IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.021 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2025

DOI: 10.17148/IJIREEICE.2025.13353

Forecasting Stock Prices using ARIMA algorithm

BAVANEESWAR. K[1], Mrs. MENAKA.P[2]

Department of Information Technology, Dr N.G.P Arts and Science College, Coimbatore, Tamil Nadu, India.[1]

Assistant Professor (SG), Department of Information Technology, Dr N.G.P Arts and Science College, Coimbatore, Tamil Nadu, India.[2]

Abstract: The stock prediction project presented here aims to develop an interactive application for analysing and forecasting stock prices using historical data. The project utilizes Python's Tkinter for a graphical user interface, enabling users to input National Stock Exchange (NSE) company symbols and fetch stock data from Yahoo Finance. The application retrieves one year of historical stock prices, including closing prices and trading volumes. The project further enhances analysis by incorporating moving averages—50day and 200-day—to provide trend insights. For predictive analytics, the system employs the ARIMA (Auto Regressive Integrated Moving Average) model, a well-established time-series forecasting technique. This model is trained on the stock's closing prices and generates future price predictions for the next 30 days. The results are visually represented through Matplotlib, with historical prices, moving averages, trading volumes, and forecasted prices displayed on interactive charts. This project is particularly useful for investors, traders, and financial analysts seeking to understand stock trends and predict potential future movements. The intuitive interface allows users to seamlessly interact with stock data, view market trends, and make data-driven decisions. The integration of real-time data fetching, statistical modeling, and visualization makes this application a powerful tool for stock market analysis.

I. INTRODUCTON

Stock market prediction is a challenging yet essential task in financial analysis, influencing investment decisions, risk management, and portfolio strategies. This project focuses on developing an interactive stock analysis and forecasting tool that utilizes historical data to predict future stock prices. The application is designed with a user- friendly graphical interface built using Python's Tkinter, enabling users to fetch stock data by entering the National Stock Exchange (NSE) company symbol. The system retrieves real-time stock data from Yahoo Finance, including key metrics such as closing prices and trading volumes, which are crucial indicators of market performance. To enhance analysis, the project incorporates technical indicators such as moving averages, specifically the 50-day and 200-day moving averages, to identify long-term and short-term trends in stock price fluctuations. Beyond historical analysis, the application employs the ARIMA (Autoregressive Integrated Moving Average) model, a widely used statistical approach in time-series forecasting, to predict future stock prices based on past trends. This model helps forecast stock prices for the next 30 days, providing insights into possible future movements and aiding traders and investors in making informed decisions. The primary objective of this project is to bridge the gap between complex financial data and user accessibility by providing a visual representation of stock trends. The integration of data fetching, predictive modeling, and graphical visualization makes this tool a valuable resource for financial enthusiasts, market analysts, and investors. By leveraging Python's powerful libraries, including yFinance, Pandas, Stats models, and Matplotlib, the project ensures accurate data processing and reliable predictions. With an emphasis on ease of use, this stock prediction system not only facilitates market analysis but also empowers users with data-driven insights to navigate the dynamic world of stock trading.

II. OBJECTIVES

The aims of this project are as to identify factors affecting share market, To generate the pattern from large set of data of stock market for prediction of Nasdaq and to predict an approximate value of share price to provide analysis for users through web application The objective of the system is to give a approximate idea of where the stock market might be headed. It does not give a long-term forecasting of a stock value. There are way too many reasons to acknowledge for the long-term output of a current stock. Many things and parameters may affect it on the way due to which long term forecasting is just not feasible. Enhance the accuracy of stock price predictions by leveraging advanced machine learning algorithms and models. Develop models that can predict potential risks and help investors make informed decisions to minimize losses. Improve market efficiency by providing timely and accurate predictions that can be used by traders and investors. Compare the performance of different machine learning algorithms to identify the most effective ones for stock price prediction. Identify and select the most relevant features that influence stock prices to improve the prediction models. Develop systems that can provide real-time stock price predictions to assist in high-frequency trading.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.021 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2025

DOI: 10.17148/IJIREEICE.2025.13353

III. CORE PRINCIPLES

3.1 ARIMA

ARIMA is an algorithm that uses time series forecasting to predict the future value of stocks. In a study presented by Tamerlan et al., in, it is demonstrated that the ARIMA model best fits the stock market index. The ARIMA model comprises three steps—identify, estimate, and diagnose. These steps can be used for forecasting any finance market such as equity, derivative, etc. . ARIMA can be combined with another algorithm, symmetric generalized autoregressive conditional heteroskedasticity (SGARCH), to improve forecasting performance. This combination has been modelled and tested on the S&P500 Index. ARIMA has even been extended for stock sentiment analysis. The formula of ARIMA, which combines AR (autoregression) and MA (moving average), is shown in Equation,

$$^{y0}(t) = k + \beta p * \omega^{D\,y0}{}_{t-1} + \cdots + \beta p * \omega^{Dy0}{}_{t-p} + \theta 1 * \varepsilon t{-}1 + \cdots + \theta q * \varepsilon(ti{-}q) + \varepsilon ti$$

where p represents the autoregressive model's given degree, D is the degree of different orders, and q is the moving average's given degree.

3.2. FB Prophet Model

FB Prophet is a time series forecasting library developed by Facebook. FB Prophet better suits data that have a null value and it relatively shows more accurate results in such situations. The formula of FB Prophet is presented in Equation:

$Yt = l(t) + sp(t) + v(t) + \varepsilon t, y\ t = g(t) + s(t) + h(t) + \varepsilon n$ (7) where $l(t)$ represents the linear trend, $sp(t)$ represents seasonal patterns, $v(t)$ represents holiday effects, and εn is the white noise error. The algorithm designed based on FB Prophet is implemented using the PyStan library Researchers have extensively used FB Prophet in stock price forecasting.

3.3 FORCASTING MODEL

The stock prediction project is structured into multiple modules, each designed to handle a specific functionality, ensuring an efficient and scalable system. These modules work together to provide real-time stock data retrieval, historical trend analysis, and predictive forecasting, all within an intuitive graphical user interface. The primary modules include the User Interface Module, Data Retrieval Module, Data Processing and Analysis Module, Visualization Module, and Forecasting Module. The User Interface Module is built using Python's Tkinter library, providing a simple and interactive platform for users to input stock symbols, fetch data, and visualize stock trends. It includes input fields, buttons for fetching and forecasting data, and a dynamic display area where graphical outputs are rendered. This module ensures a seamless user experience by allowing users to interact with the system without requiring programming knowledge. The Data Retrieval Module handles stock data extraction from Yahoo Finance using the yfinance library. When a user inputs a stock symbol and requests data, this module fetches historical stock price information, including closing prices, volume, and other relevant financial data for the past year.

It structures the data into a Pandas Data Frame, ensuring that it is organized and ready for further processing. The Data Processing and Analysis Module enhances stock data by calculating technical indicators such as 50-day and 200-day moving averages. These indicators help users identify market trends and make informed trading decisions. The module ensures that data is cleaned and prepared for visualization, eliminating errors that may arise from missing or inconsistent values. The Visualization Module utilizes Matplotlib to generate multiple charts, including the closing price trend, moving averages, and trading volume. The stock price trends are displayed in a time-series format, allowing users to see how stock prices have evolved. This module ensures that financial data is presented in a visually intuitive manner, making it easier for users to analyse stock movements. The Forecasting Module is responsible for predicting future stock prices using the ARIMA (Autoregressive Integrated Moving Average) model from the Stats models library. This module processes historical stock prices, applies the ARIMA model, and generates a 30-day forecast. The forecasted stock prices are plotted alongside historical prices, helping users visualize expected price movements and plan their investment strategies. Together, these modules create a comprehensive stock prediction system that integrates real-time data retrieval, financial analysis, predictive modelling, and graphical representation into a single, user-friendly application. The modular design ensures flexibility, allowing enhancements such as additional forecasting techniques or more advanced technical indicators to be incorporated in future versions of the system.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.021 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2025

DOI: 10.17148/IJIREEICE.2025.13353

IV. METHODOLOGY

The method used in this study to develop ARIMA model for stock price forecasting is explained in detail in subsections below. The tool used for implementation is EViews software version 5. Stock data used in this research work are historical daily stock prices obtained from two countries stock exchanged. The data composed of four elements, namely: open price, low price, high price and close price respectively. In this research the closing price is chosen to represent the price of the index to be predicted. Closing price is chosen because it reflects all the activities of the index in a trading day.

To determine the best ARIMA model among several experiments performed, the following criteria are used in this study for each stock index. Relatively small of BIC Relatively small standard error of regression (S.E. of regression) Relatively high of adjusted R^2 Q-statistics and correlogram show that there is no significant pattern left in the autocorrelation functions and partial autocorrelation functions of the residuals, it means the residual of the selected model are white noise. The subsections below described the processes of ARIMA model-development.

V. INPUT DESIGN

The input design of this stock prediction project is carefully structured to ensure seamless user interaction, accurate data retrieval, and effective stock analysis. The system allows users to enter a stock symbol corresponding to a company listed on the National Stock Exchange (NSE) through a graphical user interface (GUI) built with Tkinter. The primary input mechanism is a text entry field where users manually input the stock symbol of the company they wish to analyse. Additionally, the system references a predefined list of stock symbols stored in the symbols.txt file, which contains commonly used company names and their respective ticker symbols. This feature enables users to quickly access stock symbols without the need for manual lookup. Once the user enters a valid NSE stock symbol, the system fetches historical stock data from Yahoo Finance using the yfinance library.

The retrieved dataset includes essential financial metrics such as closing prices, trading volumes, and historical trends over the past year. The system processes this input data by structuring it into a Pandas Data Frame for further analysis, ensuring that data is stored efficiently and can be easily manipulated for calculations and visual representation. In addition to fetching stock data, the system includes another key input: user interaction with the forecasting feature. Users can initiate stock price prediction by clicking the "Forecast" button, which triggers the ARIMA model to process the historical closing prices and generate a 30-day future forecast. The forecasted values are then plotted and displayed to help users gain insights into expected stock movements. The input design ensures accuracy, user-friendliness, and automation, minimizing the need for manual data entry beyond entering stock symbols.

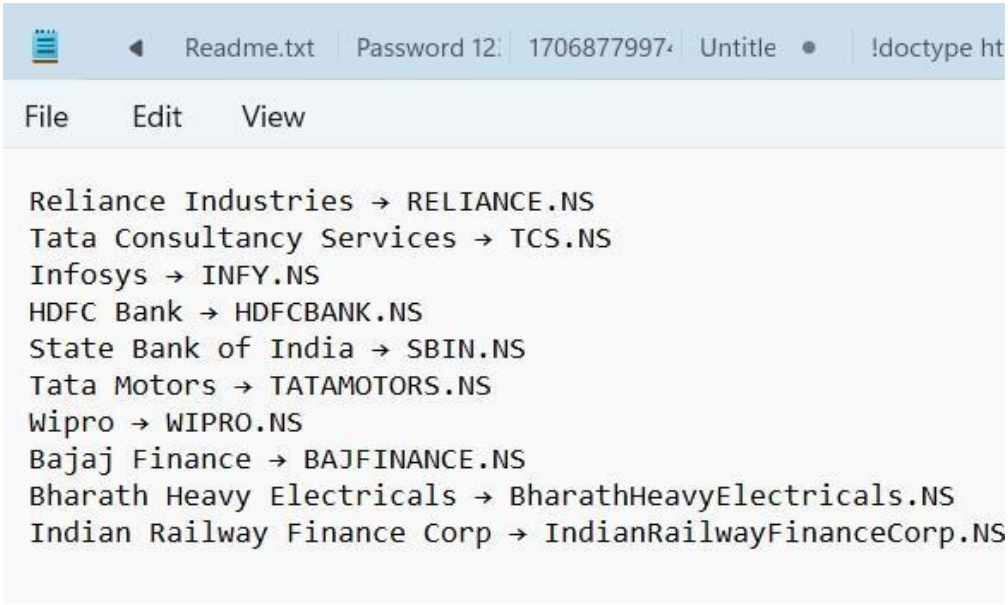


Fig 1: Stock List

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.021 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2025

DOI: 10.17148/IJIREEICE.2025.13353

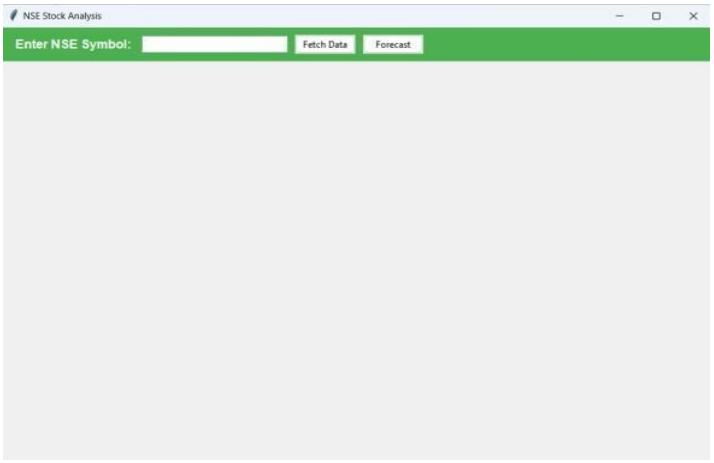


Fig 2: Login Page

VI. OUTPUT DESIGN

The output design of this stock prediction project is structured to provide users with clear, visually informative, and interactive stock analysis results. The system delivers multiple types of outputs through a graphical user interface (GUI) built with Tkinter, ensuring that users can easily interpret stock trends and forecasts. The primary output consists of stock price data retrieved from Yahoo Finance, which is processed and displayed using Matplotlib charts for better visualization. Once a user enters a valid NSE stock symbol and fetches stock data, the system generates three main types of graphical outputs. The first output is a closing price trend chart, which plots the historical closing prices of the selected stock over the past year. This graph helps users understand the stock's price movement and identify patterns over time. The second output is a moving average chart, which overlays 50-day and 200-day moving averages on the stock price graph. These moving averages provide insights into long-term and short-term trends, helping investors make informed decisions based on trend strength and momentum. The third output is a trading volume analysis chart, which represents daily stock trading volumes using a bar graph. This visualization helps users identify periods of high or low trading activity, which can indicate potential market trends. In addition to historical analysis, the system includes an ARIMA-based stock price forecasting output, which predicts the stock's future price movement for the next 30 days. Once the user clicks the "Forecast" button, the system runs the ARIMA model on historical closing prices and generates a forecasted price trend. The forecast is displayed on a separate line graph, with historical prices plotted in blue and predicted prices in red. This clear differentiation allows users to compare past stock performance with projected trends. All graphical outputs are integrated seamlessly into the Tkinter GUI using Matplotlib's FigureCanvasTkAgg, ensuring that charts are displayed dynamically within the application window

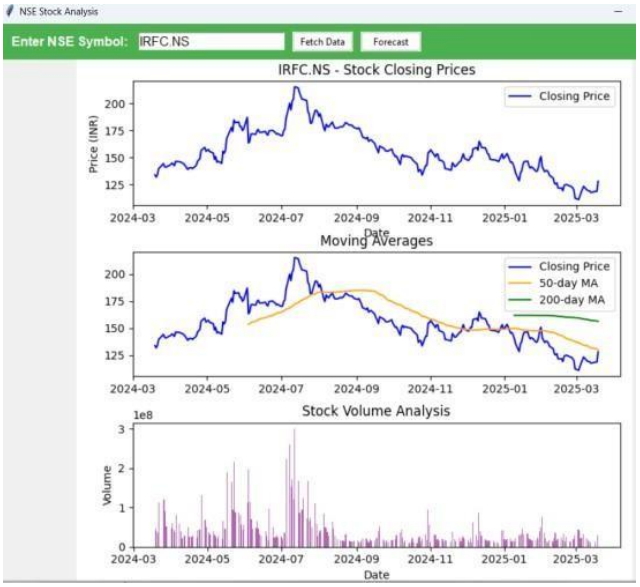


Fig 3: 200 to 50 days moment

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.021 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2025

DOI: 10.17148/IJIREEICE.2025.13353

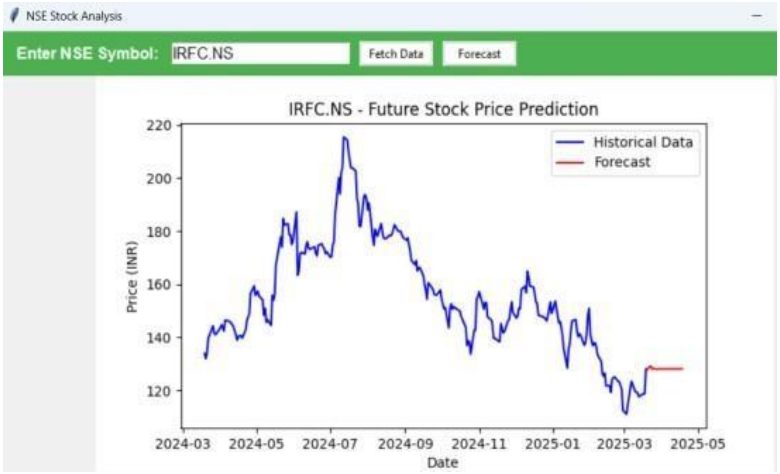


Fig 4: Forecasted Graph

VII. FILE DESIGN

The file design of this stock prediction project is structured to ensure efficient data handling, seamless user interaction, and robust predictive analysis. The project consists of multiple components, including the primary Python script for the graphical user interface and stock analysis (nse.py) and an external text file (symbols.txt) that stores predefined stock symbols for quick access. Each file plays a crucial role in the functioning of the system, ensuring smooth data retrieval, processing, and visualization. The nse.py file serves as the backbone of the project, handling all key functionalities such as fetching stock data, processing historical trends, performing predictive analysis, and displaying results through a user-friendly interface. This script utilizes Tkinter to create an interactive GUI where users can input a stock symbol and trigger data retrieval from Yahoo Finance using the yfinance library.

The retrieved data is then structured into a Pandas Data Frame, enabling efficient storage and manipulation of stock price information. The script also includes visualization components that utilize Matplotlib to generate multiple charts, such as stock closing price trends, moving averages (50-day and 200-day), and volume analysis. Additionally, the script incorporates an ARIMA-based forecasting model from the Stats models library, which analyses past stock prices and predicts future trends for 30 days, displaying the results on a dedicated graph. The symbols.txt file is a simple yet essential component that stores stock symbols and their corresponding company names for easy reference. It provides a predefined list of commonly analysed stocks, such as Reliance Industries (RELIANCE.NS), Tata Consultancy Services (TCS.NS), Infosys (INFY.NS), and others. This file enhances user convenience by allowing quick access to popular stock symbols without requiring users to manually look them up.

VIII. DATA FLOW DIAGRAM

A data flow diagram is graphical tool used to describe and analyse movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations.

A full description of a system actually consists of a set of data flow diagrams. Each component in a DFD is labelled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD'S is done in several levels. Each process in lower-level diagrams can be broken down into a more detailed DFD in the next level. The lop-level diagram is often called context diagram. It consists of a single process bit, which plays vital role in studying the current system.

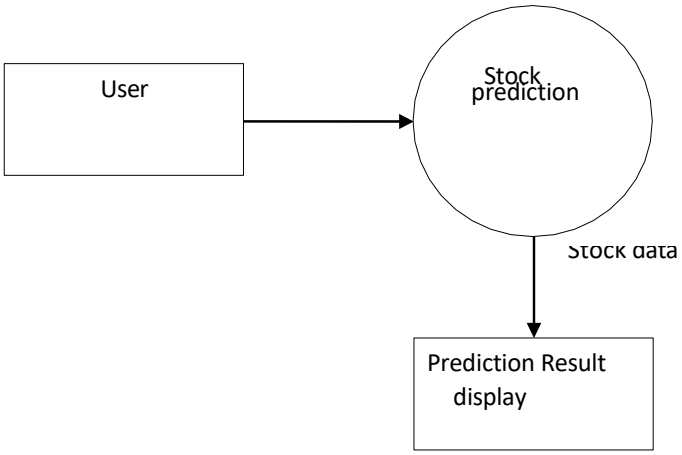
The process in the context level diagram is exploded into other process at the first level DFD.A DFD is also known as a "bubble Chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So, it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in system.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.021 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 3, March 2025

DOI: 10.17148/IJIREEICE.2025.13353



IX. CONCLUSION

The stock prediction project successfully integrates real-time stock data retrieval, historical trend analysis, and predictive forecasting into a user- friendly graphical interface. By leveraging Python's powerful libraries such as yfinance, pandas, matplotlib, and stats models, the system enables users to analyse stock market trends efficiently and make data-driven investment decisions. The implementation of the ARIMA model for forecasting provides a robust statistical approach to predicting future stock prices, offering valuable insights for traders and investors. Additionally, the inclusion of moving averages and volume analysis enhances the system's ability to identify market trends and potential investment opportunities. The intuitive GUI built with Tkinter ensures that users can easily interact with the system, fetch stock data, visualize trends, and generate forecasts without requiring advanced technical expertise. The system addresses key limitations of traditional stock analysis methods by automating data retrieval, reducing manual effort, and presenting results in a visually clear manner. Through rigorous testing, the application has been validated for accuracy, reliability, and performance, ensuring seamless integration of different modules. Overall, this project demonstrates the potential of technology-driven financial analysis by combining statistical modeling, data visualization, and user-friendly design

BOOK REFERENCES

[1]. Chatfield, C. (2003). The Analysis of Time Series: An Introduction. Chapman & Hall/CRC.
[2]. Hamilton, J. D. (1994). Time Series Analysis. Princeton University Press.
[3]. Shreve, S. E. (2004). Stochastic Calculus for Finance I & II. Springer.
[4]. Murphy, J. J. (1999). Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications. New York Institute of Finance.
[5]. Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2015). Time Series Analysis: Forecasting and Control. Wiley.
[6]. Hochreiter, S., & Schmid Huber, J. (1997). Long Short- Term Memory. Neural Computation.
[7]. Patel, S. (2018). Hands-On Artificial Intelligence for Banking: Implement Machine Learning Algorithms for Financial Services with Python. Packt Publishing.
[8]. Tsay, R. S. (2005). Analysis of Financial Time Series. Wiley-Interscience.

WEBSITE REFERENCES

[1]. Yahoo Finance – https://finance.yahoo.com (For real-time and historical stock data)
[2]. National Stock Exchange of India (NSE) – https://www.nseindia.com (For stock market information and company listings)
[3]. Python Official Documentation – https://docs.python.org (For Python programming references)
[4]. yFinance Library – https://pypi.org/project/yfinance/ (For fetching stock data in Python)
[5]. Pandas Library – https://pandas.pydata.org (For data manipulation and analysis)
[6]. Matplotlib Documentation – https://matplotlib.org (For data visualization)