S&P 500 Stock Recommendation System

Aditya Agrawal, Shresht Venkatraman, Viveka Dhanda, Marija Vukić

Aim

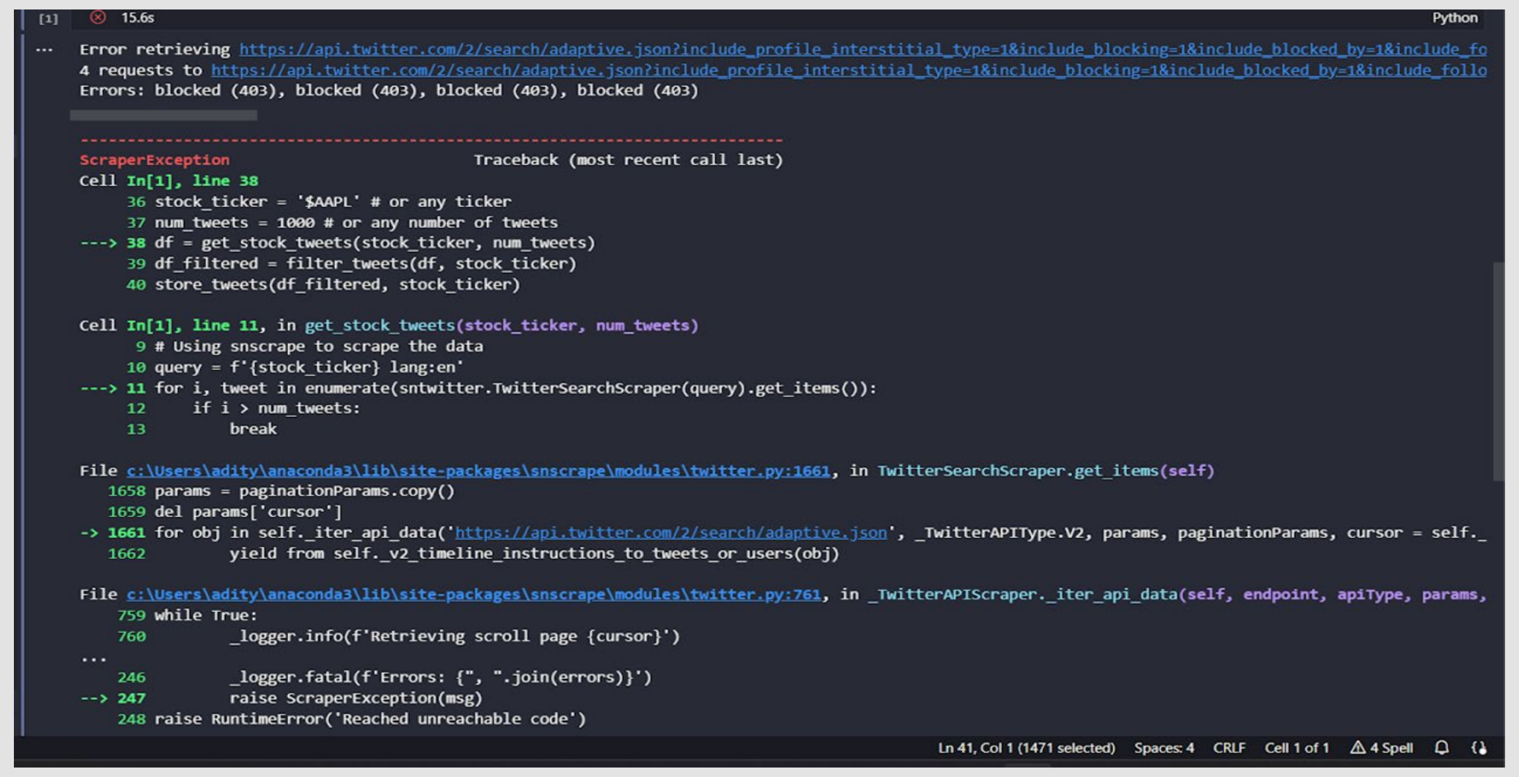
The goal of this project is to enhance the process of stock portfolio allocation by incorporating machine learning, sentiment analysis, and portfolio optimization techniques. Our intention is to predict the trajectory of stock prices while delivering robust portfolio recommendations that consider the larger sentiment environment associated with each stock.

Methodology

Our methodology comprises three core stages. Firstly, an **LSTM** model forecasts stock price movements using historical data. Secondly, **sentiment analysis** is conducted on the same stock from news data. These two outputs, which encapsulate both quantitative and qualitative insights, are finally integrated into an **Efficient Frontier model**, to determine an optimally balanced portfolio composition. By incorporating LSTM forecasting, sentiment analysis, and portfolio optimization, our project hopes to provide authentic stock recommendations while advancing our understanding of data science in the realm of stocks.

Challenges

1. **Efficient Market Hypothesis:** This theory suggests that all current stock-prices already reflect all available information about a given Stock. Therefore future price-movements move in a '*Random-Walk*', i.e an unpredictable manner. This makes traditional predictive ML models difficult to develop.
2. **Non Stationarity**: Movement of Stock Prices across time are *Non-Stationary*, i.e. the distribution (mean and variance) of our data set changes over time. Difficult to develop a prediction model that is accurate and usable across extended time periods due to differing distributions of our training and testing data sets
3. **Predicting External Events:** Pandemics, wars, election results and their effects on individual stocks and their effect on prices of a given stock cannot be easily predicted.



```
[1]  ⊗ 15.6s                                                                    Python
...  Error retrieving https://api.twitter.com/2/search/adaptive.json?include_profile_interstitial_type=1&include_blocking=1&include_blocked_by=1&include_fo
     4 requests to https://api.twitter.com/2/search/adaptive.json?include_profile_interstitial_type=1&include_blocking=1&include_blocked_by=1&include_follo
     Errors: blocked (403), blocked (403), blocked (403), blocked (403)

     ---------------------------------------------------------------------------
     ScraperException                          Traceback (most recent call last)
     Cell In[1], line 38
          36 stock_ticker = '$AAPL' # or any ticker
          37 num_tweets = 1000 # or any number of tweets
     ---> 38 df = get_stock_tweets(stock_ticker, num_tweets)
          39 df_filtered = filter_tweets(df, stock_ticker)
          40 store_tweets(df_filtered, stock_ticker)

     Cell In[1], line 11, in get_stock_tweets(stock_ticker, num_tweets)
           9 # Using snscrape to scrape the data
          10 query = f'{stock_ticker} lang:en'
     ---> 11 for i, tweet in enumerate(sntwitter.TwitterSearchScraper(query).get_items()):
          12     if i > num_tweets:
          13         break

     File c:\Users\adity\anaconda3\lib\site-packages\snscrape\modules\twitter.py:1661, in TwitterSearchScraper.get_items(self)
        1658 params = paginationParams.copy()
        1659 del params['cursor']
     -> 1661 for obj in self._iter_api_data('https://api.twitter.com/2/search/adaptive.json', _TwitterAPIType.V2, params, paginationParams, cursor = self._
        1662         yield from self._v2_timeline_instructions_to_tweets_or_users(obj)

     File c:\Users\adity\anaconda3\lib\site-packages\snscrape\modules\twitter.py:761, in _TwitterAPIScraper._iter_api_data(self, endpoint, apiType, params,
         759 while True:
         760         _logger.info(f'Retrieving scroll page {cursor}')
     ...
         246         _logger.fatal(f'Errors: {", ".join(errors)}')
     --> 247         raise ScraperException(msg)
         248 raise RuntimeError('Reached unreachable code')

                                   Ln 41, Col 1 (1471 selected)  Spaces: 4  CRLF  Cell 1 of 1  ⚠ 4 Spell
```

Model 1: Historical Data Analysis

LSTM: LSTMs (Long Short-Term Memory) are a type of Recurrent Neural Network (RNN) that can effectively capture sequential patterns in data. In the context of stock prediction, we attempt to use historical stock price data to train an LSTM model to learn patterns and trends in price movements.

Why we selected it:

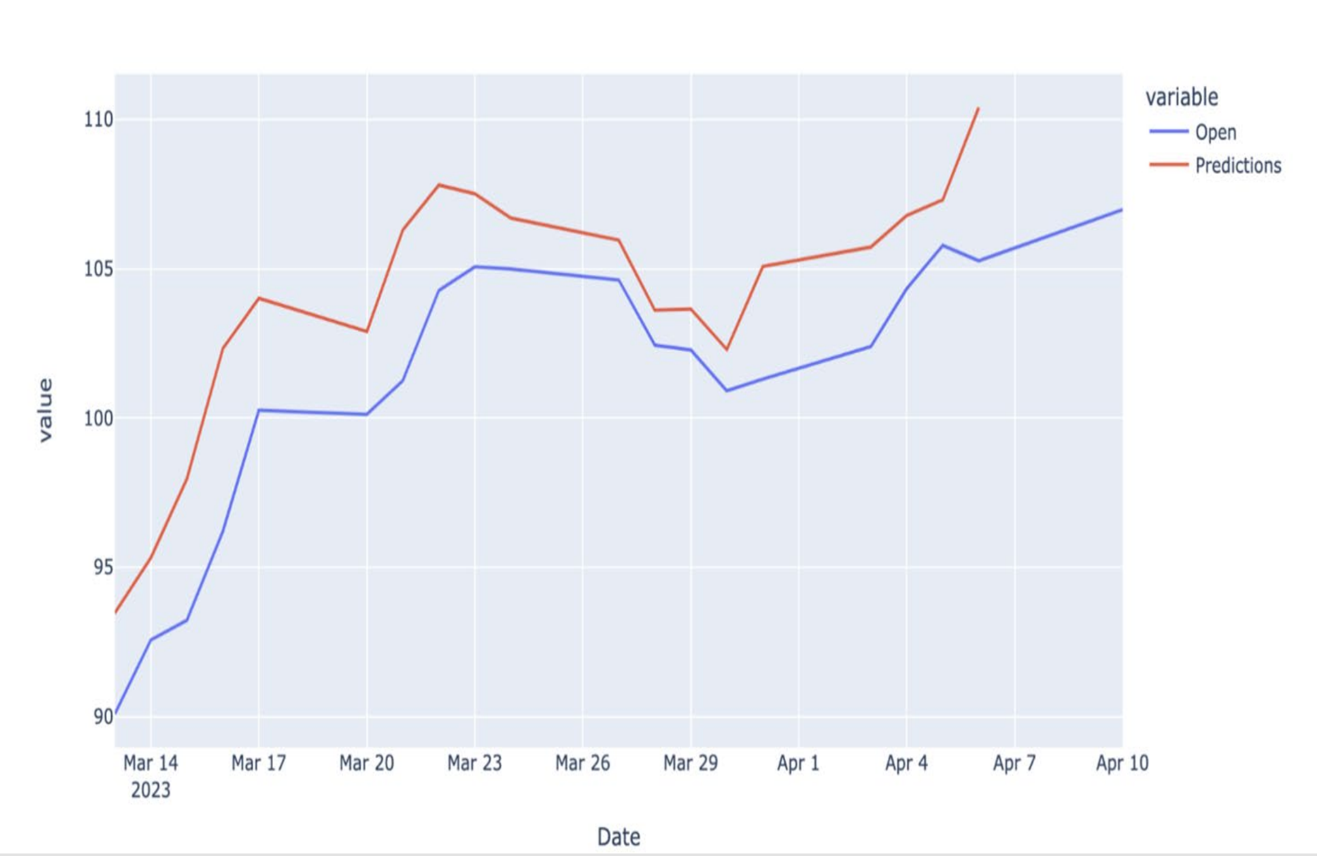
In order to overcome the *'Non-Stationarity Problem'* and incorporate long-term historical stock price data into our prediction model. LSTMs are useful in that they can enhance the modelling of long-term dependencies by incorporating memory cells and gating mechanisms, LSTMs can selectively 'remember' or 'forget' information, enabling them to capture and utilize long-range dependencies in the sequential price data.

Pros: Decent Accuracy

Cons:. Doesn't address the Random Walk Hypothesis, limiting its prediction capacity. Furthermore, historical price, in general, isn't a strong predictor of future price, limiting its real-world applicability in areas like hedge funds or quantitative analysis.

Results:

The LSTM model predicted the direction of Google's [GOOGL] opening price movements with reasona-ble accuracy, albeit over predicting by an average of $3.04 per day, with a maximum error of $6.



Model 2: Sentiment Analysis

Sources: We first systematically extract data big news and financial websites like **Motley Fool, Yahoo Finance, Bloomberg, The Wall Street Journal, Reuters, Fortune, Business Wire, TheStreet.com**, and sort for news on anything that might affect any given stock. It then processes and runs sentiment scores on **100+ articles per stock** and finally returns the sentiment scores and over a week's period of time. This methodology inspects the other side of trading: emotions, and gives us a sense of recent market happenings that provide for a deeper analysis.

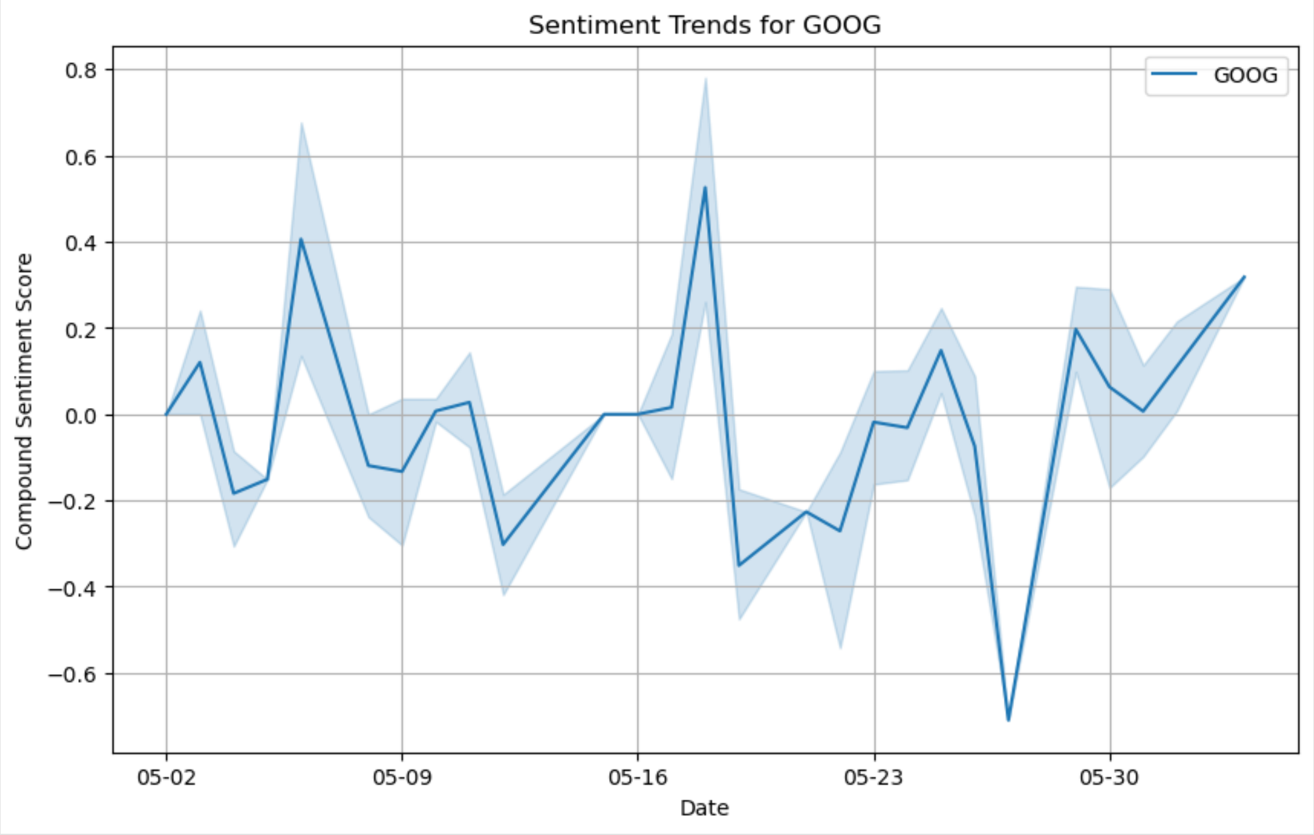
VADER: We used VADER, rule-based sentiment analysis tool. Strengths include simplicity, ease of interpretation, and low computational resource requirements. However, it struggles with complex sentences and sarcasm. Luckily, news channels seem to have straightforward sentiment.

Pros: Dynamic and complete overview of stock sentiment. Many news sources.

Cons: Reliance on textual analysis may miss nuanced implications of financial metrics. Accuracy may be influenced by potential biases in sourced articles.

Results:

On the right, this graph shows the sentiment trends for GOOGL. It scraped 100+ articles and builds an aggregated insight on stock sentiment is for that day. The above is for the past week.


Sentiment Trends for GOOG

Model 3: Recommendation System

Overcoming the Efficient Markets Problem:

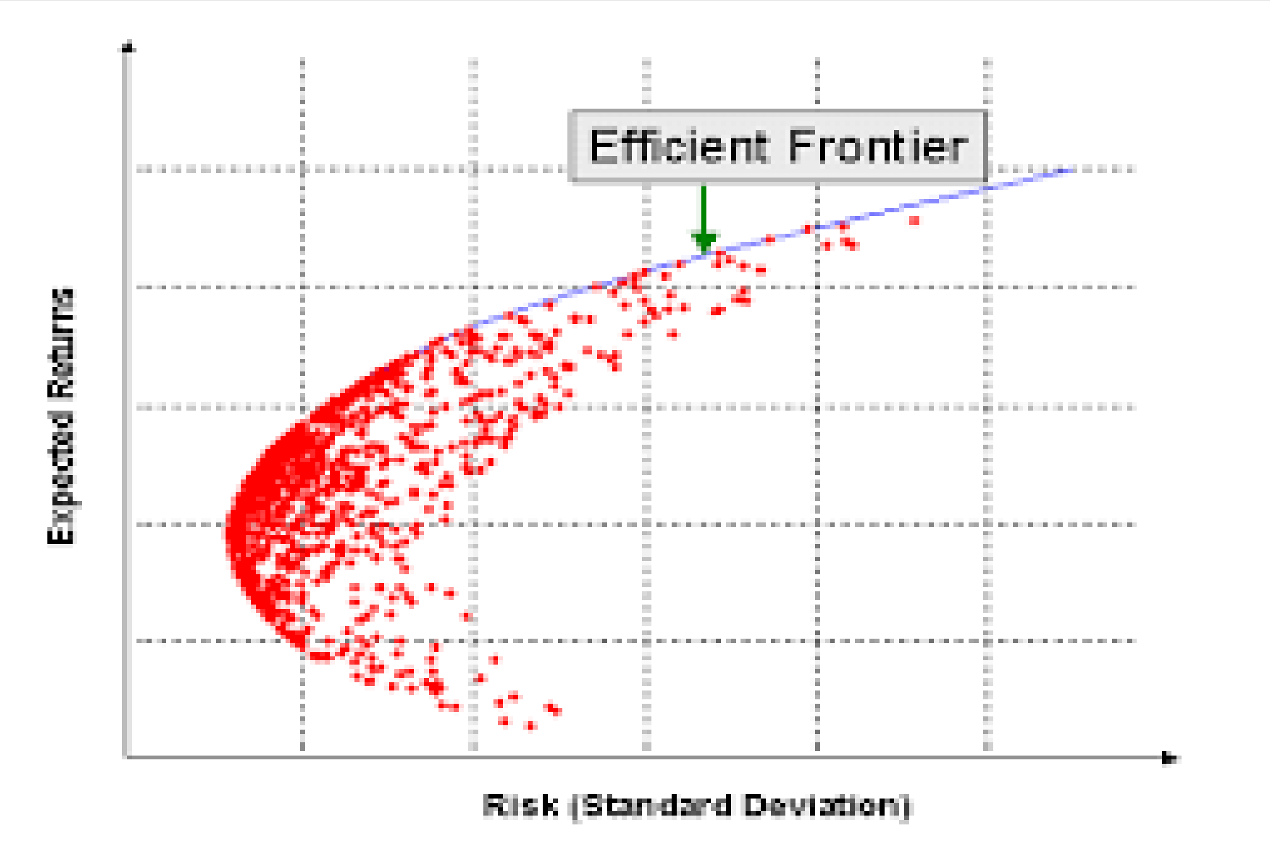
Since we know that Stock prices cannot be predicted consistently using only historical price movements, we make use of Modern Portfolio Theory to find a way to recommend Stocks to include in your portfolio without relying only Predicted Price.

Instead of 'Predicted Price' we focus on 2 Parameters: **Return and Risk**.

Portfolio Return is defined as the weighted average of Returns of all stocks in a given portfolio. Portfolio Risk refers to the potential for the actual returns of a portfolio to deviate from the expected or desired returns. It is measured as the weighted average of the standard deviation of the Portfolio of stocks.

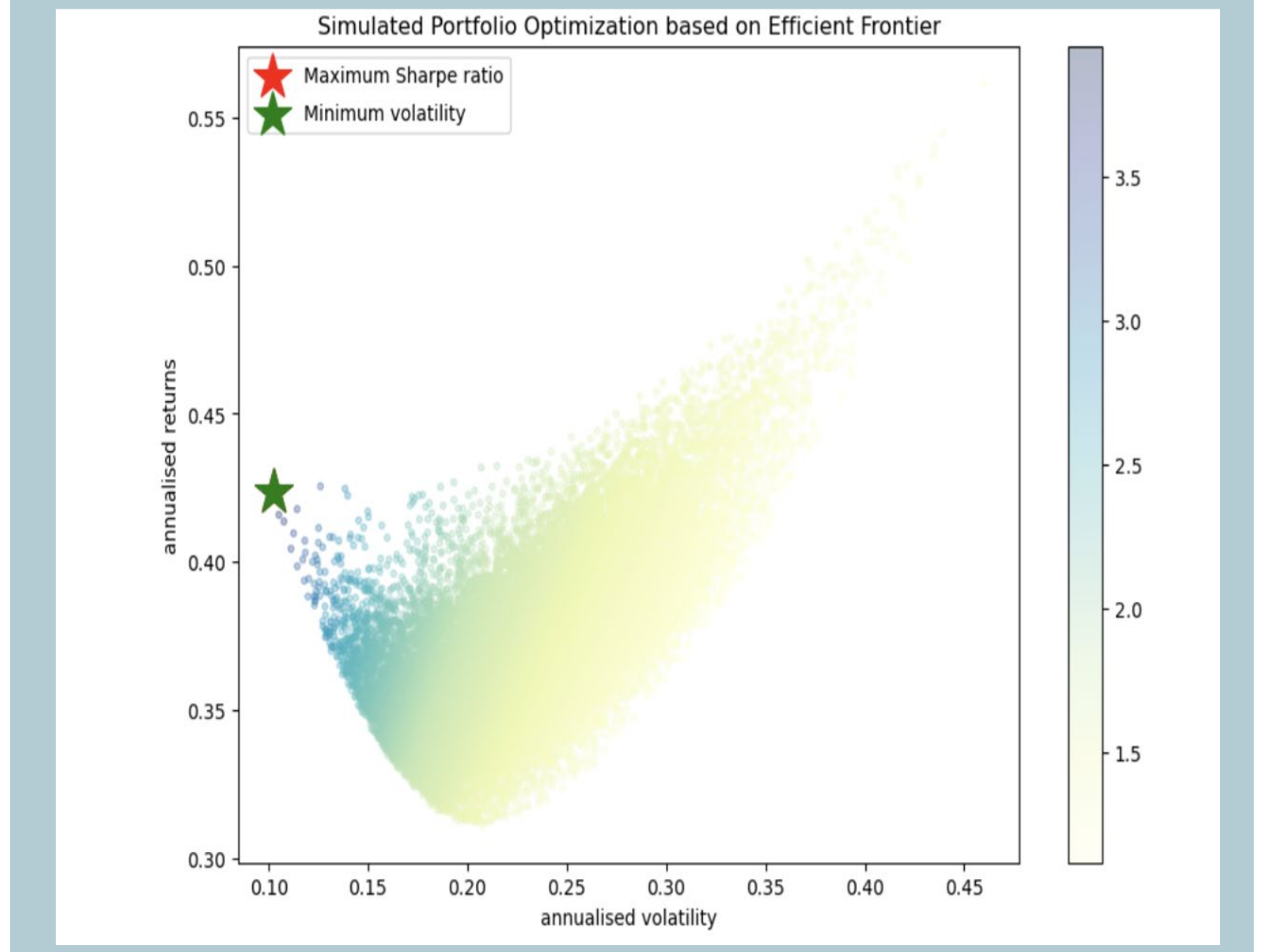
An Optimal Portfolio - Minimizes Risk & Maximises Returns.

The Efficient Frontier graphically illustrates the optimal portfolio combinations that achieve the highest expected return for a given level of risk or the lowest level of risk for a given expected return. The x-axis represents the standard deviation or volatility, which is a measure of the portfolio's risk, while the y-axis represents the expected return.



Once we predict the performance of a given stock using our Predictive Models (LSTM, News Sentiment) we feed the best performing stocks into our Efficient Frontier Model and that graphs the optimal set of Portfolios involving these stocks and presents the optimal ratio of stocks given.

Results:


Simulated Portfolio Optimization based on Efficient Frontier

```
--------------------------------------------------------------------------------
Maximum Sharpe Ratio Portfolio Allocation

Annualised Return: 0.42
Annualised Volatility: 0.1


            AAPL   AMZN  GOOGL  META
allocation  2.78  92.47   3.99  0.75
--------------------------------------------------------------------------------
Minimum Volatility Portfolio Allocation

Annualised Return: 0.42
Annualised Volatility: 0.1


            AAPL   AMZN  GOOGL  META
allocation  2.78  92.47   3.99  0.75
```

Future Steps

1. Enhanced Sentiment Analysis: Include more data sources, refine NLP for financial language.
2. Deep Learning Models: Explore use of CNNs or transformer models for stock predictions.
3. Portfolio Optimization Methods: Investigate advanced models considering transaction costs, taxes.
4. Real-time Application: Develop a tool for real-time portfolio recommendations.
5. Model Interpretability: Increase transparency in decision-making process.
6. Back testing and Performance Metrics: Back test models with historical data, establish performance metrics.