

Summer Project On
Stock Market Prediction using Twitter
Sentimental Analysis

By

Sanket Nitnaware (2021510039)
Kevin Rego (2021510056)

Under the guidance of
Internal Supervisor

Prof. Sakina Salmani



Department of Master Of Computer Application
Sardar Patel Institute of Technology
Autonomous Institute Affiliated to Mumbai University
2022-23

CERTIFICATE OF APPROVAL

This is to certify that the following students

Sanket Nitnaware (2021510039)
Kevin Rego (2021510056)

Have satisfactorily carried out work on the project
entitled

**“Stock Market Prediction using
Twitter Sentimental Analysis”**

Towards the fulfillment of project, as laid down
by
Sardar Patel Institute of Technology
during year
2022-23.

Project Guide:
Prof. Sakina Salmani

PROJECT APPROVAL CERTIFICATE

This is to certify that the following students

Sanket Nitnaware (2021510039)
Kevin Rego (2021510056)

Have successfully completed the Project report on

**“Stock Market Prediction using Twitter
Sentimental Analysis”,**

which is found to be satisfactory and is approved

at

**SARDAR PATEL INSTITUTE OF TECHNOLOGY,
ANDHERI (W), MUMBAI**

INTERNAL EXAMINER

EXTERNAL EXAMINER

HEAD OF DEPARTMENT

PRINCIPAL

Contents

Abstract	i
Objectives	i
List Of Figures	ii
List Of Tables	ii
1 Introduction	1
1.1 Problem Definition	1
1.2 Objectives and Scope	1
1.2.1 Objectives	1
1.2.2 Scope	1
1.3 Existing System	2
1.4 Proposed System	2
1.5 System Requirements	3
2 Software Requirement Specification (SRS) and Design	4
2.1 Purpose	4
2.2 Definition	4
2.3 Overall Description	4
2.3.1 Product Functions	4
2.3.2 User Characteristics	4
3 Project Analysis and Design	5
3.1 Methodologies Adapted	5
3.1.1 Work Breakdown Structure	6
3.1.2 PERT Chart	7
3.1.3 Gantt Chart	8
4 Project Implementation and Testing	9
4.1 Home - Landing View	9
4.2 Select Box	10
4.3 Plotting	11
4.4 Predictions	12
4.5 Code 1	13
4.6 Code 2	13
4.7 Code 3	14
4.8 Code 5	14
4.9 Code 6	15
5 Test Cases	16
6 Limitations	17

7	Future Enhancements	17
8	User Manual	18
9	Bibliography	19
9.1	Web References	19

Abstract

This is a Machine Learning project that offers stock investment recommendations based on Machine Learning predictions from last year's values of any market symbol and also based on Twitter sentiment analysis from retrieved tweets containing the symbol as text inside of it.

The User can view recommendations of different stock based on the past 3 years data of desired company.

The app will provide Stock investors with the ability to predict stock in realtime with sentimental analysis of tweets about the company.

Objectives

The Web Based Application "Stock Recommendation App" is used

- To provide Investors a user friendly platform for getting recommendations.
- To provide Plotting of the prediction graph.
- To provide a facility to view different companies stock predictions.

List of Figures

3.1.1	Diagrammatic Representation of Waterfall Model	5
3.1.2	Work Breakdown Structure	6
3.1.3	PERT Chart	7
3.1.4	Gantt Chart	8
4.1.1	Home View	9
4.2.1	Stocks	10
4.3.1	Graph View	11
4.4.1	Stock Predictions	12

List of Tables

1.5.1	Hardware Requirements on Server Side	3
1.5.2	Hardware Requirements on Client Side	3
1.5.3	Software Requirements on Server Side	3
1.5.3	Software Requirements on Client Side	3
6.1	Test Case - Login and Register	16

1 Introduction

1.1 Problem Definition

Stock exchange is a subject that is highly affected by economic, social, and political factors. There are several factors e.g. external factors or internal factors which can affect and move the stock market. Stock prices rise and fall every second due to variations in supply and demand. Various Data mining techniques are frequently involved to solve this problem. But technique using machine learning will give more accurate, precise and simple way to solve such issues related to stock and market prices. “Stock Price Prediction Using Twitter Sentiment Analysis” a method for predicting stock prices is developed using news articles. The changes in stock prices of a company, the rises and falls, are correlated with the public opinions being expressed in tweets about that company. Understanding author’s opinion from a piece of text is the objective of sentiment analysis. Positive news and tweets in social media about a company would definitely encourage people to invest in the stocks of that company and as a result the stock price of that company would increase. A prediction model for finding and analysing correlation between contents of tweets and stock prices and then making predictions for future prices can be developed by using machine learning

1.2 Objectives and Scope

1.2.1 Objectives

The Web based application “Stock Recommendation App” is

- To provide Investors a user friendly platform for getting recommendations.
- To provide Plotting of the prediction graph.
- To provide a facility to view different companies stock predictions.

1.2.2 Scope

The user can select the desired company of which they want to predict the stock from the select menu provided.

Our System is being made for reducing the information loss and smoothening the communication between Training and Placement Co-ordinator and students so that they both have all the required information in their hand.

1.3 Existing System

Currently system which use web scraping are there

Some of the disadvantages of existing system are as follows :

- Accuracy
Such systems are not accurate as they are heavily dependent upon the predictions made by others.

1.4 Proposed System

The User is the Investor who wants to invest in a particular company.

This application will show the predictions of various companies along with their graphs and plottings. So that the User can select correct company for investing.

Some of the advantages of our system are as follows :

- Predicitons
This application uses machine learning in order to make the predictions.
- Accuracy
A sentimental analysis of the tweets related about the company is done to combine it with the predictions made by the Machine learning model. This helps to increase the accuracy of the predicitons.

1.5 System Requirements

- Hardware Requirements on Server Side

Table 1.5.1: Hardware Requirements on Server Side

Processor	Dual Core Processor or Above
RAM	Minimum 4 GB RAM
Storage	Minimum 10 GB Hard Disk Space for smooth run

- Hardware Requirements on Client Side

Table 1.5.2: Hardware Requirements on Client Side

Device	Android Device with Touch Screen minimum 5" inch Display
Processor	Dual Core Processor or Above
RAM	Minimum 2 GB RAM
Storage	Minimum 250 MB Storage Space

- Software Requirements on Server Side

Table 1.5.3: Software Requirements on Server Side

Operating System	OS Independent
Database	Firestore

- Software Requirements on Client Side

Table 1.5.3: Software Requirements on Client Side

Any Web Browser	Chrome, Firefox etc.
Server	Not Required

2 Software Requirement Specification (SRS) and Design

2.1 Purpose

The purpose of our project is to develop a web application that can help user (investor) to see the predictions of current stock trends.

This lets the investors to make the decisions much more accurately using the prediction charts provided.

2.2 Definition

To build a Stock Prediction App so that investors can predict the stocks easily.

2.3 Overall Description

2.3.1 Product Functions

The product function includes:

1. To give the user options of the company.
2. To download the data of the company.
3. To perform sentimental analysis on the company.
4. To provide correct predictions to the user.

2.3.2 User Characteristics

There is only one user:

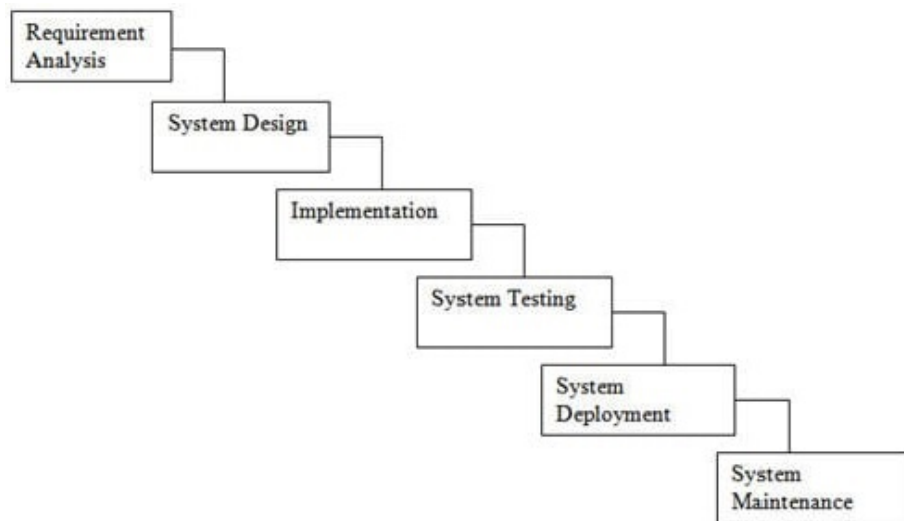
- Investor: He/She can select the company to view the stock trends and predictions.

3 Project Analysis and Design

3.1 Methodologies Adapted

In Waterfall model, very less customer interaction is involved during the development of the product. Once the product is ready then only it can be demonstrated to the end users.

Once the product is developed and if any failure occurs then the cost of such issues is very high, because we need to update everything from document till the logic.



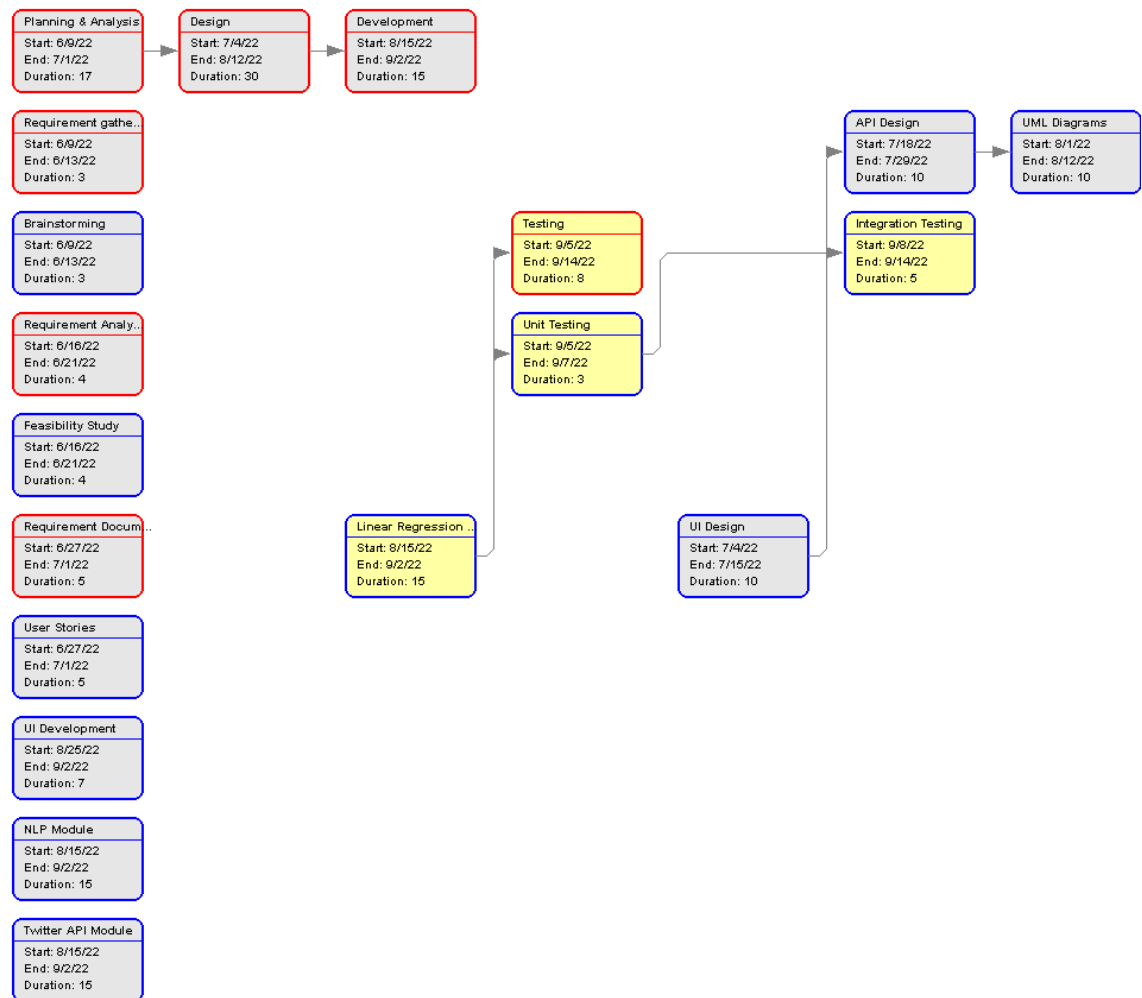
3.1.1: Diagrammatic Representation of Waterfall Model

3.1.1 Work Breakdown Structure

Name	Begin d...	End date	Resources
▼ Planning & Analysis	6/9/22	7/1/22	
▼ Requirement gathering	6/9/22	6/13/22	
Brainstorming	6/9/22	6/13/22	
▼ Requirement Analysis	6/16...	6/21/22	
Feasibility Study	6/16...	6/21/22	
▼ Requirement Documentation	6/27...	7/1/22	
User Stories	6/27...	7/1/22	
▼ Design	7/4/22	8/12/22	
UI Design	7/4/22	7/15/22	
API Design	7/18...	7/29/22	
UML Diagrams	8/1/22	8/12/22	
▼ Development	8/15...	9/2/22	
UI Development	8/25...	9/2/22	
NLP Module	8/15...	9/2/22	
Linear Regression Module	8/15...	9/2/22	
Twitter API Module	8/15...	9/2/22	
▼ Testing	9/5/22	9/14/22	

3.1.2: Work Breakdown Structure

3.1.2 PERT Chart

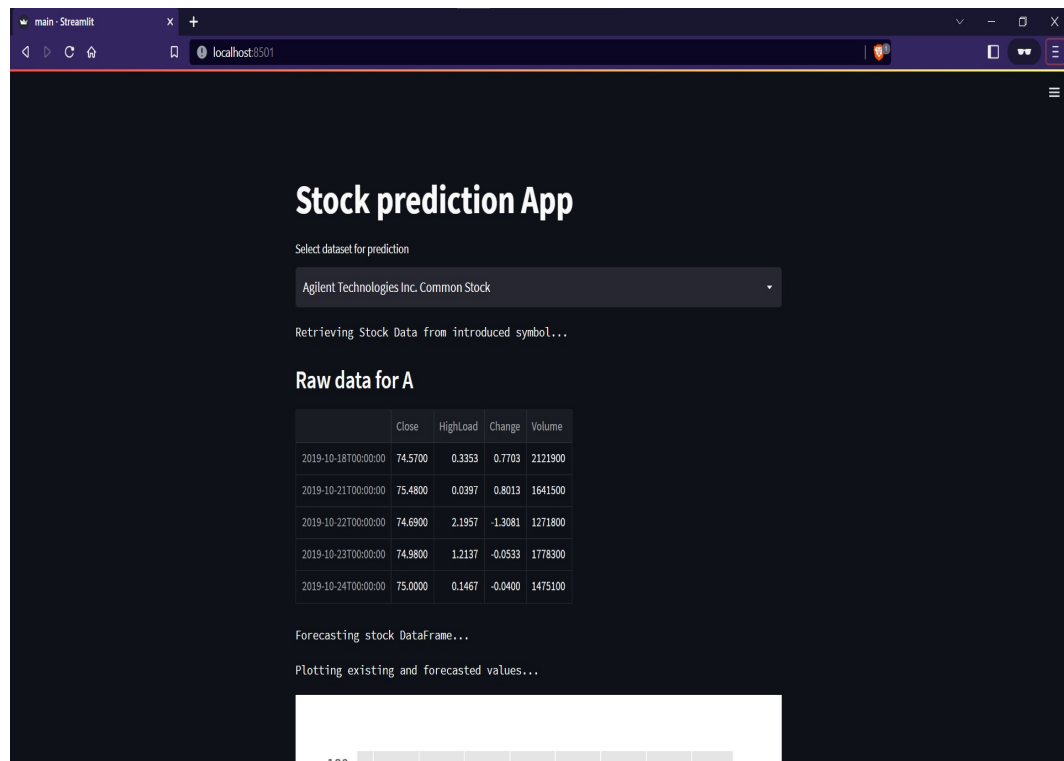


3.1.3: PERT Chart

Sanket Nitnaware (2021510039)
Kevin Rego (2021510056)

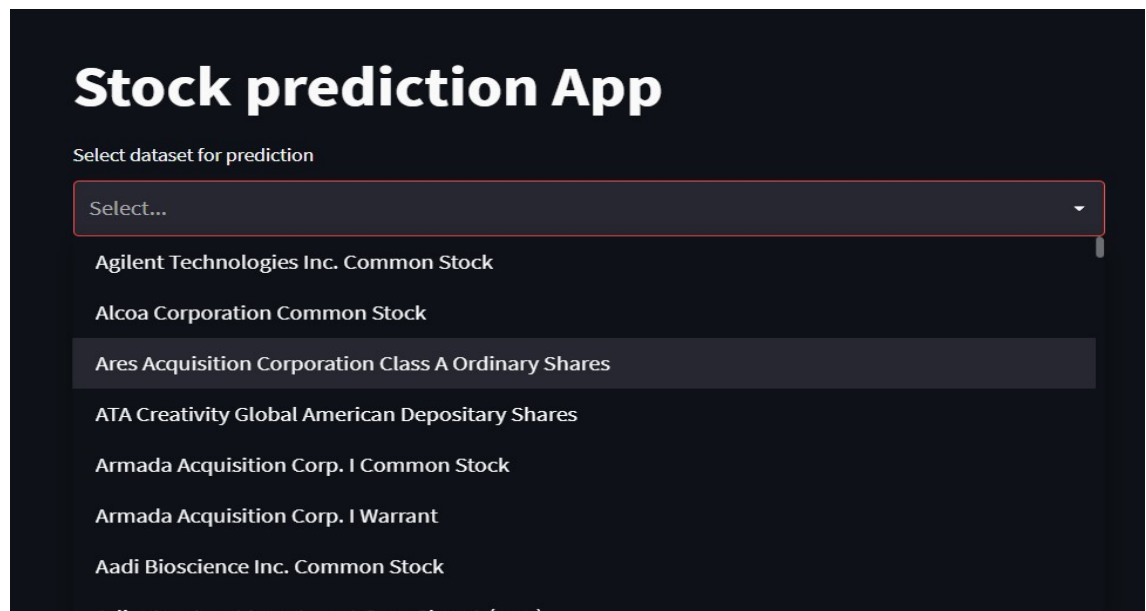
4 Project Implementation and Testing

4.1 Home - Landing View



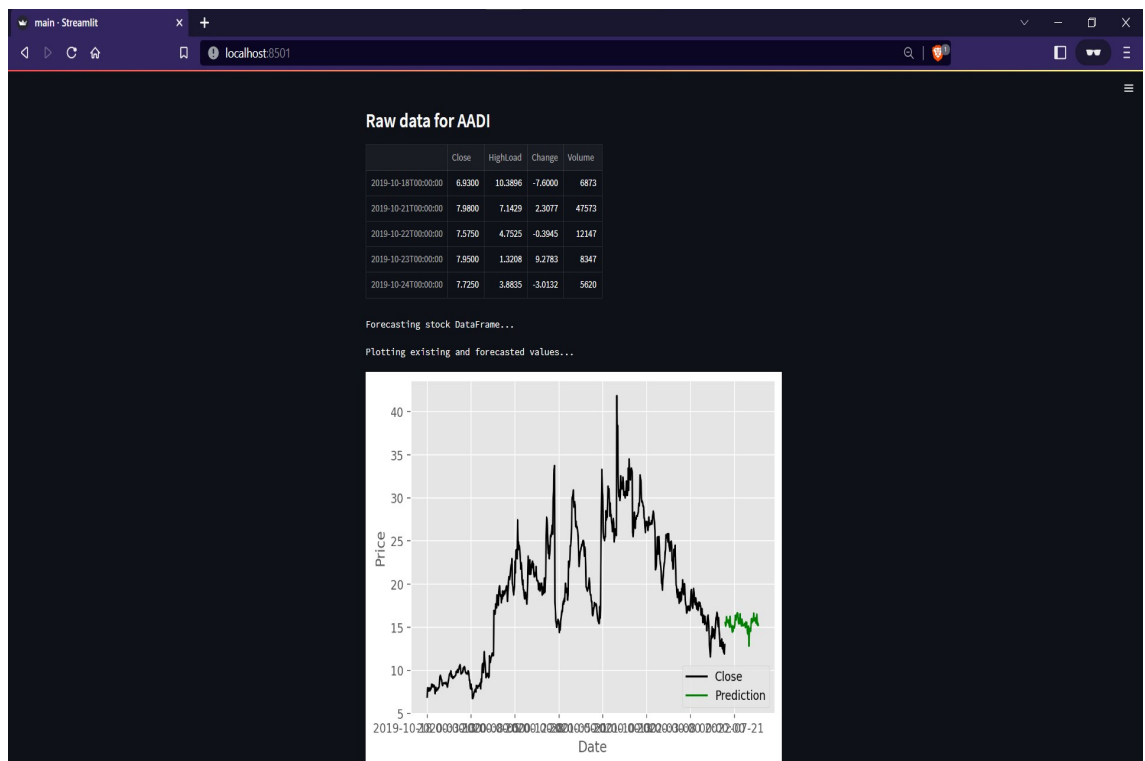
4.1.1: Home View

4.2 Select Box



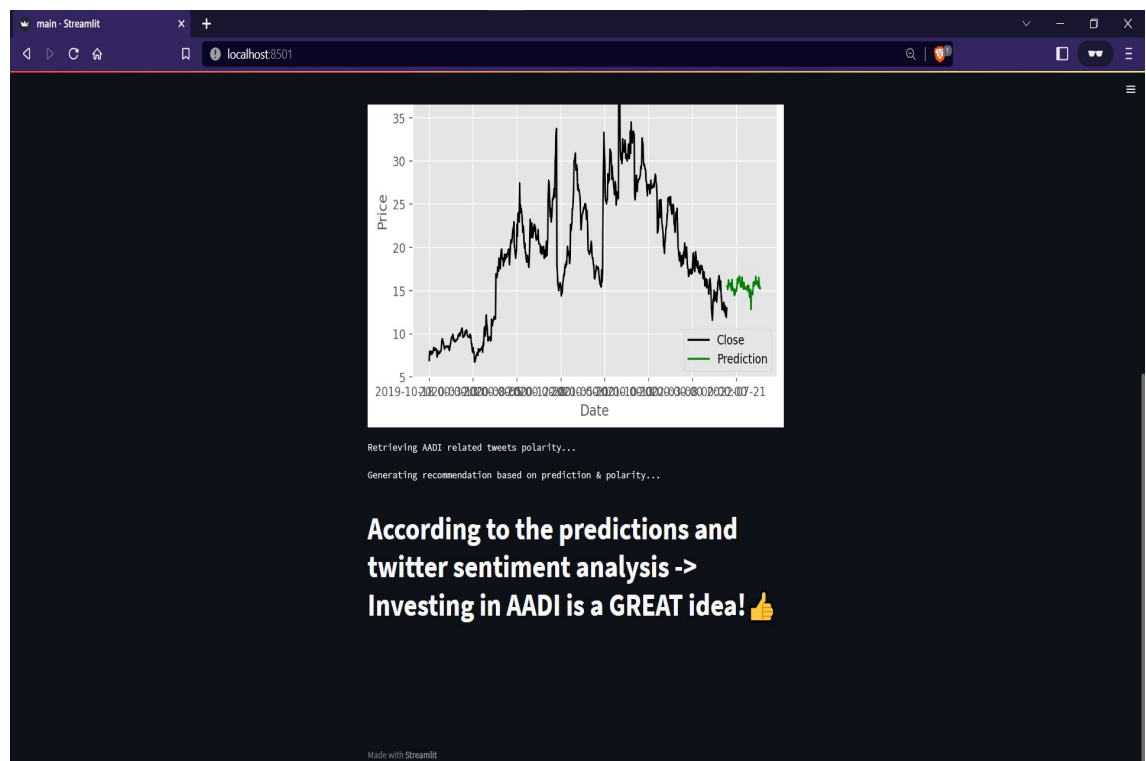
4.2.1: Stocks

4.3 Plotting



4.3.1: Graph View

4.4 Predictions



4.4.1: Stock Predictions

4.5 Code 1

```
import datetime as dt
import math
import yfinance as yf
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import tweepy
from matplotlib import style
from sklearn import preprocessing
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from textblob import TextBlob
import configparser
import constants as ct
from Tweet import Tweet
import streamlit as st
from plotly import graph_objs as go

# To check if the stock symbol introduced via prompt is valid (or if it exists) I check if it appears in a Company List CSV that contains every stock symbol available:

def check_stock_symbol(flag=False, companies_file='new.csv'):
    df = pd.read_csv(companies_file, usecols=[0,1])
    print(df)
    while flag is False:
        for index in range(len(df)):
            if df['Name'][index] == symbol:
                flag = True
                symbols = df['Symbol'][index]
    return flag, symbols
```

4.6 Code 2

```
# create the Pandas DataFrame of the introduced symbol stock market values from the last year from now.

# @st.cache
def get_stock_data(symbol, from_date, to_date):
    data = yf.download(symbol, start=from_date, end=to_date)
    df = pd.DataFrame(data)
    df = df[['Open', 'High', 'Low', 'Close', 'Volume']]
    df['HighLoad'] = (df['High'] - df['Close']) / df['Close'] * 100.0
    df['Change'] = (df['Close'] - df['Open']) / df['Open'] * 100.0
    df = df[['Close', 'HighLoad', 'Change', 'Volume']]
    st.subheader("Raw data for " + symbol)
    st.write(df.head())
    return df

# Modelling using LinearRegression

def stock_forecasting(df):
    forecast_col = 'Close'
    forecast_out = int(math.ceil(0.1*len(df)))
    df['label'] = df[forecast_col].shift(-forecast_out)

    X = np.array(df.drop(['label'], axis=1))
    X = preprocessing.scale(X)
    X_forecast = X[-forecast_out:]
    X = X[:-forecast_out]

    df.dropna(inplace=True)
    y = np.array(df['label'])

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5)

    clf = LinearRegression(n_jobs=-1)
    clf.fit(X_train, y_train)
    accuracy = clf.score(X_test, y_test)
    forecast = clf.predict(X_forecast)

    df['Prediction'] = np.nan
```

4.7 Code 3

```
last_date = df.iloc[-1].name
last_date = dt.datetime.strptime(str(last_date), "%Y-%m-%d %H:%M:%S")

for pred in forecast:
    last_date += dt.timedelta(days=1)
    df.loc[last_date.strftime("%Y-%m-%d")] = [np.nan for _ in range(len(df.columns) - 1)] + [pred]
return df, forecast_out

def forecast_plot(df):
    df['Close'].plot(color='black')
    df['Prediction'].plot(color='green')
    plt.legend(loc=4)
    plt.xlabel('Date')
    plt.ylabel('Price')
    st.pyplot(plt)
    plt.show()

# Twitter sentiment analysis

def retrieving_tweets_polarity(symbol):
    config = configparser.ConfigParser()
    config.read('config.ini')

    api_key = config['twitter']['api_key']
    api_key_secret = config['twitter']['api_key_secret']
    access_token = config['twitter']['access_token']
    access_token_secret = config['twitter']['access_token_secret']
    auth = tweepy.OAuthHandler(api_key, api_key_secret)
    auth.set_access_token(access_token, access_token_secret)
    user = tweepy.API(auth)

    tweets = tweepy.Cursor(user.search_tweets, q=str(symbol), lang='en').items(ct.num_of_tweets)

    tweet_list = []
    global_polarity = 0
```

4.8 Code 5

```
for tweet in tweets:
    tw = tweet._json['text']
    # st.write(tw)
    blob = TextBlob(tw)
    polarity = 0
    for sentence in blob.sentences:
        polarity += sentence.sentiment.polarity
    global_polarity += sentence.sentiment.polarity
    tweet_list.append(tweet.tw, polarity)
global_polarity = global_polarity / len(tweet_list)
return global_polarity

# Recommendation Function

def recommending(df, forecast_out, global_polarity):
    if df.iloc[forecast_out-1]['Close'] < df.iloc[-1]['Prediction']:
        if global_polarity > 0:
            st.markdown("# According to the predictions and twitter sentiment analysis -> Investing in %s is a GREAT ideal" % str(symbol) + ":thumbsup:")
        elif global_polarity < 0:
            st.markdown("# According to the predictions and twitter sentiment analysis -> Investing in %s is a BAD ideal" % str(symbol) + ":thumbsdown:")
        else:
            st.markdown("# According to the predictions and twitter sentiment analysis -> Investing in %s is a BAD ideal" % str(symbol) + ":thumbsdown:")

if __name__ == "__main__":
    # Start
    st.title("Stock prediction App")
    d = pd.read_csv('new.csv', usecols=[0, 1])
    # Get stock list
    stocks = []
    for index in range(len(d)):
        stocks.append(d['Name'][index])

    symbol = st.selectbox("Select dataset for prediction", stocks)
    style.use('ggplot')
    (flag, symbol) = check_stock_symbol(False, 'new.csv')
    if flag:
        actual_date = dt.date.today()
```

4.9 Code 6

```
if __name__ == "__main__":
    # start
    st.title("Stock prediction App")
    d = pd.read_csv('new.csv', usecols=[0, 1])
    # get stock list
    stocks = []
    for index in range(len(d)):
        stocks.append(d['Name'][index])

    symbol = st.selectbox("Select dataset for prediction", stocks)
    style.use('ggplot')
    (flag, symbol) = check_stock_symbol(False, 'new.csv')
    if flag:
        actual_date = dt.date.today()
        past_date = actual_date - dt.timedelta(days=365 * 3)
        actual_date = actual_date.strftime("%Y-%m-%d")
        past_date = past_date.strftime("%Y-%m-%d")
        st.text("Retrieving Stock Data from introduced symbol...")
        dataframe = get_stock_data(symbol, past_date, actual_date)
        st.text("Forecasting stock DataFrame...")
        (dataframe, forecast_out) = stock_forecasting(dataframe)
        st.text("Plotting existing and forecasted values...")
        forecast_plot(dataframe)
        st.text("Retrieving " + symbol + " related tweets polarity...")
        polarity = retrieving_tweets_polarity(symbol)
        st.text("Generating recommendation based on prediction & polarity...")
        recommending(dataframe, forecast_out, polarity)
    else:
        st.text("Please enter correct name")
```

5 Test Cases

Table 6.1: Test Case - Login and Register

Test Case ID	Test Case Name	Test Data	Expected Output	Actual Output	Result
1	User enters company name	Enters the correct company name	Shows Prediction	Prediction	Pass
2	User enters company name	Enters the wrong company name	Shows No Result	No Result	Pass

6 Limitations

- It needs internet to be accessed.
- It supports limited Company

7 Future Enhancements

- More Companies to be added.
- Integrate with real stock applications for predictions.
- Can make this app fully offline.

8 User Manual

Part 1 – Open Web Application

Upon opening the web Application, the user will see the home screen with a select box consisting of company names.

Part 2 – Select company

User can select any company of which they want to see the predictions.

Part 3 – View Predicions

User upon selecting the company can view the prediction generated by the application.

9 Bibliography

9.1 Web References

- [1.] <https://cs229.stanford.edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimen.pdf>
- [2.] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
- [3.] <https://docs.streamlit.io/>
- [4.] <https://docs.streamlit.io/>
- [5.] <https://www.draw.io/>
- [6.] <https://www.geeksforgeeks.org/>