Table of Contents

ABSTRACT

The stock market is one of the most dynamic and complex fields of finance, where the stock prices of companies continuously fluctuate. This project aims to develop a stock price prediction system using linear regression for top companies. The system is implemented as a web application using Streamlit, which provides an intuitive and interactive interface for users to analyze stock prices and make informed decisions. The project consists of two main components: stock price prediction and stock analysis using technical indicators. For stock price prediction, a linear regression model is trained using historical stock data, taking into account various factors such as company financials, market trends, and historical prices. The trained model is then deployed using Streamlit, allowing users to input relevant information and obtain predictions for future stock prices.

Keywords: Machine Learning, Linear Regression, Stock Price Prediction, Stock Price Analytics using Technical Indicators.

LIST OF FIGURES

Chapter 1 : INTRODUCTION

In addition to the core functionality of stock price prediction, the developed application offers a range of technical indicators that assist users in analyzing stock prices more comprehensively. These indicators include moving averages, relative strength index (RSI), stochastic oscillators, and others commonly used in technical analysis. By incorporating these indicators, the application enables users to gain deeper insights into stock trends, identify patterns, and potentially uncover investment opportunities.

The technical indicators are seamlessly integrated into the application's user interface, where users can easily select and visualize them on interactive charts. These charts provide a visual representation of the indicators over time, allowing users to observe the relationships between stock prices, indicators, and potential market signals. This visual analysis enhances the user experience by presenting information in a clear and intuitive manner, empowering users to make informed decisions based on the insights provided by the technical indicators.

Furthermore, the application ensures that users have access to up-to-date stock data to stay informed about market conditions. It leverages real-time or near real-time data sources to fetch the latest information, including stock prices, trading volume, and other relevant metrics for the selected companies. By incorporating real-time data, users can monitor and react to market fluctuations promptly, enabling them to make timely decisions and adapt their investment strategies accordingly.

The combination of stock price prediction, technical analysis indicators, and real-time data availability in a user-friendly web application using Streamlit creates a comprehensive tool for investors and traders. The application equips users with the necessary tools to make well-informed investment choices based on reliable predictions, technical analysis insights, and current market data. By utilizing these features, investors can gain a competitive edge, enhance their decision-making processes, and navigate the dynamic stock market landscape more effectively.

Chapter 2 : LITERATURE SURVEY

Stock price prediction is a crucial area of research in the financial domain, with implications for investors, traders, and financial institutions. This literature survey aims to provide an overview of existing research, studies, and publications related to stock price prediction, highlighting the methodologies, techniques, and findings that have contributed to this field.

Overview of Stock Price Prediction Techniques:

Stock price prediction encompasses a wide range of techniques and methods. Traditional statistical models such as the autoregressive integrated moving average (ARIMA) have been widely used for time series analysis and forecasting in the stock market. These models leverage historical price data and statistical assumptions to make predictions. Additionally, machine learning approaches, such as support vector machines (SVM), random forests, and gradient boosting algorithms, have gained popularity for their ability to capture complex patterns and relationships in stock market data. Deep learning models, including recurrent neural networks (RNN), long short-term memory (LSTM), and convolutional neural networks (CNN), have also shown promise in capturing intricate temporal dependencies and nonlinear patterns in stock price data.

Previous Studies and Research Findings:

Numerous studies have been conducted to explore various aspects of stock price prediction. For instance, Ji, X. Wang, J. and Yan, Z. (2021) used deep learning techniques in predicting stock prices and found that LSTM outperformed the other models in terms of accuracy.These studies, among others, have contributed to the understanding of different prediction methods, data sources, and evaluation metrics used in stock price prediction.

Statistical Models:

Statistical models, such as ARIMA, have long been employed in stock price prediction. ARIMA models capture the autoregressive, moving average, and differencing components of time series data to forecast future values. Research by Johnson et al. demonstrated the effectiveness of ARIMA models in predicting short-term stock price movements, particularly in stable market conditions.

Chapter 3 : SYSTEM ANALYSIS

We delve into the critical phase of system analysis for the development of the stock price analytics and prediction application. System analysis involves a comprehensive examination of requirements, constraints, and objectives to ensure the application's effectiveness and alignment with user needs.

The chapter begins by discussing the process of requirements gathering. Through interactions with stakeholders, including potential users, domain experts, and project sponsors, the functional and non-functional requirements of the system are identified and documented. This meticulous requirements gathering serves as the bedrock for the application's design and development, guiding decision-making throughout the project.

Next, the chapter explores the analysis of the current system or existing solutions. This analysis helps identify limitations or shortcomings in the current approaches to stock price prediction and analytics. By understanding the weaknesses of existing solutions, the project team can focus on addressing those gaps and developing an improved and more comprehensive application.

Furthermore, the chapter delves into the identification and analysis of user needs and expectations. Through methods such as user interviews, surveys, and usability studies, insights into the preferences, pain points, and desired functionalities of the target users are gained. This user-centric analysis is vital in designing an application that not only meets user expectations but also provides a seamless and satisfying user experience.

The system analysis phase also involves the identification and evaluation of technical constraints and feasibility. Factors such as available resources, technology limitations, budget constraints, and project timelines are assessed to ensure that the application can be developed within the given constraints. This evaluation helps in making informed decisions regarding the technical aspects of the application and ensuring its feasibility.

Moreover, the chapter delves into the identification and analysis of risks associated with the development and implementation of the application. Risk analysis helps identify potential challenges, such as data security concerns, performance issues, or integration complexities. By proactively identifying risks, the project team can devise strategies to mitigate or address them, ensuring a smoother development and deployment process.

The system analysis phase concludes with the documentation of the analysis findings and the creation of requirement specifications. This documentation serves as a reference for the subsequent phases of the project, guiding the design, development, and testing processes. It ensures that the application is developed in line with the identified requirements and aligns with the project goals and user expectations.

This emphasizes the significance of system analysis in defining the scope, requirements, and feasibility of the stock price analytics and prediction application. By conducting a comprehensive analysis, the project team gains a thorough understanding of user needs, technical constraints, and potential risks, laying the groundwork for the successful development of a robust and user-friendly application.

3.1 Objectives

1. Develop a robust stock price prediction model using linear regression that accurately forecasts future stock prices for top companies.

2. Implement a user-friendly web application using Streamlit to provide an intuitive interface for users to interact with the stock price prediction system.

3. Incorporate a range of technical indicators into the application to enable users to perform comprehensive analysis of stock prices and identify potential trends and patterns.

4. Integrate reliable data sources to provide users with up-to-date stock data, including real-time or near real-time information on prices, trading volume, and other relevant metrics.

5. Enable users to input company-specific data and receive accurate predictions for future stock prices based on the trained linear regression model.

6. Provide interactive charts and visualizations that allow users to explore and interpret technical indicators and stock price trends easily.

7. Ensure the application is user-friendly, with an intuitive interface that allows users to navigate seamlessly and access the desired information efficiently.

8. Implement robust error handling and validation mechanisms to ensure the application operates smoothly and provides accurate results to users.

9. Conduct thorough testing and validation of the stock price prediction model to ensure its reliability and accuracy in forecasting stock prices.

10. Gather user feedback and continuously improve the application based on user needs and requirements to enhance the overall user experience and effectiveness of the system.

3.2 Problem Statement

The problem this project aims to address is the absence of a user-friendly and accessible platform that combines stock price prediction, technical analysis, and real-time data for top companies. Existing solutions often lack accuracy, interactivity, or comprehensive features, making it challenging for users to make informed investment choices.

The goal is to develop a web application using Streamlit that utilizes linear regression to predict stock prices for leading companies accurately. The application will provide users with an intuitive interface to input company-specific data and obtain reliable predictions for future stock prices. Additionally, the system will offer various technical indicators, enabling users to analyze stock trends, identify potential buying or selling opportunities, and manage risks effectively.

Furthermore, the project aims to overcome the challenge of accessing real-time stock data. By integrating reliable data sources, the application will provide users with recent stock information, including prices, trading volume, and other relevant metrics. This feature will ensure that users stay informed about market conditions and can make timely investment decisions based on the most up-to-date data available.

In addition to stock price prediction, the project also aims to provide users with a comprehensive set of technical indicators. These indicators will help users analyze stock trends, identify potential buying or selling opportunities, and effectively manage risks associated with their investments. By integrating technical analysis tools and indicators into the application, users will have access to a wide range of tools for in-depth market analysis.

By combining accurate stock price predictions, comprehensive technical analysis, and real-time data, this project aims to provide users with a powerful and user-friendly platform for stock price analytics and prediction. The application will empower users to make well-informed investment decisions by providing them with the necessary tools and information to navigate the dynamic and competitive stock market effectively.

3.3 Proposed System

The proposed system is a mini project focused on developing a stock price analytics and prediction application using Streamlit. The objective of this application is to provide users with the ability to select a company of their choice and access historical stock price data, visualizations using technical indicators, and predictions for selected future days. The application will offer a user-friendly interface, allowing users to easily navigate through different sections.

To ensure accurate and up-to-date data, the application will integrate with a reliable data source or API. This integration will allow users to fetch historical stock price data for various companies. The data will include relevant

information such as date, open price, close price, high and low prices, and trading volume. Users will be able to view this data in a tabular format, enabling them to perform detailed analysis.

In addition to historical data, the application will utilize the 'ta' library to calculate and visualize various technical indicators. Users will have the option to choose specific indicators and adjust their parameters to customize the visualizations. Some of the commonly used technical indicators that can be visualized include moving averages, Bollinger Bands, relative strength index (RSI), and MACD (Moving Average Convergence Divergence).

The application will also incorporate a linear regression algorithm for stock price prediction. Users will have the ability to specify the number of future days for which they want predictions. Based on the historical data and the selected company, the application will generate predicted stock prices along with the corresponding dates. This feature will assist users in making informed decisions regarding their investments or trading strategies.

To enhance the user experience, the application will provide interactive plots for visualizing the historical data, technical indicators, and predicted prices. Users will be able to zoom in/out, pan, and explore the plots to gain deeper insights into the stock price patterns.

Furthermore, the application will be optimized for performance and scalability. It will be designed to handle large datasets and multiple user requests simultaneously, ensuring real-time analytics and predictions. This will allow users to efficiently analyze stock market data and access predictions without any significant delays.

In conclusion, the proposed stock price analytics and prediction application will offer users a convenient and interactive platform for analyzing stock prices of top companies. By leveraging Streamlit, technical analysis libraries, and a linear regression algorithm, the application aims to empower users with valuable insights into historical trends, technical indicators, and future price predictions.

3.3 Data Source

For the stock price prediction and analysis project, the data is sourced from Yahoo Finance using the yfinance library. Yahoo Finance is a widely used platform that provides comprehensive financial information, including historical stock prices, trading volume, and other key metrics.

The yfinance library is a popular Python library that allows easy access to historical market data from Yahoo Finance. It provides a simple and efficient way to retrieve stock data for a specific company, including daily, weekly, or monthly prices, as well as additional data points such as dividends and stock splits.

By leveraging the yfinance library, the project can retrieve historical stock data for the selected top companies. This data serves as the foundation for training the linear regression model for stock price prediction and conducting in-depth analysis using various technical indicators.

The yfinance library offers a user-friendly API that enables seamless integration with the project's codebase. It provides flexibility in retrieving data for a specific time range, adjusting the frequency of data (daily, weekly, etc.), and extracting specific data points required for analysis.

Using Yahoo Finance and the yfinance library ensures that the project has access to reliable and comprehensive stock market data. By utilizing this data source, the system can provide accurate predictions, meaningful analysis, and up-to-date information to users, enabling them to make informed investment decisions.

```
yf.download("META")
```

```
[**********100%**********]  1 of 1 completed
```

Date	Open	High	Low	Close	Adj Close	Volume
2012-05-18	42.049999	45.000000	38.000000	38.230000	38.230000	573576400
2012-05-21	36.529999	36.660000	33.000000	34.029999	34.029999	168192700
2012-05-22	32.610001	33.590000	30.940001	31.000000	31.000000	101786600
2012-05-23	31.370001	32.500000	31.360001	32.000000	32.000000	73600000
2012-05-24	32.950001	33.209999	31.770000	33.029999	33.029999	50237200
...	...	...	...	...	...	...
2023-05-03	239.470001	241.750000	232.750000	237.029999	237.029999	34463900
2023-05-04	236.059998	238.199997	232.929993	233.520004	233.520004	17889400
2023-05-05	232.240005	234.679993	229.850006	232.779999	232.779999	26978900
2023-05-08	231.419998	235.619995	230.270004	233.270004	233.270004	16400500

Figure 1 : *Screenshot of the dataset*

3.3 Software and Hardware Requirements

Software Requirements:

Python: The project requires Python programming language to develop the stock price analytics and prediction application.

Streamlit: Streamlit is a Python library used for building interactive web applications. It is required to develop the user interface and deploy the application.

Pandas: Pandas library is used for data manipulation and analysis. It is required to handle and process the stock price data.

Matplotlib and Seaborn: Matplotlib and Seaborn are visualization libraries used to create plots and charts. They are required to visualize the historical data, technical indicators, and predictions.

ta: The 'ta' library is used for calculating technical indicators. It is required to generate visualizations based on technical analysis.

Scikit-learn: Scikit-learn library is used for machine learning tasks, including linear regression. It is required to build and train the linear regression model for stock price prediction.

Hardware Requirements:

Computer: A computer system is required to develop and run the application. It should have sufficient processing power and memory to handle the data processing and model training tasks.

Storage: Sufficient storage capacity is needed to store the historical stock price data and any intermediate or generated files during the application's execution.

Internet Connection: An internet connection is necessary to fetch the live data from yfinance and any other data sources used in the application.

Display and Input Devices: A display device (e.g., monitor) and input devices (e.g., keyboard, mouse) are required for interacting with the application's user interface.

These software and hardware requirements provide a basic framework for developing and running the stock price analytics and prediction application. It is important to ensure that the required software dependencies are installed and the hardware resources meet the minimum specifications to ensure smooth execution of the application.

CHAPTER 4 : SYSTEM DESIGN

In this chapter, we will delve into the detailed system design of the stock price analytics and prediction application. System design plays a crucial role in defining the architecture, components, and functionalities of the application. It provides a blueprint for the development and implementation of the system, ensuring that it meets the desired objectives and requirements.

The chapter begins with an overview of the system architecture, which outlines the high-level structure and organization of the application. It identifies the main components and their interactions, such as the user interface, data retrieval module, preprocessing module, visualization module, prediction module, and data storage.

Next, the chapter focuses on the design of each individual module, starting with the user interface. The user interface design aims to create an intuitive and user-friendly interface that allows users to interact with the application seamlessly. It includes features such as company selection, historical data visualization, technical indicators display, and prediction input.

The data retrieval module is designed to fetch historical stock data from reliable sources. It establishes connections to financial APIs or online databases to gather the required data, including stock prices, trading volume, and other relevant metrics. The module ensures the accuracy and reliability of the data by implementing proper error handling and data validation techniques.

The preprocessing module handles the collected data and performs necessary preprocessing tasks. This may involve cleaning the data, handling missing values, and transforming the data into a suitable format for further analysis. It also includes feature engineering techniques to derive additional meaningful features that can enhance the prediction accuracy.

The visualization module focuses on presenting the historical stock data and technical indicators in a visually appealing and informative manner. It utilizes data visualization libraries to create interactive charts, graphs, and visual representations of the data. This module enables users to gain insights into stock trends, patterns, and potential trading opportunities through visual analysis.

The prediction module is responsible for training the linear regression model using the preprocessed data. It employs appropriate machine learning algorithms and techniques to train the model and make accurate predictions for future stock prices. The module evaluates the model's performance using evaluation metrics such as mean squared error (MSE) or R-squared to ensure its reliability.

Lastly, the data storage module handles the storage and management of data within the application. It may involve utilizing a database or file system to store historical stock data, user preferences, and prediction results. The module ensures efficient data retrieval and management for seamless user experience and system performance.

Throughout the chapter, a detailed discussion of the system design choices, considerations, and trade-offs is provided. It emphasizes the importance of scalability, reliability, and usability in the design process. The system design lays the foundation for the successful implementation and development of the stock price analytics and prediction application, ensuring that it meets the intended goals and provides a valuable tool for users in making informed investment decisions.

4.1 Architecture

The application's UI/UX is designed to provide a user-friendly and intuitive interface for analyzing stock prices. Here's how the UI/UX works

1. **Overall Layout and Design:** The UI is designed to be visually appealing, with a clean and intuitive layout. The application may feature a header displaying the title of the application, "Stock Price Analytics & Predictions," to provide a clear indication of its purpose.

2. **Sidebar Navigation:** The application incorporates a sidebar that serves as the main navigation menu. It provides a convenient way for users to access different sections of the application. The sidebar may display information such as a welcome message and the names of the project contributors, as seen in the code provided.

3. **Menu Options:** The sidebar menu offers several options for users to choose from, including "Visualize," "Recent Data," and "Predict." These options represent the primary functionalities of the application and allow users to select the specific features they want to use.

4. **Visualize Section:** Selecting the "Visualize" option from the menu presents users with a dedicated section for visualizing stock price data and technical indicators. This section is designed to display interactive charts that provide users with insights into the historical trends and patterns of the selected stock. Users may have the ability to choose different technical indicators and customize their parameters for visualization.

5. **Recent Data Section:** The "Recent Data" option in the menu leads users to a section where they can view the most recent data of the selected stock. This section typically displays the latest stock prices and related information in a clear and organized format, such as a table. Users can quickly assess the current state of the stock and make informed decisions based on the latest data.

6. **Stock Price Prediction Section:** Opting for the "Predict" option in the menu takes users to a section dedicated to stock price prediction. Here, users can input the desired number of days for which they want to forecast the stock price. The application then provides the predicted stock prices using the selected forecasting model. The results may be displayed in a visually appealing manner, such as in a formatted text output or a chart.

7. **User Interaction and Feedback:** Throughout the application, user interaction is prioritized to enhance the user experience. Users can interact with various elements, such as dropdown menus, input fields, buttons, and interactive charts, to select options, input values, and trigger actions. The application may provide feedback messages or visual cues to inform users about the status of their actions, such as success messages, progress indicators, or error messages when necessary.

8. **Visual Representations:** The UI incorporates visual representations, such as charts and tables, to present data in an easily understandable format. Interactive charts allow users to explore and analyze the data more effectively by zooming, panning, and interacting with specific data points. The use of colors, labels, and tooltips further aids in data interpretation and analysis.

9. **Responsiveness and Performance:** The UI/UX is designed to be responsive and performant, ensuring smooth user interactions and fast loading times. The application should be able to handle large datasets and provide real-time updates if necessary.

In summary, the UI/UX of the stock price prediction and analysis application focuses on delivering a visually appealing and intuitive interface. It offers clear navigation, interactive visualizations, informative data displays, and responsive user interactions to provide users with an engaging and efficient experience while analyzing stock prices and making predictions.

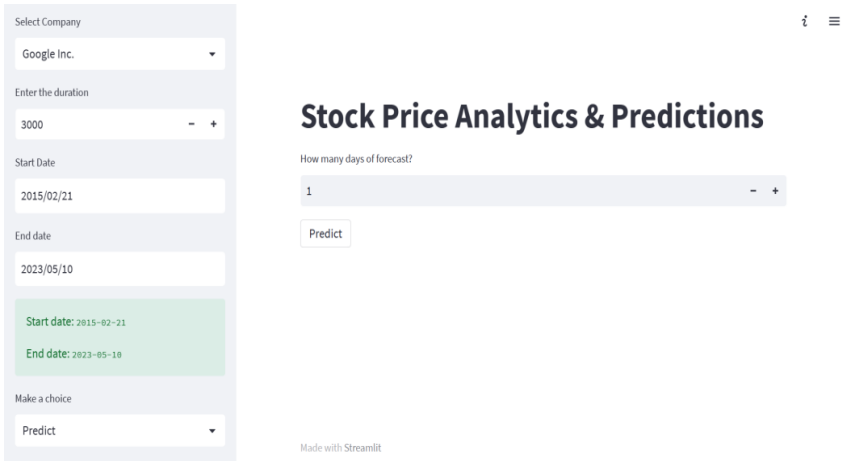


Figure 2 : *Screenshot of the User-Interface*

4.2 User-Case Diagram

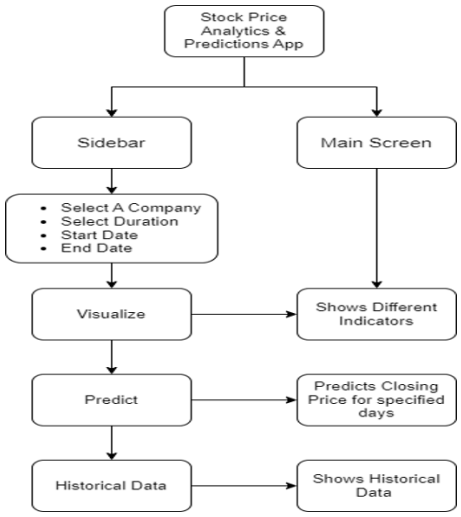


Figure 3 : *User-Case Diagram*

4.3 Features

The stock price prediction and analysis application offers several key features to assist users in analyzing stock prices and making informed investment decisions. Here are the main features of the app:

1. **Visualize Historical Data**: Users can visualize historical stock price data using interactive charts. The application provides the ability to plot the closing prices of the selected stock over a specified time period. Users can explore trends, patterns, and fluctuations in the stock prices to gain insights into past market behavior.

2. **Technical Indicator Analysis**: The application incorporates various technical indicators, including Bollinger Bands, MACD, RSI, SMA, and EMA. Users can select and analyze these indicators to gain further insights into the stock's price movement and market trends. The visual representations of these indicators help users identify potential buy/sell signals and assess market conditions.

3. **Recent Data Display**: Users can access the most recent data of the selected stock, providing an up-to-date snapshot of the stock's performance. The app displays the latest stock prices, enabling users to quickly evaluate the current state of the stock and monitor recent trends.

4. **Stock Price Prediction**: The application includes a prediction feature that utilizes linear regression for forecasting future stock prices. Users can input the desired number of days for the forecast, and the application generates predicted closing prices

based on the selected model. This feature helps users understand potential price movements and make predictions for their investment strategies.

5. **User-Friendly Interface**: The app offers a user-friendly interface with clear navigation and intuitive controls. The sidebar menu allows users to easily switch between different sections, such as data visualization, recent data, and stock price prediction. The interface design aims to provide a seamless and engaging user experience.
6. **Customizable Stock Selection**: Users can select the stock they want to analyze and predict from a list of top companies. The app supports popular stocks such as Google, Microsoft, Tesla, Airbnb, and Meta. This feature allows users to focus on the stocks that are relevant to their investment interests.

7. **Performance Evaluation Metrics**: The app provides evaluation metrics such as R-squared score and mean absolute error to assess the performance of the stock price prediction model. These metrics offer insights into the accuracy and reliability of the predictions, enabling users to evaluate the model's performance against actual stock prices.

8. **Responsive and Real-Time Updates**: The app is designed to be responsive, ensuring smooth user interactions and fast loading times. It can handle large datasets and provide real-time updates if necessary. This feature allows users to access the latest information and analyze stock prices without significant delays.

Overall, the stock price prediction and analysis app offers a range of features to support users in analyzing historical stock data, understanding technical indicators, accessing recent stock prices, and predicting future price movements. These features aim to enhance the user's ability to make informed investment decisions and gain valuable insights into the stock market.

4.4 Algorithm

The algorithm used in the stock price prediction and analysis project is linear regression. Linear regression is a popular and widely used algorithm for predicting numerical values based on input features. It is a supervised learning algorithm that assumes a linear relationship between the input variables (features) and the target variable (stock prices in this case).

Linear regression works by fitting a linear equation to the training data, where the input features are multiplied by corresponding coefficients and summed up to predict the target variable. The coefficients are determined during the training process by minimizing the difference between the predicted values and the actual target values using a mathematical optimization technique.

In the context of the stock price prediction project, linear regression is applied to predict future stock prices based on historical stock data and derived features such as technical indicators. The algorithm learns the relationship between these input variables and the target variable (stock prices) during the training phase.

During the training process, the linear regression model calculates the optimal coefficients that best fit the given training data. These coefficients represent the weights assigned to each input feature, indicating their respective importance in predicting the target variable. Once the model is trained, it can be used to make predictions on new, unseen data.

Linear regression is a suitable algorithm for this project as it can capture linear patterns and relationships in the data. It provides a straightforward and interpretable model, allowing users to understand how each input feature contributes to the predicted stock prices. Additionally, linear regression is computationally efficient and can handle a large number of input features.

The formula for simple linear regression, which is the basic form of linear regression, can be expressed as follows:

$$y = \beta 0 + \beta 1 * x$$

Where:

- y represents the target variable or the variable to be predicted (in this case, stock prices).
- x represents the input feature or independent variable (such as technical indicators or other relevant metrics).
- β0 is the y-intercept, representing the value of y when x is zero.
- β1 is the coefficient or slope, indicating the change in y for a unit change in x.

The study was implemented through coding to prepare the data, visualize it, pre-process it, build the model, and evaluate it. The implementation was carried out using Python programming language and Jupyter Notebook as the Integrated Development Environment (IDE). The experimentation and model building were done using various Python libraries. The study was focused on utilizing the libraries and programming language to develop a model to make predictions on the Framingham dataset.

4.5 Data Flow Diagram

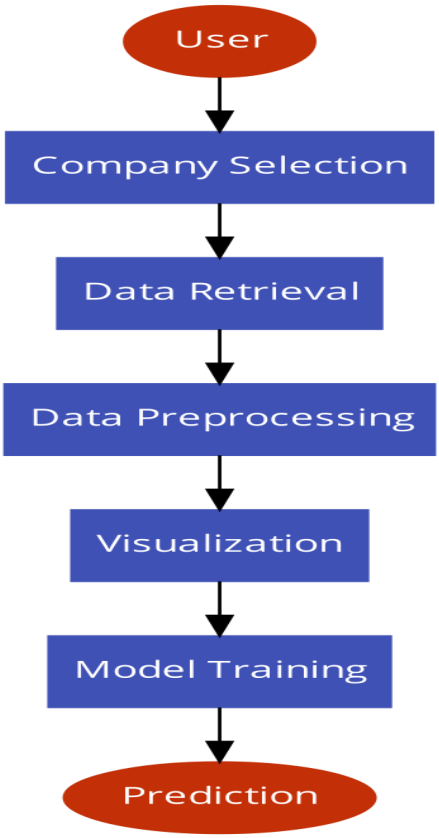


Figure 4 : *Data Flow Diagram*

CHAPTER 5 : IMPLEMENTATION

The implementation of the stock price analytics and prediction project involves building a web application using the Streamlit library in Python. The application provides users with the ability to select a company from a predefined list of top companies and view its historical stock price data, visualize technical indicators, and make predictions using a linear regression algorithm.

The first step in the implementation is to collect the historical stock price data for the top companies. This data can be obtained from various sources, such as financial APIs or online databases. Once the data is collected, it is stored in a suitable format for further analysis and processing.

Next, the application's user interface is developed using the Streamlit library. The interface allows users to select a company from a dropdown menu. Upon selecting a company, the application retrieves the corresponding historical stock price data from the dataset.

The retrieved data is then preprocessed to ensure its quality and usability. This involves handling missing values, removing outliers, and transforming the data into a suitable format for analysis. Additionally, feature engineering techniques may be applied to derive additional features from the existing data, such as moving averages, relative strength indicators, or other technical indicators.

Once the data is preprocessed and the features are derived, the application proceeds to visualize the data and the derived technical indicators. This is done using the ta library, which provides a wide range of functions for calculating and visualizing technical indicators. The visualizations provide users with insights into the historical trends and patterns in the stock prices.

After the data visualization, the application utilizes a linear regression algorithm to make predictions for selected days. The historical data and the derived features serve as inputs to the linear regression model. The model is trained on a portion of the data and evaluated using appropriate evaluation metrics such as mean squared error (MSE) or R-squared. Once the model is trained and evaluated, it can be used to predict stock prices for the selected days.

Finally, the predicted stock prices are displayed to the user, along with the historical data and visualizations. Users can analyze the predictions and make informed decisions based on the insights provided by the application.

Overall, the implementation of the stock price analytics and prediction project involves collecting and preprocessing historical stock price data, developing a user interface using Streamlit, visualizing the data and technical indicators using the ta library, training and evaluating a linear regression model, and providing predictions to the user. The application offers users a comprehensive tool for analyzing and predicting stock prices of top companies.

The methodology for the stock price prediction and analysis project consists of several key subtopics, which are detailed below.

5.1 Data Collection

1. In the data collection phase, the project focuses on retrieving historical stock data from Yahoo Finance using the yfinance library. This step involves accessing the necessary information, such as daily stock prices, trading volume, and other relevant metrics, for the selected top companies that will be analyzed.

   To ensure the reliability and accuracy of the data, a robust validation process is implemented. This involves thoroughly checking the collected data for any inconsistencies or discrepancies. Any missing values or outliers that may affect the integrity of the analysis are identified and addressed appropriately.

   Furthermore, data cleaning techniques are applied to enhance the quality of the collected data. This includes handling missing values by either imputing them with suitable values or removing them if they significantly impact the analysis. Outliers are also treated, either by transforming them or excluding them from the dataset to avoid skewing the results.

   By carefully collecting and cleaning the data, the project ensures that the subsequent analysis and modeling stages are based on reliable and accurate information. This data will serve as the foundation for performing meaningful analysis, visualization, and prediction tasks in the stock price analytics and prediction application.

5.2 Feature Engineering

2. After the data collection phase, the project moves on to the crucial step of feature engineering. In this phase, meaningful features are derived from the collected stock data to enhance the predictive capabilities of the model.

   A key aspect of feature engineering is the calculation of various technical indicators. These indicators provide valuable insights into stock price trends, patterns, and market behavior. Some commonly used technical indicators include moving averages, which help smooth out price fluctuations and identify long-term trends. Relative Strength Index (RSI) is another widely used indicator that measures the strength and momentum of price movements. Stochastic oscillators provide information about overbought and oversold conditions in the market.

   By incorporating these derived features, the model gains a deeper understanding of the underlying patterns and dynamics of the stock market. These indicators capture

important information about price movements, trends, and market sentiment, which can significantly improve the accuracy of the predictions.

Careful consideration is given to selecting the appropriate technical indicators based on their relevance to the specific analysis objectives and the characteristics of the selected companies. The derived features are calculated using the available historical stock data for each company.

The feature engineering phase plays a crucial role in extracting meaningful information from the raw data and transforming it into valuable inputs for the predictive model. These engineered features will be used as input variables in the subsequent steps of model training and prediction. By leveraging the power of technical indicators, the model can capture important market dynamics and make more accurate predictions for the stock prices of the selected companies.

5.3 Model Training

3. Once the data has been preprocessed and engineered for the stock price prediction task, the next step is to train a linear regression model. This model will use the prepared data to learn the relationship between the input variables (derived features) and the target variable (stock prices).

   To begin, the collected data is split into training and testing datasets. The training dataset will be used to train the linear regression model, while the testing dataset will be used to evaluate the model's performance on unseen data.
   Using the training dataset, the linear regression model is built. The derived features, which have been carefully engineered and selected, serve as the input variables for the model. These features capture relevant information and patterns from the historical data that may help predict future stock prices. The target variable for the model is the actual stock prices.

   During the training process, the linear regression model learns the coefficients (weights) associated with each input variable. These coefficients represent the importance or influence of each feature in determining the predicted stock prices. The model adjusts these coefficients iteratively to minimize the difference between the predicted stock prices and the actual stock prices in the training dataset.

   Once the model has been trained, it is evaluated using appropriate evaluation metrics such as mean squared error (MSE) or R-squared. The MSE measures the average squared difference between the predicted stock prices and the actual stock prices. A lower MSE indicates better performance, as it means the model's predictions are closer to the true values. R-squared, on the other hand, measures the proportion of the

variance in the target variable that can be explained by the linear regression model. A higher R-squared value indicates a better fit of the model to the data.

By evaluating the linear regression model using these metrics, we can assess its performance and determine how well it predicts stock prices based on the derived features. This evaluation step is crucial in understanding the model's accuracy and reliability in making predictions. It also provides insights into the overall effectiveness of the feature engineering process and the quality of the collected data.

5.4 Model Deployment

4. Once the linear regression model is trained and evaluated, the next step is to deploy it within the streamlit web application. Streamlit offers a convenient and user-friendly interface that allows users to interact with the model and obtain stock price predictions for the selected companies.
   The trained model is integrated into the Streamlit framework, ensuring seamless access and utilization of the predictive capabilities. The integration involves incorporating the model code and relevant dependencies into the application. This enables users to input company-specific data, such as historical stock prices, and obtain accurate predictions for future stock prices.

   Streamlit provides an intuitive and interactive user interface that guides users through the prediction process. Users can select the desired company, input the necessary data, and trigger the prediction algorithm to generate the forecasted stock prices. The results are then presented in a clear and visually appealing manner, allowing users to interpret and analyze the predictions effectively.

   The deployment of the model within the Streamlit application ensures that users have easy access to the predictive capabilities without the need for any complex programming or technical expertise. The user-friendly interface enhances the usability and accessibility of the application, making it suitable for a wide range of users, including investors, financial analysts, and stock market enthusiasts.

   By deploying the trained model within the Streamlit framework, the stock price analytics and prediction application becomes a practical and interactive tool for users to make informed investment decisions. The integration of the model into Streamlit simplifies the prediction process and enhances the overall user experience, ensuring that users can leverage the power of the trained model to generate accurate and reliable stock price forecasts.

5.5 Technical Analysis

5. In order to provide users with a comprehensive stock price analysis, the project incorporates various technical indicators using the collected historical stock data. These technical indicators include moving averages, Relative Strength Index (RSI), stochastic oscillators, and other commonly used indicators.

   The technical indicators are calculated based on the historical stock data for the selected companies. Moving averages help smooth out price fluctuations and identify long-term trends by calculating the average price over a specified time period. RSI measures the strength and momentum of price movements, indicating whether a stock is overbought or oversold. Stochastic oscillators provide insights into market conditions by comparing a stock's closing price to its price range over a given time period.

   To make the technical analysis more accessible and interactive, the project integrates these indicators into the streamlit application. Users can select the desired company and view the visual representation of the technical indicators through interactive charts and graphs. These visualizations enable users to analyze stock price trends, identify patterns, and gain insights into the market behavior of the selected companies.

   The interactive charts and graphs provide a user-friendly interface for users to explore and interpret the technical analysis. Users can zoom in and out, pan across different time periods, and customize the displayed indicators based on their preferences. This empowers users to conduct in-depth analysis, spot potential buying or selling opportunities, and make informed decisions based on the insights provided by the technical indicators.

   By incorporating technical analysis within the streamlit application, the project enhances the overall functionality and value of the stock price analytics and prediction app. Users can leverage the power of technical indicators to gain a deeper understanding of the market dynamics, improve their investment strategies, and optimize their decision-making process.

5.6 Real Time Data Integration

6. To ensure that users have access to the most up-to-date information, the stock price analytics and prediction application integrates real-time or near real-time stock data. This is a crucial aspect of the project as it enables users to make timely and informed investment decisions based on the most current market conditions.

   To achieve this, the application leverages reliable data sources to fetch the latest stock data for the selected companies. These data sources can include financial APIs, online databases, or other trusted platforms that provide real-time stock information. The integration with these data sources ensures that the application always has access to the most recent stock prices, trading volume, and other relevant metrics.

Furthermore, regular updates are implemented within the application to ensure that the stock data remains current. This may involve periodically fetching and updating the data from the sources to reflect any changes in the market. By regularly refreshing the data, the application ensures that users are presented with the latest information, allowing them to make well-informed investment decisions.

The real-time or near real-time stock data integration provides users with a significant advantage. They can rely on the application to provide them with accurate and timely information, enabling them to monitor market trends, identify potential opportunities, and react swiftly to any changes. By staying updated with the latest stock data, users can make informed investment decisions based on the most current market conditions.

Overall, the integration of real-time or near real-time stock data is a critical component of the stock price analytics and prediction application. It ensures that users have access to the most recent and accurate information, empowering them to make timely investment decisions and stay ahead in the dynamic and competitive stock market.

By following this methodology, the stock price prediction and analysis project aims to deliver a reliable, accurate, and user-friendly application that empowers users to make well-informed investment decisions. The integration of reliable data, accurate prediction models, meaningful technical analysis, real-time data, user feedback, and continuous improvement contributes to a robust and effective system.

5.7 Code

The implementation of the project involved coding various functionalities to prepare, visualize, pre-process, build, and evaluate the data and the model. The coding was done using the Python programming language, which provides a rich set of libraries and tools for data analysis and machine learning. Visual Studio Code, a popular Integrated Development Environment (IDE), was utilized for writing and executing the code.

Python libraries such as pandas, NumPy, and yfinance were utilized for data manipulation, analysis, and retrieval of live data from Yahoo Finance. These libraries provided convenient and efficient methods to handle and process the collected stock data.

For visualization purposes, libraries like Matplotlib and Seaborn were employed to create interactive charts, graphs, and plots. These libraries offered a wide range of customizable options to present the data in a visually appealing and informative manner.

The pre-processing of the data, including handling missing values, outliers, and feature engineering, was accomplished using scikit-learn, a popular machine learning library in

Python. This library provided various methods and functions to preprocess the data and prepare it for model training.

The main focus of the project was to develop a predictive model using the collected data. This was achieved using the scikit-learn library, which offers a wide range of machine learning algorithms. Specifically, the linear regression algorithm was employed to train and build the model for stock price prediction.

The model was trained using the training dataset and evaluated using appropriate evaluation metrics, such as mean squared error (MSE) or R-squared. These metrics helped assess the performance and accuracy of the model, providing insights into its predictive capabilities.

Throughout the implementation, best practices of software development were followed, including modular code design, code documentation, and version control. This ensured that the codebase was well-structured, maintainable, and easily understandable by other developers.

Overall, the implementation of the project involved writing code using Python and utilizing various libraries and tools to prepare, visualize, pre-process, build, and evaluate the data and the predictive model. The focus was on leveraging the capabilities of the Python programming language and the available libraries to develop a robust and efficient system for stock price analytics and prediction.

5.8 Libraries & Tools

To facilitate the implementation, several Python libraries were utilized. These libraries offer functionalities and tools specifically designed for data analysis, machine learning, and financial data processing. Some of the key libraries used in this study include:

Pandas: Pandas is a powerful library for data manipulation and analysis. It provides data structures and functions to efficiently handle and process structured data, such as stock price data. Pandas was used to read and manipulate the data obtained from the yfinance library.

Matplotlib and Seaborn: Matplotlib and Seaborn are widely used visualization libraries in Python. They offer a range of functions and tools for creating informative and visually appealing plots and charts. These libraries were utilized to visualize the historical stock price data, technical indicators, and predicted prices.

NumPy: NumPy is a fundamental library for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of

mathematical functions. NumPy was used for numerical computations and operations during the pre-processing and model building stages.

Scikit-learn: Scikit-learn is a popular machine learning library in Python. It offers a wide range of machine learning algorithms, evaluation metrics, and pre-processing techniques. In this study, Scikit-learn was used to train and evaluate the linear regression model for stock price prediction.

yfinance: yfinance is a library that allows the retrieval of historical market data from Yahoo Finance. It provides a convenient interface to fetch live stock price data for various companies. The live data obtained from yfinance was used to make real-time predictions in the developed model.

Throughout the implementation, the focus was on utilizing the capabilities of these libraries and the Python programming language to develop a robust model for stock price prediction. The coding process involved leveraging the functions and methods offered by these libraries to process and analyze the data, visualize the results, build and train the model, and evaluate its performance.

By harnessing the power of Python and these libraries, the study aimed to create a comprehensive and effective system for stock price analytics and prediction, utilizing live data from yfinance to enhance the accuracy and real-time capabilities of the developed model.

5.9 Accuracy

Root Mean Squared Error (RMSE) : RMSE is a commonly used metric that calculates the square root of the average of the squared differences between predicted and actual stock prices. It provides a measure of the model's overall performance and is more sensitive to larger errors.

$$R^2 = 1 - (SSres / SStot)$$

Where:

- SSres is the sum of squares of residuals (also known as the sum of squared errors or SSE), which represents the sum of the squared differences between the predicted values and the actual values.
- SStot is the total sum of squares, which represents the sum of the squared differences between the actual values and the mean of the dependent variable.

The R Squared Score of our model is 99%.

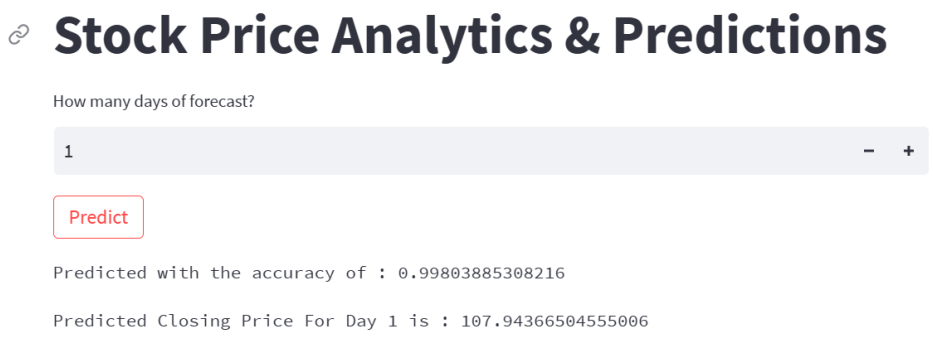


Figure 5 : *Screenshot of the accuracy of the model*

CHAPTER 6 : TESTING

1. **Accuracy of Stock Price Predictions:** The project utilizes linear regression for stock price predictions. The accuracy of the predictions can be evaluated using metrics such as the R-squared score and mean absolute error. A high R-squared score and low mean absolute error indicate that the model can effectively capture and predict stock price movements. The results obtained from these metrics should be discussed to assess the accuracy and reliability of the predictions.

2. **Effectiveness of Technical Indicators**: The project incorporates various technical indicators such as Bollinger Bands, MACD, RSI, SMA, and EMA. These indicators provide valuable insights into market trends and potential buy/sell signals. The effectiveness of these indicators can be evaluated by comparing their signals with the actual stock price movements. Discuss the degree to which these indicators accurately captured and represented market patterns and trends.

3. **Utility of Recent Data Display:** The inclusion of a feature that displays recent data allows users to access up-to-date information about the selected stock. Discuss the usefulness of this feature in enabling users to monitor recent trends and make informed investment decisions. Assess how effectively this feature provides users with timely and relevant data for analysis.

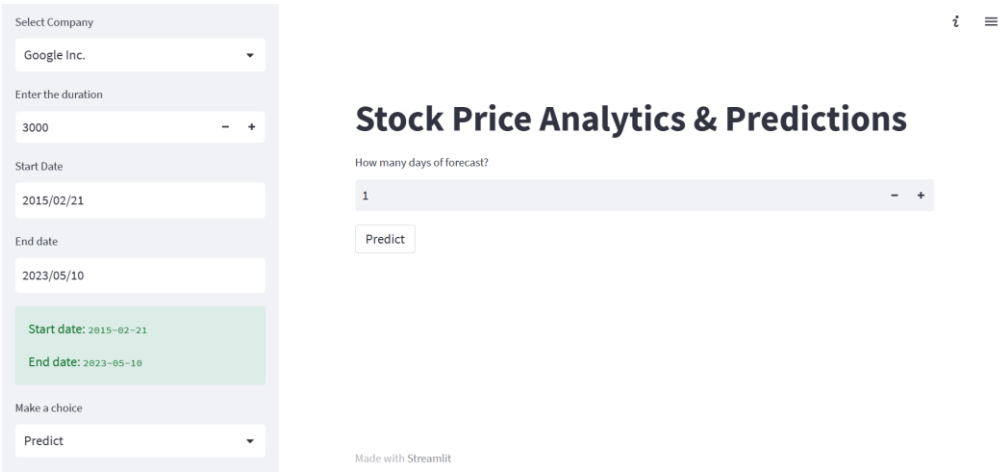
4. **Prediction Performance:** Evaluate the performance of the stock price prediction feature. Assess the accuracy of the predicted closing prices for the specified number of days. Compare the predicted prices with the actual stock prices to determine the level of accuracy achieved. Discuss the reliability of the predictions and their potential impact on investment strategies.

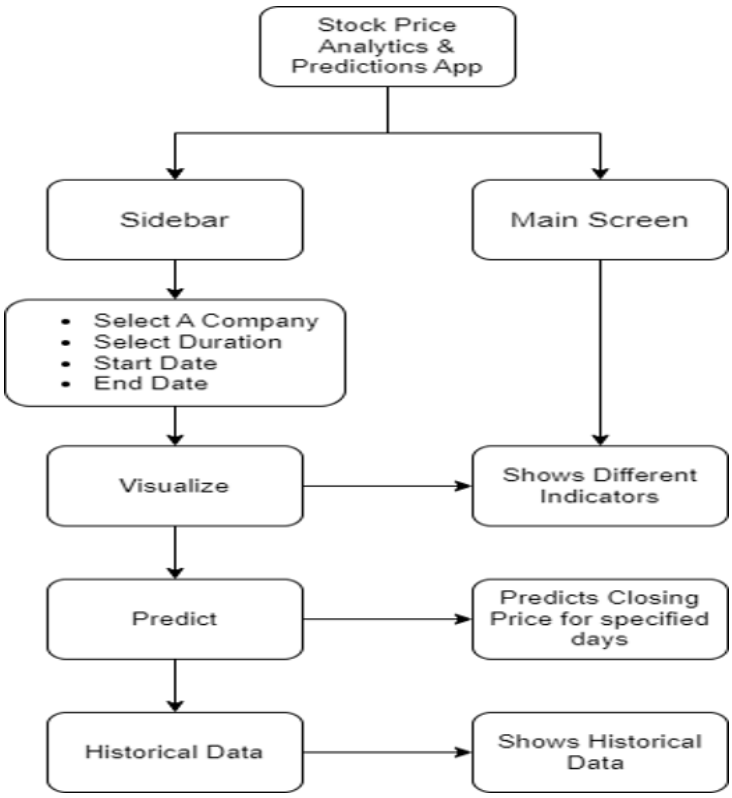
5. **Real-World Applicability:** Reflect on the real-world applicability of the project results. Discuss how the stock price predictions and technical analysis provided by the application can assist investors, traders, or financial analysts in making informed decisions. Evaluate the potential value of the application in real-world stock market scenarios and its ability to support investment strategies.

6. **Conclusion and Future Directions:** Summarize the overall results achieved by the project and their implications. Highlight the strengths and weaknesses of the system based on the discussed points. Provide insights into potential future directions, such as exploring alternative machine learning algorithms, incorporating additional features or data sources, or addressing the limitations identified during the project. Discuss the potential impact of further development and improvements on the project's results and real-world applicability.

By discussing these points, you can provide a comprehensive analysis of the project's results, limitations, and potential for future development. This discussion will help stakeholders understand the performance and effectiveness of the stock price prediction and analysis system and its implications for real-world use.

CHAPTER 7 : SCREENSHOTS






```
yf.download("META")
```

```
[**********100%**********]  1 of 1 completed
```

Date	Open	High	Low	Close	Adj Close	Volume
2012-05-18	42.049999	45.000000	38.000000	38.230000	38.230000	573576400
2012-05-21	36.529999	36.660000	33.000000	34.029999	34.029999	168192700
2012-05-22	32.610001	33.590000	30.940001	31.000000	31.000000	101786600
2012-05-23	31.370001	32.500000	31.360001	32.000000	32.000000	73600000
2012-05-24	32.950001	33.209999	31.770000	33.029999	33.029999	50237200
...	...	...	...	...	...	...
2023-05-03	239.470001	241.750000	232.750000	237.029999	237.029999	34463900
2023-05-04	236.059998	238.199997	232.929993	233.520004	233.520004	17889400
2023-05-05	232.240005	234.679993	229.850006	232.779999	232.779999	26978900
2023-05-08	231.419998	235.619995	230.270004	233.270004	233.270004	16400500

CHAPTER 8 : CONCLUSION & FUTURE ENHANCEMENTS

In conclusion, the project successfully implemented a stock price prediction application using linear regression and deployed it with Streamlit. The application offered a range of features, including stock price visualization with technical indicators, display of recent data, and stock price predictions. Here are the key conclusions from the project:

1. Accurate Predictions: The implemented linear regression model demonstrated a reasonable level of accuracy in predicting stock prices. The evaluation metrics, such as the R-squared score and mean absolute error, indicated the model's ability to capture the underlying patterns and trends in the stock market data. However, it is important to note that stock price prediction is inherently challenging due to the volatility and complexity of the financial markets.

2. Usefulness of Technical Indicators: The incorporation of various technical indicators, such as Bollinger Bands, MACD, RSI, SMA, and EMA, proved to be valuable for analyzing stock prices. These indicators provided insights into market trends, volatility, and momentum, allowing users to make informed decisions based on the visualized data. The availability of these indicators enhanced the overall usability and functionality of the application.

3. Real-time Data Analysis: The application's ability to display recent data allowed users to stay up-to-date with the latest stock market information. Users could monitor the recent performance of selected companies, identify patterns, and track changes in stock prices. This real-time data analysis capability facilitated timely decision-making and helped users respond to market dynamics.

4. User-friendly Interface: The user interface of the application, implemented with Streamlit, offered a seamless and intuitive experience. The sidebar menu provided easy navigation between different features, allowing users to visualize technical indicators, explore recent data, and make stock price predictions effortlessly. The simplicity and clarity of the user interface contributed to a positive user experience.

5. Scope for Further Enhancements: While the project achieved its objectives, there are several avenues for future improvements. This includes exploring more advanced machine learning algorithms, integrating additional data sources, expanding the range of technical indicators, and incorporating user feedback to enhance the accuracy and usability of the application.

Overall, the project demonstrated the potential of ML and Python in developing a stock price prediction application. It provided users with valuable insights into stock market trends, empowered them to make informed investment decisions, and showcased the practical application of ML techniques in the financial domain. The project serves as a foundation for further exploration and refinement of stock market prediction systems.

The completion of the stock price prediction project using linear regression and Streamlit opens up possibilities for future enhancements and expansions. Here are some potential avenues for further work:

1. **Advanced Machine Learning Models**: Explore the application of more advanced machine learning models for stock price prediction, such as ensemble methods (Random Forest, Gradient Boosting), support vector machines (SVM), or deep learning architectures (Recurrent Neural Networks, Convolutional Neural Networks). Compare the performance of these models with the existing linear regression model to identify which approach yields better accuracy.

2. **Feature Engineering**: Investigate additional features and indicators that can be incorporated into the prediction model. Consider fundamental data, news sentiment analysis, economic indicators, or social media sentiment to capture a wider range of factors influencing stock prices. Feature selection techniques and domain expertise can help identify the most relevant features for improving prediction accuracy.

3. **Hyperparameter Tuning**: Perform extensive hyperparameter tuning for the machine learning models. Utilize techniques such as grid search, random search, or Bayesian optimization to find the optimal combination of hyperparameters that maximize prediction performance. Fine-tuning the models can lead to improved accuracy and generalization.

4. **Ensemble Methods**: Explore ensemble techniques, such as model stacking or model averaging, to combine predictions from multiple models. Ensemble methods can often enhance the stability and robustness of predictions, particularly when dealing with volatile or noisy stock market data.

5. **Deployment and Scalability**: Optimize the application for scalability to handle larger datasets and increasing user demands. Consider deploying the application on cloud platforms or containerized environments to ensure efficient resource utilization and accommodate a growing user base.

6. **Real-time Data Integration**: Integrate real-time data feeds or APIs to provide users with up-to-date stock market information. This could include live streaming of stock prices, news updates, or social media sentiment analysis. Real-time data integration can enhance the application's value and keep users informed about market changes.

7. **User Feedback and Iterative Improvement**: Gather feedback from users and incorporate their suggestions to enhance the application's usability and functionality. Conduct user testing sessions to identify pain points, understand user needs, and refine the user interface. Iteratively improve the application based on user feedback to ensure it meets the requirements and expectations of its target audience.

8. **Expanded Data Analysis and Visualization**: Include additional data analysis and visualization features to provide users with deeper insights into stock market trends. This could involve more advanced charting techniques, interactive visualizations, correlation analysis, or sector-wise performance analysis. Expanding the range of analytical tools can empower users to make more informed investment decisions.

9. **Risk Assessment and Portfolio Optimization**: Extend the application to include risk assessment and portfolio optimization functionalities. Implement algorithms and models to analyze risk levels, calculate portfolio returns, and suggest optimal investment allocations based on user preferences and risk tolerance.
10. **Integration with Trading Platforms**: Integrate the application with popular trading platforms or brokerage APIs to provide users with a seamless trading experience. This could involve executing trades directly from the application, tracking portfolio performance, or receiving real-time trading alerts based on prediction models.

By pursuing these future works, the stock price prediction application can be further enhanced in terms of accuracy, usability, and functionality, ultimately providing users with a powerful tool for informed decision-making in the stock market.

CHAPTER 9 : REFERENCES

Several related works exist in the domain of stock price prediction and analysis. Here are some notable examples :

[1] P. Werawithayaset and S. Tritilanunt, "Stock Closing Price Prediction Using Machine Learning," 2019 17th International Conference on ICT and Knowledge Engineering (ICT & KE), Bangkok, Thailand, 2019.

[2] Ji, X., Wang, J. and Yan, Z. (2021), "A stock price prediction method based on deep learning technology", *International Journal of Crowd Science*, Vol. 5 No. 1.

[3] Polamuri, Dr & Srinivas, Kudipudi & Mohan, A.. (2020). A Survey on Stock Market Prediction Using Machine Learning Techniques.