**ABSTRACT**

This Project includes the description of indicators which can be used for technical analysis of Indian market Nifty stocks. The indicators which have been used in this study are Moving Averages, Moving Averages cross rules and Moving Averages Convergence/Divergence. Later this study also includes the usage and application of Moving Average on Nifty stocks. Additionally, the analysis demonstrates that these indicators are the tools for successful trading and profit generation. An investor in the stock market would be interested in analysing the stock price movements. Prices in the stock market fluctuate due to continuous buying and selling in the market. There are basically two approaches used in analysing the share price movements. They are fundamental approach and technical approach. Both these approaches have the same objective of buying at lower price and selling at a higher price to gain good return on investment.

**CONTENTS**

1. **Introduction …………………………………………..**

Overview

Introduction

Scope

1. **Requirement analysis ………………………………….**

Existing System

Proposed System

System analysis and Requirement

Hardware Requirement

Software Requirement

1. **Software Requirement Specification .......................................**

Functional Requirement

Non-Functional Requirement

1. **Analysis and Design ……………………………………….**

Block diagram

Circuit diagram

1. **Implementation………………………………………………** Source Code

Snap Shot

1. **Testing …………………………………………………………** WhiteBoxing

Black Boxing

1. **Conclusion……………………………………………………**

**8. Reference …………………………………………………….**

# Chapter 1

# 1.1 INDRODUCTION

Data analysis is the process of gathering raw data and converting it into information that the users can use to make decisions.It entails inspecting, cleansing, transforming, and modeling data to uncover valuable information, draw conclusions, and aid decision-making.In today’s business world, data analysis plays an important role in making scientific decisions and assisting businesses in operating more efficiently.

Stock Market Analysis and Prediction is the project on technical analysis, visualization, and prediction using data provided by Google Finance. By looking at data from the stock market, particularly some giant technology stocks and others. Used pandas to get stock information, visualized different aspects of it, and finally looked at a few ways of analyzing the risk of a stock, based on its previous performance history. Predicted future stock prices through a Monte Carlo method.

## 1.2 Plan of Implementation

The project can be broken down into 7 main steps which are as follows:

1. Understand the dataset.
2. Clean the data.
3. Analyse the candidate columns to be Features.
4. Process the features as required by the model/algorithm.
5. Train the model/algorithm on training data.
6. Test the model/algorithm on testing data.
7. Tune the model/algorithm for higher accuracy.

## 1.3 Problem Statement

The stock market appears in the news every day. You 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.

Previous methods of stock predictions involve the use of Artificial Neural Networks and Convolution Neural Networks which has an error loss at an average of 20%.

In this report, we will see if there is a possibility of devising a model using Recurrent Neural Network which will predict stock price with a less percentage of error. And if the answer turns to be YES, we will also see how reliable and efficient will this model be.

## 1.4 Objective

This is a High-quality Indian stock exchange data of almost 20 years. It includes all giants in the Indian stock market of both BSE and NSE. It is very challenging to forecast the stock market even if large data is available.

This data is scrap after the large lockdown held in India due to the COVID-19 pandemic. This data is capable to cover the big recessions of the Indian market 2008 and 2020(COVID-19).

## 1.5 Existing System

There are two basic types of stock analysis: [fundamental analysis](https://www.investopedia.com/terms/f/fundamentalanalysis.asp) and [technical analysis](https://www.investopedia.com/terms/t/technicalanalysis.asp). Fundamental analysis concentrates on data from sources, including financial records, economic reports, company assets, and market share. To conduct fundamental analysis on a public company or sector, investors and analysts typically analyze the metrics on a company’s [financial statements](https://www.investopedia.com/terms/f/financial-statements.asp) – [balance sheet](https://www.investopedia.com/terms/b/balancesheet.asp), [income statement](https://www.investopedia.com/terms/i/incomestatement.asp), [cash flow statement](https://www.investopedia.com/terms/c/cashflowstatement.asp), and [footnotes](https://www.investopedia.com/terms/f/footnote.asp). These statements are released to the public in the form of a [10-Q](https://www.investopedia.com/terms/1/10q.asp) or [10-K report](https://www.investopedia.com/terms/1/10-k.asp) through the database system, [EDGAR](https://www.investopedia.com/terms/e/edgar.asp), which is administered by the U.S. [Securities and Exchange Commission](https://www.investopedia.com/terms/s/sec.asp) (SEC).

## 1.6 Proposed System

 It explores main concepts from basic to expert level which can help you achieve better grades, develop your academic career, apply your knowledge at work, or do research as an experienced investor. All of this while referencing the best practitioners in the field.

Become a Stock Technical Analysis Expert in this Practical Course with Python

* Read or download S&P 500® Index ETF prices data and perform technical analysis operations by installing related packages and running code on Python IDE.
* Compute lagging stock technical indicators or overlays such as moving averages, Bollinger bands, parabolic stop, and reverse.
* Calculate leading stock technical indicators or oscillators such as average directional movement index, commodity channel index, moving averages convergence/divergence, rate of change, relative strength index, stochastic oscillator, and Williams %R.
* Determine single technical indicator-based stock trading opportunities through price, double, bands, centerline, and signal crossovers.
* Define multiple technical indicators based on stock trading occasions through price crossovers confirmed by bands crossovers.
* Outline long (buy) or short (sell) stock trading strategies based on single or multiple technical indicators trading openings.
* Evaluate stock trading strategies performances by comparing them against the buy and hold benchmark.
* Calculate leading stock technical indicators or oscillators such as average directional movement index, commodity channel index, moving averages convergence/divergence, rate of change, relative strength index, stochastic oscillator, and Williams %R.
* Determine single technical indicator-based stock trading opportunities through price, double, bands, centerline, and signal crossovers.

# CHAPTER 2

# LITERATURE SURVEY

## The following papers were studied in order to get an overview of the techniques that were applied earlier to predict the stock market.

**LSTM Fully Convolutional Networks for Time Series Classification**

-Fazle Karim, Somshubra Majumdar, Houshang Darabi and Shun Chen [1]

With the proposed models, we achieve a potent improvement in the current state-of-the-art for time series classification using deep neural networks. Our baseline models, with and without fine-tuning, are trainable end-to-end with nominal preprocessing and are able to achieve significantly improved performance.

LSTM-FCNs are able to augment FCN models, appreciably increasing their performance with a nominal increase in the number of parameters. An LSTM-FCNs provide one with the ability to visually inspect the decision process of the LSTM RNN and provide a strong baseline on their own. Fine-tuning can be applied as a general procedure to a model to further elevate its performance.

**Learning Long term Dependencies with Gradient Descent is difficult**

-Yoshua bengio, Patrice Simard and Paolo Frasconi [10]

Recurrent networks are very powerful in their ability to represent context, often outperforming static network. But the factor off gradient descent of an error criterion may be inadequate to train them for a task involving long-term dependencies. It has been found that the system would not be robust to input noise or would not be efficiently trainable by gradient descent when the long-term context is required. The theoretical result presented in this paper holds for any error criterion and not only from mean square error.

**Improving N Calculation of the RSI Financial Indicator Using Neural Networks**

-Alejandro Rodríguez-González, Fernando Guldris Iglesias, Ricardo Colomo-Palacios Giner Alor-Hernandez, Ruben Posada-Gomez [8]

There has been growing interest in Trading Decision Support Systems in recent years. In spite of its volatility, it is not entirely random, instead, it is nonlinear and dynamic or highly complicated and volatile. Stock movement is affected by the mixture of two types of factors: determinant (e.g. gradual strength change between buying side and selling side) and random (e.g. emergent affairs or daily operation variations).

**Stock Trend Prediction Using Simple Moving Average Supported**

**by News Classification**

-Stefan Lauren Dra. Harlili S., M.Sc. [5]

The simple moving average is one of many time series analysis technique. Time series analysis is a method of timely structured data processing to find statistics or important characteristics for many reasons. The simple moving average shows stock trend by calculating the average value of stock prices on specific duration. The prices that are used are closing prices at the end of the day. This technique can avoid noises and therefore smooth the trend movement.

The main objective of financial news classification is to classify and calculate each news’ sentiment value. The positive news is marked by sentiment value which is greater than 0, while negative news is marked by less than 0 sentiment value. If there are news having 0 sentiment value, they will be omitted as their neutralism does not affect the stock trend.

**VISUALIZING AND UNDERSTANDING RECURRENT NETWORKS**

-Andrej Karpathy, Justin Johnson, Li Fei-Fei [4]

Character-level language models have been used as an interpretable test bed for analyzing the predictions, representations training dynamics, and error types present in Recurrent Neural Networks. In particular, the qualitative visualization experiments, cell activation statistics and comparisons to finite horizon n-gram models demonstrate that these networks learn powerfully, and often interpretable long-range interactions on real-world data.

that further architectural innovations may be needed to address the remaining errors.

**LSTM: A Search Space Odyssey**

-Klaus Greff, Rupesh K. Srivastava, Jan Koutn´ık, Bas R. Steunebrink, Jurgen Schmidhuber

This paper reports the results of a large-scale study on variants of the LSTM architecture. We conclude that the most commonly used LSTM architecture (vanilla LSTM) performs reasonably well on various datasets. None of the eight investigated modifications significantly improves performance.

The forget gate and the output activation function are the most critical components of the LSTM block. Removing any of them significantly impairs performance. We hypothesize that the output activation function is needed to prevent the unbounded cell state to propagate through the network and destabilize learning. This would explain why the LSTM variant GRU can perform reasonably well without it: its cell state is bounded because of the coupling of input and forget gate.

**The difficulty of training recurrent neural networks**

-Razvan Pascanu, Tomas Mikolov, Yoshua Bengio [6]

We provided different perspectives through which one can gain more insight into the exploding and vanishing gradients issue. We put forward a hypothesis stating that when gradients explode we have a cliff-like structure in the error surface and devise a simple solution based on this hypothesis, clipping the norm of the exploded gradients.

The effectiveness of our proposed solutions provides some indirect empirical evidence towards the validity of our hypothesis, though further investigations are required. In order to deal with the vanishing gradient problem, we use a regularization term that forces the error signal not to vanish as it travels back in time.

**Deep Sparse Rectifier Neural Networks**

-Xavier Glorot Antoine Bordes Yoshua Bengio [7]

Sparsity and neurons operating mostly in a linear regime can be brought together in more biologically plausible deep neural networks. Rectifier units help to bridge the gap between unsupervised pre-training and no pre-training, which suggests that they may help in finding better minima during training.

# Chapter 3: Data And Tools

## 3.1 Data Used

### 3.1.1 CHOOSING THE DATASET

For this project, we chose the Google stocks. The Google stocks is a large index traded on the New York stock exchange. All companies in the index are large publicly traded companies, leaders in each of their own sectors. The index covers a diverse set of sectors featuring companies such as Microsoft, Visa, Boeing, and Walt Disney. It is important to use a predefined set of companies rather than a custom selected set so that we do leave ourselves open to methodology errors or accusations of fishing expeditions. If we had selected a custom set of companies, it could be argued that the set was tailored specifically to improve our results. Since the aim of the project is to create a model of stock markets in general. Google was chosen because it is well known. The components provided a good balance between available data and computational feasibility.

### 3.1.2 GATHERING THE DATASETS

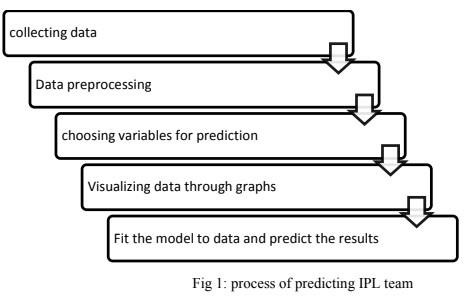
A primary dataset will be used throughout the project. The dataset will contain the daily percentage change in stock price. Luckily, daily stock price data is easy to come by. Google and Yahoo both operate websites which offer a facility to download CSV files containing a full 14 daily price history. These are useful for looking at individual companies but cumbersome when accessing large amounts of data across many stocks. For this reason, Quandl was used to gather the data instead of using Google and Yahoo directly. Quandl is a free to use website that hosts and maintains vast amounts of numerical datasets with a focus specifically on economic datasets, including stock market data which is backed by Google and Yahoo. Quandl also provides a small python library that is useful for accessing the database programmatically. The library provides a simple method for calculating the daily percentage change daily in prices.

. Quandl also provides a small python library that is useful for accessing the database programmatically. The library provides a simple method for calculating the daily percentage change daily in prices.

# Chapter 4

# APPROACH AND DESIGN

The below figure explains the approach we have taken into building the predictive model using machine learning algorithms.



## 4.1 Data Collection

Data collection is the process of gathering and measuring information from countless different sources. In order to use the data, we collect to develop practical machine learning solutions.

Collecting data allows you to capture a record of past events so that we can use data analysis to find recurring patterns. From those patterns, you build predictive models using machine learning algorithms that look for trends and predict future changes.

The Indian Stock Analysis official website is the principal basis of data for this project. The data was web scrapped from the website and kept in the appropriate format using a python library called beautiful soup. The dataset has the columns regarding match-number, Indian Stock Analysis season year, the place where match has been held and the stadium name, the match winner details, participating teams, the margin of winning and the umpire details, player of the match etc.

**4.2 Data Preprocessing**

### 4.2.1 Data cleaning

There are some null values in the dataset in the columns such as winner, city, venue etc. Due to the presence of these null values, the classification cannot be done accurately. So, we tried to replace the null values in different columns with dummy values.

### 4.2.2 Choosing Required Attributes

This step is the main part where we can eliminate some columns of the dataset that are not useful for the estimation of match winning team. This is estimated using feature importance. The considered attributes have the following feature importance.

### 

## 4.3 Data Visualization

* The data which has been collected is used for visualizing for the better understanding of the information.

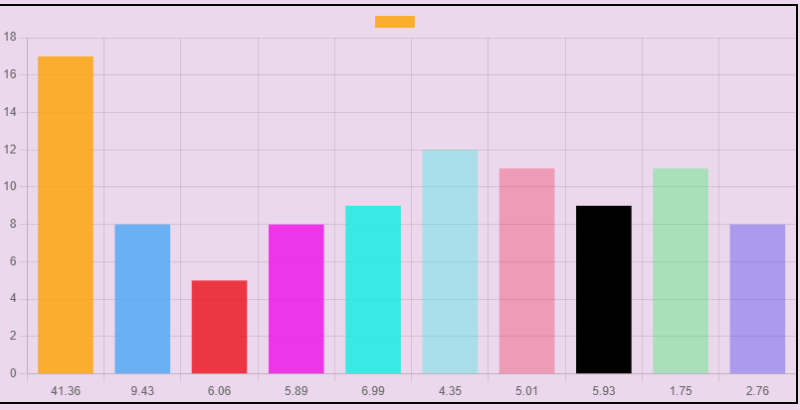


Fig 2 - Relation between toss winning and match winning

* Matplotlib Library is used here for visualizing the graphs

# Chapter 5:

# Integrated Summary

The most interesting task is to predict the market. So many methods are used for completing this task. Methods vary from very informal ways to many formal ways a lot. This tech. are categorized as:

- Prediction Methods  
- Traditional Time Series  
- Technical Analysis Methods  
- Machine Learning Methods  
- Fundamental Analysis Methods  
- Deep Learning

The criteria for this category are the kind of tool and the kind of data that these methods are consuming in order to predict the market. What is mutual to the technique is that they are predicting and hence helping the market's future behavior.

## 5.1 Technical Analysis Methods

Technical analysis is used to attempt to forecast the price movement of virtually any tradable instrument that is generally subject to forces of supply and demand, including stocks, bonds, futures and currency pairs. In fact, technical analysis can be viewed as simply the study of supply and demand forces as reflected in the market price movements of a security. It is most commonly applied to price changes, but some analysts may additionally track numbers other than just prices, such as trading volume or open interest figures.

Over the years, numerous technical indicators have been developed by analysts in attempts to accurately forecast future price movements. Some indicators are focused primarily on identifying the current market trend, including support and resistance areas, while others are focused on determining the strength of a trend and the likelihood of its continuation. Commonly used technical indicators include trendlines, moving averages and momentum indicators such as the moving average convergence divergence (MACD) indicator.

## 5.2 Fundamental Analysis Techniques

Fundamental analysis uses real, public data in the evaluation a security's value. Although most analysts use fundamental analysis to value stocks, this method of valuation can be used for just about any type of security. For example, an investor can perform fundamental analysis on a bond's value by looking at economic factors such as interest rates and the overall state of the economy. He can also look at information about the bond issuer, such as potential changes in credit ratings.

For stocks and equity instruments, this method uses revenues, earnings, future growth, return on equity, profit margins, and other data to determine a company's underlying value and potential for future growth. In terms of stocks, fundamental analysis focuses on the financial statements of the company being evaluated. One of the most famous and successful fundamental analysts is the so-called "Oracle of Omaha", Warren Buffett, who is well known for successfully employing fundamental analysis to pick securities. His abilities have turned him into a billionaire.

## 5.3 Traditional Time Series Prediction

Time series analysis can be useful to see how a given asset, security or economic variable changes over time. It can also be used to examine how the changes associated with the chosen data point compare to shifts in other variables over the same time period.

For example, suppose you wanted to analyze a time series of daily closing stock prices for a given stock over a period of one year. You would obtain a list of all the closing prices for the stock from each day for the past year and list them in chronological order. This would be a one-year daily closing price time series for the stock.

Delving a bit deeper, you might be interested to know whether the stock's time series shows any seasonality to determine if it goes through peaks and valleys at regular times each year. The analysis in this area would require taking the observed prices and correlating them to a chosen season. This can include traditional calendar seasons, such as summer and winter, or retail seasons, such as holiday seasons.

**5.4 Machine Learning Methods**

Various sectors of the economy are dealing with huge amounts of data available in different formats from disparate sources. The enormous amount of data, known as Big Data, is becoming easily available and accessible due to the progressive use of technology. Companies and governments realize the huge insights that can be gained from tapping into big data but lack the resources and time required to comb through its wealth of information. In this regard, Artificial Intelligence (AI) measures are being employed by different industries to gather, process, communicate and share useful information from data sets. One method of AI that is increasingly utilized for big data processing is Machine Learning.

The various data applications of machine learning are formed through a complex algorithm or source code built into the machine or computer. This programming code creates a model which identifies the data and builds predictions around the data it identifies. The model uses parameters built into the algorithm to form patterns for its decision-making process. When new or additional data becomes available, the algorithm automatically adjusts the parameters to check for a pattern change, if any. However, the model shouldn’t change.

How machine learning works can be better explained by an illustration in the financial world. Traditionally, investment players in the securities market like financial researchers, analysts, asset managers, individual investors scour through a lot of information from different companies around the world to make profitable investment decisions. However, some pertinent information may not be widely publicized by the media and may be privy to only a select few who have the advantage of being employees of the company or residents of the country where the information stems from. In addition, there’s only so much information humans can collect and process within a given time frame. This is where machine learning comes in.

## 5.5 DEEP LEARNING

An artificial intelligence function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in Artificial Intelligence (AI) that has networks which are capable of learning unsupervised from data that is unstructured or unlabeled. Also known as Deep Neural Learning or Deep Neural Network.

BREAKING DOWN 'Deep Learning'

The digital era has brought about an explosion of data in all forms and from every region of the world. This data, known simply as Big Data, is gotten from sources like social media, internet search engines, e-commerce platforms, online cinemas, etc. This enormous amount of data is readily accessible and can be shared through fine tech applications like cloud computing. However, the data, which normally is unstructured, is so vast that it could take decades for humans to comprehend it and extract relevant information. Companies realize the incredible potential that can result from unraveling this wealth of information and are increasingly adapting to Artificial Intelligence (AI) systems for automated support.

One of the most common AI techniques used for processing Big Data is Machine Learning. Machine learning is a self-adaptive algorithm that gets better and better analysis and patterns with experience or with newly added data. If a digital payments company wanted to detect the occurrence of or potential for fraud in its system, it could employ machine learning tools for this purpose.

## 5.5.1 ARTIFICIAL NEURAL NETWORKS (ANN)

A computing system that is designed to simulate the way the human brain analyzes and process information. Artificial Neural Networks (ANN) is the foundation of Artificial Intelligence (AI) and solves problems that would prove impossible or difficult by human or statistical standards. ANN has self-learning capabilities that enable it to produce better results as more data becomes available.

# Chapter 6

# SYSTEM REQUIREMENT SPECIFICATION

A System Requirement Specification (SRS) is basically an organization’s understanding of a customer or potential client’s system requirements and dependencies at a particular point prior to any actual design or development work. The information gathered during the analysis is translated into a document that defines a set of requirements. It gives the brief description of the services that the system should provide and also the constraints under which, the system should operate. Generally, SRS is a document that completely describes what the proposed software should do without describing how the software will do it. It’s a two-way insurance policy that assures that both the client and the organization understand the other’s requirements from that perspective at a given point in time.

SRS document itself states in precise and explicit language those functions and capabilities a software system (i.e., a software application, an ecommerce website and so on) must provide, as well as states any required constraints by which the system must abide. SRS also functions as a blueprint for completing a project with as little cost growth as possible. SRS is often referred to as the “parent” document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it.

Requirement is a condition or capability to which the system must conform. Requirement Management is a systematic approach towards eliciting, organizing and documenting the requirements of the system clearly along with the applicable attributes. The elusive difficulties of requirements are not always obvious and can come from any number of sources.

## 6.1 Functional Requirements

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality.

Following are the functional requirements on the system:

The whole process can be handled at minimal human interaction with android and web both.

The application automatically receives the captured data from server.

## 6.2 Non Functional Requirement

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviours. They may relate to emergent system properties such as reliability, response time and store occupancy. Non-functional requirements arise through the user needs, because of budget constraints, organizational policies, the need for interoperability with other software and hardware systems or because of external factors such as :-

* Performance Requirements
* Design Requirements
* Security Constraints
* Basic Operational Requirements

### 6.2.1 Product Requirements

Platform independency: A progressive web app will be developed and deployed so that users with a smartphone or a computer can access the voting site to cast their vote.

Ease of use: The progressive web app provides an interface which is easy to use and eliminates the need for the voter to go to a voting booth.

Modularity: The complete product is broken up into modules and welldefined interfaces are developed to explore the benefit of flexibility of the product.

Robustness: This software is being developed in such a way that the overall performance is optimized, and the user can expect the results within a limited time with utmost relevancy and correctness.

## 6.3 System Configuration

### 6.3.1 H/W System Configuration:

* **Processor** - Pentium – IV
* **Speed** - 1.1 GHz

|  |  |
| --- | --- |
| • **RAM** | - 256 MB (min) |
| •  **Hard Disk** | - 20 GB |

### 5.3.2 S/W System Configuration:

* **Operating System** - XP/7/8/8.1/10
* **Coding Language** - Python, HTML, JavaScript

## 5.4 Hardware Requirements

• **Processors** **-** Pentium IV Processor

|  |  |  |  |
| --- | --- | --- | --- |
| **•** | **Speed** |  | **-** 3.00 GHZ |
| **•** | **RAM** |  | **-** 2 GB |
| • | **Storage** |  | **-** 20 GB |

## 5.5 Software Requirements

* **Operating system** **-** Windows 10 Professional.
* **IDE** **-** Visual Studio.
* **Front End -** Django.
* **Back End** - Chart Js, NumPy, Pandas, Papa Parse.

# Chapter 6

# SYSTEM DESIGN

Design is a meaningful engineering representation of something that is to be built. It is the most crucial phase in the developments of a system. Software design is a process through which the requirements are translated into a representation of software. Design is a place where design is fostered in software Engineering. Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. Design is the perfect way to accurately translate a customer’s requirement in the finished software product. Design creates a representation or model, provides details about software data structure, architecture, interfaces and components that are necessary to implement a system. The logical system design arrived at as a result of systems analysis is converted into physical system design.

## 6.1 System development methodology

System development method is a process through which a product will get completed or a product gets rid from any problem. Software development process is described as a number of phases, procedures and steps that gives the complete software. It follows series of steps which is used for product progress. The development method followed in this project is waterfall model.

### 6.1.1 Model phases

The waterfall model is a successive programming improvement process, in which advance is seen as streaming relentlessly downwards (like a waterfall) through the periods of Requirement start, Analysis, Design, Implementation, Testing and upkeep.

**Prerequisite Analysis:** This stage is worried about gathering of necessity of the framework.

This procedure includes producing record and necessity survey.

**Framework Design:** Keeping the prerequisites at the top of the priority list the framework details are made an interpretation of into a product representation. In this stage the fashioner underlines on calculation, information structure, programming design and so on.

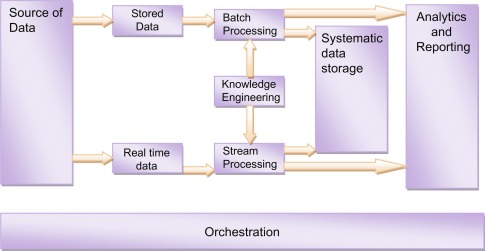
**Coding:** In this stage developer begins his coding with a specific end goal to give a full portray of item. At the end of the day framework particulars are just changed over into machine coherent register code.

**Usage:** The execution stage includes the genuine coding or programming of the product. The yield of this stage is regularly the library, executables, client manuals and extra programming documentation.

**Testing:** In this stage all projects (models) are coordinated and tried to guarantee that the complete framework meets the product prerequisites. The testing is worried with check and approval.

**Support:** The upkeep stage is the longest stage in which the product is upgraded to satisfy the changing client need, adjust to suit change in the outside environment, right mistakes and oversights beforehand undetected in the testing stage, improve the proficiency of the product.

### 6.1.2 System Architecture



# Chapter 7

# IMPLEMENTATION

The goals of the implementation phase is to translate the design of the system produce during the phase, into coded form in a given programming language, which can then be executed by a computer performing the computation specified by the design the coding phase affects both testing and maintenance profoundly. Well written code can reduce the testing and maintains cost.

A crucial phase in the system lifecycle is the successful implementation of the system design. Implementation simply means converting the system designs into operation. Implementation is the process of bringing the developed system into operational use and providing it to the user.

This stage is considered to be most crucial stage in the development of a successful system since a new system is developed and the users are get information in effective manner

Implementation is a stage in which the design is converted into working system that is it is the stage of the project where theoretical design is turned into a working system .The implementation involves careful planning, investing of the current system and its constraint on implementation, design of methods to achieve the changeover.

The Project is implemented in different phases as follows

* First phase includes table design for database module.
* Second phase includes coding for modules.
* Third phase includes the integration of modules.
* Fourth phase includes connection establishment between the front end and back end.
* Fifth phase includes error handling and message generator.
* First phase includes table design for database module.
* Second phase includes coding for modules.
* Third phase includes the integration of modules.
* Fourth phase includes connection establishment between the front end and back end.
* Fifth phase includes error handling and message generator.

The coding was done with the following characteristics in mind

* Code efficiency
* Memory efficiency
* Response time
* Security
* Maintainability
* Efficient and consistent logic

# Chapter 8

# Coding

**Views.py**

import contextlib

from django.shortcuts import render, HttpResponse, redirect

from django.core.files.storage import FileSystemStorage

from django.contrib import messages

from collections import Counter

import os

import pandas as pd

import numpy as np

from pandas.\_libs import missing

from pandas.core.indexing import check\_bool\_indexer

# Create your views here.

def index(request):

    global column\_id,clean\_data,file\_directory

    # return HttpResponse('This is index Page')

    if request.method == 'POST':

        uploaded\_files=request.FILES['choose\_csv']

        clean\_data = request.POST.get('clean\_data')

        print(clean\_data)

        print(uploaded\_files)

        if uploaded\_files.name.endswith('csv'):

            save\_file=FileSystemStorage()

            name\_file=save\_file.save(uploaded\_files.name, uploaded\_files) #this is name of the file

            current\_directory=os.getcwd()    #To get the current directory of the folder

            file\_directory=current\_directory+'\media\\'+name\_file

            readFile(file\_directory)

            return cleanData(request)

    return render(request,'index.html')

def readFile(filename):

    global rows,columns,my\_file,data,missing\_values,data\_html

    my\_file = pd.read\_csv(filename, sep=',',engine='python')       #using to read the csv file with columns seperators of delimitor(,)

    data=pd.DataFrame(data=my\_file,index=None)

    data\_html = data.to\_html()

    context = {'loaded\_data': data\_html,}

    print(data)

    #rows and columns

    rows=len(data.axes[0])              # to count number of rows

    columns=len(data.axes[1])            # to count number of columns

    # finding missing values

    missing\_signs = ['?','0','--']

    null\_data = data[data.isnull().any(axis=1)]             # To finding the null values

    missing\_values=len(null\_data)

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Function for cleaning the Data  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

def cleanData(request):

    null\_message = "found " + str(rows) + " rows " + str(columns) + " and columns.\nMissing data are: " + str(missing\_values)

    context = {

        'loaded\_data': data\_html,

        'cleanMessage':null\_message,

    }

    messages.warning(request, null\_message)

    return render(request,'index.html')

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Function for Graph Plotting  in graph\_page.html \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

def graphPage(request):

    global column\_id,clean\_data,file\_directory

    if request.method == 'POST':

        uploaded\_files=request.FILES['choose\_csv']

        column\_id = request.POST.get('column\_id')

        print(uploaded\_files)

        print(column\_id)

        if uploaded\_files.name.endswith('csv'):

            save\_file=FileSystemStorage()

            name\_file=save\_file.save(uploaded\_files.name, uploaded\_files) #this is name of the file

            current\_directory=os.getcwd()    #To get the current directory of the folder

            file\_directory=current\_directory+'\media\\'+name\_file

            readFile(file\_directory)

            return graphPlotPage(request)

    return render(request,'graph\_page.html')

def readFile(filename):

    global rows,columns,my\_file,data,missing\_values,data\_html

    my\_file = pd.read\_csv(filename, sep=',',engine='python')       #using to read the csv file with columns seperators of delimitor(,)

    data=pd.DataFrame(data=my\_file,index=None)

    data\_html = data.to\_html()

    context = {'loaded\_data': data\_html,}

    print(data)

    #rows and columns

    rows=len(data.axes[0])              # to count number of rows

    columns=len(data.axes[1])            # to count number of columns

    # finding missing values

    missing\_signs = ['?','0','--']

    null\_data = data[data.isnull().any(axis=1)]             # To finding the null values

    missing\_values=len(null\_data)

def graphPlotPage(request):

    dashboard = []

    for x in data[column\_id]:           # Go to the specified input column\_id row by row and saved this in the dashboard array[]

        dashboard.append(x)

    my\_dashboard = dict(Counter(dashboard))

    print('my dashboard',my\_dashboard)

    keys = my\_dashboard.keys()

    values = my\_dashboard.values()

    listkeys = []

    listvalues = []

    for x in keys:

        listkeys.append(x)

    for y in values:

        listvalues.append(y)

    context = {

        'listkeys':listkeys,

        'listvalues':listvalues,

    }

    return render(request,'graph\_page.html',context)

# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Function for Displaying the CSV Table Rows  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

def TableRows(request):

    global column\_id,clean\_data,file\_directory

    if request.method == 'POST':

        uploaded\_files=request.FILES['choose\_csv']

        numrows = request.POST.get('numrows')

        numrows=int(numrows)

        print(numrows)

        if uploaded\_files.name.endswith('csv'):

            save\_file=FileSystemStorage()

            name\_file=save\_file.save(uploaded\_files.name, uploaded\_files) #this is name of the file

            current\_directory=os.getcwd()    #To get the current directory of the folder

            file\_directory=current\_directory+'\media\\'+name\_file

            my\_file = pd.read\_csv(file\_directory, sep=',',engine='python')       #using to read the csv file with columns seperators of delimitor(,)

            data=pd.DataFrame(data=my\_file,index=None)

            # data\_html = data.to\_html()

            firstRows=data.head(numrows).to\_html()

            lastRows=data.tail(5).to\_html()

            context={

                'firstRow':firstRows,

                'lastRow':lastRows,

            }

            return HttpResponse(firstRows)

    return render(request,'table.html')

def AnalyzeData(request):

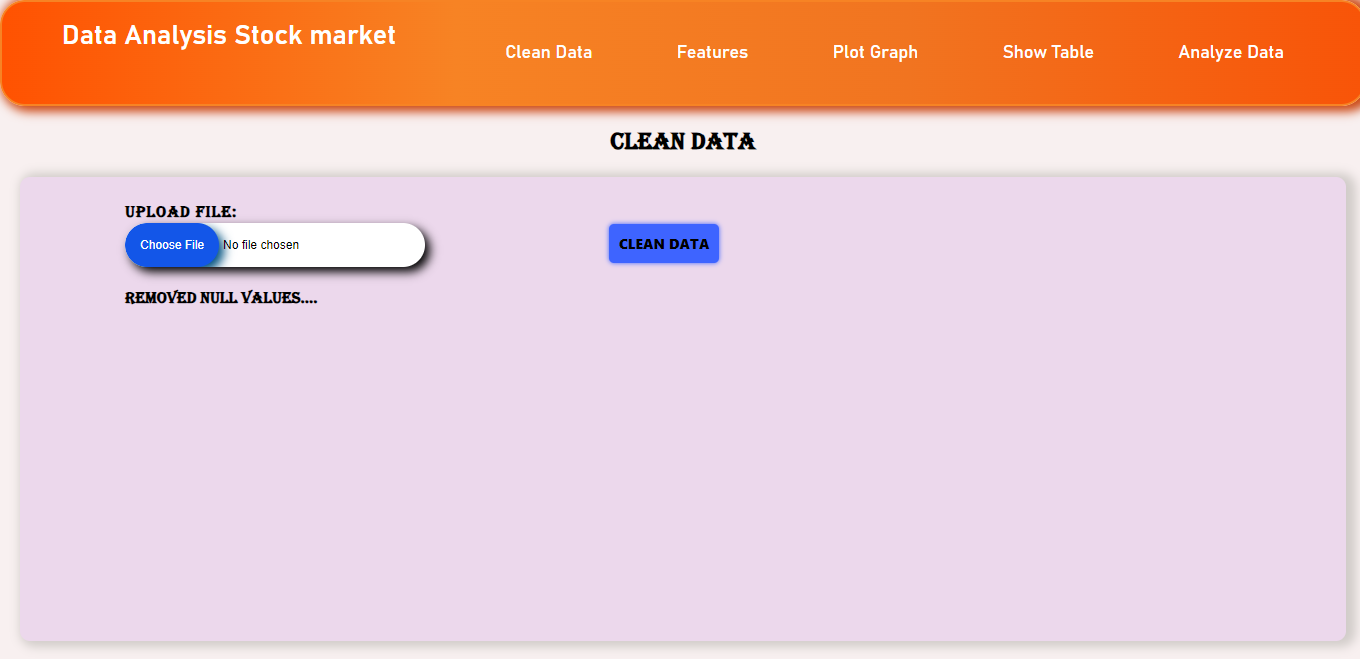
    return render(request,'analyseData.html')

def UiDesign(request):

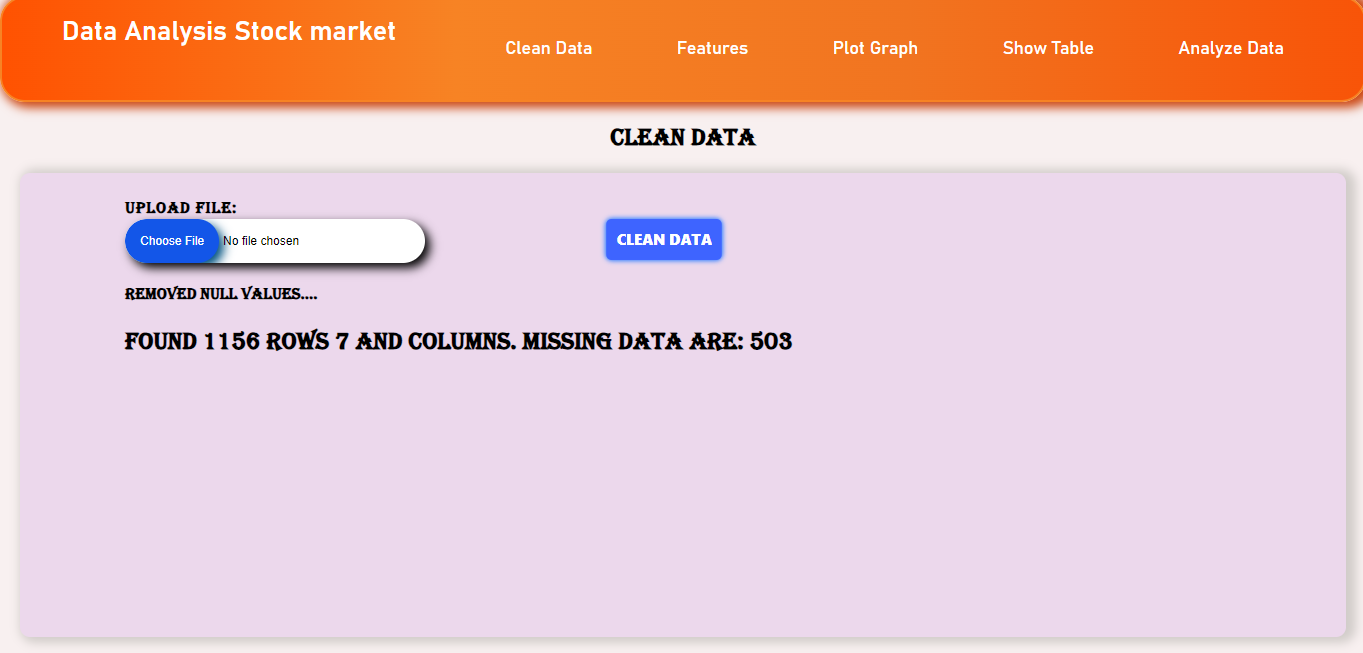
    return HttpResponse(request,'uiDesign.html')

# Chapter 8

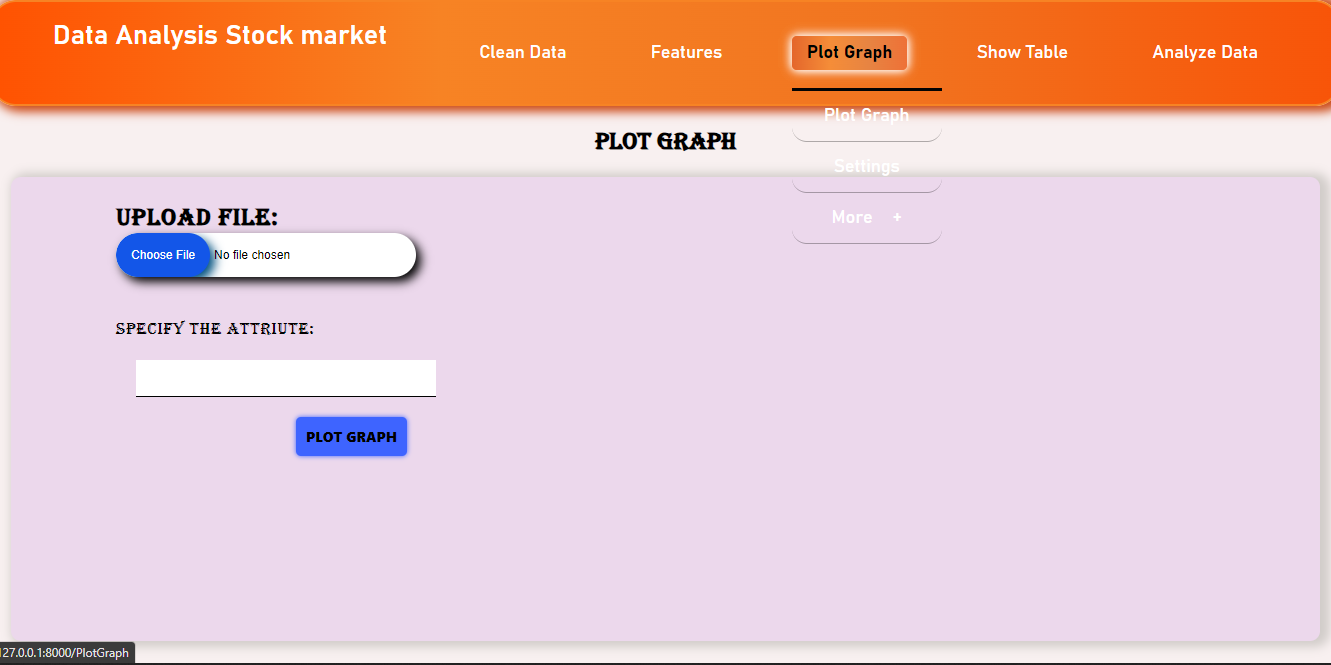
# SnapShots:

**Index Page** ****

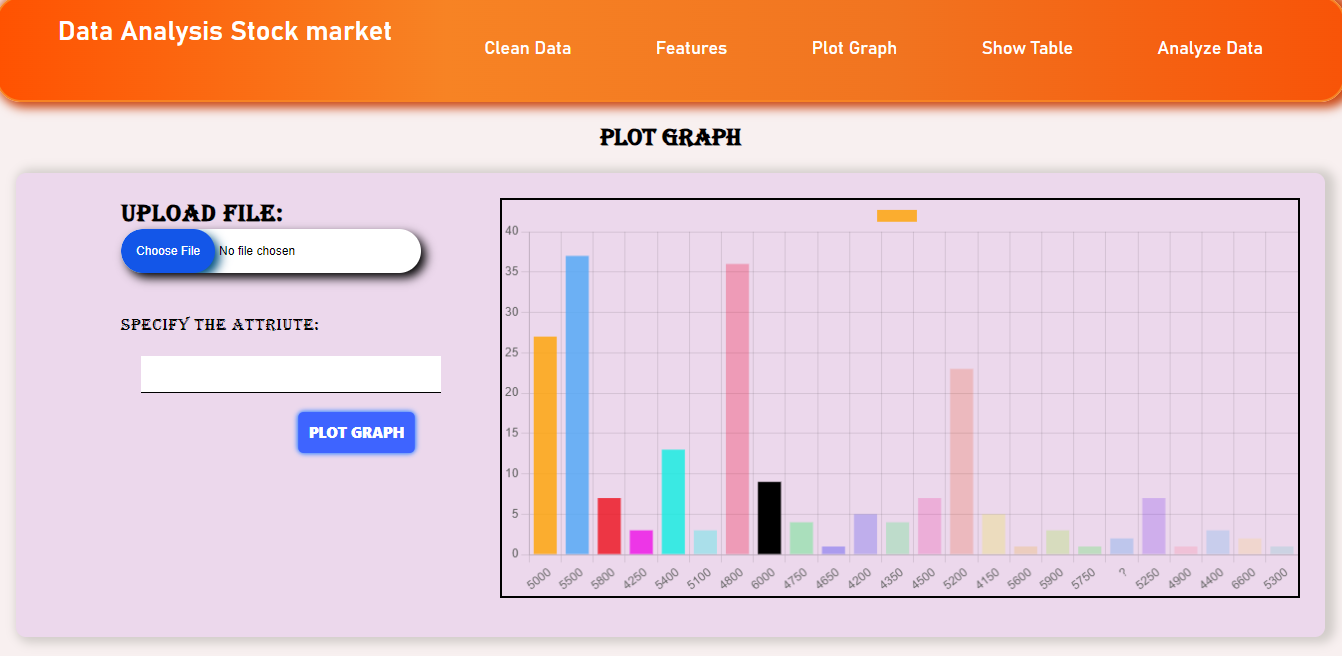
**Clean Data**

****

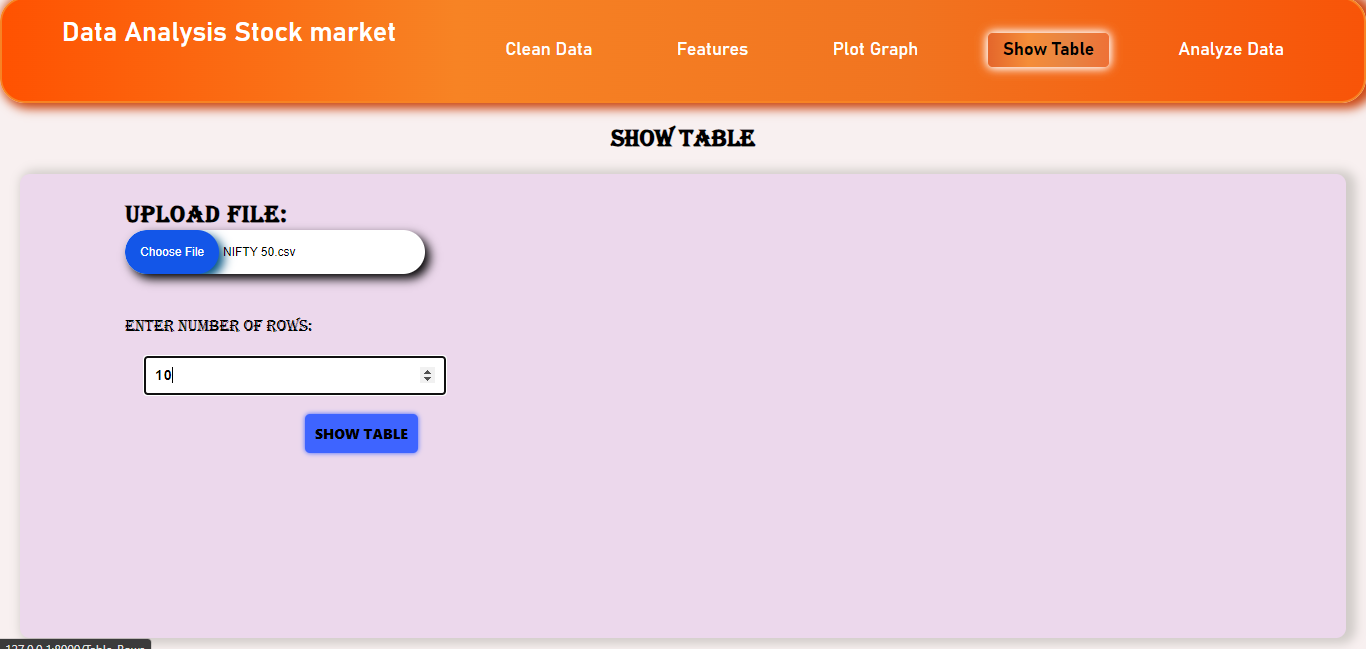
**Plot Graph:**

****

**Plot Graph (**Type Column name**):**

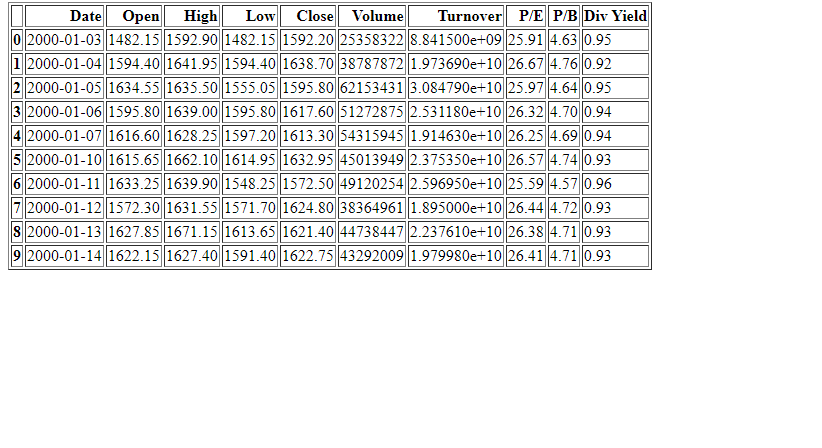
****

**Show Table:**

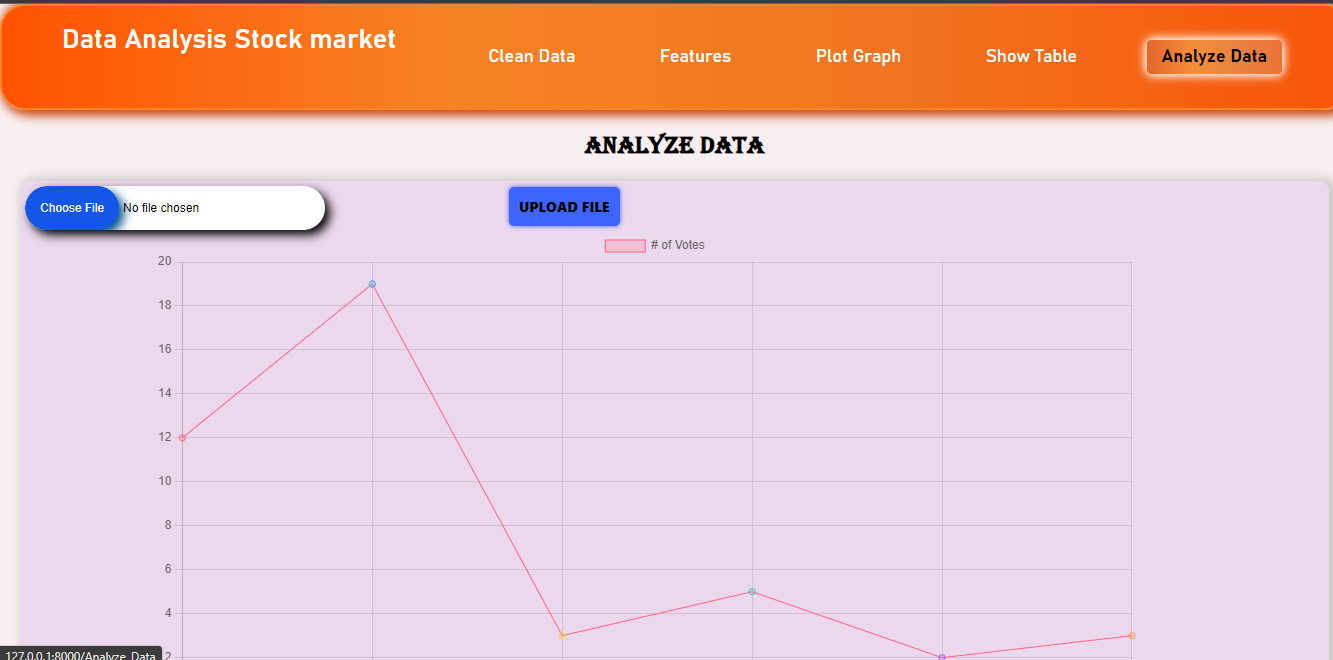
****

**Show Table:**

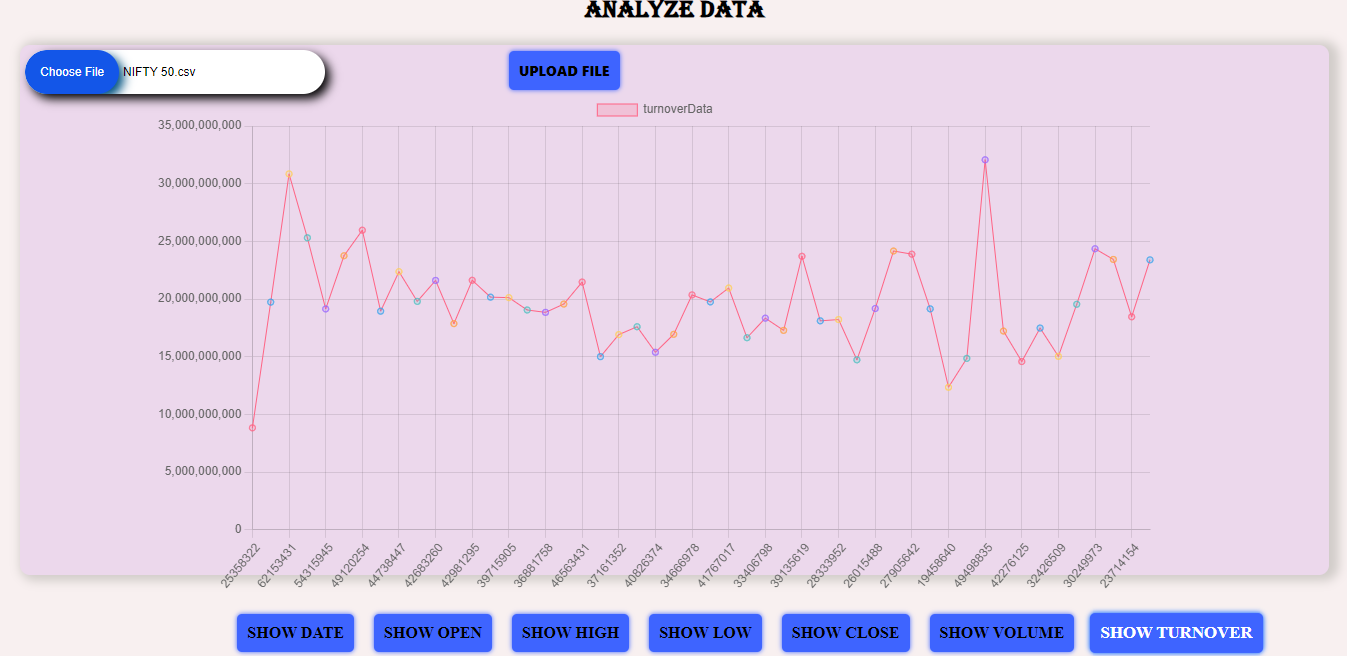
Specify Column Attribute



**Analyze Data:**

****

**TurnOver**

****

# Chapter 9

# Conclusion:

Financial markets provide a unique platform for trading and investing, where trades can be executed from any device that can connect to the Internet. With the advent of stock markets, people have the opportunity to have multiple avenues to make their investment grow. Not only that, but it also gave rise to different types of funds like mutual funds, hedge funds and index funds for people and institutions to invest money according to their risk appetite. Governments of most countries invest a part of their healthcare, employment, or retirement funds into stock markets to achieve better returns for everyone. Online trading services have already revolutionised the way people buy and sell stocks. The financial markets have evolved rapidly into a strong and interconnected global marketplace. These advancements bring forth new opportunities and the data science techniques offer many advantages, but they also pose a whole set of new challenges. In this paper, we propose a taxonomy of computational approaches to stock market analysis and prediction, present a detailed literature study of the state-of-the-art algorithms and methods that are commonly applied to stock market prediction, and discuss some of the continuing challenges in this area that require more attention and provide opportunities for future development and research. Unlike traditional systems, stock market today are built using a combination of different technologies, such as machine learning, expert systems, and big data which communicate with one another to facilitate more informed decisions. At the same time, global user connectivity on the internet has rendered the stock market susceptible to customer sentiments, less stable due to developing news, and prone to malicious attacks. This is where further research can play an important role in paving the way how stock markets will be analysed and made more robust in the future. A promising research direction is to explore various algorithms to evaluate whether they are powerful enough to predict for the longer term, because markets act like weighing machines in the long run having less noise and more predictability. Hybrid approaches that combine statistical and machine learning techniques will probably prove to be more useful for stock prediction.

# References:

* Abu-Mostafa, Yaser S., and Amir F. Atiya. 1996. Introduction to financial forecasting. Applied Intelligence 6: 205–13.
* Arévalo, Rubén, Jorge García, Francisco Guijarro, and Alfred Peris. 2017. A dynamic trading rule based on filtered flag pattern recognition for stock market price forecasting. Expert Systems with Applications 81: 177–92.
* Babu, M. Suresh, N. Geethanjali, and B. Satyanarayana. 2012. Clustering Approach to Stock Market Prediction. International Journal of Advanced Networking and Applications 3: 1281
* Ballings, Michel, Dirk Van den Poel, Nathalie Hespeels, and Ruben Gryp. 2015. Evaluating multiple classifiers for stock price direction prediction. Expert Systems with Applications 42: 7046–56. [CrossRef]
* Bao, Wei, Jun Yue, and Yulei Rao. 2017. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PLoS ONE 12: e0180944. [CrossRef] [PubMed]
* Cervelló-Royo, Roberto, Francisco Guijarro, and Karolina Michniuk. 2015. Stock market trading rule based on pattern recognition and technical analysis: Forecasting the DJIA index with intraday data. Expert Systems with Applications 42: 5963–75. [CrossRef]
* Creighton, Jonathan, and Farhana H. Zulkernine. 2017. Towards Building a Hybrid Model for Predicting Stock Indexes. Paper presented at the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, December 11–14.
* Fama, Eugene F. 1970. Efficient Capital Markets: A Review of Theory and Empirical Work. The Journal of Finance 25: 383–417. [CrossRef]