"Stock Price Prediction using Time Series Analysis and Sentimental Analysis"

A Project Report Submitted to
Rajiv Gandhi Proudyogiki Vishwavidyalaya

Towards Partial Fulfillment for the Award of
Bachelor of Technology in *Computer Science Engineering*

Submitted by:	***Guided by:***
Suyash Jain(0827CS181221)	**Dr. Praveen Bhanodia**
Yagvendra Khichi(0827CS181241)	**Professor CSE**
Yashwant Patidar(0827CS181248)	**AITR, Indore**
Yogesh Sharma(0827CS181249)	

Acropolis Institute of Technology & Research, Indore

Jan - June 2022

EXAMINER APPROVAL

The project entitled **“Stock Price Prediction using Time Series Analysis and Sentimental Analysis”** submitted by Suyash Jain (0827CS181221), Yagvendra Khichi (0827CS181241), Yashwant Patidar (0827CS181248) and Yogesh Sharma (0827CS181249) has been examined and is hereby approved towards partial fulfillment for the award of ***Bachelor of Technology degree in Computer Science Engineering*** discipline, for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.

(Internal Examiner) **(External Examiner)**

Date: **Date:**

GUIDE RECOMMENDATION

This is to certify that the work embodied in this project entitled **"Stock Price Prediction using Time Series Analysis and Sentimental Analysis"** submitted by Suyash Jain (0827CS181221), Yagvendra Khichi (0827CS181241), Yashwant Patidar (0827CS181248) and Yogesh Sharma (0827CS181249) is a satisfactory account of the Bonafide work done under the supervision of **Dr. Praveen Bhanodia** is recommended towards partial fulfillment for the award of the Bachelor of Technology (Computer Science Engineering) degree by Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal.

(Project Guide) **(Project Coordinator)**

STUDENTS UNDERTAKING

This is to certify that project entitled **“Stock Price Prediction using Time Series Analysis and Sentimental Analysis”** has been developed by us under the supervision of **Dr. Praveen Bhanodia.** The whole responsibility of work done in this project is ours. The sole intension of this work is only for practical learning and research.

We further declare that to the best of our knowledge; this report does not contain any part of any work which has been submitted for the award of any degree either in this University or in any other University / Deemed University without proper citation and if the same work found then we are liable for explanation to this.

Suyash Jain(0827CS181221)

Yagvendra Khichi(0827CS181241)

Yashwant Patidar(0827CS181248)

Yogesh Sharma (0827CS181249)

Acknowledgement

We thank the almighty Lord for giving me the strength and courage to sail out through the tough and reach on shore safely.

There are number of people without whom this projects work would not have been feasible. Their high academic standards and personal integrity provided me with continuous guidance and support.

We owe a debt of sincere gratitude, deep sense of reverence and respect to our guide and mentor **Dr. Praveen Bhanodia,** Professor, CSE, AITR, Indore for his motivation, sagacious guidance, constant encouragement, vigilant supervision and valuable critical appreciation throughout this project work, which helped us to successfully complete the project on time.

We express profound gratitude and heartfelt thanks to **Dr. Kamal Kumar Sethi**, Head of Department, CSE, AITR, Indore for his support, suggestion and inspiration for carrying out this project. I am very much thankful to other faculty and staff members of CSE Department, AITR Indore for providing me all support, help and advice during the project. We would be failing in our duty if do not acknowledge the support and guidance received from **Dr. S C Sharma**, Director, AITR, Indore whenever needed. We take opportunity to convey my regards to the management of Acropolis Institute, Indore for extending academic and administrative support and providing me all necessary facilities for project to achieve our objectives.

We are grateful to **our parent** and **family members** who have always loved and supported us unconditionally. To all of them, we want to say "Thank you", for being the best family that one could ever have and without whom none of this would have been possible.

Suyash Jain (0827CS181221)

Yagvendra Khichi (0827CS181241)

Yashwant Patidar (0827CS181248)

Yogesh Sharma (0827CS181249)

Executive Summary

Stock Price Prediction using Time Series Analysis and Sentimental Analysis

This project is submitted to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (MP), India for partial fulfillment of Bachelor of Engineering in Computer Science Engineering branch under the sagacious guidance and vigilant supervision of

Dr. Praveen Bhanodia.

The "Stock Price Prediction using Time Series Analysis and Sentimental Analysis" has been developed to override the problems prevailing in the practicing Stock Prediction. Stock price forecasting is a popular and important topic in financial and academic studies. Stock price prediction has always attracted people interested in investing in share market and stock exchanges because of the direct financial benefits. It is also an important research topic in finance. Prediction of stock market returns is a very complex issue depends on so many factors such company financial status and national policy etc. These days stock prices are affected due to many reasons like company related news, political, social economic conditions. This software is supported to eliminate and, in some cases, reduce the hardships faced by these existing systems. Moreover, this system is designed for the particular need of the company to carry out operations in a smooth and effective manner. The application is reduced as much as possible to avoid errors while entering the data. Thus, by this all it proves it is user- friendly.

Key words: Stock price prediction, Sentimental Analysis, Long Short Term Memory.

“Dream is not the thing you see in sleep but is that thing that doesn’t let you sleep”

-A P J Abdul Kalam

List of Figures

Table of Contents

Chapter 1. Introduction

Introduction

Stock market forecasting is a difficult undertaking since it necessitates in-depth knowledge of events, historical data analysis, and the impact of news on stock price movements. The problem is made even more difficult by the extreme volatility of stock price patterns.

The whole point of stock price forecasting is to invest money with a calculated assessment of risk. However, a comprehensive analysis of the general background of stock prediction is lacking in the literature.

The tool "Stonks" was created to address the issues that arise in the practice of stock prediction. In the current machine learning arena, stock price prediction is an important task. To tackle the challenges, several ways have been presented. The majority of them are focused on a stock's time series analysis, while some are based on news feelings to predict a stock's value. Our tool provides both options to the user i.e., prediction using historical data as well as public sentiment about the company on Social Media Platform (Twitter).

1.1 Overview

The project will be useful for investors to invest in stock market based on various factors. The project target is to create an application that can analyze the historical data of the companies and implement these values to the model created to determine the value the particular stock will have in near future with suitable accuracy.

The main feature of the project is to generate an approximate forecasting of stock price and create a general idea of future values based on historical data by generating a pattern and also analyze the public sentiments about the company through Twitter API and generate a graphical representation of behavior of stocks.

1.2 Background and Motivation

Stock Price Analysis of stocks using previous years data will be useful for new investors to invest in stocks based on various factors considered by our software. Stock price of a company depends on various factors such as supply and demand, company's earning or quarterly corporate results, inflation, public sentiments, behavioral factors of traders and investors and economic condition of country.

For people willing to try their luck in stock market, "Stonks" is a generalized suggestion tool which predicts price of stock analyzing above listed factors and will help in choosing stock with better returns.

1.3 Problem Statement and Objectives

The Existing systems works on various methods from regression to classification. There are many drawbacks such as existing system fails when there are rare outcomes or predictors, as the algorithm is based on bootstrap sampling. The previous results indicate that the stock price is unpredictable when the traditional classifier is used.

The existing system reported highly predictive values, by selecting an appropriate time period for their experiment to obtain highly predictive scores. It does not perform well when there is a change in the operating environment. It does focus on external events in the environment, like news events or social media.

The objective of the project are as follows:

- To predict future stock price
- To generate pattern from previous years large dataset for the prediction of stock price.
- To Analysis the public sentiment about a particular company.
- To provide a platform for investors from which they can select which company to invest in.

The project will be useful for investors to invest in the stock market based on various factors. The project target is to create an application that can analyze the historical data of the companies and implement these values to the model created to determine the value of that particular stock will have in near future with suitable accuracy and also analyze public sentiments about that company on social media platform (Twitter).

The main feature of the project is to generate an approximate forecasting of stock price and create a general idea of future values based on previous data by generating a pattern and extracting public perception about that company from Twitter.

1.4 Scope of the Project

Stock Price Analysis of stocks using historical data will be useful for new investors to invest in stocks based on various factors considered by our software. Stock price of a company depends on various factors such as supply and demand, company's earning or quarterly corporate results, inflation, behavioral factors of traders and investors and economic condition of country and public sentiments.

For people willing to try their luck in stock market, "Stonks" is a generalized suggestion tool which predicts price of stock analyzing above listed factors and will help in choosing stock with better returns.

1.5 Team Organization

- **Suyash Jain:**
    - I worked as the project manager, prepared dataset for the project and helped in implementation.
- **Yagvendra Khichi:**
    - I worked on documentation of the project and helped in functional testing of the project.
- **Yashwant Patidar:**
    - I worked on testing, implementation and documentation of the project.
- **Yogesh Sharma:**
    - I worked on the implementation, documentation of the project and created model for project.

1.6 Report Structure

The project **Stonks** is primarily concerned with the **predicting stock price** and whole project report is categorized into five chapters.

Chapter 1: Introduction - Introduces the background of the problem followed by rationale for the project undertaken. The chapter describes the objectives, scope and applications of the project. Further, the chapter gives the details of team members and their contribution in development of project which is then subsequently ended with report outline.

Chapter 2: Review of Literature - Explores the work done in the area of Project undertaken and discusses the limitations of existing system and highlights the issues and challenges of project area. The chapter finally ends up with the requirement identification for present project work based on findings drawn from reviewed literature and end user interactions.

Chapter 3: Proposed System - Starts with the project proposal based on requirement identified, followed by benefits of the project. The chapter also illustrate software engineering paradigm used along with different design representation. The chapter also includes block diagram and details of major modules of the project. Chapter also gives insights of different type of feasibility study carried out for the project undertaken. Later it gives details of the different deployment requirements for the developed project.

Chapter 4: Implementation - Includes the details of different Technology/ Techniques/ Tools/ Programming Languages used in developing the Project. The chapter also includes the different user interface designed in project along with their functionality. Further it discusses the experiment results along with testing of the project. The chapter ends with evaluation of project on different parameters like accuracy and efficiency.

Chapter 5: Conclusion - Concludes with objective wise analysis of results and limitation of present work which is then followed by suggestions and recommendations for further improvement.

Chapter 2. Review of Literature

Review of Literature

In the topic of stock price prediction, a lot of effort has been done; yet, getting the desired out comes is not straightforward. Following are some of the significant conclusions drawn from the literature review:

Stock market forecasting is a difficult undertaking since it necessitates in-depth knowledge of events, historical data analysis, and the impact of news on stock price movements. The problem is made even more difficult by the extreme volatility of stock price patterns. Several studies based on stock price prediction also goes into the usage of approaches for extracting

opinions. It also highlights the application of domain knowledge in both textual feature extraction methodologies. It also emphasizes the need of using deep neural network-based prediction approaches to uncover the hidden relationship between textual and numerical data. These studies are significant and unique in that it develops a comprehensive framework for stock market forecasting and identifies the strengths and flaws of existing methods. It covers a wide range of open issues and research directions that the research community may find useful. There are several studies such as Forecasting Stock trend using Time Series data, using Textual Data, using Sentimental Analysis, Using Numerical and Textual Data[1]. Each has different methods for prediction and considers various parameters and precision. Stock market patterns are exceedingly volatile, which makes forecasting difficult.

Significant work has been done in the field of Stock Price Prediction; however, it is not easy to achieve desired results. The review of literature leads to draw certain major findings which are as under:

"What other people think" has always been an important piece of information for most of us during the decision-making process. The Internet and the Web have now (among other things) made it possible to find out about the opinions and experiences of those in the vast pool of people that are neither our personal acquaintances nor well-known professional critics — that is, people we have never heard of. And conversely, more and more people are making their opinions available to strangers via the Internet. The interest that individual users show in online opinions about products and services, and the potential influence such opinions wield, is something that is driving force for this area of interest. And there are many challenges involved in this process which needs to be walked all over in order to attain proper outcomes out of them. In this survey we analysed basic methodology that usually happens in this process and measures that are to be taken to overcome the challenges being faced.

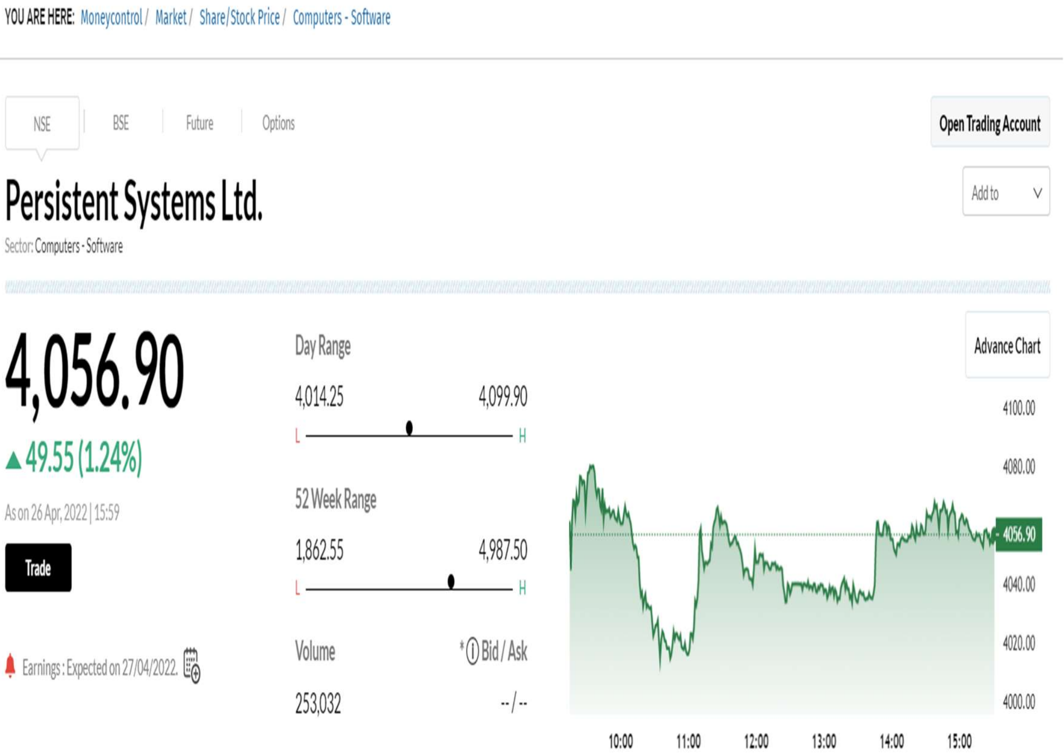
Stock market prediction is a challenging task as it requires deep insights for extraction of news events, analysis of historic data, and impact of news events on stock price trends. The challenge is further exacerbated due to the high volatility of stock price trends. However, a detailed overview that discusses the overall context of stock prediction is elusive in literature. To address this research gap, this paper presents a detailed survey. All key terms and phases of generic stock prediction methodology along with challenges, are described. A detailed literature review that covers data pre-processing techniques, feature extraction techniques, prediction techniques, and future directions is presented for news sensitive stock prediction. This work investigates the significance of using structured text features rather than unstructured and shallow text features. It also discusses the use of opinion extraction techniques. In addition, it emphasizes the use of domain knowledge with both historical data and sentimental analysis. This is approaches of textual feature extraction. Furthermore, it highlights the significance of deep neural network-based prediction techniques to capture the hidden relationship between textual and numerical data. This survey is significant and novel as it elaborates a comprehensive framework for stock market prediction and highlights the strengths and weaknesses of existing approaches. It presents a wide range of open issues and research directions that are beneficial for the research community Stock market trends are extremely volatile in nature that makes prediction quite hard.

2.1 Preliminary Investigation

2.1.1 Current System

In Current system, it reported highly predictive values, by selecting an appropriate time period for their experiment to obtain highly predictive scores. The existing system does not perform well when there is a change in the operating environment. It doesn't focus on external events in the environment, like news events or social media. It exploits only one data source, thus highly biased. The stock market prediction process is filled with uncertainty and can be influenced by multiple factors. Therefore, the stock market plays an important role in business and finance. The technical and fundamental analysis is done by sentimental analysis process. Social media data has a high impact due to its increased usage, and it can [6] The vast majority of the stockbrokers while making the prediction utilized the specialized, fundamental or the time series analysis. Overall, these techniques couldn't be trusted completely, so there emerged the need to give a strong strategy to financial exchange prediction. To find the best accurate result, the methodology chose to be implemented as machine learning and AI along with supervised classifier. Results were tried on the binary classification utilizing SVM classifier with an alternate set of a feature list. The greater part of the Machine Learning approach for taking care of business [2] issues had their benefit over factual techniques that did exclude AI, despite the fact that there was an ideal procedure for specific issues. Swarm Intelligence [2] optimization method named Cuckoo search was most easy to accommodate the parameters of SVM. The proposed hybrid cs-svm strategy exhibited the performance to create increasingly exact outcomes in contrast with ANN. Likewise, the SC-SVM display [2] performed better in the forecasting of the stock value prediction. Prediction stock cost utilized parse records to compute the predicted, send it to the user, and autonomously perform tasks like buying and selling shares utilizing automation concept. Naïve Bayes Algorithm was utilized. [8].

The technical and fundamental or the time series analysis is used by the most of the stockbrokers while making the stock predictions. The programming language is used to predict the stock market using machine learning is Python. In this paper we propose a Machine Learning (ML) approach that will be trained from the available stocks data and gain intelligence and then uses the acquired knowledge for an accurate prediction. In this context this study uses a machine learning technique called Support Vector Machine (SVM) to predict stock prices for the large and small capitalizations and in the three different markets, employing prices with both daily in this market. Moneycontrol.com today gets over 17 million visitors every month across all its platforms-web, mobile and tablets that makes it the largest online financial platform in India. But while we've radically changed and evolved, the belief and passion to be the best and the most insightful hasn't. That, we hope, keeps us ticking. This we believe is merely the start of our journey.



YOU ARE HERE: Moneycontrol / Market / Share/Stock Price / Computers - Software

Persistent Systems Ltd.

Sector: Computers - Software

4,056.90

▲49.55 (1.24%)

As on 26 Apr, 2022 | 15:59

Day Range: L 4,014.25 – H 4,099.90

52 Week Range: L 1,862.55 – H 4,987.50

Volume: 253,032

Bid / Ask: -- / --

Earnings : Expected on 27/04/2022.

Fig 2.1 Money Control Graph

This figure shows the analysis of the company on money control web and it contains today’s closing price and the percentage of increment in market value. There are day range which shows current day range and the 52-week range which shows 52-week range and current volume of the shares. There is a graph which shows the trend of the stock price.

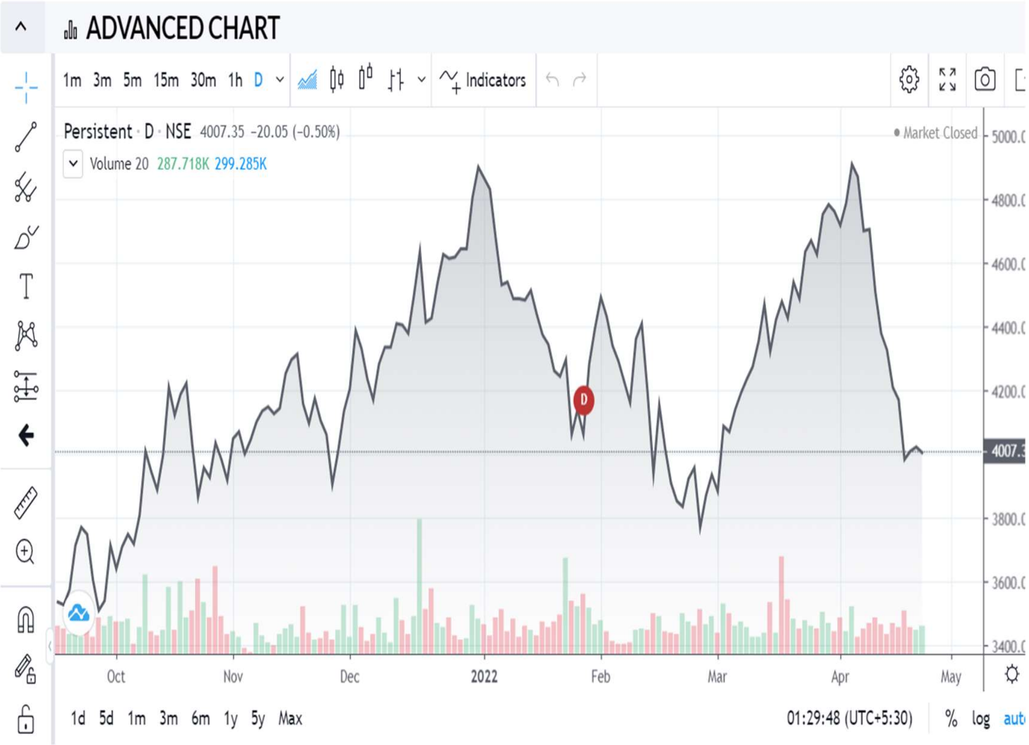
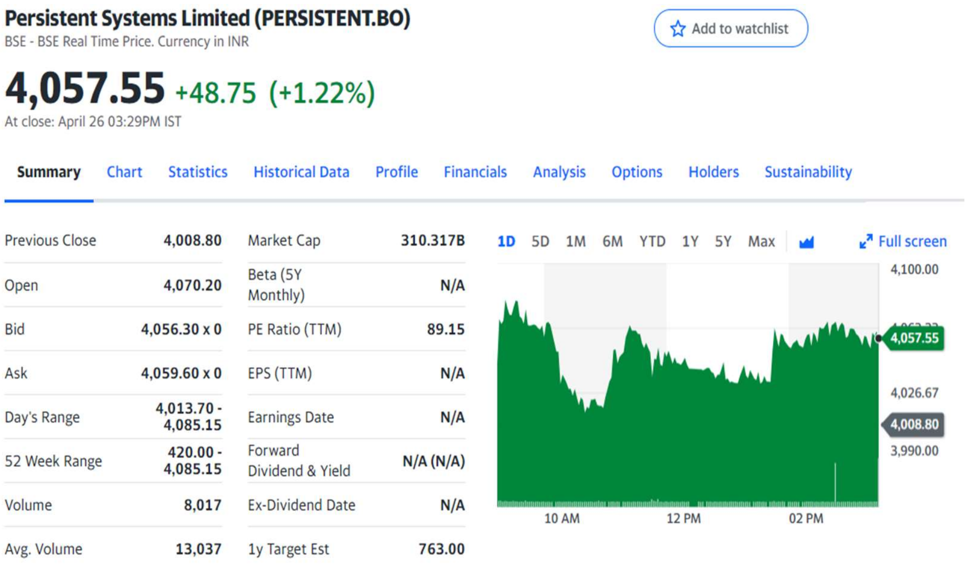


Fig 2.2 Money Control Advance Chart

This figure shows the advance chart of the particular company and here as like D point, we can select points go through the insight of that point and every point show values of the stock price at that point.

I'm Yahoo! I have the best search results around. Yahoo! Finance is a media property that is part of the Yahoo! Group. We are experts in providing news, analysis and information on the stock market. I have a great network of professionals who can help me with my project. Our site provides up-to-date financial news, data, and commentary. We also include stock quotes, press releases, financial reports, and original content. The site offers tools to help manage your finances online. We also post original stories written by our team of staff journalists.

Yahoo! Finance is a media property that is part of the Yahoo! network. It provides financial news, data and commentary including stock quotes, press releases, financial reports, and original content. It also offers some online tools for personal finance management. In addition to posting partner content from other web sites.



Persistent Systems Limited (PERSISTENT.BO)

BSE - BSE Real Time Price. Currency in INR

4,057.55 +48.75 (+1.22%)

At close: April 26 03:29PM IST

Previous Close	4,008.80	Market Cap	310.317B
Open	4,070.20	Beta (5Y Monthly)	N/A
Bid	4,056.30 x 0	PE Ratio (TTM)	89.15
Ask	4,059.60 x 0	EPS (TTM)	N/A
Day's Range	4,013.70 - 4,085.15	Earnings Date	N/A
52 Week Range	420.00 - 4,085.15	Forward Dividend & Yield	N/A (N/A)
Volume	8,017	Ex-Dividend Date	N/A
Avg. Volume	13,037	1y Target Est	763.00

Fig 2.3 Yahoo Finance Historical data

This figure shows the analysis of the company on Yahoo finance web and it contains today's open price and previous closing price and the percentage of increment in market value. There are day range which shows current day range and the 52 week range which shows 52 week range and current volume of the shares. There is a graph which shows the trend of the stock price.

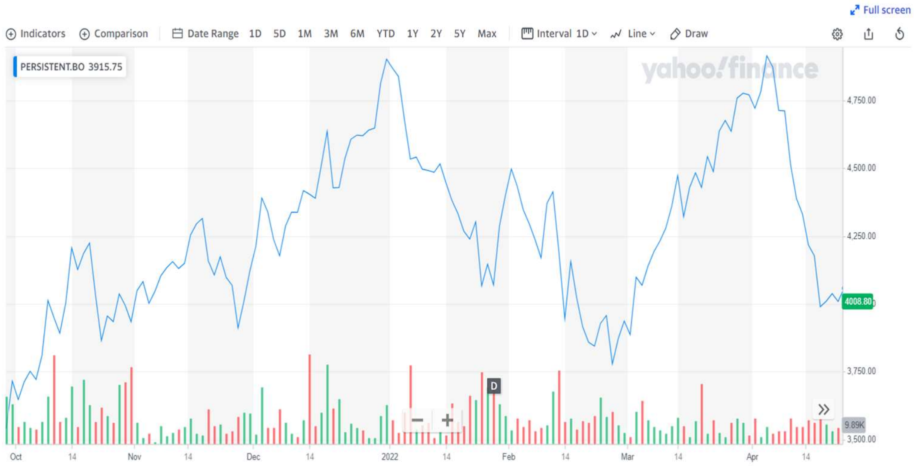


Fig 2.4 Yahoo Finance Graph

This figure shows the chart of the particular company and here as like D point we can select points go through the insight of that point and every point shows values of the stock price at that point.

2.2 Limitations of Existing Systems

The limitations of these are as follows:

- The existing system fails when there are rare outcomes or predictors, as the algorithm is based on bootstrap sampling.
- The previous results indicate that the stock price is unpredictable when the traditional classifier is used.
- The existence system reported highly predictive values, by selecting an appropriate time period for their experiment to obtain highly predictive scores.
- The existing system does not perform well when there is a change in the operating environment.
- It doesn't focus on external events in the environment, like news events or social media.
- It exploits only one data source, thus highly biased.
- The existing system needs some form of input interpretation, thus need of scaling.
- It doesn't exploit data pre-processing techniques to remove inconsistency and incompleteness of the data.

There is manual method too that can predict the price such as statistical, sentiment analysis.

2.3 Requirement Identification and Analysis for Project

Significant work has been done in the field of Stock Price Prediction; however, it is not easy to achieve desired results. The review of literature leads to draw certain major findings which are as under:

Stock market prediction is a challenging task as it requires deep insights for extraction of news events, analysis of historic data, and impact of news events on stock price trends. The challenge is further exacerbated due to the high volatility of stock price trends. However, a detailed overview that discusses the overall context of stock prediction is elusive in literature. All key terms and phases of generic stock prediction methodology along with challenges, are described. A detailed literature review that covers data preprocessing techniques, feature extraction techniques, prediction techniques, and future directions is presented for news sensitive stock prediction. This work investigates the significance of using structured text features rather than unstructured and shallow text features. It also discusses the use of opinion extraction techniques. In addition, it emphasizes the use of domain knowledge with both approaches of textual feature extraction. Furthermore, it highlights the significance of deep neural network-based prediction techniques to capture the hidden relationship between textual and numerical data. This survey is significant and novel as it elaborates a comprehensive framework for stock market prediction and highlights the strengths and weaknesses of existing approaches. It presents a wide range of open issues and research directions that are beneficial for the research community.

Stock market trends are extremely volatile in nature that makes prediction quite hard. This volatile nature attracts researchers to investigate sophisticated techniques for better prediction. Prediction of stock market trends with high accuracy generates significant revenue. Fundamental and technical analyses are two basic approaches used for stock trend prediction. Technical analysis inspects past data and volumes of stock prices while fundamental analysis not only considers stock statistics but also evaluates industry's performance, political events, and economic circumstances (Patel et al., 2015; Milosevic,2016) [1]. Fundamental analysis of Stock market trends is extremely volatile in nature that makes prediction quite hard. This volatile nature attracts researchers to investigate sophisticated techniques for better prediction.

Prediction of stock market trends with high accuracy generates significant revenue. Fundamental and technical analyses are two basic approaches used for stock trend prediction. Technical analysis inspects past data and volumes of stock prices while fundamental analysis not only considers stock statistics but also evaluates industry's performance, political events, and economic circumstances (Patel et al., 2015; Milosevic,2016) [2]. Fundamental analysis is more realistic because it evaluates the market in a broader scope.

This survey puts emphasis on research work based on fundamental analysis, where textual data is considered along with stock price historical data for stock prediction. There are many sources of textual data like news, tweets, and annual reports, etc. which could be analyzed to mine significant information. Textual data, especially news, is a better source of hidden information than numeric data because it permits to predict financial trends with its justification (Chan & Franklin, 2011) [5].

For instance, a news article on accompanies with words or phrases like “resignation", "risk of default” helps the investor to predict a decrease in the company’s stock prices. Furthermore, news about many Uncertain factors can affect stock market trends (Nassirtoussi et al., 2015) [4]. For instance, economic and political shocks, war, civil unrest, terrorism, and natural disasters etc.

Therefore, there is a great need for better knowledge discovery mechanisms from textual data. Feature extraction is a fundamental step in prediction where input data is reduced into more manageable form for further processing. Most of the previous work on news sensitive stock trend prediction adopted shallow features extraction techniques which are unstructured and where words are represented as features. For instance, Bag-of-Words (Bow), noun phrases, and named entities (Schumaker & Chen, 2009) [1].

This is contrary to the structured feature extraction technique where a combination of words, nouns, and verbs are used. Unlike structured feature extraction techniques, shallow feature extraction techniques are not able to capture a complete event in the form of structured entity- relation information. Consequently, shallow features make it complicated to represent the impact of news events on stock market trend prediction (Ding et al., 2014) [8].

Neural networks can be tricky to use for many practitioners compared to other methods whose properties are already well understood. This has remained a hurdle for newcomers to the field since a lot of practical choices are based on the intuitions of experts, as well as experiences gained over time. With this study, we have attempted to back some of these intuitions with experimental results. We have also presented new insights, both on architecture selection and hyper parameter tuning for LSTM networks which have emerged as the method of choice for solving complex sequence learning problems. In future work, we plan to explore more complex modifications of the LSTM architecture.

Events extracted from news articles may play a significant role in stock market trend prediction. Sophisticated Natural Language Processing (NLP) technologies enable more accurate structured representation of events than shallow features. But structured representation of events increases sparsity, which most probably decreases the predictive power (Ding et al., 2015) [9].

2.4 Conclusion

This chapter reviews the literature surveys that have been done during the research work. The related work that has been proposed by many researchers has been discussed. The research papers related to Stock Prediction till 2021 have been shown. We 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 the proposed method, we are using 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 the stock market and the prediction will produce higher and accurate results. In our above model we are getting accurate results which will be more useful to stock analysts, Business analysts, Stock Market Investors.

Chapter 3. Proposed System

Proposed System

3.1 Proposal

The Proposed system is “Stonks”. A web application through which user can predict the future price of a stock and can analyze the pattern from previous data. The project also accounts various factors such as sentiment, market cap and performance of company to predict the stock price. The prediction methods can be roughly divided into two categories, statistical methods and artificial intelligence methods. Statistical methods include logistic regression model, ARCH model, etc. Artificial intelligence methods include multi-layer perceptron, convolutional neural network, naive Bayes network, back propagation network, single-layer LSTM, support vector machine, recurrent neural network, etc. They used Long short-term memory network (LSTM).

The features of the project are as follows:

- Predict stock price: - User can select a company from a list of companies and here the Time series analysis model will be trained and predict the stock price.
- Analyze the previous data: - The model will use historical data which used as dataset and after analyzing historical data result will be displayed.
- Compare stocks over time using graphs: -Python library Seaborn and Matplotlib are used for printing the graph and visual aspects. Here we show historical data comparison graph.
- Analyze public sentiments: - Through Flair a Python Library analyze sentiments of the company and display visuals.

The project uses Time Series Analysis algorithm for prediction. Python is used to develop the software. Various python library such as pandas, seaborn, matplotlib, NumPy, PyTorch etc. are also used for data cleaning and data visualization. Time-series prediction is a common technique widely used in many real-world applications such as weather forecasting and financial market prediction. It uses the continuous data in a period of time to predict the result in the next time unit. Many timeseries prediction algorithms have shown their effectiveness in practice. The most common algorithms now are based on Recurrent Neural Networks (RNN), as well as its special type - Long-short Term Memory (LSTM) and Gated Recurrent Unit (GRU). Stock market is a typical area that presents time-series data and many researchers' studies on it and proposed various models. In this project, LSTM model is used to predict the stock price.

3.2 Benefits of the company

- The challenges that are overcome by this system:
- The application can predict the open and the close price of the stock.
- The application can analyze public sentiments about the company.
- The application is web based so user can access it whenever and from wherever he/she wants.
- The application has graph to analysis the trend and user can zoom in and zoom out to get clear picture.
- The application has list of company names so user can select the company for which he wants to predict.

3.3 UML Diagrams

3.3.1 Use Case Diagram

In the Unified Modelling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors.

An effective use case diagram can help your team discuss and represent:

- Scenarios in which your system or application interacts with people, organizations, or external systems.
- Goals that your system or application helps those entities (known as actors) achieve.

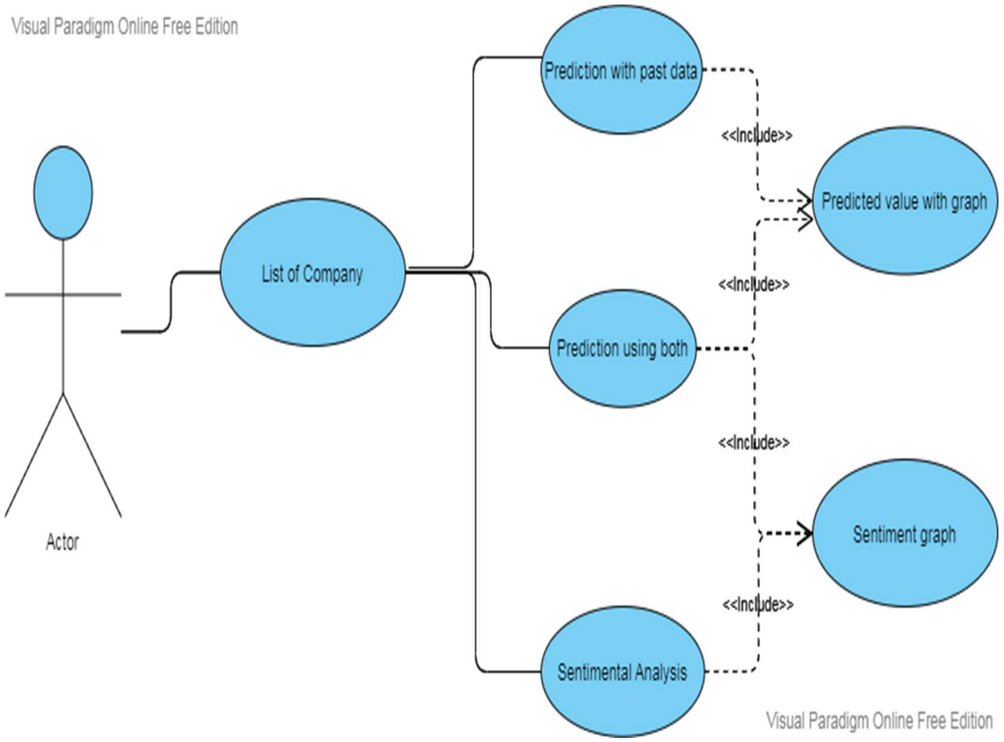

Visual Paradigm Online Free Edition
Prediction with past data
<<Include>>
Predicted value with graph
List of Company
<<Include>>
Prediction using both
<<Include>>
Sentiment graph
Actor
<<Include>>
Sentimental Analysis
Visual Paradigm Online Free Edition

Fig 3. 1 Use-Case Diagram

Actor: - The end user of the system.

List of company: - There are the list of companies in which user can select.

Predictions: - Here we can predict historical data, sentimental analysis and using both options are there.

Result: -After predictions final results will be displayed in visuals or data manner.

3.3.2 Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios. Sequence diagrams can be useful references for businesses and other organizations.

Try drawing a sequence diagram to:

- Represent the details of a UML use case.
- Model the logic of a sophisticated procedure, function, or operation.

- See how objects and components interact with each other to complete a process.
- Plan and understand the detailed functionality of an existing or future scenario.

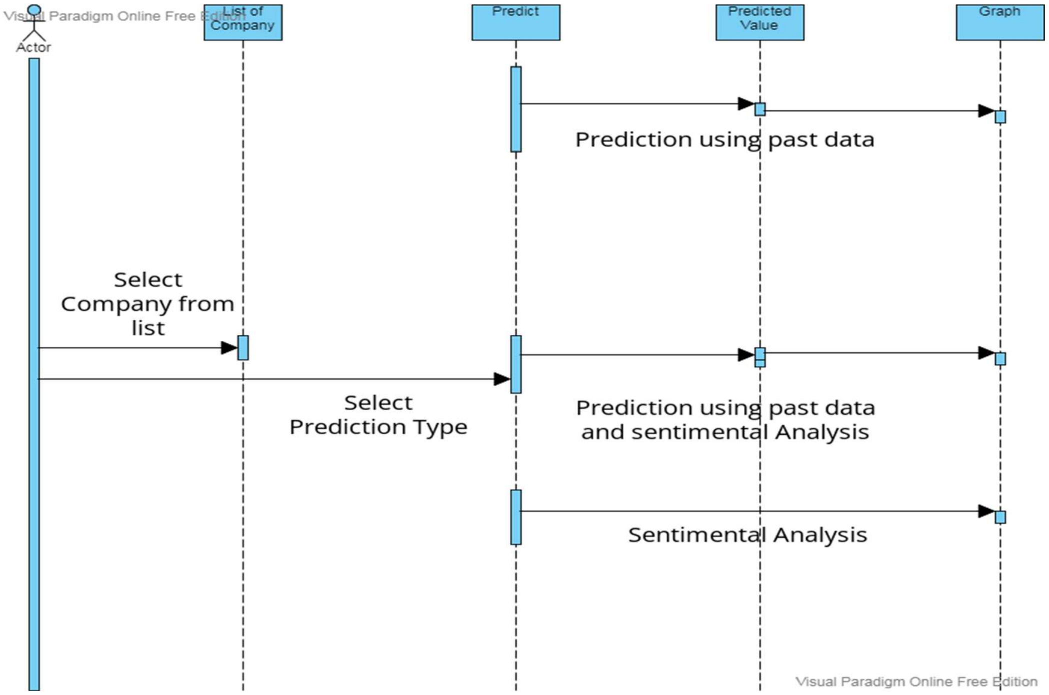

Visual Paradigm Online Free Edition
Actor
List of Company
Predict
Predicted Value
Graph
Prediction using past data
Select Company from list
Select Prediction Type
Prediction using past data and sentimental Analysis
Sentimental Analysis
Visual Paradigm Online Free Edition

Fig 3.2 Sequence Diagram

Actor: - The end user of the system who will select company from list.

List of company: - There are the list of companies in which user can select.

Predictions: - Here we can predict historical data, sentimental analysis and using both options are there.

Result: -After predictions final results will be displayed in visuals or data manner.

3.3.3 Activity Diagram

An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

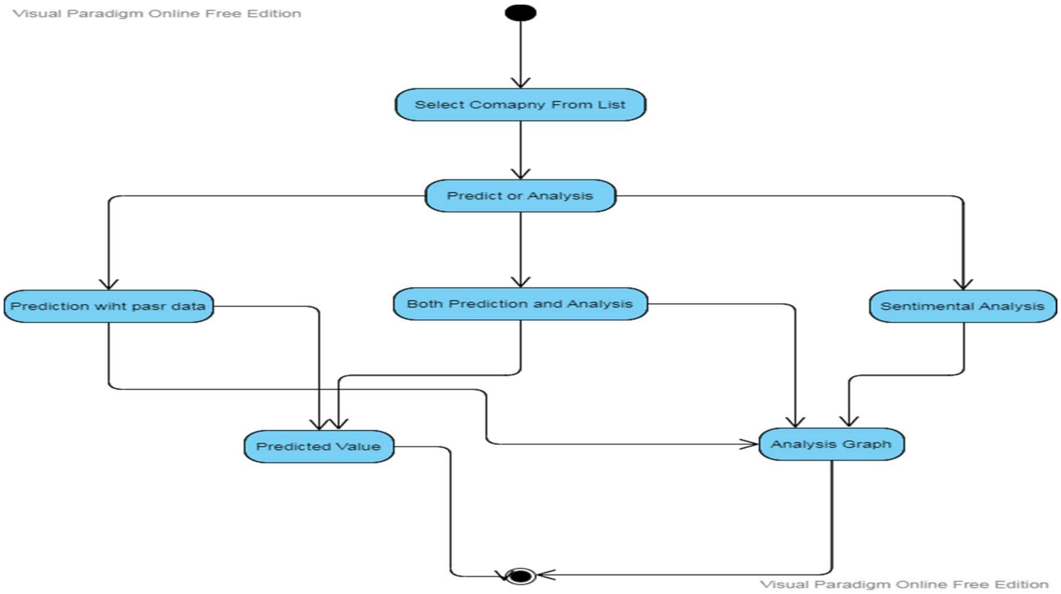

Visual Paradigm Online Free Edition
Select Comapny From List
Predict or Analysis
Prediction wiht pasr data
Both Prediction and Analysis
Sentimental Analysis
Predicted Value
Analysis Graph
Visual Paradigm Online Free Edition

Fig 3.3 Activity Diagram

The end user of the system who will select company from list. There are the list of companies in which user can select. Here we can predict historical data, sentimental analysis and using both options are there. After predictions final results will be displayed in visuals or data manner.

3.4 Feasible Study

A feasibility study is an analysis of how successfully a system can be implemented, accounting for factors that affect it such as economic, technical and operational factors to determine its potential positive and negative outcomes before investing a considerable amount of time and money into it.

3.4.1 Technical

For stock price prediction, there is a need to load and process the data from the csv files. For this, the kind of framework used must be the one that is capable of loading and cleaning the data. The library used in this is Pandas, which is a highly used python library for loading the csv data for training and testing of the model. There is a requirement of data transformation in stock price prediction models and for feature selection for this purpose scikit learn is the best library which is an open-source machine learning library. For developing the predictive model keras as python deep neural network library works on the top of tensorflow and Theano library. For front end web the streamlit library a open source python framework for easy machine learning model deployment is used. visualization is carried out using the plotly library of the python.

For making the system technically feasible, there is a requirement for a GPU-built system for better performance.

3.4.2 Economical

For stock prediction application, there is a need of a high performing GPU enabled system for better training and test of the data.

The streamlit, keras, plotly frameworks used works great with GPU built systems which is an expensive side.

Since this is a web application, this needs to be deployed on the external server which adds cost to the project.

3.4.3 Operational

The main motto of Stonks is to predict stock price and public sentiment of a company.

The system is able to do that accurately and efficiently making the system operationally feasible.

3.5 Design Representation

3.5.1 Data Flow Diagram

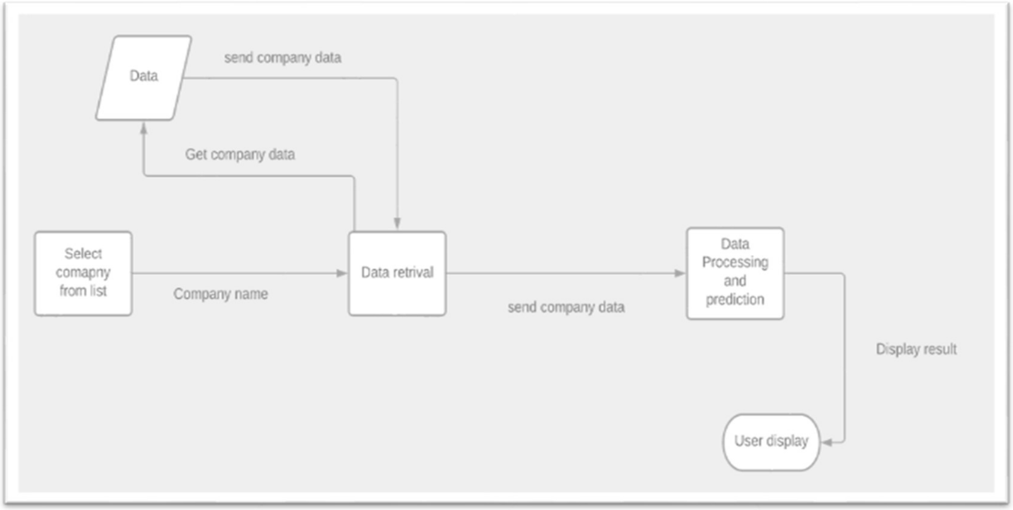


Fig 3.4 Data Flow Diagram

The above diagram is a data flow diagram that shows how the data will flow. First, user have to select one company that he/she want the prediction and after this data of the company selected will be processed and our model will train the data of company and predict the output.

3.6 Deployment Requirements

There are various requirements (hardware, software and services) to successfully deploy the system. These are mentioned below:

3.6.1 Hardware

- A device with Operating System: - Any desktop device that have a browser.
- The Internet connection is required.

3.6.2 Software

- Windows 7 or later operating system.
- Any Web Browser
- Streamlit: - An open-source library for web development using python for deploying machine learning models.
- External Server: - Streamlit Cloud Server

3.6.3 Deployment Diagram

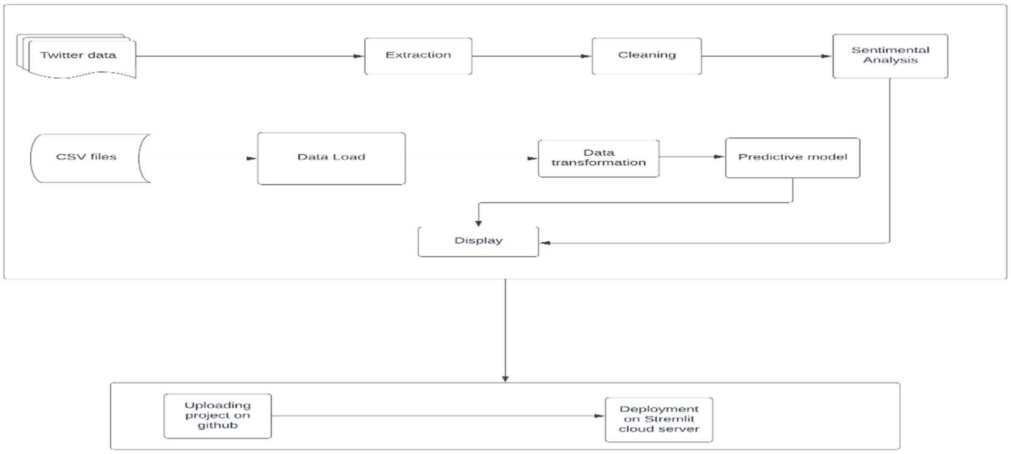


Fig 3.5 Deployment Diagram

This figure shows the deployment structure of the project.

Chapter 4. Implementation

Implementation

Stonks –Stock Price Predictor using Time Series analysis and LSTM, the main feature of the project is to generate an approximate forecasting of stock price and create a general idea of future values based on previous data by generating a pattern and analyze the public sentiments about the company through Twitter API and generate a graphical representation.

4.1 Technology Used

4.1.1 Python

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

4.1.2 Pandas:

Pandas is a software library written for the Python programming language for data manipulation and analysis. It offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.

4.1.3 Seaborn:

Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas' data structures.
Seaborn helps you explore and understand your data. Its plotting functions operate on data frames and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them.

4.1.4 LSTM:

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. This is a behavior required in complex problem domains like machine translation, speech recognition, and more.

LSTMs are a complex area of deep learning. It can be hard to get your hands around what LSTMs are, and how terms like bidirectional and sequence-to-sequence relate to the field.

4.1.5 Keras:

Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. Up until version 2.3, Keras supported multiple backends, including TensorFlow, Microsoft Cognitive Toolkit, Theano.

4.1.6 Scikit-learn:

Scikit-learn (Sklearn) 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.

4.1.7 Streamlit:

Streamlit 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.

4.1.8 Plotly

Plotly provides online graphing, analytics, and statistics tools for individuals and collaboration, as well as scientific graphing libraries for Python, R, MATLAB, Perl, Julia, Arduino, and REST.

4.1.9 Sentimental Analysis

Sentiment analysis is the process of classifying whether a block of text is positive, negative, or neutral. Sentiment analysis is a word context mining that shows a brand's social sentiment and helps determine if the products it manufactures are in demand in the market.

4.1.10 Flair (Sentimental Analysis Pretrained Model)

Flair is a lightweight, open-sourced natural language processing (NLP) library developed by Zalando Research. The Flair framework is built directly on top of one of the best deep learning frameworks, PyTorch.

4.2 Tools Used

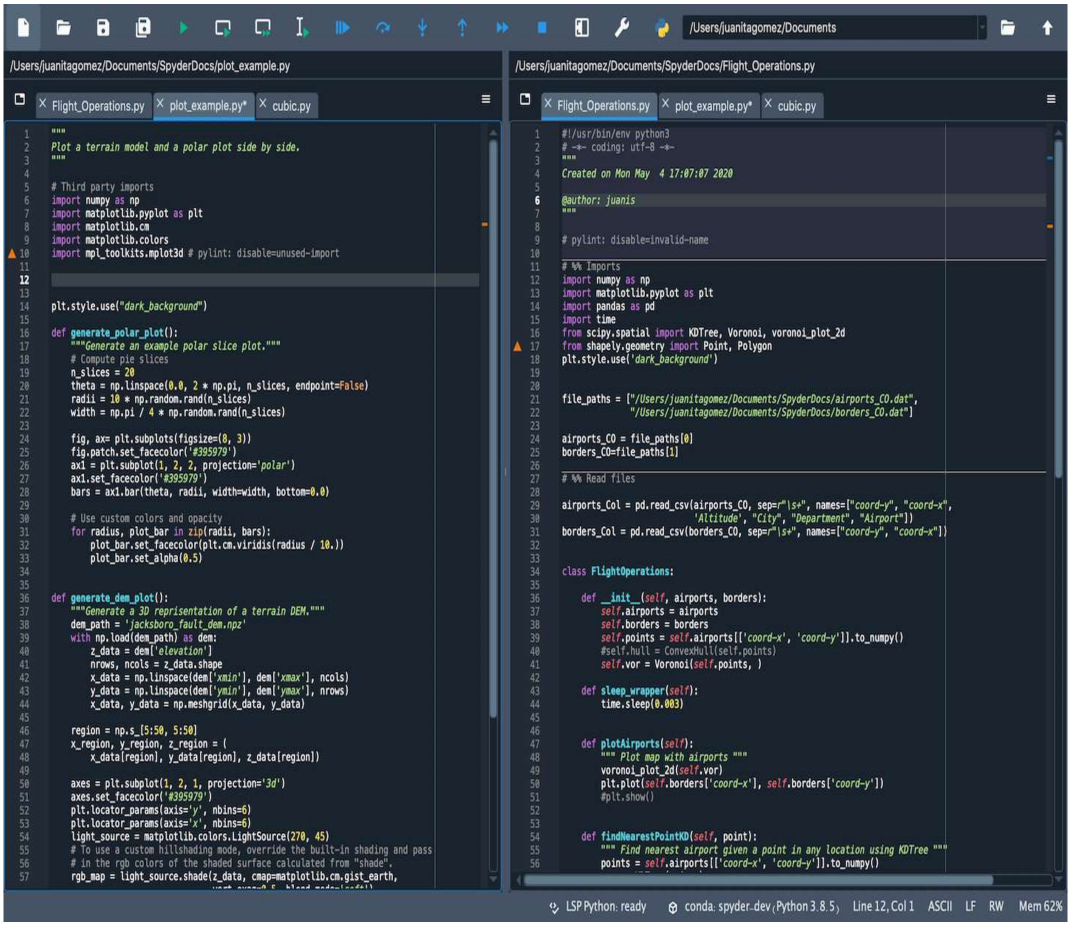
4.2.1 Anaconda:

Anaconda is a distribution of the Python and R programming languages for scientific computing that aims to simplify package management and deployment. The distribution includes data- science packages suitable for Windows, Linux, and macOS.

4.2.2 Spyder:

Spyder is an open-source cross-platform integrated development environment for scientific programming in the Python language.

Spyder's multi-language **Editor** integrates several powerful tools right out of the box for an easy to use, efficient editing experience. The Editor's key features include syntax highlighting (pygments); real-time code and style analysis (pyflakes and pycodestyle); on-demand completion, calltips and go-to-definition features (rope and jedi); a function/class browser, horizontal and vertical splitting, and much more.


/Users/juanitagomez/Documents
/Users/juanitagomez/Documents/SpyderDocs/plot_example.py
Flight_Operations.py
plot_example.py*
cubic.py
/Users/juanitagomez/Documents/SpyderDocs/Flight_Operations.py
LSP Python: ready
conda: spyder-dev (Python 3.8.5)
Line 12, Col 1
ASCII
LF
RW
Mem 62%

```python
"""
Plot a terrain model and a polar plot side by side.
"""

# Third party imports
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.cm
import matplotlib.colors
import mpl_toolkits.mplot3d # pylint: disable=unused-import



plt.style.use("dark_background")

def generate_polar_plot():
    """Generate an example polar slice plot."""
    # Compute pie slices
    n_slices = 20
    theta = np.linspace(0.0, 2 * np.pi, n_slices, endpoint=False)
    radii = 10 * np.random.rand(n_slices)
    width = np.pi / 4 * np.random.rand(n_slices)

    fig, ax= plt.subplots(figsize=(8, 3))
    fig.patch.set_facecolor('#395979')
    ax1 = plt.subplot(1, 2, 2, projection='polar')
    ax1.set_facecolor('#395979')
    bars = ax1.bar(theta, radii, width=width, bottom=0.0)

    # Use custom colors and opacity
    for radius, plot_bar in zip(radii, bars):
        plot_bar.set_facecolor(plt.cm.viridis(radius / 10.))
        plot_bar.set_alpha(0.5)


def generate_dem_plot():
    """Generate a 3D reprisentation of a terrain DEM."""
    dem_path = 'jacksboro_fault_dem.npz'
    with np.load(dem_path) as dem:
        z_data = dem['elevation']
        nrows, ncols = z_data.shape
        x_data = np.linspace(dem['xmin'], dem['xmax'], ncols)
        y_data = np.linspace(dem['ymin'], dem['ymax'], nrows)
        x_data, y_data = np.meshgrid(x_data, y_data)

    region = np.s_[5:50, 5:50]
    x_region, y_region, z_region = (
        x_data[region], y_data[region], z_data[region])

    axes = plt.subplot(1, 2, 1, projection='3d')
    axes.set_facecolor('#395979')
    plt.locator_params(axis='y', nbins=6)
    plt.locator_params(axis='x', nbins=6)
    light_source = matplotlib.colors.LightSource(270, 45)
    # To use a custom hillshading mode, override the built-in shading and pass
    # in the rgb colors of the shaded surface calculated from "shade".
    rgb_map = light_source.shade(z_data, cmap=matplotlib.cm.gist_earth,
```

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Mon May  4 17:07:07 2020

@author: juanis
"""

# pylint: disable=invalid-name

# %% Imports
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import time
from scipy.spatial import KDTree, Voronoi, voronoi_plot_2d
from shapely.geometry import Point, Polygon
plt.style.use('dark_background')


file_paths = ["/Users/juanitagomez/Documents/SpyderDocs/airports_CO.dat",
              "/Users/juanitagomez/Documents/SpyderDocs/borders_CO.dat"]

airports_CO = file_paths[0]
borders_CO=file_paths[1]

# %% Read files

airports_Col = pd.read_csv(airports_CO, sep=r"\s+", names=["coord-y", "coord-x",
                           'Altitude', "City", "Department", "Airport"])
borders_Col = pd.read_csv(borders_CO, sep=r"\s+", names=["coord-y", "coord-x"])


class FlightOperations:

    def __init__(self, airports, borders):
        self.airports = airports
        self.borders = borders
        self.points = self.airports[['coord-x', 'coord-y']].to_numpy()
        #self.hull = ConvexHull(self.points)
        self.vor = Voronoi(self.points, )

    def sleep_wrapper(self):
        time.sleep(0.003)


    def plotAirports(self):
        """ Plot map with airports """
        voronoi_plot_2d(self.vor)
        plt.plot(self.borders['coord-x'], self.borders['coord-y'])
        #plt.show()


    def findNearestPointKD(self, point):
        """ Find nearest airport given a point in any location using KDTree """
        points = self.airports[['coord-x', 'coord-y']].to_numpy()
```

Figure 4.1 Spyder Editor [16]

4.2.3 cmd:

CMD is the default command-line interpreter for the OS/2, eComStation, ArcaOS, Microsoft Windows, and ReactOS operating systems. The name refers to its executable filename.

4.3 Language Used

4.3.1 Python:

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

4.4 Screenshots

The Following are the screenshots of the result of the project:

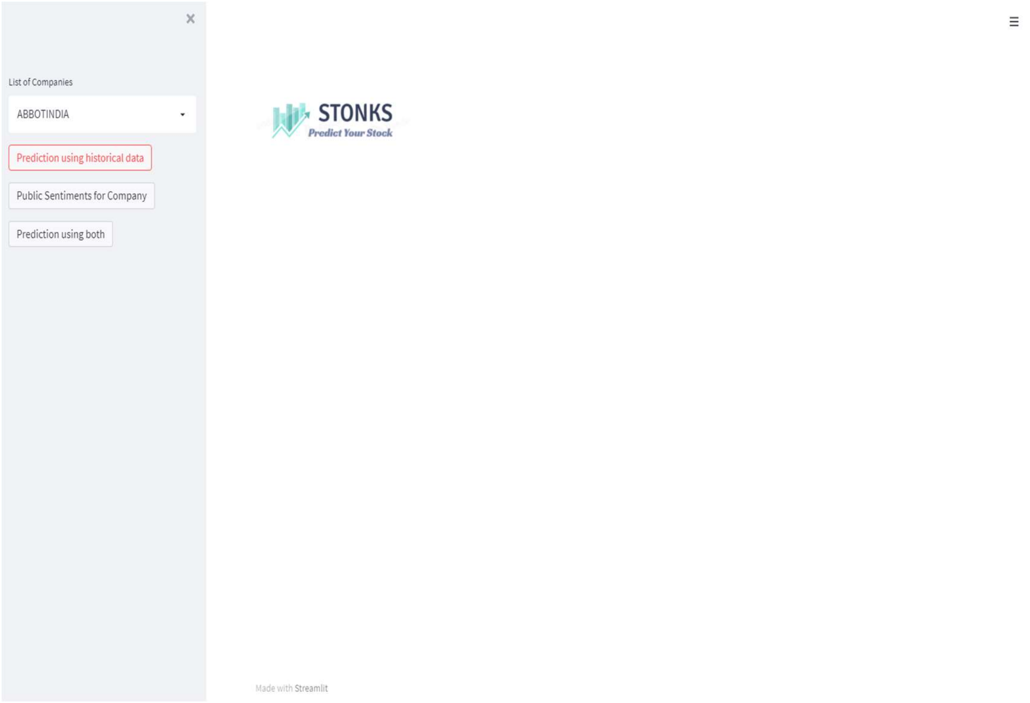
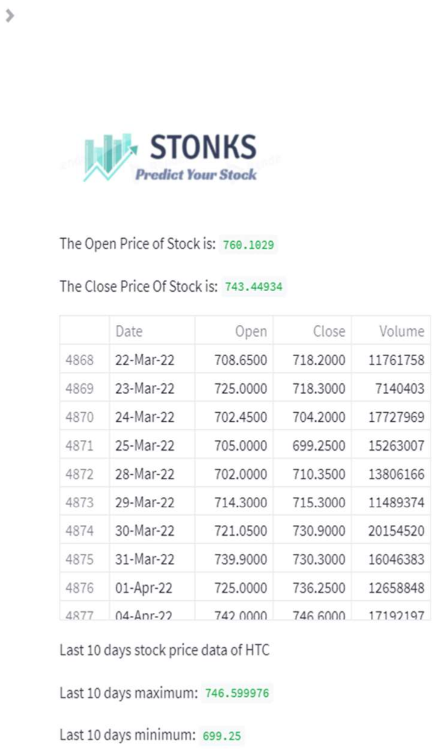


Figure4.2 Screenshot 1

This is the front page of our application where the user will land when user access the application. The above screenshot shows the side menu and the main display screen. The side menu contains the list of companies the user can select. There are 3 mode the user can choose. First mode is the prediction by past data. The user can predict the stock price from the historical data present. Second mode is sentimental Analysis from which user can analyze the recent public sentiments about the company selected. Third mode is the using both is both users can predict as well as analyze the sentiment.



The Open Price of Stock is: `760.1029`

The Close Price Of Stock is: `743.44934`

	Date	Open	Close	Volume
4868	22-Mar-22	708.6500	718.2000	11761758
4869	23-Mar-22	725.0000	718.3000	7140403
4870	24-Mar-22	702.4500	704.2000	17727969
4871	25-Mar-22	705.0000	699.2500	15263007
4872	28-Mar-22	702.0000	710.3500	13806166
4873	29-Mar-22	714.3000	715.3000	11489374
4874	30-Mar-22	721.0500	730.9000	20154520
4875	31-Mar-22	739.9000	730.3000	16046383
4876	01-Apr-22	725.0000	736.2500	12658848
4877	04-Apr-22	742.0000	746.6000	17192197

Last 10 days stock price data of HTC

Last 10 days maximum: `746.599976`

Last 10 days minimum: `699.25`

Figure4.3 Screenshot 2

The above figure 4.3 shows the last 10 days data of the chosen company. User can analyze the trend of last 10 days of the stock price of the company. This picture also shows the predicted open and close price of the selected company using the TSA algorithm and LSTM neural network. The table contains the date, Close price, Open price and the volume of the stock.The Figure also shows the last 10 days highest and lowest stock price of the company.

Open graph of ONGC :

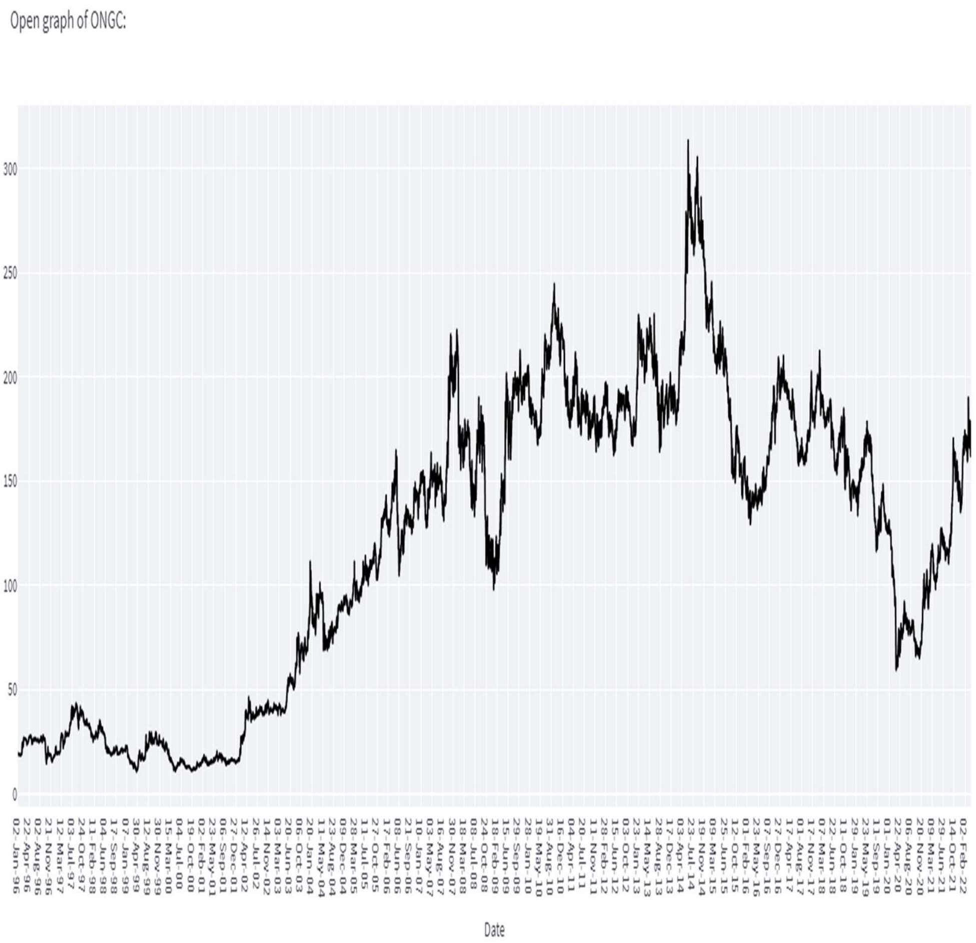


Figure4.4 Screenshot 3

This picture shows the Open Price Graph of the selected company.

The above figure 4.4 shows the Open Stock price graph of the user selected company. On X-axis of the graph shows the date and Y-axis of the graph shows the Open price of the company's stock. The highest point shows the highest price the company stock hit. The lowest point shows the lowest price of the company stock the stock achieved.

Close graph of ONGC :

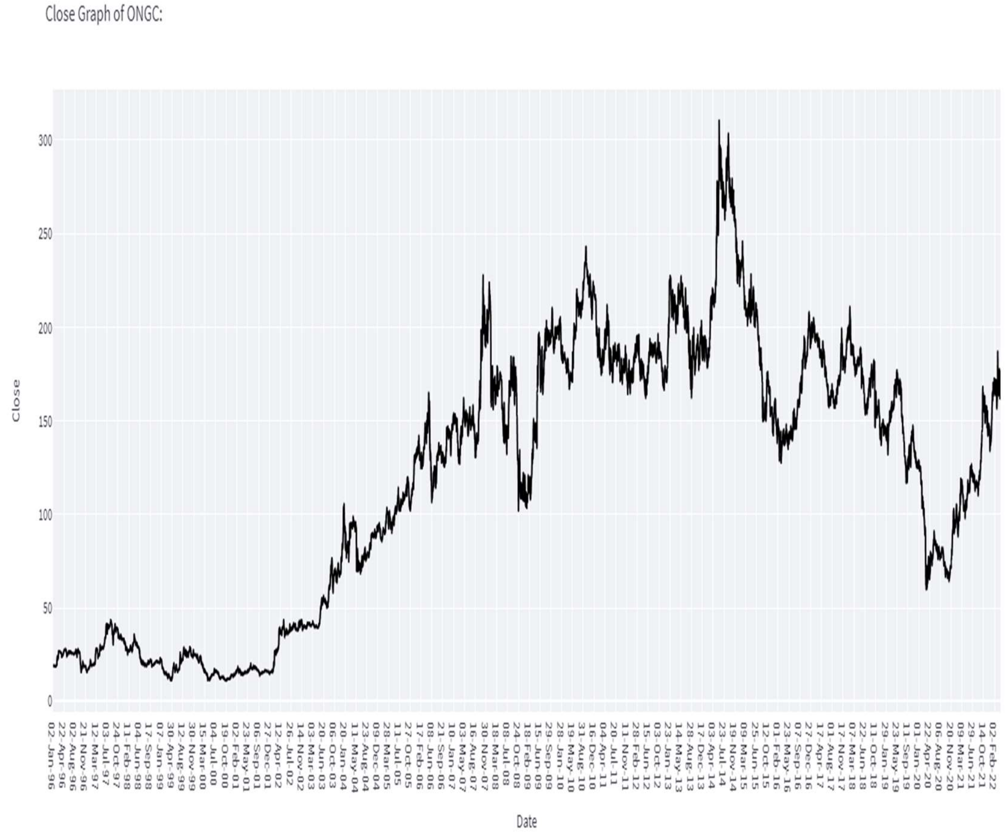
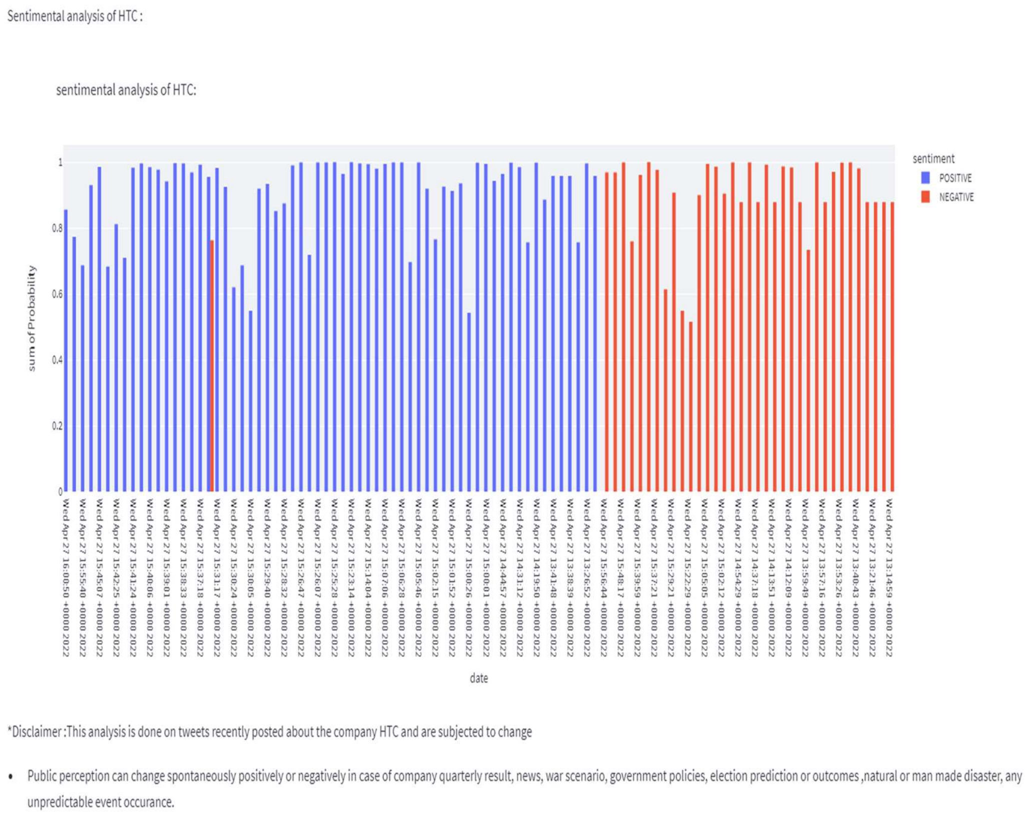


Figure4.5 Screenshot 4

This picture shows the Close Price Graph of the selected company.

The above figure 4.5 shows the Close Stock price graph of the user selected company. On X-axis of the graph shows the date and Y-axis of the graph shows the Close price of the company's stock. The highest point shows the highest price of the company stock. The lowest point shows the lowest price of the company share.



Sentimental analysis of HTC :

sentimental analysis of HTC:
sentiment
🟦 POSITIVE
🟥 NEGATIVE
sum of Probability
1
0.8
0.6
0.4
0.2
0
date
Wed Apr 27 16:00:50 +0000 2022
Wed Apr 27 15:55:40 +0000 2022
Wed Apr 27 15:45:07 +0000 2022
Wed Apr 27 15:42:25 +0000 2022
Wed Apr 27 15:41:24 +0000 2022
Wed Apr 27 15:40:06 +0000 2022
Wed Apr 27 15:39:01 +0000 2022
Wed Apr 27 15:38:33 +0000 2022
Wed Apr 27 15:37:18 +0000 2022
Wed Apr 27 15:31:17 +0000 2022
Wed Apr 27 15:30:24 +0000 2022
Wed Apr 27 15:30:05 +0000 2022
Wed Apr 27 15:29:40 +0000 2022
Wed Apr 27 15:28:32 +0000 2022
Wed Apr 27 15:26:47 +0000 2022
Wed Apr 27 15:26:07 +0000 2022
Wed Apr 27 15:25:28 +0000 2022
Wed Apr 27 15:23:14 +0000 2022
Wed Apr 27 15:14:04 +0000 2022
Wed Apr 27 15:07:06 +0000 2022
Wed Apr 27 15:06:28 +0000 2022
Wed Apr 27 15:05:46 +0000 2022
Wed Apr 27 15:02:15 +0000 2022
Wed Apr 27 15:01:52 +0000 2022
Wed Apr 27 15:00:26 +0000 2022
Wed Apr 27 15:00:01 +0000 2022
Wed Apr 27 14:44:57 +0000 2022
Wed Apr 27 14:31:12 +0000 2022
Wed Apr 27 14:19:50 +0000 2022
Wed Apr 27 13:41:48 +0000 2022
Wed Apr 27 13:38:39 +0000 2022
Wed Apr 27 13:26:52 +0000 2022
Wed Apr 27 15:56:44 +0000 2022
Wed Apr 27 15:48:17 +0000 2022
Wed Apr 27 15:39:59 +0000 2022
Wed Apr 27 15:37:21 +0000 2022
Wed Apr 27 15:29:21 +0000 2022
Wed Apr 27 15:22:29 +0000 2022
Wed Apr 27 15:05:05 +0000 2022
Wed Apr 27 15:02:12 +0000 2022
Wed Apr 27 14:54:29 +0000 2022
Wed Apr 27 14:37:18 +0000 2022
Wed Apr 27 14:13:51 +0000 2022
Wed Apr 27 14:12:09 +0000 2022
Wed Apr 27 13:59:49 +0000 2022
Wed Apr 27 13:57:16 +0000 2022
Wed Apr 27 13:53:26 +0000 2022
Wed Apr 27 13:40:43 +0000 2022
Wed Apr 27 13:21:46 +0000 2022
Wed Apr 27 13:14:59 +0000 2022

*Disclaimer :This analysis is done on tweets recently posted about the company HTC and are subjected to change

- Public perception can change spontaneously positively or negatively in case of company quarterly result, news, war scenario, government policies, election prediction or outcomes ,natural or man made disaster, any unpredictable event occurance.

Figure4.6 Screenshot 5

This Figure shows the sentimental analysis of the selected company.

The graph is about the sentiment of the company available on social media platform (Twitter). On X-axis the date and time of tweet is displayed and on Y-axis the Probability of the tweet is displayed and color is based on the behavior of the tweet. The graph is a bar graph, and every bar shows the character and its probability of the tweet.

4.1 Testing

Testing is the process of evaluation of a system to detect differences between given input and expected output and also to assess the features of the system. Testing assesses the quality of the product. It is a process that is done during the development process.

4.1.1 Strategy Used

Tests can be conducted based on two approaches –

Functional Testing

Software testing is a method used to verify the functionality of a software application. This can be done by checking whether the function is working as expected. In functional testing, we systematically give values to functions and check that their outputs match our expectations. Functional testing is a black box testing technique that confirms that an application or system behaves as we expect. The application has been tested and is functioning as intended.

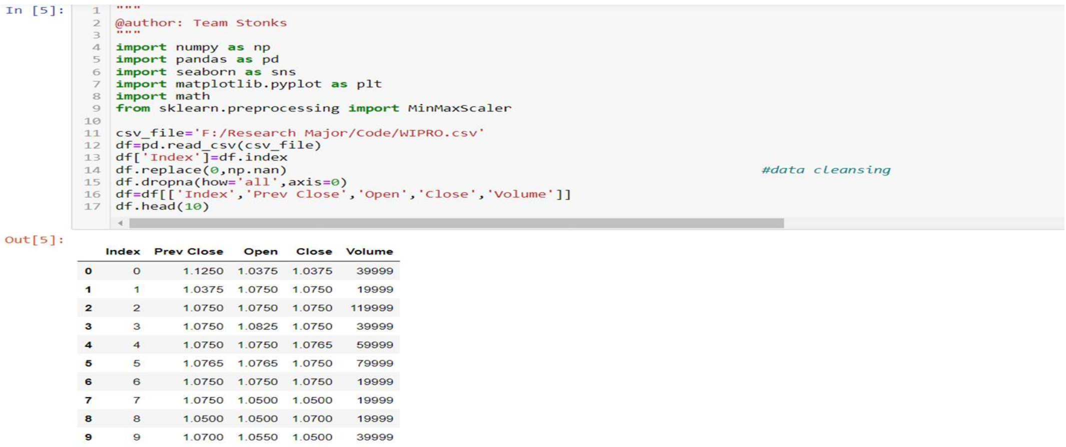
Functional testing is a critical part of software development and can help ensure that the application functions as intended. The tester is only testing the program, not the system.

Implementation Testing

Implementation is the process of creating an action for the formulated plan. Before we start implementing, the plan must be completed, and our goals must be clear.

The texting method used here is Black Box Testing. It is carried out to test functionality of the program. It is also called 'Behavioral' testing. The tester in this case, has a set of input values and respective desired results. On providing input, if the output matches with the desired results, the program is tested 'ok', and problematic otherwise.

We have tested the model on ITC.csv file. We have tested our model for the prediction and using the R2 score we tested the accuracy of the model. We have also built a graph between actual value and predicted value.

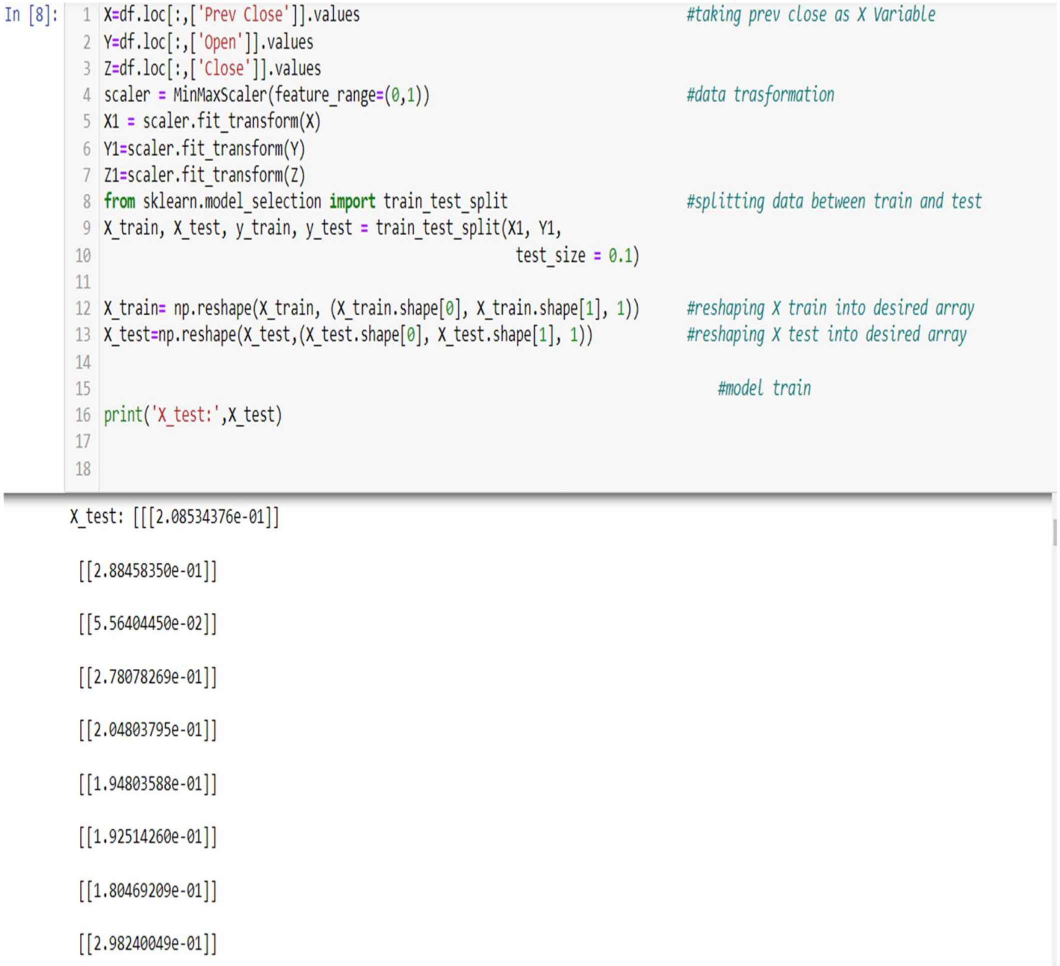


```
"""
@author: Team Stonks
"""
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import math
from sklearn.preprocessing import MinMaxScaler

csv_file='F:/Research Major/Code/WIPRO.csv'
df=pd.read_csv(csv_file)
df['Index']=df.index
df.replace(0,np.nan)                                                  #data cleansing
df.dropna(how='all',axis=0)
df=df[['Index','Prev Close','Open','Close','Volume']]
df.head(10)
```

```
   Index  Prev Close    Open   Close  Volume
0      0      1.1250  1.0375  1.0375   39999
1      1      1.0375  1.0750  1.0750   19999
2      2      1.0750  1.0750  1.0750  119999
3      3      1.0750  1.0825  1.0750   39999
4      4      1.0750  1.0750  1.0765   59999
5      5      1.0765  1.0765  1.0750   79999
6      6      1.0750  1.0750  1.0750   19999
7      7      1.0750  1.0500  1.0500   19999
8      8      1.0500  1.0500  1.0700   19999
9      9      1.0700  1.0550  1.0500   39999
```

Figure4.7 Screenshot 6

The above figure 4.7 shows the test case of the importing of various python libraries used to create the prediction model for the application. In the module we read the csv file containing the data of the company using pandas read_csv function. This converts the csv file to a dataframe. The pandas is also used to display the cleaned data.

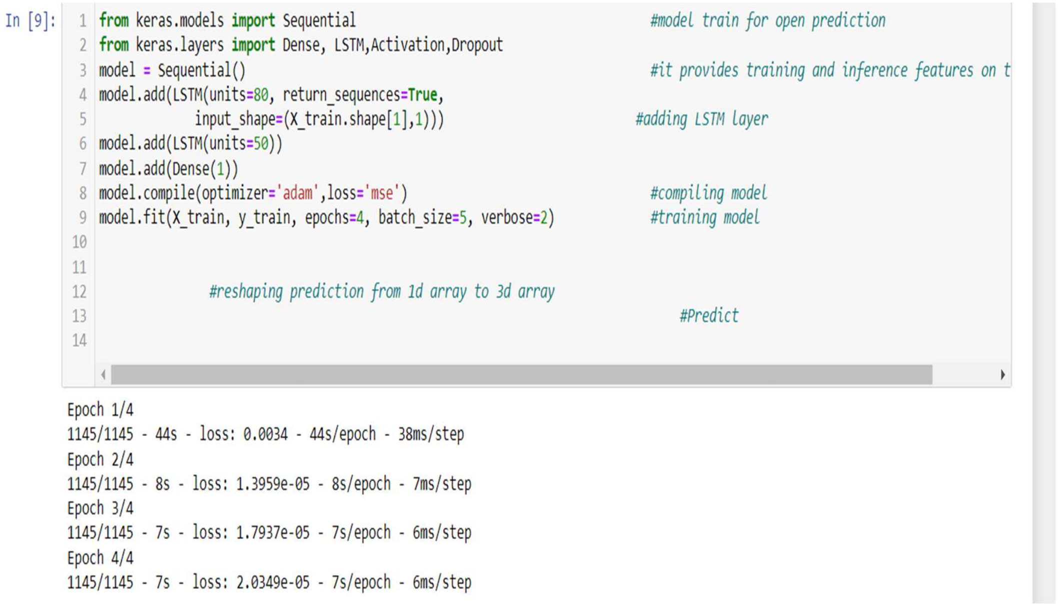


```
In [8]:
X=df.loc[:,['Prev Close']].values                                      #taking prev close as X Variable
Y=df.loc[:,['Open']].values
Z=df.loc[:,['Close']].values
scaler = MinMaxScaler(feature_range=(0,1))                             #data trasformation
X1 = scaler.fit_transform(X)
Y1=scaler.fit_transform(Y)
Z1=scaler.fit_transform(Z)
from sklearn.model_selection import train_test_split                   #splitting data between train and test
X_train, X_test, y_train, y_test = train_test_split(X1, Y1,
                                                    test_size = 0.1)

X_train= np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))    #reshaping X train into desired array
X_test=np.reshape(X_test,(X_test.shape[0], X_test.shape[1], 1))          #reshaping X test into desired array

                                                                          #model train
print('X_test:',X_test)
```

```
X_test: [[[2.08534376e-01]]

 [[2.88458350e-01]]

 [[5.56404450e-02]]

 [[2.78078269e-01]]

 [[2.04803795e-01]]

 [[1.94803588e-01]]

 [[1.92514260e-01]]

 [[1.80469209e-01]]

 [[2.98240049e-01]]
```

Figure4.8 Screenshot 7

The above figure 4.8 shows the data transformation into required form. Pandas, NumPy, Sklearn libraries are used for the data transformation. First data is selected and stored in various variables using pandas. Sklearn feature MinMaxscaler is used to transform each data into a range of 0 to 1. The data then is split into test data and train data in ration of 1:9.



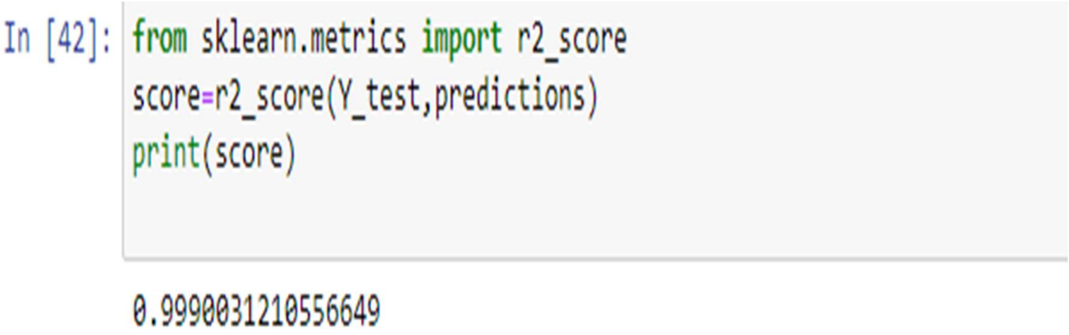
```
In [9]:
from keras.models import Sequential                                  #model train for open prediction
from keras.layers import Dense, LSTM,Activation,Dropout
model = Sequential()                                                 #it provides training and inference features on t
model.add(LSTM(units=80, return_sequences=True,
               input_shape=(X_train.shape[1],1)))                  #adding LSTM layer
model.add(LSTM(units=50))
model.add(Dense(1))
model.compile(optimizer='adam',loss='mse')                           #compiling model
model.fit(X_train, y_train, epochs=4, batch_size=5, verbose=2)       #training model


                #reshaping prediction from 1d array to 3d array
                                                                       #Predict

```

```
Epoch 1/4
1145/1145 - 44s - loss: 0.0034 - 44s/epoch - 38ms/step
Epoch 2/4
1145/1145 - 8s - loss: 1.3959e-05 - 8s/epoch - 7ms/step
Epoch 3/4
1145/1145 - 7s - loss: 1.7937e-05 - 7s/epoch - 6ms/step
Epoch 4/4
1145/1145 - 7s - loss: 2.0349e-05 - 7s/epoch - 6ms/step
```

Figure4.9 Screenshot 8

The above figure 4.9 shows the testing of model train. The model is trained on training data containing 90% of the total data. The model is prepared using keras library of python.

The model runs 4 times with batch size of 5. The optimizer for the model is adam which is the top optimizer used today. MSE or mean squared error for calculating loss in the model.



```
In [42]:
from sklearn.metrics import r2_score
score=r2_score(Y_test,predictions)
print(score)
```

```
0.9990031210556649
```

Figure4.10 Screenshot 9

This picture shows the R2 score of the model.

The above Figure 4.10 shows the comparison of the predicted value and the actual value and the total accuracy of the model. For this we used the sklearn library of the python. The R2 score is used to measure the accuracy of the model we developed.

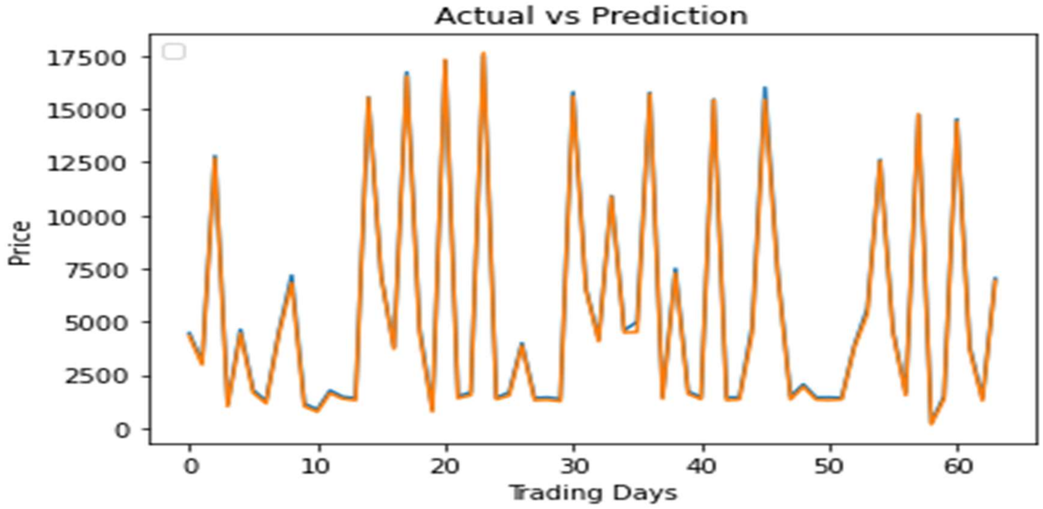


Figure4.11 Actual vs Predicted values

The above graph shows the testing of the predicted value verses the actual value of the company using graph. The graph is developed using the seaborn library of the python created upon the matplotlib library of the python. The above graph justifies the R2 score we calculated in the figure 4.8.

Chapter 5. Conclusion

Conclusion

5.1 Conclusion

We suggested a more accurate model that employs RNN and LSTM to forecast the trend in stock prices and uses sentimental analysis to extract the perception of people about a company. In the buried layer of the network, LSTM introduces the memory cell, a computational unit that substitutes typical artificial neurons. The accuracy of prediction is improved in this work by increasing the Epochs and batch size. We use test data to forecast in the proposed method, and the predicted outcomes are more accurate than the test data. The proposed method can track and anticipate the stock market, with the prediction yielding greater and more accurate outcomes. We are getting accurate findings in our above model, which will be more valuable to stock analysts, business analysts, and other professionals.

5.2 Limitations of the Work

The limitations of the project are below:

- The training of data is slow.
- Software excludes the minor factors (unpredictable events) that may impact the outcome.
- The application cannot predict on real-time.

5.3 Suggestion and Recommendations for Future Work

- Making a generalized app**.**
- Adding content validation.
- Automatically adding the historical data.
- Adding all the related news links.
- Adding the prediction of gold, silver etc.
- We can change domain to SIP, mutual fund.
- We can collaborate with investing application to direct user to their website directly after the prediction.

Bibliography

[1]. Shriram.S, Dr. K. Anuradha, Dr. K.P. Uma. “Future Stock Price using Recurrent Neural Network, LSTM and Machine Learning”, 2021

[2]. Usmani S, Shamsi JA. “News sensitive stock market prediction: literature review and suggestions”, 2021

[3]. Dev Shah, Haruna Isah and Farhana Zulkerine. “Stock market Analysis: A Review and Taxonomy of Prediction Techniques”, 2019

[4]. K. Hiba Sadia, Aditya Sharma, Adarrsh Paul, Sarmishtha Padhi, Saurav Sanyal. “Stock Market Prediction Using Machine Learning Algorithms”, 2019.

[5]. Prit Modi, Shaival Shah, Himani Shah. “Big Data Analysis in Stock Market Prediction”, IJERT 2019.

[6]. Yanmin Zhu, Yanyan Shen, Hao Li. “Stock Price Prediction Using Attention-based Multi-Input”, ACML 2018.

[7]. Ashish Sharma, Dinesh Bhuriya, Upendra Singh. "Survey of Stock Market Prediction Using Machine Learning Approach", ICECA 2017.

[8]. Kordonis, John &amp; Symeonidis, Symeon & Arampatzis, Avi. “Stock Price Forecasting via Sentiment Analysis on Twitter”, 2016

[9]. Kalyani Joshi, Bharthi H.N, Jyothi Rao. “Stock Trend Prediction using News Sentiment Analysis”, 2016

[10]. Anshul Mittal, Arpit Goel. “Stock Prediction using Twitter Sentimental Analysis”, 2011

[11]. Available at
https://www.analyticsvidhya.com/blog/2021/10/machine-learning-model-deployment-using-streamlit/

[12]. Available at https://github.com/Rajat-dhyani/Stock-Price-Predictor

[13]. Available at https://medium.com/codex/stock-price-prediction-a-modified

[14]. Available at https://www.moneycontrol.com/news/trends

[15]. Available at https://finance.yahoo.com/

[16]. Available at http://docs.spyder-ide.org/current/panes/editor.html

Appendix A

Project Plan

Gantt Chart

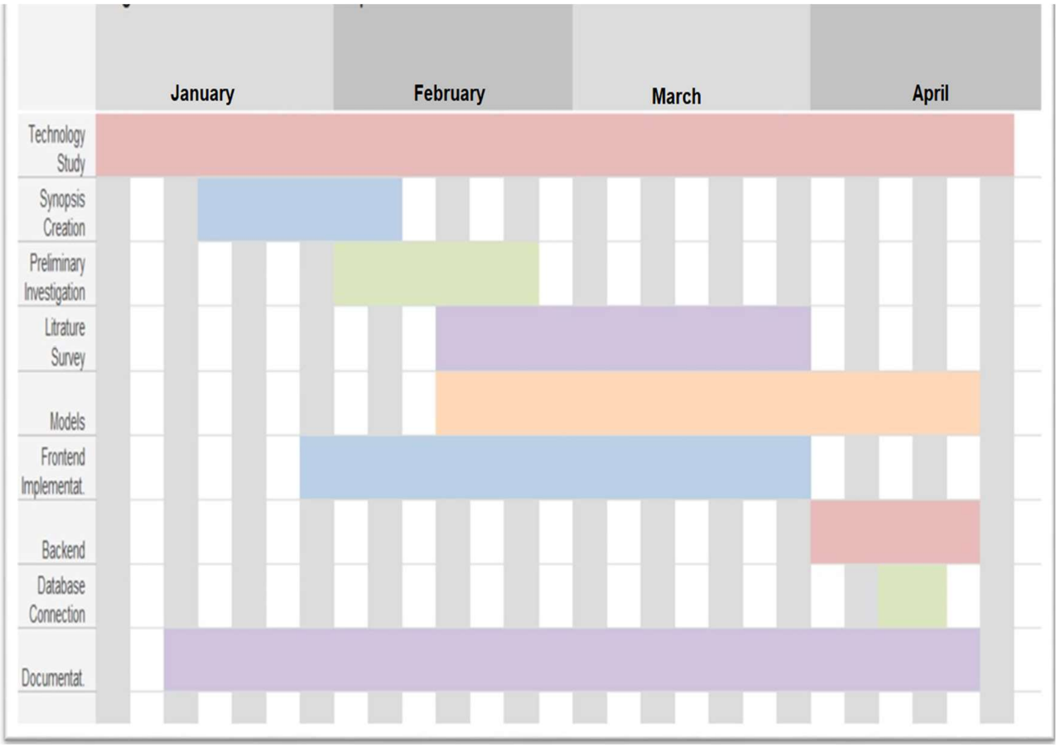


Figure5.1 Gantt Chart

This picture represents the timeline of work done during the entire duration of project.

Appendix B

Guide Interaction Sheet

Dates	Discussion
23/02/2022	Discussed about the project and how to proceed
28/02/2022	Discussed about the diagrams and implementation
10/03/2022	Discussed about the documentation and frontend implementation
16/03/2022	Discussed about the presentation and diagrams, poster and other technical things
22/03/2022	Discussed about the documentation and got it approved after making suggested changes
29/03/2022	Discussed about the GUI and got it approved
04/04/2022	Discussion to show complete implementation of project and made changes suggested
16/04/2022	Research paper discussion and Final approval of project

Appendix C

SOURCE CODE

Stonks.py

```
"""
@author: Team Stonks
"""
import streamlit as st
from Stonks_Complete import *
import numpy as np
import pandas as pd
import plotly.graph_objects as go
import matplotlib.pyplot as plt
import plotly.express as px
import math
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, LSTM,Activation,Dropout
import base64
from sentimental_analysis import *

def lineplot(df,value,title):                                 #function for creating graph
  fig = px.line(                                            #initiate line graph
    df,                                                    #Data Frame
    x = df['Date'],                                          #Columns from the data frame
    y = df[value],
    title=title                                              #csv file Name

  )
  fig.update_layout(                                          #graph layout
      width=1400,
      height=650,
      paper_bgcolor="white",)
  fig.update_traces(line_color = "Coral")                       #display text color on graph
  st.plotly_chart(fig)                                        #to plot graph

def sentiplot(df,title):
  fig = px.histogram(
    df,                                                    #Data Frame
    x = df['date'],                                          #Columns from the data frame
    y = df['Probability'],
    color=df['sentiment'],
    title=title

  )
  fig.update_layout(                                          #graph layout
      width=1400,
      height=650,
      paper_bgcolor="white",)

  fig.update_layout(barmode='group')                             #to transform bar graph into
readable form
  st.plotly_chart(fig)

def main():

  st.set_page_config('STONKS','LOGO_stonks.png','wide')          # front end elements of the
web page
                                              #for stonks logo
  main_bg = "OIP.webp"                                        #background page
  main_bg_ext = "webp"                                        #file extension



  st.markdown(                                                #html script for background
image
  f"""
  <style>
     .reportview-container{{
                 background-image:url(./stonk.jpg)
     }}
     }}
  </style>
  """,
  unsafe_allow_html=True
  )

  st.image('stonks_logo.png',width=250)                          #logo file

  menu = ['CIPLA',                                           #companies names for prediction
'GAIL','BPCL','COALINDIA','HCLTECH','ITC','ONGC',
'TATAMOTORS','HINDUNILVR','HDFCBANK','CENTRALBK','ADANIPORTS','ADANIPOWER','BANKB
ARODA','ABBOTINDIA',
'WIPRO','PERSISTENT','UCOBANK','AXISBANK','KOTAKBANK','SBIN','NTPC','HTC','BANKINDIA'
]
  menu.sort()                                                   #Sort in alphabetical order
  choice = st.sidebar.selectbox("List of Companies",menu)

  if st.sidebar.button("Prediction using historical data"):              #Only Historical data
                                                                         #prediction

    Open,Close=Close_predict(choice+'.csv')                          #initializing prediction module

    st.write("The Open Price of Stock is:",Open[0])
    st.write("The Close Price Of Stock is:",Close[0])

    df=pd.read_csv(choice+'.csv')
    df.replace(0,np.nan)
    df.dropna(how='all',axis=0)
    df=df[['Date','Open','Close','Volume',]]
    st.dataframe(df.tail(10))                                       #print recent 10 values from csv
    st.write(" ")
    st.write("Open graph of ",choice,':')
    open_title="Open graph of "+choice+':'
    lineplot(df,'Open',open_title)
    st.write(" ")
    st.write("Close graph of ",choice,':')
    close_title='Close Graph of '+choice+':'

    lineplot(df,'Close',close_title)

  if st.sidebar.button("Public Sentiments for Company"):                #Only sentimental Analysis
    st.write("Sentimental analysis of ",choice,':')

    analysis_title='sentimental analysis of '+choice+':'

    DF=Senti_analyze(choice)                               #initializing sentimental analysis   module

    sentiplot(DF, analysis_title)

  if st.sidebar.button("Prediction using both"):                        #Prediction and Analysis both
```

```
        Open,Close=Close_predict(choice+'.csv')

        st.write("The Open Price of Stock is:",Open[0])
        st.write("The Close Price Of Stock is:",Close[0])

        df=pd.read_csv(choice+'.csv')
        df.replace(0,np.nan)
        df.dropna(how='all',axis=0)
        df=df[['Date','Open','Close','Volume',]]
        st.dataframe(df.tail(10))                              #print recent 10 values from csv
        st.write(" ")
        st.write("Open graph of ",choice,':')
        open_title="Open graph of "+choice+':'
        lineplot(df,'Open',open_title)
        st.write(" ")
        st.write("Close graph of ",choice,':')
        close_title='Close Graph of '+choice+':'

        lineplot(df,'Close',close_title)

        st.write("Sentimental analysis of ",choice,':')

        analysis_title='sentimental analysis of '+choice+':'

        DF=Senti_analyze(choice)                               #Sentimental analysis module

        sentiplot(DF, analysis_title)



if __name__=='__main__':
    main()
```

Stonks_open.py

```
"""
@author: Team Stonks
"""
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import math
from sklearn.preprocessing import MinMaxScaler

                                                              #data preprocess
def process_data(csv_file):                                       #read csv
  df=pd.read_csv(csv_file)
  df['Index']=df.index
  df.replace(0,np.nan)                                            #data cleansing
  df.dropna(how='all',axis=0)
  df=df[['Index','Prev Close','Open','Close','Volume']]



  X=df.loc[:,['Prev Close']].values                               #taking prev close as X Variable
  Y=df.loc[:,['Open']].values
  Z=df.loc[:,['Close']].values
  scaler = MinMaxScaler(feature_range=(0,1))                        #data transformation
  X1 = scaler.fit_transform(X)
  Y1=scaler.fit_transform(Y)
  Z1=scaler.fit_transform(Z)
  from sklearn.model_selection import train_test_split                #splitting data between train
and test
  X_train, X_test, y_train, y_test = train_test_split(X1, Y1,
                              test_size = 0.1)
  X_train= np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))
                                                              #reshaping X train into desired array
  X_test=np.reshape(X_test,(X_test.shape[0], X_test.shape[1], 1))     #reshaping X test into desired
array

                                            #model train
  from keras.models import Sequential                                #model train for open
prediction
  from keras.layers import Dense, LSTM,Activation,Dropout
  model = Sequential()                                              #it provides training and inference
features on this model
  model.add(LSTM(units=80, return_sequences=True,
          input_shape=(X_train.shape[1],1)))                          #adding LSTM layer
  model.add(LSTM(units=50))
  model.add(Dense(1))

  model.compile(optimizer='adam',loss='mse')                            #compiling model
  model.fit(X_train, y_train, epochs=4, batch_size=5, verbose=2)        #training model



  Value=Z1[-1]                                                     #taking previous close for prediction

  Value=np.reshape(Value,(Value.shape[0], Value.shape[0], 1))          #reshaping prediction from
1d array to 3d array
                                                                 #Predict
  predictions = model.predict(X_test)                                 #prediction value of X test
  predictions = scaler.inverse_transform(predictions)
  Y_test=scaler.inverse_transform(y_test)

  v=model.predict(Value)
  value = scaler.inverse_transform(v)

  return v,value,predictions
```

Stonks_Complete.py

```
"""
@author: Team Stonks
"""
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import math
from sklearn.preprocessing import MinMaxScaler
from stonks_open import *
from pathlib import PurePath
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
                                                            #data preprocess
def Close_predict(csv_file):                                   #reading csv file
   df=pd.read_csv(csv_file)
   df['Index']=df.index                                        #for creating index in dataframe
   df.replace(0,np.nan)                                        #for replacing nan values with 0
   df.dropna(how='all',axis=0)                                  #for dropping NA values from csv
   df=df[['Index','Open','Close','Volume']]                          #for extracting specific data



   X=df.loc[:,['Open']].values                                # x is open price - Independent variable
   Y=df.loc[:,['Close']].values                               # y is close price - Target Variable

   scaler = MinMaxScaler(feature_range=(0,1))                #data transformation in values between 0
to 1
   X1 = scaler.fit_transform(X)                           # transforming open value
   Y1=scaler.fit_transform(Y)                             #transforming close value

   from sklearn.model_selection import train_test_split              #data splitting in test and train
datasets
   X_train, X_test, y_train, y_test = train_test_split(X1, Y1,
                                  test_size = 0.1)
   X_train= np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))  #reshaping X train dataset
into
```

```
                                                                    # desired array
X_test=np.reshape(X_test,(X_test.shape[0], X_test.shape[1], 1))      #reshaping X test dataset
                                                                    #into desired array


print('Stonks_open running')
value,S_open,graph=process_data(csv_file)                      #initaillizing open prediction model


                                                                    #Model train
print('Stonks_Close running')
from keras.models import Sequential                                #close price model train
from keras.layers import Dense, LSTM,Activation,Dropout

model = Sequential()                                            #function for combining layers
                                                                    #of LSTM
model.add(LSTM(units=100, return_sequences=True,
        input_shape=(X_train.shape[1],1)))                        #LSTM Input Gate
model.add(LSTM(units=40))                                        #Hidden Layer
model.add(Dense(1))                                              #Dense Layer for adjusting bais
                                                                    #and weights

model.compile(optimizer='adam',loss='mse')                           #compiling model
model.fit(X_train, y_train, batch_size=5,epochs=4 ,verbose=2)        #model training dataset


                                                                     #Predict
predictions = model.predict(X_test)                                  #predicting on X test
predictions = scaler.inverse_transform(predictions)
Y_test=scaler.inverse_transform(y_test)
value=np.reshape(value,(value.shape[0], value.shape[0], 1))        #predicted open price of stock
v=model.predict(value)                                           #predict close price from open

S_close=scaler.inverse_transform(v)
return S_open[0],S_close[0]                                      #returning values
```

Sentimental_analysis.py

```
"""
@author: Team stonks
"""
import requests
import flair
import regex as re
import pandas as pd


def get_data(tweet):                                                #get data from twitter
  data = {
    'id': tweet['id_str'],                                           #extracting id of tweet
    'created_at': tweet['created_at'],                                 #timestamp of tweet
    'text': tweet['full_text']                                        #content of tweet
  }
  return data



def Senti_analyze(company):                                         #sentimental Anlysis
  params = {                                                       #parameter
  'q': company,                                                    #company name or acronym
  'tweet_mode': 'extended',                                          #to extract full tweet
  'lang': 'en',                                                    #language of tweet
  'count':'10000'                                                    #number of tweets to extract
}



BEARER_TOKEN='AAAAAAAAAAAAAAAAAAAAAH3BbAEAAAANhrInokh03CpxLwZNoSP1qfpQSc
%3Dn4eeAxqx51TqnbPu2w60mJ0B51tswZH9l6kX7Nn4phyO8DoLrp'


  response=requests.get(
    'https://api.twitter.com/1.1/search/tweets.json',                    #initiating request for
tweets
    params=params,
        headers={'authorization': 'Bearer '+BEARER_TOKEN}                    #For authorization
        )



    df = pd.DataFrame()                                                      #insert tweets to dataframe
    for tweet in response.json()['statuses']:                                  #Json key
        row = get_data(tweet)
        df = df.append(row, ignore_index=True)                        #append in dataframe  ignoring    indices

    sentiment_model = flair.models.TextClassifier.load('en-sentiment')       #initializing flair model
    tweets=df['text']                                                        #tweets in dataframe for screening

    probs = []                                                               #probablity
    sentiments = []                                                          #sentiment list



    for tweet in tweets.to_list():
        sentence = flair.data.Sentence(tweet)                                # make prediction
        sentiment_model.predict(sentence)                                    # extract sentiment prediction
        probs.append(sentence.labels[0].score)                               # numerical score 0-1
        sentiments.append(sentence.labels[0].value)                          # 'POSITIVE' or 'NEGATIVE'



    tweets['probability'] = probs                                            # add probability and
sentiment predictions to tweets dataframe
    tweets['sentiment'] = sentiments



    senti=pd.DataFrame()                                                     #dataframe
    senti['date']=df['created_at']                                           #inserting creation date

    senti['sentiment']=sentiments[0:]

    senti['Probability']=probs[0:]

    return senti
```