Arun Rai

July 6, 2023

C964: Computer Science Capstone

Task 2 parts A, B, C, and D

[Part A: Project Proposal for Business Executives 3](#_Toc139818686)

[Project Recommandation 5](#_Toc139818687)

[Problem Summary 5](#_Toc139818688)

[Application Benefits 6](#_Toc139818689)

[Application Description 6](#_Toc139818690)

[Data Description 7](#_Toc139818691)

[Objectives and Hypothesis 7](#_Toc139818692)

[Methodology 7](#_Toc139818693)

[Funding Requirements 8](#_Toc139818694)

[Data Precautions 9](#_Toc139818695)

[Developer’s Expertise 9](#_Toc139818696)

[Part B: Project Proposal 10](#_Toc139818697)

[Problem Statement 10](#_Toc139818698)

[Customer Summary 10](#_Toc139818699)

[Existing System Analysis 11](#_Toc139818700)

[Data 11](#_Toc139818701)

[Project Methodology 12](#_Toc139818702)

[Project Outcomes 13](#_Toc139818703)

[Implementation Plan 14](#_Toc139818704)

[Evaluation Plan 16](#_Toc139818705)

[Resources and Costs 17](#_Toc139818706)

[Timeline and Milestones 18](#_Toc139818707)

[Part C: Application 19](#_Toc139818708)

[Part D: Post-implementation Report 20](#_Toc139818709)

[A Business (or Organization) Vision 20](#_Toc139818710)

[Datasets 21](#_Toc139818711)

[Data Product Code 22](#_Toc139818712)

[Training and Testing Model 24](#_Toc139818713)

[Data Analysis impact on Descriptive and Non-Descriptive Methods 25](#_Toc139818714)

[Objective (or Hypothesis) Verification 25](#_Toc139818715)

[Effective Visualization and Reporting 26](#_Toc139818716)

[Accuracy Analysis 29](#_Toc139818717)

[Application Testing 31](#_Toc139818718)

[Application Files 32](#_Toc139818719)

[User Guide 32](#_Toc139818720)

[Summation of Learning Experience 33](#_Toc139818721)

[Sources 34](#_Toc139818722)

# Part A: Project Proposal for Business Executives

Arun Rai

2285 Stonybrook Dr.

Yuba City, CA 95991

arai12@wgu.edu

916-432-3211

July 6, 2023

John Smith

Chief Investment Officer

Smith Capital Management

123 Ellis Street

New York, NY 10001

Subject: Proposal for a Data Product to Improve Trading Performance

Dear John Smith,

I hope you are doing well. I am writing to address a crucial challenge our hedge fund currently faces and to propose a solution that has the potential to significantly enhance our trading returns and win rates on discretionary trades.

It has come to my attention that our hedge fund's returns have been lower than previous years, particularly in the discretion/fundamental trading team. As you know, discretionary trades rely on the judgment and expertise of our traders. However, human decision-making can be influenced by various factors, leading to suboptimal outcomes. We need to find a way to improve the accuracy and consistency of our discretionary trades.

I would like to introduce a data product called a quantitative machine learning model that utilizes historical stock market data, specifically daily bars, to predict whether a stock will close green or red the next day. This model employs advanced algorithms to analyze vast amounts of information and identify patterns that can guide our trading decisions.

Implementing this proposed solution will yield several advantages for our hedge fund. By leveraging the predictive power of machine learning algorithms, the data product will increase our probability of making correct trade calls. Access to more reliable and insightful information will enable us to make more informed decisions, leading to a higher number of winning trades. Ultimately, this will contribute to better returns and position our hedge fund to outperform the market.

I have a done comprehensive overview of the costs associated with this project, even though some of terms here may be technical I want to provide a very transparent view of what are the costs of this project. Hardware investments, amounting to $10,000, cover essential infrastructure like servers, computers, storage devices, and networking equipment. All software licenses, development tools, and required libraries are open source and incur no additional expenses. Labor costs involve a development team of three professionals working for three months at an average rate of $100 per hour, resulting in an estimated cost of $36,000. Project management, overseen by a project manager for the same duration, amounts to $14,400 at an average rate of $120 per hour. Quality assurance and testing efforts, conducted by testers for one month at an average rate of $80 per hour, are projected to cost approximately $8,100. The estimated environment costs encompass a $2,000 expenditure for deployment, $2,400 for one year of cloud-based hosting services at $200 per month, and an annual expense of $5,000 for ongoing maintenance and support.

My professional and academic background (B.S in Computer Science) and experience make me well-suited to develop and implement this solution. I have successfully delivered multiple algorithms for financial market trading, including live trading algorithms and back testing systems that assist traders in identifying effective strategies. Unlike regular software engineers, I possess an in-depth understanding of financial markets, including terminology and associated risks. This knowledge enables me to collaborate effectively with team members from various departments, ensuring seamless integration and successful outcomes.

In conclusion, I strongly believe that the proposed data product will revolutionize our trading approach and significantly improve our hedge fund's performance. The benefits of enhanced decision-making capabilities and increased trading success justify the investment in the necessary resources.

Thank you for considering this proposal. I would be grateful for an opportunity to discuss the details further and address any questions or concerns you may have. Please find attached additional documentation supporting my proposal.

Yours sincerely,

Arun Rai

## Project Recommandation

Arun Rai

2285 Stonybrook Dr.

Yuba City, CA 95991

arai12@wgu.edu

916-432-3211

July 6, 2023

John Smith

Chief Investment Officer

Smith Capital Management

123 Ellis Street

New York, NY 10001

Subject: Follow-up Proposal for Project Implementation

Dear John Smith,

I hope this letter finds you well. I am writing to provide further details on the proposed project that aims to address the organizational needs of our hedge fund. This project centers around the development of a command line interface that will revolutionize the trading decision process and ultimately lead to higher gains for our fund.

### Problem Summary

The project entails the creation of a user-friendly command line interface that allows traders to enter a stock and obtain quantitative data and analysis instantly. This information includes the current price, current gain YTD%, and a prediction on whether the stock will close green or red the following day. By providing these features in an easily accessible format, we aim to simplify the trading decision process and assist traders in making more informed choices, ultimately resulting in higher returns for the fund.

The command line interface will be deployed on the trading floor, where live trades are placed and managed. Accessibility is crucial in this setting, as traders require instant access to pertinent information before making trading decisions. The project is particularly needed to support the discretion/fundamental traders on the floor, who have experienced lower returns over the past year. Introducing new technology at this juncture will enhance their trading environment and provide them with the tools necessary to improve their performance. It is worth noting that the quantitative traders at our hedge fund, who rely on automated systems, are not present on the trading floor and therefore do not require the same level of human intervention.

The proposed project aligns with the business objectives of our hedge fund, which are centered around achieving higher returns for our investors. By harnessing the power of cloud computing, machine learning, and big data, our application will equip traders with cutting-edge technology to analyze stocks and make informed trading decisions. This will ultimately lead to increased trading success, improved returns for the fund, and the ability to outperform the market.

The deliverables for this project include the development of a command line interface that enables users to choose a specific stock and obtain quantitative analysis for that stock. Additionally, a predictive model utilizing machine learning techniques will be created to determine whether the selected stock is likely to close higher (green) or lower (red) the following day. Also, users will be able to view images that describe the data such as pie charts and histograms. By delivering these components, the objective is to enhance the performance of the hedge fund, ultimately leading to increased returns. The command line interface will provide users with valuable insights and data-driven information on stocks, enabling informed investment decisions, while the predictive model will leverage advanced algorithms to forecast potential stock movements, assisting in optimizing trading strategies and generating higher profits for the hedge fund.

### Application Benefits

Expanding on the benefits mentioned earlier, the command line interface will provide several advantages to our organization. Firstly, traders will have immediate access to quantitative data and analysis, enabling them to make decisions based on reliable and up-to-date information. Secondly, the predictive capabilities of our machine learning algorithms will enhance the accuracy of trade predictions, leading to a higher win rate. Additionally, by integrating this application into our trading environment, we empower our traders to take advantage of data-driven insights and optimize their trading strategies.

### Application Description

From a technical standpoint, our application will utilize historical stock market data obtained through the Yahoo Finance API endpoint. The data will primarily consist of quantitative information such as historical stock market close prices. It will be structured using nested dictionaries, where each dictionary represents a different stock, and each key within the dictionary represents a specific date and its corresponding stock close price.

The application will employ machine learning methods, such as decision tree and random tree algorithms, to identify patterns and predict future stock prices. It will detect anomalies and outliers, alerting traders to sudden price movements that may require caution. By combining historical data analysis, statistical anomalies, and machine learning techniques, our application aims to provide traders with reliable insights into stock performance.

### Data Description

The raw data for this project is sourced from Yahoo Finance through their API endpoint. The data type is quantitative and specifically consists of historical stock market closing prices. The chosen data structure will mostly be arrays, the array will contain objects that each represent one daily bar in the stock market which contains an Open price, High price, Low price, Close price, and Date. In terms of variables, time is considered an independent variable, while the dependent variable is the stock price. An important aspect of this data analysis is accounting for anomalies or limitations. For instance, outliers may be present in the stock data, and the data product incorporates checks for such anomalies. One notable anomaly that the product examines is the percentage difference in stock prices between two consecutive days. Generally, the stocks traded by our hedge fund, particularly blue-chip stocks, tend to experience relatively small daily price fluctuations. Therefore, if the data product detects a sudden spike in the percentage difference, exceeding a threshold like 10%, it will issue a cautionary warning to the trader that there may be false data detected, alerting them to the occurrence and urging caution in response.

### Objectives and Hypothesis

The desired outcome of this project is to achieve increased returns, particularly within the discretion/fundamental department of our hedge fund, leading to improved overall investor returns. While we cannot provide an accuracy prediction or hypothesis within the scope of this proposal, we anticipate that pairing our data product with live traders will result in returns of approximately 25% per year and a win rate of 75%, surpassing the current rates of 15% return per year and 50-60% win rate respectively. The accuracy and prediction will be difficult to quantify without live trades paired with real traders because this is not a standalone product.

### Methodology

The chosen methodology for the project is Agile development. Given the dynamic nature of the market, agile methodologies allow for flexibility, rapid iterations, and continuous feedback integration. The project will follow a phased approach, including setting up a connection to Yahoo Finance for data retrieval, implementing mathematical equations to obtain non-prescriptive data, incorporating machine learning libraries, enhancing the user-friendly command line interface, debugging, and gathering feedback from traders. The methodology ensures constant adaptation to market needs and efficient delivery of the project.

**1. Project Setup:**

-Establish a connection to Yahoo Finance API to retrieve real-time and historical stock data.

-Set up the development environment, including necessary libraries and dependencies.

**Data Acquisition and Preprocessing:**

-Retrieve the required stock data, such as historical prices, volumes, and relevant financial indicators.

-Implement mathematical equations and calculations to derive non-predictive data, such as moving averages, standard deviations, and other statistical measures.

**2. Machine Learning Model Development:**

-Integrate machine learning libraries, such as scikit-learn or TensorFlow, into the project.

-Prepare the data by cleaning, transforming, and encoding it for use in the machine learning algorithms.

-Utilize appropriate supervised learning techniques, such as classification models, to train the predictive model.

-Fine-tune the model parameters and evaluate its performance using suitable evaluation metrics.

**3. Command Line Interface Development:**

-Design and implement a user-friendly command line interface (CLI) that enables users to interact with the program.

-Develop functionality for users to input their chosen stock and access quantitative analysis and predictions.

-Incorporate visualizations, such as pie charts and histograms, to enhance data representation.

**4. Testing and Debugging:**

-Conduct rigorous testing of the command line interface, ensuring its stability and functionality.

-Debug any issues or errors encountered during testing.

-Collect feedback from traders or users to identify areas for improvement and enhancement.

**5. Deployment and Delivery:**

-Package the project into a deployable form, ensuring all dependencies are included.

-Create documentation and user guides to aid in the deployment and utilization of the software.

-Deliver the finalized project, including the command line interface, predictive model, and supporting documentation, to the hedge fund.

**6. Post-Delivery Support:**

-Provide ongoing support and maintenance for the software, addressing any potential issues or bugs that may arise.

-Incorporate updates and enhancements based on user feedback or evolving market requirements.

### Funding Requirements

The resources and costs associated with the project include hardware and software expenses. The hardware investment amounts to $10,000, covering servers, computers, storage devices, and networking equipment. On the software front, all licenses, development tools, and necessary libraries are open source and come at no additional cost. In terms of labor, the development team, comprising three developers, dedicated three months to the project, with an average rate of $100 per hour. The estimated cost for their services is $36,000. Project management was handled by a project manager for the same duration, at an average rate of $120 per hour, resulting in an estimated cost of $14,400. Quality assurance and testing were conducted by testers for one month, with an average rate of $80 per hour, totaling an estimated cost of $8,100. Regarding the environment costs, deployment required $2,000 for server setup and configuration. Cloud-based hosting services were employed, amounting to $200 per month, leading to an estimated cost of $2,400 for one year. Lastly, ongoing maintenance and support expenses were projected at $5,000 per year.

### Data Precautions

Data security and privacy were paramount considerations in the development of this project. To safeguard the sensitive information, we implemented several precautions. Firstly, a username and password authentication system were integrated at the start of the program. This ensured that only authorized individuals could access the application and its functionalities. To enhance security, we utilized hashing techniques to store and verify user credentials, providing an additional layer of protection against unauthorized access. The main reason for this layer of security was the need to protect the results generated by the predictive model, as they hold valuable insights for our hedge fund. With the focus on user-end security, we took measures to ensure the confidentiality and integrity of the data. This project itself does not employ code encryption, as it is unlikely that people in the hedge fund setting have high amounts of technical knowledge to extract information from the source code.

### Developer’s Expertise

As the developer of this project, I possess a strong academic background, holding a B.S. in Computer Science. Furthermore, I have acquired substantial professional expertise and experience in delivering algorithms for financial market trading, including live trading algorithms and back testing systems. This has provided me with an in-depth understanding of financial markets, including terminology and associated risks. My diverse skill set, encompassing Python programming, server maintenance, data structures, machine learning, and effective communication, positions me well to deliver this solution seamlessly in collaboration with team members from various departments.

Thank you for considering this follow-up proposal. I would be delighted to discuss the project in more detail and address any questions or concerns you may have. Attached to this letter, you will find additional documentation supporting the project.

Yours sincerely,

Arun Rai

# Part B: Project Proposal

## Problem Statement

The problem that the hedge fund will face in the future is the lack of advanced technology to provide quantitative data and analysis, essential for supporting traders in making decisions based on objective evidence rather than short-term emotions or intuition. Specifically, discretionary, and fundamental traders, who heavily rely on human instinct and intuition, have been underperforming compared to other trader groups within the hedge fund, with returns falling below the target of 20% per year, currently standing at less than 15%. To address this issue, the hedge fund aims to introduce cutting-edge technology encompassing quantitative analysis and machine learning capabilities, which will provide traders with increased conviction, objective evidence, and diverse perspectives, ultimately resulting in more successful trades. The task at hand involves developing user-friendly software tailored for non-technical traders, enabling them to access and interpret quantitative data for any desired stock. This will be accomplished by establishing a streamlined data stream to acquire the latest historical stock market prices, implementing machine learning algorithms and equations to manipulate and transform the data into meaningful insights, and creating an intuitive user interface for seamless interaction.

## Customer Summary

The clients or customers of the proposed data product will be the fundamental and discretionary traders at our hedge fund. These traders will employ a distinct trading style that will not solely rely on historical evidence but instead prioritize future thinking, intuition, and considerations such as current market trends, politics, and the uniqueness of the trading environment. Although quantitative analysis has not been their primary approach in the past, the integration of quantitative analysis and machine learning techniques will prove highly beneficial for these traders in the future. The application will provide them with an additional layer of insights, objective evidence, and alternative perspectives that will complement their discretionary decision-making process. Through advanced algorithms and models, quantitative analysis will unveil underlying patterns and correlations in extensive historical and real-time data, empowering traders to make well-informed predictions and strategic choices. Machine learning algorithms will further enhance their abilities by learning from historical market behavior and generating predictive analytics to forecast market movements and identify potential trading opportunities which will lead to more gains for the hedge fund. By incorporating quantitative analysis and machine learning, the proposed data product will augment the decision-making capabilities of fundamental and discretionary traders, enabling them to harness data-driven insights while retaining their expertise in interpreting market nuances and unique factors.

## Existing System Analysis

The current tools or applications used by the client are the platforms provided by big-name brokers, which are available to the public. These platforms, although widely accessible, have certain shortcomings that highlight the need for our solution. The platforms provided by these brokers are often basic and lack comprehensive features. They typically offer limited functionality, providing users with only the current stock price and historical price data. They do not delve into in-depth quantitative analysis of the stocks being viewed, nor do they extract meaningful statistical insights that can help identify patterns in the market or predict future prices. Moreover, these platforms do not employ machine learning algorithms for price prediction. By contrast, our proposed platform will provide our traders with a wealth of information, presented in a user-friendly manner similar to what they are accustomed to. Our platform will equip them with the tools necessary to access quantitative analysis and gain an edge in the market. They will have access to valuable insights, comprehensive statistical analysis, and predictive capabilities, enabling them to make more informed decisions, achieve higher success rates in their trades, and ultimately outperform the majority of traders relying solely on the limited offerings of big-name broker platforms.

## Data

The raw data set for this project consists of financial data in the form of stock prices. The data is structured in rows, where each row represents one trading day and contains attributes such as the opening price, highest price, lowest price, closing price, timestamp, and trading volume. These attributes provide essential information for analyzing and understanding stock market trends and patterns.

Throughout the application development life cycle, data will be collected, processed, and managed in a specific manner. Instead of storing the data on external devices like hard drives or servers, it will be stored in memory. Whenever a user queries a stock, the data will be retrieved from the memory, ensuring real-time and up-to-date information. This approach eliminates the need for maintaining databases and simplifies the overall data management process. Although it involves frequent queries from the data provider, Yahoo Finance, this approach offers greater simplicity and reduces security concerns. As the data set is not very large, this in-memory storage and retrieval mechanism is efficient and ensures the most recent data is always available.

Regarding data anomalies, mechanisms will be implemented to handle outliers, incomplete data, and other irregularities. Traders will actively participate in the verification process by visually inspecting line charts generated from the data. This visual analysis will allow them to quickly identify any glaring inconsistencies or anomalies in the data. Additionally, algorithms will be in place during data processing, ensuring that unusual price movements or timestamps are flagged and handled appropriately. By combining both human inspection and automated algorithms, the system will effectively handle data anomalies, ensuring the integrity and reliability of the data used for quantitative analysis and decision-making processes.

## Project Methodology

We will apply the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology with an agile approach. This widely recognized and accepted methodology provides a structured approach for developing and deploying machine learning solutions. We will follow the CRISP-DM framework, including phases such as business understanding, data understanding, data preparation, modeling, evaluation, and deployment. These phases will guide us through tasks like defining project objectives, exploring, and preparing the data, building, and refining machine learning models, evaluating their performance, and finally deploying the command line interface application. By adhering to this standard methodology, we ensure a systematic and well-organized implementation process for our project.

To ensure a comprehensive understanding of the CRISP-DM methodology's application to our specific project, we will follow a detailed approach in each phase:

Business Understanding: The project's primary objective is to enhance the returns for our hedge fund by incorporating a quantitative approach to stock price prediction in addition to the existing fundamental analysis. This will involve exploring and leveraging machine learning and statistical techniques to generate accurate predictions for blue-chip stocks.

Data Understanding: We will utilize financial stock price data, which includes timestamps, numeric values, and character information. The data will be sourced from various providers, including in-house stored data, Yahoo Finance, NASDAQ, and Polygon. Additionally, we will ensure regular data updates at the end of each trading day. During this phase, we will conduct thorough data exploration to understand its quality, format, and relevance to the project.

Data Preparation: As we gather data from different sources, we anticipate potential challenges in standardizing the data. We will address issues such as adjusting timestamp formats, accounting for stock-splits, managing time zones, and handling holidays. This is also where data anomalies will be addressed and how to solve them such as algorithms to detect large price movements in stocks that have low volatility to possibly signal bad data. These preprocessing steps will ensure the data is consistent and suitable for subsequent modeling.

Modeling: In the modeling phase, we will apply a combination of general statistical methods and machine learning techniques. Our approach will involve utilizing descriptive statistics such as mean, median, mode, range, maximum, minimum, percentage increase, and percentage loss to gain insights into the data. Additionally, we will employ machine learning algorithms such as linear regression and decision trees to develop predictive models that capture the relationships between variables and generate accurate predictions.

Evaluation/Deployment: To evaluate the performance of our model, we will analyze the hedge fund's returns after the model's implementation. We will also assess the success of specific trades executed based on the model's predictions. Verification data will be utilized to cross-check the accuracy of the predictions against actual outcomes. Additionally, robust data update mechanisms and other precautions will be implemented to ensure the reliability and effectiveness of the deployed solution.

## Project Outcomes

The main deliverable of this project will be a Python program featuring a command line interface designed for traders' interaction. Traders will have the ability to select a stock of their choice, focusing on blue-chip stocks, and the program will prompt them with specific questions regarding the stock. The program will provide various deliverables, including statistical analysis and predictions for the upcoming trading session. Visual representations derived from the stock price data will be included, such as a 10-year price chart, a pie chart displaying the number of green and red years within the past decade, and a histogram illustrating the volatility of each trading year. The statistical analysis component will generate numeric values derived from the financial data, such as Year-to-Date percentage gain, percentage gain over the last 10 years, percentage gain over the last 30 days, and largest drawdown, among others. Moreover, the prediction model will offer forecasts on whether the selected stock will close green or red the next day, providing traders with additional insights to strengthen their trading decisions. The model will also provide results from previous days, enabling traders to evaluate the accuracy of its predictions and make informed judgments based on historical performance.

**User Guide:**

**Installation and Setup:**

Ensure that you have Python installed on your system (version 3.0 or higher).

Download the program files from the provided source.

Open your preferred command line interface (e.g., Terminal, Command Prompt).

Navigate to the directory where the program files are located.

Install the required dependencies by running the command: pip install -r requirements.txt.

**Running the Program:**

Open your command line interface.

Navigate to the directory where the program files are located.

Run the program by executing the command: python stock\_analysis.py.

Login by entering the username and password in Part D (user guide) of this paper.

The program will display a welcome message and prompt you to select a stock of your choice (limited to blue-chip stocks).

**Stock Selection and Analysis:**

Enter the symbol or code for the desired stock when prompted.

The program will present you with various analysis options.

Select the desired analysis by entering the corresponding number or keyword.

**Available Analysis Options:**

**Statistical Analysis:**

YTD Percentage Gain: Displays the percentage gain of the stock from the beginning of the current year.

10-Year Percentage Gain: Shows the percentage gain of the stock over the past 10 years.

30-Day Percentage Gain: Provides the percentage gain of the stock in the last 30 days.

Largest Drawdown: Indicates the largest drop or decline experienced by the stock during the selected period.

**Prediction Model:**

Next Day Prediction: Forecasts whether the selected stock will close green or red the following trading day(or current day if market is still open).

Historical Prediction Accuracy: Displays the accuracy of the prediction model based on previous trading days as well as the results of previous trading days that went into the calculation.

**Data Visualization (displayed after Statistical Analysis is selected):**

The program will also generate visual representations of the stock price data for better insights:

10-Year Price Chart: Displays a line chart showing the price movement of the stock over the past 10 years.

Green/Red Years Pie Chart: Presents a pie chart depicting the distribution of green and red years within the last decade.

Volatility Histogram: Shows a histogram representing the volatility of each trading year.

## Implementation Plan

**General Strategy:** The general strategy for implementing the project will involve a combination of agile and CRISP-DM methodologies to ensure an iterative and structured approach to development. The project will emphasize regular collaboration, flexibility, and adaptability while following the data mining process.

**Phases of the Rollout:**

**1. Project Initiation:**

-Define project objectives, requirements, and success criteria.

-Set up the development environment and establish project infrastructure.

-Identify the key stakeholders and their roles.

**2. Agile Development Iterations:**

-Adopt an agile framework (such as Scrum or Kanban) for iterative development.

-Plan and prioritize features and functionalities based on user stories and feedback.

-Conduct regular sprints or work cycles to develop and refine the program incrementally.

-Continuously collaborate with stakeholders to gather feedback and make necessary adjustments.

**3. CRISP-DM Phases:**

-Business Understanding: Understand the objectives, requirements, and constraints of the project. Define the scope and success criteria.

-Data Understanding: Explore and analyze the raw data set, identify relevant attributes, assess data quality, and perform initial data preprocessing.

-Data Preparation: Cleanse, transform, and integrate the data for further analysis and modeling. Handle missing data, outliers, and inconsistencies.

-Modeling: Apply machine learning algorithms to build predictive models based on the prepared data. Fine-tune the models and evaluate their performance.

-Evaluation: Assess the effectiveness and accuracy of the models through validation and testing. Determine the strengths and limitations of the models.

-Deployment: Integrate the developed models and features into the program. Conduct further testing and quality assurance to ensure stability and functionality.

-Maintenance: Provide ongoing support, monitoring, and updates as required. Address any issues or bugs that arise during real-world usage.

**Dependencies:** The project implementation will depend on several factors, including:

-Availability of the raw data set from the data provider (Yahoo Finance API).

-Access to relevant software libraries and tools for data analysis, machine learning, and visualization.

-Adequate computing resources and infrastructure to handle data processing and storage requirements.

**Testing and Distribution:**

-Testing: Implement a robust testing strategy, including unit testing, integration testing, and system testing, to ensure the functionality, accuracy, and reliability of the program. Conduct regular reviews and quality checks during each development iteration.

-Distribution: Once the program is stable and meets the desired quality standards, it can be distributed to the traders within the hedge fund. This can be done through deployment on internal servers or by providing executable files or installation packages.

By combining the agile and CRISP-DM methodologies, the project implementation will maintain a balance between flexibility, responsiveness, and structured data analysis. Regular feedback and collaboration with stakeholders will guide the development process, while adherence to the CRISP-DM phases will ensure a systematic approach to data understanding, preparation, modeling, evaluation, deployment, and maintenance.

## Evaluation Plan

**Verification Methods:**

**1. Stage: Project Initiation**

-Verification Method: Review and approval by project stakeholders to ensure alignment with objectives, requirements, and success criteria.

**2. Stage: Agile Development Iterations**

-Verification Method: Continuous integration and unit testing to verify the functionality and correctness of newly developed features.

-Peer code reviews to identify and address any coding errors, inconsistencies, or best practice violations.

-User acceptance testing (UAT) by stakeholders to validate that the implemented features meet their expectations and requirements.

**3. Stage: CRISP-DM Phases**

-Verification Method: Data quality assessment and data preprocessing checks to ensure the accuracy, completeness, and consistency of the data.

-Model evaluation through various validation techniques such as cross-validation, hold-out validation, or k-fold validation, to assess the performance and accuracy of the predictive models.

-Regular meetings and checkpoints with stakeholders to verify that the project is progressing as planned, meeting the defined objectives and success criteria.

**Validation Method:** Upon completion of the project, a comprehensive validation process will be conducted to ensure the overall effectiveness and suitability of the developed solution.

**1. Validation Method: User Testing**

-Traders and end-users will engage in extensive testing of the application.

-They will perform various tasks and scenarios to evaluate the usability, functionality, and performance of the software.

-Feedback and observations from users will be collected and analyzed to identify any areas for improvement or necessary adjustments.

**2. Validation Method: Performance Testing**

-Stress testing and load testing will be conducted to evaluate the system's performance under heavy usage and ensure it can handle the expected workload.

-Performance metrics such as response time, throughput, and resource utilization will be measured and compared against defined benchmarks.

**3. Validation Method: Comparison and Benchmarking**

-The developed application will be compared to existing tools or platforms commonly used by traders in terms of functionality, usability, and accuracy.

-Benchmarking against industry standards and best practices will be performed to assess the solution's performance and capabilities.

Through a combination of verification methods during the development stages and a thorough validation process at project completion, the evaluation plan ensures the quality, effectiveness, and alignment of the developed solution with stakeholder expectations and project objectives. Feedback from users and performance testing will provide valuable insights for future improvements and optimizations.

## Resources and Costs

**1. Hardware and Software Costs:**

-Hardware: $10,000 for servers, computers, storage devices, and networking equipment.

-Software: Free, all software licenses, development tools, and necessary libraries are open source.

**Estimated Labor Time and Costs:**

-Development Team: 3 developers working for 3 months at an average rate of $100 per hour. Estimated cost: $36,000.

-Project Management: Project manager working for 3 months at an average rate of $120 per hour. Estimated cost: $14,400.

-Quality Assurance and Testing: Testers working for 1 month at an average rate of $80 per hour. Estimated cost: $8,100.

**Estimated Environment Costs:**

-Deployment: $2,000 for server setup and configuration.

-Hosting: $200 per month for cloud-based hosting services. Estimated cost for 1 year: $2,400.

-Maintenance: $5,000 per year for ongoing maintenance and support.

## Timeline and Milestones

|  |  |  |
| --- | --- | --- |
| **Milestone** | **Start Date** | **End Date** |
| Project Initiation (Requirements Gathering) | July 1, 2023 | July 7, 2023 |
| Setting up Data Connections | July 8, 2023 | July 15, 2023 |
| Machine Learning Model Development | July 16, 2023 | August 1, 2023 |
| Image/Statistical Analysis Development | August 2, 2023 | August 15,2023 |
| Develop Command Line Interface | August 16, 2023 | September 1, 2023 |
| Validation and Testing | September 2, 2023 | September 10, 2023 |
| Deployment and Release | September 11, 2023 | September 19, 2023 |
| Maintenance and Support | September 20, 2023 | Ongoing |

# Part C: Application

\final\_capstone

\account\_registration.py

\config.ini

\main.py

\requirements.txt

\C964\_task\_2\_written.docx

# Part D: Post-implementation Report

## A Business (or Organization) Vision

The problem faced by the business was the lack of advanced technological tools to provide traders with quantitative data and analysis, hindering their decision-making process. Discretionary and fundamental traders, who heavily relied on human instinct and intuition rather than historical evidence, were underperforming compared to other trader groups at the hedge fund, resulting in returns below the desired target of 20%. To address this problem, the organization envisioned developing an application that would leverage quantitative analysis and machine learning to enhance trading decisions and improve overall returns.

The application developed by the organization successfully resolved the problem by providing traders with a powerful tool for quantitative analysis and predictive insights. Through a command-line interface, traders could interact with the application and select the desired stock for analysis. The application offered a range of features, including statistical analysis, visualizations, and predictive modeling. Traders could access statistical metrics such as percentage gains over different time frames, largest drawdowns, and volatility histograms. Moreover, the application incorporated a predictive model that determined the likelihood of a selected stock closing green or red the next day, adding a data-driven dimension to traders' decision-making process.

By using the application, traders could solve the problem of relying solely on intuition and subjective analysis. They could leverage the power of quantitative analysis and machine learning to gain objective insights and enhance their trading decisions. The application empowered traders to make informed choices based on statistical data, visualizations of stock price trends, and predictive modeling results. They could assess the historical performance of stocks, identify patterns and trends, and predict potential future movements. This enabled traders to have a competitive edge in the market, make more winning trades, and ultimately achieve higher returns for the hedge fund.

For example, one morning, a trader is analyzing the market after the release of overnight earnings reports. Based on their interpretation of the information, they perceive a bullish sentiment and consider buying a particular stock the following day. However, this trader has experienced mixed results with this strategy in the past, with an accuracy rate of around 50%. This time, they decide to leverage the predictive model provided by our solution. To their surprise, the model predicts that the stock will close in the red the next day, contrary to their initial bullish assumption. The trader now finds themself with conflicting information, prompting them to think twice about the trade. As a result, they exercise caution and refrain from placing any trade at all. This decision proves to be wise as the stock ultimately declines, validating the prediction made by the model. By using our application and heeding the insights from the predictive model, the trader avoids a potentially losing trade and demonstrates the value of leveraging quantitative analysis for informed decision-making.

## Datasets

The raw data used for this project consisted of financial data in the form of historical stock prices. The raw data set included attributes such as the opening price, highest price, lowest price, closing price, volume, timestamp, and other relevant information. The data was organized in a structured format (Pandas DataFrame), with each row representing a specific date and containing the corresponding stock price attributes. The raw data set was collected from the Yahoo Finance API, ensuring its accuracy and reliability.

To process the raw data, minimal processing was required. The primary focus was on making the data accessible and usable by the algorithm. This involved extracting the necessary attributes from the raw data set and storing them in an appropriate data structure(arrays) for analysis. The data was processed to ensure it could be efficiently retrieved and manipulated by the algorithm during runtime. Additionally, any missing or incomplete data points were handled by applying appropriate handling techniques, such as imputation or excluding those data points from the analysis.

The processed data was organized and structured to facilitate efficient access and manipulation by the algorithm. The data was made accessible through a memory-based storage approach, eliminating the need for external devices or databases. This allowed for real-time retrieval and processing of the most up-to-date data for each stock query. By streamlining the data processing and making it readily available within the application, the algorithm could effectively leverage the processed data for quantitative analysis and prediction modeling, enabling traders to make informed decisions based on the latest market information.

**Raw Data in form of Pandas DataFrame (Stock: AAPL):**

A screenshot of a computer screen

Description automatically generated

**Processed Data in form of an array (Stock: AAPL, ctrl + scroll wheel to zoom):**

A black and white screen

Description automatically generated

The code utilizes the yfinance library's yf.download function to dynamically query Yahoo Finance for historical stock data. As the data is fetched in real-time during code execution (in order to always get the latest stock data as prices are always changing), there are no pre-existing datasets used. Therefore, no access to specific datasets can be provided as the data is obtained dynamically from Yahoo Finance. However, the code that is used to query the data is provided below.

(“company” is the stock symbol in string formal, and “start” is a datetime object that represents the start of historical data. The “end” is not specified as this tells the query to get the latest data available. The variable “data” will be a Pandas Data Frame.)

****

## Data Product Code

To process the raw data, the code implemented a streamlined approach. It extracted the necessary attributes from the raw data set, such as opening prices, high and low prices, closing prices, timestamps, volume and then stored them in an array. This extraction process ensured that relevant data was available for further analysis. The code also handled any incorrect data points through appropriate algorithms that identify unusual patterns like large unrealistic price fluctuations.

For statistical analysis, the code implemented mathematical operations on the raw data obtained from the Yahoo Finance API. Various attributes, such as the highest and lowest prices, were extracted and used to calculate relevant statistical metrics. These calculations included determining percentage gains over different timeframes, identifying the largest drawdowns, and assessing volatility. By applying mathematical operations to the data, the code generated valuable insights into the performance and behavior of the selected stocks. These statistical metrics provided traders with quantitative measures that aided in their decision-making process. Additionally, the code incorporated visualizations (using Matplotlib) to complement the statistical analysis. Pie charts were utilized to depict the distribution of green and red years, offering a visual representation of positive and negative outcomes. Histograms were employed to showcase the volatility of trading years, allowing traders to identify the range and frequency of price movements. Furthermore, price charts were generated to illustrate the historical trends of stock prices over a specified time period, enabling traders to visualize patterns and trends. These visualizations enhanced the understanding of the data and facilitated the interpretation of statistical analysis results.

*Statistical Analysis Example (Stock: GOOG):*

A screenshot of a computer

Description automatically generated

*Visualization Example (Stock: GOOG)*

A screenshot of a graph

Description automatically generated

Furthermore, the code implemented a predictive model using the decision tree algorithm from the scikit-learn library. The decision tree algorithm is a powerful tool for classification tasks, making it suitable for predicting whether a stock will close green or red the following day. The code trained the decision tree model using historical data and relevant features, such as the closing price, open price, high price, low price, and volume. It then utilized the trained model to make predictions based on the provided input. The decision tree algorithm offered a straightforward and interpretable approach for predicting stock outcomes, contributing to the application's reliability and effectiveness.

For the predictive model, the code implemented the decision tree algorithm using daily bars as the analytical method. The decision tree algorithm is a supervised learning technique that is well-suited for classification tasks, making it an appropriate choice for predicting whether a stock will close green or red the next day. The algorithm works by constructing a tree-like model of decisions and their possible consequences based on input features.

The decision tree algorithm was deemed appropriate for the project due to several reasons. Firstly, it can handle both numerical and categorical data, allowing for the inclusion of various features relevant to stock price prediction, such as closing price, volume, etc. This flexibility ensured that the model could capture the diverse factors influencing stock price movements. Additionally, decision trees are easy to interpret and provide transparent decision-making paths, enabling traders to understand and validate the reasoning behind the model's predictions.Furthermore, decision trees excel in handling nonlinear relationships and interactions between features, making them suitable for capturing complex patterns in financial data. The algorithm's ability to perform well with noisy and incomplete data also made it well-suited for the project, as financial data can often contain fluctuations and missing values. The decision tree algorithm provided a robust and versatile method for predicting stock outcomes based on historical data, enhancing the application's predictive capabilities, and assisting traders in making informed trading decisions.

# Training and Testing Model

The decision tree model was trained and tested using a specific set of features and labels. The features included the open, high, low, close, and volume for each trading day, while the label represented whether the price closed green or red for that particular day. The training and testing processes were conducted separately, ensuring the model's reliability and accuracy.

The training data spanned from January 2, 2013, to December 31, 2022, encompassing a substantial historical period. This data was used to train the decision tree model initially, enabling it to learn from the patterns and relationships within the provided features and labels.

The testing data, on the other hand, covered a more recent timeframe, starting from January 1, 2023, and extending up to the present day. What sets this model apart is its dynamic nature. For each trading day in the testing period, the model was retrained using the latest available data, ensuring it was continuously updated with the most recent market trends. This approach allowed the model to adapt to the evolving dynamics of the financial market.

Once retrained with the latest data, the model utilized the knowledge acquired during training to make predictions for the next trading day. Specifically, it would look back at the preceding 100 trading days from the current day and apply the learned methods and patterns to those historical data points. Based on this analysis, the model provided a conclusion of whether the price would close green or red for the next trading day.

By training and testing the decision tree model with historical and up-to-date data, the application ensured that traders received reliable and timely predictions based on the latest market conditions. This iterative and dynamic approach enhanced the model's predictive capabilities and allowed it to adapt to changing market trends, providing valuable insights for informed trading decisions.

# Data Analysis impact on Descriptive and Non-Descriptive Methods

The data analysis performed in this project played a pivotal role in supporting the selection and improvement of both descriptive and non-descriptive methods. Through the examination of histograms and price charts, a notable observation emerged regarding the dynamic nature of the market, with significant changes occurring within short timeframes despite long periods of consistency. This insight led to the decision to retrain the decision tree model with the latest available data, acknowledging the constant evolution and evolving trends within the market. By incorporating this adaptive approach, the model could capture the most recent market dynamics and enhance its predictive capabilities. The data analysis also provided valuable insights into the market's behavior, guiding the selection of relevant features and patterns for statistical metrics, such as percentage gains and volatility. These findings further informed the project's methodology, ensuring the alignment of methods with the dynamic nature of the financial market and ultimately enhancing the accuracy and reliability of the application's predictions.

Access to all libraries used in this project will be available in the "requirements.txt" file for evaluators to replicate and develop the project.

## Objective (or Hypothesis) Verification

The project had two main objectives: first, to improve the returns of the hedge fund, and second, to develop a working predictive model with an accuracy of at least 50%. The primary goal was to enhance the performance of the hedge fund's traders by providing them with data-driven insights and a reliable tool for making informed trading decisions. The technical objective aimed to develop and implement a predictive model that could accurately forecast whether a selected stock would close green or red the following day.

The objective of improving the returns of the hedge fund cannot be definitively proven without long-term usage by the traders. However, the project laid the foundation for achieving this objective by equipping the traders with quantitative analysis, statistical metrics, and the predictive model. By leveraging these tools, the traders gained valuable insights and increased confidence in their decision-making processes.

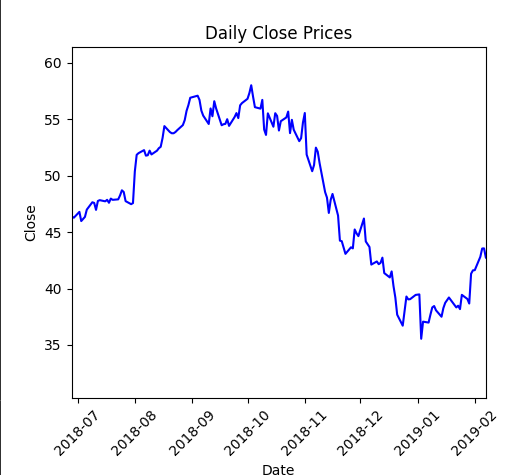
Regarding the technical objective, the project successfully met and even exceeded expectations in certain instances. For instance, when testing the predictive model on the stock AAPL over the last six months, it achieved an impressive accuracy rate of 61%. This level of accuracy was achieved through rigorous experimentation with different machine learning models and variations in input sizes of training data. The thorough testing process allowed for the identification of the optimal solution that yielded the highest accuracy. It is also worth noting that a higher accuracy model was not the goal of this project, that is because we do not want the prediction model to override the discretionary and fundamental traders’ decision making. Also, when the accuracy of model used for the stock market is too high, around 80%+ that can signal that you are overfitting the data and usually gives much different accuracy rates when applied to the live market.

## Effective Visualization and Reporting

Statistical analysis and visualizations played a crucial role in supporting the decision tree development process by providing insights and patterns within the data. During data exploration, it was observed that blue-chip stocks tend to exhibit extended uptrends with minimal downside volatility. For instance, analyzing the price data of AAPL from 2010 to 2018 revealed a sustained uptrend with limited downside fluctuations. This observation led to the idea of applying machine learning models that take numeric data (price), such as the decision tree, to identify patterns during these uptrends.

Further data analysis delved into specific statistical metrics, such as the largest drawdown for stocks. It was discovered that the largest drawdowns often had a relatively short duration, as exemplified by AAPL's largest drawdown lasting around three months. This information was taken into consideration for the decision tree model, highlighting the importance of incorporating the latest available data as soon as it became accessible. By continuously updating and training the model with the most recent data, potential changes in market conditions could be captured promptly, allowing for timely adjustments to trading strategies.

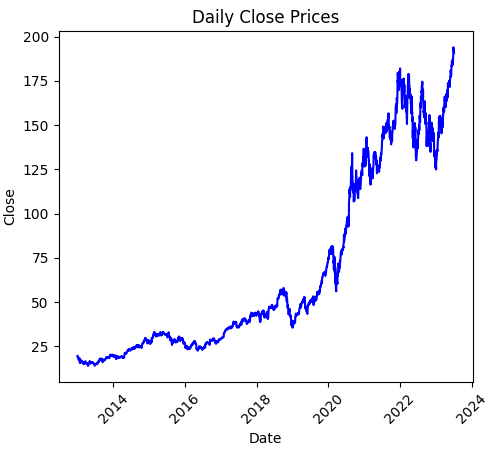
*AAPL Largest Drawdown (Using zoom function on matplotlib):*



The analysis of price charts revealed the phenomenon of sustained long-term growth, wherein certain stocks exhibited consistent upward trends that persisted for several consecutive years. This observation played a pivotal role in the development of the decision tree model by highlighting the importance of capturing and incorporating these sustained growth patterns into the algorithm's training process (adding volume on the training features which usually increases as price goes up). Recognizing the significance of these trends, the decision tree model was designed to identify and leverage such periods of sustained growth as potential predictors of future positive outcomes. By incorporating historical data from these extended periods of upward trends, the model could better assess the likelihood of future price increases and make more accurate predictions. This utilization of sustained growth patterns from the price charts allowed the decision tree model to capture and incorporate valuable insights into its decision-making process, leading to improved accuracy and effectiveness in predicting the future performance of selected stocks.

Histograms play a critical role in providing valuable insights into the volatility of a stock's price movements, complementing the information extracted from the price chart. While price charts visually depict the trend and magnitude of price fluctuations, histograms offer a different perspective by showcasing the frequency and distribution of these movements. This additional information is vital because it reveals volatility patterns that may not be readily apparent from a single price chart. The histograms provide a comprehensive overview of price fluctuations, allowing traders and data analysts to identify key aspects such as periods of high volatility, frequency of extreme price swings, and clustering of price movements. By examining the histograms, the significance of price fluctuations in relation to the stock's current price level becomes apparent. For instance, a drawdown of $20 from $100 to $80 may seem significant when viewed in isolation on a price chart, representing a 20% decline. However, when considering a long-term perspective, such as after 5 or 10 years when the price has reached $500, that same $20 decrease appears relatively minor, representing only a 4% decline compared to the latest data on the chart. This insight helps prevent traders and data analysts from overlooking key aspects of the chart and considering the overall context when analyzing price fluctuations. In the development of the decision tree model, the analysis of histograms prompted the consideration of using more training data and more retraining (to not miss short price fluctuations to the downside) to capture all periods of negative performance (red years) that may have been initially overlooked when relying solely on the price chart. This comprehensive analysis contributed to the model's accuracy and robustness by ensuring a more complete representation of the stock's performance over time.

*AAPL drawdown on price chart zoomed out (Notice how it looks less significant than 2022 drawdown even though it was larger percentage wise.)*





The pie chart, another visualization used in the analysis, provided a distinct representation of a stock's performance over several years that may not be readily extracted from the price chart alone. While price charts emphasize the strongest trends, the pie chart offers an equal-weighted percentage comparison of green (positive) and red (negative) years. This representation becomes particularly valuable when examining stocks such as TSLA, where a significant portion of the gains occurred between 2020 and late 2021. As a result, these years dominate the space on the price chart, potentially overshadowing the performance of previous years and obscuring any poor performance. However, the pie chart treats each year equally in terms of weight percentage, ensuring a balanced representation of green and red years. This allowed for a comprehensive evaluation of a stock's long-term performance, prompting the consideration of sufficient data encompassing both short-term and long-term trends. The analysis of pie charts played a crucial role in aiding the development of the decision tree model by providing more years in the training data for a balanced representation of a stock's performance over several years. This allowed the decision tree model to capture the overall performance of a stock, regardless of specific periods of exceptional gains or losses. By incorporating the insights from the pie charts, the decision tree model could consider the long-term performance of stocks and avoid biases towards certain periods or trends. This comprehensive understanding enhanced the accuracy and robustness of the model's predictions, enabling more informed decision-making in the financial market.

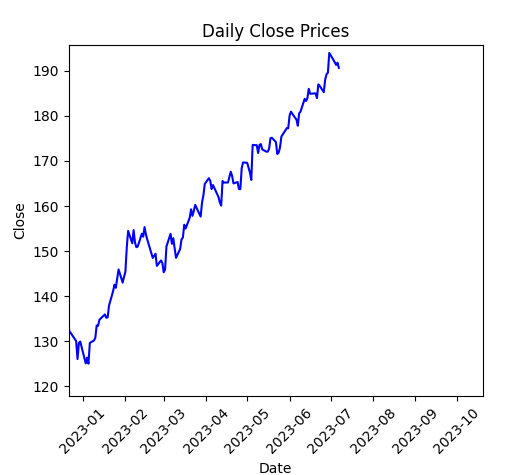
In summary, the statistical analysis and visualizations provided a comprehensive overview of the data, enabling the identification of key patterns and characteristics. The exploration of blue-chip stock trends and analysis of drawdown durations informed the decision tree development process. By leveraging this knowledge, the model was designed to adapt to extended uptrends and respond promptly to changing market conditions. Also, the visualizations of the price chart, histogram, and pie chart further enriched the analysis process. The price charts revealed sustained long-term growth trends, highlighting the compounding effect of price increases over time. The histograms provided insights into price volatility, capturing the frequency and distribution of price fluctuations. Meanwhile, the pie charts offered a balanced representation of a stock's performance, considering both positive (green) and negative (red) years equally. Together, these visualizations enhanced the understanding of market trends, volatility, and overall stock performance. By incorporating insights from these analyses, the decision tree model was refined to consider short-term and long-term trends, resulting in improved accuracy and adaptability.

## Accuracy Analysis

The metric used to assess the predictive model was accuracy, which was represented as a percentage. Accuracy was calculated by dividing the number of winning predictions by the total number of predictions made. This metric provided an evaluation of how well the model performed in correctly predicting whether a selected stock would close green or red on the next trading day. By comparing the number of correct predictions to the total predictions made, the accuracy metric offered a quantitative measure of the model's ability to forecast the direction of stock prices. A higher accuracy percentage indicated a stronger predictive performance (but not too high as that can indicate overfitting), while a lower percentage indicated the need for further refinement and improvement. The accuracy metric served as a reliable benchmark to gauge the effectiveness and reliability of the predictive model in generating accurate trading signals for decision-making in the financial market.

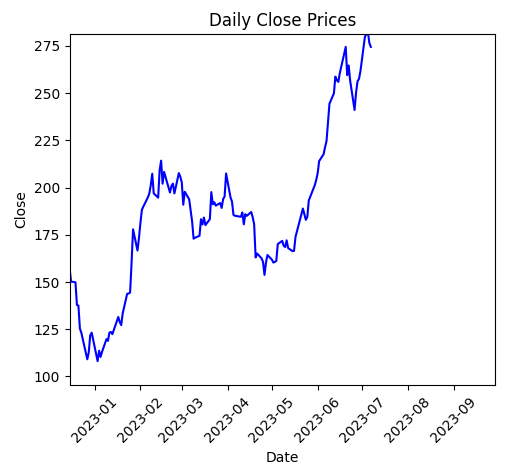
The non-descriptive method employed in this project exhibited a satisfactory level of accuracy. Most stocks achieved an accuracy percentage of 50% or higher, which aligned with the project's expectations. The intention of this model was not to achieve the highest accuracy percentage but to complement the judgment and decision-making of fundamental and discretionary traders, leveraging their expertise to further enhance the accuracy of trading decisions. Having a moderate accuracy percentage for a quantitative model is desirable, as it reduces the risk of overfitting and increases the likelihood of the model performing effectively over longer periods. Notably, the stock AAPL demonstrated the highest accuracy percentage of 61%. This outcome can be attributed to the unique qualities of AAPL, such as its lower volatility compared to other stocks and more predictable price movements characterized by relatively steady growth. These findings suggest that the decision tree model may be particularly effective with stocks exhibiting stable and predictable linear growth patterns rather than those with high volatility.

*AAPL chart during testing data showing linear growth.*

**

As an example, demonstrating the non-descriptive method, the stock TSLA showcased an accuracy percentage of 50%. This relatively moderate accuracy can be attributed to the inherent volatility of TSLA as a stock, which often experiences significant price fluctuations. The statistical analysis highlighted the nature of TSLA's volatility, with a notable drawdown of -75%. This drawdown exemplifies the unpredictable and volatile nature of TSLA's price movements, which can pose challenges for accurate predictions. The 50% accuracy indicates that the non-descriptive method was able to correctly predict the direction of TSLA's closing prices in half of the cases, considering the dynamic and unpredictable nature of the stock. Despite the lower accuracy compared to more stable stocks, the non-descriptive method still offers valuable insights and potential trading signals when used in conjunction with fundamental and discretionary analysis.

*TSLA chart during testing data showing volatile unpredictable growth.*

**

## Application Testing

The application was tested by executing the statistical analysis, predictive model, and visualizations for each stock in the list, employing a brute force method to ensure accuracy and reliability. Testing was conducted during both market open and closed hours to verify the functionality and consistency of data retrieval from the Yahoo Finance API. When testing the application after its development phase, it was observed that the statistical analysis did not include the current trading day's prices in its calculations. This discrepancy was traced back to the "end" variable (endpoint or cutoff date for fetching financial data from the Yahoo Finance API.), which had not been updated since the development phase to include the latest data. To address this, an optimization was implemented by removing the "end" variable altogether and ensuring that the application always displayed the most up-to-date data. These testing results were crucial in identifying and rectifying this issue, leading to an improved version of the application that provided real-time, accurate analysis for traders.

## Application Files

\final\_capstone

\account\_registration.py

\config.ini

\main.py

\requirements.txt

\C964\_task\_2\_written.docx (written portion of project which contains instruction, etc.)

The submission package consists of a compressed zip file named "final\_capstone" containing the necessary files for the project. Upon extracting the zip file, the folder structure will be organized as follows: The root folder named "final\_capstone" contains several files, including "account\_registration.py" which holds the Python code for user account registration, "config.ini" serving as the configuration file, and "main.py" which contains the main Python script for the project and what will be used to launch the program. Additionally, there is a "requirements.txt" file that lists the required Python packages for the project. The "C964\_task\_2\_written.docx" file will be included as the written portion of this project containing documentation and instructions for how to run the program.

## User Guide

1. Install Python: Ensure that Python is installed on your system. You can download the latest version of Python from the official Python website (https://www.python.org) and follow the installation instructions for your operating system.

2. Install Required Libraries: Open the command prompt (CMD) and navigate to the directory where the requirements.txt file is located. Copy the file location including the filename and run the command pip install -r (location of requirements.txt), for example pip install -r Z:\final\_capstone\requirements.txt. Now you have installed the necessary Python libraries listed in the requirements.txt file. This will ensure that all dependencies are installed.

3. Configure the config\_file Location: Open the "main.py" file using a text editor and go to line 417. Modify the value of the config\_file variable to specify the directory path where the "config.ini" file is located on your machine. Save the changes.

4. Run the Application: Open the command prompt (CMD) and navigate to the directory where the "main.py" file is located. Copy the location of "main.py" (including the file name) and paste it into the CMD prompt. Press Enter to execute the command and start the application.

5. Login and User Interface: The program will be live in the command prompt. Follow the instructions displayed in the user interface. For the initial login question, enter the following credentials:

Username: username

Password: test

6. Stock Selection: Follow the instructions provided by the user interface to enter the stock of your choice for analysis.

7. Analysis Selection: Follow the instructions provided by the user interface to select the type of analysis you want to perform.

8. Data Analysis and Restart: After analyzing the data, the program will redirect you to step 6 to restart the process for a different stock. Repeat steps 6 and 7 as desired.

**Notes:**

-An internet connection is required for this program to work.

-There is no need to modify or configure the "config.ini" and "account\_registration.py" files. These files are already set up for use and provided for educational purposes to demonstrate username and password storage and account generation. In a real situation, unauthorized users would not have access to the "account\_registration.py" file. Also, please note that only one account can be stored in "config.ini", and the program will not work if the file is empty.

## Summation of Learning Experience

My previous academic and professional experiences have significantly prepared me for this project. During my computer science degree program, I gained extensive experience in backend development, data analysis, data structure and algorithms, and mathematical concepts that are crucial for working with financial data and machine learning algorithms. The rigorous coursework and hands-on projects honed my skills in handling complex datasets, implementing algorithms, and performing statistical analysis. This foundation allowed me to grasp the technical aspects of this project efficiently.In addition, my prior experience in developing live trading algorithms and back testing systems provided me with a deeper understanding of financial markets and trading strategies. This familiarity with the intricacies of trading and financial terminology proved invaluable when working with the financial data and designing the predictive model. Understanding the nuances of the market dynamics and being able to interpret the data effectively enabled me to make informed decisions throughout the project.

To complete this project, I had a solid foundation in statistical analysis, data visualization using libraries like Matplotlib, and working with financial data from previous projects and experiences. However, one area that required additional learning was the implementation of the predictive model. While I had a high-level understanding of machine learning concepts from previous coursework, applying them specifically to financial data was a new challenge. To bridge this gap, I delved into various machine learning libraries such as scikit-learn, TensorFlow, and PyTorch. Through experimentation and exploration, I gained insights into how to preprocess and feed training and testing data into the algorithms. This hands-on experience helped me understand the nuances of working with financial data and enabled me to achieve success in utilizing the decision tree algorithm with scikit-learn. The process of acquiring new knowledge and skills in predictive modeling allowed me to overcome the learning curve and successfully incorporate machine learning techniques into the project.

This experience has greatly contributed to my concept of lifelong learning by reinforcing the importance of continually acquiring new knowledge and skills to adapt to evolving challenges and industries. Throughout the project, I encountered new concepts, technologies, and methodologies that required me to step out of my comfort zone and expand my expertise. The need to learn and explore machine learning libraries, understand the intricacies of financial data analysis, and grasp trading strategies deepened my appreciation for the value of continuous learning. It highlighted the dynamic nature of the field and the necessity to stay updated with the latest developments since the stock market is always changing and is not an exact science. This project reinforced my belief that learning is a continuous journey, and embracing lifelong learning is essential to personal growth and professional success. By recognizing the need for ongoing education, I am committed to pursuing further knowledge and skills in order to adapt to future challenges, explore new opportunities, and make meaningful contributions in the ever-changing landscape of technology and finance.

# Sources

No sources were used in this project.