Project - Quantitative Finance

José Ángel García Sánchez, Sarra Ben Yahia, Yasmina Moussa, Souhayla Hedri

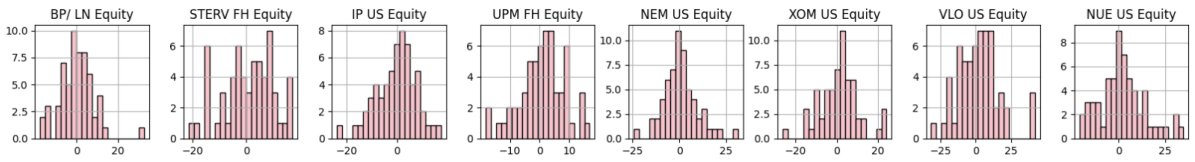
Part 1

1. Data description

First, we imported the “returns” database and proceeded to clean the data by replacing the '#N/A N/A' values with nan and then using the backward fill method, "Backward Fill", to replace these nan values. The reason for using the backward fill method in a temporal series is that financial data, being time-series data, is often dependent on previous values and can display trends or patterns. This method preserves the continuity of the series and minimizes the disruption to the underlying patterns or trends by using the previous value in the series to fill in the missing data.
After the data was cleaned, we performed a comprehensive descriptive analysis of our 60 stocks, including calculating the mean, standard deviation, skewness, kurtosis, and creating histograms, etc.

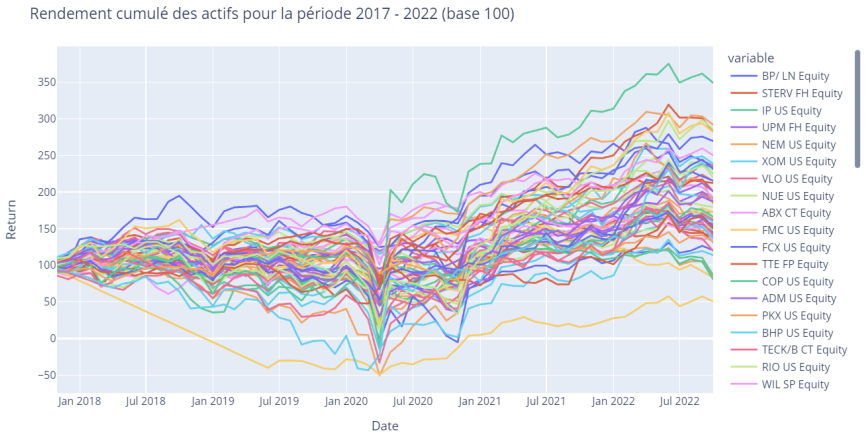
Statistical data analysis (for 8 actions)

	count	mean	std	min	25%	50%	75%	max	skewness	kurtosis
BP/ LN Equity	60.0	0.340750	8.215153	-17.42225	-4.775288	0.240264	4.845354	32.56261	0.717428	3.084020
STERV FH Equity	60.0	0.523926	9.403090	-21.38297	-4.456746	1.802086	7.828605	17.16692	-0.344520	-0.449652
IP US Equity	60.0	-0.226696	7.744599	-23.83469	-4.766816	0.846089	4.732543	17.51734	-0.437574	0.545165
UPM FH Equity	60.0	0.933991	7.545553	-18.17581	-3.391470	1.360282	6.047841	16.86093	-0.378970	0.161751
NEM US Equity	60.0	0.828072	9.153569	-24.11597	-4.713372	-0.518354	4.899593	31.36042	0.535311	1.808924
XOM US Equity	60.0	0.992708	9.524520	-26.18585	-4.474271	1.752606	5.669578	24.13793	0.086027	0.944225
VLO US Equity	60.0	1.916323	14.380837	-31.53208	-7.380648	2.168741	9.631668	41.78483	0.539300	1.032024
NUE US Equity	60.0	1.997453	12.152629	-20.80527	-6.005346	1.085803	8.014501	34.86191	0.536364	0.376308



2. Cumulative performance of our returns (base 100)

We have visualized the cumulative performance of our returns, where the starting value is set at 100. This representation provides us with a clear picture of the growth or decline in value over time, giving us a comprehensive understanding of the overall performance of our investments.



We observe a noticeable downward trend in the returns of all our stocks, as exemplified by the sharp decline between January and July 2020.

3. Correlation

The next step in our analysis was to examine the correlation between our selected stocks. By examining the relationship between different stocks, we can identify whether they move in tandem or independently of each other. This information is crucial in determining the composition of a well-diversified portfolio.

Our analysis revealed that stocks belonging to the same sector, such as energy or basic materials, tend to have a high degree of correlation. This means that their prices tend to move in the same direction and can be influenced by similar economic and market factors. To differentiate between strongly and weakly correlated stocks, we established a threshold of 0.7. This means that any stock pair with a correlation coefficient of greater than 0.7 was considered strongly correlated, while those with a coefficient below 0.7 were considered weakly correlated.

Having this information allows us to make informed decisions about which stocks to include in our portfolio. For example, it may be beneficial to limit the number of strongly correlated stocks in our portfolio to minimize the overall risk of the portfolio. On the other hand, including weakly correlated stocks can help to diversify the portfolio and reduce risk. This is because if one stock experiences a significant drop in value, the impact on the portfolio can be mitigated by the performance of other stocks that are not highly correlated.

4. Risk indicators

We have also calculated some synthetic risk indicators for a market risk equal to zero:

	sharpe_ratio	beta	treynor	var	var_gaussian	lpm_omega
BP/ LN Equity	0.041478	0.583136	0.584340	13.766847	13.853475	4.448327
STERV FH Equity	0.055719	0.469633	1.115607	15.346383	15.990633	5.225471
IP US Equity	-0.029272	0.407588	-0.556189	12.673869	12.512036	3.645603
UPM FH Equity	0.123780	0.313425	2.979948	12.663237	13.345321	4.706767
NEM US Equity	0.090464	0.094445	8.767784	11.424765	15.884354	5.404857
XOM US Equity	0.104227	0.858644	1.156134	14.055087	16.659150	5.754968
VLO US Equity	0.133255	1.194142	1.604770	19.950176	25.570696	9.106742
NUE US Equity	0.164364	0.567058	3.522487	15.882053	21.986748	8.073767

Our analysis of the Sharpe ratios showed that the expected return per unit of risk in these stocks is not very high. The benchmark Sharpe ratio was calculated to be 0.099. This means that if the Sharpe ratio of our stock is higher than the benchmark, the investment is expected to provide higher risk-adjusted returns compared to the benchmark index.

Additionally, we calculated the beta of each stock to assess the volatility of the stock relative to the benchmark index. A beta greater than one indicates that the stock is more volatile than the market. In our case, VLO US had a high beta, making it a very risky stock with the potential for high returns, whereas NEM US had a low beta, making it a less risky option with lower potential returns.

The Treynor ratios, another measure of performance, were not very high for most of the stocks. However, NEM US had a high Treynor ratio, which is consistent with its low beta. We also calculated the Value at Risk (VaR) and the Gaussian Value at Risk (gVaR) to assess the potential loss in an investment. The gVaR is based on the normal distribution, whereas the VaR is based on historical data. For example, the gVaR for VLO US indicated that there is a 95% probability that the loss will not exceed 25.5, while the VaR showed a loss of 19.9. Finally, we calculated the Omega, which measures the overall performance of a stock. A higher Omega value indicates that the returns from the investment will be higher.

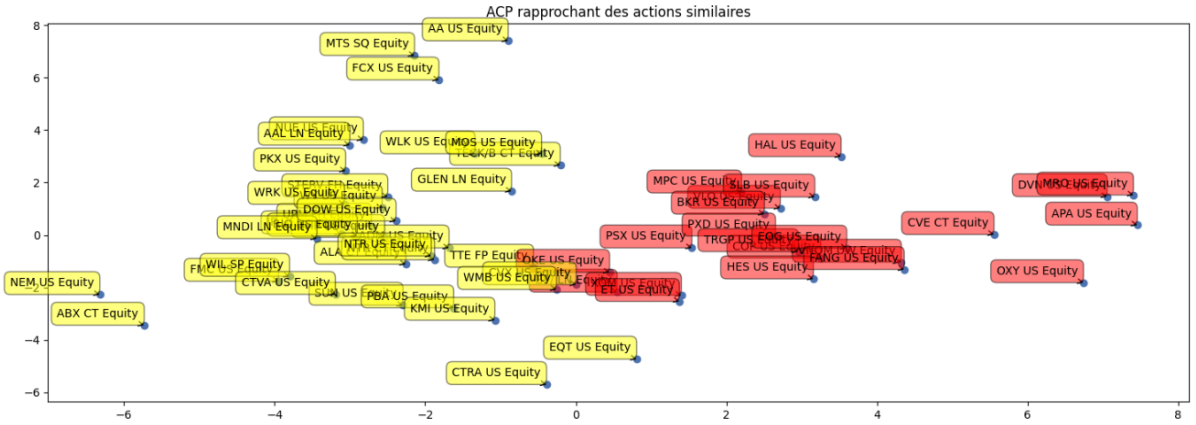
Part 2

We have selected a diverse set of 58 stocks from various sectors including Energy, Basic Materials, Utilities, and others from the DATA folder.

1. K-means clustering

To begin our analysis, we utilized returns-based K-means clustering as our initial approach. In order to determine the optimal number of clusters, we used the silhouette method and determined that K=2 was the most appropriate choice. To ensure that our data was appropriately scaled, we normalized the data prior to performing the K-means clustering. Unfortunately, the results obtained from the K-means clustering were not satisfactory, as the plot plan was not ideal.

Given these challenges, we decided to implement a principal component analysis (PCA) as a preprocessing step before performing the K-means clustering. This technique allowed us to reduce the dimensionality of the data and enabled us to gain a more insightful understanding of the underlying structure of the data. The resulting graph from the PCA-K-means analysis provided a clear and interpretable visualization of the clustered data, which was not possible with the initial K-means results.

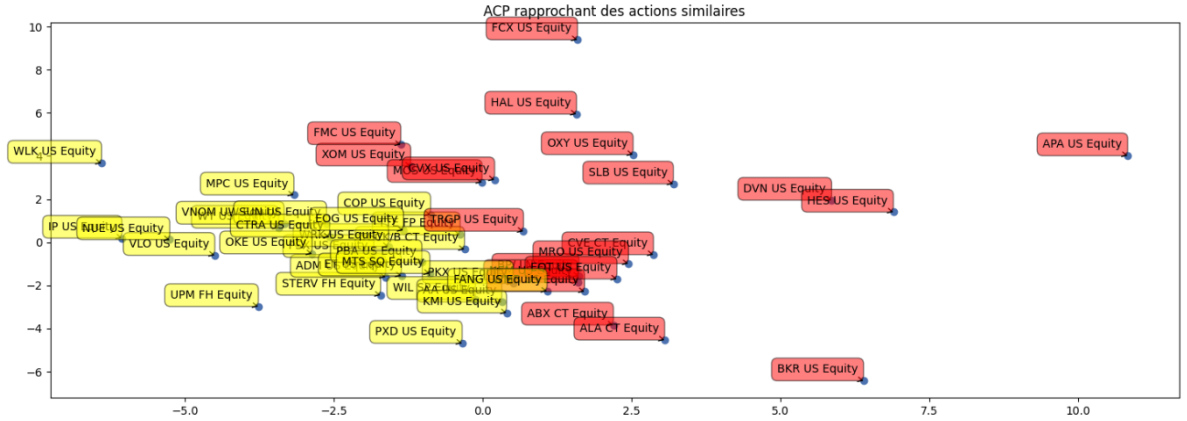


The observation made through the use of K-means clustering and PCA has revealed a significant correlation between the stocks that belong to the same cluster and those in the same sector. This finding highlights the importance of considering sectoral affiliation in the analysis of stock performance and its potential impact on clustering outcomes. The visualization of the clustering results through a PCA plot plan helps to further illustrate this correlation and provide a clear understanding of the relationships between stocks. This information can be useful for investors looking to make investment decisions by taking into account not just individual stock performance but also their sectoral associations.

2. CAH Clustering

Next, we proceeded to perform the Cluster Analysis on Hierarchies (CAH) based on the ROIC values. In order to ensure the accuracy of our analysis, we meticulously examined the presence of missing values in our database. To address this issue, we chose to remove any columns that contained more than 60% missing values, and then replaced the remaining missing values using a combination of the "Backward Fill" and "Forward Fill" methods.

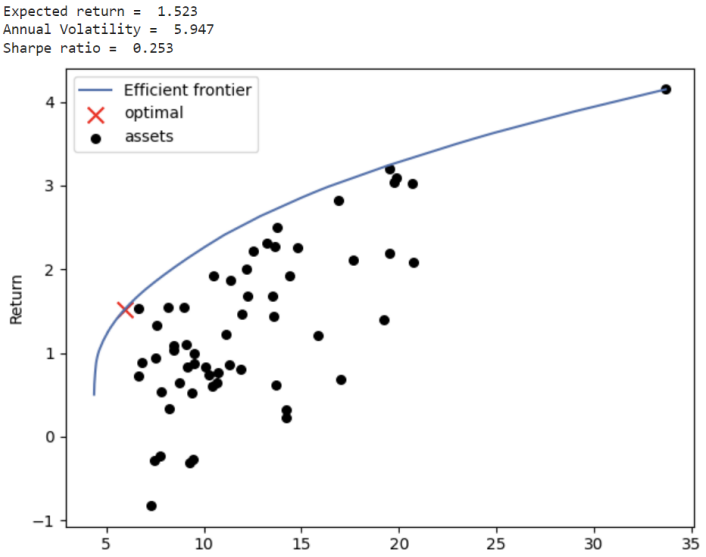
Subsequently, to enhance the interpretability of our results, we normalized the data and applied a principal component analysis (PCA) before utilizing the Agglomerative Clustering technique to arrive at the final graphical representation of our analysis.



Our initial findings have been corroborated by this analysis, as it has been demonstrated that the stocks which belong to the same cluster are also part of the same sector. This highlights the consistency and reliability of our observations, further strengthening our conclusion.

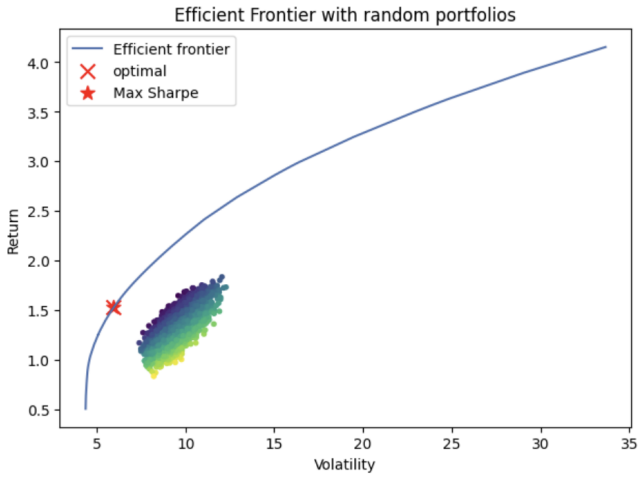
Part 3

1. Stocks on the same volatility return plan



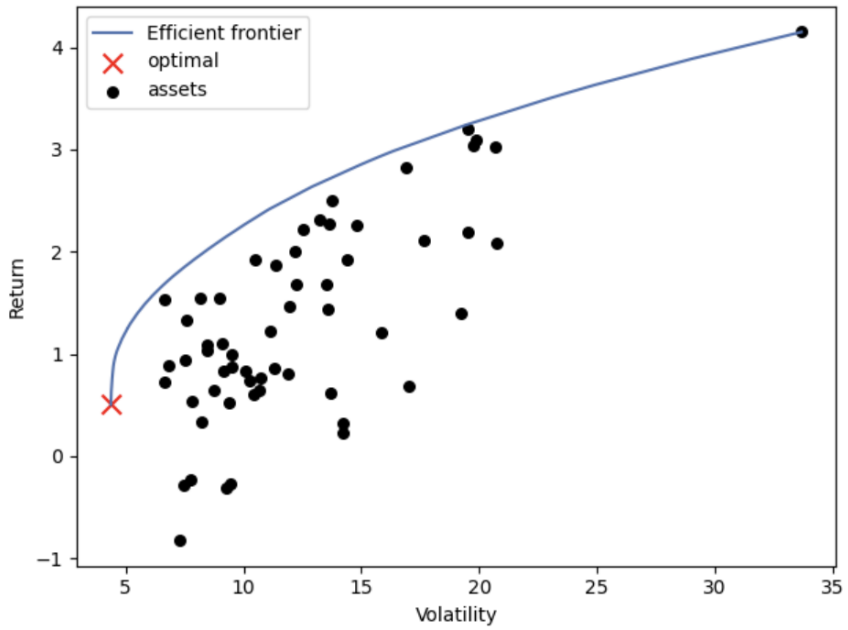
We have visualized the relationship between the expected returns and volatility of our stocks on a scatter plot, where the x-axis represents the volatility and the y-axis represents the expected returns. The efficient frontier is a line plotted on this graph, representing portfolios that offer the highest expected return for a given level of risk. The portfolios situated on the efficient frontier are considered "optimal." The graph enables us to make informed investment decisions by providing a clear comparison between risk and return. On the graph, the portfolio with the maximum Sharpe Ratio is represented by the red cross and is considered the "optimal portfolio," offering the best balance between risk and return.

2. Portfolios that can be put together on a plan return risk



On this graph, we are able to visualize all of the possible portfolios that can be created using our stocks, as well as the most optimal portfolio. This representation provides us with a comprehensive understanding of the relationship between risk and return for each potential portfolio.

3. Min Variance Portfolio

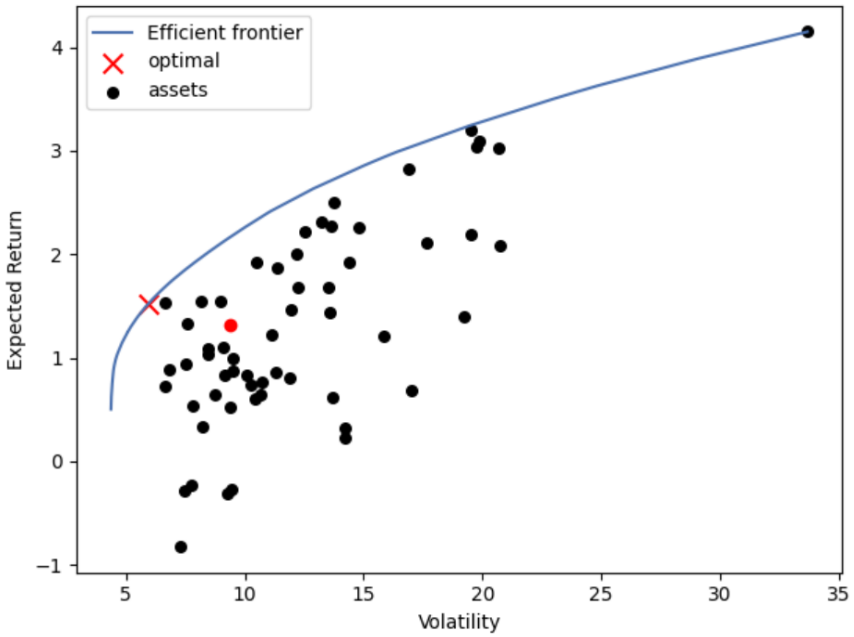


On this graph, we visually depict the minimum volatility portfolio which can be identified by the red cross on the efficient frontier. As the name suggests, this portfolio aims to provide the lowest level of risk to investors, as evidenced by its relatively low annual volatility of 4.373. Despite the reduced risk, the expected return of 0.503 and Sharpe ratio of 0.11, while lower than the portfolio with the maximum Sharpe ratio, still offer a viable investment opportunity for those seeking a balance between risk and reward.

The min variance portfolio consists of the following stocks and their respective weights :

- ➔ ABX CT Equity: 0.01788
- ➔ FMC US Equity: 0.02822
- ➔ BP/ LN Equity: 0.0948
- ➔ MNDI LN Equity: 0.09503
- ➔ CTVA US Equity: 0.10703
- ➔ CTRA US Equity: 0.1257
- ➔ WIL SP Equity: 0.16959
- ➔ UPM FH Equity: 0.17878
- ➔ NEM US Equity: 0.18297

4. Equi weighted Portfolio



In this graph, we depict the equally weighted portfolio represented by the red dot, which is characterized by assigning equal proportions to each individual security or asset within the portfolio. This approach results in each security or asset receiving an equal weight, irrespective of its market value or the outstanding number of shares.

The expected return for this portfolio stands at 1.323, with an annual volatility of 0.141, and a Sharpe ratio of 9.385. Compared to the other two portfolios, this strategy presents a balanced outcome in terms of expected return and a comparatively lower annual volatility, making it a less risky option.

Part 4

Abstract

This project aimed to create a portfolio of energy stocks by combining web scraping of Twitter data and sentiment analysis. The Twitter data was used to gain insight into the public opinion on various energy stocks, and the sentiment analysis was applied to understand the sentiment towards these stocks. Based on this analysis, an optimal portfolio was created to provide the highest returns while considering the public sentiment towards each stock. The combination of web scraping and sentiment analysis provided a comprehensive and data-driven approach to portfolio creation, with a focus on the energy sector.

Introduction

Twitter, founded in 2006, is a social media platform that has evolved into one of the largest sources of news and opinions. With over 330 million monthly active users, it is utilized by individuals, businesses, and organizations to express their thoughts and share news updates. From a data extraction perspective, Twitter can be accessed through its website which uses a combination of HTML, CSS, and JavaScript. To scrape data, one must send HTTP requests to the Twitter server and parse the HTML response using tools like Selenium. However, it's crucial to note that scraping Twitter data without permission is against the company's terms of service and comes with security challenges, such as CAPTCHA, IP blocking, and rate limiting.

Twitter's significance in the stock market has increased over the years. Investors and traders keep a close eye on the platform to stay informed about companies and their stock prices as it is a hub for real-time news and updates. A tweet from a prominent figure like Elon Musk can significantly impact the stock price of the company. Moreover, algorithms that employ web-scraped data are used in automated trading, analyzing tweets and other data in real-time to make investment decisions based on market sentiment.

Studies have demonstrated the potential of Twitter data in decision-making in the energy sector. For instance, research has shown that Twitter sentiment can impact energy stock returns, especially during periods of high market uncertainty and volatility. Another study found a correlation between energy stock prices and the sentiment of tweets about those companies.

These findings emphasize the value of considering social media sentiment when analyzing energy stocks as it can provide valuable insights into market sentiment and aid in predicting future stock performance. Our approach is influenced by the methods outlined in the Bloomberg article "Embedded value in Bloomberg News & Social Sentiment Data".

Methodology

1. Webscraping

The web scraping script is a program that automates the process of collecting information from a Twitter account. It does this by using a library in Python called selenium, along with a tool called Chrome web driver. The script starts by opening up an incognito window in Chrome and maximizing the window size. Then, it turns off any pop-up notifications. Next, it logs into the Twitter account using an email and password.

Once logged in, the script begins to gather information from individual tweets on the Twitter account. This information includes the username of the person who posted the tweet, the date it was posted, the text of the tweet, the number of replies to the tweet, the number of times it was retweeted, and the number of likes the tweet received.
The program has several functions that perform specific tasks. One function is used to extract information from each tweet card, another opens and sets up Chrome in incognito mode, another turns off pop-up notifications, and another logs into the Twitter account. There are also additional functions to handle exceptions and detect any suspicious activity.

2. Preprocessing

The code is a step-by-step process that helps prepare Twitter data for analysis. It starts by taking a collection of tweets and making sure certain elements are organized in a consistent way. This includes fixing data types, removing extra characters, and filling in any missing values. Additionally, the code filters out tweets that aren't written in English and simplifies the language by removing hashtags, links, and mentions. The end result is a clean, organized set of tweets that are ready to be analyzed and understood.

3. AI Models

Our code leverages the power of two pre-trained artificial intelligence models for sentiment analysis on stock market-related tweets.
The first model used in the project is the "Stock-Sentiment-BERT" model, which is based on the BERT architecture and fine-tuned on stock market-related tweets for sentiment analysis. It is trained to predict the sentiment of tweets related to publicly traded companies as either positive, negative, or neutral. This fine-tuning process enables the model to understand the context and nuances of stock market related tweets and make more accurate predictions about the sentiment expressed in these tweets.

The second model is the pre-trained language model "finBERT," developed by Prosus AI and made available through the Hugging Face library. finBERT is also a BERT-based model, fine-tuned on financial domain-specific data, including financial articles, news, and SEC filings. This fine-tuning process allows the model to have a deep understanding of financial language and terminology, enabling it to perform various NLP tasks such as sentiment analysis, named entity recognition, and question answering. The model's versatility allows it to be fine-tuned for specific financial tasks and used in various applications beyond just sentiment analysis.

By utilizing these two pre-trained models, the project is able to accurately predict the sentiment of tweets related to publicly traded companies, which is crucial for making informed investment decisions. The project uses these predictions to create an optimal portfolio strategy that takes into account the sentiment expressed in the tweets and other financial data.

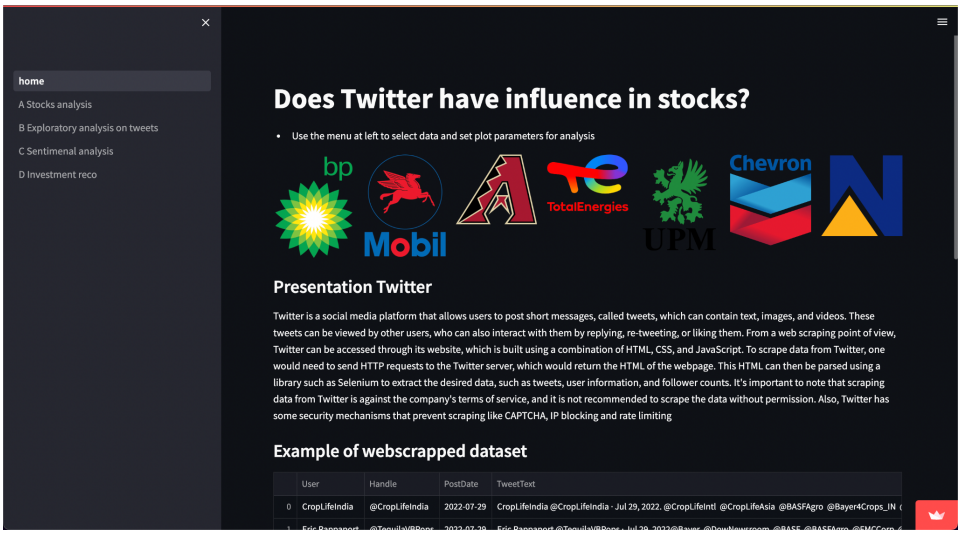
The code uses sentiment analysis models to classify tweets as positive or negative, and adds the sentiment predictions as new columns in the dataframe of tweets. The PortfolioModel class performs financial analysis by combining data from two sources to determine positive and negative sentiment ratios and returns of financial indices. It converts sentiment columns into numerical values for further processing.

4. Creating the portfolio

To construct the optimal portfolio, the code starts by counting the number of positive tweets received by each company in a specified year and month. Then, it calculates the positive ratio by dividing the number of positive tweets by the total number of positive and negative tweets. The method uses a positive ratio threshold of 0.5 to determine whether to invest in long or short positions. If the positive ratio is greater than 0.5, the portfolio is bullish and invests in long positions. If it's less than 0.5, the portfolio is bearish and invests in short positions. This method also calculates the cumulative returns for both long and short positions and returns a dataframe containing the position and return information.

Results

Uncover the results of our analysis with this interactive Streamlit app. You can try different portfolios, stocks, and options to see how they impact the outcome. Take your time to explore and understand the information presented in a dynamic and user-friendly format.



Streamlit

Enhancements

The algorithm's enhancement involves adding key features that increase its effectiveness and reliability. One such feature is the ability to recommend short selling, allowing investors to benefit from market downturns and negative sentiments, which can provide stability during periods of market volatility. Traditionally, sentiment analysis algorithms only focused on identifying positive sentiments, limiting investors' options.

Another improvement is to have a finer granularity on times, by calculating returns at 15-minute intervals instead of monthly. This provides investors with real-time, accurate and relevant market sentiment analysis, enabling good investment decisions.

By incorporating these two enhancements, the portfolio's risks can be reduced. The ability to recommend short selling and the increased time granularity provide a more comprehensive strategy that considers both positive and negative sentiments, reducing the risks of investment and providing investors with a more robust and reliable approach to making investment decisions. Ultimately, incorporating these improvements into the sentiment analysis algorithm offers investors a powerful tool for creating a more successful portfolio.

Additionally, incorporating a diverse range of news sources ensures a more comprehensive analysis of market sentiments, reducing the risk of bias towards a particular viewpoint. Integrating machine learning algorithms into the sentiment analysis process can also improve accuracy, as these algorithms learn from past market trends and make more accurate predictions based on the information. Finally, including sentiment analysis of other relevant factors, such as economic indicators, political events, and industry expert sentiments, offers a more comprehensive analysis of market sentiments.