BIG DATA PROJECT

Uncovering Stock Behavior Patterns: A Comprehensive Analysis

Introduction

In today's dynamic financial market, the capability to discern patterns and relationships within an increasing amount of stocks and data is paramount for informed decision-making in order to optimize portfolio performance and reduce risks.

To address this challenge, we tried to develop a community detection algorithm to identify subsets of stocks that exhibit similar behaviors.

In particular, through this algorithm we tried to uncover hidden dependencies and relationships among stocks that may not be immediately obvious, by analyzing a comprehensive stock price dataset spanning over the last few years.

What is a Community Detection Algorithm

Before we delve into the technicalities of the project let's first explain what a community detection algorithm is and how it can be useful for stock analysis.

A community detection algorithm is a computational technique used to identify cohesive groups or communities within a network. In the context of our project, we partitioned the stock market (network) into groups of nodes (stocks) that are densely connected internally but sparsely connected to nodes of other groups.

By grouping stocks into communities based on their price correlations, we can:

- identify sectors or industries with similar performance patterns
- detect market trends and anomalies
- optimize portfolio diversification strategies
- help investors identify potential investment opportunities and mitigate risk by spreading investments across stocks with different behavior patterns.

Methodology

1. **Data Collection**: we started by collecting historical stock data from 1st January 2020 to 1st January 2024, using Yahoo Finance. This dataset was a useful source of information on stock prices and movements.
2. **Data preprocessing**: we preprocess the collected data to ensure its quality and consistency. Some steps we performed were: handling missing values and organizing the data in a format suitable for analysis.
3. **Exploratory data analysis**: to gain insights into the behavior of individual stocks and understand the historical performance of each stock for further analysis.
4. **Development and application of a community detection algorithm**: this was the key component of our methodology. Specifically, we used the Louvain algorithm, a powerful technique in network science, to partition the stock market into distinct communities (stocks) based on correlations between stock prices to unveil similar stock behaviors.

Metrics for evaluation

In order to evaluate the effectiveness of our approach we considered the following metrics:

1. **Separability**: measures the degree to which identified communities are distinct from each other within the financial network. Higher separability indicates clearer boundaries between communities, suggesting that stocks within each community exhibit similar behavior patterns distinct from those in other communities.
2. **Clustering Coefficient**: it helps us assess the internal cohesion and structure of the identified communities, specifying the strength of relationships among stocks within each community. A higher clustering coefficient suggests that stocks within a community are densely interconnected, forming cohesive clusters.
3. **Fraction over Median Degree**: measures the proportion of stocks within a community that have a degree (number of connections) greater than the median degree of the entire network. Communities with a high fraction over median degree indicate that a significant portion of stocks within the community are highly connected, potentially indicating strong interdependencies or commonalities in behavior.
4. **Conductance**: evaluates the level of connectivity between a community and the rest of the network. Lower conductance values indicate that a community

is well-isolated from the rest of the network, with few connections to stocks outside it. High conductance values may suggest that a community is less cohesive or distinct, with greater overlap or interaction with stocks from other communities.

5. **Normalized Cut**: assesses the quality of the partitioning of the network into communities by considering both internal cohesion and external separation. A lower normalized cut value indicates a more effective partitioning of the network into cohesive and distinct communities.
6. **Cut Ratio**: it provides insights into the level of isolation and coherence of individual communities within the financial network. A lower cut ratio indicates that a community has fewer edges (links/connections) connecting it to stocks outside the community, implying a tighter and internally cohesive structure.

Implementation steps

Data collection

We tried to use Kafka to set up a data pipeline for fetching, producing, and consuming stock market data, facilitating the flow of data from Yahoo Finance to a Kafka topic for further processing or analysis.

Steps performed:

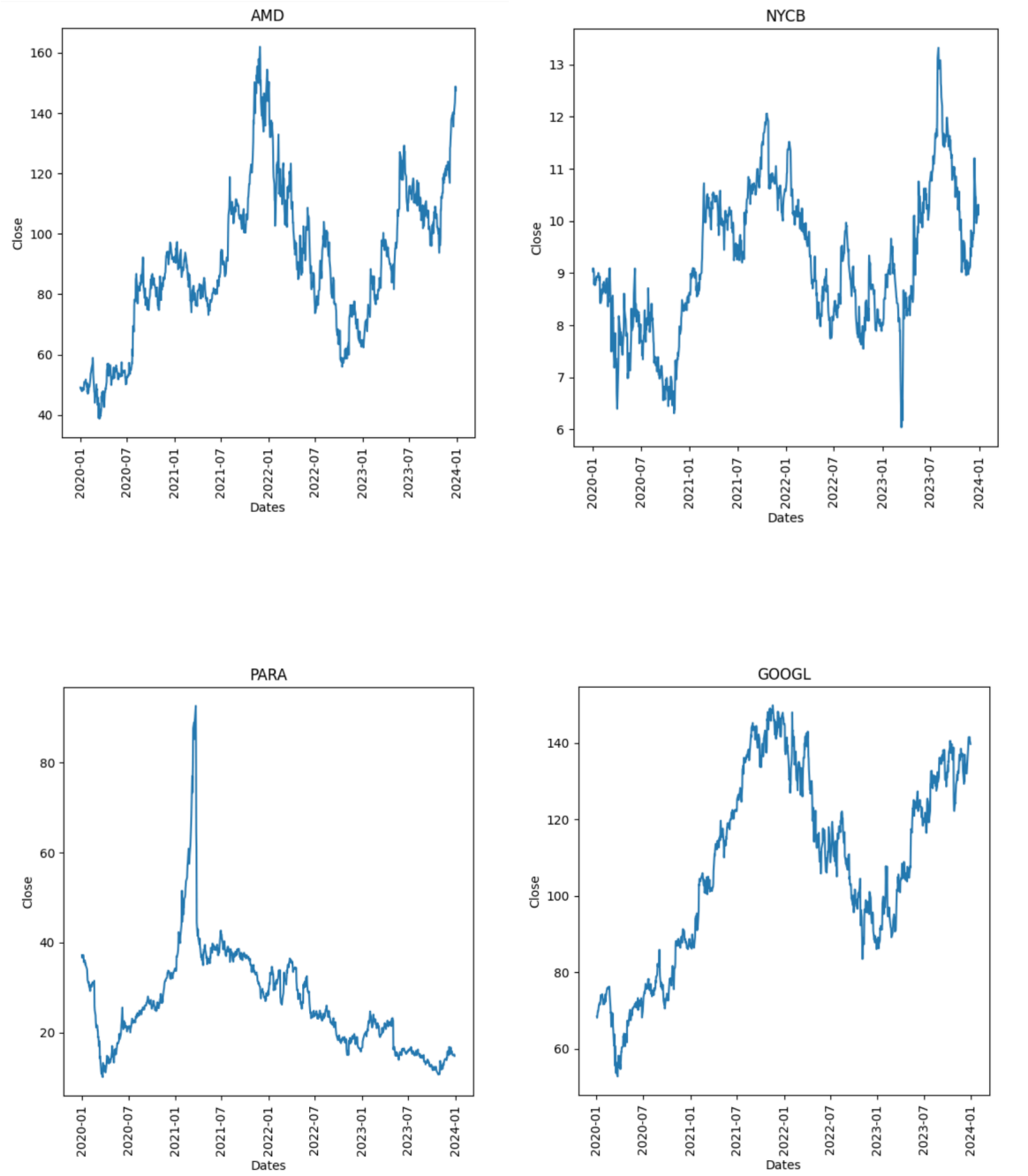
1. Kafka configuration: we set up the configuration parameters for Kafka
2. Creating a Kafka Producer: for producing messages to the Kafka topic “stock_Data” and we used JSON serialization to encode the stock data into UTF-8 encoded bytes before sending.
3. Downloading and producing the stock data from Yahoo Finance to the topic created.
4. Creating a Kafka Consumer: for consuming messages from the Kafka topic.

Data analysis

Stock behavior over time

After fetching and consolidating the stock data, we performed basic exploratory data analysis by generating line plots for the adjusted closing prices ('Adj Close') of each

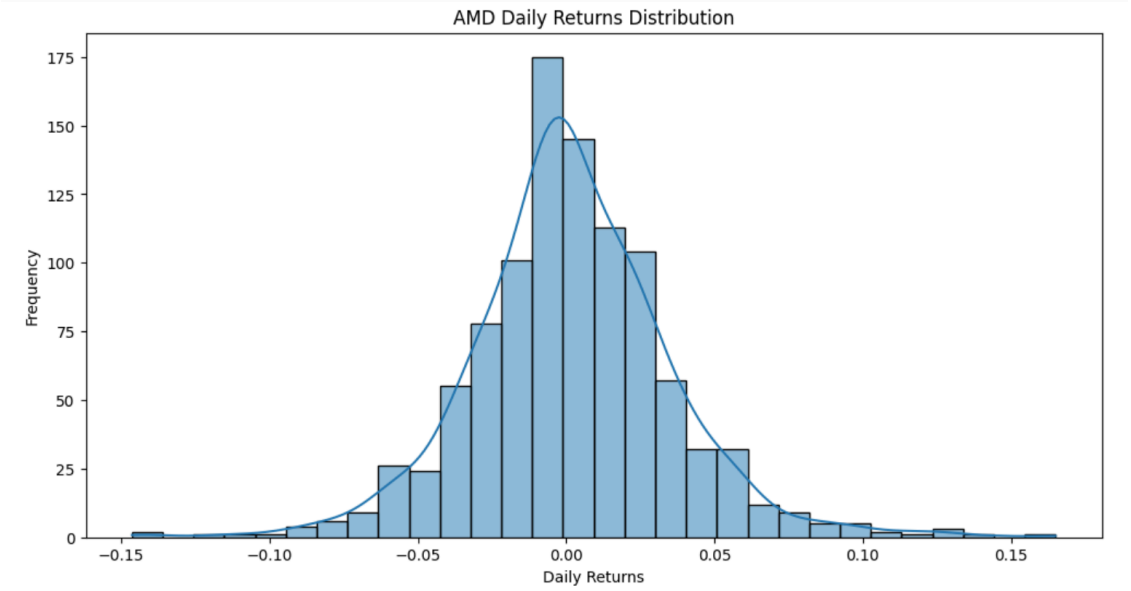
stock against time. These visualizations offer insights into the historical trends and patterns of each stock's closing prices over the specified time period.

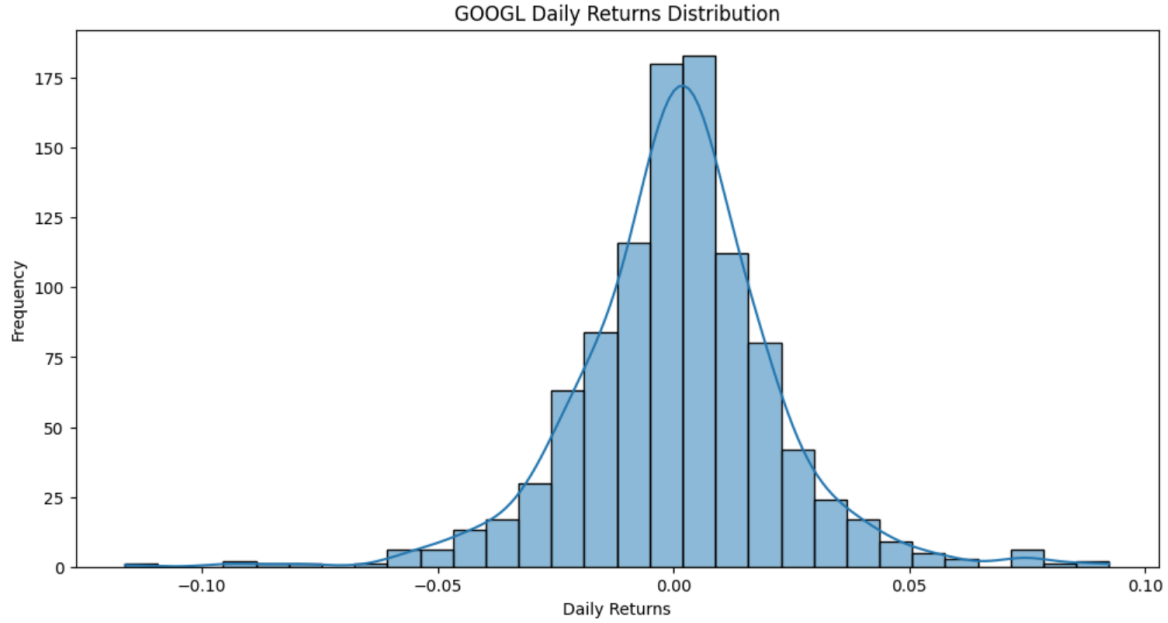


Daily returns distribution and Average daily returns

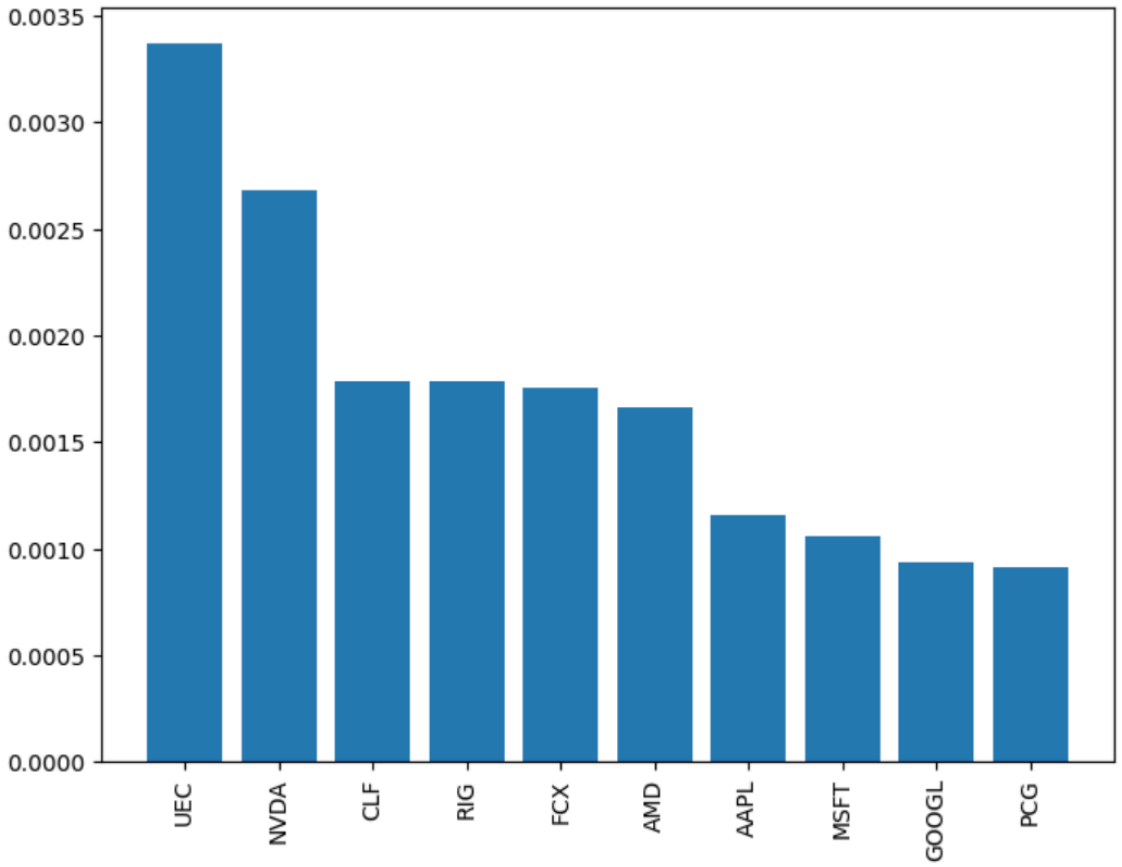
Then we analyzed the daily returns of each stock and identified the top 10 stocks with the highest average daily returns.

For each stock ticker in the list of tickers, we calculated the daily returns by taking the percentage change in the closing prices (`'Close'`) of the stock. This is achieved using the `pct_change()` method. The `dropna()` method is then used to remove any NaN (Not a Number) values from the resulting series. The histogram is then used to visualize the distribution of daily returns for each stock, providing insights into the volatility and frequency of returns.





Then we calculated the Average Daily Returns, identifying the top 10 performing stocks and visualized them in a bar plot.



The x-axis of the plot represents the ticker symbols of the top stocks, while the y-axis represents their corresponding average daily returns.

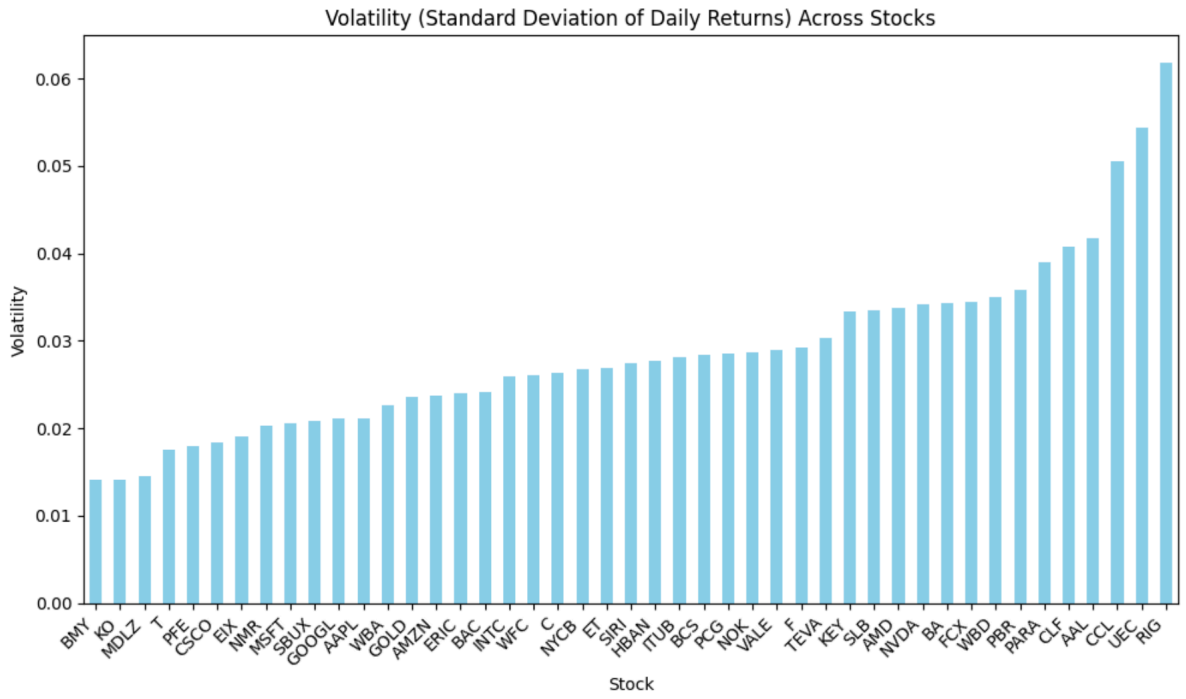
The plot provides a clear visual comparison of the top-performing stocks, aiding in decision-making processes such as portfolio optimization or investment strategies.

Volatility

In the following graph, we studied the volatility of each stock by calculating the standard deviation of daily returns and then collected the values in one single plot for better comparisons among stocks.

This step is important for two main reasons:

1) Risk Assessment: Volatility measures the variability or risk associated with an investment. Stocks with higher volatility tend to have larger price fluctuations, indicating higher risk.
2) Pattern Recognition: Stocks with similar volatility profiles may exhibit similar patterns in their price movements. In this way we can identify groups of stocks that move in similar ways over time.



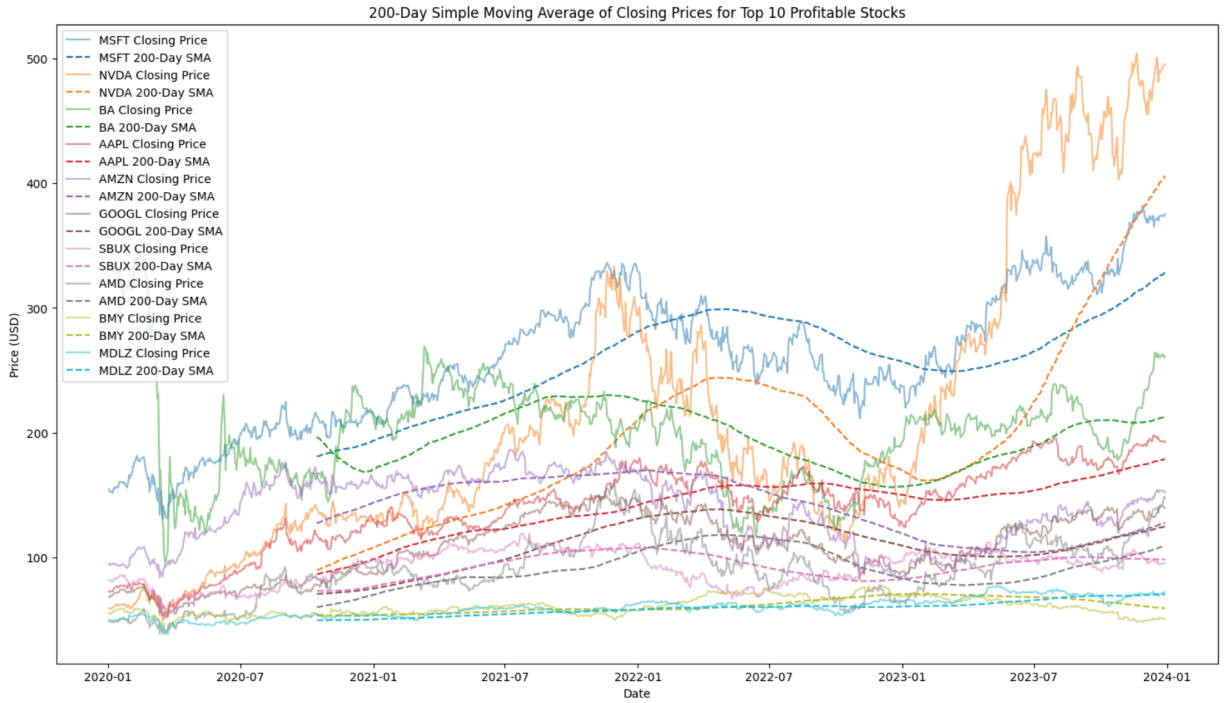
Simple Moving Averages of closing prices

A simple moving average (SMA) is an arithmetic moving average calculated by adding recent prices and then dividing that figure by the number of time periods in the calculation average.

A simple moving average helps identify whether the stock is trending upwards, downwards, or remaining relatively stable over time. In fact, if the simple moving average points up, this means that the stock price is increasing, if it is pointing down, it means that the stock price is decreasing.

Here we calculated the 200-day SMA of closing prices for the first 10 profitable stocks.This gave us an overview of stock trends by smoothing out short term fluctuations in order to understand which stocks are moving in a similar direction.

Stocks with similar SMAs may belong to the same sector, industry, or have similar market dynamics.



Community detection algorithm

Correlation matrix of stocks

To develop our community detection algorithm, we first calculated the correlation matrix of closing prices for our stocks. This is achieved using the `corr()` method on the DataFrame `closing_prices`.

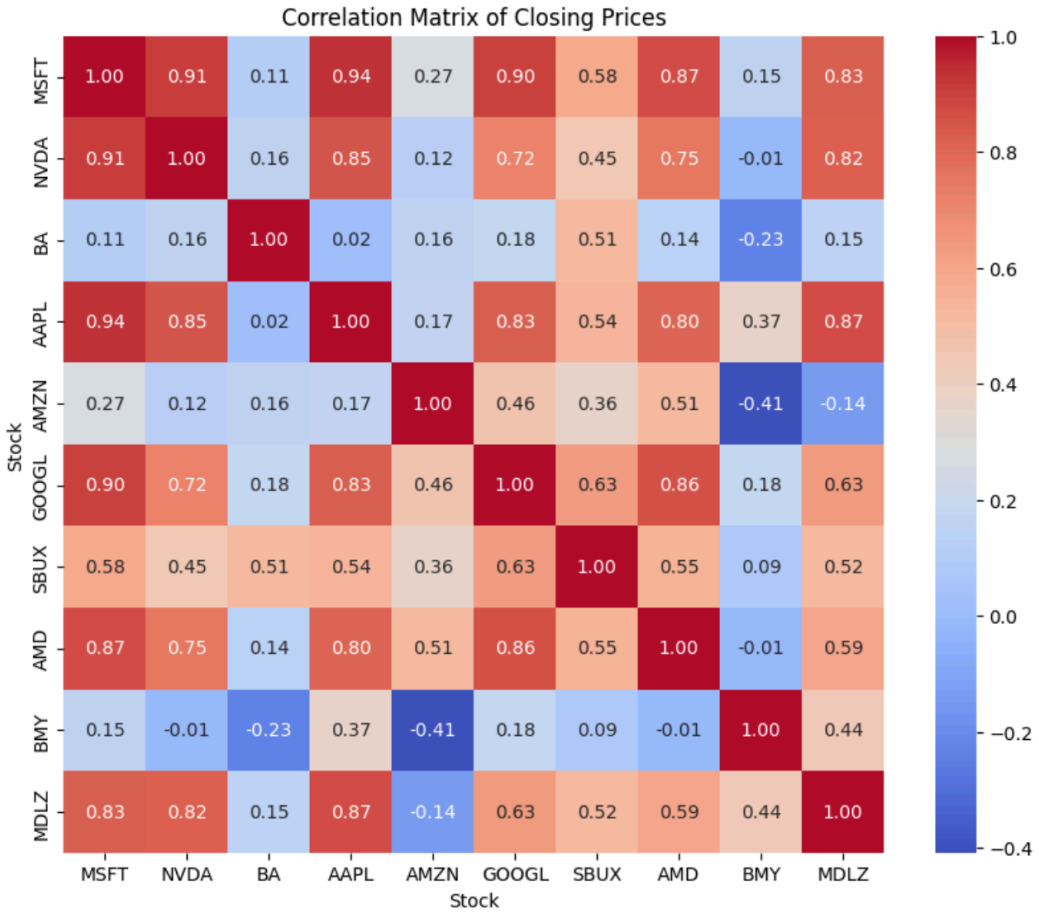
After, we visualized the correlation matrix as a heatmap using `sns.heatmap()`.

Each cell in the heatmap represents the correlation coefficient between two stocks. The correlation coefficient ranges from -1 to 1.

Cool Colors (e.g., Blue): represent negative correlation coefficients, indicating that the prices of the two stocks tend to move in opposite directions.

Warm Colors (e.g., Red): represent positive correlation coefficients, indicating that the prices of the two stocks tend to move in the same direction. The darker shades of red typically indicate stronger positive correlations.

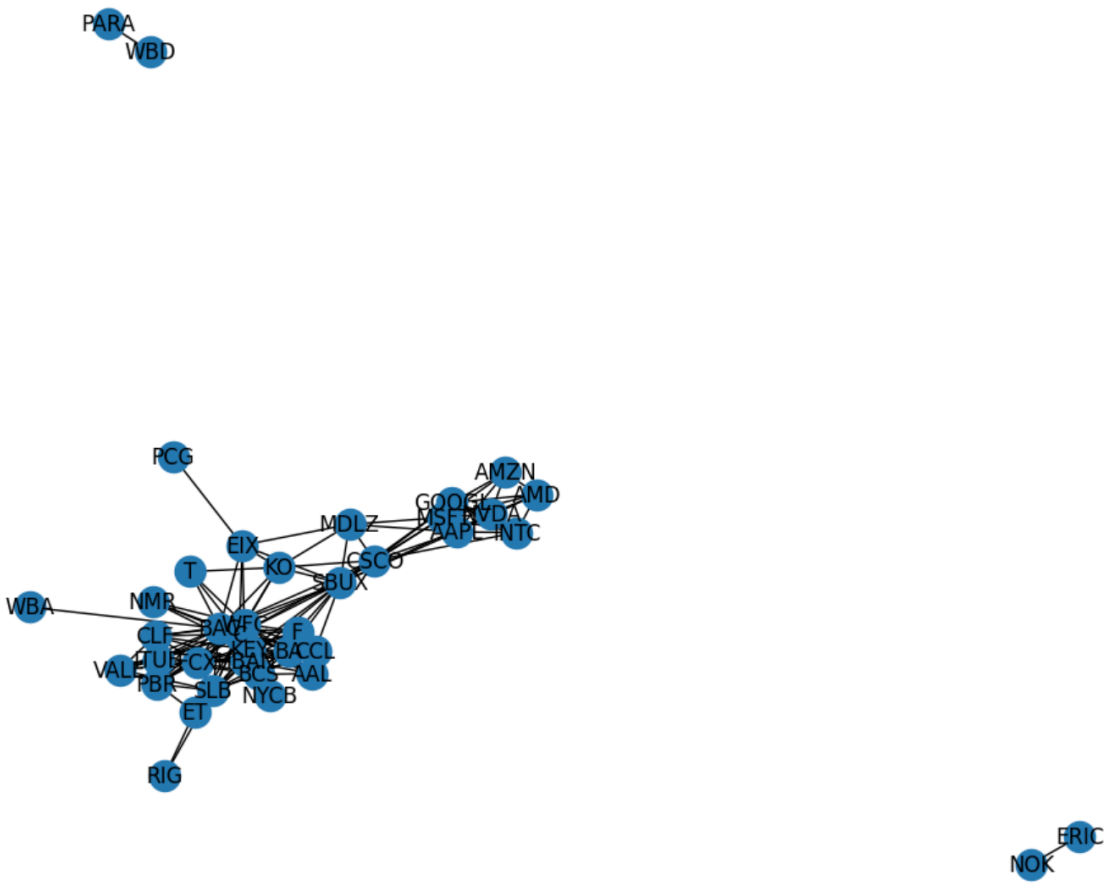
Neutral Colors (e.g., White): These colors represent correlation coefficients close to zero, indicating little to no correlation between the prices of the two stocks.



*In the graph above, we only reported the correlation matrix for the 10 most profitable stocks.

Stock network graph

In the following graph we created and visualized a stock network where stocks are represented as nodes, and the presence of an edge between two stocks indicates a strong correlation between their price movements. The threshold value 0.5 that we set determines which correlations are considered strong enough to be included in the network.

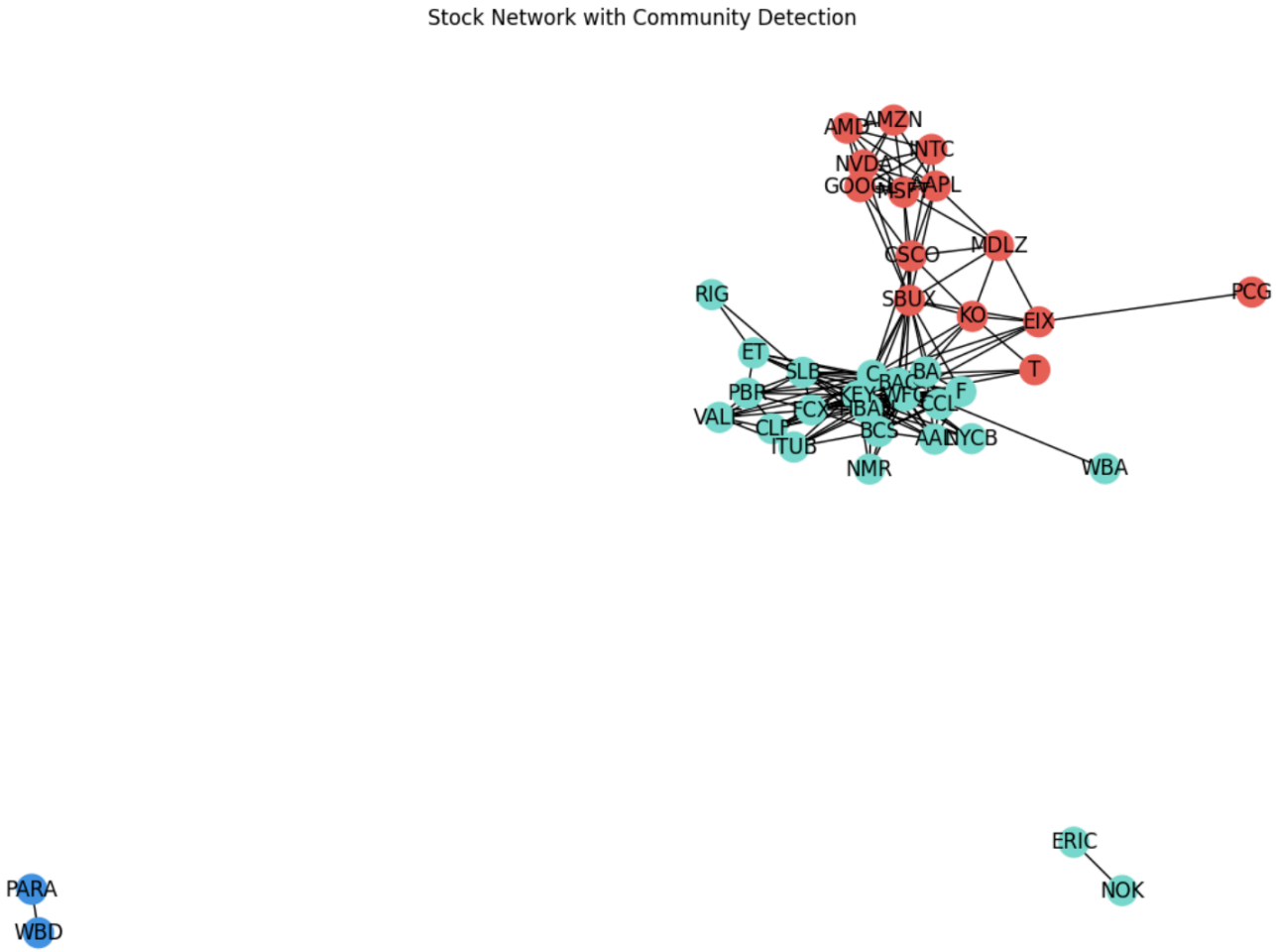


Louvain community detection algorithm

We applied the Louvain community detection algorithm to the stock network in order to partition the nodes (stocks) into different communities based on their connectivity patterns.

Each node (stock) is represented by a circle. Nodes belonging to the same community are assigned the same color. The connections (edges) between nodes represent the correlations between stocks.

Tickers connected together suggest that there are correlations between them, meaning that there are price movements that are correlated across the entire set of stocks being analyzed. However, the different colors indicate that despite being interconnected, the stocks tend to form distinct communities or groups based on their correlations, which may represent different sectors or other underlying relationships in the stock market.



Conclusion

In the following image we show the results of the evaluations metrics obtained by implementing the Louvain community detection algorithm.

```
Clustering Coefficient: 0.6348609517637859
Fraction Over Median Degree: 0.48717948717948717
Normalized Cut: 0.1144578313253012
Separability: 0.8855421686746988
Conductance: 0.1144578313253012
Cut ratio: 0.1144578313253012
```

These results give us several key insights into the effectiveness of the implemented method and the structure of the interconnected stocks.

The *clustering coefficient* of 0.634 indicates a relatively high level of clustering or interconnectedness within the communities identified by the algorithm. This

suggests that stocks within the same community tend to have more connections among themselves compared to stocks in different communities.

The *fraction over median degree*, at 0.487, indicates that nearly half of the stocks have a number of connections higher than the median degree in the network. This suggests that there is a significant presence of stocks with above-average connectivity, potentially indicating influential or central stocks within their respective communities.

The *normalized cut* of 0.114 reflects the quality of the partitioning achieved by the algorithm, with a lower value indicating a better division of the network into communities. A low normalized cut suggests that there are relatively few edges connecting nodes from different communities.

Furthermore, the *separability* value of 0.886 indicates a strong separation between the identified communities, implying that the stocks within each community exhibit similar behavior compared to those in other communities.

The *conductance* value of 0.114 further supports the idea of well-defined communities, in fact a low level of conductance indicates better separation between communities and higher cohesive and internally connected stock subsets.

Finally, the *cut ratio* of 0.114 indicates that we have a moderate level of separation between the detected communities, with a relatively low proportion of edges connecting different subsets of stocks.

Overall, these results highlight the importance and effectiveness of the Louvain community detection algorithm in uncovering meaningful patterns and relationships within stocks, providing valuable insights for portfolio diversification, risk management, and investment decision-making.