Pairs Trading Algorithm for Financial Markets

This document contains the strategy, analysis, report and results related to the project

[Shashank Yadav (22123041)]

Overview :

Pairs Trading strategy :

This project develops a pair trading strategy using clustering techniques to identify pairs of stocks with correlated price movements. Pair trading involves identifying two stocks with a historical correlation, betting that their prices will revert to this historical relationship over time. The primary objective is to leverage machine learning algorithms to cluster stocks, generate trading pairs, and analyze the performance of the trading strategy.

Collecting the Data :

Historical stock price data was sourced from Yahoo Finance, covering a broad spectrum of the stock market to ensure a diverse set of data. I have obtained 6 years worth of data for the S&P 500 stock by using Yahoo Finance. S&P is a stock market index that measures the stock performance of 500 large US companies.

ticker	A	AAL	AAPL	ABBV	ABNB	ABT	ACGL	ACN	ADBE
Date									
2018-01-02	64.520721	51.647564	40.615883	73.593399	NaN	52.570995	29.433332	139.834564	177.699997
2018-01-03	66.162399	51.014027	40.608822	74.745018	NaN	52.687248	29.459999	140.479965	181.039993
2018-01-04	65.666092	51.335663	40.797440	74.318787	NaN	52.597839	29.570000	142.143387	183.220001
2018-01-05	66.715988	51.316174	41.261929	75.612534	NaN	52.749847	29.453333	143.315918	185.339996
2018-01-08	66.859138	50.809349	41.108669	74.401054	NaN	52.597839	29.456667	144.461182	185.039993
...	...	...	...	...	...	...	...	...	...

Missing values were addressed using the forward fill method to maintain data continuity. Data was standardized using "`StandardScaler`" to ensure uniformity and comparability across

different stocks. I dropped the columns with missing data greater than 20%, and the columns with missing values less than 20% ,then filled them with the last known data.

Applying Machine Learning Model for Pair Generation:

I've harnessed the power of clustering algorithms to bring a fresh perspective to stock analysis. You might wonder, why clustering? By grouping stocks with similar price movements, clustering simplifies the task of identifying potential trading pairs. This method hinges on the idea that stocks within the same cluster will continue to move in sync over time. The result? Stocks are neatly organized into clusters, ensuring high intra-cluster correlation, and making the selection of trading pairs a breeze.

In our clustering endeavor, we're focusing on the volatility and performance of stocks. To achieve this, we aim to capture their variance and returns on an annual basis.

To prepare our data for the algorithm, I utilized the `StandardScaler` from the `Scikit-Learn` library to scale the variables, ensuring they have a mean of 0 and a variance of 1. This step is crucial for maintaining consistency and accuracy in our analysis.

	returns	volatility
A	0.162080	0.297546
AAL	-0.078995	0.590943
AAPL	0.319156	0.321000
ABBV	0.173726	0.275831
ABT	0.148595	0.259761

➡

	returns	volatility
A	-0.037322	-0.557946
AAL	-2.153431	2.601144
AAPL	1.341452	-0.305404
ABBV	0.064903	-0.791758
ABT	-0.155693	-0.964783

Next, we'll compare two clustering algorithms to generate pairs. For this task, we'll employ unsupervised learning methods and focus on two models:

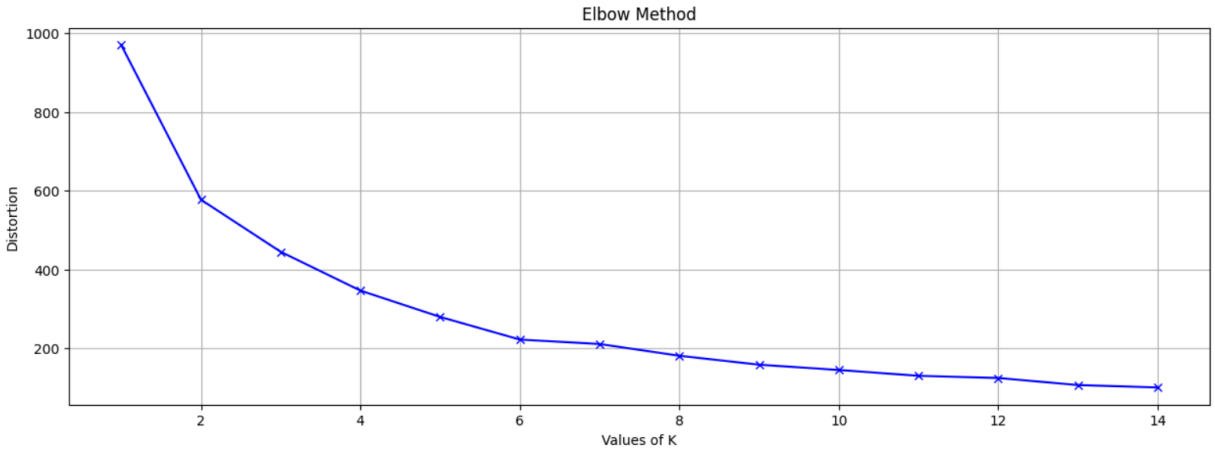
1. K-means Clustering

2. Hierarchical Clustering

This comparison will help us determine which approach is more effective for our pair generation needs.

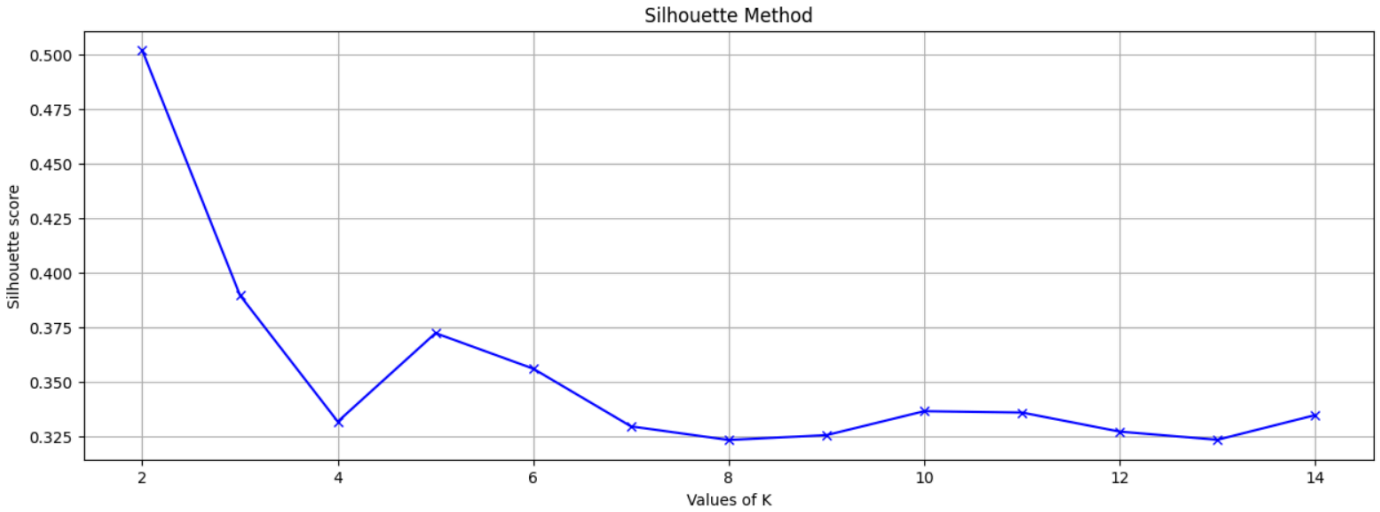
First, since we don't know the optimal number of clusters, we'll use the `“elbow method”` to determine it. This technique helps us identify the point where adding more clusters no longer significantly improves the model, ensuring we select the most appropriate number of clusters for our analysis.

Using K-Means Clustering :



By observing the chart generated from the elbow method, we conclude that the optimal number of clusters falls between 4 and 6. After the 6th iteration, additional clusters become less informative. To confirm the exact number of clusters, we utilized the `kneed library`, which pinpointed the optimal number as 5.

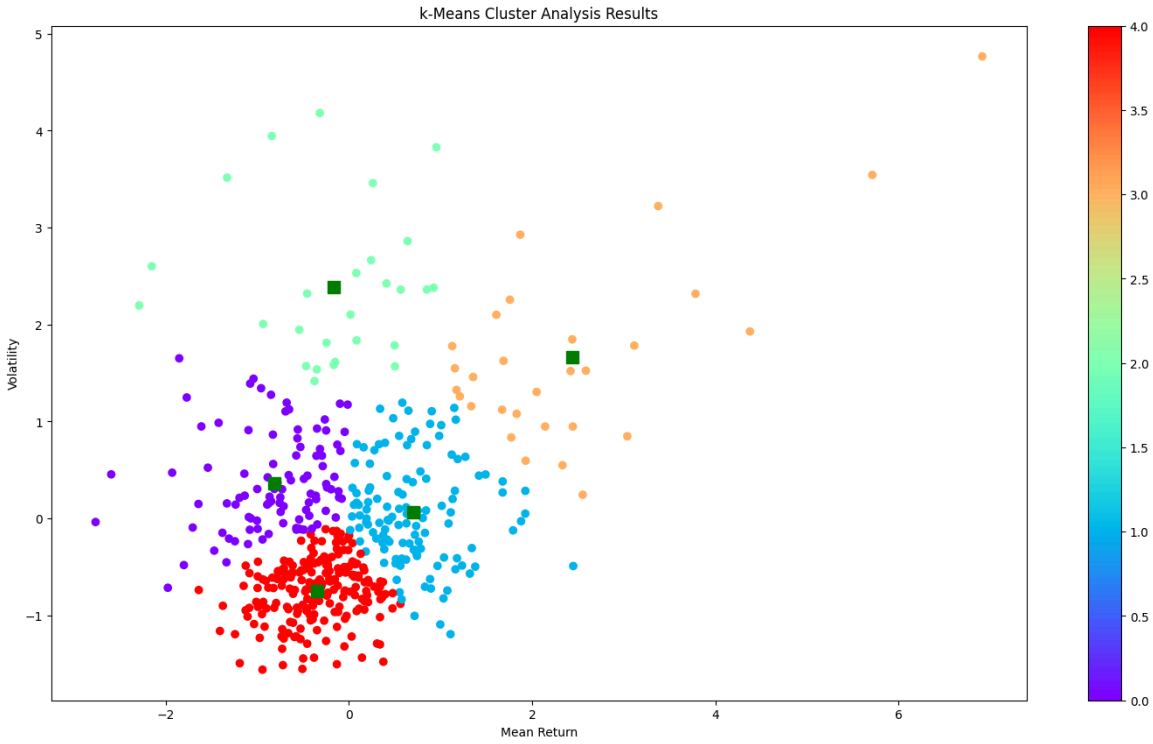
Thus, for our K-Means clustering task, we'll proceed with 5 clusters, ensuring our grouping of stocks is both efficient and informative.

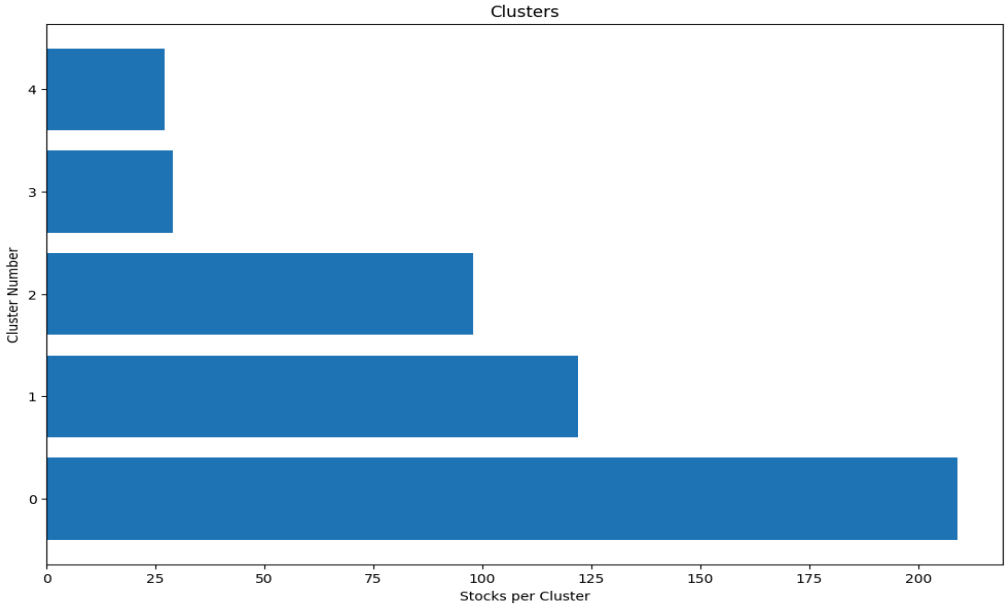


The silhouette method also indicates that 4 clusters might be optimal. However, I'll proceed with the K-means algorithm using 5 clusters, as confirmed by the kneed library.

Building the K-Means Algorithm with 5 Clusters :

With the optimal number of clusters established, it's time to implement the K-Means algorithm using 5 clusters. Let's dive into the results and see how the stocks are grouped, aiming for high intra-cluster correlation and effective pair selection.

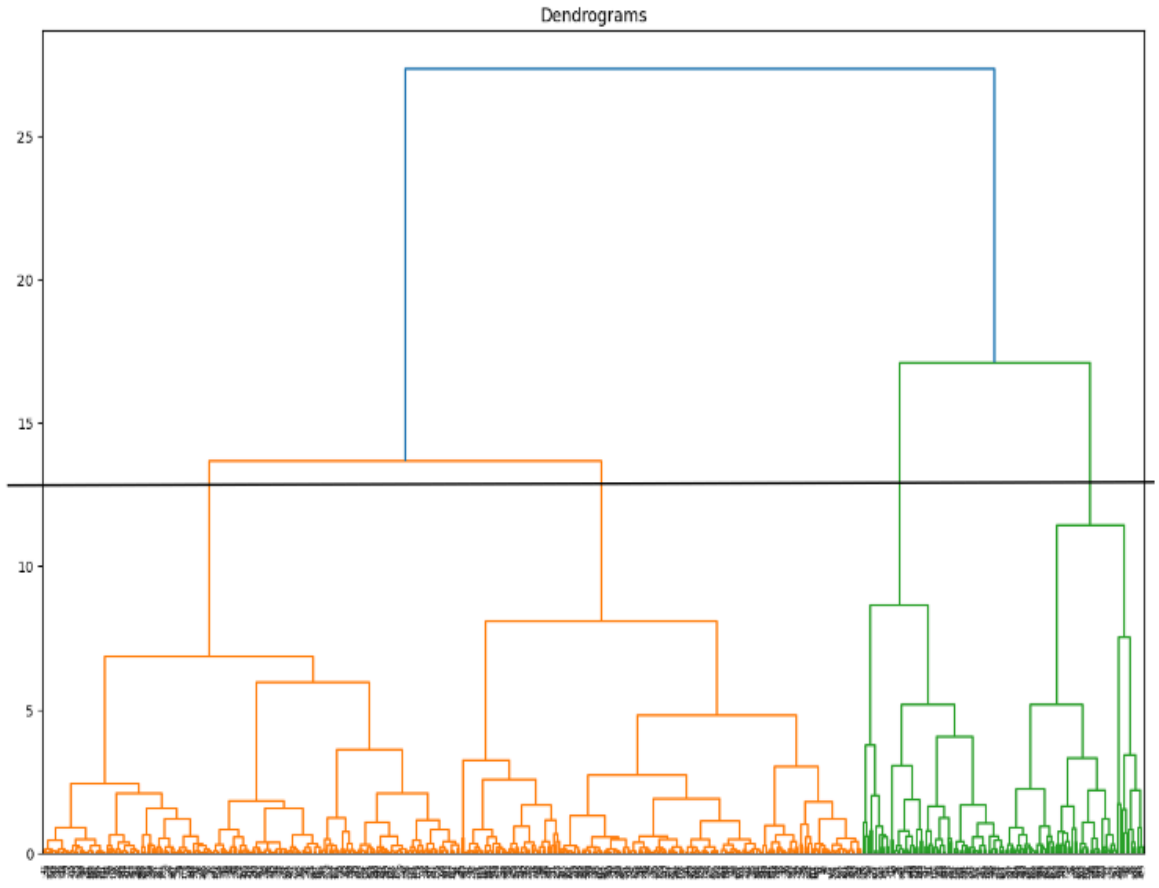




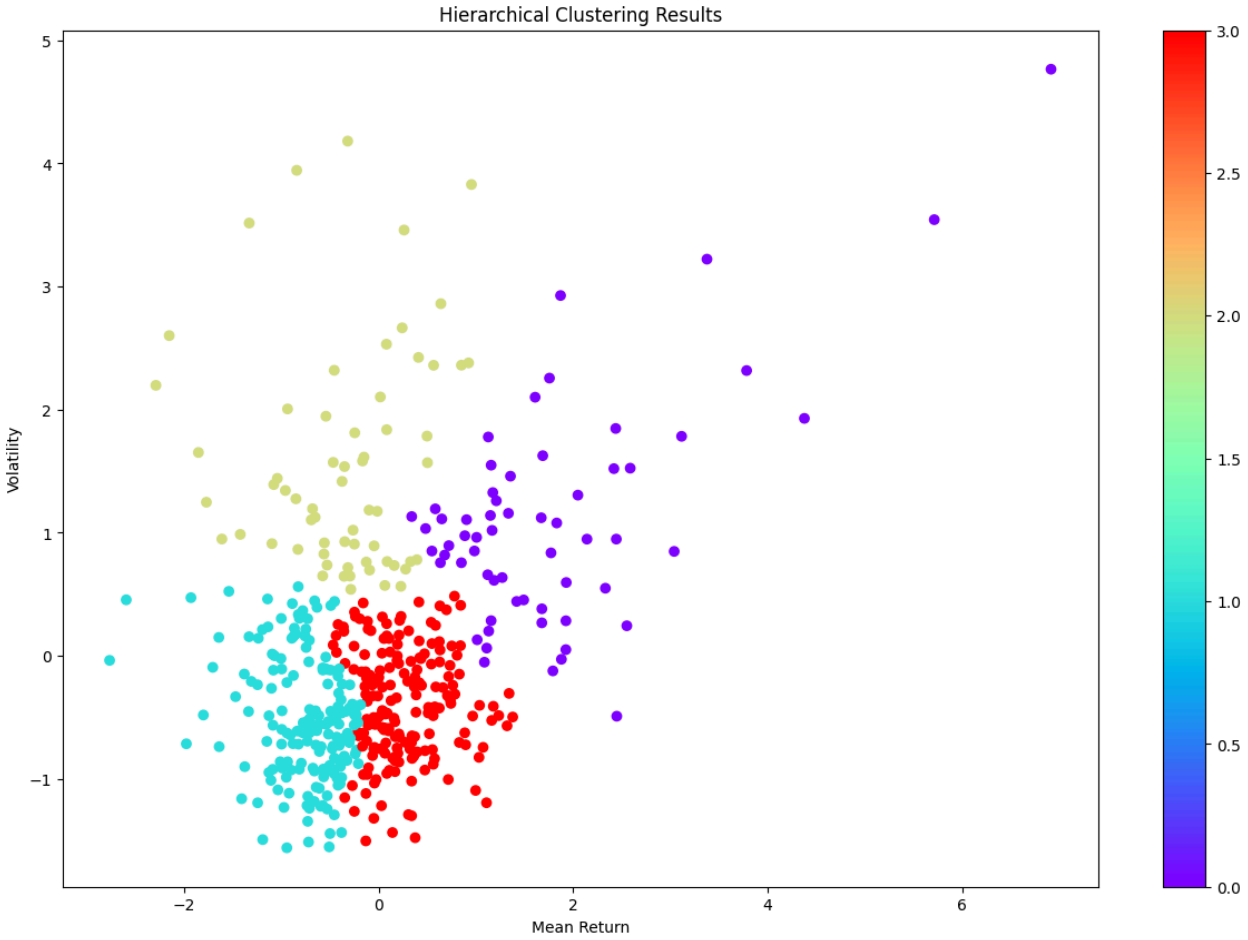
Using Hierarchical Clustering :

Hierarchical Clustering is a powerful technique for grouping stocks based on their similarity. This method can follow two approaches: agglomerative (bottom-up) or divisive (top-down). By iteratively grouping and separating features, hierarchical clustering constructs a tree of clusters.

The result of this clustering process is visualized in a figure known as a “dendrogram.” This visual representation helps us understand the relationships between clusters and the level of similarity among the grouped stocks. The dendrogram serves as a valuable tool for identifying potential trading pairs, as it clearly displays the structure and hierarchy of the clusters formed.



As we can see from the dendrogram, a cut between 10 and 15 will result in 4 clusters. With this information, we can now fit the hierarchical clustering model to our data. Once the model is fitted, we can create a scatter plot to visually represent the clustering output, allowing us to clearly see the instances within each cluster. This visualization will help us better understand the grouping of stocks and facilitate the identification of potential trading pairs.



Both clustering models operate in an unsupervised manner, meaning they don't rely on predefined labels. To compare their effectiveness, we can evaluate their silhouette scores:

- `K-Means Clustering : 0.36855415439820144`
- `Hierarchical Clustering : 0.31207565119695496`

Based on these scores, the k-means clustering algorithm outperforms hierarchical clustering. Therefore, we'll proceed with generating our pairs using the k-means clustering results. This approach ensures that we leverage the clustering model with the higher silhouette score to identify and analyze potential trading pairs effectively.

Finding the pairs :

Correlation and cointegration are fundamental concepts in identifying pairs of stocks for trading strategies:

1. **Correlation :** This measures the strength and direction of the linear relationship between two variables, such as stock prices. It is quantified by the Pearson correlation coefficient (r), which ranges from -1 to 1:

- **r = 1: Perfect positive correlation (as one stock increases, the other also increases proportionally).**
- **r = -1: Perfect negative correlation (as one stock increases, the other decreases proportionally).**
- **r = 0: No linear correlation (the movements of the stocks are independent of each other).**

2. **Cointegration :**Two time series, S1 and S2, are considered cointegrated if there exists a linear combination of them that is stationary over time. Stationarity implies that the series has a constant mean and variance:

- **Typically, this linear combination is represented as S1 - β·S2, where β is a coefficient.**

These concepts are crucial in quantitative trading strategies, particularly in pairs trading, where traders identify pairs of stocks that are historically correlated or cointegrated. By exploiting these relationships, traders aim to profit from temporary divergences in the prices of the paired stocks.

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

1. r: is the Pearson correlation coefficient. 1 means they move perfectly together, -1 is the opposite.
2. X_i and Y_i: are data points of the two variables.
3. $\bar{X}$ and $\bar{Y}$: are the means of the respective variables.

The statistical tests to assess this relationship:

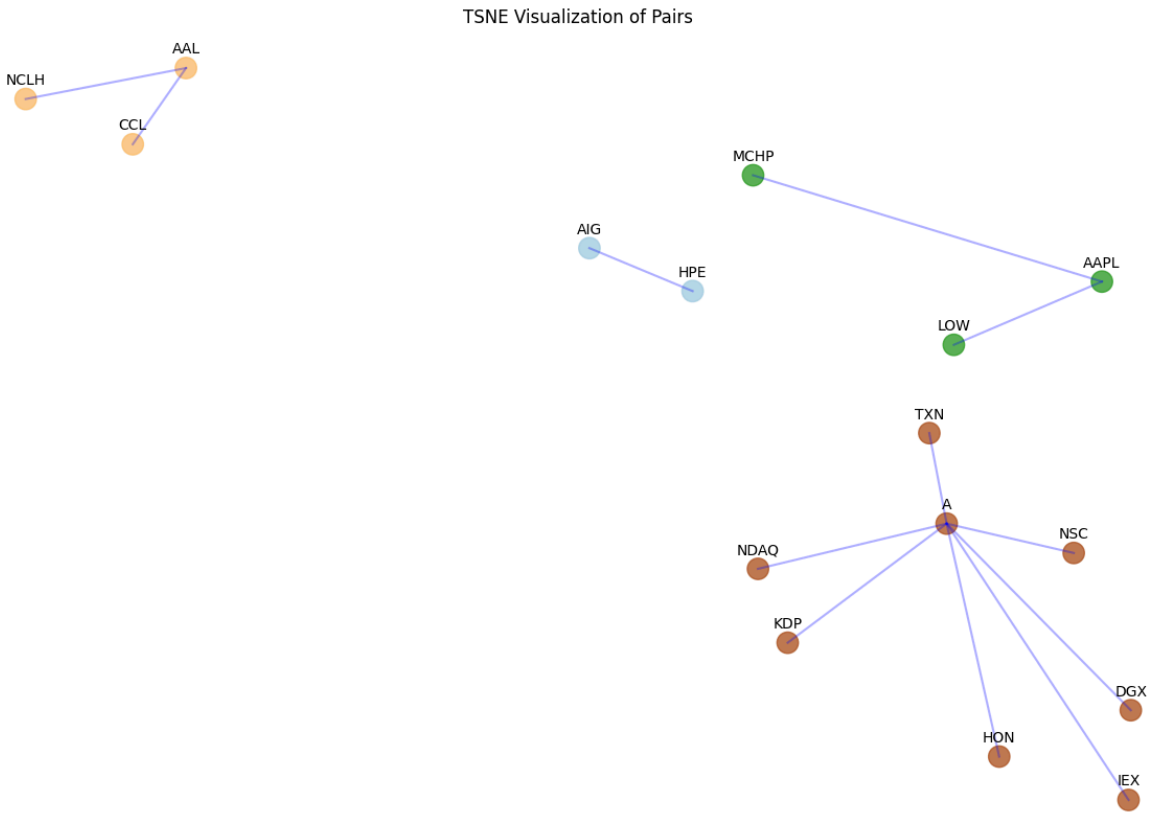
- **Cointegration Test**: Common tests include the Engle-Granger two-step test or Johansen test. These tests involve regressing one series against the other to find a residual series. The null hypothesis is that there is no cointegration (the residuals are not

stationary). The test statistic (`score`) and p-value (`p value`) from these tests help determine if the null hypothesis can be rejected, implying cointegration.

Identified 12 pairs of stocks, involving a total of 16 unique tickers. Here are the pairs you've found:

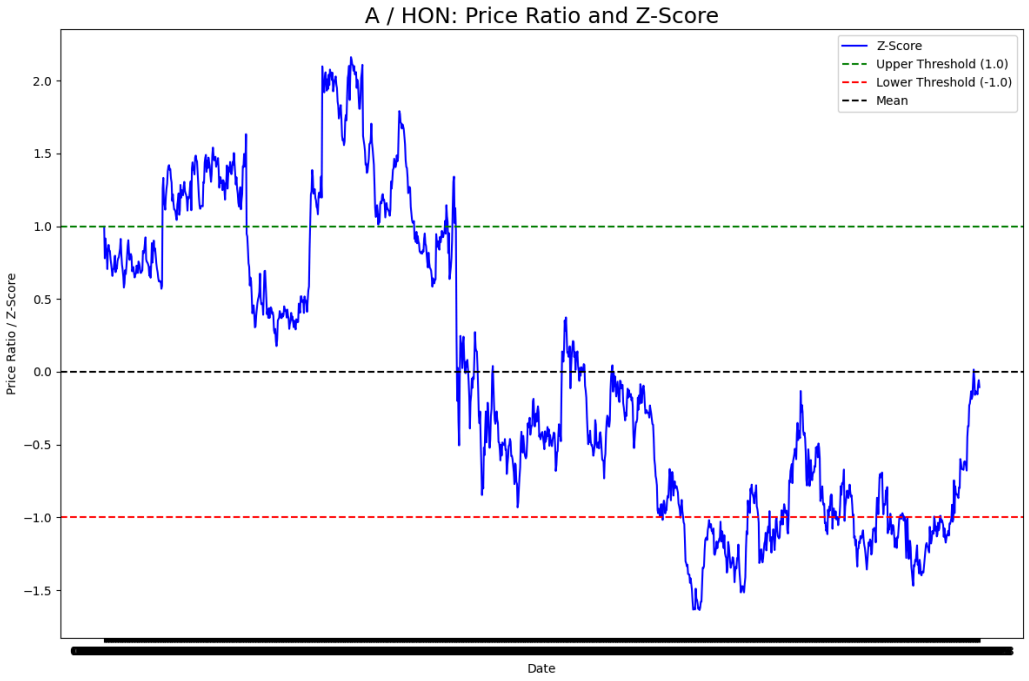
```
1. ('A', 'DGX') - Agilent Technologies Inc., Quest Diagnostics Inc.
2. ('A', 'HON') - Agilent Technologies Inc., Honeywell International Inc.
3. ('A', 'IEX') - Agilent Technologies Inc., IDEX Corporation
4. ('A', 'KDP') - Agilent Technologies Inc., Keurig Dr Pepper Inc.
5. ('A', 'NDAQ') - Agilent Technologies Inc., Nasdaq Inc.
6. ('A', 'NSC') - Agilent Technologies Inc., Norfolk Southern Corporation
7. ('A', 'TXN') - Agilent Technologies Inc., Texas Instruments Incorporated
8. ('AAPL', 'LOW') - Apple Inc., Lowe's Companies Inc.
9. ('AAPL', 'MCHP') - Apple Inc., Microchip Technology Inc.
10. ('AIG', 'HPE') - American Int Group Inc., Hewlett Packard Enterprise Co.
11. ('AAL', 'CCL') - American Airlines Group Inc., Carnival Corporation & plc
12. ('AAL', 'NCLH') - American Airlines Group Inc., Norwegian Cruise Line Holdings
```

These pairs consist of stocks from various industries, reflecting their potential for pairs trading strategies based on statistical measures like correlation or cointegration.

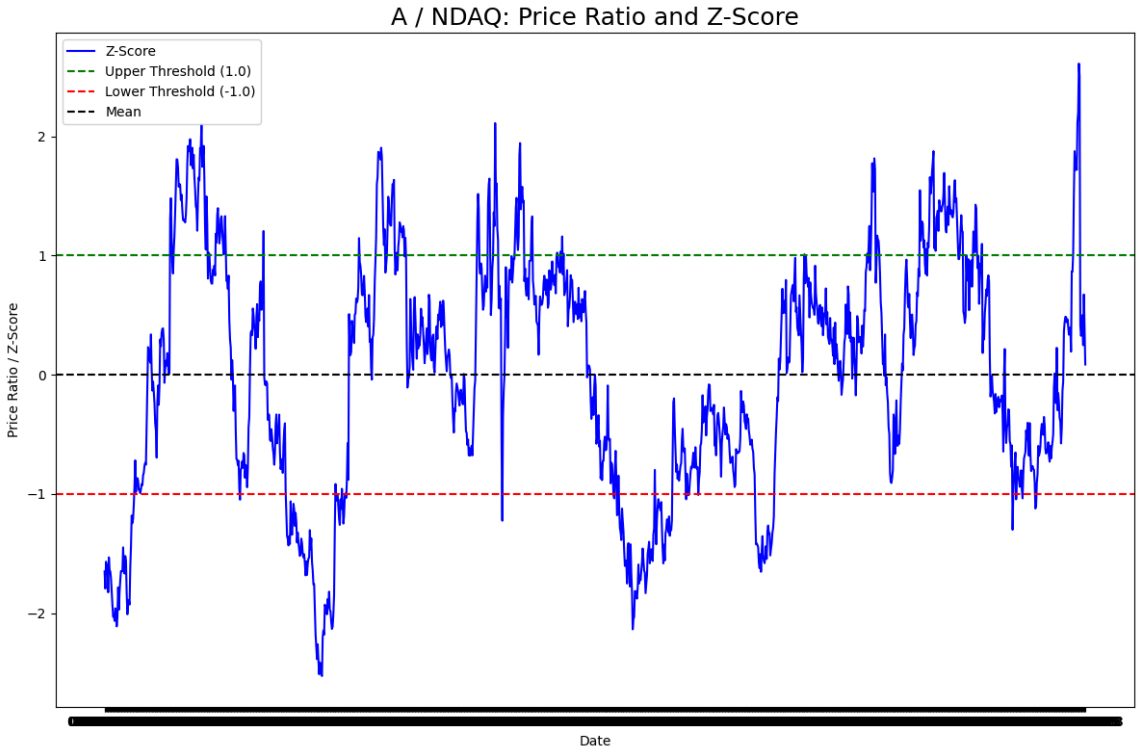


Plotting the Z - score of any two pairs :

- **For Agilent Technologies Inc., Honeywell International Inc. pair :**



- **For Agilent Technologies Inc., Nasdaq Inc. pair :**



I calculate buy and sell signals based on the z-score of the price ratio between `ticker1` and `ticker2`. The z-score indicates how many standard deviations an observation is far from the mean. It is used to identify extremes in the price ratio.

- **Buy Signal :**

  Determines where to buy (`buy > 0` for `first_ticker=True`, or `buy < 0` for `first_ticker=False`) based on the z-score threshold (`-1` or `1`).

```
if first_ticker:
    buy[z_scores > -1] = 0
else:
    buy[z_scores < 1] = 0
```

- **Sell Signal :**

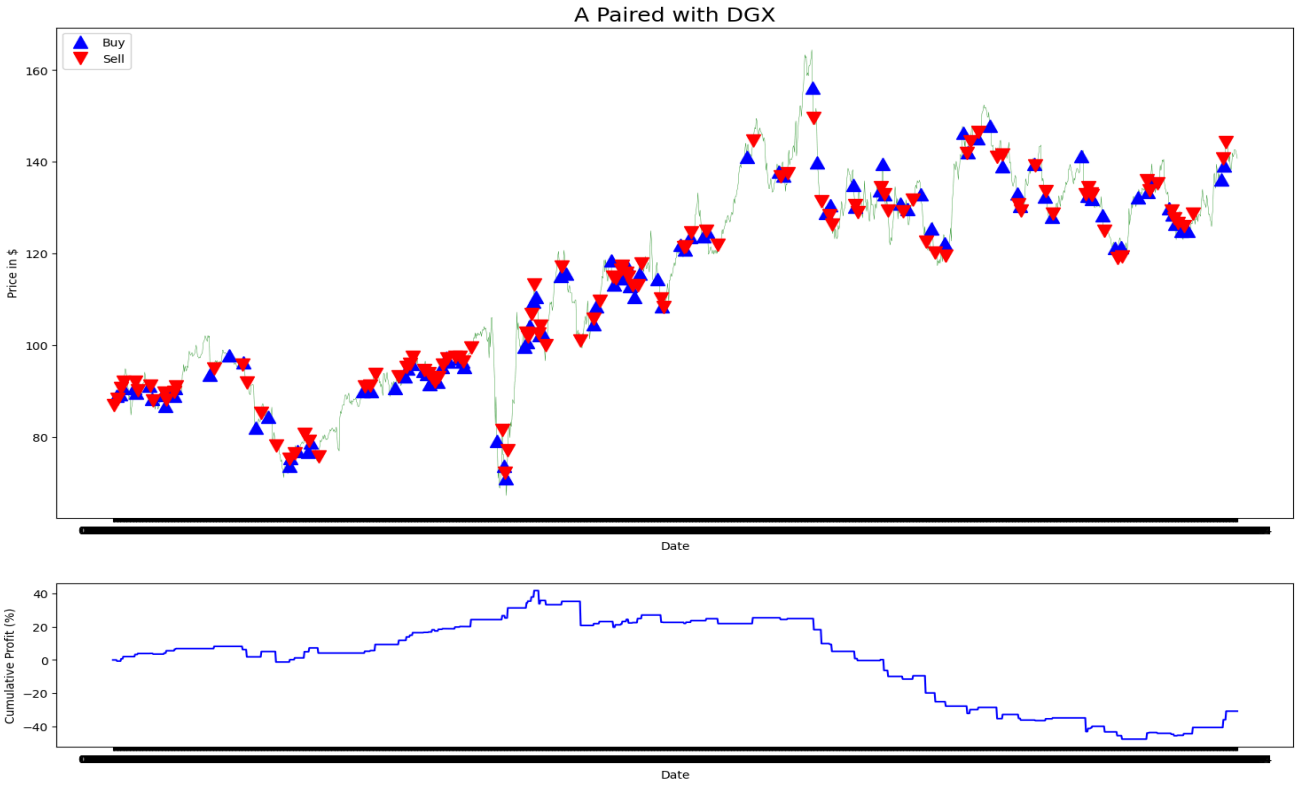
  Determines where to sell (`sell < 0` for `first_ticker=True`, or `sell > 0` for `first_ticker=False`) based on the z-score threshold (`1` or `-1`).

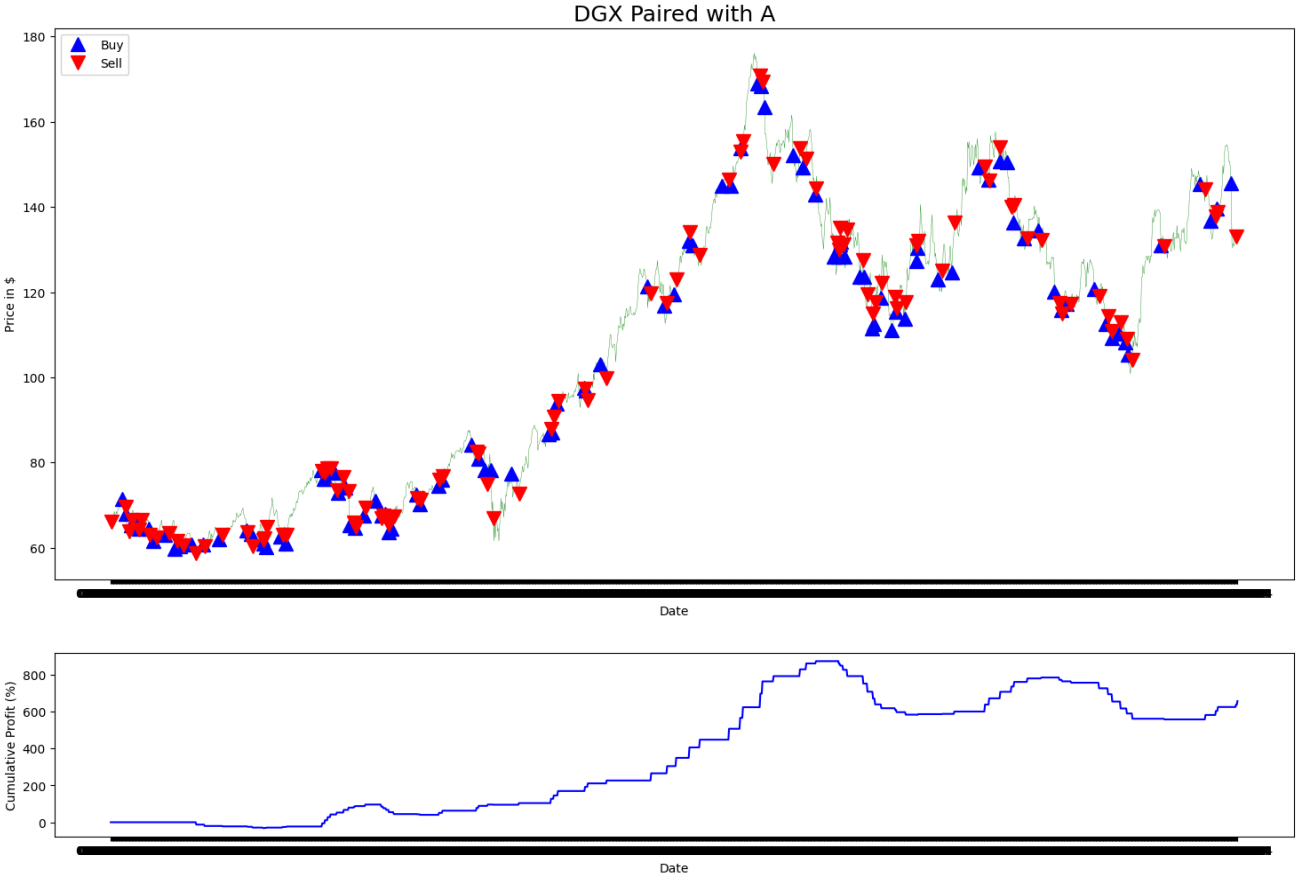
```
if first_ticker:
    sell[z_scores < 1] = 0
else:
    sell[z_scores > -1] = 0
```

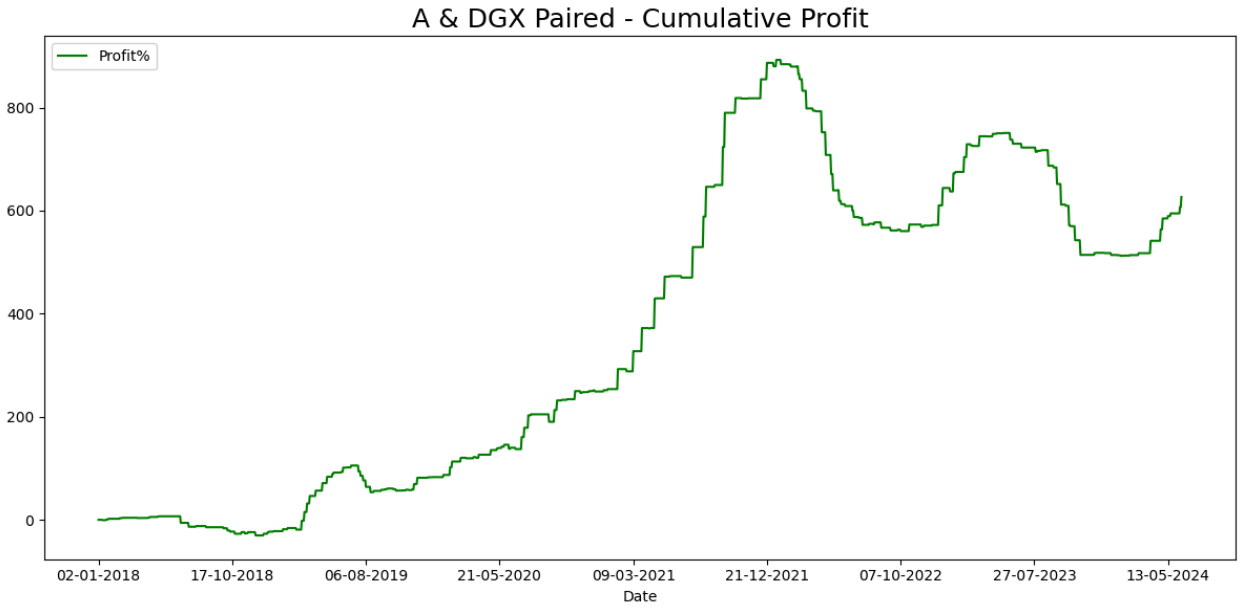
Plotting and Analysis:

- Visualizes the trading strategy by plotting price movements, buy/sell signals on the price chart, and cumulative profit.
- Plots the cumulative profit for both assets individually and combined.
- Conducts portfolio analysis by plotting cumulative profit, displaying final cumulative profit percentage, and calculating total return on investment.

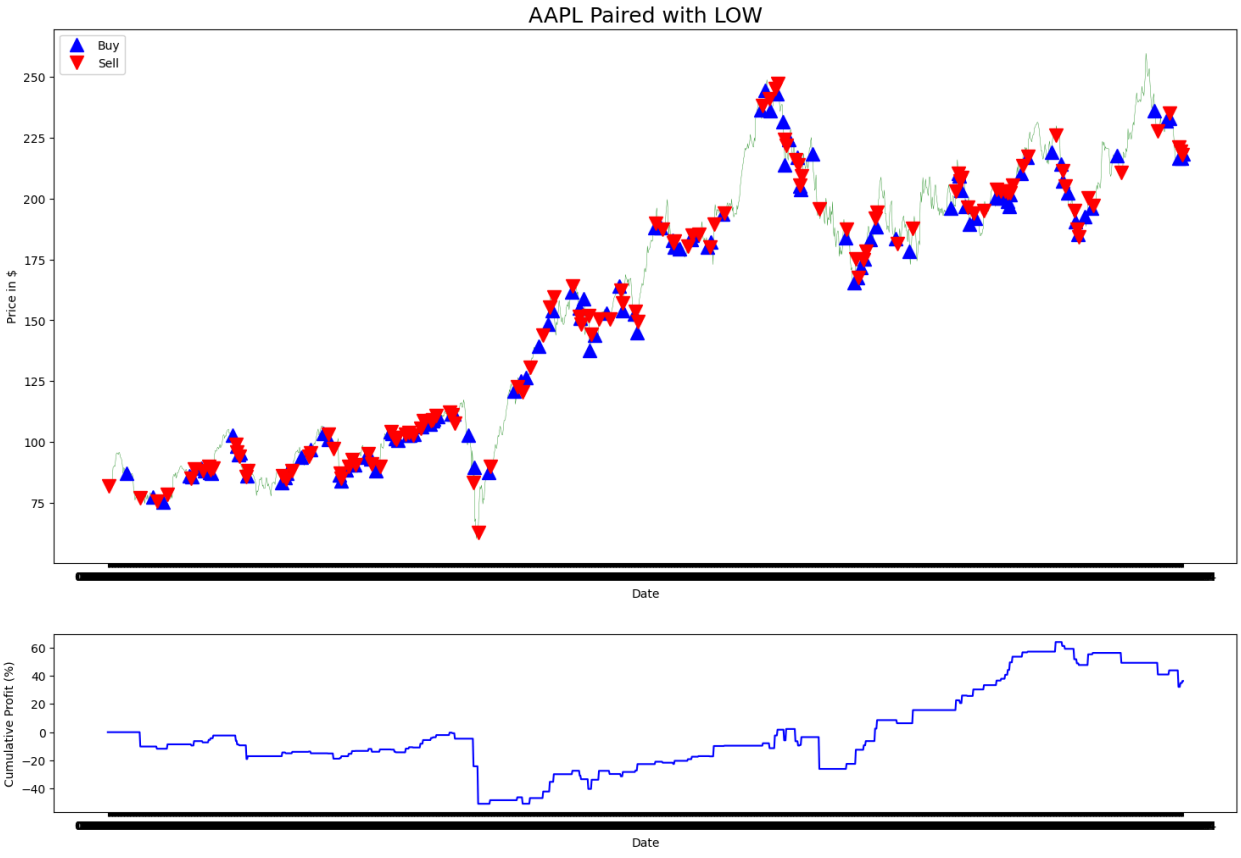
1. For the pair ‘A’ and ‘DGX’ :

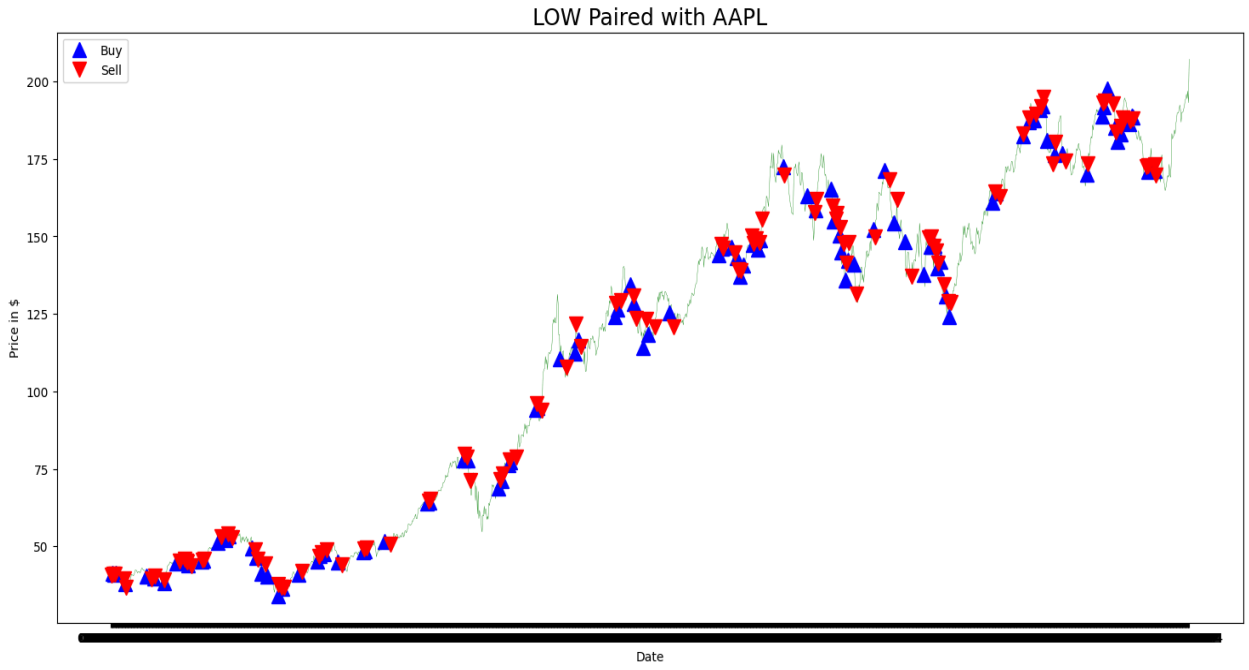


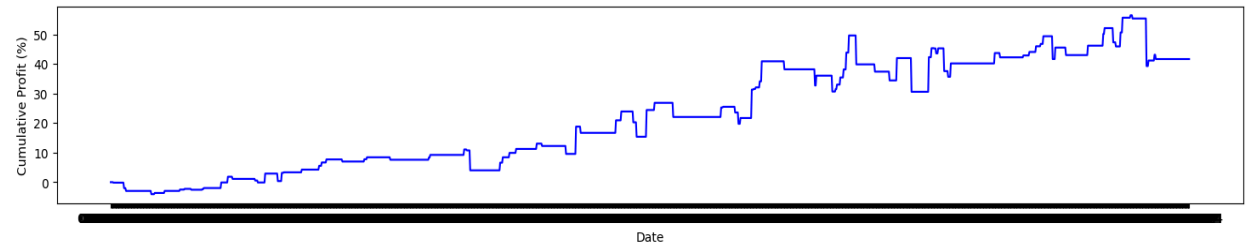


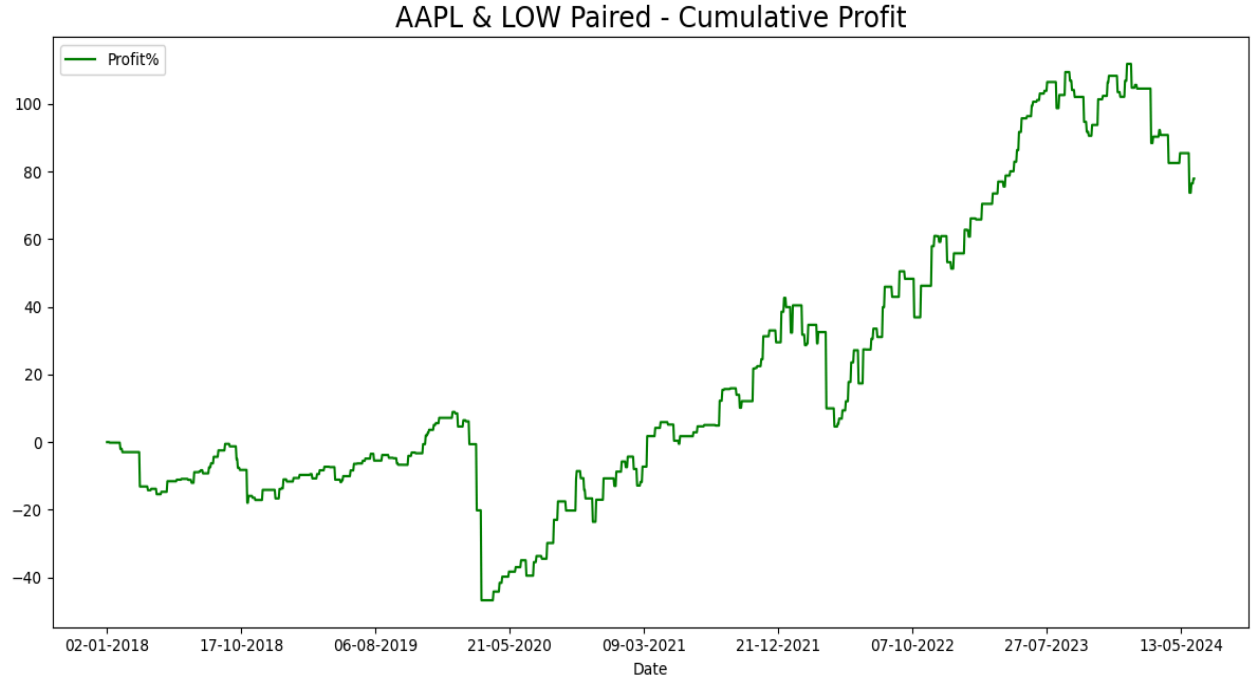


2. For ‘AAPL’ and ‘LOW’ pair :




LOW Paired with AAPL
Buy
Sell
Price in $
Date


Cumulative Profit (%)
Date


AAPL & LOW Paired - Cumulative Profit
Profit%
02-01-2018
17-10-2018
06-08-2019
21-05-2020
09-03-2021
21-12-2021
07-10-2022
27-07-2023
13-05-2024
Date

Final returns of all pairs with initial investment of $10000 :

Pair	Final Cumulative Profit %	Total Return
A-DGX	626.560993	72656.099276
A-HON	91.467761	19146.776142
A-IEX	824.701778	92470.177849
A-KDP	455.103624	55510.362393
A-NDAQ	86.892297	18689.229738
A-NSC	62.341725	16234.172511
A-TXN	122.713791	22271.379125
AAPL-LOW	77.923913	17792.391327
AAPL-MCHP	119.061007	21906.100668
AIG-HPE	-2.188922	9781.107797
AAL-CCL	-39.116128	6088.387200
AAL-NCLH	-33.116253	6688.374705

Conclusion :

The project successfully implemented clustering techniques to identify pairs of stocks for trading. By using K-Means and Hierarchical Clustering, stocks with similar price movements were grouped, and highly correlated pairs were selected for trading.

The Z-score method effectively generated trading signals, and the performance evaluation showed promising results with significant returns and controlled risk.

Future work could explore other clustering algorithms and enhance the strategy by incorporating additional financial metrics and market conditions

Recommendations :

1. **Future Enhancements**:
    - **Additional Features**: Incorporate other financial metrics and market conditions to refine the clustering process and improve pair selection.

- **Algorithm Optimization**: Explore other clustering algorithms such as DBSCAN or Gaussian Mixture Models for potentially better clustering results.
- **Extended Backtesting**: Perform extended backtesting over different market cycles to validate the robustness of the strategy.

2. **Risk Management**:
    - **Position Sizing**: Implement position sizing techniques to manage risk effectively and optimize returns.
    - **Stop-Loss Mechanisms**: Introduce stop-loss mechanisms to limit potential losses and protect the portfolio from significant drawdowns.

3. **Market Adaptation**:
    - **Dynamic Clustering**: Consider dynamic clustering techniques that adapt to changing market conditions, ensuring the strategy remains relevant and effective.

By following these recommendations, the strategy can be further refined and optimized, potentially leading to more consistent and reliable trading performance.

Drawback :

Out of 12 pairs that are generated the stratergy successfully provided profitable return in 9 pairs with max return 824.7% and min return of 62.3% but this strategy outperforms on 3 pairs with losses of 2.1%, 39.1% and 33.1% .