**Algorithmic Trading (Equal Weights S&P500)**

1. **Introduction**

This is my first personal project on algorithmic trading.

The strategy for this project is assigning equal weights to all the companies in the S&P500 (a static list).

The hypothesis for this strategy is that smaller-cap stocks may have more room for growth or are more likely to be mispriced, compared to the mega-cap stocks that dominate the S&P 500's market-cap-weighted index.

How this hypothesis may work:

1. Diversification of Risk: Equal-weighting reduces the concentration of risk by spreading it evenly across all 500 companies, limiting the dominance of large-cap stocks.
2. Smaller-Cap Exposure: The strategy increases exposure to mid- and small-cap stocks, potentially leading to higher returns compared to large-cap stocks over certain periods.

The output of this project is a excel sheet recommending the number of shares to buy for each S&P500 company.

**2. Methodology**

* Data processing
  + Loaded a csv file of 505 S&P500 companies

A screenshot of a computer

Description automatically generated

*Figure 1: Static List of S&P500 Companies*

* + Output of APIs

A white text with black numbers

Description automatically generated

*Figure 2: Response of API to get Stock Price of companies*

A close-up of a computer error

Description automatically generated

*Figure 3: Response of API to get Market Capitalisation of companies*

* Looping through all S&P500 companies
  + Created a loop to obtain the relevant data (price and market capitalisation) of all S&P500 companies
  + Created a DataFrame to store the data

\* only looped for first 5 companies to get result

A screenshot of a cellphone

Description automatically generated

*Figure 4: DataFrame consisting of data on S&P500 companies*

* User Interface
  + Using input function of python, users can input their portfolio size
  + Calculated position size by dividing portfolio size with number of companies
  + Calculated number of shares to buy of respective companies by dividing position size with stock price

A screenshot of a graph

Description automatically generated

*Figure 5: Data Frame with updated data on number of shares to buy*

* Saving output into an excel file
  + By using xlsxwriter library in python, an excel sheet with the recommended trades will be created and saved

A screen shot of a chart

Description automatically generated

*Figure 6: Excel Sheet of recommended trades*

**3. Learning Summary**

* Algorithmic Trading Process is broken down generally into these steps:
  + Collecting Data
  + Developing a hypothesis for a strategy
  + Backtesting the strategy\* (did not do for this project)
  + Implement strategy in production\* (did not do for this project)
* **How I can improve this project:**
  + Purchasing an API to get real-time list of S&P500 companies
  + Purchasing an API that allows for batch calls (optimises the speed at which the data is collected)