STOCK PRICE PREDICTION

A Project Progress Report

Submitted in partial fulfillment for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by

ANINGI LIKHITH	(N170307)
ABOTHULA SATISH	(N170455)
DANTA JEEVAN SAI	(N170466)
POKURI NAGA MANEENDRA	(N170465)
VAJJA VINOD KUMAR	(N170482)

Under the Esteem Guidance of

Mrs. M Baby Anusha

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Rajiv Gandhi University of Knowledge Technologies – Nuzvid

Nuzvid, Krishna, Andhra Pradesh – 521202.

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Rajiv Gandhi University of Knowledge Technologies – Nuzvid

Nuzvid, Krishna, Andhra Pradesh – 521202.

CERTIFICATE OF COMPLETION

This is to certify that the work entitled, **"Stock Price Prediction"** is the bonafied work of **A.LIKHITH (ID No*:* N170307), D.JEEVAN SAI (ID No*:* N170466), A.SATISH *(*ID No*:* N170455),P.NAGA MANEENDRA *(*ID No*:*N170465), V.VINOD KUMAR (ID No :N170482)** carried out under my guidance and supervision for 3rd year mini project of **Bachelor of Technology** in the department of Computer Science and Engineering under RGUKT IIIT Nuzvid. This work is done during the academic session April 2022 – September 2022, under our guidance.

Mrs. M Baby Anusha

Assistant professor

Department of CSE

RGUKT Nuzvid

Mr. Chiranjeevi Sadu

Assistant Professor,

Department of CSE

Head of the Department

RGUKT Nuzvid.

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Rajiv Gandhi University of Knowledge Technologies – Nuzvid

Nuzvid, Krishna, Andhra Pradesh – 521202.

CERTIFICATE OF EXAMINATION

This is to certify that the work entitled, **"Stock Price Prediction"** is the bonafied work of **A.LIKHITH (ID No*:* N170307), D.JEEVAN SAI (ID No*:* N170466), A.SATISH *(*ID No*:* N170455),P.NAGA MANEENDRA *(*IDNo*:*N170465), V.VINOD KUMAR (ID No :N170482)** and here by accord our approval of it as a study carried out and presented in a manner required for its acceptance in final year of **Bachelor of Technology** for which it has been submitted. This approval does not necessarily endorse or accept every statement made, opinion expressed or conclusion drawn, as a recorded in this thesis. It only signifies the acceptance of this thesis for the purpose for which it has been submitted.

Mrs. M Baby Anusha

Assistant Professor,

Department of CSE,

Project Examiner

RGUKT-Nuzvid

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

Rajiv Gandhi University of Knowledge Technologies – Nuzvid

Nuzvid, Krishna, Andhra Pradesh – 521202.

DECLARATION

we "**A.LIKHITH (ID No*:* N170307), D.JEEVAN SAI (ID No*:* N170466), A.SATISH *(*ID No*:* N170455),P.NAGA MANEENDRA *(*IDNo*:*N170465), V.VINOD KUMAR (ID No :N170482)"** hereby declare that the project report entitled **"Stock Price Prediction"** done by us under the guidance of Mrs. M Baby Anusha, Assistant Professor is submitted for the partial fulfillment for the award of degree of Bachelor of Technology in Computer Science and Engineering during the academic session April 2022 – September 2022 at RGUKT-Nuzvid.

We also declare that this project is a result of our own effort and has not been copied or imitated from any source. Citations from any websites are mentioned in the references. The results embodied in this project report have not been submitted to any other university or institute for the award of any degree or diploma.

Date: 12-09-2022

Place: Nuzvid

A LIKITH (N170307)

A SATISH (N170455)

D JEEVAN SAI (N170466)

P NAGA MANEENDRA(N170465)

V VINOD KUMAR(N170482)

ACKNOWLEDGEMENT

We would like to express our profound gratitude and deep regards to our guide **Mrs. M Baby Anusha** for her exemplary guidance, monitoring and constant encouragement to us throughout the Project Period. We shall always cherish the time spent with her during the course of this work due to the invaluable knowledge gained in the field of reliability engineering.

We are extremely grateful for the confidence bestowed in us and entrusting our project entitled **"Stock Price Prediction"**

We express gratitude to Mr. Chiranjeevi sadhu (HOD of CSE) and other faculty members for being source of inspiration and constant encouragement which helped us in completing the project successfully.

Table of Contents

ABSTRACT

A stock market, equity market or share market is the aggregation of buyers and sellers of stocks (also called shares), which represent ownership claims on businesses. The task of predicting stock prices is one of the difficult tasks for many analysts and in fact for investors. For a successful investment, many investors are very keen in predicting the future ups and down of share in the market. Good and effective prediction models help investors analysts to predict the future of the stock market. In this project, We had proposed Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) model by using Machine and Deep Learning models to predict stock market prediction. In present, there are several models to predict the stock market but they are less accurate. We had proposed a model that uses RNN and LSTM to predict the trend in stock prices that would be more accurate. LSTM introduces the memory cell, a unit of computation that replaces traditional artificial neurons in the hidden layer of the network. In this work by increasing the Epochs and batch size, the accuracy of prediction is more. In proposed method, We are using a test data that is used to predict which gives results that are more accurate with the test data. The proposed method is capable of tracing and prediction of stock market and the prediction will produce higher and accurate results.

***Keywords*:** Stock Market Prediction, Recurrent Neural Network (RNN), ,Long Short Term Memory (LSTM), Epochs, batch size, Stock Price.

INTRODUCTION

Our project is recurrent neural network based Stock price prediction using machine learning. For a successful investment, many investors are very keen in predicting the future ups and down of share in the market. Good and effective prediction models help investors and analysts to predict the future of the stock market. In this project, We had proposed a Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) model by using Machine and Deep Learning models to predict stock market prediction. In present, there are several models to predict the stock market but they are less accurate. We had proposed a model that uses RNN and LSTM to predict the trend in stock prices that would be more accurate. LSTM introduces the memory cell, a unit of computation that replaces traditional artificial neurons in the hidden layer of the network. In this work by increasing the Epochs and batch size, the accuracy of prediction is more. In proposed method, We are using a test data that is used to predict which gives results that are more accurate with the test data. The proposed method is capable of tracing and prediction of stock market and the prediction will produce higher and accurate results.

1.1 TensorFlow:

TensorFlow is an open-source software library. **TensorFlow** was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well!

TensorFlow is basically a software library for numerical computation using data flowgraphs where nodes in the graph represent mathematical operations and edges in the graph represent the multidimensional data arrays (called **tensors**) communicated between them.

Tensor flow 2.0 has been beta since last year, and it is a completely different universe as its predecessor Tensor flow 1.0 but even in 2020 it is important to understand the history and evolution of TF library to understand how did it get from here, and why did it choose Keras as a high-level API. It is important to understand what is a compute graph as it is a super useful concept in Deep Learning and that you can still visualize and inspect it in Tensor Board.

1.2 Numpy

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

Numpy for Machine Learning:

NumPy library is an important foundational tool for studying Machine Learning. Many of its functions are very useful for performing any mathematical or scientific calculation. As it is known that mathematics is the foundation of machine learning, most of the mathematical tasks can be performed using NumPy.

1.3 Pandas

Pandas is an open-source library that is built on top of NumPy library. It is a Python package that offers various data structures and operations for manipulating numerical data and time series. It is mainly popular for importing and analyzing data much easier. Pandas is fast and it has high-performance & productivity for users.

Pandas for Machine Learning:

Pandas is one of the tools in Machine Learning which is used for data cleaning and analysis. It has features which are used for exploring, cleaning, transforming and visualizing from data.

1.4 Sciket Learning:

Scikit-learn is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

Sciket Learning in Machine Learning:

- Simple and efficient tools for data mining and data analysis. It features various classification, regression and clustering algorithms.
- Accessible to everybody and reusable in various contexts.
- Built on the top of NumPy, SciPy, and matplotlib.

1.5 Matplot Lib:

Matplotlib is easy to use and an amazing visualizing library in Python. It is built on NumPy arrays and designed to work with the broader SciPy stack and consists of several plots like line, bar, scatter, histogram, etc.

Matplot Lib in Machine Learning:

Data Visualization Is An Important Technique In Machine Learning To Understand The Data Through Various Plotting Techniques. However, It Helps Us To Get Insights About The Trends, Patterns, Outliers In Datasets Which Helps Us To Develop A Better Machine Learning Model With High Performance.
With High Volume Of Data It Is Difficult For Machine Learning Engineers To Understand The Structure, Pattern And Distribution Of Data, So Data Visualization Methods Are Employed To Get Statistical Inference About The Dataset.

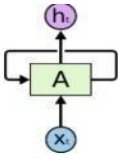
1.6 Pysimple GUI :

PySimple GUI is a Python package that enables Python programmers of all levels to create GUIs. You specify your GUI window using a "layout" which contains widgets (they're called "Elements" in PySimple (GUI). Your layout is used to create a window using one of the 4 supported frameworks to display and interact with your window. Supported frameworks include tkinter, Qt, WxPython, or Remi. The term "wrapper" is sometimes used for these kinds of packages.

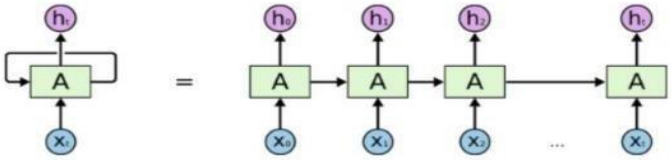
PySimple GUI is a python library that wraps tkinter, Qt (pyside2), wxPython and Remi (for browser support), allowing very fast and simple-to-learn GUI programming. PySimple GUI defaults to using tkinter, but the user can change to another supported GUI library by just changing one line.

1.7 Recurrent Neural Network:

RNN is recurrent in nature as it performs the same function for every input of data while the output of the current input depends on the past one computation. After producing the output, it is copied and sent back into the recurrent network. For making a decision, it considers the current input and the output that it has learned from the previous input. Unlike feed forward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. In other neural networks, all the inputs are independent of each other. But in RNN, all the inputs are related to each other.

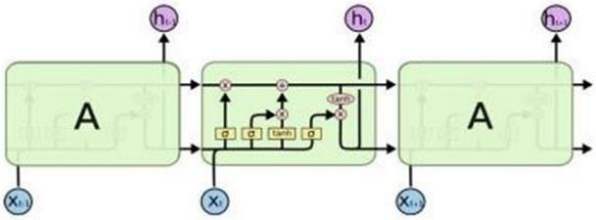


Recurrent Neural Networks have loops.

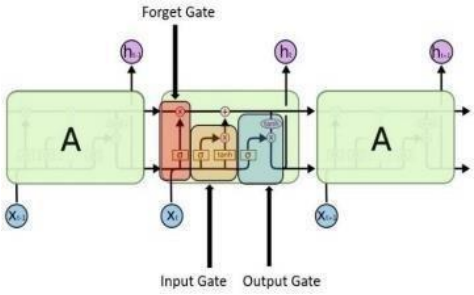


An unrolled recurrent neural network.

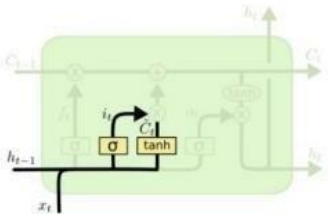
1.8 Long Short Term Memory(LSTM):



LSTM's have a Nature of Remembering information for a long periods of time is their Default behavior. look at the below figure that says Every LSTM module will have 3 gates named as Forget gate, Input gate, Output gate.

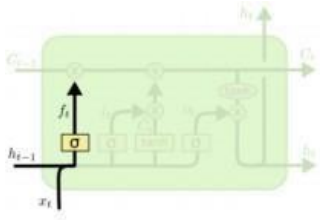


1) Input gate — discover which value from input should be used to modify the memory. Sigmoid function decides which values to let through 0,1. And the function gives weightage to the values which are passed deciding their level of importance ranging from-1 to1.



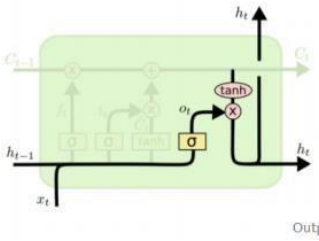
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

. **2)Forget gate** — discover what details to be discarded from the block. It is decided by the sigmoid function. it looks at the previous state(ht-1) and the content input (Xt) and outputs a number between 0(omit this) and 1(keep this) for each number in the cell state Ct−1.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

3)Output gate — the input and the memory of the block is used to decide the output. Sigmoid function decides which values to let through 0,1. and tanh function gives weightage to the values which are passed deciding their level of importance ranging from-1 to 1 and multiplied with output of Sigmoid



$$o_t = \sigma\left(W_o \left[h_{t-1}, x_t\right] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

REQUIREMENTS AND ANALYSIS

2.1 Software components:

- **Python:**Python offers concise and readable code. While complex algorithms and versatile workflows stand behind machine learning and AI, Python's simplicity allows developers to write reliable systems.

- **Numpy:** NumPy is a Python library used for working with arrays.

- **Pandas:** Pandas is an open-source library that is built on top of NumPy library. It is a Python package that offers various data structures and operations for manipulating numerical data and time series.

- **Pysimple GUI :** The model is Used to Create a Simple Graphical User Interface to our Project with this we add buttons for the Companies and from here we can select the company to predict its stock price.

- **Scikit Learning:** Scikit-learn is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.

- **Matplot Lib:** Matplotlib is easy to use and an amazing visualizing library in Python. It is built on NumPy arrays and designed to work with the broader SciPy stack and consists of several plots like line, bar, scatter, histogram, etc.

- **Tensorflow:** TensorFlow is an open-source framework developed by Google researchers to run machine learning, deep learning and other statistical and predictive analytics workloads. Like similar platforms, it's designed to streamline the process of developing and executing advanced analytics applications for users such as data scientists, statisticians and predictive modelers

- **VS code Application**

- **Windows OS 64-bit.**

2.2 Functional requirements

- The software should shall do pre-processing (like verifying for missing data values) on input for model training.
- The software shall use LSTM as main component of the software.
- It processes the given input data by producing the most possible outcomes of a CLOSING STOCK PRICE.

2.3 Non-Functional requirements

• Usability: It defines the user interface of the software in terms of simplicity of understanding the user interface of stock prediction software, for any kind of stock trader and other stakeholders in stock market.

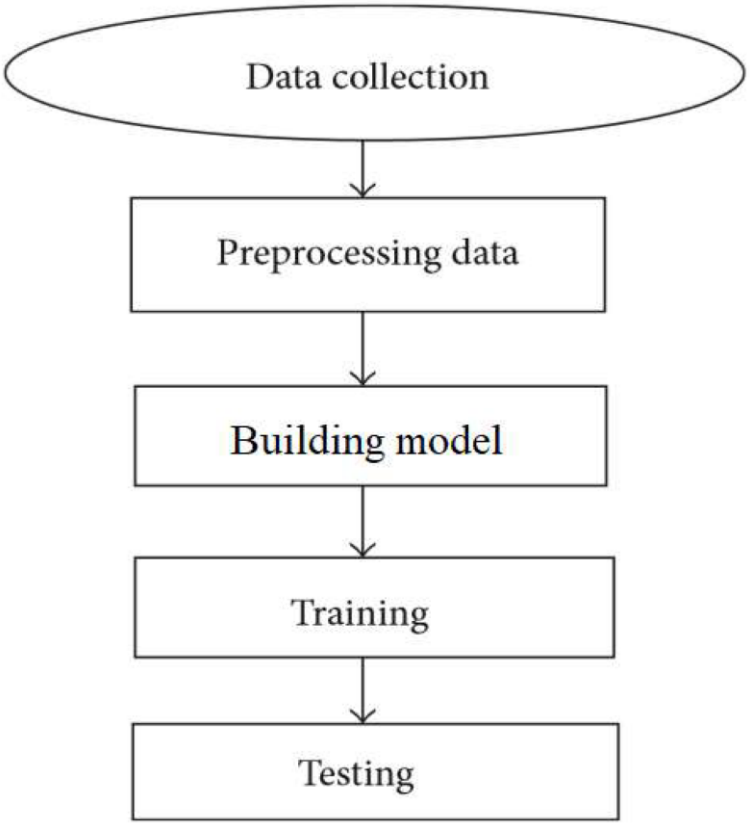
• Efficiency: maintaining the possible highest accuracy in the closing stock prices in shortest time with available data.

PROPOSED MODEL AND FLOW OF THE PROJECT

3.1 Proposed model

The Field of the proposed model is MACHINE LEARNING, DEEP LEARNING and this model is FUTURE STOCK PRICE PREDICTION USING RECURRENT NEURAL NETWORK, LSTM AND MACHINE LEARNING. The Stock Price prediction model can predict more accurate than other existing models and this project is also different in a way that we had created a Graphical User Interface (GUI) where we can select the Companies and predict the Stock price by taking last 60 days in training data and Predicting the one of future . In this Project We had created a graph which consists of the Predicted values and the Actual values. And also in the model there is a special feature where it can display the close values of the predicted next day.

3.2 Flow of the project



3.3 Advantages and disadvantages

3.3.1 Advantages :

- This Project can be used by general people to study stock prices without any pre requested knowledge.
- This can be used to predict stock prices more accurately than other models.

3.3.2 Disadvantages :

- This will fail in some exceptional cases, as stock market is extremely volatile as there are numerous factors affecting the stock values.

IMPLEMENTATION

4.1 Import all necessary libraries :

```
1   import numpy as np
2   import matplotlib.pyplot as plt
3   import pandas as pd
4   import pandas_datareader as web
5   import datetime as dt
6   #
7   import PySimpleGUI as sg
8   #
9   from sklearn.preprocessing import MinMaxScaler
10  from tensorflow.keras.models import Sequential
11  from tensorflow.keras.layers import Dense,Dropout,LSTM
12
```

We import all the necessary modules and write the import statements

4.2 GUI code : By using pysimple gui we create a simple graphical user interface to the code.

```
15
16  dic={"AMAZON":"AMZN","FLIPKART":"FPKT","RELIANCE":"RELIANCE","META":"META","MAXLINEAR":"MXL"}
17  #
18  sg.theme('DarkAmber')
19  layout=[[sg.Text('Select respective company button to predict it shares')],
20                [],
21                [sg.Button('Amazon',key='-A-'),sg.Button('Netflix',key='-N-'),sg.Button('apple',key='-AA-'),sg.Button('Meta',key='-Me-'),sg.Button('Maxlinear',key='-Mx-')],
22                [sg.Text('',key='-result-')],
23                [sg.Button('Exit',button_color=('white','black'),font=('Helvetica', 14))]
24
25                ]
26  window=sg.Window('miniprojet',layout,icon='Logonew.ico')
```

```
while True:
        event,values=window.read()
        if event is None or event=='Exit':
                break
        if event=='-A-':
                print('amazon')
                company="AMZN"
                #data=pd.read_csv('C:\\Users\\Likhith\\Desktop\\e3s2\\miniproject\\Amzn.csv')
                break
        if event=='-AA-':
                print('apple')
                company="AAPL"
                #data=pd.read_csv('C:\\Users\\Likhith\\Desktop\\e3s2\\miniproject\\Apl.csv')
                break
        if event=='-N-':
                print('Netflix')
                company='NFLX'
                #data=pd.read_csv('C:\\Users\\Likhith\\Desktop\\e3s2\\miniproject\\Ntflix.csv')
                break
        if event=='-Me-':
                print('Meta')
                company='META'
                #data=pd.read_csv('C:\\Users\\Likhith\\Desktop\\e3s2\\miniproject\\Meta.csv')
                break


         if event=='-Mx-':
                 print('Maxilinear')
                 company='MXL'
                 #data=pd.read_csv('C:\\Users\\Likhith\\Desktop\\e3s2\\miniproject\\Mxl.csv')
                 break

#event,values=window.read()
#
#print(dic)
print("choose the company's ticker from the above dictionary ")
#company=input("enter ticker symbol  :")
```

4.3 Loading of datasets : We used pandas reader to load the data from the “Yahoo” Financial API.

We loaded the data between two particular dates .We loaded the data from 2013 to 2022 . 9 years dataset of a selected company loaded.

```
start = dt.datetime(2013,1,1)
end   = dt.datetime(2022,1,1)
data = web.DataReader(company,'yahoo',start,end)
```

4.4 Preparation of data :

We prepared the data for the Neural Network by Min max scaler .We scaled the data in a range of 0 to 1.

```
scaler= MinMaxScaler(feature_range=(0,1))
scaled_data= scaler.fit_transform(data['Close'].values.reshape(-1,1))


prediction_days = 60

x_train=[]
y_train=[]

for x in range(prediction_days,len(scaled_data)):
    x_train.append(scaled_data[x-prediction_days:x,0])
    y_train.append(scaled_data[x,0])

x_train,y_train= np.array(x_train),np.array(y_train)

#x_train is going to work with neural network. so we have to reshaped it

x_train=np.reshape(x_train,(x_train.shape[0],x_train.shape[1],1))

```

4.5 Model Building:

```
model=Sequential()

#we always one lstm layers and one dropout layers and one lstm layer and one dropout layer
#in the end we add dense layer,this in one unit
#that one unit is going to predict the stock price

#if we add more layers and more units we have to train the model for longer time
#if we use more layers it may overfit of sophistication

model.add(LSTM(units=50,return_sequences=True,input_shape=(x_train.shape[1],1)))
model.add(Dropout(0.2))
model.add(LSTM(units=50,return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(units=50))
model.add(Dropout(0.2))
model.add(Dense(units=1)) #prediction of the next closing value

model.compile(optimizer='adam',loss='mean_squared_error')
model.fit(x_train,y_train,epochs=100,batch_size=32)

```

4.6 Testing the Model :

```
test_start = dt.datetime(2022,1,1)

test_end = dt.datetime.now()

test_data = web.DataReader(company,'yahoo',test_start,test_end)


actual_prices= test_data['Close'].values


#total_data set which combines  training dataset and testing dataset


total_dataset = pd.concat((data['Close'],test_data['Close']),axis=0)

#we need to give the input to the model so it can predict the next price

model_inputs= total_dataset[len(total_dataset)- len(test_data)- prediction_days:].values
model_inputs= model_inputs.reshape(-1,1)
model_inputs= scaler.transform(model_inputs)
```

```
148  x_test =[]
149
150  for x in range(prediction_days,len(model_inputs)):
151          x_test.append(model_inputs[x-prediction_days:x,0])
152
153  x_test  = np.array(x_test)
154  x_test =  np.reshape(x_test,(x_test.shape[0],x_test.shape[1],1))
155
156
157  predicted_prices = model.predict(x_test)
158
159  #predicted prices are scaled,so we need to reverse scaled them. so we can get back to the actucal prices.
160
161  predicted_prices = scaler.inverse_transform(predicted_prices)
162
```

4.7 Visualising the predictions :

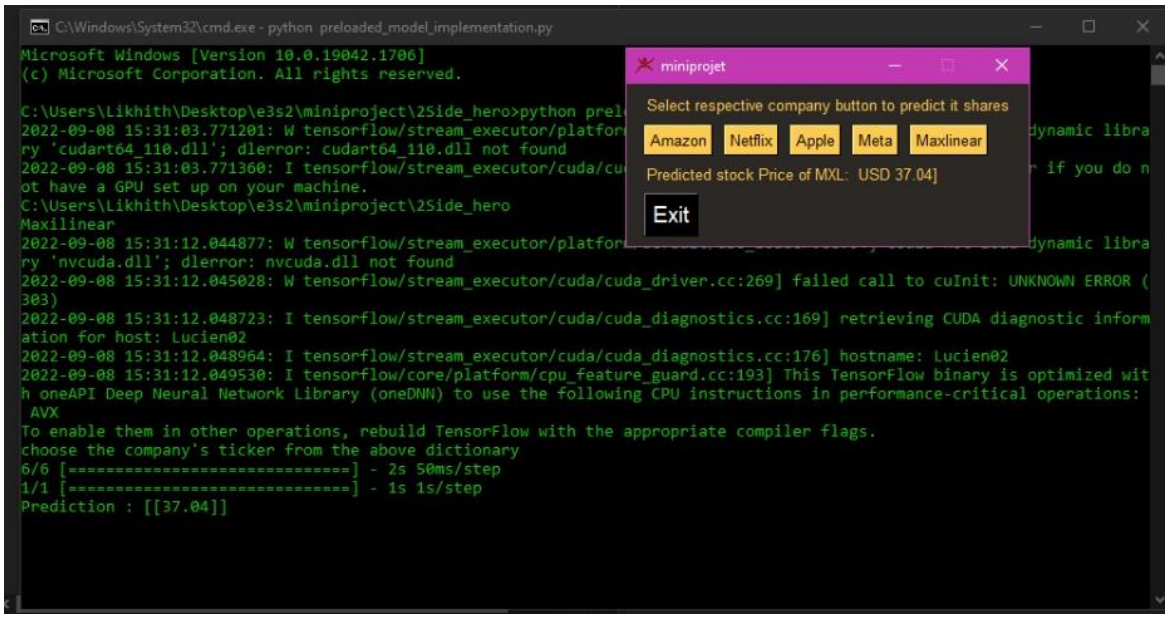
By using the “Matplot lib” library we visualize the Predicted and Actual Prices of the company.

```
166
167  plt.plot(actual_prices,color='black',label=f"Actual {company} Price")
168  plt.plot(predicted_prices,color='green',label=f"Prediction {company} Price")
169  plt.title(f"{company} Share Price")
170  plt.xlabel('Time')
171  plt.ylabel(f'{company} Share price')
172  plt.legend()
173  plt.show()
174
```

4.8 Predicting next day stock price :

```
#Predict the next day

real_data=[model_inputs[len (model_inputs)+1 - prediction_days : len(model_inputs+1),0]]
real_data=np.array(real_data)
real_data=np.reshape(real_data,(real_data.shape[0],real_data.shape[1],1))

prediction = model.predict(real_data)
prediction = scaler.inverse_transform(prediction)
print(f"Prediction : {prediction}")
window['-result-'].update(f'Predicted stock Price of {company}: {" Rs. "+str(prediction[0])[1:6]}')
while True:
        event,values=window.read()

        if event=='Exit'or event==sg.WIN_CLOSED:
                break
#saving the model
#model.save('Mxlmodel')

window.close()
```

OUTPUT



CONCLUSION

- The Companies future stock price has been successfully predicted.
- The GUI is working well and user friendly.
- The system works well even if the models are saved and reloaded.

REFERENCES

- **https://www.alpharithms.com/predicting-stock-prices-with-linear-regression-214618/**
- **https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM**
- **https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html**
- **https://numpy.org/learn/**
- **https://pandas.pydata.org/docs/user_guide/reshaping.html**
- **https://pypi.org/project/PySimpleGUI/**
- **https://matplotlib.org/stable/api/index**