Pairs Trading Strategy

Dharun Suryaa Nagarajan, Rohan Benjamin Varghese
DS5230 - Final Project
Northeastern University - Silicon valley campus
PROJECT GIT REPOSITORY

Introduction

In the dynamic landscape of investment, the pursuit of maximizing returns while minimizing risk stands as the paramount objective. Over the years, a plethora of strategies and techniques have been devised to navigate the complex terrain of financial markets. Among these, the 'Pairs-Trading' strategy has emerged as a prominent tool in the arsenal of modern hedge funds, renowned for its simplicity and market-neutral attributes. Termed as a form of statistical arbitrage, pairs trading hinges on the meticulous observation of correlation dynamics between pairs of stocks known to exhibit a historical relationship. By strategically opening long and short positions based on this historical behavior, investors aim to capitalize on the convergence of prices over time. Crucially, the identification of these correlated pairs is facilitated by unsupervised learning algorithms, adding a layer of sophistication to the execution of this strategy. This project employs advanced clustering techniques such as K-Means and DBSCAN to identify these pairs effectively.Through rigorous evaluation, we aim to determine the superior algorithm for identifying correlated stock pairs and thereby enhance the precision and efficiency of pairs trading, ultimately optimizing investment outcomes in today's dynamic financial landscape

1 Methods

1.1 Pre-processing

In our next stage, we want to pre-select eligible stocks that enable us to sail through further steps. First, we removed stocks that were de-listed, exchanged, or merged during our sample period since those stocks are no longer in market. Next, we removed stocks that have negative prices which will be problematic for further analysis. Stocks that are constantly trading at-low-volume also have to be removed since improper trading executions can largely change their stock prices and altered history. Finally, we remove stocks that have more than half missing prices, so that we have enough available data for imputation. A similar approach was performed on the financial fundamentals of datasets. In the end, there are 1795 eligible stocks for further analysis.

In this step, we imputed the missing values in our pre-processed dataset. We worked with the time series data and the financial ratios separately. We imputed both of them using means, although in a slightly different way. For the time series data of stock prices, missing values were replaced by the mean of all the available stock prices for that stock in the training period. Since the financial ratios individually have different bounds[3] we imputed missing values in the financial ratios dataset with the average of all available data for the particular ratio.

1.2 Dimensionality Reduction

There are several non-linear dimensionality reduction algorithms like PCA, t-Distributed Stochastic Neighbor Embedding (t-SNE), CAE (Convolution AutoEncoder)[1]. In our approach, we tried to experiment with both them but since the stock data is linear[2] in nature we went with PCA for the below reasons.
Given the context 1500+ daily data points, individual stocks as rows, and stock prices as data points, PCA is likely a good starting point for dimensionality reduction for the following reasons. Linear Nature of Stock Data- Stock prices often have linear or quasi-linear trends over time[3]. PCA can capture key components that explain the most variance in the data. Interpretability - PCA provides interpretable components, allowing you to understand which dates or periods contribute most to variance. Simplicity and Efficiency - PCA is computationally efficient and easy to implement, allowing you to quickly reduce dimensionality and perform further analysis or visualization.
PCA was performed independently on the time series stock price data and the financial ratios. After PCA, the time series data is reduced to 15 principal components and the financial ratios are reduced to 5 principal components. We retained more than 99% of the variance in either case. Here are two plots illustrating the proportion of variance captured by the top singular values:

1.3 K-means clustering

K-means is a partition-based clustering algorithm that assigns data points to clusters by minimizing

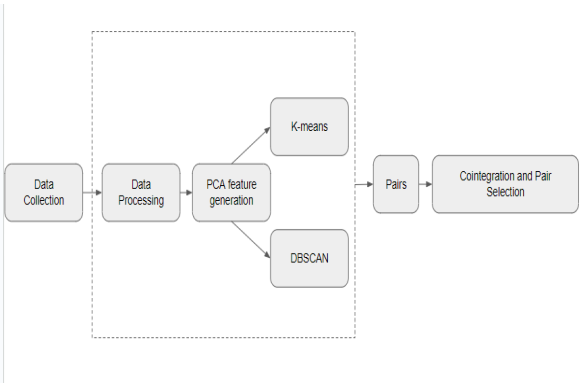

Figure 1: Approach

within-cluster variance. The steps are as follows: Initialization: To select the K initial data points, we used three scoring methods Distortion Score, Silhouette Score, Calinski Harabz Score. From them the max number of clusters given was 10 so we chose 10 as k. Each data point was assigned to the nearest centroid based on Euclidean distance:

$$d(x, c) = \sqrt{\sum_{j=1}^{m}(x_j - c_j)^2}$$

Recalculated the centroids based on the new assignments. This process was repeated until the centroids no longer changed. To determine potential pairs for trading, we identified securities within each cluster, focusing on those with the closest distances to the centroids.

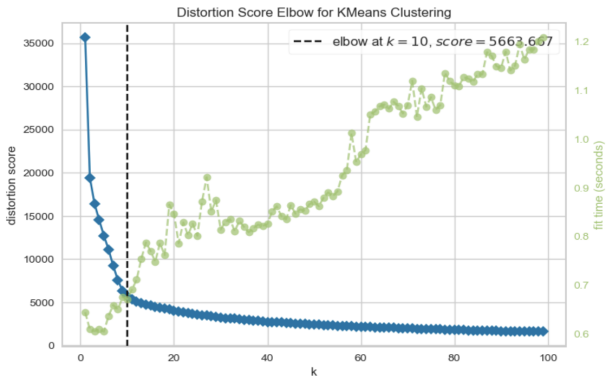

Figure 2: Distortion Score

1.4 DBSCAN clustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm that groups data based on the concept of density connectivity. The key parameters are ϵ- The maximum distance between two points for them to be considered in the same cluster and $minPts$- The minimum number of points required to form a dense region. A point is a "core point" if it has at least $minPts$ points within distance ϵ. if $|N(x, \epsilon)| = y|d(x, y) > minPts$, the point is considered a core point and forms the basis of a new cluster. The clustering process involves recursively adding points within distance ϵ to form connected clusters. Points that do not meet the density requirements are labeled as noise.

After clustering, we identified potential pairs for trading by examining securities within the same cluster. Unlike K-means, DBSCAN can form clusters with varying shapes, making it suitable for complex data distributions.

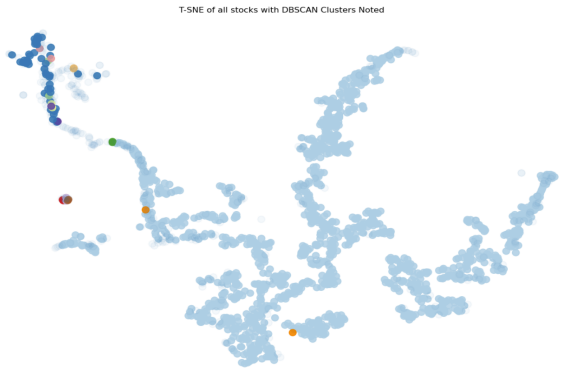

Figure 3: DBSCAN T-SNE

1.5 Cointegration and Pair Selection after DBSCAN

The key of finding valid pairs is to find the cointegration of two selecting stocks. As we will go in detail later, we want to find two stocks that their time series of prices follows a linear relationship but not always. The spread of two selecting stocks should be a mean-reverting process, meaning that their spread tends to drift towards its mean function over time.

To find such pairs, ADF test is performed(or Augmented Dicky Fuller Test) to every pairs in each clusters to find co-integrated pairs. ADF test is usually used in time series analysis. In this case, ADF test helps us determine whether the spread of two stocks is stationary or not. A stationary process is very valuable to model Pairs Trading strategies. For instance, in this case, if the spread is stationary, we know that the difference in their stock process will drift to the mean (which is zero in our case) over time if it is temporarily derailed, and this is the time window for us to make money.

2 Data set

2.1 Data Source

The datasets are provided by Wharton Research Data Services (WRDS). We mainly obtained the daily stock

files from file from CRSP and quarterly fundamentals from Compustats for our purpose. Initially, our dataset consists of stock price files from 3000 stocks. Those stocks value and size are large enough to restore the whole market value, representing approximately 95% of the total market shares. We performed this pre-screening process to avoid the 'small-cap' trap in the market. Currently, there are more than 6000 active stocks in the U.S. Stock Market but most of them are micro-valued. In reality, investors often cautiously avoid investing in those stocks, since trading, even a small number of shares might have unpredictable effects on their stock prices. We set the sample period from 2010-01-01 to 2015-12-31 for training strategies and use sample period 2016-01-01 to 2019-12-31 for back-testing.

2.2 Additional Datasets and Applications

IBES (Institutional Brokers' Estimate System): Provides analyst estimates, recommendations, and earnings forecasts. Used for understanding market sentiment and analyst behavior.[2] Y-finance (Trade and Quote Data): Offers detailed intra-day trading data, which can be used to study market micro structure, trading patterns, and liquidity.[3] Alternative Datasets (e.g., Social Media Sentiment, News Data)[2]: Emerging datasets that offer insights into market sentiment and external influences on stock performance. All of the above mentioned datasets have been used in various other applications like Financial Large Language Models, Backtesting Trading Strategies, Risk Assessment and Portfolio Management, etc.

```
Data columns (total 18 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   permno          3666 non-null   int64
 1   bm              3666 non-null   float64
 2   pe_exi          3666 non-null   float64
 3   npm             3666 non-null   float64
 4   cfm             3666 non-null   float64
 5   roa             3666 non-null   float64
 6   roe             3666 non-null   float64
 7   GProf           3666 non-null   float64
 8   totdebt_invcap  3666 non-null   float64
 9   capital_ratio   3666 non-null   float64
 10  debt_ebitda     3666 non-null   float64
 11  lt_debt         3666 non-null   float64
 12  cash_debt       3666 non-null   float64
 13  fcf_ocf         3666 non-null   float64
 14  de_ratio        3666 non-null   float64
 15  at_turn         3666 non-null   float64
 16  staff_sale      3666 non-null   float64
 17  ptb             3666 non-null   float64
```

Figure 4: Data Schema

3 Results

3.1 K-means Clustering Results

In Figure 2 we see that the distortion score gave a k=10, whereas the Silhouette Score, Calinski Harabz Score gave k=2 as the elbow result. Smaller clusters lead to bigger cluster density so in-order to increase decrease the cluster density to focus on good pairs K-means parameter was give a K value of 10. Using the pairs given out by clusters, the next was to test their linearity visually (fig 5 and fig 6) as well as using scores like Silhouette Score, DBI (Davies-Bouldin index).

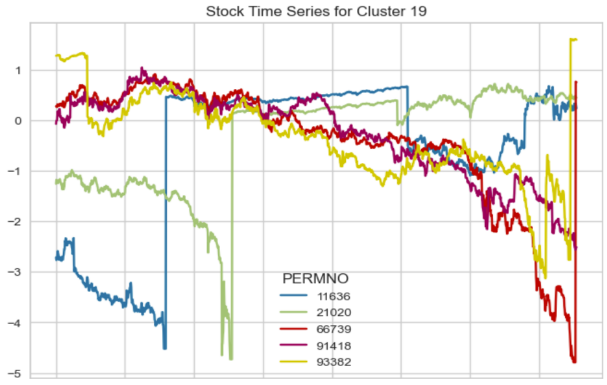

Figure 5: K-means cluster

3.2 DBSCAN Clustering Results

The DBSCAN algorithm was paramterized by eps = 1 and minPoints = 2 which resulted in the formation of 12 clusters. A simple visualization of the cluster in the form of a T-SNE plot is shown in fig 3. The results of the DBSCAN looked more in correlation visually and the evaluations metrics proved the same.

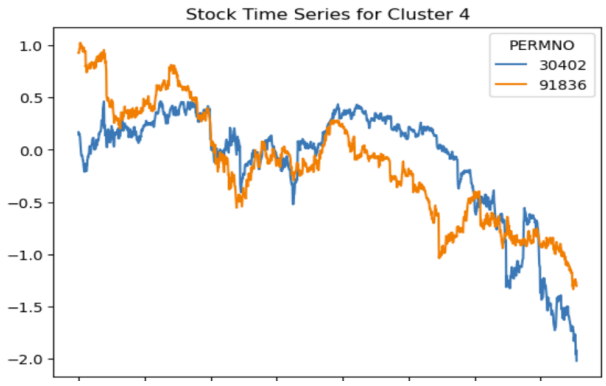

Figure 6: DBSCAN cluster

3.3 Comparison and Pair Selection

With Higher dimensions Silhouette Score does not give the true clustering efficiency so DBI was chosen as another metric to test the efficiency on. With 55 dimensions both has low Silhouette Score but DBSCAN performed better relative to DBI. In lower dimensions(15 columns) the Silhouette Score, DBI was good for DBSCAN. This led the next stage to co-integration using clusters generated by DBSCAN. The results of the positive ADF test results are shown in Fig 8.

Metrics	K- Means	DBSCAN
Silhouette Score (closer to 1 is better)	-0.402	0.529
Davies Bouldin Index(DBI) (closer to 0 is better)	3.66	0.738

Figure 7: Result Camparison

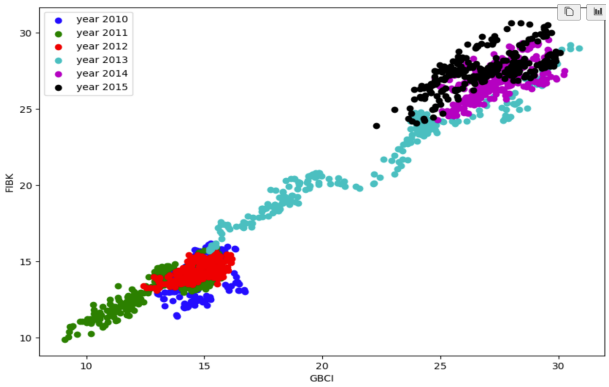

Figure 8: Co-integration result

4 Discussion

In this project, we explored two different clustering algorithms K-means and DBSCAN. These are foundational models that are equipped for fast clustering. Other advanced clustering techniques, like Gaussian Mixture Models (GMM), hierarchical clustering, or deep learning-based approaches (such as autoencoders for clustering), offer more flexibility and adaptability, particularly in complex datasets or high-dimensional spaces.

Agglomerative Clustering was used in [1] but we tried k-means and DBSCAN and as a result DBSCAN did show better performance. The performance of Agglomerative clustering was not mentioned, the DBSCAN model gave 875 pairs compared to 497[1] of Agglomerative technique.

Firm characteristics are revealed to be an important source of information in identifying pairs.Clustering stocks based on their characteristics as well as past returns significantly reduces the volatility and downside risk of the strategies and improves performance.Each cluster consists of only one or few sectors, which implies that stocks in the same industry tend to behave in a similar manner.

PCA and Auto-encoders were explored as different techniques. Since the data was quasi-linear we went with PCA as final approach but with more non-linear information adding value to pairs-trading, CAE could perform better in those cases [1]. Co-integration was tried out as a reference from [3] but a more complex method proposed in [1] can create RL based trading strategy that shows better risk management and returns too.

The next step would be to handle risks by doing rigor backtesting on the training data, stress testing and continuous monitoring. Include more features technical and fundamental that could improve clustering and help incorporate CAE, and RL based approaches to better the strategy. Compare the performance with the market(S & P index) to realize the market efficiency.

5 References

[1] Roychoudhury, Raktim and Bhagtani, Rahul and Daftari, Aditya, Pairs Trading Using Clustering and Deep Reinforcement Learning (July 9, 2023).

[2] Shihao Gu, Bryan Kelly, Dacheng Xiu, Empirical Asset Pricing via Machine Learning (May 5, 2020).

[3] Chulwoo Han, Zhaodong He, Alenson Jun Wei Toh, Pairs trading via unsupervised learning (September 28, 2022)

6 Appendix

6.1 Firm Characteristics

1. PERMNO - permanent identification number
2. BM - Book/Market
3. PE-EXI - Price to Earnings Ratio (Diluted, Excluding Extraordinary Items)
4. NPM - Net Profit Margin
5. CFM - Cash Flow Margin
6. ROA - Return on Assets
7. ROE - Return on Equity
8. GProf - GNU Profile
9. Capital Ratio - Capital Ratio
10. debt-ebitda - Debt-to-EBITDA (Earnings Before Interest, Taxes, Depreciation, and Amortization)
11. lt-debt - Long-Term Debt
12. cash-debt - Cash to Debt
13. fcf-ocf - Free Cash Flow to Operating Cash Flow
14. de-ratio - Debt-to-Equity Ratio

15. at-turn - Asset Turnover
16. staff-sale - Staff-to-Sales Ratio
17. ptb - Price-to-Book Ratio

6.2 Github Repository

Code - https://github.com/dharun4772/Pairs-Trading-Strategy

Plots - https://github.com/dharun4772/Pairs-Trading-Strategy/tree/main/Picture

7 Statement of contributions

In this project, we, Dharun Suryaa Nagarjan and Rohan Benjamin Varghese, collaborated closely to leverage our individual strengths and expertise towards achieving our common goals. Dharun spearheaded the implementation of the K-means clustering algorithm, investing significant effort in understanding its nuances and fine-tuning parameters to extract meaningful insights from the data. Concurrently, Rohan took charge of implementing and optimizing the DBSCAN algorithm, exploring its applicability and effectiveness in our specific context.

Beyond our individual responsibilities, we worked hand-in-hand for the data retrieval and preprocessing tasks. Moreover, our collaboration extended to the documentation phase, where we meticulously recorded our methodologies, experimental setups, and results. Ultimately, this paper reflects the culmination of our joint endeavors, showcasing not only our individual contributions but also the strength of our collaborative partnership.