**Chapter 1: Introduction**

* 1. **Background and Context**

In a rapidly changing world where almost everyone is affected by rising inflation, people are increasingly becoming aware of the importance of investing, and their own financial goals. The purpose of this project is to create a data-driven solution for such individuals to help them select equity portfolios in the Indian market. In contrast to conventional investment methods, this strategy will analyse market benchmark(s) and their constituents and find outperforming assets within those indices. The end product will be a code repository and a web application that will display portfolios suggested by the strategy developed. Using that, a potential investor can select a portfolio that aligns with their financial goals and make a well-informed investment decision. With the help of historical data, this project intends to offer a quantitative investment strategy that could be customized to suit the specific demands and risk tolerances of individual investors.

Traditional algorithmic trading strategies often rely on simplistic buy/sell signals. In contrast, the primary objective here is to utilize historical data, quantitative analysis, and self-defined metrics to construct investment portfolios that offer varying risk-return profiles. This approach is critical in today's investment landscape, where traditional methods may no longer suffice. The project proposes a multi-faceted methodology that goes beyond standard market analysis. It includes an extensive backtesting phase, spanning various market cycles, to validate the robustness of the strategies developed. Additionally, the strategies developed in this project are not limited to selecting optimal portfolios but also give a date on which the portfolio should be rebalanced, i.e., which existing stocks must be sold which must be bought, and on which date.

* 1. **Motivation and Significance**

Modern investors want more than simply financial rewards; they also want empowerment, ownership, and the capacity to make data-driven decisions. In the exponentially growing Indian financial market, there are many investment options, which can be overwhelming for individual investors. This project intends to help simplify investing for them with a structured approach. I am also motivated by the idea that historical data can greatly assist investment decisions. My project will use strategies that consider various statistical factors to make investment choices more organized. The goal is to provide individuals with the tools they need to make informed decisions in the dynamic Indian financial markets. Throughout the project, I will try to keep my code as modular as possible, making it easier to adapt the same to other markets and customizable to tweak the strategy in the future.

* 1. **Problem Statement**

#### To create a novel approach to portfolio selection and time-based rebalancing to outperform market benchmark indices.

* 1. **Objectives**

#### Develop a historical data-driven investment strategy that focuses on analysing market indices and identifying a good combination of assets within those indices.

#### Create a user-friendly app/interface that allows individual investors to choose their portfolios based on the strategy’s recommendations.

#### Evaluate the performance of the strategy through backtesting and historical data analysis, providing investors with insights into potential portfolio performance.

#### Scope and Limitations

The scope of this project includes data collection, data wrangling, data analysis, formation of strategies, backtesting of strategies, analysis of results, and creation of a web application for the same. The scope does not include consideration of market microstructure to a complete extent, although some aspects of market microstructure have been considered. These include the consideration of current market prices of the stocks for weight allocation of the total investment amount and rebalancing costs that include brokerages, taxes, etc. There are a few more identified limitations of the project that are beyond the scope of the project such as making the investment strategies survivorship-bias free. This will be discussed in detail in a later section of this report.

**Chapter 2: Review of Related Works**

#### Chapter 3: Methodology

#### 3.1. Strategy Development and Overview of Variations

#### The foremost step in developing strategies is to choose a base universe, i.e., a common pool of stocks that act as the foundation for the strategies to be built upon. The selection of the base universe is an essential step in strategy development because it can highly influence the performance of the strategies. The selection of the base universe for this project was made by focusing on getting a large pool of diverse stocks that include large-cap, mid-cap, small-cap, and even micro-cap companies. The idea was to create a strategy that capitalised on the extremely high returns of small-cap and micro-cap companies while giving a safety net by including the more stable and less risky large-cap and mid-cap companies.

#### The most commonly referred benchmark indices for the Indian market include SENSEX and NIFTY 50, which only include 30 and 50 companies respectively, that too from a very small segment of companies with an extremely large market capital. Apart from the fact that both of these indices don’t include companies with diverse market capital, they also include a very small number of companies and building a strategy on the analysis of such few companies wouldn’t have yielded the kind of results that the project aimed for.

#### Hence, my focus shifted to considering the constituents of indices like NIFTY 500 or NIFTY TOTAL MARKET which include 500 and 750 companies respectively. NIFTY 500 includes the top 500 listed companies in India by market capitalization and it was a very good base to start with given that it represents 96.1% of the total free-float market capitalization of India and 96.5% of the total turnover of the National Stock Exchange (NSE). This would include not just large-cap and mid-cap companies but also some small-cap companies. However, I wanted to include companies that are emerging and even smaller by market capitalization and do not fall in the top 500 which is why I chose the NIFTY TOTAL MARKET index which included the same 500 companies from the NIFTY 500 but also include 250 other companies from the NIFTY MICROCAP 250 index. Having 750 companies gave both, a huge and diverse market-cap pool for the base universe for the strategy.

#### The strategy can be broken down into three main components. This section will get into the details of all of these three components and the variations between them and then discuss how they are used together in conjunction for successful outcomes.

#### 3.1.1. Stock Selection Methods:

There are a total of 8 stock selection methods defined for this project, coded from S1 to S8. Odd methods, i.e., S1, S3, S5, and S7 have a holding/rebalancing period of 1 quarter (3 months), whereas the even methods, S2, S4, S6, and S8, have a holding/rebalancing period of 1 month. Except for the differences in the holding periods of the methods, the consecutive methods have no other difference. For example, S1 and S2 are the same except for the difference in their holding period, and so on till S7 and S8. However, a different holding period would lead to differences in the way the stock prices are being analysed.

*Step 1:* The foremost step in all of the methods is to generate periods of the duration of the defined holding period for the previous 3 years of the buying date. This means for methods with a one-month holding period, all the one-month periods will be generated for the last three years of the buying date that start every 10, 20, and 30 days, i.e., for a buying date of 01/01/2024 the code would create all the one-month periods starting every 1st, 11th, and 21st of every month from January 2021 to December 2023. This would result in about 36 one-month periods in a year, and a total of 108 different one-month periods in three years. Similarly, for methods with a three-month holding period, all the three-month periods will be generated for the last three years of the buying date that start every 30 days, i.e., for a buying date of 01/01/2024, the code would create all 3 month periods starting every 1st of every month from January 2021 to December 2023. This would result in about 12 three-month periods in a year, and a total of 36 different three-month periods in three years.

*Step 2:* This step includes a combination of various filters applied to all the.

There are 6 different filters in total:

* Filter 1: select stocks with daily average returns of all the quarterly/monthly periods > 0.
* Filter 2: select stocks with daily average returns of all the quarterly/monthly periods in the last ‘X’ year > 0.
* Filter 3: select stocks with a maximum limit of either 10/15 (depending on the strategy) on total non-positive returns count for all the quarterly/monthly periods.
* Filter 4: select stocks with maximum no non-positive returns count for all the quarterly/monthly periods in the last ‘X’ year.
* Filter 5: select 50 stocks that have the least non-positive returns count for all the quarterly/monthly periods in the last ‘X’ year.
* Filter 6: select 30 stocks that have the highest daily average returns in the last ‘X’ year.

NOTE: The use of ‘X’ in some of these filters is related to the third component of the strategy which is ‘Analysis Time Frames’. For now, you can consider them to be a number less than 3, which is the total years of data considered before the buying date.

|  |  |  |  |
| --- | --- | --- | --- |
| Stock Selection Method | Holding/Rebalancing Period | Step 1 - Period Generation | Step 2 - Filters Applied |
| S1 | 1 quarter (3 months) | - generate 3-month periods  - starting every 1st of a month in the last 3 years. | Filters 1-4 (Limit of 15 on Filter 3) |
| S2 | 1 month | - generate 1-month periods  - starting every 1st, 11th, and 21st of a month in the last 3 years. | Filters 1-4 (Limit of 15 on Filter 3) |
| S3 | 1 quarter (3 months) | - generate 3-month periods  - starting every 1st of a month in the last 3 years. | Filters 1-4 (Limit of 10 on Filter 3) |
| S4 | 1 month | - generate 1-month periods  - starting every 1st, 11th, and 21st of a month in the last 3 years. | Filters 1-4 (Limit of 10 on Filter 3) |
| S5 | 1 quarter (3 months) | - generate 3-month periods  - starting every 1st of a month in the last 3 years. | Filters 1, 2, and 4 |
| S6 | 1 month | - generate 1-month periods  - starting every 1st, 11th, and 21st of a month in the last 3 years. | Filters 1, 2, and 4 |
| S7 | 1 quarter (3 months) | - generate 3-month periods  - starting every 1st of a month in the last 3 years. | Filters 5 and 6 |
| S8 | 1 month | - generate 1-month periods  - starting every 1st, 11th, and 21st of a month in the last 3 years. | Filters 5 and 6 |

*Table 3.1: Stock Selection Methods*

*Methods S1 and S2:* They have the first 4 filters in a sequential order with a limit of 15 on the third filter. These were the first two methods that I started the stock selection process with.

*Methods S3 and S4:* They have the first 4 filters in a sequential order with a limit of 10 on the third filter. These variations were created after noticing an impact on the results making the requirement of filter 3 more selective. Keeping the limit to 15 allowed many companies to get selected that had an extremely high variance, and could lead to losses.

*Methods S5 and S6:* They have the filter 1, 2, and 4, in the same order but do not have filter 3. After noticing the impact of making the process more selective in S3 and S4, it seemed like the logical move to remove the filter 3 requirement as it was focusing on the full 3-year performance and not the recent performance which was more important considering that the strategies were either for a monthly rebalance or a quarterly rebalance. Filter 3 with a limit of 10 selected very few companies most of the time (around 3 or 4) and that was not an adequate number to move forward with the weight allocation.

*Method S7 and S8:* After looking at the results from the previous methods, I got more and more clarity and insight on what filters are beneficial and how can the other filters be tweaked to get a good filter of stocks. S7 and S8 have only 2 filters which are filters 5 and 6 in that same order. Here, the focus had completely shifted to only looking at the more recent data and not the entire 3-year data which improved the performance a lot and some of the best outcomes of the project were through S7 and S8 which will be discussed later in the backtesting results and analysis section.

*Step 3:* After the stocks are filtered in step 2 through various possible methods, they go ahead for weight allocation.

**3.1.2. Weight Allocation Methods:**

There are a total of 11 weight allocation methods defined for this project, coded from A1 to A11. A1 is the most basic weight allocation method, i.e., to equally weigh all the stocks. Methods A2, A3, A6, and A7 are based on the mean-variance analysis given by the Markowitz Portfolio Theory where A2 and A3 aim to maximize returns with slightly modified constraints, and A6 and A7 aim to minimize the risk by minimizing variance with slightly modified constraints. Method A4 and A5 are tangency portfolios that aim to maximize the risk-adjusted returns, i.e., the Sharpe ratio, with some variations in the constraints. Methods A8 to A11 are some of the more complicated and self-devised methods that aim to factor in the weightage given to stock for different objectives (returns, risk, and risk-adjusted returns) and try to pick stocks that have a significant weightage in all of those cases. Although the fundamental idea behind the methods A8 to A11 are the same, there are some differences in the constraints that will be discussed in this section. The weight allocation strategies are also based on the historical stock price data of a specific period ‘Y’, which is again related to the third component of the strategy which is ‘Analysis Time Frames’.

*Method A1:* Weight all stock selected equally

*Method A2:* If the number of stocks selected in the first component of the strategy is more than 8 then this method runs an optimization function to maximize the returns with the constraints of not letting the annual standard deviation of the portfolio exceeding 40% and while making sure that no single stock gets a weightage of more than 25% of the total funds, and then picks the 8 highest weighted stocks from the obtained portfolio and run the optimization process again on those 8 stocks.

*Method A3:* Method A3 is the same as Method A2 with the only difference being more leniency on one of the constraints, i.e., no single stock should get a weightage of more than 40%, instead of 25% in A2.

*Method A4:* If the number of stocks selected in the first component of the strategy is more than 8 then this method runs an optimization function to maximize the Sharpe ratio while making sure that no single stock gets a weightage of more than 25% of the total funds, and then picks the 8 highest weighted stocks from the obtained portfolio and run the optimization process again on those 8 stocks.

*Method A5:* Method A5 is the same as Method A4 with the only difference being more leniency on the constraint, i.e., no single stock should get a weightage of more than 40%, instead of 25% in A4.

*Method A6:* If the number of stocks selected in the first component of the strategy is more than 8 then this method runs an optimization function to minimize the variance with the constraints of having a minimum annual return of the portfolio of 30% while making sure that no single stock gets a weightage of more than 25% of the total funds, and then picks the 8 highest weighted stocks from the obtained portfolio and run the optimization process again on those 8 stocks.

*Method A7:* Method A7 is the same as Method A6 with the only difference being more leniency on one of the constraints, i.e., no single stock should get a weightage of more than 40%, instead of 25% in A6.

*Method A8:* Method A8 applies a minimized variance optimization on the selection stocks with no single stock being allocated more than 25% of the total fund and a constraint on the annual returns being a minimum of 20%. At the same time, it applies a maximized returns optimization on the same stocks with no single stock being allocated more than 25% of the total fund and a constraint on the annual standard deviation being a maximum of 20%. From the two portfolios obtained, it picks the 4 highest weighted stocks from each to get a total of 8 stocks (4 giving a high return, and 4 giving a low risk). On these 8 stocks, the tangency portfolio maximises the Sharpe ratio, i.e., the highest risk-adjusted returns with a constraint on no single stock being more than 25% of the total fund.

*Method A9:* Method A9 is the same as Method A8 with the only difference being more leniency on one of the constraints, i.e., no single stock should get a weightage of more than 40%, instead of 25% everywhere in A8.

*Method A10:* Method A10 applies a minimized variance optimization on the selection stocks with no single stock being allocated more than 40% of the total fund and a constraint on the annual returns being a minimum of 30%. At the same time, it applies a maximized returns optimization on the same stocks with no single stock being allocated more than 40% of the total fund and a constraint on the annual standard deviation being a maximum of 30%, and simultaneously it also applies a maximized Sharpe ratio optimization with the same constraint of no single stock being allocated more than 40% of the funds. From the three portfolios obtained, it picks all the stocks with a minimum of 10% weightage in all of them to get the best of all optimizations. After which it runs a maximized returns optimization again on all of these stocks with constraints of no single stock being allocated more than 20% of the total fund and the maximum limit on the annual standard deviation being 30%.

*Method A11:* Method A11 is the same as Method A10 with the only difference being more leniency on one of the constraints, i.e., no single stock should get a weightage of more than 25%, instead of 40% everywhere in A10.

NOTE: The calculations and optimization processes involving the Sharpe ratio use the Indian Government’s bond yield historical data for deducing a risk-free rate of return which is used in the calculation of the Sharpe ratio.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Weight Allocation Method | Summary | | | |
| A1 | - equally weight all selected stocks | | | |
| A2 | - objective: maximize returns  - constraints:  1. no stock > 25%  2. annual standard deviation < 40%  - if selected stocks > 8, then choose top 8 by weight and repeat the optimization | | | |
| A3 | - same as A2 with the only change in one of the constraints being no stock > 40% instead of 25% | | | |
| A4 | - objective: maximize Sharpe ratio  - constraint: no stock > 25%  - if selected stocks > 8, then choose top 8 by weight and repeat the optimization | | | |
| A5 | - same as A4 with the only change in the constraint being no stock > 40% instead of 25% | | | |
| A6 | - objective: minimize variance  - constraints: no stock > 25%, annual returns > 30%  - if selected stocks > 8, then choose top 8 by weight and repeat the optimization | | | |
| A7 | - same as A6 with the only change in one of the constraints being no stock > 40% instead of 25% | | | |
| A8 | - objective 1: minimize variance  - constraints: no stock > 25%, annual returns > 20% | | - objective 2: maximize returns  - constraints: no stock > 25%, annual standard deviation < 20% | |
| - pick top 4 stocks by weight from of the portfolios obtained to get 8 stocks and apply the optimization for objective 3.  - objective 3: maximize Sharpe ratio  - constraint: no stock > 25% | | | |
| A9 | - same as A8 with the only change in one of the constraints being no stock > 40% instead of 25% for all 3 objectives. | | | |
| A10 | - objective 1: minimize variance  - constraints:  1. no stock > 40%  2. annual returns > 30% | - objective 2: maximize returns  - constraints:  1. no stock > 40%  2. annual standard deviation < 30% | | - objective 3: maximize Sharpe ratio  - constraint: no stock > 40% |
| - pick all the stocks with a minimum weightage of 10% in all of the three portfolios and optimize for objective 4.  - objective 4: maximize returns  - constraints:  1. no stock > 20%  2. annual standard deviation < 30% | | | |
| A11 | - same as A10 with the only change in one of the constraints being no stock > 25% instead of 40% for the first 3 objectives, constraints for objective 4 remains the same. | | | |

*Table 3.2. Weight Allocation Methods*

**3.1.3. Analysis Time Frames:**

Analysis Time Frames, within the context of stock selection and portfolio management, relate to the precise time periods that are used to evaluate historical data in order to guide decisions for buying or selling the securities. To make accurate data-backed judgements about the selection and weight allocation or securities for a portfolio, choosing the correct time span to analyse its performance and risk is highly important. Given the time varying nature of financial markets, selecting the appropriate time range can have a big impact on how investment strategies perform.

*Performance Evaluation:* For the same asset, different time periods may reveal different performance trends. While long-term analysis may show stability and development potential, short-term analysis may emphasise volatility and the possibility for rapid gains.

*Risk Assessment:* Over a variety of time periods, an asset's associated risk can change dramatically. While long-term analysis may show resilience and stability, short-term analysis may reveal more volatility.

Since my project involves two kinds of strategies, rebalancing every month and rebalancing every quarter, it does not require a huge analysis period. While some traditional methods suggest that the analysis period should be at least 10 times longer than the holding period of the portfolio, some suggest 5 times, and some suggest 3 times. Which is why I decided to experiment with different analysis time frames. It is important to note that as described above, the stock selection method is a completely independent process to the weight allocation method except for the fact that which stocks to allocate weight to depends on which stock got selected. Apart from that, the time frames for both of these processes need not necessarily be the same and can be different. These are the periods referred to as ‘X’ and ‘Y’ in the above sections. As a base case, I started with an analysis time frame of 1 year each (for stock selection and weight allocation), and then I tried both, increasing the period, and decreasing the period to compare

the results. The results, which will be discussed in much more detail in later sections, were much better for smaller periods, which was expected since the holding periods of both types of strategies are monthly and quarterly, which aren’t huge periods. I also experimented with taking larger time frames for the stock selection methods and smaller ones for the weight allocation methods and vice-versa. Considering the results, I went ahead with decreasing the time frames more and eventually reached 10 different combinations of periods for stock selection and weight allocation.

|  |  |  |
| --- | --- | --- |
| Analysis Time Frame Combination | Historical Time Period Considered for  Stock Selection Methods | Historical Time Period Considered for  Weight Allocation Methods |
| T1 | 1 year | 1 year |
| T2 | 1 year | 2 years |
| T3 | 2 years | 1 year |
| T4 | 2 years | 2 years |
| T5 | 6 months (0.5 year) | 6 months (0.5 year) |
| T6 | 1 year | 6 months (0.5 year) |
| T7 | 6 months (0.5 year) | 1 year |
| T8 | 3 months (0.25 year) | 3 months (0.25 year) |
| T9 | 6 months (0.5 year) | 3 months (0.25 year) |
| T10 | 3 months (0.25 year) | 6 months (0.5 year) |

*Table 3.3. Analysis Time Frames*

*Method T1:*

Stock Selection Historical Time Frame(X): 1 year

Weight Allocation Historical Time Frame (Y): 1 year

*Method T2:*

Stock Selection Historical Time Frame(X): 1 year

Weight Allocation Historical Time Frame (Y): 2 years

*Method T3:*

Stock Selection Historical Time Frame(X): 2 years

Weight Allocation Historical Time Frame (Y): 1 year

*Method T4:*

Stock Selection Historical Time Frame(X): 2 years

Weight Allocation Historical Time Frame (Y): 2 years

*Method T5:*

Stock Selection Historical Time Frame(X): 6 months (0.5 years)

Weight Allocation Historical Time Frame (Y): 6 months (0.5 years)

*Method T6:*

Stock Selection Historical Time Frame(X): 1 year

Weight Allocation Historical Time Frame (Y): 6 months (0.5 years)

*Method T7:*

Stock Selection Historical Time Frame(X): 6 months (0.5 years)

Weight Allocation Historical Time Frame (Y): 1 year

*Method T8:*

Stock Selection Historical Time Frame(X): 3 months (0.25 years)

Weight Allocation Historical Time Frame (Y): 3 months (0.25 years)

*Method T9:*

Stock Selection Historical Time Frame(X): 6 months (0.5 years)

Weight Allocation Historical Time Frame (Y): 3 months (0.25 years)

*Method T10:*

Stock Selection Historical Time Frame(X): 3 months (0.25 years)

Weight Allocation Historical Time Frame (Y): 6 months (0.5 years)

**3.2. Strategy Combinations and Evaluation Process**

Having devised 8 stock selection methods, 11 weight allocation methods, and 10 analysis time frame combinations, I thought of trying every combination of all of them to come up with the best possible strategy out of all of them that consider so many different parameters. This gave me a huge total of 880 unique strategies for every combination of S1 to S8, A1 to A11, and T1 to T10.

For easy referencing, I decided to give a unique code to each of the 880 strategies that is a combination of their stock selection method, weight allocation method, and analysis time frame combination, which is given in the table on the next page.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No. | Stock Selection Method | Weight Allocation Method | Analysis Time Frame | Combination Code |
| 1 | S1 | A1 | T1 | S1-A1-T1 |
| 2 | S1 | A1 | T2 | S1-A1-T2 |
| … | … | … | … | … |
| 110 | S1 | A11 | T10 | S1-A11-T10 |
| … | … | … | … | … |
| 880 | S8 | A11 | T10 | S8-A11-T10 |

*Table 3.4. Strategy Combinations*

To evaluate these strategies I decided to backtest all 880 strategies for 4 years from 1st January 2020 to 1st January 2024. Based on the results of the evaluation I had 4 different selection criteria defined to find the best of them. These 4 criteria were:

1. Strategy with the highest 4Y CAGR among the ones with the highest success rate
2. Strategy with the lowest 4Y variance among the ones with the highest success rate
3. Strategy with the highest 4Y CAGR
4. Strategy with the lowest 4Y variance

Another important component of the evaluation process is deciding the backtesting dates. The backtesting dates would vary depending on the strategy’s rebalancing rule/holding period. As described earlier in the ‘Stock Selection Methods’ section, for the strategies with quarterly rebalancing, the backtesting would start with buying the first portfolio on the 1st of January 2020 and rebalancing it on the 1st of every month

**3.3. Tools and technologies used**

For this project a variety of tools and techniques were used. The foundation of the project is laid on Python programming language in addition to many in-built and external libraries and modules.

The external libraries and modules include:

* Pandas: Pandas library is used for data collection, wrangling, and analysis. Pandas helps in creating and managing data in the form of data frames.
* Numpy: Numpy is used for large scientific calculations including calculations on multi-dimensional arrays and matrices along with a huge collection of high-level mathematical functions that can be used. Using Numpy for the project helped with reducing time and space complexity and optimizing the computing processes involved in the project.
* Yfinance: Yfinance is an open-source API by Yahoo Finance that was the primary source for all the data needed for the project. The YFinance API enables Python programmers to access both live and stock data including opening and closing prices and volumes for analysis and strategy creation.
* Matplotlib: Matplotlib is a Python charting toolkit that is used to create static, interactive, and animated visuals. It has features for generating many different plots and charts, including scatter plots, bar charts, line charts, and histograms. Matplotlib played a key role in this project's data and result visualisation, which helped to clarify patterns, trends, and anomalies in the dataset.
* Seaborn: Seaborn is a high-level interface for creating visually appealing and educational statistical visualisations is offered by this Python visualisation toolkit, which is built on top of Matplotlib. Making intricate graphs from data in arrays and data frames is one of its main uses. Seaborn was utilised in this project to produce more complex statistics charts and to improve the plots' aesthetic appeal.
* Scipy: The library Scipy is used for scientific calculations in python. It contains features for optimization, linear algebra, integration, differential equation solvers, etc., which are common in engineering and operations research related projects. In this project, Scipy was used within various weight allocation methods to cater for various objectives such as maximizing returns, maximizing Sharpe ratio, and minimizing risk.
* Python-dateutil: Python-dateutil is an extension to the standard in-built datetime module in Python. It offers a wide range of operations on dates in various formats, which is particularly useful in a project like this where the entire strategy is built on time-series data analysis.
* Streamlit: Streamlit is a framework available in Python that helps create minimalistic and interactive web applications to Python programmers without having to use any frontend web development programming languages like HTML, CSS, or JavaScript. It enables programmers to create interactive apps in Python itself and in this project it was used to build an interface that allows users to explore the strategies and get live recommendations for their portfolios.

The in-built libraries and modules include:

* Datetime: Functions for modifying dates and times are provided by the datetime module. In order to properly handle time-series data, it was utilised in this project to work with dates along with the extension Python-dateutil.
* Sys: The sys module gives access to functions that have a close relationship with the Python interpreter as well as some variables that are utilised or maintained by the interpreter. It was applied to system-specific functions and parameters in this project, like changing the directory of the Python interpreter to access data files in an external directory.
* OS: The OS module gives users access to functionality that depends on their operating system. It was utilised in this project to handle files and directories by interfacing with the operating system along with the sys module.
* Random: For a variety of distributions, the random module applies pseudo-random number generators. It was employed to produce random numbers, which help assign random values when necessary or for use in simulations.
* Math: The mathematical functions specified by the C standard are accessible through the math module. It was used for constants and elementary mathematical operations that need not be defined repeatedly in the code.
* Ast: The ast module facilitates the processing of Python abstract syntax grammar trees by Python applications. It is not frequently utilised in data projects, but when it is, it can be quite helpful for the project's dynamic expression evaluation, like the conversion of strings from a data file to dictionaries.
* JSON: Data encoding and decoding in JSON is made simple by the JSON module. This project utilised it to parse JSON files or data, which is frequently found in web-based data sources. This project also used the JSON library to save backtesting results and read them back for evaluation and analysis.
* Statistics: Functions for computing mathematical statistics of numerical data are available in the statistics module. This project utilises it to do fundamental statistical operations like variance, median, and mean calculations.

­­­

Chapter 4: