

STOCK MARKET PREDICTION USING MACHINE LEARNING MODELS & SENTIMENT ANALYSIS OF TWEETS

A Project Report

Submitted in partial fulfilment of the requirements

for the award of the Degree of

MASTER OF SCIENCE (INFORMATION TECHNOLOGY)

By

Pankaj Narayan Manachekar

Seat Number :- 4135581

Under the esteemed guidance of

Ms. Beena Kapadia

Assistant Professor, Department of Information Technology



DEPARTMENT OF INFORMATION TECHNOLOGY

VIDYALANKAR SCHOOL OF INFORMATION TECHNOLOGY

(Affiliated to University of Mumbai)

MUMBAI, 400 037

MAHARASHTRA

2022 - 2023

VIDYALANKAR SCHOOL OF INFORMATION TECHNOLOGY

(Affiliated to University of Mumbai)

MUMBAI-MAHARASHTRA-400037

DEPARTMENT OF INFORMATION TECHNOLOGY



CERTIFICATE

This is to certify that the project entitled , "**Stock Market Prediction Using Machine Learning Models & Sentiment Analysis of Tweets**", is bona fide work of **PANKAJ NARAYAN MANACHEKAR** bearing **Seat No :- 4135581** submitted in partial fulfilment of the requirements for the award of degree of MASTER OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

Internal Guide

Coordinator

Internal Examiner

External Examiner

Date:

College Seal

Principal

Research Paper

Stock Market Prediction Using Machine Learning Models & Sentiment Analysis Of Tweets

Mr. Chaitanya Vijay Mane

M.Sc. I.T.

Vidyalankar School Of Information

Technology, Wadala

manechaitanya224@gmail.com

Mr. Pankaj Narayan Manachekar

MSc. I.T.

Vidyalankar School Of Information

Technology, Wadala

pnkj.manchekar@gmail.com

Abstract :- Stock market forecasting is very important in the planning of business activities. Stock price prediction has attracted many researchers of various fields including information technology, statistics, business, finance, and various operations. In this paper we present a detail survey on stock price trends. All key terms and phases of generic stock prediction methodology along with challenges, are described. Various types of literature review cover many data pre-processing techniques, feature selection techniques, prediction techniques, and future directions is presented for news sensitive stock prediction. Different types of studies have been shown that the vast amount of online information in the public domain such as Wikipedia usage pattern, news stories from the mainstream media, and social media discussions can have an observable effect on investors' opinions towards financial markets. The development of a dictionary-based sentiment analysis model, and evaluation model for gauging various effects of news sentiments on stocks for the pharmaceutical market. Using only news sentiments, we achieved a directional accuracy of 70.59% in predicting the trends in short-term stock price movement. We believe that the reason the model is able to achieve this accuracy for this particular sector by researching and leveraging domain expertise. This Stock prediction is a challenging task as it requires deep insights for extraction of news events and stock price trends.

Keywords :- Stock Market Prediction, Sentiment Analysis, Feature Selection, Arima Model, LSTM Model, Twitter Analysis.

I. INTRODUCTION :-

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 the most common 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. Fundamental analysis is more realistic because it evaluates the market in a broader scope. This survey puts emphasis research based on fundamental analysis, where textual data is considered along with stock price historical data for stock trend prediction.

Twitter is the important and popular social media where anyone can post tweets about any event. This is an open platform where people may express their views/opinions or emotions freely. Due to less internet charges, less expensive portable devices and increase social importance; people have a twitter account. Most of them tweet on different events. In the social networking age people express their opinions and feelings through twitter. So twitter contains a huge amount of data.

Twitter sentiment analysis is one of recent and challenging research areas. As social media like twitter contains a huge amount of text sentiment data in the form of tweets it is useful to identify sentiments or opinions of people about specific events. Sentiment analysis or opinion mining is useful for review of movies, products, customer services, opinions about any event etc.

Dataset Example for stock prediction (Apple Dataset) :-

The paper deals with the development of a specific company's stock price. The aim of the paper is to use the prediction method for a detailed analysis and evaluation of the development of Apple Inc. stock prices. Daily data from 2000 to 2020. The data, from the period of 2000 - 2020 show a gradual increase in Apple's stock prices. The most common factor leading to the increase in stock prices is the launch of a new product or service on the global market.

II. LITERATURE REVIEW OF SENTIMENT ANALYSIS:-

This section briefly outlines the research on stock prediction techniques. It summarizes the techniques that only consider numerical financial data for stock prediction. Then it discusses the feature extraction from textual data.

Stock trend prediction using textual data.

Unstructured form of textual data makes the model difficult for extracting the data mining techniques to mine information from text. Moreover, these text mining techniques can be further classified as fact mining and opinion mining techniques. Literature regarding text processing techniques is discussed shallow feature based text processing, event extraction based text processing, and sentiment analysis.

Sentiment analysis based text processing.

Stock prediction of sentiment analysis is an attractive area of research as it gives deeper analysis of textual data. In [Sehgal & Song \(2007\)](#), Yahoo financial message board is used as a source of textual data for predicting stock trend. They inferred public sentiments from web messages and proved its correlation with stock trend. Naïve bayes, decision tree, and bagging algorithms are used as prediction algorithm. They also added an important contribution of trust value parameter. It is calculated using author's past performance related to correct predictions. It should have a trust value unreliable sentiments are filtered which further enhances prediction accuracy. It can be viewed by experiments in [Wu et al. \(2012\)](#) that the sentiment analysis based features along with technical indicators enhances the prediction performance. They used pointwise mutual information (PMI) measurement to extract sentiment analysis based features. PMI measures strength of semantic association between words and seed words from positive and negative class. But technical analysis can be logically improved by examining different combinations of technical indicators.

ML techniques for stock market prediction using numerical and textual data.

In literature, machine learning based stock market techniques are divided into shallow learning and deep learning techniques. Research papers of stocks are discussed in earlier section in the context of text mining approaches. In this section, these research papers are outlined under machine learning categories.

The given table below contains specific information about various researches that have carried through out the stocks using sentimental analysis.

REFERENCE	DATA SOURCE	Numerical Data	Textual Data	Algorithms Used	Accuracy
1. Ding et al. (2016)	S&P 500 through Yahoo Finance, News articles from Reuter's website from October 2006, to November 2013	Stock price data	Knowledge driven event embedding (KGEB)	KGEB-RNN	Accuracy = 66.93%
2.Vargas, De Lima & Evsukoff (2017)	S&P 750 index series .Yahoo Finance, News articles from Reuter's website from 20-10-2006, to 2-11-2013	Technical data	Word embedding and sentence embedding	Combines LSTM with RNN	Accuracy for word embedding and technical indicator = 61% Accuracy for sentence embedding and technical indicator = 62%
3.Deng et al. (2019)	DJIA index from 08/08/2008 to 01/01/2016. Stock price data from Yahoo Finance, news headlines from Reddit WorldNews Channel	Stock price data	Knowledge driven event embedding	Knowledge Driven Temporal	Accuracy = 71.8%
4.Jin, Yang & Liu (2019)	Apple stocks from Yahoo finance, Stock comment dataset from stocktwits	Stock price data	CNN as a base learner for sentiment index	EMD based enhance LSTM (EMD-LSTM) with attention layer	RMSE = 3.196534 MAPE = 1.65 MAE = 2.396121 R ₂ = 0.977388
5.Li, Wu & Wang (2020)	Hong Kong Exchange daily prices from January 2003 to March 2008, FINET news	Stock price Data & Technical data	News sentiment analysis using sentiment lexicon	LSTM	Test Accuracy for 3 out of 4 sectors is comparatively better using domain specific dictionary.

6. (Chan & Franklin, 2011)	Permits to predict financial trends with its justification	Stock price data	permits to predict financial trends with its justification	LSTM	Accuracy = 51.8%
7. Chen et al.(2019)	Tokyo Stock Price Index (TOPIX),	Stock price data	Proposed financial event dictionary	Structured Stock Prediction Model(SSPM), Multi-Task Structured Stock Prediction Model(MSSPM)	SSPM Accuracy = 66.4%
8. Chen et al. (2015)	Financial news from Reuters	Stock price data	fine grained event using dictionary	Using BiLSTM, self-attention and Conditional Random Fields (CRF) etc.	MSSPM Accuracy = 65.7
9. (Nassirtoussi et al., 2015)	News-headlines	Tweet Sentiment Analysis	FOREX market prediction - multi-layer dimension algorithm with semantics.	Tweepy	Positive : 50% Neutral : 60% Negative : 65%
10. (Schumaker & Chen, 2009)	Textual analysis of stocks	financial news - the AZF in text format	ACM Transactions of various Information Systems	LSTM	Accuracy = 78.8%
11. Pasupa K, Sunhem W. 2016	Application of support vector regression in indonesian stock price prediction	feature selection using various swarm optimisation	Modelling and Simulation	LSTM	Accuracy = 81.8%
12. Rustam Z, Kintandani P. 2019.	Financial distress prediction using svm	ensemble vs. individual	Applied Soft Computing	RNN	Accuracy = 88.3%
13. Schumaker RP, Chen H. 2009	A comparative form of machine learning, deep learning	a decade survey on the necessity	recent developments, and potential future directions	ARIMA	Accuracy = 41.6%
14. Sehgal V, Song C. 2007	Deep learning for stocks and various purposes	Prediction of financial news.	News - headlines categorization scheme for unlabelled data	ARIMA	Accuracy = 48.1%
15. Sun J, Li H. 2012	. A novel time-series model	based on empirical mode decomposition for forecasting TAIEX	Forecasting the given models	ARIMA	Accuracy = 93.3%

Finally, the table outlines state of the art techniques in the context of news sensitive stock prediction model. By observing this table, it can be deduced that hybrid approaches for feature extraction and prediction perform better by combining strengths of different approaches.

III. RESEARCH METHODOLOGY :-

The sources of papers for this study are IEEE Xplore AND Researchgate. Total 20 articles presented in this research paper which are related to sentiment analysis. The algorithms used for predicting purposes are mainly Long ShortTerm Memory (LSTM) & RNN.

Stock market prediction seems like a complicated problem because there are various factors that are still left unaddressed and do not seem to be statistical at first. But to our rescue there are various machine learning algorithms by using which we could efficiently predict current trends in the stock market by using thereferences from the previous data. Here the dataset that we are going to use has been collected from Yahoo finance.

This dataset consists of nearly 9,00,000 records related to the stock prices required and many other values that are relevant to each other. This data predicted the stock prices at some intervals of time for each day in a year. Many sections such as volume, date etc were included in it. Inorder to simulate and analyze only one company's data was taken into account.

The data considered or taken into account was readily available in the csv format which was first read and converted into a data frame by making use of one of the most popular libraries, Pandas in Python.

In the due course, one specific company's data was pulled out by separating data depending on the symbol field. After this, the data was segregated into testing and training data sets by performing normalization by using yet another popular Python library known as Sklearn library.

The test set was placed as 20 percent of the dataset that was available. Although Machine learning has various algorithms that could be used for predicting the stock prices so we can make use of two main algorithms known as RNN and LSTM.

IV. DISCUSSION, RESULTS AND ANALYSIS:

The sentiment analysis can be applied in many different domains. The major use of sentiment analysis is to analyze the texts which are available on the social networking sites, where people share their views on particular product or topic.

Companies use the sentiment analysis tools to analyze market, by analyzing the customer feedbacks. Now-a-days even politicians are appointing data analysts to analyze political party related texts available on the news reporting websites and also in social networking sites. Some major applications of sentiment analysis are as stated below,

1. In Review-Related Websites.
2. The sentiment analysis can be used Sub-Component Technology.
3. It can be used in Business and Government Intelligence.
4. Sentiment analysis can be across Different Domains.

Feature Selection :-

Feature selection is the process of eliminating the redundant set of features, while selecting only those features from the dataset, which are most useful or which are most relevant. The use of bigrams and trigrams, will result in a problem resulting in the increment in the features. Most common features are redundant and noisy in nature.

Twitter Analysis :-

Twitter data is also automatically classified into positive, negative and neutral according to query term used in consumer review tweets. In the paper author uses Parts Of Speech (POS) polarity technique and tree kernel technique.

Research work uses different types resources such as hand dictionary of emotions and dictionary collected from web. Author used different types classification and feature extraction algorithm.

ARIMA Model :-

This ARIMA model was introduced by Box and Jenkins in 1970. ARIMA models with time series data. The model is most important financial forecasting method. Models from ARIMA have been effective in generating short-term forecasts. The future value of variable in ARIMA model is linear combination of past values and past errors.

Long Short Term Memory (LSTM) Model :-

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.

Linear Regression Model :-

A linear regression model describes the relationship between a dependent variable, y , and one or more independent variables, X . The dependent variable is also called the response variable. Independent variables are also called explanatory or predictor variables.

RNN Model :-

A recurrent neural network (RNN) is a type of artificial neural network which uses sequential data or time series data. These deep learning algorithms are commonly used for ordinal or temporal problems, such as language translation, natural language processing (NLP), speech recognition, and image captioning; they are incorporated into popular applications such as Siri, voice search, and Google Translate.

Data Collection :-

The data collection from twitter, news headline and yahoo finance are collected for analysis.

Data Pre-processing :-

Stock data is extracted is not completely understandable because of public holidays and weekends where the stock market does not function. There are missing in the stock value. These empty values can be approximately using simple way. Consider, the stock values on a day is x and the next value present is y with some missing in between. So, the first value is estimated as $(x+y)/2$ and the same method is used to fill the missing values.

Classification :-

Use a bag of words containing information on sentiment (positive, negative, neutral) along with sentiment scores. After this, we adopt negation detection measures to differentiate between “good” and “not good”. In this blog we will be trying to do sentiment analysis on twitter dataset and categorizing them into positive, negative and neutral behavior of people. If the entire review has a positive, joyful attitude on if something is mentioned with positive connections. So, it is considered as a positive statement. If the entire comment has a negative, sad or if something mentioned with negative connections. So, it is considered as a negative statement. If the review expresses no personal opinion in the comments and reviews transmits information. After the feature extraction we perform sentiment analysis using naïve bayes classifier.

The obtained sentiment analysis data along with stock market data are combined and given as input to the training model. the stock market values are fetched using yahoo finance. The XG-Boost classifier evaluates both the data and predicts the stock market value.

Results :-

Sentiment Analysis – LSTM The accuracy of the model was 72% which is quite high enough and the output of this sentiment analysis will be combined to predict the stock price.

Stock Prediction Multiple Linear Regression The MSE and RMSE for train model was 2.664 and 1.632, the MSE and RMSE for the test model was little bit higher for 15.450 and 3.930. The graph difference of prediction and actual result, the difference just a little between actual and prediction
Comparison of Actual and Predict using Multi Linear Regression.

Stock Prediction -LSTM The MSE and RMSE for train model was 473.875 and 21.768, the MSE and RMSE for the test model was little bit higher for 74.181 and 8.612. The gap between actual and prediction is higher than using the Multi Linear Regression. Comparison Actual and Predict using LSTM.

Stock Prediction -ARIMA The MSE and RMSE for the ARIMA model was 24.770529 and 4.977. We can conclude that ARIMA forecast cannot predict when there is a high edge at the end of forecast.

Discussion :-

Based on our research, even though RMSE and MSE from ARIMA model is low enough, but for FREN stock that having a trend conditions, it cause the big difference of the actual and the prediction. This conclude that the theory limitation of ARIMA was approved on this research.

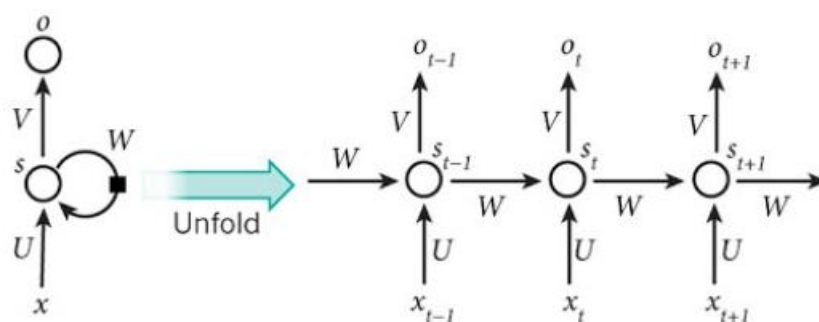
Even though the result of Multi Linear Regression and LSTM giving quite high accuracy than ARIMA, the prediction almost always close to the actual price but the prediction almost always lower than the actual, hopefully future works can improve with parameter tuning so the line of prediction is equal or higher than the actual price.

Comparative Result with LSTM and RNN Model :-

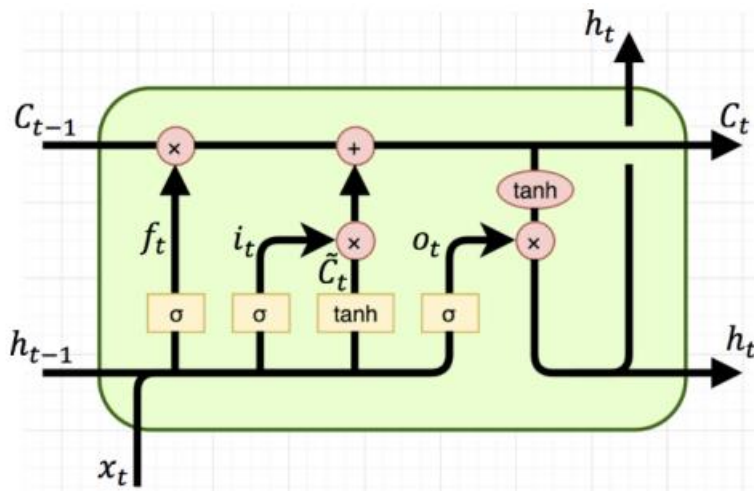
RNNs are a type of neural network that are designed to process sequential data, such as text, audio, or time series data. They can “remember” or store information from previous inputs, which allows them to use context and dependencies between time steps.

RNN can provide considerably good prediction for the temporal stock data. The hidden states of RNN are given by Equations (1) and (2) [32]. $S_t = \tanh(Wx_t + U S_{t-1} + b)$ (1) $o_t = c + V S_t$

where x_t is the input vector at time t ; b and c are bias values; W , U , and V denote inputto-hidden, hidden-to-hidden, and hidden-to-output weight matrices, respectively. While working with time-series data (like the stock market), an attention mechanism can be utilized that can divide the given data into parts so that decoder can utilize specific parts while generating new values. Figure 1 shows the generalized RNN architecture.



LSTMs are a type of RNN that use special type of memory cell and gates to store and output information. The gates in an LSTM network are controlled by sigmoid activation functions. These gates allow the network to selectively store or forget information. LSTMs are effective at storing and accessing long-term dependencies. They are slower to train and run than other types of RNNs.



Stock Prediction :-

This step uses 3 kind of model which is Multiple Linear Regression, LSTM and ARIMA.

The proposed workflow and conventional workflow in this study presented.

- I. Multiple Linear Regression. The configuration in this steps was splitting the data between train and test with 8:2 ratio to forecast the stock with variable stock, exchange rate and sentiment data.
- II. LSTM The configuration for LSTM model using several step by using input shape with 1 loop back, 1 dense for the LSTM model, by finding the loss of mean_squared_error with Adam optimizer with 20 epochs.
- III. ARIMA For the ARIMA model still using the Rapid Miner with $p=2, d=0, q=1$ due the rapid miner can optimize the processing better, the variable used only historical of low_prc only.

V. CONCLUSION :-

This paper provides a review and comparative analysis of different stock market prediction parameter techniques. These techniques are used to evaluate stock market performance and trends. The stock market forecasting system is to increase accuracy. In this study to analyze a novel approach to improve prediction of results of stock, it means we will combine two or more methods to construct a novel approach method.

Investors in the stock market are always searching for new techniques to outperform the stock market and make a good profit. Researchers around the world continue to conduct research in this area to meet the demand of the investors. In this research, we have tried to analyze the performance of different versions of recurrent neural networks (ARIMA , LR, RNN & LSTM) in time series analysis in the domain of stock price forecasting.

This paper presents an extensive study of stocks prediction using news and stock prices. It presents a generic approach to implement news sensitive stock prediction model and identifies three main phases. In each phase, challenges are identified and in search of opportunities existing literature is reviewed. This work has four major contributions. The first contribution is to provide literature review on this topic. This work elaborates existing research paper and assesses their strengths and limitations.

ARIMA performs better in terms of slightly fluctuating and highly fluctuating data, while LSTM produces better results when the pattern is not too flat or too sharp. From the results, we can say that both LSTM and ARIMA give better performance than RNN because RNN suffers from vanishing gradient problem and also has less control on the input and output. Investors should therefore try updated versions of RNN instead of traditional RNN to predict stock price movement. Investors can also have up to 5 days to figure out their buying and selling points.

In this paper we investigated how sentiment-analysis of the twitter data is correlated to predict the stock price for all the companies which are taken. Sentiment analysis is a very important aspect in data analytics and the advanced text processing. Each word in a given text is related to adjacent or any other related words. Merely by looking at the positive or negative words in a text, we cannot conclude with a sentiment which it results in. As of now we have seen that by implementing LSTM, ARIMA, Twitter analysis, feature selection, we have got highly improvement in the given sentiment classification.

VI. REFERENCES :-

- [1] <https://monkeylearn.com/blog/sentiment-analysis-machine-learning/>
- [2] <https://data-flair.training/blogs/python-sentiment-analysis/>
- [3] <https://www.geeksforgeeks.org/what-is-sentiment-analysis/>
- [4] M.Govindarajan, Sentiment Analysis Movie Reviews o f HybridMethod of Naive Bayes or Genetic Algorithm , *International International of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970) Volume-3 Number-4 Issue-13 December-2013*
- [5] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. The *Proceedings the Workshop on Language in Social Media (LSM 2011)*, pages 30–38, Portland, Oregon. Association for Computational Linguistics.
- [6] <https://cran.r-project.org/web/packages/> accessed 15-Feb-2016
- [7] Singh, Vivek & Piryani, Rajesh & Uddin, Ashraf & Waila, Pranav. (2013). Sentiment analysis reviews: A new feature-based heuristic for aspect-level sentiment classification. 712-717. 10.1109/iMac4s.2013.6526500
DOI:10.1109/iMac4s.2013.6526500
- [8] Pak, Alexander & Paroubek, Patrick. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. *Proceedings of LREC*. 10.
- [9] Sangeeta, Twitter Data Analysis Using FLUME & HIVE on Hadoop Frame Work Special Issue on International of the Recent Advances in Engineering & Technology (IJRAET) V-4 I-2 For National Conference , Recent Innovations in Science, Technology & Management (NCRISTM) ISSN (Online): 2347-2812, Gurgaon Institute of Technology and Management, Gurgaon 26th to 27th February 2016
- [10] Varsha Sahayak, Vijaya Shete, Apashabi Pathan, Sentiment Analysis of Twitter Data, *International Innovative Research* DOI:10.21817/ijet/2017/v9i3/1709030151
- [11] Citius: A Naive-Bayes Strategy for Sentiment Analysis - English Proceedings of 15th International , the Workshop on Semantic (SemEval 2014), pages 171–175, Dublin, Ireland, August 23-24, 2014
- [12] Hemalatha, Dr. G. P Saradhi Varma, Dr. A.Govardhan, "Sentiment Engineering, Indian Institute of Technology (BHU), Varanasi, India. ISSN ONLINE(2320-9801) PRINT (2320-9798)
- [13] <http://sentiwordnet.isti.cnr.it/>
- [14] Nassirtoussi AK, Aghabozorgi S, Wah TY, Ngo DCL. 2015. Text mining news-headlines of forexs market prediction - multi-layer dimension algorithm with semantics. *Expert Systems with various types of Applications* 42(1):306–324.
- [15] Patel Shah, Thakkar P. 2015. Predicting price index movement data preparation and machine learning techniques. *Expert Systems with the Applications* 42(1):259–268.
- [16] Picasso A, Merello S, Ma Y, Oneto L, Cambria E. 2019. Technical analysis and sentiment tweets of embeddings for market prediction. *Expert Systems with Applications* 135:60–70.
- [17] Rustam Z, Kintandani P. 2019. Application of arima, support vector regression in stock price prediction with feature selection using particle swarm. *Modelling and the various Simulation in Engineering* 2019(4):1–5.
- [18] Schumaker RP, Chen H. 2009. Textual analysis of stock prediction using breaking news - the AZFin text system.
- [19] Sumathy K, Chidambaram M. 2013. Text mining: concepts, applications, tools and issues-an overview. *International Journal of various types of Computer Applications* 80(4):29–32.
- [20] Sun J, Li H. 2012. Financial distress prediction arima, lstm and using support vector machines - ensemble individual. *Applied Soft Computing* 12(8):2254–2265.

ACKNOWLEDGEMENT

It gives me immense pleasure in expressing my heartfelt thanks to the people who were part of this project in numerous ways. I owe my thanks to all those who gave endless support right from the conception of the project idea to its implementation, it would not have materialized without the help of many.

The dedication, hard work, patience and correct guidance makes any task proficient & a successful achievement. Intellectual and timely guidance not only helps in trying productive but also transforms the whole process of learning and implementing into an enjoyable experience.

I would like to thank our Principal “**Dr. Rohini Kelkar**” and vice principal “**Mr. Asif Rampurawala**” for providing this opportunity, a special thanks to our MSc IT coordinator “**Ms. Beena Kapadia**” for their support, blessings and for being a constant source of inspiration to us. With immense gratitude, I would like to convey my special honour and respect to “**Ms. Beena Kapadia**” (**Project Guide**) who took keen interest in checking the minute details of the project work and guided us throughout the same.

A sincere thanks to the non-teaching staff for providing us with the long lab timings that we could receive along with the books and with all the information we needed for this project, without which the successful completion of this project would not have been possible.

Finally, I wish to avail this opportunity & express a sense of gratitude and love to my friends and my beloved parents for their support, strength and help for everything.

Mr. Pankaj Narayan Manachekar

DECLARATION

I hereby declare that the project entitled, **“Stock Market Prediction Using Machine Learning Models & Sentiment Analysis Of Tweets”** done at Vidyalankar School of Information Technology, has not been in any case duplicated to submit to any other universities for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **MASTER OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Pankaj Narayan Manachekar

TABLE OF CONTENTS

CHAPTER 1	1
INTRODUCTION	1
1.1 BACKGROUND	1
1.2 PROBLEM STATEMENT	1
1.3 PURPOSE , SCOPE AND APPLICABILITY	2
1.4 FEASIBILITY STUDY	2
1.4.1 ECONOMIC FEASIBILITY	3
1.4.2 TECHNICAL FEASIBILITY	3
1.4.3 OPERATIONAL FEASIBILITY	3
1.5 OBJECTIVES	3
 CHAPTER 2	4
SURVEY OF TECHNOLOGIES	4
2.1 INTRODUCTION	4
2.2 LITERATURE REVIEW	4
2.3 COMPARATIVE ANALYSIS	5
2.4 RESEARCH GAP	5
 CHAPTER 3	6
REQUIREMENTS AND ANALYSIS	6
3.1 PROBLEM DEFINITION	6
3.2 REQUIREMENT SPECIFICATION AND ANALYSIS	6
3.3 PLANNING AND SCHEDULING.....	7
3.3.1 GANTT CHART.....	8
3.4 SOFTWARE AND HARDWARE REQUIREMENTS	9
3.4.1 SOFTWARE REQUIREMENTS	9
3.4.2 HARDWARE REQUIREMENTS	12
3.5 PRELIMINARY PRODUCT DESCRIPTION	12

CHAPTER 4	13
SYSTEM DESIGN	13
4.1 BASIC MODULES	13
4.2 DIAGRAMS	14
4.2.1 CLASS DIAGRAM	14
4.2.2 USE CASE DIAGRAM	15
4.2.3 ACTIVITY DIAGRAM	16
4.2.4 BLOCK DIAGRAM	17
4.3 USER INTERFACE DESIGN	18
4.4 TEST CASES DESIGN	23
4.4.1 TEST CASES FOR LOGIN AND REGISTRATION	23
CHAPTER 5	25
IMPLEMENTATION AND TESTING	25
5.1 IMPLEMENTATION APPROACHES	25
5.2 CODING DETAILS	26
5.3 TESTING APPROACHES	36
5.3.1 UNIT TESTING	36
5.3.2 INTEGRATION TESTING	36
CHAPTER 6	37
RESULTS AND DISCUSSION	37
6.1 TEST REPORTS	37
CHAPTER 7	38
CONCLUSIONS	38
7.1 CONCLUSION	38
7.2 FUTURE SCOPE OF THE PROJECT	38
REFERENCES	39

CHAPTER 1 : INTRODUCTION

1.1 Background

The prediction of stock market prices and trends is a problem of interest. The pricing of shares on the stock exchange has dynamic behaviour often driven by the law of supply and demand for action. Uncertainty is the common characteristic that most stock markets have hence the long-term and short-term future states.

Past studies conducted in this subject relied on historical prices collected from companies following their fluctuation characteristics. The theories of efficient market hypothesis state that the movements in financial markets depend on current events, news and releases of products and their impact on the stock values of different companies. Therefore, extracting sentiments and opinions from business and financial news is useful as it may assist in stock-market price prediction.

Nowadays social media is representing the opinions and sentiments of the public about current events. Sentiment analysis entails having a complete understanding of an author's opinion expressed in a text. This study involves taking existing non-quantifiable statistics on financial news and public sentiments on companies.

The collected data is used in predicting the trends on future stocks. The assumption here is that the opinions and news have a significant impact on the changes in the stock markets with an attempt to establish the correlations between public sentiments, opinions, stock trends and company news.

1.2 Problem Statement

Stock market prediction relies on factors such as interest rates, economic activities and related markets that influence the demand and supply of the trading volume. Currently, Stockbrokers who execute trades and advise clients, rely on their experience, technical analysis (price trends) or fundamental analysis in picking their stocks.

With the value of trade money involved, the improper investment could easily mean great losses for investors, especially if they keep making wrong decisions. Lack of guaranteed returns has also led to the reluctance by potential investors to participate in the market. It is therefore desirable to have a model that can guide on the most likely next day prices (prediction) as a basis for making any investment decision.

This study proposes text mining of financial news and public sentiments and opinions from social media such as twitter. The combination of market data and news features together helps improve the accuracy of predictions. Regardless, already existing systems have failed to effectively integrate news features together with market data. With this, the results obtained are converted into numeric forms that feed the prediction process.

1.3 Purpose, Scope and Applicability

1.3.1 Purpose :-

Stock Price Prediction using machine learning helps you discover the future value of company stock and other financial assets traded on an exchange. The entire idea of predicting stock prices is to gain significant profits. Stock market prediction aims to determine the future movement of the stock value of a financial exchange. The accurate prediction of share price movement will lead to more profit investors can make.

1.3.2 Scope :-

Stock market prediction means determining the future scope of market. The project is limited to only the company's shares listed on the NSE. Additionally, the company should have traded for at least five years to ensure there is data consistency.

1.3.3 Applicability :-

The stock market prediction has extra advantages for novice traders as they are the kind of traders who are more prone to making mistakes and facing severe losses in the market compared to experienced traders. You can better analyse and predict the stock market by gaining a complete understanding of the same.

1.4 Feasibility Study.

A significant consequence of starter examination is the affirmation that the framework request is feasible. This is possible just if it is viable inside limited resource and time. The various potential outcomes that must be dismembered are-

- Operational Feasibility
- Economic Feasibility
- Technical Feasibility

1.4.1 Economic feasibility :-

Monetary Feasibility or Cost-advantage is an assessment of the budgetary resistance for a PC based endeavour. As gear was presented from the most punctual beginning stage and for heaps of purposes along these lines the cost on the undertaking of hardware is low.

1.4.2 Technical feasibility :-

As demonstrated by Roger S. Pressman, Technical Feasibility is the assessment of the specific resources of the affiliation. The system is made for the stage independent condition. Python code, Html, CSS, coming up short immediately of visual studio code are used to develop the structure. The structure is really down to earth for development and can be made with the present office.

1.4.3 Operational feasibility :-

Operational Feasibility deals with the examination of prospects of the framework to be made. This framework operationally assists customers in sufficiently foreseeing stock estimation of an association, with the objective that customers can settle on up their stock exchanging decisions similarly as improve the gauge model reliant on the comprehension.

1.5 Objectives.

This project aims at predicting future price movements of the stock market using financial news and peoples' opinions posted on the social media platforms hence getting sentiments that will aid in stock price forecasting. To investigate how the stock market NSE operates. To analyze the current methods used in the stock market prediction. To evaluate current methods, use in text mining and processing from social media. To develop a model for stock market price prediction based on sentiment on social media. To test and validate the stock market price prediction model.

CHAPTER 2 : SURVEY OF TECHNOLOGIES

2.1 Introduction

Stock Market Prediction using Machine Learning using Sentiment Analysis project takes help from various methods and studies done by various domain experts, to make this project a reality. There is mixing and parameter tuning of these methods to make it according to the project needs. This section of the document reviews existing related literature and previous research and studies on predicting the stock exchange market using sentiment analysis and other methods.

2.2 Literature Review

Over the past two decades many important changes have taken place in the environment of stock markets. The development of powerful communication and leading facilities has enlarged the scope of selection for investors as well as for users. Sentiment Analysis is an information extraction task that aims to obtain writer's feelings expressed in positive, negative or neutral comments.

The various machine learning techniques on providing a positive or negative sentiment on a tweet. The author uses different techniques are Arima Model, Linear Regression etc. Twitter API is used to analyze sentiment in the tweet data set and the machine learning models techniques would be used for predicting market movement.

Another hypothesis which is currently under survey is, whether the early indicators extracted from online sources (blogs, twitter feeds etc.) can be used to predict changes in economic and commercial indicators.

Most of the stock prediction approaches have been built on technical and fundamental analyses of stocks. In recent studies, it has been evident that there is a strong correlation between news articles related to a company and its stock price movements.

Sentiment analysis has increasingly received significant research attention. Opinion mining has emerged as a key tool to comprehend the sentiments of the target audience in order to build superior predictive models. An elaborative review of the evolution of sentiment analysis was conducted by with special context to research topics and highly cited papers.

2.3 Comparative Analysis

Comparing and analysing the methods and techniques implemented in this project with many of the previous work done on this or related topic gives various insights and results that helped us to improve on the methods implemented on this project by either implementing hybrid techniques or improving by providing home-made data.

Stock market prediction is considered as one of the most promising research area that is attaining the attention of various researchers. The vital information which is available for access is assumed to have predictive relationships to the future stock returns. The present work gives information to the investors so that the decision could be made better during the purchase of stocks.

The factors that contribute towards the decision are the historical prices of stocks and tweet comments regarding the same. The proposed method uses four methods for predicting the stock market status, namely, Linear Regression (LR) , Long Short-Term Memory (LSTM), Arima Model (AM) approaches.

2.4 Research Gap

As far as our project goes, comparing it with various papers and studies done online people and domain experts have been working on these kinds of methods with different applications have covered the various technological gap but the idea that we are implementing these techniques on generally shows a great research gap as there have been no instances where resource finding used these techniques to ease the whole process of searching the resources.

During past years many researchers have given contribution in this field but there is still a need to do the research for stock selection based on fundamental and technical analysis, this provides strong motivation for a system that can efficiently extract data from different web sources can be done using web crawling and capable of prediction of stock price, which would be helpful for individual for selection of stock for trading and investment.

Research gaps and challenges between existing techniques are listed and detailed, which helps researchers to upgrade future works. Here, this paper provides an overview of the applications of machine learning in stock market forecasting to determine what can be done in the future

CHAPTER 3 : REQUIREMENTS AND ANALYSIS

3.1 Problem Definition

Stock Market is one of the problem you have hear about it every time it reaches a new high or a new low. The rate of investment and business opportunities in the Stock market can increase if an efficient algorithm could be devised to predict the short term price of an individual stock. A prediction model for finding and analysing correlation between contents of tweets and stock prices and then making predictions for future prices can be.

3.2 Requirements Specification / Analysis.

Based on the objectives as well as the user requirements, this section outlines the various requirements to be met in the research.

a. Functional requirements :-

These are functions or processes the proposed system and its components must perform. They are a definition of what users of the system expect form it. For the system, the functional requirements include :-

- The system should allow a user to select the stock to be predicted.
- The system should crawl the historical and current price on a company's stock price
- The system should retrieve financial news and sentiments pertaining the company's stock from twitter.
- The system should perform pre-processing of the tweets to clean then and store them in a comma separated values (csv) file.
- The system should be able to generate an approximate share price for the next trading day.

b. Non-functional requirements :-

Unlike the functional requirements, non-functional requirements place constraints or limits in how the proposed system will achieve its functional requirements. They describe how well the system does its functions and are classified based on the needs of the users. The non-functional requirements of the system include :-

- Usability- The intended users of the proposed system are the stockbrokers from different accredited trading firms. The interaction with the system will be simple to allow stock price prediction.
- Reliability - The reliability of the model will highly depend on the accuracy of the data collected (stock). As this data will be used to train the model which will be used in prediction.
- Interoperability – This is the degree to which the developed system will be able to facilitate of couple the different interfaces with other systems.
- Response time – this is defined as the time between the end of a request by a user and start of the response. For the proposed system, the response time should be fast.
- Scalability – This describes the degree in which the system is able to expand its processing or functional capabilities outward or upward with the aim of supporting business growth and user requirements.
- Persistent storage- the proposed system components and devices should be able to retain data or information after device's power have been shut down or eliminated.

3.3 Planning And Scheduling

3.3.1 Gantt Chart :-

A Gantt chart makes it simple to create, view and monitor project activities and tasks over a given time. Typically, each activity, process or task in a Gantt chart is represented by a horizontal bar scaled parallel to a calendar and/or dates.

The start and end of each bar represent the start and the end of that particular activity. Gantt charts help in quickly understanding the different tasks in a project, their schedule (start and end date) and any overlapping tasks or activities.

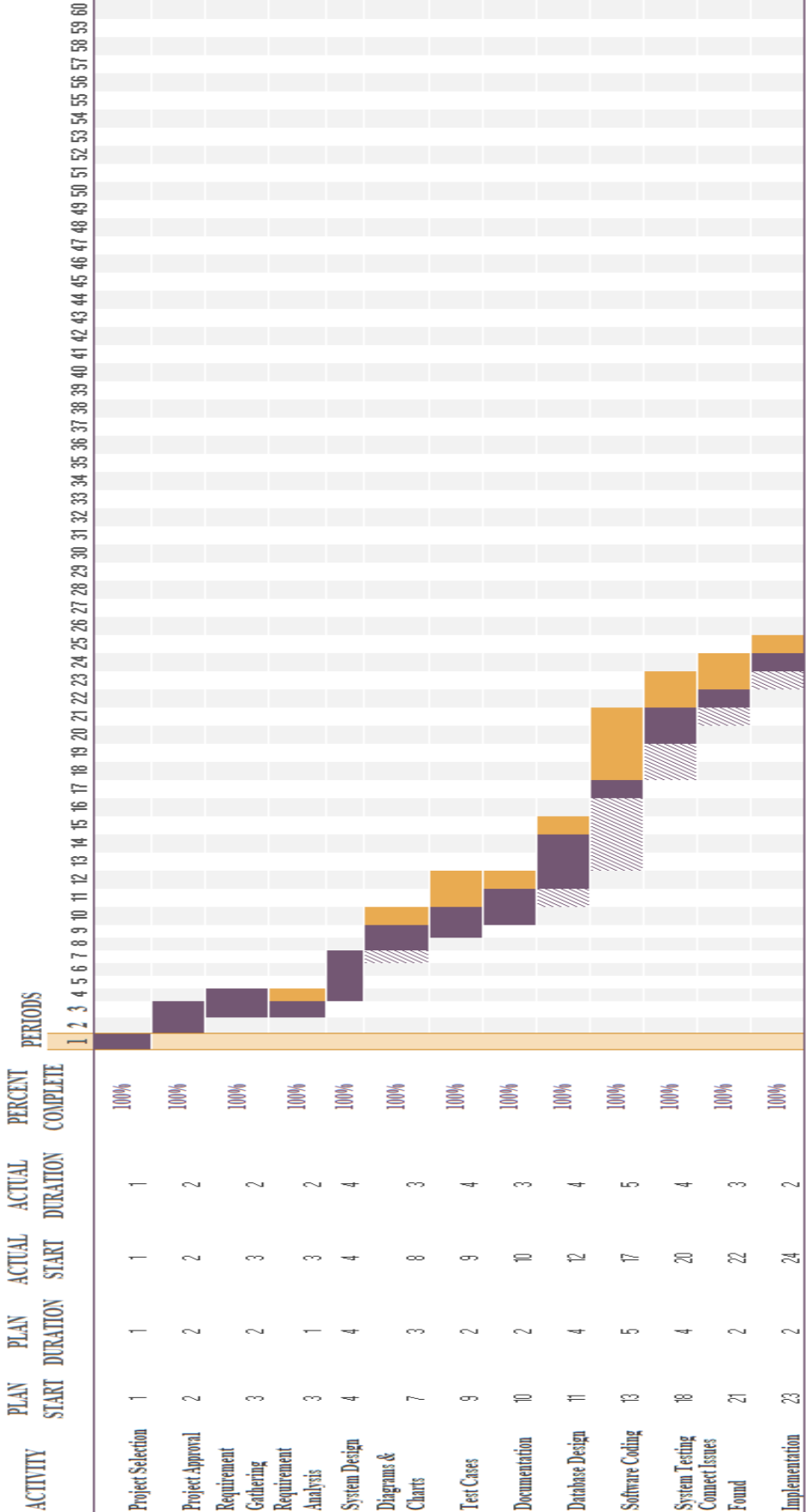
A Gantt chart is constructed with a horizontal axis representing the total time span of the project, broken down into increments (for example, days, weeks, or months) and a vertical axis representing the tasks that make up the project (for example, if the project is outfitting your computer with new software, the major tasks involved might be: conduct research, choose software, install software).

STOCK MARKET PREDICTION USING M.L. MODELS & SENTIMENT ANALYSIS OF TWEETS

17FEB2021 = 1/1/2021

Period Highlight: 1

Plan Duration Actual Start %Complete Actual (beyond plan) %Complete (beyond plan)

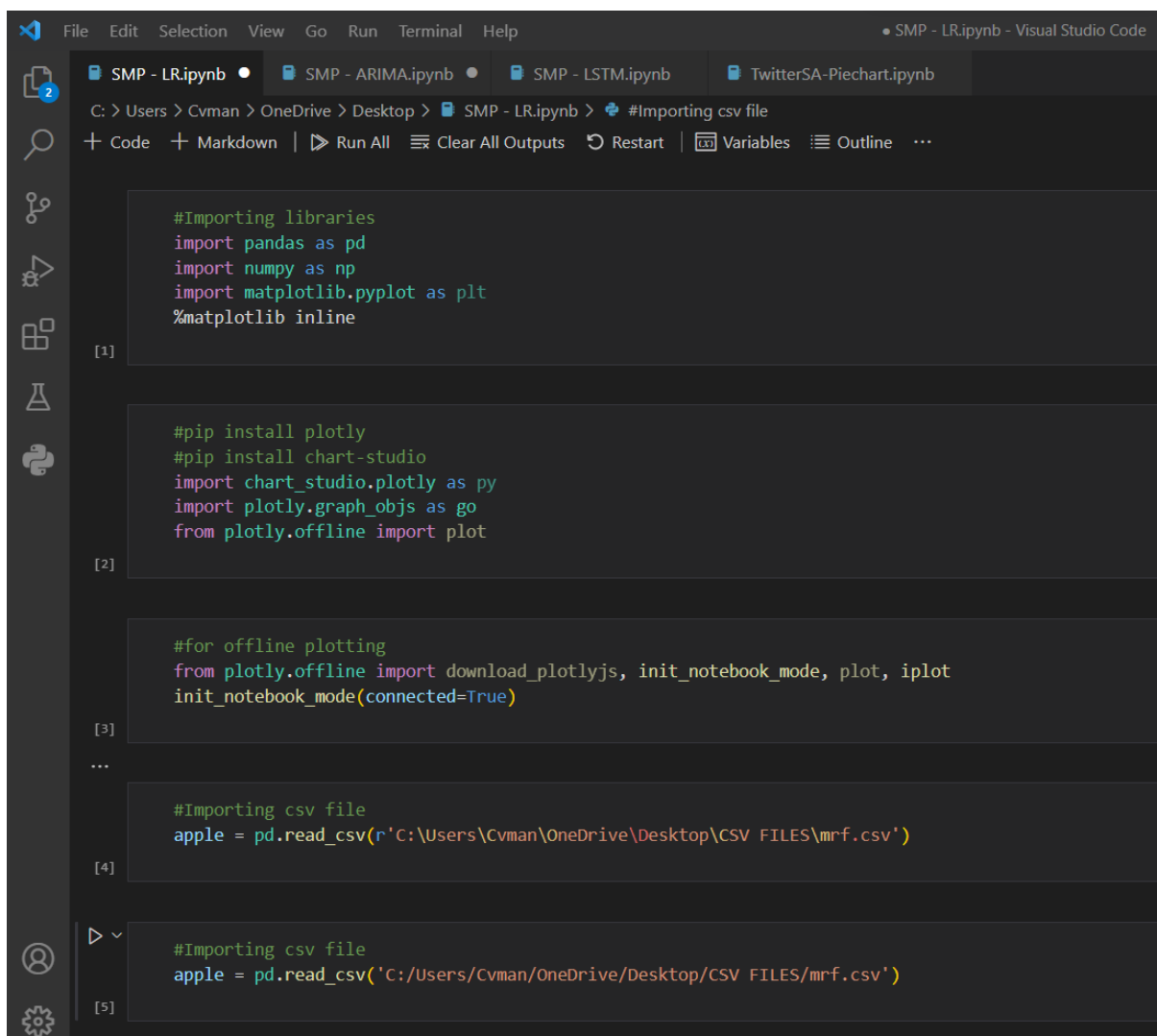


3.4 Software And Hardware Requirements

3.4.1 Software Requirements :-

Visual Studio Code :-

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages and runtimes (such as C++, C#, Java, Python, PHP, Go, .NET).



```
File Edit Selection View Go Run Terminal Help
SMP - LR.ipynb SMP - ARIMA.ipynb SMP - LSTM.ipynb TwitterSA-Piechart.ipynb
C:\> Users > Cvman > OneDrive > Desktop > SMP - LR.ipynb > #Importing csv file
+ Code + Markdown | ▶ Run All ⌵ Clear All Outputs ↺ Restart | 📄 Variables 📖 Outline ...

#Importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

[1]

#pip install plotly
#pip install chart-studio
import chart_studio.plotly as py
import plotly.graph_objs as go
from plotly.offline import plot

[2]

#for offline plotting
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)

[3]

...

#Importing csv file
apple = pd.read_csv(r'C:\Users\Cvman\OneDrive\Desktop\CSV FILES\mrf.csv')

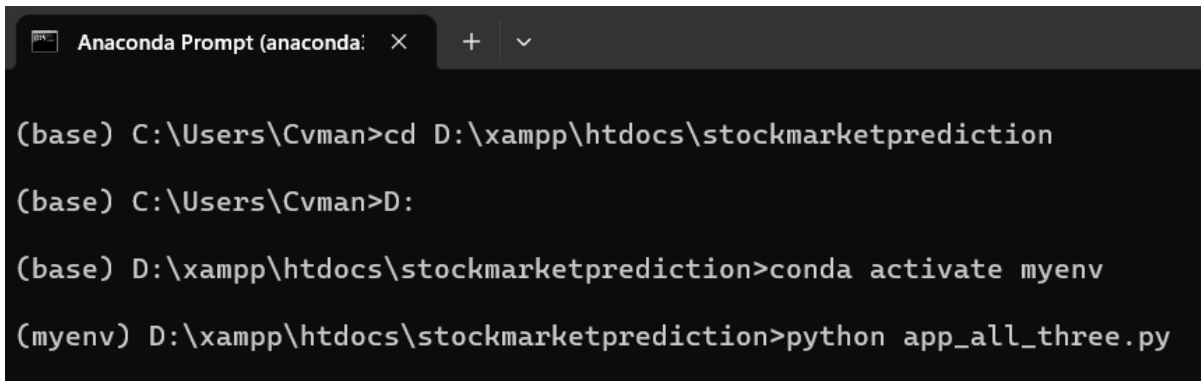
[4]

▶ #Importing csv file
apple = pd.read_csv('C:/Users/Cvman/OneDrive/Desktop/CSV FILES/mrf.csv')

[5]
```

- Python 3.10 or 3.11 in Visual Studio Code which is used for data pre-processing, model training-testing and prediction.
- In Visual Studio Code various extensions are being used. The system should collect data from web sources like Yahoo Finance.

Anaconda Prompt :-



```
Anaconda Prompt (anaconda: x + v
(base) C:\Users\Cvman>cd D:\xampp\htdocs\stockmarketprediction
(base) C:\Users\Cvman>D:
(base) D:\xampp\htdocs\stockmarketprediction>conda activate myenv
(myenv) D:\xampp\htdocs\stockmarketprediction>python app_all_three.py
```

Steps for running the python file and installing libraries as well as creating new python my environment :-

```
(base) C:\Users\Cvman>cd D:\xampp\htdocs\stockmarketprediction
```

```
(base) C:\Users\Cvman>D:
```

```
(base) C:\xampp\htdocs\stockmarketprediction>conda create -n myenv
python=3.11
```

conda update -n base -c defaults conda #### - if you want to update the new version of conda.

environment location: C:\Users\Cvman\anaconda3\envs\myenv

Proceed ([y]/n)? y

```
(base) C:\xampp\htdocs\Stocks>conda activate myenv
```

```
(myenv) C:\xampp\htdocs\Stocks>python app_all_three.py
```

```
(myenv) C:\xampp\htdocs\Stocks>pip install plotly
```

```
(myenv) C:\xampp\htdocs\Stocks>pip install pandas
```

```
(myenv) C:\xampp\htdocs\Stocks>pip install matplotlib
```

(myenv) C:\xampp\htdocs\Stocks>pip install numpy

(myenv) C:\xampp\htdocs\Stocks>pip install statsmodels

(myenv) C:\xampp\htdocs\Stocks>pip install sklearn

(myenv) C:\xampp\htdocs\Stocks>pip install datetime

(myenv) C:\xampp\htdocs\Stocks>pip3 install -U scikit-learn

(myenv) C:\xampp\htdocs\Stocks>pip install chart_studio

(myenv) C:\xampp\htdocs\Stocks>pip install keras

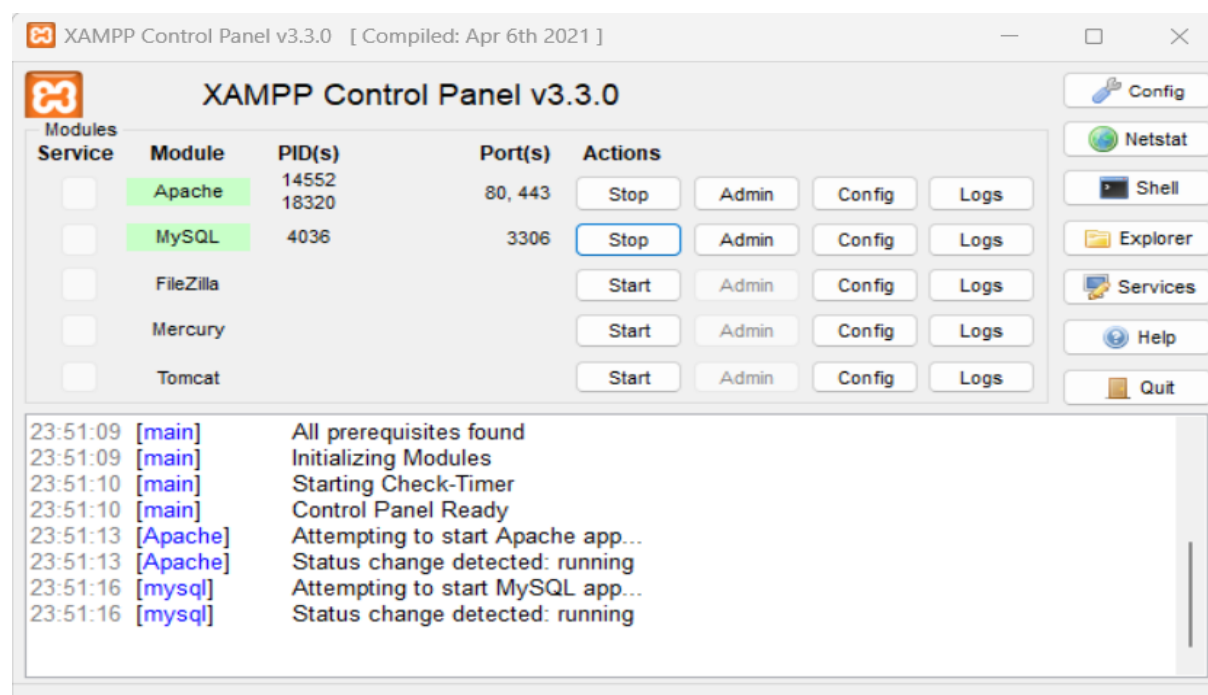
(myenv) C:\xampp\htdocs\Stocks>pip install tensorflow

(myenv) C:\xampp\htdocs\Stocks>pip install flask

(myenv) C:\xampp\htdocs\Stocks>pip install -U kaleido

(myenv) C:\xampp\htdocs\Stocks>python app_all_three.py

Xampp Control Panel :-



3.4.2 Hardware Requirements :-

- **Windows 10/11 :-** For using an Visual Studio Code, you'll most likely need the 64-bit version of Windows 10. Visual Studio Code require this as a minimum and don't work with 32-bit Windows.
- **Processor :-** Most Visual Studio Code require an Intel processor, but some also have support for AMD processors. The Intel processor should have support for 2 GHZ or faster CPU.
- **Memory :-** You'll need at least 8 GB RAM to use an Visual Studio Code. For some, the minimum memory requirement may be higher. It's important to note that 4 GB of disk storage would not make up for memory as that's a requirement.
- **Storage :-** Every emulator has different storage requirements, but you should keep at least 15 GB in mind. You'll need more of it once you start using the Visual Studio Code and download all your extensions. Extensions will also get stored on the hard drive, so make sure you have sufficient space. It's always better to have more space than required.

3.5 Preliminary Product Description

In this project, we investigate the impact of sentiment expressed through Twitter tweets on stock price prediction. Twitter is the social media platform which provides a free platform for each individual to express their thoughts publicly. Specifically, we fetch the live twitter tweets of the particular company using the API. All the stop words, special characters are extracted from the dataset. Thus, the tweets are classified into positive, negative, and neutral tweets. To predict the stock price, the stock dataset is fetched from yahoo finance API. The stock data along with the tweets data are given as input to the machine learning model to obtain the result. ARIMA , Linear Regression and LSTM is used as a model to predict the stock market price. The effectiveness of the proposed project on stock price prediction is demonstrated through experiments on several companies like Apple, Amazon, Microsoft using live twitter data and daily stock data. The goal of the project is to use historical stock data in conjunction with sentiment analysis of news headlines and Twitter posts, to predict the future price of a stock of interest.

CHAPTER 4 : SYSTEM DESIGN

4.1 Basic Modules

➤ Login & Registration :-

User will be able to login and register through unique username and password.

➤ Home Page :-

User will see their own dashboard page in which they can check flat details, user details, bill details as well as complaints and notices. User will know more about stock market information. User can also download various stocks of company's list.

➤ Predict The Stock Price Page :-

User can enter company's stock symbol or company's name to predict the specific stock prices prediction for specific company.

➤ Stock Data Prediction Page :-

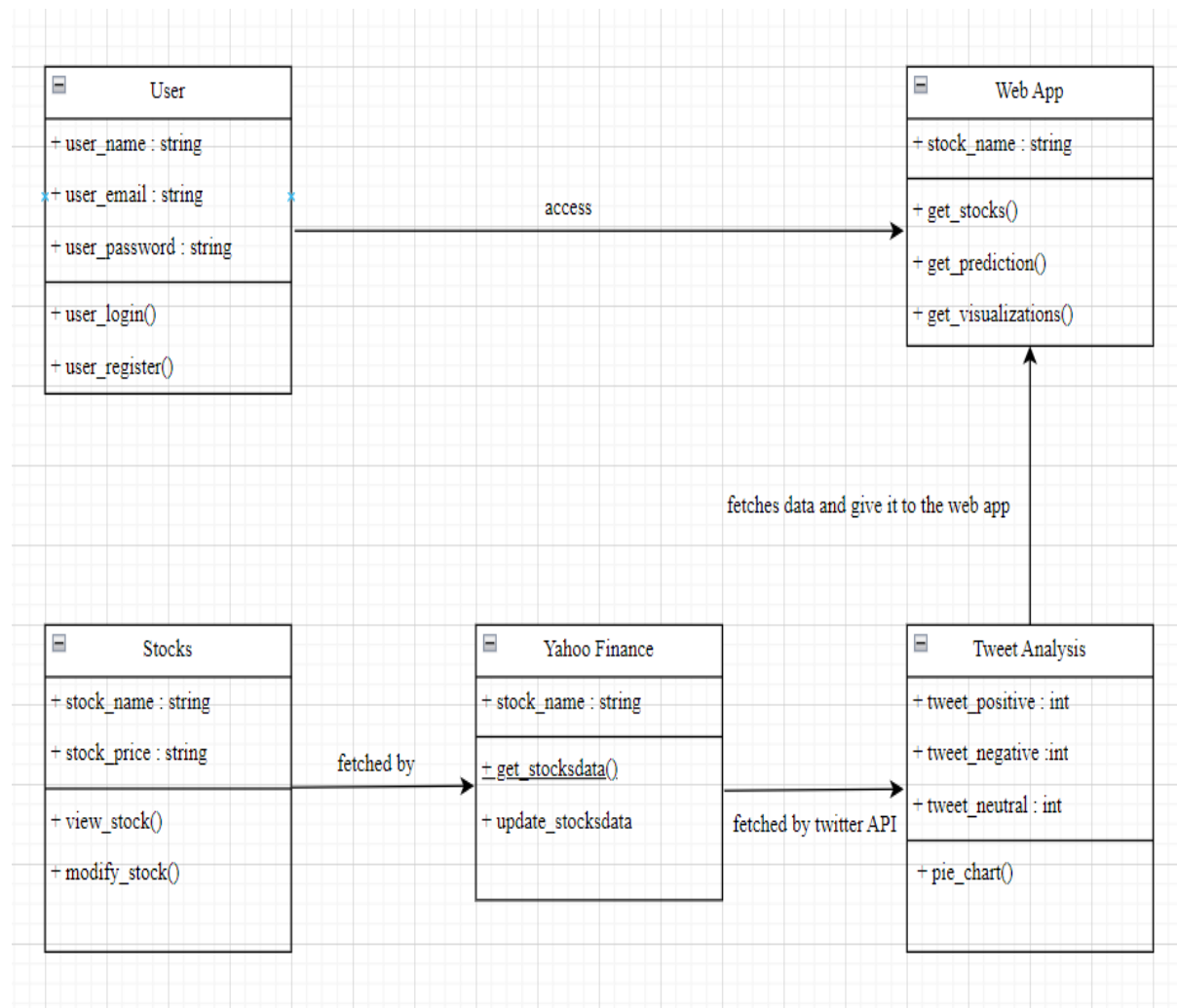
User will see recent trends for stock prices for a specific company. User will also get to see a prediction accuracy for Arima Model , Linear Regression Model and Long Short-Term Memory Model.

User can also see recent tweets and news about specific company. User can also see a sentiment analysis of pie chart for a specific company's tweets, and it can also tell you the overall review of tweets polarity whether it is positive , negative or neutral.

User can also see a predicted price of a specific company for the next 7 days. User can also see a recommendation message in which they will know that according to the machine learning predictions and sentiment analysis of the tweets whether the stocks price of specific company is expected to buy or not.

4.2 Diagrams.

4.2.1 Class Diagram :-

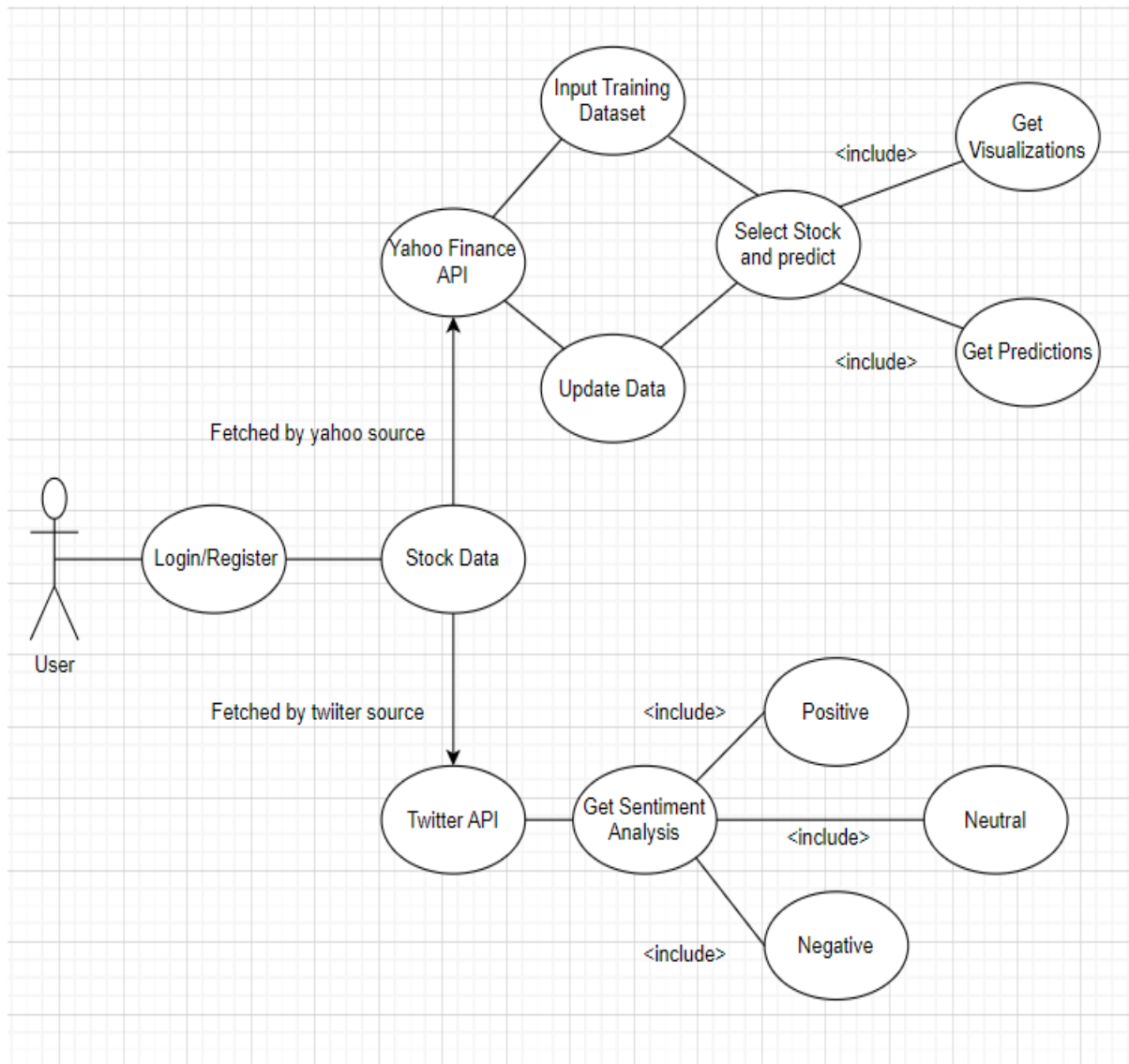


Description of Class Diagram :-

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

4.2.2 Use Case Diagram :-

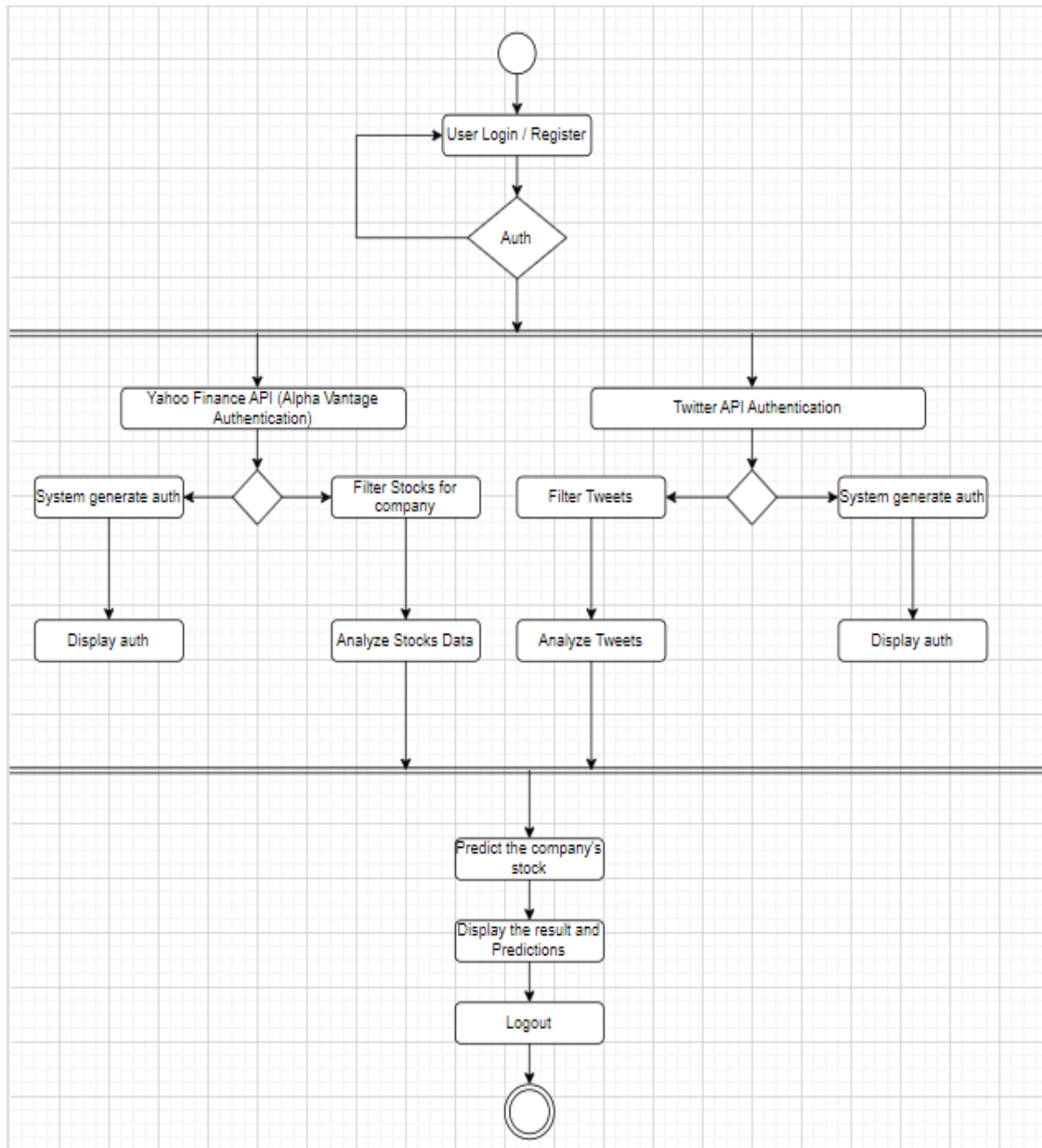


Description of Use Case Diagram :-

Use case diagrams referred as a behaviour model or diagram. It simply describes and displays the relation or interaction between the users or customers and providers of application service or the system.

It describes different actions that a system performs in collaboration to achieve something with one or more users of the system. Use case diagram is used a lot nowadays to manage the system.

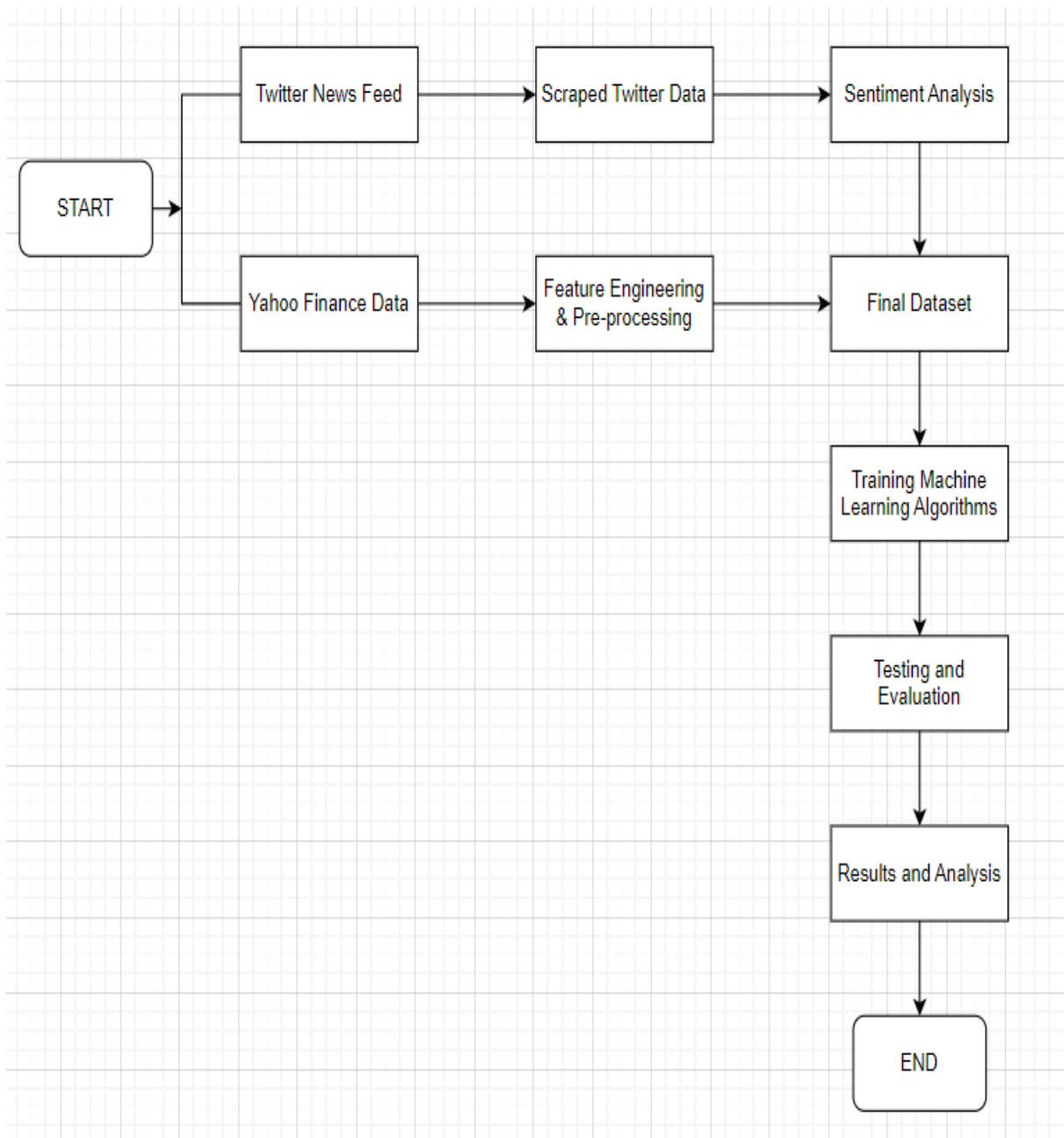
4.2.3 Activity Diagram :-



Description of Activity Diagram :-

The user login is done through authentication. It captures the dynamic behaviour of the system. Other four into four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

4.2.4 Block Diagram :-

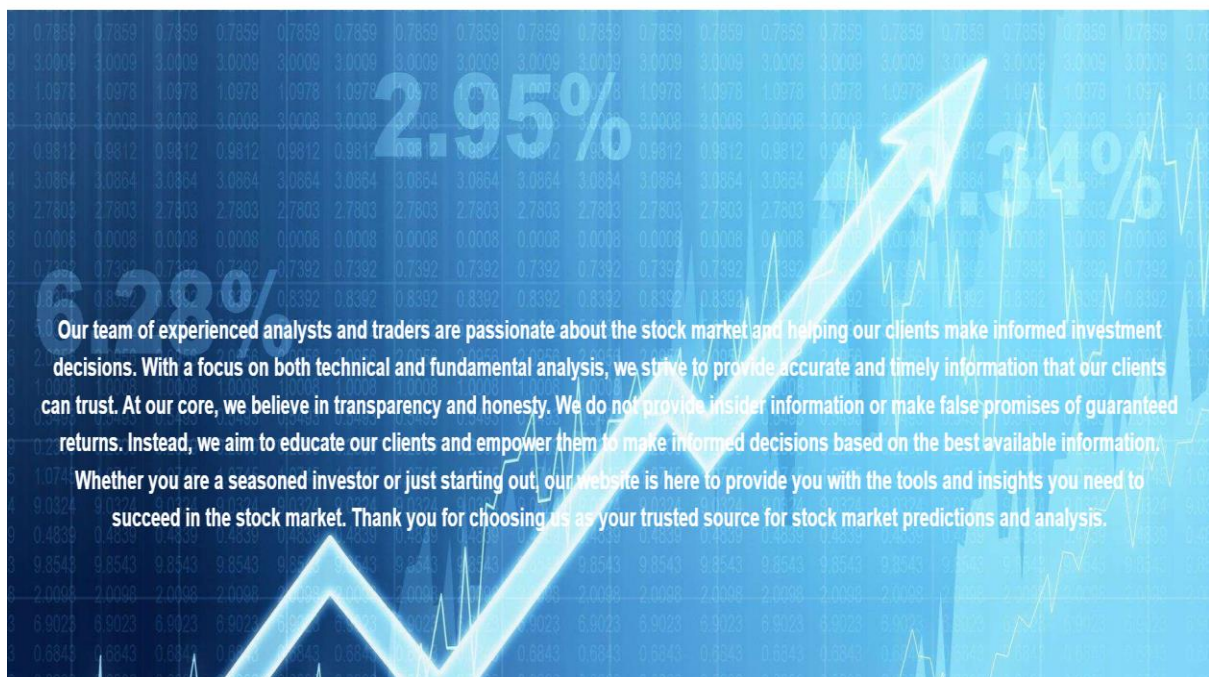
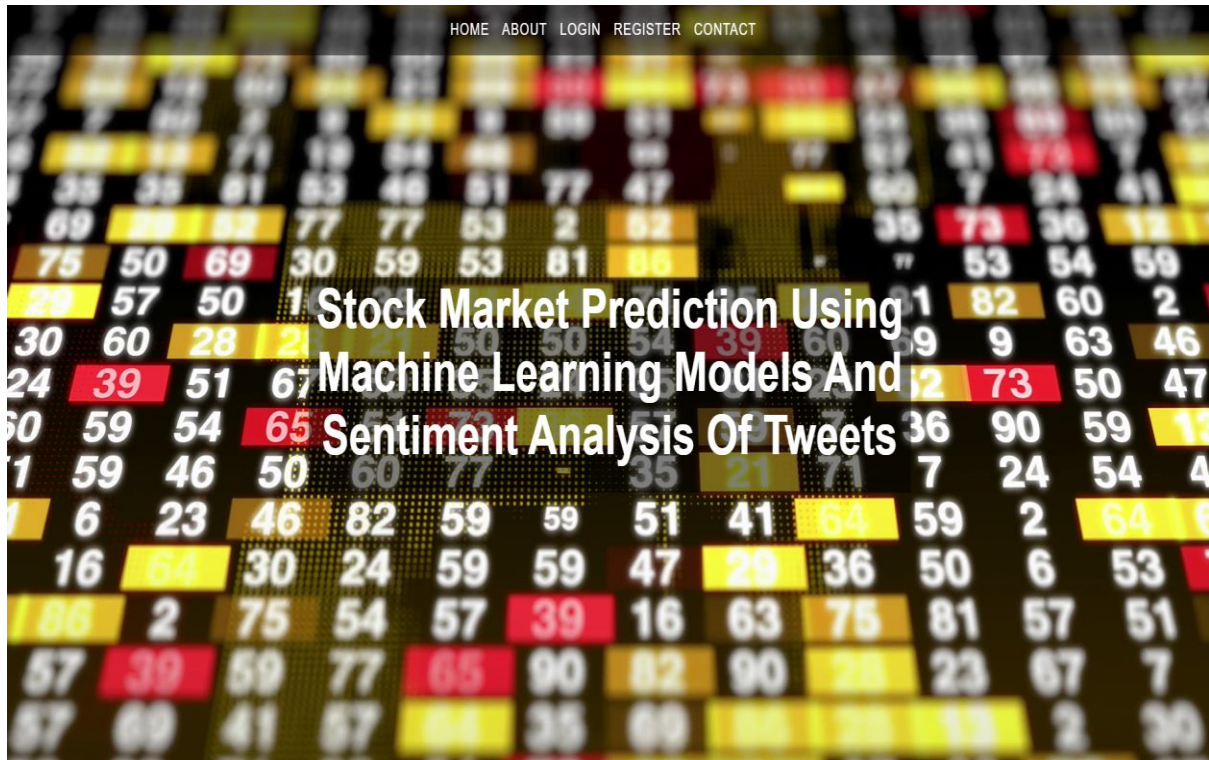


Description of Block Diagram :-

A block diagram is a visual representation of a system that uses simple, labelled blocks that represent single or multiple items, entities or concepts, connected by lines to show relationships between them.

4.3 User Interface Design.

1. Home Page :-



Contact



Name: Chaitanya Vijay Mane ,

Name: Pankaj Narayan Manchekar ,

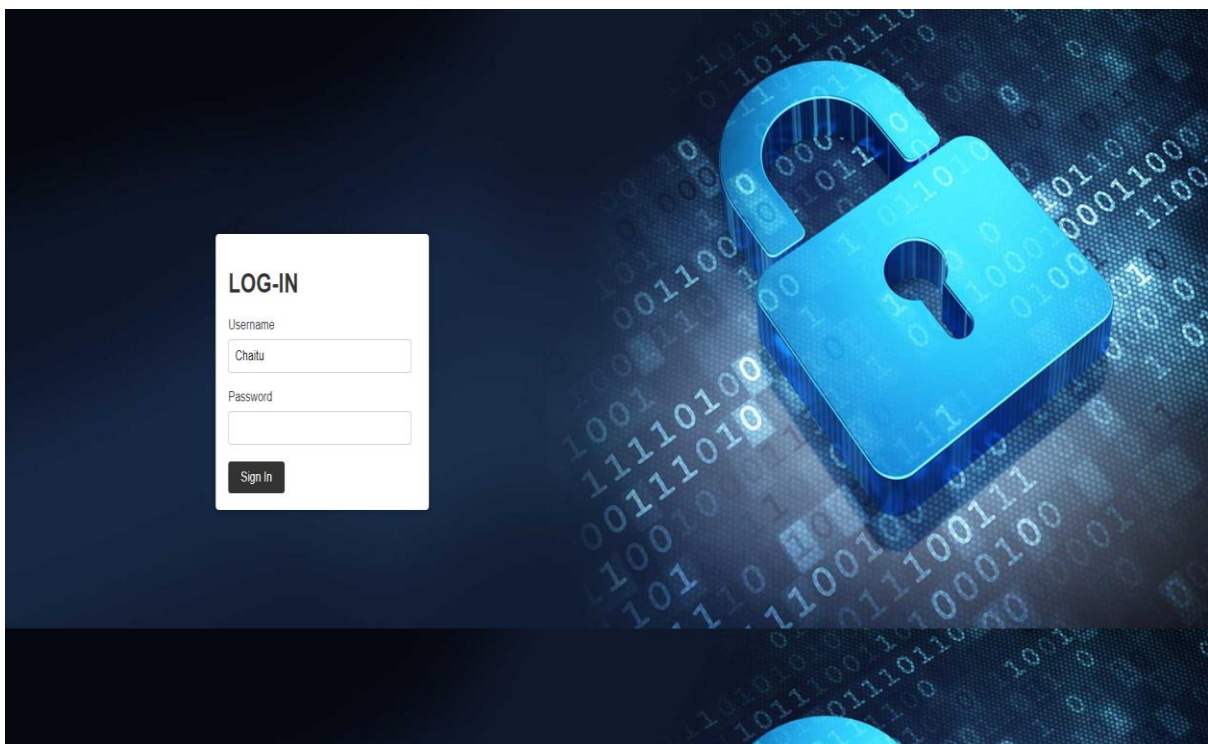
Address: Sewri Wadala Mumbai 400015 ,Address: Sewri Wadala Mumbai 400015 ,

Phone No.: 9987026493 ,

Phone No.: 8828088116 ,

Gmail: manechaitanya224@gmail.com Gmail: pnkj.manchekar@gmail.com

2. Login Page :-



3. Register Page :-



Register Here!

Name

Email

Address

Phone

Username

Password

REGISTER NOW

4. Predict The Stock Price Page :-

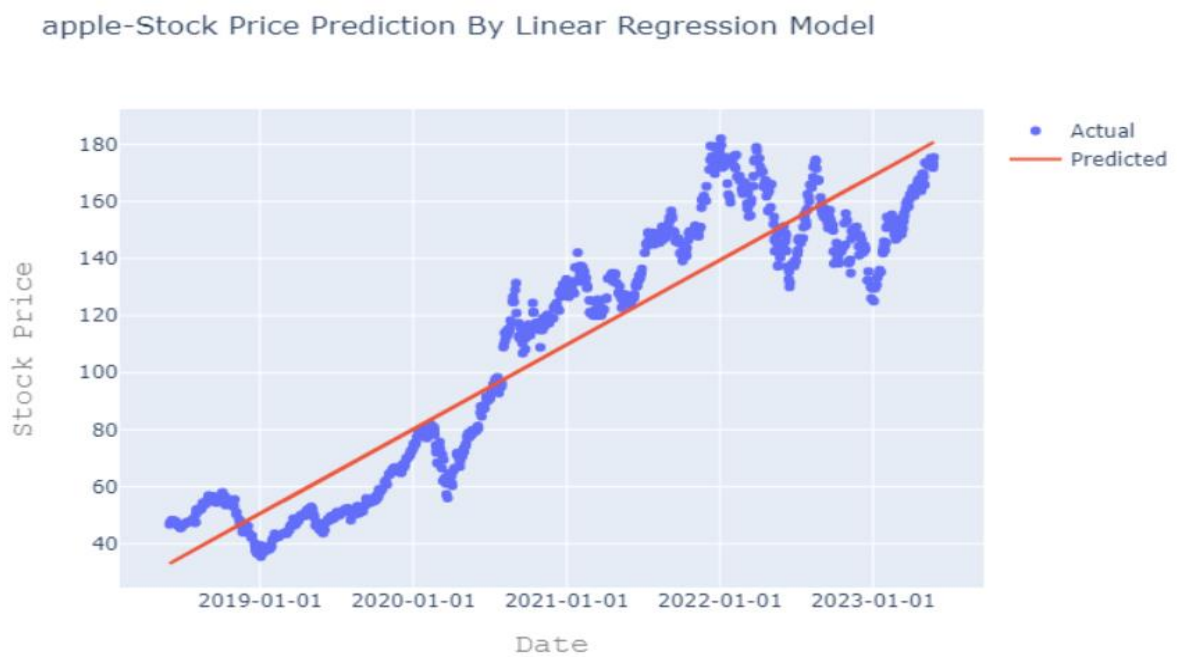
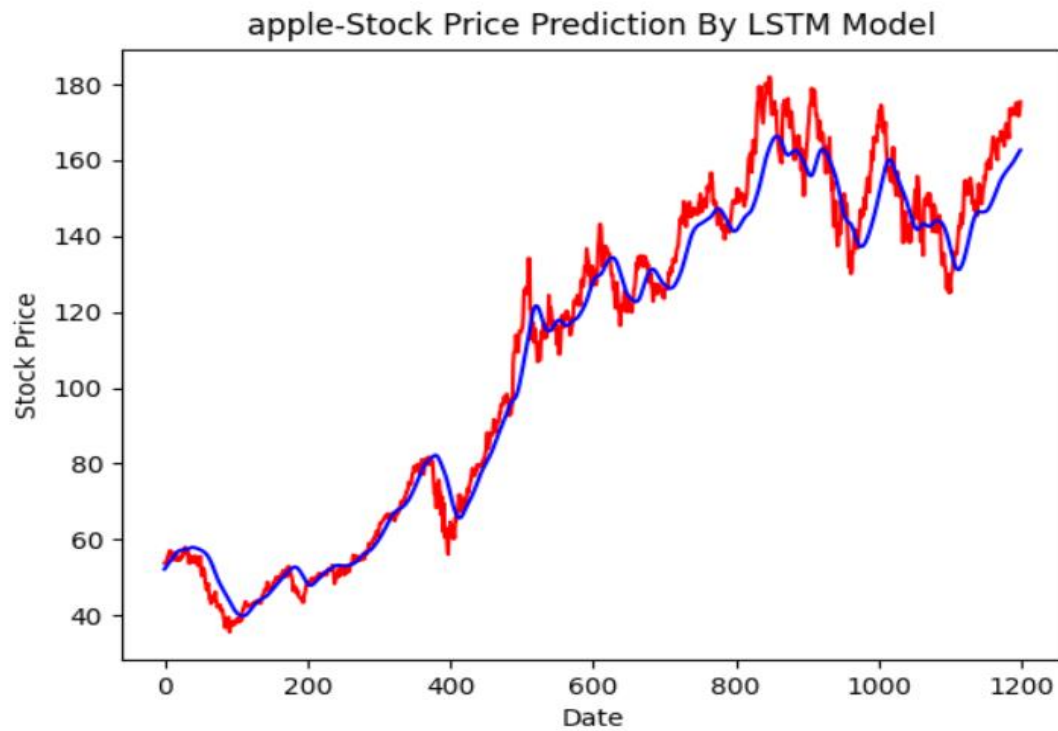


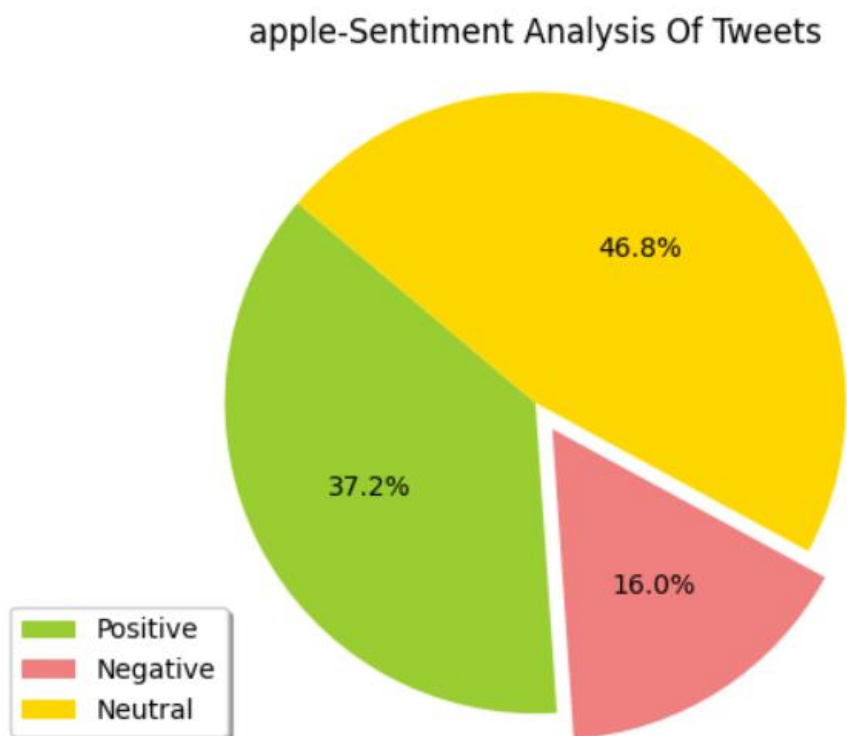
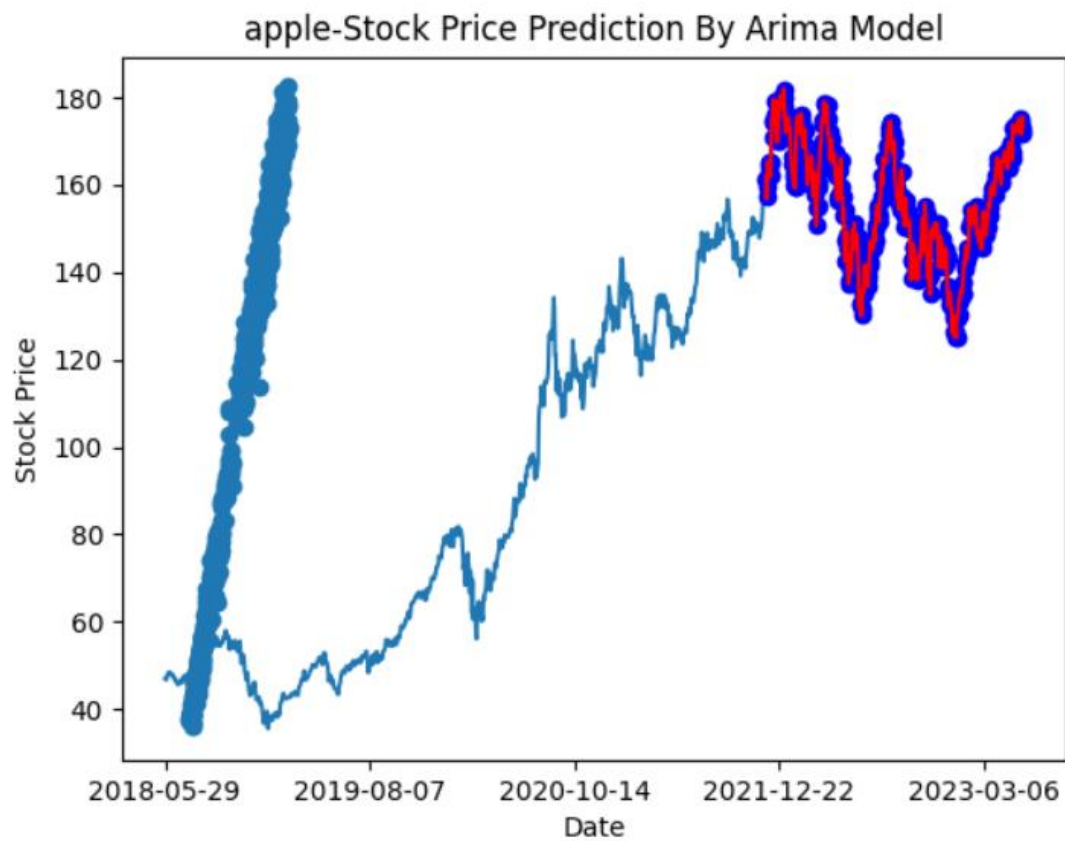
ENTER A STOCK NAME

apple

Predict The Stock Price

5. Stock Data Prediction (Machine Learning Models) :-





4.4 Test Cases Design

4.4.1 Test Cases for Login & Registration.

Test Case No.	Test Case Description	Expected Result	Pass/Fail	Actual Result
1.	Login with valid username and valid password (User).	Login should be successful, and it should show the home screen of the admin and user.	Pass	Login Successful
2.	Login with invalid username and invalid password (User).	“error” message should be displayed.	Pass	Error Message
3.	Login with valid username and invalid password (User).	“error” message should be displayed.	Pass	Error Message
4.	Login with invalid username and valid password (User).	“error” message should be displayed.	Pass	Error Message
5.	Login without entering username and password (User).	“error” message should be displayed.	Pass	Error Message

Test Cases for Prediction & Visualizations.

Test Case No.	Test Case Description	Expected Result	Pass/Fail	Actual Result
1.	Entering a valid stock company name.	Get the visualization output in the form of graphs.	Pass	Predicted Successfully
2.	Entering a non-valid stock company name.	“error” message should be displayed.	Pass	Error Message
3.	Prediction Button.	Should show the predicted output after clicking on it.	Pass	Predicted output
4.	Clicking on Home , About & Contact Button.	Should display the information that is provided in their particular section.	Pass	Displayed Successfully
5.	Clicking on Login & Registration Button.	Should display the information that is provided in their particular section.	Pass	Displayed Successfully

CHAPTER 5 : IMPLEMENTATION AND TESTING

5.1 Implementation Approaches

In this project we have developed a working prototype machine learning model with the current available requirement details and for the actual requirement of the product to develop the system. Prototype is the trimmed version of the actual product with limited features and functionalities.

Stock market prediction has been a significant area of research in Machine Learning. Machine learning algorithms such as Linear regression, Arima , Long Short Term Memory help predict the stock market. This article presents a simple implementation of analyzing stock market prediction using machine learning and analysing the proper sentiment analysis of tweets of specific stock company name to show the negative, positive and neutral reviews from the tweets.

Testing such as Unit testing, Integration testing, functional testing and System testing has been performed in order to understand the quality of the project and how efficiently it works.

It provides visual studio code editor which helps you write better code, work faster and be more productive by offering advanced code completion, refactoring, and code analysis. It also uses the anaconda prompt to run the given python file.

Visual Studio Code and the Anaconda Prompt are the two main aspects of this project. It also uses the XAMPP control panel server to run the project through Web Application. This can be done by using the Web Flask. The service module of the XAMPP server such as (Apache & MySQL) should be started before running the project so that it can run through their own localhost.

The login and the registration of the given project is being done through the phpMyAdmin . Database named as (stock.sql) has been created for the login and registration of the given stock market prediction project.

5.2 Coding Details

Importing All Python Libraries

```
import plotly.express as px
import pandas as pd
import plotly.io as pio
import tweepy
import numpy as np
import pandas as pd
import re
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import chart_studio.plotly as py
import plotly.graph_objs as go

from pandas.plotting import lag_plot
from textblob import TextBlob
from pandas import datetime
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_squared_error
from plotly.offline import plot
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_squared_error
```

WEB APP USING FLASK

```
from flask import Flask, render_template, request
app = Flask(__name__)

@app.route('/')
def index():
    return render_template('home.html')

@app.route('/calculate', methods=['POST'])
def calculate():
    number = str(request.form['number'])
    result = number
```

```

##### LINEAR REGRESSION ALGORITHM #####

#Importing csv file
apple = pd.read_csv(r'D:\\xampp\\htdocs\\stockmarketprediction\\input\\'+ number +'.csv')

#Importing csv file
apple = pd.read_csv('D:\\xampp\\htdocs\\stockmarketprediction\\input\\'+ number +'.csv')

#First 5 rows
apple.head()

#Prints information about dataframe
apple.info()

#Converting the date column into datetime format
apple['Date'] = pd.to_datetime(apple['Date'])

#Printing the functions
print(f'Dataframe contains stock prices between {apple.Date.min()} {apple.Date.max()}')
print(f'Total days = {(apple.Date.max() - apple.Date.min()).days} days')

#Description of the data in the DataFrame
apple.describe()

#Create a boxplot to visually check the outliers
apple[['Open','High','Low','Close','Adj Close']].plot(kind='box')

#Plot the graph using plotly libraries
# Setting the layout for our plot
layout = go.Layout(
    title=number+'-Stock Price Prediction By Linear Regression Model',
    xaxis=dict(
        title='Time',
        titlefont=dict(
            family='Courier New, monospace',
            size=18,
            color='#7f7f7f'
        )
    ),
    yaxis=dict(
        title='Stock Price',
        titlefont=dict(
            family='Courier New, monospace',
            size=18,
            color='#7f7f7f'
        )
    )
)

```

```

#Passing the datalist and the layout created to plot variable
apple_data = [{ 'x':apple['Date'], 'y':apple['Close']}]
plot2 = go.Figure(data=apple_data, layout=layout)

# Building the regression model
from sklearn.model_selection import train_test_split

#For preprocessing
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler

#For model evaluation
from sklearn.metrics import mean_squared_error as mse
from sklearn.metrics import r2_score

#Split the data into train and test sets
X = np.array(apple.index).reshape(-1,1)
Y = apple['Close']
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=101)

# Feature scaling
scaler = StandardScaler().fit(X_train)

#Import linear regression
from sklearn.linear_model import LinearRegression

#Creating a linear regression model
lm = LinearRegression()
lm.fit(X_train, Y_train)

#Plot actual and predicted values for train dataset
trace0 = go.Scatter(
    x = X_train.T[0],
    y = Y_train,
    mode = 'markers',
    name = 'Actual'
)
trace1 = go.Scatter(
    x = X_train.T[0],
    y = lm.predict(X_train).T,
    mode = 'lines',
    name = 'Predicted'
)
apple_data = [trace0,trace1]
layout.xaxis.title.text = 'Time'
plot2 = go.Figure(data=apple_data, layout=layout)
pio.write_image(plot2, 'static/LR.png')

```

LONG SHORT TERM MEMORY ALGORITHM

#Importing csv file

```
apple = pd.read_csv(r'D:\\xampp\\htdocs\\stockmarketprediction\\input\\'+ number +'.csv')
```

#Importing csv file

```
apple = pd.read_csv('D:\\xampp\\htdocs\\stockmarketprediction\\input\\'+ number +'.csv')
```

#First 5 rows

```
apple.head()
```

#Prints information about dataframe

```
apple.info()
```

#Remove rows that contain null values & #implicit conversion

```
apple["Close"]=pd.to_numeric(apple.Close,errors='coerce')
```

```
apple = apple.dropna()
```

```
trainData = apple.iloc[:,4:5].values
```

#Prints information about dataframe

```
apple.info()
```

#Rescale the data

```
sc = MinMaxScaler(feature_range=(0,1))
```

```
trainData = sc.fit_transform(trainData)
```

```
trainData.shape
```

#Dataflow training

```
x_train = []
```

```
y_train = []
```

#60 : timestep / 1258 : length of the data

for i in range (60,1149):

```
    x_train.append(trainData[i-60:i,0])
```

```
    y_train.append(trainData[i,0])
```

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

#adding the batch size axis

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

```
x_train.shape
```


#Building a ML model that contains four layers of LSTM network followed by dropout layer and using optimizer.

```
model = Sequential()

model.add(LSTM(units=100, return_sequences = True, input_shape
=(x_train.shape[1],1)))
model.add(Dropout(0.2))

model.add(LSTM(units=100, return_sequences = True))
model.add(Dropout(0.2))

model.add(LSTM(units=100, return_sequences = True))
model.add(Dropout(0.2))

model.add(LSTM(units=100, return_sequences = False))
model.add(Dropout(0.2))

model.add(Dense(units =1))
model.compile(optimizer='adam',loss="mean_squared_error")
```

#Training the data using epochs and batch size

```
hist = model.fit(x_train, y_train, epochs = 20, batch_size = 32, verbose=2)
```

#Visualize the loss that occurs during the epoch

```
plt.plot(hist.history['loss'])
plt.title("Training model loss")
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train'], loc='upper left')
```

#Importing csv file

```
appledata = pd.read_csv(r'D:\\xampp\\htdocs\\stockmarketprediction\\input\\'+ number
+'.csv')
```

#Importing csv file

```
appledata = pd.read_csv('D:\\xampp\\htdocs\\stockmarketprediction\\input\\'+ number
+'.csv')
```

#Remove rows that contain null values & #implicit conversion

```
appledata["Close"]=pd.to_numeric(appledata.Close,errors='coerce')
appledata = appledata.dropna()
appledata = appledata.iloc[:,4:5]
y_test = appledata.iloc[60:,0:].values
```

```

#input array for the model
#Converting x_test data into numpy array and printing it's shape
inputClosing = appledata.iloc[:,0:].values
inputClosing_scaled = sc.transform(inputClosing)
inputClosing_scaled.shape
x_test = []
length = len(appledata)
timestep = 60
for i in range(timestep,length):
    x_test.append(inputClosing_scaled[i-timestep:i,0])
x_test = np.array(x_test)
x_test = np.reshape(x_test,(x_test.shape[0],x_test.shape[1],1))
x_test.shape

#Predicting the model and passing the x_test data
y_pred = model.predict(x_test)
y_pred

#Plot the data between actual and predicted prices
predicted_price = sc.inverse_transform(y_pred)

plt.clf()
plt.cla()

#Plot the graph and visualize the actual and predicted stock prices
plt.plot(y_test, color = 'red', label = 'Actual Stock Price')
plt.plot(predicted_price, color = 'blue', label = 'Predicted Stock Price')
plt.title(number+'-Stock Price Prediction By LSTM Model')
plt.xlabel('Time')
plt.ylabel('Stock Price')
plt.savefig('static/LS.png')

```

ARIMA ALGORITHM

#Importing csv file

```
apple = pd.read_csv(r'D:\\xampp\\htdocs\\stockmarketprediction\\input\\'+ number +'.csv')
```

#Importing csv file

```
apple = pd.read_csv('D:\\xampp\\htdocs\\stockmarketprediction\\input\\'+ number +'.csv')
```

#First 5 rows

```
apple.head()
```

#Let's check if there is some cross-correlation in out data.

```
plt.figure()
```

```
lag_plot(apple['Open'], lag=3)
```

```
plt.title('Google Stock - Autocorrelation plot with lag = 3')
```

#Plotting the stock price evolution over time.

```
plt.plot(apple["Date"], apple["Close"])
```

```
plt.xticks(np.arange(0,1260, 300), apple['Date'][0:1260:300])
```

```
plt.title("Stock price over time")
```

```
plt.xlabel("time")
```

```
plt.ylabel("price")
```

#Building the predictive ARIMA model

#training(70%) and testing(30%)

#default arima parameters p=4,d=1,q=0

```
train_data, test_data = apple[0:int(len(apple)*0.7)], apple[int(len(apple)*0.7):]
```

```
training_data = train_data['Close'].values
```

```
test_data = test_data['Close'].values
```

```
history = [x for x in training_data]
```

```
model_predictions = []
```

```
N_test_observations = len(test_data)
```

```
for time_point in range(N_test_observations):
```

```
    model = ARIMA(history, order=(4,1,0))
```

```
    model_fit = model.fit()
```

```
    output = model_fit.forecast()
```

```
    yhat = output[0]
```

```
    model_predictions.append(yhat)
```

```
    true_test_value = test_data[time_point]
```

```
    history.append(true_test_value)
```

```
MSE_error = mean_squared_error(test_data, model_predictions)
```

```
print("Testing Mean Squared Error is {}".format(MSE_error))
```

```
#Plot the graph and visualize the actual and predicted stock prices
test_set_range = apple[int(len(apple)*0.7):].index
plt.plot(test_set_range, model_predictions, color='blue', marker='o',
linestyle='dashed',label='Predicted Price')
plt.plot(test_set_range, test_data, color='red', label='Actual Price')
plt.title(number+'-Stock Price Prediction By Arima Model')
plt.xlabel('Time')
plt.ylabel('Stock Price')
plt.legend()
plt.savefig('static/Arima.png')
plt.clf()
plt.cla()
```

SENTIMENT ANALYSIS OF TWEETS

#Importing the csv file

```
twitter = pd.read_csv('D:\\xampp\\htdocs\\stockmarketprediction\\input\\twitter\\'+ number  
+'.csv')
```

#Print the variable

```
print(twitter)
```

#First 5 rows

```
twitter.head()
```

#Cleaning up tweets using regular expression.

```
def cleanUpTweet(txt):
```

```
    txt = re.sub(r'@[A-Za-z0-9_]+', '', txt)
```

```
    txt = re.sub(r'#', '', txt)
```

```
    txt = re.sub(r'RT : ', '', txt)
```

```
    txt = re.sub(r'https?:\.[A-Za-z0-9\.\.]+', '', txt)
```

```
    return txt
```

#Cleaned up tweets

```
twitter['Tweet']=twitter['Tweet'].apply(cleanUpTweet)
```

#Using the textblob

```
def getTextSubjectivity(txt):
```

```
    return TextBlob(txt).sentiment.subjectivity
```

#Using the textblob

```
def getTextPolarity(txt):
```

```
    return TextBlob(txt).sentiment.polarity
```

#Getting sub and pol for tweet

```
twitter['Subjectivity']=twitter['Tweet'].apply(getTextSubjectivity)
```

```
twitter['Polarity']=twitter['Tweet'].apply(getTextPolarity)
```

#First 20 rows

```
twitter.head(20)
```

#Removing the column

```
twitter = twitter.drop(twitter[twitter['Tweet']=="].index)
```

#First 20 rows

```
twitter.head(20)
```

```

#TextAnalysis
def getTextAnalysis(a):
    if a<0:
        return "Negative"
    elif a==0:
        return "Neutral"
    else:
        return "Positive"

#Getting the textanalysis
twitter["Score"]=twitter['Polarity'].apply(getTextAnalysis)

#First 20 rows
twitter.head(20)

#Calculating % of positive review
positive=twitter[twitter['Score']=="Positive"]
print(str(positive.shape[0]/(twitter.shape[0])*100)+"% of positive tweets")
pos=positive.shape[0]/twitter.shape[0]*100

#Calculating % of nrgative review
negative=twitter[twitter['Score']=="Negative"]
print(str(negative.shape[0]/(twitter.shape[0])*100)+"% of negative tweets")
neg=negative.shape[0]/twitter.shape[0]*100

#Calculating % of neutral review
neutral=twitter[twitter['Score']=="Neutral"]
print(str(neutral.shape[0]/(twitter.shape[0])*100)+"% of neutral tweets")
neutrall=neutral.shape[0]/twitter.shape[0]*100

#Show graph using pie-chart
explode=(0,0.1,0)
labels='Positive','Negative','Neutral'
sizes=[pos,neg,neutrall]
colors=['yellowgreen','lightcoral','gold']

#Plotting the pie chart
plt.pie(sizes,explode=explode,colors=colors,autopct='%1.1f%%',startangle=140)
plt.legend(labels,loc=(-0.05,0.05),shadow = True)
plt.axis('equal')
plt.title(number+'-Sentiment Analysis Of Tweets')
plt.savefig('static/tweet.png')

return render_template("calculate.html")

if __name__ == '__main__':
    app.run()

```

5.3 Testing Approaches

5.3.1 Unit Testing :-

Unit Testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually scrutinized for proper operation. Software developers and sometimes QA staff complete unit tests during the development process.

It is concerned with functional correctness of the standalone modules. Unit testing is performed on each module of the system i.e. login page, registration page , home page, prediction page and the final one is the prediction output page.

5.3.2 Integration Testing :-

In integration testing all the modules are combined together and executed. This helps to understand if the all the modules work accordingly and is error free and runs in the required sequence i.e. as the user logs into the account the home page of the system opens which consists of the buttons for the a fore mentioned modules of the system.

The user can select any of the modules, when a particular module is selected the system takes the user to that particular module page i.e. the stock name of the company will be inserted and after clicking on the prediction button it will show the proper machine learning model output on the same page by using the code of machine learning algorithm.

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements.

CHAPTER 6 : RESULTS AND DISCUSSIONS

6.1 Test Reports

Test Report is a document which contains a summary of all test activities and final test results of a testing project.

Test report is an assessment of how well the Testing is performed. Based on the test report, stakeholders can evaluate the quality of the tested product and make a decision on the software release.

Depending on the test cases we can drive that the outcome of the test cases is positive and the system developed is successful in performing the required functions.

All the functionalities are cross checked with valid and invalid inputs and with the help on those outputs this test report is derived. If any invalid inputs are entered the system alters the user to enter valid information.

Multiple combinations of inputs (valid, invalid) have been entered to check systems accuracy and its response.

If the valid input is enter no error message is displayed and with invalid input error message is displayed, this helps the user to enter the necessary and required information.

Results :-

Mean Squared Error (MSE) & Accuracy.

The Mean Squared Error (MSE) and the Accuracy can be differ by one another by downloading specific historical prices of the stock company in which the time period may changes from any year to the current year and the frequency can be kept depending on the user such as on daily , weekly or monthly requirements.

CHAPTER 7 : CONCLUSIONS

7.1 Conclusion

We investigated how sentiment analysis of the twitter data is correlated to the prediction of the stock market price for all the companies which are taken. The result obtained after the prediction process clearly specifies that, we have obtained the accurate value which matches with the actual and predicted stock price appropriately.

Thus, social media such as twitter can be used as a source to predict the stock market price with maximum accuracy. Furthermore, the machine learning model such as ARIMA , LSTM , LINEAR REGRESSION provides more accurate values compared with other models. Thus, using sentiment analysis of twitter data and stock data from yahoo finance API, we predict the stock market price which is helpful for predicting future stock price. In the future, we plan to further improve the work in the following areas.

Our analysis is limited to 5 companies. An expansion to broader set of companies or all Twitter data might yield more insights into the data, leading to more effective application in stock price prediction. Finally, the current project examines correlation at daily granularity because the stock data are only available at the daily level. It will be interesting to study correlations at a finer granularity such as hourly.

7.2 Future Scope of the Project

The Future work regarding this study would include using the models on different stock markets across the world. Furthermore, using a data range of more than one year may provide more accurate results. Additionally, analyzing the models in different economic situations may allow us to better see the productivity of the models. Besides, the use of a neural network for classifying the sentimental analysis tweets API may offer better results. The stock market prediction can be done with the live forecasting in the upcoming days such as by using the tick symbol directly from yahoo finance website. This prediction will be live forecasting while using the machine learning algorithms. The Sentiment analysis in this project is being done by the offline data using the CSV files. So in the future work of the sentiment analysis can be done by using the live twitter data API, because it needs access from the twitter application developer to fetch the twitter data.

REFERENCES

- [1] <https://www.analyticsvidhya.com/blog/>
- [2] J. Bollen and H. Mao. Twitter mood as a stock market predictor. IEEE Computer, 44(10):91–94.
- [3] <https://machinelearningmastery.com/>
- [4] Fazel Zarandi M.H, Rezaee B, Turksen I.B and Neshat E. “A Type-2 Fuzzy Model for Stock Market Analysis.”, 2007.
- [5] <https://machinelearningmastery.com/>
- [6] V. S. Pagolu, K. N. Reddy, G. Panda and B. Majhi, “Sentiment analysis of Twitter data for predicting stock market movements,” “Proceedings of 2016 International Conference on Signal Processing, Communication, Power and Embedded System, pp. 1345-1350, 2016.
- [7] <https://www.kaggle.com/code/>
- [8] <https://finance.yahoo.com/>
- [9] L. Breiman, Random forests. Machine Learning, 45(1):5-32, 2001
- [10] <https://towardsdatascience.com/>
- [11] <https://www.geeksforgeeks.org/>
- [12] R. Goonatilake and S. Herath, The volatility of the stock market and news, International Research Journal of Finance and Economics, 2007, 11: 53-65.
- [13] Spandan Ghose Chowdhury, Soham Routh , Satyajit Chakrabarti, News Analytics and Sentiment Analysis to Predict Stock Price Trends.
- [14] <https://monkeylearn.com/blog/sentiment-analysis-of-twitter/>
- [15] <https://www.researchgate.net/StockMarketPredictionUsingTwitterSentimentAnalysis>