A MINI-PROJECT REPORT ON

STOCK MARKET PREDICTION

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING
IN
INFORMATION TECHNOLOGY
BY

NEIL DESHPANDE (T190058554)
SUDEEP MANGALVEDHEKAR (T190058631)
PRANAV BHAGWAT (T190058681)
VISHWAJIT SHELKE (T190058717)

UNDER THE GUIDANCE OF
PROF. DEEPALI D. LONDHE

DEPARTMENT OF INFORMATION TECHNOLOGY
PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE.
SR. NO 27, PUNE-SATARA ROAD, DHANKAWADI, PUNE - 411043.

Academic Year 2021-22

CERTIFICATE

This is to certify that the project report entitled

STOCK MARKET PREDICTION

Submitted by

NEIL DESHPANDE (T190058554)
SUDEEP MANGALVEDHEKAR (T190058631)
PRANAV BHAGWAT (T190058681)
VISHWAJIT SHELKE (T190058717)

is a bonafide work carried out by them under the supervision of Prof. Deepali D. Londhe and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University for the award of the Degree of Bachelor of Engineering (Information Technology)

This project report has not been earlier submitted to any other Institute or University for the award of any degree or diploma.

Prof. Deepali D. Londhe
Internal Guide
Department of Information Technology

Dr. A. Ghotkar
Head of Department
Department of Information Technology

Prof. Deepali D. Londhe
Internal Guide
Date:

Place:
Date:

ACKNOWLEDGEMENT

It is our pleasure to present this mini-project report on the problem statement of “Stock Market Prediction” in the Big Data and Deep Learning domain. Firstly, we would like to thank our Principal R. Sreemathy, the Head of I.T. Department Prof. A. Ghotkar and our internal guide Prof. Deepali D. Londhe for guiding us with this project.

We also thank our friends and family who have helped us discuss and further explore our ideas.

NEIL DESHPANDE SUDEEP MANGALVEDHEKAR

PRANAV BHAGWAT VISHWAJIT SHELKE

ABSTRACT

This mini-project has been performed on the topic of ”Stock Market Prediction” using APIs (Application Programming Interface) to fetch real-time data, Deep Learning LSTM (Long Short Term Memory Network) model to predict stock prices and a web page made using Streamlit to visualize the data and communicate the results. We have chosen this topic to work with time series data as it can be applied to other fields such as patient history in healthcare, employee experience in corporate sector and more. This mini-project successfully predicts the stock prices for Apple, Google and Microsoft using data from the NASDAQ stock exchange. It can further be extended to include more companies data and improve the model's performance in both accuracy and compute time.

Contents

List of Figures

List of Tables

1 INTRODUCTION

We chose ”Stock Market Prediction” as our mini-project topic since it encompasses a plethora of useful techniques that any developer in the Big Data and Deep Learning fields must know. These are:

1. Working with real-time data
   By using the API (Application Programming Interface) of a stock market index, we were able to access real world data in real time.

2. Working with Deep Learning models
   By using Long Short Term Memory Models, we were able to predict the future i.e. possible prices of a particular stock.

3. Working with dashboards
   By using a web application to communicate our results, we learned how to use dashboards to communicate our working and findings.

A stock market predictor deals with time-series data i.e. data that is intrinsically tied to the date/time on which it was created. Most of the real world data is time-series in nature, like the records of a patient at a hospital, a student's marks and an employee's records.
Knowing how to deal with such data is important and this project can be extended to different domains by changing the dataset and a few data cleaning steps which makes it incredibly useful to perform.

2 BACKGROUND AND LITERATURE REVIEW

For this particular topic, we had to look into a variety of finance related concepts so as to fully understand the data and its meaning such as:

1. Stock: In finance, stock consists of all of the shares into which ownership of a corporation or company is divided. A single share of the stock means fractional ownership of the corporation in proportion to the total number of shares.

2. Stock Index: In finance, a stock index, or stock market index, is an index that measures a stock market, or a subset of the stock market, that helps investors compare current stock price levels with past prices to calculate market performance.

3. Open High Low Close (OHLC): In stock trading, the high and low refer to the maximum and minimum prices in a given time period. Open and close are the prices at which a stock began and ended trading in the same period.

4. Volume: For stock trading, volume is the total amount of trading activity.

Apart from the financial aspect, we also had to learn how to work with time-series data and research which technique would be best for predicting the future prices. The techniques were:

1. Linear Regression:
   The most basic machine learning algorithm that can be implemented on this data is linear regression. The linear regression model returns an equation that determines the relationship between the independent variables and the dependent variable.

   The equation for linear regression can be written as:

   $y = \theta_1 x_1 + \theta_2 x_2 + ...\theta_n x_n$

   Here, $x_1, x_2, ...x_n$ represent the independent variables, the coefficients $\theta_1, \theta_2, ...\theta_n$ represent the weights and y is the dependent variable.

2. ARIMA (Auto-Regressive Integrated Moving Averages): [3]
   ARIMA models work on the following assumptions –

   a.) The data series is stationary, which means that the mean and variance should not vary with time. A series can be made stationary by using log transformation or differencing the series.

   b.) The data provided as input must be a univariate series, since ARIMA uses the past values to predict the future values.

   ARIMA has three components – AR (autoregressive term), I (differencing term) and MA (moving average term).

   a.) AR term refers to the past values used for forecasting the next value. The AR term is defined by the parameter ‘p’ in arima. The value of ‘p’ is determined using the PACF (Partial Autocorrelation) plot.

b.) MA term is used to defines number of past forecast errors used to predict the future values. The parameter ‘q’ in ARIMA represents the MA term. ACF(Auto Correlation) plot is used to identify the correct ‘q’ value.

c.) Order of differencing specifies the number of times the differencing operation is performed on series to make it stationary. Test like ADF (Augmented Dickey Fuller) and KPSS (Kwiatkowski-Phillips-Schmidt-Shin) can be used to determine whether the series is stationary and help in identifying the d value.

3. Deep Learning:
Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. Deep learning has aided image classification, language translation, speech recognition. It can be used to solve any pattern recognition problem and without human intervention.
Deep learning systems require large amounts of data to return accurate results; accordingly, information is fed as huge data sets. When processing the data, artificial neural networks are able to classify data with the answers received from a series of binary true or false questions involving highly complex mathematical calculations.

a.) Long Short Term Memory networks (LSTMs): [2]
These are a special kind of Recurrent Neural Network (RNN) which overcome RNN’s inability to learn long-term dependencies. LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series.
It uses a Forget Gate to selectively forget previous inputs, Input Gate to accept input and Output Gate to produce output.

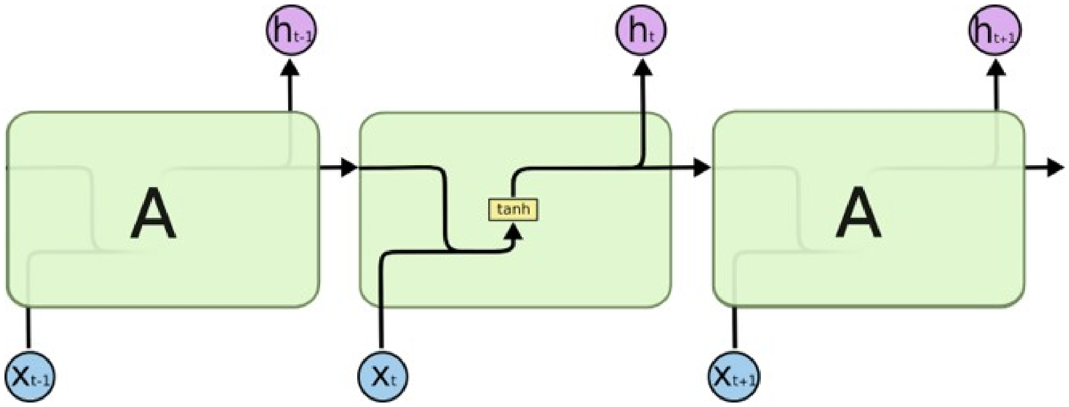


Figure 1: A single RNN module [2]

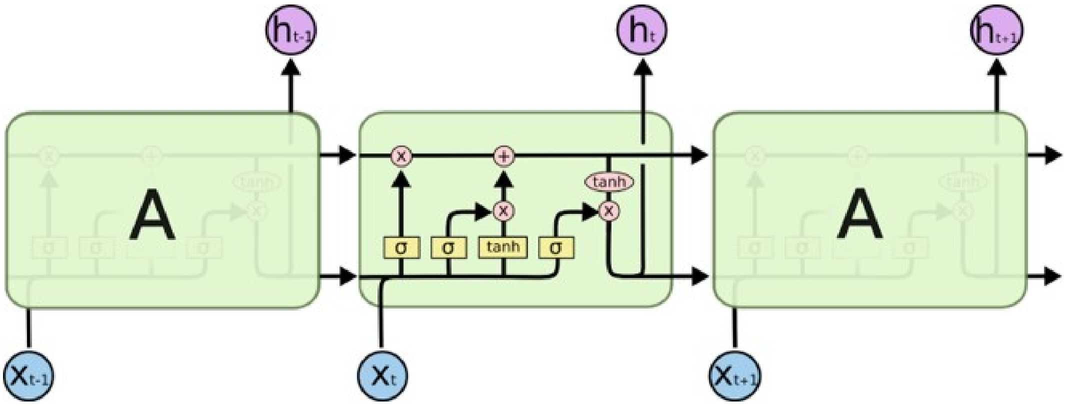


Figure 2: A single LSTM module [2]

3 REQUIREMENT SPECIFICATION AND ANALYSIS

To perform Stock Market Prediction, the following components are needed:

1. A dataset for the relevant company's stock
2. A machine learning or deep learning model which will be used to predict future stock prices
3. A user interface to display the results

3.1 Dataset

Stock markets are very particular in many ways. The markets open and close at a fixed time each day and are closed on the weekend. Barring any unique occasion, the data of a company's stocks must be available in the stock exchange to be used by the model.
This data must be in a time series format as the performance of a stock is heavily dependent on the day since many factors like the politics of the country, the performance of rival companies and more affect a company's stock price aside from its own performance.
To retrieve this data, there must be some API (Application Programming Interface) of a stock exchange index available and it should be capable of providing real-time data.

3.2 Prediction Model

Forecasting is the process of predicting the future using current and previous data. The major challenge is understanding the patterns in the sequence of data and then using this pattern to analyse the future. If we were to hand-code the patterns, it would be tedious and changes for the next data. Deep Learning has proven to be better in understanding the patterns in both structured and unstructured data.
To understand the patterns in a long sequence of data, we need networks to analyse patterns across time. Recurrent Neural Networks (RNNs) are the ones usually used for learning such data. However, due to the limitations of RNNs with long-term dependencies, we must use their improved variation of Long Short Term Memory Networks (LSTMs). They are capable of understanding long and short term dependencies or temporal differences.

3.3 User Interface

To communicate the results of the model, a user interface is essential. This user interface must have labelled graphs and other content to easily explain to the viewer what is being done. The graph itself must also be interactive so as to provide more granular detail as opposed to just visually showing trends in the data.

4 DESIGN AND IMPLEMENTATION

4.1 Alpha Vantage API

The Alpha Vantage API [1] is a method to obtain historical and real-time data for several markets. We used it for the NASDAQ API. You can access the data directly in Python or any other programming language of your choosing. From there, you can manipulate the data or store it for later use. It offers historical and real-time data for stocks, forex, and cryptocurrencies. Several time frames are available ranging from 1-minute bars up to monthly.

We used Alpha Vantage [5] by first creating an account and getting an API key which we used to authenticate our requests. We then import the appropriate part of the library and instantiate the class within it. For example:

```
from alpha_vantage.timeseries import TimeSeries
app = TimeSeries()
```

We then use the API URLs for the endpoints defined in the documentation as follows:

```
import requests

# replace the "demo" apikey below with your own key
# from https://www.alphavantage.co/support/#api-key
url = 'https://www.alphavantage.co/query?function=
    TIME_SERIES_DAILY&symbol=IBM&apikey=demo'
r = requests.get(url)
data = r.json() # or r.content()

print(data)
```

4.2 Exploratory Data Analysis

After retrieving the data, we receive a requests object which is then saved as a csv (Comma Separated Values) and parsed as a Pandas DataFrame. The first 5 rows of the DataFrame can be seen as:

index	timestamp	open	high	low	close	volume
0	2022-04-29	2351.56	2379.20	2293.8800	2299.33	1684655
1	2022-04-28	2342.30	2408.77	2302.8778	2388.23	1839547
2	2022-04-27	2287.46	2350.00	2262.4850	2300.41	3111906
3	2022-04-26	2455.00	2455.00	2383.2370	2390.12	2284104
4	2022-04-25	2388.59	2465.56	2375.3850	2465.00	1726090

Table 1: df.head() for the data retrieved for Google from TIME_SERIES_DAILY

As shown above, the values retrieved are:

1. timestamp: This is the date associated with the stock's details
2. open: This is the opening price of that stock for its timestamp

3. high: This is the highest price of that stock for its timestamp

4. low: This is the lowest price of that stock for its timestamp

5. close: This is the closing (final) price of that stock for its timestamp

6. volume: This is the amount of that stock traded for its timestamp

4.3 Model Training And Testing

We have created a LSTM model using tensorflow and keras in Python.

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

model = Sequential()
model.add(
    LSTM(10,
        activation='relu',
        input_shape=(look_back,1))
)
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')

num_epochs = 25
model.fit_generator(train_generator, epochs=num_epochs, verbose=1)
```

It is a sequential model which uses one LSTM layer with 10 nodes and Rectified Linear Unit (relu) as its activation function. It is connected to a Dense layer of 1 node i.e the output layer. We have used the Adam algorithm for optimization and Mean Squared Error as the loss function (see section 5.1) for 25 epochs. Instead of using model.fit(), we use model.fit_generator() because we have created a data generator.

4.4 Streamlit

Streamlit [4] is an open source app framework in Python language. It helps us create web apps for data science and machine learning in a short time. It is compatible with major Python libraries such as scikit-learn, Keras, PyTorch, SymPy(latex), NumPy, pandas, Matplotlib etc.
With Streamlit, no callbacks are needed since widgets are treated as variables. Data caching simplifies and speeds up computation pipelines. Streamlit watches for changes on updates of the linked Git repository and the application will be deployed automatically in the shared link.
We have used Streamlit to create a web page that allows users to:

1. Select a company from list (AAPL = Apple, GOOG = Google and so on)

2. The stock information for that company in the NASDAQ exchange received by the API is displayed as an interactive line plot.

3. Either for close price, open price, high price or low price a graph showing its prediction is displayed as an interactive line plot.

5 OPTIMIZATION AND EVALUATION

5.1 Error Calculation and Model Optimization

```
model.compile(optimizer='adam', loss='mse')
```

We have used Mean Squared Error (MSE) as our loss function for the Neural Network. The purpose of loss functions is to compute the quantity that a model should seek to minimize during training. MSE was used as it is defined for regression i.e. prediction problems in the Keras API documentation.

We have used Adaptive Moment Estimation (adam) as our optimization algorithm. It is an algorithm for optimization technique for gradient descent. The method is really efficient when working with large problem involving a lot of data or parameters. It requires less memory and is efficient. Intuitively, it is a combination of the 'gradient descent with momentum' algorithm and the Root Mean Square Propagation (RMSP) algorithm.

5.2 Time Saving Techniques

To optimize performance by reducing compute time, we performed the following:

1. After the initial training, the LSTM model used is serialized and deserialized using pickle library separately for each stock. Thus, the model is effectively cached when not in use and can be loaded whenever its associated stock is chosen by the user.

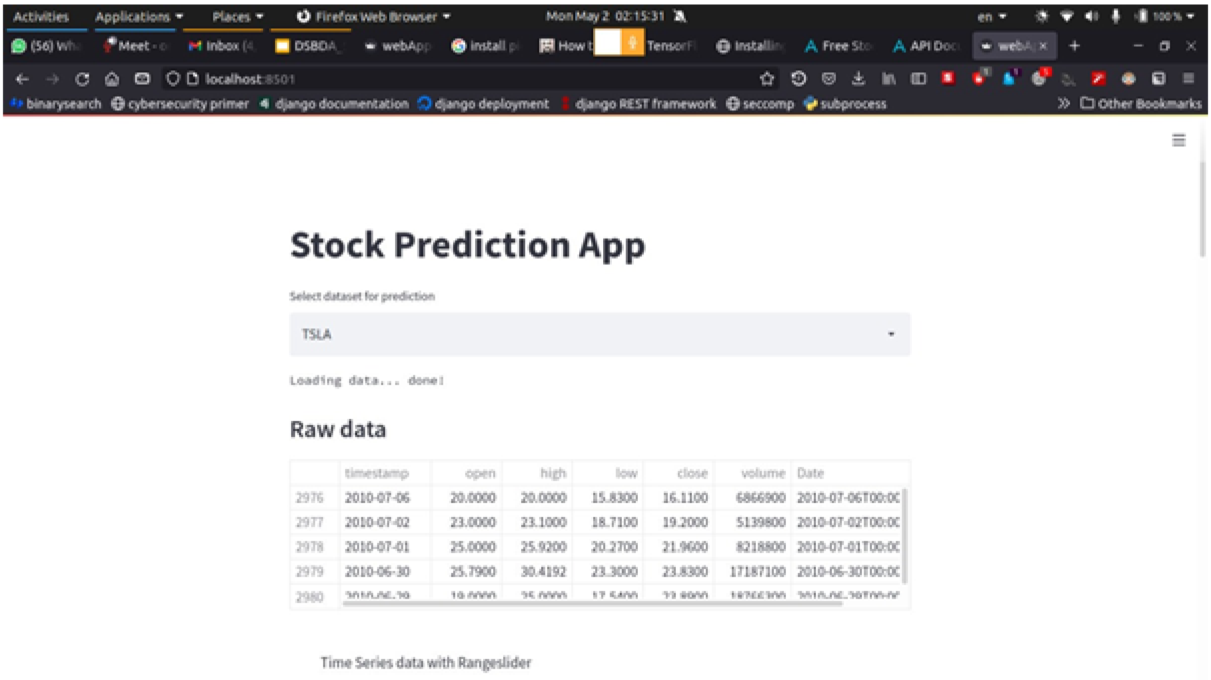
2. When the user checks prediction for stock A and then switches to stock B, as the separate models are cached the model for A is not retrained for B.

3. When the user checks prediction for stock A, then B and then A again, the model for A picks up where it left off and doesn't start from scratch.

4. The Streamlit local cache is used to save the dataset of the current stock selected by the user for the model to use.

6 RESULT

The result of our project is a web page which allows users to select which company they want see data about and its associated graphs for the datasets and predictions.



Stock Prediction App

Select dataset for prediction

TSLA

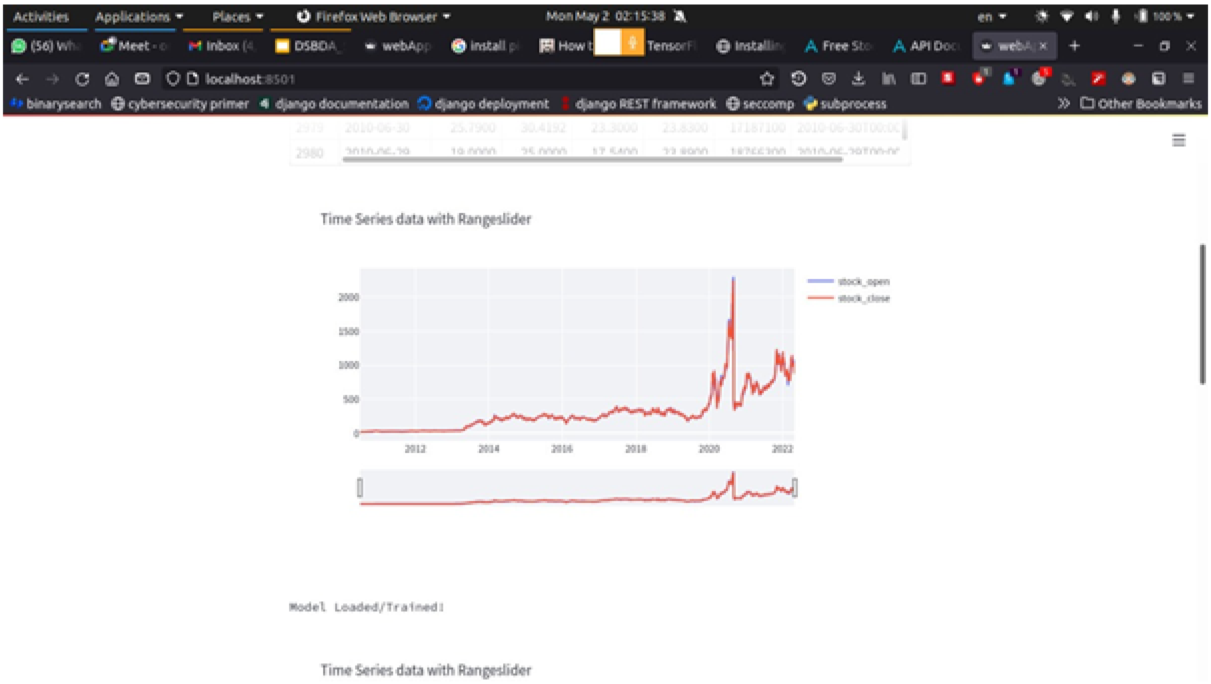
```
Loading data... done!
```

Raw data

	timestamp	open	high	low	close	volume	Date
2976	2010-07-06	20.0000	20.0000	15.8300	16.1100	6866900	2010-07-06T00:0C
2977	2010-07-02	23.0000	23.1000	18.7100	19.2000	5139800	2010-07-02T00:0C
2978	2010-07-01	25.0000	25.9200	20.2700	21.9600	8218800	2010-07-01T00:0C
2979	2010-06-30	25.7900	30.4192	23.3000	23.8300	17187100	2010-06-30T00:0C
2980	2010-06-29	19.0000	25.0000	17.5400	23.8900	18766300	2010-06-29T00:0(

Time Series data with Rangeslider

Figure 3: Company Stock Select Menu and Raw Data

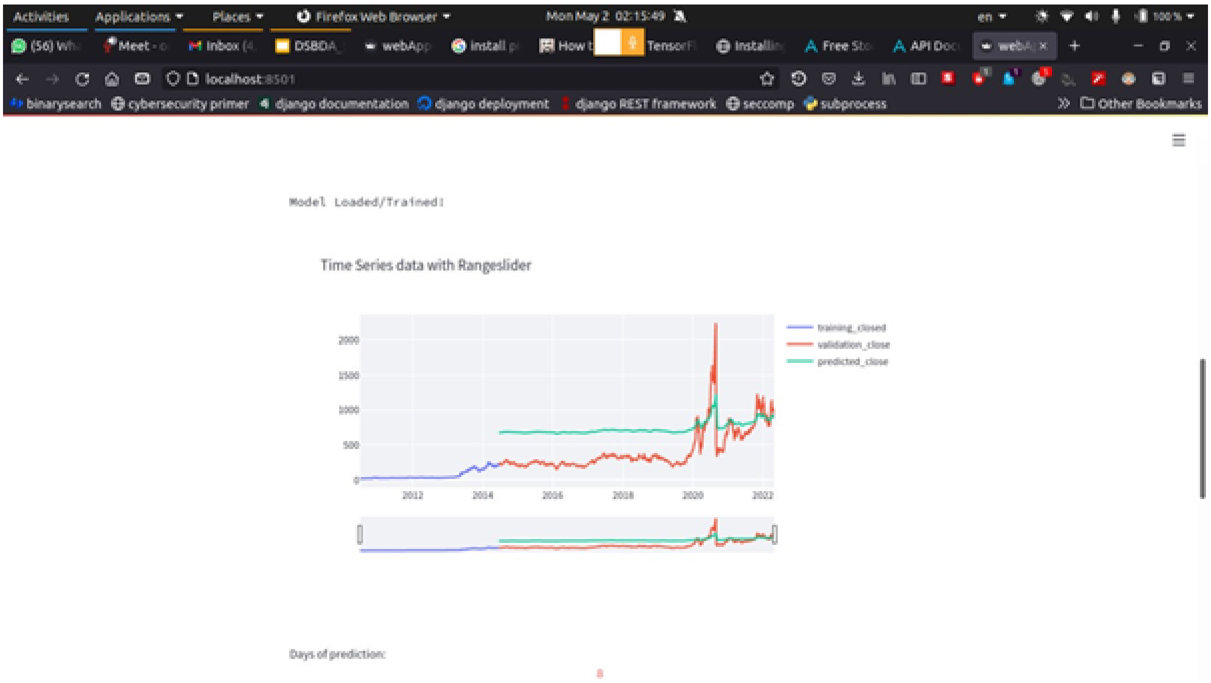


Time Series data with Rangeslider

```
Model Loaded/Trained!
```

Time Series data with Rangeslider

Figure 4: Line Plot of Raw Time Series Data for Close Price

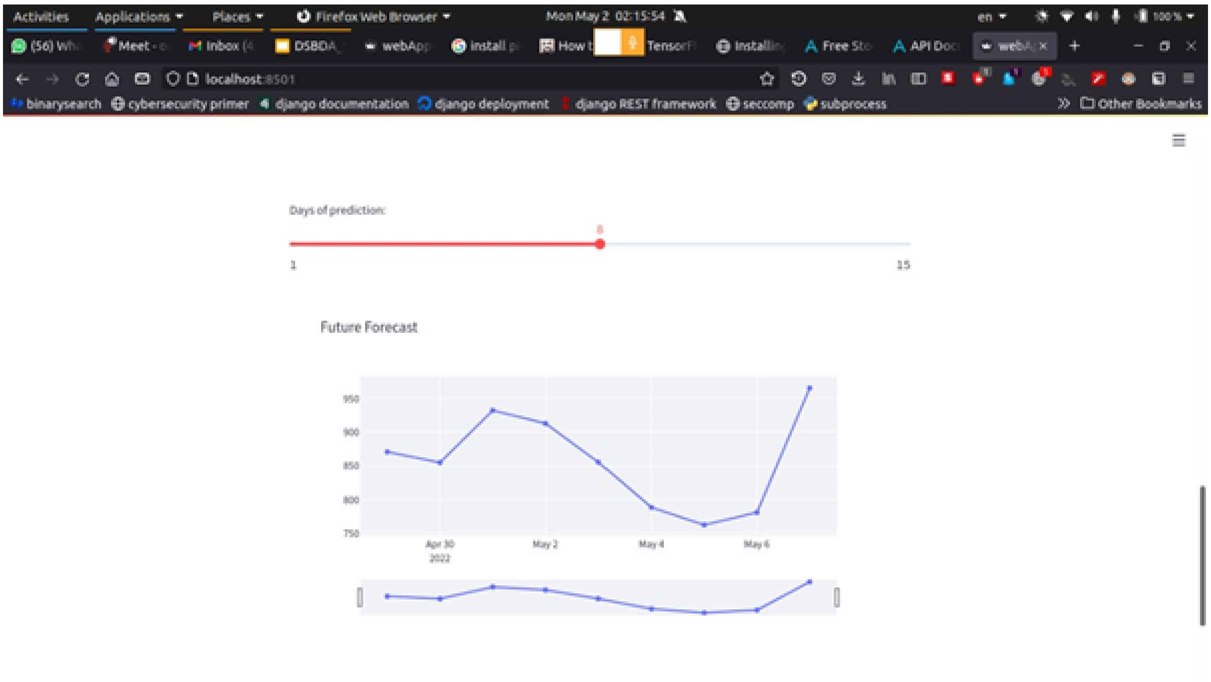


```
Model Loaded/Trained!
```

Time Series data with Rangeslider

Days of prediction:

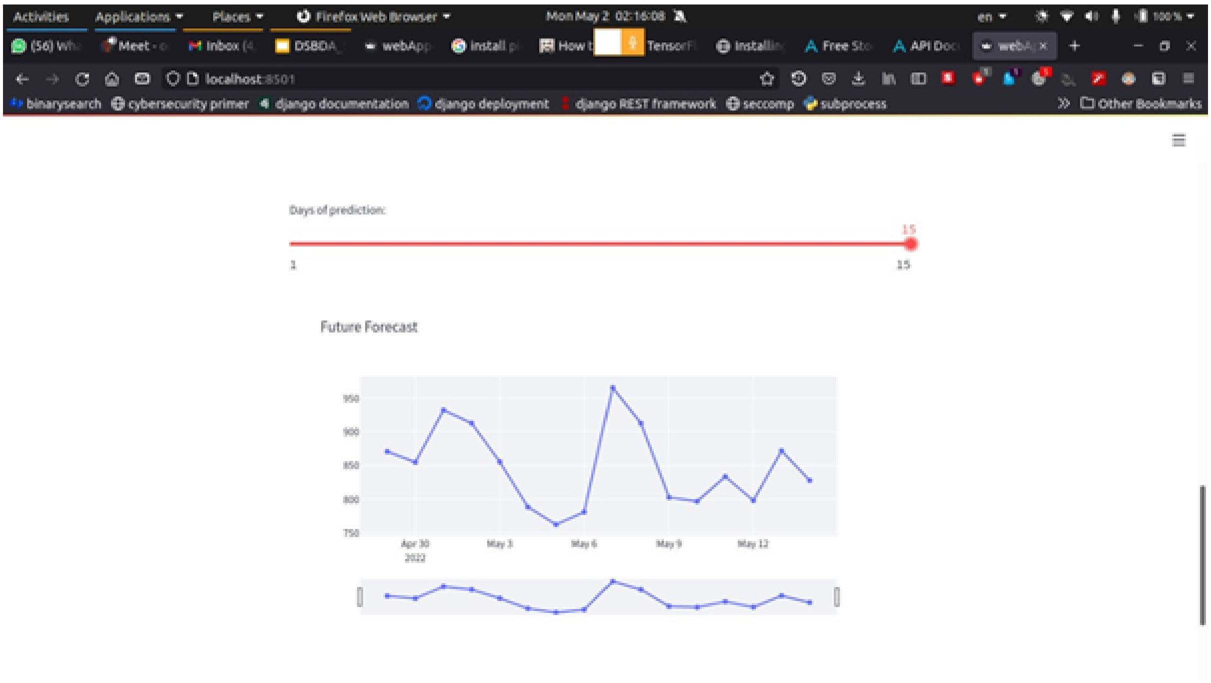
Figure 5: Line Plot of Training, Validation and Predicted Close Price



Days of prediction: 8

Future Forecast

Figure 6: Modifying days of prediction using slider (1)



Days of prediction: 15

Future Forecast

Figure 7: Modifying days of prediction using slider (2)

7 CONCLUSIONS AND FUTURE WORK

Thus, in this mini-project, we have successfully created a stock market predictor using an API to retrieve data for a company's stock in the NASDAQ stock exchange. We have used a LSTM based Deep Learning model that has been trained on the received time-series data and is capable of predicting the next prices. Finally, we have communicated our results through interactive data visualizations on a locally hosted web page to make it a well-rounded application. Regarding future scope, it is possible to extend the number of companies' stocks being considered and provide a separate model for each. Furthermore, multiple stock exchanges can be considered to cross-check predictions to increase accuracy.

References

[1] Jignesh Davda. Alpha vantage introduction guide - algotrading101 blog, Jan 2021.

[2] Christopher Olah. Understanding lstm networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs/. Accessed: 2021-10-30.

[3] Aishwarya Singh. A gentle introduction to handling a non-stationary time series in python. https://www.analyticsvidhya.com/blog/2018/09/non-stationary-time-series-python/, 2018.

[4] Streamlit. Streamlit. https://www.streamlit.io/.

[5] Alpha Vantage. Free stock apis in json excel — alpha vantage. https://www.alphavantage.co/.