statXarb: Statistical Arbitrage using Clustering for Market Decisions

Deniz Qian
dq2024@nyu.edu

Henry Ying
hy1672@nyu.edu

Chynna Hernandez
ch4262@nyu.edu

Shreya Guda
sg7999@nyu.edu

Abstract

In the dynamic realm of financial markets, banks, and other financial institutions actively engage in both long and short positions on financial instruments, seeking to leverage pricing inefficiencies for profit. These pricing inefficiencies arise from a complex interplay of factors such as market sentiment, economic indicators, geopolitical events, and inter-market relationships. A deep understanding of these dynamics and the ability to predict market trends not only provides a significant competitive edge but also enhances these institutions' profitability and risk management strategies.

The overarching goal of this project is to develop a sophisticated predictive model that forecasts stock market trends and uncovers potential inefficiencies in security pricing through the lens of statistical arbitrage. This model aims to serve as a pivotal tool for financial institutions, empowering them with the insights needed to make informed trading decisions, thereby optimizing profitability while minimizing risk.

1 Introduction

1.1 Background

In the modern financial landscape, statistical arbitrage strategies have evolved to incorporate advanced machine learning and artificial intelligence techniques. The increasing availability of high-frequency trading data, coupled with advancements in computational power, has enabled the development of more sophisticated models capable of processing vast amounts of information in real-time. These models not only enhance the accuracy of predictions but also allow for the continuous adaptation to changing market conditions.

The importance of statistical arbitrage extends beyond mere profitability. It plays a crucial role in market efficiency by correcting mispricings. Moreover, the ability to accurately predict market trends and identify arbitrage opportunities provides financial institutions with a significant competitive edge, enhancing their risk management capabilities and overall financial stability.

This paper builds on the foundational principles of statistical arbitrage by leveraging modern machine learning techniques to automate the decision-making process in forming arbitrage groups. By utilizing a large dataset of NYSE trades and quotes, and incorporating methods such as stock embeddings, this research aims to advance the field of quantitative trading and provide new insights into the dynamics of financial markets.

1.2 Related Works

Developing statistical arbitrage models is not a novel idea. Numerous researchers have explored and contributed to the advancement of these models, leveraging various techniques and datasets to enhance predictive accuracy and trading performance.

For instance, Krauss, Do, and Huck (2017) explored the application of deep learning algorithms to enhance the predictive power of statistical arbitrage strategies. By utilizing neural networks and other advanced machine learning methods, their research demonstrated improved accuracy in identifying arbitrage opportunities compared to traditional approaches (Krauss et al., 2017).

Recent advancements in natural language processing (NLP) have further enriched the field of statistical arbitrage. Research by Wang et al. (2020) introduced Stock2Vec, an approach that leverages NLP techniques to capture the semantic relationships between stocks. By embedding stock data into a continuous vector space, this method facilitates the identification of similar stocks and enhances the clustering process, ultimately improving the effectiveness of arbitrage strategies (Wang et al., 2020).

Correlation matrices are the most common and widely used approach in statistical arbitrage to understand the relationships between financial instruments. These matrices are pivotal in portfolio management, particularly for constructing statistical arbitrage strategies. In their innovative study, Cartea et al. (2023) apply graph clustering algorithms to the market residual returns correlation matrix to enhance statistical arbitrage portfolios. They effectively group stocks with similar price movements into clusters, facilitating the creation of portfolios that exploit mean-reversion within these clusters. Their approach generates portfolios with notable annualized returns and robust Sharpe ratios by employing several clustering methods, including Spectral and SPONGE clustering. This method showcases the importance of correlation matrices in identifying profitable trading opportunities and advances portfolio management beyond traditional industry classification-based strategies. (Álvaro Cartea et al., 2023).

In summary, the literature on statistical arbitrage encompasses various methodologies and applications. Previous research has extensively utilized correlation matrices to identify relationships between financial instruments and neural networks to enhance predictive power. This project explores the effectiveness of newer techniques, such as embeddings, in improving the automation and accuracy of decision-making processes in statistical arbitrage. By incorporating a large dataset of NYSE trades and quotes, we seek to determine whether these modern approaches can provide significant advancements over traditional methods.

1.3 Overview

Specifically, this project aims to explore the ways to automate the decision making process as it relates to forming statistical arbitrage groups for financial models.

While this idea has been employed before, our project differs in that we used a rather large dataset: 41 files of NYSE trades and quotes data. Additionally, we used modern techniques such as stock embeddings to explore other ways to automate statistical arbitrage groups using machine learning techniques.

1.4 NYSE TAQ files

The NYSE TAQ data consists of all trades and quotes for all issues listed and traded on US regulated exchanges for a single trading day (New York Stock Exchange, 2023). For the purposes of our project, the following columns were processed: Time, Exchange, Symbol Trade Volume, Trade Price.

2 Methods and Implementation

2.1 Data preprocessing

As part of document preprocessing, we extracted 9580 unique symbols from the NYSE TAQ files from January 1, 2024 to Februrary 29, 2024.

2.1.1 Security processing

To extract the ticker symbols from the data, pandas dataframes were applied to obtain the list of unique symbols (McKinney et al., 2020). Only securities with respective sector labels were chosen. Using the capital IQ excel plugin, we extracted the sector labels for each of the respective securities: Communication Services, Consumer Staples, Consumer Discretionary, Energy, Financials, Health Care, Industrials, Information Technology, Materials, Real Estate and Utilies.

2.1.2 Time stamp processing

The NYSE TAQ Time and Participant Timestamps are fixed length field of ASCII numeric characters in the following format: HHMMSSxxxxxxxxx where HHMMSS represents two-digit values for hours, minutes and seconds since midnight, and the 9 x's represent a nine-digit nanosecond value. With this in mind, we extracted the hour, minute, seconds and nanoseconds respectively from the files as part of preprocessing. We also extracted additional time related features such as the month, day and year from the file name.

2.1.3 Normalization

For normalization, the trade price and trade volume were log scaled. Previous research showed that using log scaling method generally had better results for both price and count data(Nicholas Venuti and Forbes, 2017). As a result, we applied a moving average using a window of 50 and applied log scaling using the numpy library (Harris et al., 2020).

2.2 Feature engineering

Features such as Volume Price Ratio, Trade Volume and the hour of the trade were observed. With this in mind, the Volume Price Ratio and hour were engineered to optimize the time series data.

Volume Price Ratios are used to help determine the balance between a stocks demand and supply (Investopedia, 2022). Here, we added the Volume Price Ratio as a feature by dividing the Trade Volume and Trade Price columns and log scaling was applied. A high volume price ratio can indicate strong interest or sentiment in a stock at its current price, potentially signaling an impending price movement. This is useful in statistical arbitrage which often relies on accurately predicting short-term price movements.

Furthermore, trade volume was used because trade volumes can indicate liquidity and active interest in a stock. This is essential for executing statistical arbitrage strategies that often require quick movement to and from the market. Volume can also reflect the impact of institutional trading, which can drive significant price movements.

Cyclical encoding was also used for the time related data. Here, we extracted the hour and applied sine and cosine functions on the hour the order transcribed. Given the periodic nature of time related data, we opted to use the hour as a feature to generate embeddings (Lewinson, 2024). Financial markets often show different behaviors at various times, such as at market open or close, or during specific hours when trading volumes peak. This is particularly useful in statistical arbitrage, where the timing of trades can be as crucial as the selection of securities.

2.3 Building the embedding model

The model is designed with a multi-layer neural network specifically tailored to handle a large number of symbols effectively. It starts with an input layer that receives single integers, each representing a symbol index. This is crucial for managing categorical data in neural networks. Following the input layer, an embedding layer maps each integer to a dense vector of predefined size, facilitating a more nuanced representation of the symbols. The embedding layer is critical as it allows the model to learn an optimal representation for each symbol based on the network's subsequent feedback. The total number of unique symbols, plus one for zero indexing, sets the input dimension of this layer, ensuring all possible symbols are accounted for.

After the embedding layer, the output is flattened. This step is necessary to convert the multi-dimensional embedding outputs into a flat vector that can be fed into subsequent dense layers, bridging the gap between categorical input and continuous processing layers.

The network includes multiple dense layers with L2 regularization. L2 regularization is employed to minimize the complexity of the model by penalizing the square values of the parameters, effectively preventing overfitting by discouraging large weights (Contributors, 2023). Each dense layer is followed by a LeakyReLU activation function, which introduces non-linearity to the model's learning process. Unlike standard ReLU, LeakyReLU allows a small, negative gradient (i.e. 0.01 in our case) when the input is less than zero, which helps maintain gradient flow during training and can improve model performance on problems where activation outputs are often zero.

Dropout layers are strategically placed after each dense layer to randomly ignore a set percentage of neurons during training. This randomness helps to break up coincidental patterns that are not useful in generalizing the data, further supporting the model's robustness against overfitting.

These features combine to form a powerful model capable of learning complex, high-dimensional mappings from symbolic data to useful embeddings, which can then be applied in various tasks requiring nuanced interpretation of input symbols.

2.4 Training the embedding model

We trained the embedding model on 4 features: volume price ratio, trade volume, hour sin, and hour cos. These features were chosen because they helped to capture both market sentiment and behavioral patterns in trading data, both of which are essential for developing robust statistical arbitrage strategies.

Initially, a combination of both dense layers and LSTM layers were used to take advantage of the sequential data. However, LSTM proved to be memory intensive alternative methods were explored, namely early stopping and reduce on learning plateau.

Early stopping is a mechanism that hastens the neural network. When it is noticed that the model stops learning before all epochs are finished, it will stop the model from learning for the remaining epochs. This helped our code because algorithm to halt whenever overfitting begins to occur (Goodfellow et al., 2016).

Reduce on learning plateau is another way we can optimize our algorithm. This means we decrease the learning rate once the loss function stops going down to attempt to get even closer to the optimum. The scheduler reads the metric quantity and decides to decrease the learning rate if no im-

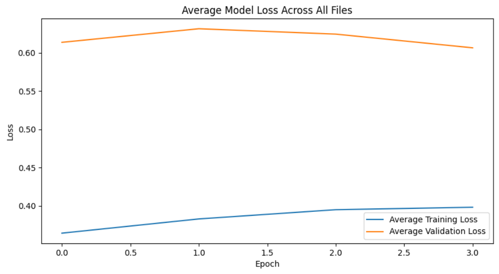

Figure 1: Average training and validation loss

provement is seen for a number of epochs, known as patience (PyTorch, 2024). This works because we eventually take lower and lower steps so that our improvement can be as precise as possible. The end result after using these techniques is a model that minimizes the loss more than our initial model with LSTM.

Given the large scale of our data, dimensionality reduction played a crucial role in our model understanding the financial trading data. Specifically, the focus was on two scaled features: volume price ratios and logarithmically scaled trade volume. After normalizing these features using the StandardScaler, PCA was applied to transform and reduce the dimensionality of the scaled data to two principal components. The purpose of this was to capture the most significant variance present in the features with fewer dimensions, which helps in simplifying the dataset while retaining essential information.

3 Evaluation metrics

3.1 Measuring the loss

The loss per day was recorded after the embeddings were processed. After reviewing the average training history over time, our model revealed that the average training loss decreased as the number of epochs increased. This behavior tells that the model is learning from the training data over time. In contrast, the average validation loss increases slightly as the epochs progress. This divergence between training and validation loss is suggestive of overfitting. Thus, the model might learning to perform well on the training data but not generalizing well to new data represented by the validation set.

3.2 Clustering

Once the embeddings are obtained, they were then clustered. For clustering, we simply compared the Euclidean distance between the embeddings. There were two types we considered: Spectral Clustering, and K-Means Clustering. Each clustering method for several different values of k, and found that K-Means Clustering overall performed better. The elbow method was used to determine the number of clusters in the dataset. This method showed that the optimal number of clusters was 4.

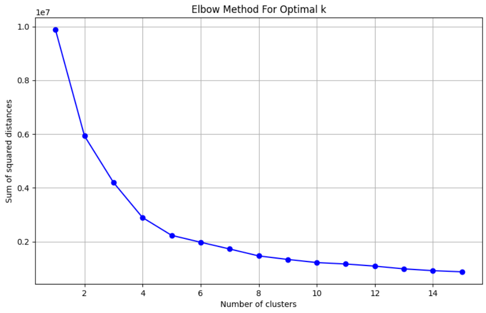

Figure 2: The elbow method was used to determine the number of clusters in the dataset. The following shows the sum of squared distances of samples to their nearest cluster center for different numbers of clusters. There is an observable bend around 4 clusters, after which the rate of decrease in sum of squared distances becomes gradual.

3.2.1 Cluster purity

The purity for each cluster is calculated by dividing the size of the largest sector by the total size of the cluster. This gives a measure of how much the largest sector dominates the cluster. The purity value ranges from 0 to 1, where 1 indicates that the cluster contains only items from a single sector, and lower values indicate a more mixed cluster. The final cluster purity score was 0.4257. A purity score of 0.4257 indicates that, on average, about 42.57% of the members of each cluster belong to the most common class within that cluster. Although this suggests some homogeneity, it also indicates that the clusters contain a significant amount of overlapping classes or are not perfectly homogeneous.

To help reduce the dimensionality of the embeddings we used t-sne to help visualize the data. The purple cluster is a little big and pretty close to the rest of the clusters, which is why our silhouette score and cluster purity are in the middle. However, the other three clusters are very well defined and separated. T-sne preserves each data point's feature and location relative to the other data, so it is a very good way of visualizing high dimensional vectors.

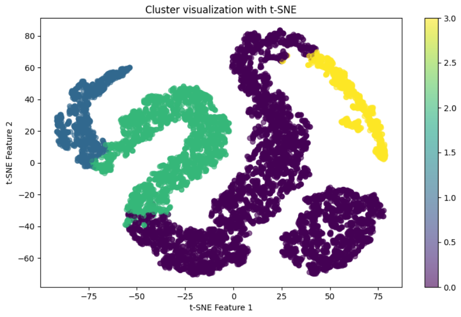

Figure 3: Each cluster in the t-SNE plot refers to a group of financial symbols that share similar trading patterns related to trade volume, volume price ratio and the hour of the trade

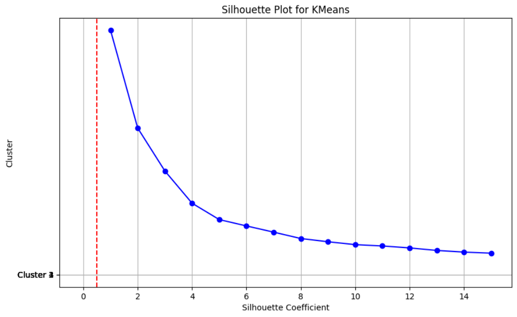

Figure 4: The silhouette scores indicate that while the clustering performs well at lower numbers of clusters, increasing the cluster count leads to a decrease in effectiveness. This aligns with the elbow plot suggesting that fewer clusters might be more effective for this dataset.

3.2.2 Silhouette Scores

The clusters were evaluated using Silhouette Scores as a metric. This number determines how well defined the clusters are. In other words, a higher silhouette score entails more clear clusters and each data item is much closer to its own cluster center than the others. This helps us determine statistical arbitrage groups because it indicates the consistency and reliability of the grouping based on the similarity of data points within each cluster. By confirming that each item in a cluster is more similar to its own group than to those in other clusters, we can confidently identify and exploit patterns, leading to more precise and profitable trading strategies. The silhouette scores were plotted for analysis. Our K-Means silhouette score yielded 0.607. This informs us that on average, the clusters are reasonably distinct and well-separated from each other. Ultimately, our kmeans clustering method performed well in distinguishing between different groups in the dataset.

4 Future Work

For future work and enhancement, having the model observe more features is crucial. Here, we observed only trade price, trade volume, and volume price ratios. We would like to expand this in the future to perhaps include features like price volatility and latency.

5 Conclusion

This study presents a comprehensive approach to leveraging statistical arbitrage using advanced machine learning techniques, specifically focusing on clustering models to optimize financial market decisions. By utilizing a large dataset of NYSE trades and quotes alongside innovative methodologies such as stock embeddings, we have demonstrated the potential to enhance trading strategies' predictive accuracy and efficiency.

Our results highlight the effectiveness of embedding models in capturing the subtle dynamics of the market, which traditional models may overlook. The application of clustering algorithms, particularly K-Means, has allowed us to identify coherent groups that exhibit statistically significant mean reversion patterns, which can be exploited for profitable trading strategies.

Despite facing challenges such as data scale and model complexity, the research provided valuable insights into the practical applications of machine learning in financial markets. Our work contributes to the ongoing discussion on integrating AI technologies in finance, suggesting that such tools can significantly improve the decision-making processes in high-frequency trading environments.

6 Limitations

The present study is subject to various constraints that should be acknowledged. Our project revealed that our model was over fitted and did not take to adding new data very well. Additionally, given the large scale of the dataset, further enhancements are required to be able to scale the project as well. Memory efficiency played a pivotal role in our design and further memory enhancements are required to be able to process a dataset of this nature.

Furthermore, the time constraints imposed on the project prohibited a comprehensive exploration of alternative methodologies. A more exhaustive examination, including exploring additional time series and quantitative related features, was constrained by the limitations of the allotted time frame. We acknowledge the necessity for future research endeavors to explore the performance of our algorithm across varied datasets. This includes a comprehensive investigation of our use of Stock2Vec embeddings and applying PCA to our model.

7 Ethics Statement

The research project was conducted with a commitment to upholding the highest ethical standards in its design, implementation, and reporting. The study adhered to the terms of service implemented by the Intercontinental Exchange, Inc. Importantly, no data mining or scraping activities were undertaken on the NYSE TAQ website or Capital IQ during the course of this research.

8 Acknowledgements

We extend our sincere appreciation to Professor Saining Xie for his guidance in this research. We also express gratitude to Richard Huddleston and Nicholaas Venuti, who served as our mentors for this project and provided invaluable support.

9 Link to Code Repository

https://github.com/shreyaguda/ML_Project

References

PyTorch Contributors. 2023. torch.nn.leakyrelu — pytorch 1.12 documentation.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, et al. 2020. Numpy: Array programming with python. *Nature*, 585(7825):357–362.

Investopedia. 2022. Volume price trend indicator – vpt.

Christopher Krauss, Xuan Anh Do, and Nicolas Huck. 2017. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the s&p 500.

Eryk Lewinson. 2024. Three approaches to encoding time information as features for ml models.

W. McKinney et al. 2020. pandas-dev/pandas: Pandas. *Zenodo*.

New York Stock Exchange. 2023. Daily taq client specification. https://www.nyse.com/publicdocs/nyse/data/Daily_TAQ_Client_Spec_v3.3b.pdf. Accessed: 2024-05-12.

Richard Huddleston Ahmed Jedda Nicholas Venuti, Ambika Sukla and Ian Forbes. 2017. Statistical arbitrage group indentification.

PyTorch. 2024. torch.optim.lr_scheduler.reducelronplateau. https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html.

Xing Wang, Yijun Wang, Bin Weng, and Aleksandr Vinel. 2020. Stock2vec: A hybrid deep learning framework for stock market prediction with representation learning and temporal convolutional network.

Álvaro Cartea, Mihai Cucuringu, and Qi Jin. 2023. Correlation matrix clustering for statistical arbitrage portfolios.