop\stock-prediction> streamlit run app.py

an now view your Streamlit app in your browser.

 URL: http://localhost:8501
rk URL: http://192.168.201.116:8501
```

Results

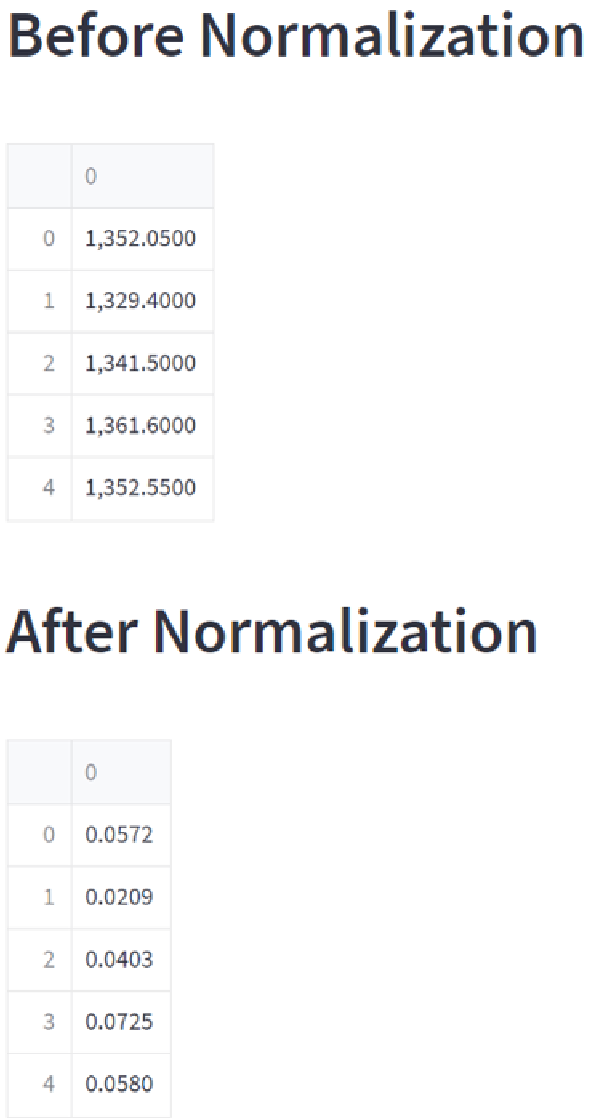


Stock Market Prediction using LSTM

Enter Start Date: 2021/05/01

Enter End Date: 2022/09/09

Enter Stock Symbol: INFY

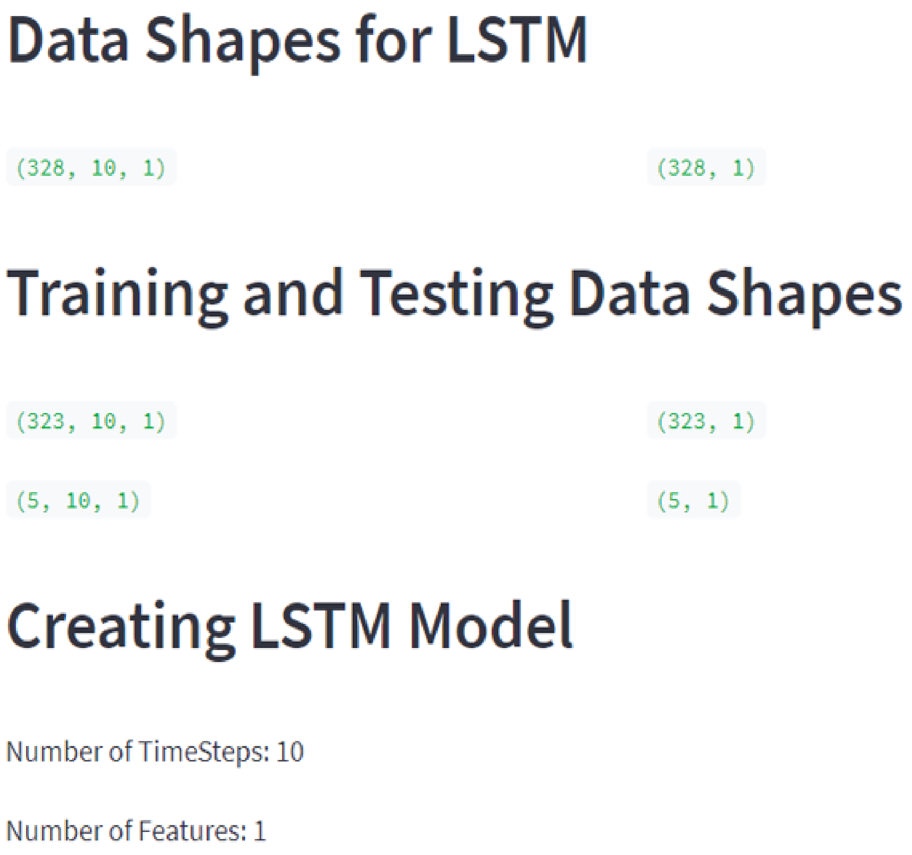


Before Normalization

	0
0	1,352.0500
1	1,329.4000
2	1,341.5000
3	1,361.6000
4	1,352.5500

After Normalization

	0
0	0.0572
1	0.0209
2	0.0403
3	0.0725
4	0.0580



Data Shapes for LSTM

```
(328, 10, 1)                                          (328, 1)
```

Training and Testing Data Shapes

```
(323, 10, 1)                                          (323, 1)

(5, 10, 1)                                            (5, 1)
```

Creating LSTM Model

Number of TimeSteps: 10

Number of Features: 1

CONCLUSION

- We learn that our project is to gain significant profits and predicting how the stock market will perform is hard in our daily life.

- The outcome is successful prediction of stock's future price could yield significant profit.

- We achieve the future value of company stock and other financial assets traded on an exchange.

Thank You!