Simulation Engine for ML-enhanced Pairs Trading Strategies for Decision Making

Xinyue Lu

Master of Science

School of Informatics
University of Edinburgh

2023

Abstract

This study aims to develop a simulation engine to capitalize on statistical arbitrage opportunities by integrating clustering algorithms and ML-based forecasting models, helping investors' decision making. First, a k-means clustering algorithm is employed to streamline stock pairs selection by categorizing stocks into classes. Additionally, the incorporation of a moving average for the trading threshold ensures adaptability to evolving market trends, overcoming the limitations of traditional fixed-threshold arbitrage methods. In signal identification, technical indicators serve as features, facilitating the learning of potential lead-lag relationships and correlations between these indicators and the price spread. This approach enhances prediction accuracy by identifying extra profitable opportunities. Lastly, through empirical testing and evaluation, the study demonstrates its effectiveness in reducing pairs selection complexity and capturing additional profitable trading opportunities in dynamic financial markets.

Acknowledgements

This project will not use any personal or sensitive data. All the data required is either stock prices or relevant financial information, such as companies' financial statements, which are all accessible from public resource. Also, the work submitted is my own, without plagiarism and collaboration.

Table of Contents

Chapter 1

Introduction

The financial markets can be described as a complicated, evolving, and non-linear dynamic system, shaped by numerous economic and non-economic elements [1]. These factors can be both predictable and random in nature, resulting in non-linear systems governed by noise and uncertainty. Sahu, Mokhade, and Bokde [2]claimed that, forecasting the behavior of the stock market is a classic but difficult topic, one that has attracted the interest of both economists and computer scientists. Therefore, to effectively analyze and understand these systems, it becomes essential to employ approaches and methodologies capable of handling non-linearity, noise, and uncertainty. A team of researchers who had previously received quantitative training developed a way to long and short stocks that are united in pairs and identified arbitrage opportunities in the equity market in 1985 under the guidance of Wall Street quant Nunzio Tartaglia [3]. The term "Statistical Arbitrage," which describes diversified portfolios traded without risk and on a short-term basis [4], then appeared on the historical scene and became well-known in modern financial markets with a wide variety of assets and rapidly moving information. A later study [5] found that statistical arbitrage is a collection of trading strategies that generates positive anticipated returns and non-negative conditional expected returns under all economic conditions using trading signals derived from historical data at predetermined intervals.

In the subject of statistical arbitrage, the pair trading method is one of the most well-known and often applied techniques. The basic premise of the pairs trading strategy is that if asset pairs consistently exhibit co-movement in the future, then when the spread exceeds a predetermined boundary, the prices of two stocks are very likely to converge in the future, at which point a reasonable profit can be made [6]. When the prices of two assets move apart, and the gap between their prices becomes larger than usual, ar-

bitrageurs can make a bet that they will eventually come back together. They do this by selling the asset that has become more expensive and buying the one that has become cheaper, and vice versa. According to [7], the idea behind pairs trading is surprisingly straightforward and involves two main steps. First, identifying two securities that have historically moved in sync during a specific formation period. Second, monitor the price difference between them during a subsequent trading period. If their prices drift apart and the price gap widens, take a short position on the one that's doing better and a long position on the one that's lagging. If these two securities tend to balance each other out over time, the price gap should eventually return to its usual level. At that point, investors can reverse the positions and make a profit.
Three basic challenges must be addressed to enable a pairs trading strategy [8]:

1. Given a variety of options, how to create portfolio pairs with comparable assets?

2. What signals indicate the occurrence of transient price variations and, consequently, the arbitrage possibilities, for particular portfolios?

3. Last but not least, how could a portfolio manager implement the trading strategy when the signals initiate it in order to reach an optimization, such as a balance between risk and reward, while taking trade limits and potential market frictions into account?

Each of these problems poses major challenges that previous studies have only partially addressed. First of all, due to its priori uncertainty about what "similarity" entails, establishing portfolio pairs for all assets is a difficult task in and of itself. It is crucial to include a variety of assets and periods, consisting of both exogenous data, such as asset qualities, and conventional data, such as price and volume, in order to handle this issue. According to [9], in order to find co-moving stocks and identify trading signals, both the distance technique and the cointegration approach concentrate on historical price pattern analysis. The distance approach, introduced in the influential work of Gatev et al. [7], analyzed all highly liquid U.S. stocks from the CRSP daily files spanning 1962 to 2002. Their methodology involved two key steps. First, they created a cumulative total return index (Pit) for each stock i, starting from the first day of a 12-month formation period. For each stock i, commencing on the first day of a 12-month formation period, they first developed a cumulative total return index (Pit). Second, they estimated the sum of squared Euclidean distances (SSD) for the price time series of all potential couples, taking into account n stocks ($\frac{n(n-1)}{2}$ combinations). Prices

were once more reset to the opening day of this trading session. Trades were entered into when the spread between the pair varied by more than two historical standard deviations σ and were ended upon mean reversion, at the conclusion of the trading period, or if either of the stocks was delisted.
Vidyamurthy [6] provides the most cited work for this approach, in which he constructs a theoretical framework for pairs trading using univariate cointegration. The framework consists of three fundamental steps:

1. Initial Pair Selection: Pre-selecting pairs that have the potential to be cointegrated. This selection can be based on statistical criteria or fundamental analysis

2. Assessment of Trade Viability: Assessing whether these selected pairs are suitable for trading using a specific proprietary approach

3. Development of Trading Rules: Designing trading rules for the chosen pairs using nonparametric methods

Second, all major patterns in the complicated and noisy time series of portfolio prices must be readily identified in order to spot and initiate successful signals. For example, the conventional method of establishing static upper and lower thresholds for price differences proves to be impractical and unreliable in real-world scenarios. This is because shifts in the fundamental factors can lead the price difference to deviate from typical ranges, potentially causing substantial losses. To address these limitations of traditional arbitrage approaches, we implement the following modifications. Typically, the log price spread deviation is tracked, which is intended to initiate entry and exit signals based on the amount of departure from equilibrium. Two other traditional approaches, the time series approach and the stochastic control approach, are predicated on the idea that a set of pairings has already been established in advance. In the time series methodology, the formation period is typically overlooked. All researchers in this field, such as Elliott [10] and Cummins [11], presume that a group of correlated securities has already been identified through previous assessments. Their primary emphasis lies in the trading period and the various ways of generating refined trading signals using time series analysis techniques. This involves modeling the spread as a process that tends to return to its mean value. Similar to the time series approach, the formation period of stochastic control approach is not given much attention. This body of literature, represented by Jurek [12] and Liu [13], is primarily concerned with pinpointing the best portfolio composition for the components of a pairs trade when

compared to other available assets. Stochastic control theory is employed to ascertain the value and optimal strategies for managing this portfolio challenge. While stochastic control approach focuses on choosing the best portfolio holdings to use in a pair of trades relative to other assets, which is frequently used to determine the ideal policy functions and values, time series approach focuses on producing the best trading signals using a variety of time-series analysis techniques [9].

1.1 Motivation and Objectives

Investors face several challenges when it comes to analyzing financial markets. One of the significant challenges is dealing with a wide range of securities, indicators, and market factors simultaneously. Financial markets involve a vast amount of data, and manually processing and analyzing all these data points can be time-consuming and prone to human errors. Another challenge is handling nonlinear relationships in the financial markets. Market dynamics are influenced by multiple interconnected factors, and these relationships can be complex and nonlinear. Traditional statistical models may struggle to capture these intricate patterns and relationships, leading to less accurate predictions.

A dynamic model, on one hand, adapts to the changing market conditions and adjusts its threshold values accordingly [7], [14], [15], [16]. Market conditions, volatility, and cointegration between securities can vary over time, making it challenging to apply a static model that uses fixed threshold values for trading decisions. By incorporating real-time or recent market data, a dynamic model can capture the current market environment more accurately, leading to improved trading decisions. The dynamic model outperforms the static model because it can respond to shifts in market trends, volatility, and cointegration relationships between assets [7]. As market conditions evolve, the static model's fixed threshold values may become outdated and lead to suboptimal trading decisions. In contrast, the dynamic model can identify more profitable trading opportunities by adjusting its thresholds based on the most up-to-date market information. In pairs trading, where the success relies on identifying temporary price divergences between two related assets, a dynamic threshold model allows traders to take advantage of changing market dynamics and optimize their trading strategies accordingly. By considering the current market conditions, the dynamic model helps traders make more informed and timely decisions, resulting in better performance compared to the static model.

Machine learning techniques, on the other hand, are well-suited to handle these challenges. Machine learning algorithms can process and analyze large volumes of data efficiently, enabling investors to explore numerous securities and market factors simultaneously. Moreover, machine learning algorithms can capture and model nonlinear relationships effectively, allowing for more accurate signal identification [17]. By leveraging machine learning, investors can gain insights from complex and dynamic market data, make informed decisions, and improve their ability to predict market movements and identify profitable investment opportunities. Thus, machine learning is crucial for addressing these challenges and enhancing investment strategies in the financial world and investors who leverage advanced machine learning techniques for signal identification may gain a competitive edge over others relying on traditional methods.

More specifically, in time series forecasting, one of the challenges is to determine the predictability of the time series under examination. If the time series is random, all forecasting methods are likely to be ineffective. For investors, predicting the behavior of a time series is crucial for making informed trading decisions. For example, if a time series shows strong mean-reverting behavior (anti-persistence), investors may have more confidence in predicting its future reversals. Conversely, if a time series has a persistent nature, predicting its future direction becomes more challenging, as trends may continue for longer periods. If the spread moves away from its mean but fails to revert back, it is best to refrain from initiating a trade, since opening a trade under such circumstances could result in significant losses. To avoid this risk, it is prudent to wait for the spread to change direction and return to its mean before considering any trading actions. In this case, our first objective is to detect and analyze time series that exhibit at least some level of predictability, by incorporating the Hurst exponent[18] into pairs selection stage of pairs trading, which provides valuable information about the long-term memory of a given time series. The Hurst exponent can be used to classify a time series into different categories based on its value, such as random series, anti-persistent series, and persistent series. This classification is significant for investors as it can help identify the nature of the time series and its predictability.

The challenge for investors also lies in determining the optimal timing for mean-reversion. If the period is too short, it may result in frequent and costly trading, leading to increased transaction costs and potential losses. Conversely, if it is too long, investors may miss out on timely opportunities for profitable trades. Additionally, the speed of mean-reversion can vary over time due to changes in market conditions, eco-

nomic factors, or other external influences [18]. This dynamic nature poses a challenge for investors as they need to continuously monitor and adapt their trading strategies to accommodate these fluctuations. Therefore, the suitable timing of mean-reversion, often measured by the half-life, is considered in this study, which indicates how long it takes for a time series to revert back to its mean value. The half-life provides insights into the speed at which the time series tends to return to its average level after experiencing a deviation[19]. A shorter half-life suggests that the time series reverts back to its mean value relatively quickly, providing more frequent trading opportunities. On the other hand, a longer half-life indicates slower mean-reversion, leading to less frequent trading opportunities. This information is valuable for investors as it helps them identify potential opportunities for trading and profit generation.

To summarize, in this study, the trade restrictions and market frictions is not within our primary focus. Instead, the overall objective of the project is to create a pairs trading simulation engine, and provide traders and investors with a comprehensive framework for portfolio construction, model selection, back-testing, and evaluation, to address the challenges, particularly in pairs selection and trading signal identification.

The study aims to utilize machine learning techniques to tackle the complexities of handling a wide range of securities, indicators, and market factors simultaneously. By leveraging machine learning algorithms, the study seeks to improve signal identification by capturing and modeling the nonlinear and dynamic relationships in financial markets more accurately, in order to assist in making informed investment decisions. In specific, the practical objectives of this project relate to addressing the following:

1. Automating tasks related to data pre-processing, pairs selection, feature engineering, model training, and to streamline the previous tasks into a workflow system to reduce the reliance on manual efforts

2. Addressing the challenge of the predictability of time series data and the suitable timing of mean-reversion in identifying potentially profitable opportunities by introducing Hurst exponent, half life, and crossing per year.

3. Addressing the challenge of model selection by evaluating the performance and generalization of different machine learning models

4. Enabling back-testing, which allows traders and investors to assess the effectiveness and profitability of a trading strategy before deploying it in real-time

In the following context, relevant literature regarding pairs trading strategy and machine learning techniques in stock price prediction will be presented first in Section 2. In Section 3, detailed programme framework and specific methodology will be given and Section 4 will describe the empirical study, including the thr analysis and evaluation about this engine. Finally, the last section will make a conclusion and discuss the potential area of improvement and future work.

Chapter 2

Background and Related Work

Extensive literature has explored the subject of pairs trading from various perspectives, including financial, statistical, and ML-based approaches. This section begins by presenting classical and traditional methods for pairs trading, followed by a review of a paper that employs machine learning algorithms for this purpose.

2.1 Classical and traditional methods concerning pairs trading

The following four approaches are commonly employed in pairs trading implementations: Distance approach, Cointegration approach, Time series approach and Stochastic control approach.

The distance approach, provided by [14], is used to create pairs by finding a suitable match who can reduce the sum of Euclidean squared distance (SSD) between the two normalized price series. The research [10] casted light on later discoveries in this field, which unambiguously described the spread following a mean-reverting Gaussian Markov chain. They developed state space models and fitted estimation mechanisms in order to parametrically handle spreads, which present mean-reverting phenomena, in pairs trading implementations. A stochastic control method mainly concentrates on figuring out the optimized pairs portfolio, when other assets are accessible. The most prominent paper [12] in this field proposed the optimal dynamic strategy for arbitrageurs with a given time series and non-myopic preferences, faced with a mean-reverting arbitrage opportunity, such as an equity pairs trade, and provided the most thorough discussion of the stochastic control approach to model arbitrage opportuni-

ties using an Ornstein-Uhlenbeck framework. They found that intertemporal hedging demands act as a crucial factor in determining the aggressiveness of arbitrageurs in trading against the mis-pricing and account for a large proportion of the total allocation to the arbitrage opportunity.

An engaging ideal pairs trading framework [6] was proposed, using parameterized trading rules for pairs trading, specifically cointegration relationships between assets. The cointegration theory is widely used for stock pairs selection purpose and several studies have confirmed its importance [20], [21], [22].

Nevertheless, [23] made note that large arbitrageurs engage in strategic trading that leads to major market distortions and encourages price manipulation, which lowers the profits made by lesser arbitrageurs. Despite this, the profitability is decreasing as a result of the increased rivalry brought on by the entry of more arbitrageurs into the markets [24]. Further evidence was provided by [25], which showed that the execution risk grows as the number of rivalry rises. As a result, the trading strategy is comparatively more important for the bulk of individual arbitrageurs who lack sufficient capital. The efficiency of the trading technique influences how likely it is that an arbitrage will succeed and generate a profit.

2.2 Dynamic trading bands in pairs trading

[14][7] employed a method to calculate trading bands based on the spread of a stock pair using information from the formation period. The historical equilibrium is determined by the mean of the spread over the entire 10-day period. To detect potential trading opportunities, upper and lower entry bands are established, representing a divergence of 2 times standard deviations from the historical mean. These fixed bands are then applied to the trading period, with the spread being monitored. A trade is started when the spread crosses the upper or lower band, and it is closed when the spread returns to the historical mean. It's vital to remember that, regardless of any patterns seen in the spread, the trading bands remain constant throughout the 5-day trading session. By using a dynamic model in 2012, [15] discovers improved performance for dynamic trading bands. It is possible to detect a divergence from the spread's trend by calculating a moving average for the mean and standard deviation that adjusts to spread trends.

2.3 Machine learning techniques for pairs trading

When taking into account statistical arbitrage opportunities, the majority of past studies that were limited to utilizing standard econometric models to arbitrage are no longer useful. The financial technology (FinTech) sector's rapid development has led to an increase in the usage of machine learning technologies to address financial problems. The results obtained by machine learning approaches have proved very promising. For example, SVM has been used for stock price prediction, portfolio optimization, and credit risk analysis and has been applied to pairs trading in the stock market with promising results. Early in 2004, [26] applied SVM on the data set from shanghai stock market in China to test its ability in forecasting stock prices.

Several research investigated the use of Artificial Neural Networks (ANN) to anticipate the spread change for three well-known spreads, including[27][28] [29]. [30] and [31] integrated the forecast and machine learning approaches. The three steps of the methodology were trading, outranking, and forecasting. Elman neural networks were used by Huck [31] to forecast 1-week returns for each asset.

More recently, [32] concentrated their research on the Indian stock market and confined their investigation to price ratio forecasting as opposed to price spread forecasting. They used three distinct machine learning algorithms in their pairs trading research, which covered the years 2012 to 2015: support vector regression (SVR), random forest (RF), and adaptive-neuro fuzzy inference system (ANFIS). The mean-reverting property of pair price movement was integrated with technical indicators in their framework. As a result, all of the algorithms they used to estimate the ratio of share prices of pairs worked efficiently. Guo and Long [33] devised an efficient pairs trading method in 2020, adding the support vector machine model from machine learning to forecast the spread trend and technical indicators (RSI, SMA) to the mix. The empirical investigation shown that the strategy is able to benefit the actual economy and assist the business in mitigating the risks brought on by price changes.

Chapter 3

Methodology

To achieve our research objectives described in Section 1, we will conduct the following two working parts: 1. data collection and preparation; 2. model adaptation and simulation.

3.1 Data collection and preparation

The dataset for this study is based on the stocks that compose the S&P 500 stock market index. We will start our research based on daily data of S&P500 component stocks. In the views of accessibility and feasibility, we choose to use the 'yfinance' package in Python to access the historical data of S&P 500 components. The reason is that this package automatically adjusts for stock splits and dividends, which is time-consuming to be handled manually. It also provides access to a wide range of financial data beyond stock prices, including corporate actions, financial statements, and analyst recommendations, which might be useful in further research.
The steps for collecting and preparing the datasets are as follows: 1. Get the list of S&P 500 components and download the historical data of for these stocks through 'yfinance'; 2. Specify time window for each period and data split for training set and testing set. Finally, the cross validation will be conducted on training set.

3.2 Model adaption and simulation

Pairs trading involve 2 key processes: identifying good pairs, following a strict criteria described in Section 3.2.1.3, and then take actions, such as entry or exit, on selected pairs.

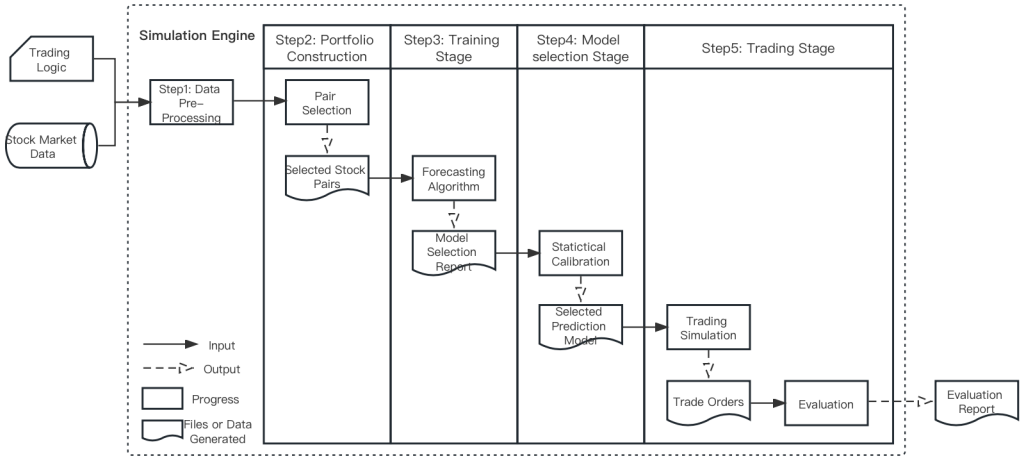


Figure 3.1: Simulation Engine Framework

3.2.1 Proposed Pairs Selection Framework

3.2.1.1 Unsurpervisied Learning - Clustering

Unsupervised learning, which is simply a statistical method, includes clustering techniques. It is a training technique that can find hidden patterns in unlabeled data. Since stocks only need to be compared for eligibility with other stocks within the same clusters, creating asset clusters can help to reduce the computational complexity of locating potential pairs. K-means clustering is a popular and successful unsupervised learning approach for grouping data points into sets [34]. It first requires the specification of K, the number of clusters. By minimizing the within-cluster sum of squares (WCSS) between the data points and their individual centroids, K centroids are then identified, and all the data points are then grouped into one of these clusters. The objective function is given by

$$WCSS = \sum_{i=1}^{N} \sum_{k=1}^{K} w_{ik} \parallel x^i - \mu_k \parallel^2 \tag{3.1}$$

where x^i refers to ith data point, μ_k the centroid of cluster k, w_{ik} = 1 if x^i belongs to cluster k, otherwise w_{ik} = 0, and N is the total number of data points. $\parallel \parallel$ denotes l_2 norm. The minimization problem for the k-means clustering involves two steps. To update the clustering of data points, WCSS is first reduced with regard to w_{ik} while keeping mu_k unchanged. The centroids are then recalculated while maintaining w_{ik} unchanged as WCSS is minimized with regard to μ_k. Up till WCSS is reduced, the aforementioned stages are repeated.

In the context of stock trading, k-means clustering can help identify pairs of assets that exhibit similar price movements or patterns over time. The main reason why this study use k-means clustering is its efficiency, which is quite crucial in pairs selection, where large amounts of historical price data are analyzed. K-means is computationally efficient and can handle large datasets with relative ease. Additionally, its disability of identifying outliers could be accepted, since each possible pairs in each cluster will be assessed through rigorous criteria as indicated in Section 3.2.1.3.

3.2.1.2 Dimensionality reduction

If the data used for clustering has a high dimensionality, it often consumes a lot of computational time during the clustering analysis. In such cases, PCA (Principal Component Analysis) can be applied for dimensionality reduction. A statistical method called principal component analysis (PCA) turns a set of potentially linked variables into a fresh set of linearly uncorrelated variables called principle components. A risk factor can be understood as being represented by each primary component. We propose applying PCA to the normalized return series, which is calculated as

$$R_{i,t} = \frac{P_{i,t} - P_{i,t-1}}{P_{i,t-1}} \tag{3.2}$$

where $P_{i,t}$ is the price series of a asset i. The number of principal components used determines the number of features in each asset's representation. However, high dimensionality of data poses two challenges, first, as the number of attributes increases, the likelihood of including irrelevant features also increases. Additionally, the curse of dimensionality arises due to the exponential increase in data volume with the addition of extra dimensions. According to [35], this effect becomes severe when the number of principal components generated by PCA exceed 15. Therefore, to address this problem, the number of dimensions in this application is limited to a maximum of 15, and this choice is made empirically as discussed in Section 4.

3.2.1.3 Pairs Selection Criteria

After creating clusters of assets to identify candidate pairs, the next step is to establish a set of rules for selecting eligible pairs for trading. The persistence of pairs' equilibrium is crucial, and to achieve this, we propose combining methods from different research works. The proposed criteria for pair selection are as follows, a pair is selected if it complies with all these four conditions. First, to ensure that the two securities forming

the pair exhibit a stable relationship, the pair must pass the Engle-Granger test, following the procedures conducted by [6] for cointegration, which involves the following procedures:

1. Fitting the fitted line using the Least Squares Linear Regression to the equation $\log(P_A) = \gamma * \log(P_B)$, where P_A and P_B refer to the closing prices respectively in the pair and the fitted parameter, γ is denoted as the cointegration ratio.

2. Constructing the spread between the two stocks after figuring out the effects of cointegration, $S_t = log(P_A) - \gamma * \log(P_B)$, at time t.

3. Checking the stationarity of the spread of pairs using an Augmented-Dickey Fuller (ADF) Test. If the pair is cointegrated, its spread should be stationary.

Second, we measure the relative likelihood of a time series to regress to the mean using the Hurst exponent (H), which confirms the spread's mean-reversion nature. A metric called the Hurst exponent gauges a time series' fractality and long-term memory. The time series can be divided into three types based on the Hurst exponent's (H) value (1) H=0.5 indicates a random series. (2) 0.5 <H <1 indicates a persistent series. (3) 0 <H <0.5 indicates an anti-persistent series, characterized by a tendency to revert to its mean value; when the series goes up, it is more likely to go down next, and vice versa. As H gets closer to 0.0, "mean reverting" becomes stronger [18]. Therefore, we demand that the Hurst exponent of a pair's spread be less than 0.5 in order to guarantee mean-reversion. The third goal is to eliminate stationary couples with inappropriate timings. Profits cannot be assured by a mean-reverting spread alone; there must be alignment between the mean-reversion time and the trading period. A key factor in [19] is the half-life of mean-reversion, which measures how long it takes for a time series to return to its mean. Because of this, we suggest excluding combinations for which the half-life takes extreme values, such as less than one day or more than one year. Last but not least, we suggest that every spread cross its mean at least twelve times annually, effectively resulting in one trade on average per month.

3.2.1.4 Pairs Selection Framework Diagram

Three building blocks of the proposed framework explains their functionality. Initially, the price series of all potential pairs' assets are input into and PCA to reduce data dimensionality. Each security is represented by a compact version derived from its price series. Using this simplified representation, the K-means algorithm clusters the

securities. Finally, we search for pair combinations within these clusters and select the ones that meet the specified conditions.

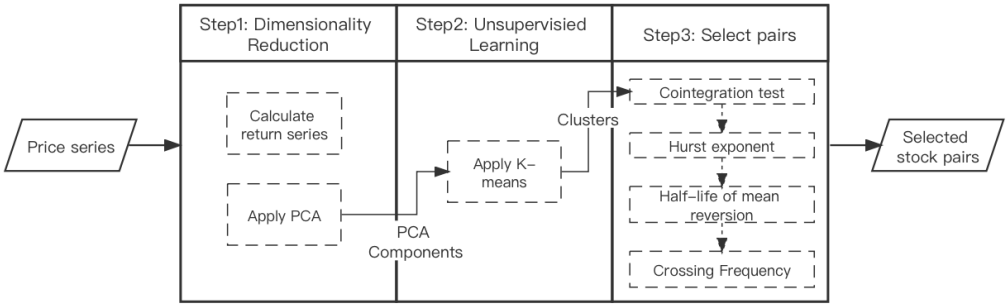


Figure 3.2: Pair selection diagram

3.2.2 Proposed Trading Framework

3.2.2.1 Trading model

Let the price of stock A and stock B at time t be P_A^t and P_B^t. We currently know the prices of both assets, and since future returns on both assets are anticipated to exhibit comparable tendencies, the time series of the two prices ought to move together.
The spread S_t at time t is indicated in Section 3.2.1.3

$$S_t = log(P_A^t) - \gamma * \log(P_B^t) \tag{3.3}$$

where γ is denoted as the cointegration ratio, the coefficient β of OLS regression of two return series. Denoting the amount that the spread deviates from the equilibrium μ, the mean of price spread as the signal. For example, considering the strategy where trades are put on and unwound on a deviation of Δ, calculated from standard deviation of price spread on either direction from the long-run equilibrium. If the signal is Δ above the mean value, then we expect that the stock A is overpriced and the stock B is underpriced. Therefore, we choose to short 1 unit of stock A and long γ units of stock B. In contrast, if the signal is Δ below the mean value, we can long 1 unit of stock A and short γ units of stock B.

$$S_t \begin{cases} \geq \mu + \Delta, & shortA, longB \\ \leq \mu - \Delta, & longA, shortB \end{cases}$$

In this way, we can recognize the statistical arbitrage opportunities and make profits through this trading strategy, which is the incremental change in the spread 2Δ.

And we follow the finding from [16], with the position closed when the spread hitting the dynamic mean. The traditional stock arbitrage approach involves setting a fixed price difference to conduct arbitrage while maintaining constant thresholds, where Δ is normally one or two times standard deviation. However, this traditional approach of setting fixed upper and lower limits for the price difference is not effective nor reliable in practice , as changes in the underlying fundamentals may cause the price difference to deviate from normal levels, resulting in significant drawdowns. To overcome the shortcomings of the traditional arbitrage approach, we make the following changes. First, we use the 10-day moving average of the price difference as its mean level, which better reflects the price difference's changing trends.Here, the moving average window is a hyperparameter, which could also be 7-day, determined by users' choice. After that, the upper and lower limits for the price difference are not fixed, instead, we take the 10-day moving average of the price difference plus/minus one standard deviation as the upper and lower thresholds to generate buy and sell signals.

3.2.2.2 Forecasting Algorithms

After the stock pairs are detected, the trading strategy will generate a trading signal by predicting whether the spread will diverge or converge in the future. A sufficient set of technical indicators described in Appendix A from 'TA-lib' package will be used as features that are relevant to the price spread of the pair. Subsequently, we will train a model to learn the relationship between these features and the price spread of the pair. The model may discover direct correlations between specific features and the price spread. For example, it may learn that when certain technical indicators, such as moving averages and RSI reach specific values or exhibit certain patterns, they tend to coincide with certain movements in the price spread. Besides, it might helpful in learning lead-lag relationships that changes in some features precede changes in the price spread. For instance, changes in trading volume or momentum indicators typically occur before significant movements in the spread. To enhance the functionality, we provide multiple choices of different binary classifier, including support vector machine, random forest, XGBoost, naive bayes, k-nearest neighbour, and logistic regression. Afterwards, each model's performance will be evaluated using the chosen evaluation metrics on the cross-validation. This will give you an initial sense of which models are performing better.

During the training stage, the SVM algorithm tries to find the hyperplane that best separates the two classes of interest, which in the proposed research relates to whether

the pair in question is in a state of convergence or divergence. Once the hyperplane is found, the class of new, unseen instances of the pair can be predicted based on their feature values. Random forest and XGBoost, which are two ensemble learning methods combining multiple decision trees to create a strong predictive model, learn complex patterns in the data and make predictions based on the collective votes of their constituent decision trees. Logistic regression estimates the coefficients of the features and calculates the probability of a pair being in a particular class (convergence or divergence) to make binary predictions based on predefined thresholds. Different from logistic regression, naive bayes applies Bayes' theorem to model the conditional probability of a pair belonging to a specific class. It assumes conditional independence of features and calculates probabilities based on occurrences of specific features, making predictions based on the highest probability. Finally, k-nearest neighbour classifier identifies pairs with similar feature values and classifies a new pair based on the majority class of its k-nearest neighbors, determining the state of convergence. If the predicted class is convergence, the trader may consider buying the underperforming security and selling the overperforming security in anticipation of a reversal towards the mean. Conversely, if the predicted class is divergence, the trader may consider exit the position or reverse it to take advantage of the trend.

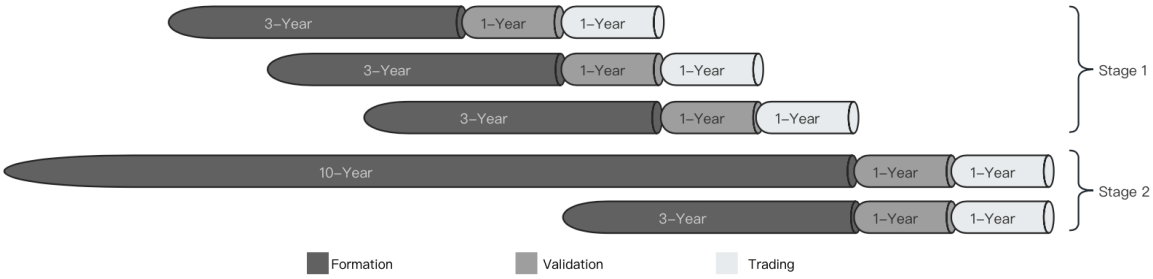
The performance and generalization will be assessed in the Model Selection Stage using evaluation metrics, including accuracy, precision, recall, F1-score and ROC curve. By evaluating the models through cross-validation and out-of-sample testing, their performance can be compared, and a quantitative assessment can be made for statistical calibration. Cross-validation is a technique used to help in estimating how well the model is likely to perform on unseen data by partitioning the available dataset into multiple subsets (k-folds, in this study we use k=5). The model is trained on a subset of the data and tested on the remaining data. This process is repeated several times, with different subsets used for training and testing each time. Out-of-sample testing, also known as testing on unseen data,is to assess how well the model generalizes to new data points, which involves evaluating the model's performance on data that it has never encountered during training or cross-validation. This helps to simulate the real-world scenario where the model encounters new, previously unseen data. The results of the evaluation will be summarized in a report that provides insights into the performance, generalization capability, calibration of each model to help users select the most appropriate model for their specific needs and requirements.

3.2.3 Simulation Engine Design

3.2.3.1 Dataset

The time periods taken into consideration for each simulating stage are shown in Figure 3.3. There are two testing sets: (i) using 10-year-long formation periods to simulate the forecasting-based trading model, which needs more formation data to fit the forecasting algorithms, and (ii) using 3-year-long formation periods for training and pairing pairs using the dynamic threshold-based trading model. In both instances, performance is validated using the second-to-last year before the strategy is applied to the test set. According to findings of Do and Faff [24], extending the initial 6-month trading period proposed in [7] to 1 year can increase profitability. Therefore, in this engine, we choose 1-year trading period to test its economical performance.

Figure 3.3: Formation and Trading Periods



In order to acquire more statistical support for the results, we advise using three separate periods in the beginning. However, a 3-year term might not be viable in the second stage due to the computing workload needed to train the forecasting algorithms. So, in order to assess how the conventional model would have fared over the same test period, we choose to take into account two options: one using a 3-year formation period and the other using a 10-year formation period.

3.2.3.2 Stage 1 - Portfolio Construction

Initially, we aim to compare the effectiveness of two distinct pairs' search techniques: one without grouping and the other involving grouping with k-means clusters. To achieve this, we employ two methodologies. For each search technique, we implement the proposed pairs selection criteria. We do not give the trading conditions much thought as we are primarily concerned with contrasting the search methodologies. As

a result, we use the dynamic threshold-based model suggested in Section 3.2.2.1 with the parameters listed in Table 3.1 The 10-day moving window is used to compute the standard deviation σ_s and mean μ_s of the spread.

Table 3.1: Dynamic Threshold-based Model Parameters

Parameters	**Values**
Long Threshold	$\mu_s - 2\sigma_s$
Short Threshold	$\mu_s + 2\sigma_s$
Exit Threshold	μ_s

We run three separate test portfolios, which simulate different trading circumstances, to assess the performance of the chosen pairs. All of the pairs found during the formation phase are included in Portfolio 1. Portfolio 2 includes the results of applying the technique in the validation set and picking just the combinations with profitable results. Last but not least, Portfolio 3 mimics a situation in which the investor is constrained to making investments in a set number of N pairs. In this instance, based on their responses from the validation set, we advise choosing the top-N pairs. Following a similar decision made by [7], N = 5 is chosen. In this situation, we are able to judge the ideal circumstances for its implementation as well as the ideal clustering process.

3.2.3.3 Stage 2 - Trading Stage

In the second phase, our objective is to compare the reliability of the dynamic threshold-based model and the forecasting-based model. Initially, we assess the forecasting performance of each algorithm. To establish a benchmark, we include a naive baseline without any model. Subsequently, we evaluate the effectiveness of the trading strategy itself, utilizing the pairs search technique that yielded more promising results based on the findings from the previous phase.

3.2.3.4 Trading Simulation

We make sure that all pairs have equal weights in the portfolio when building it. We follow the typical capital allocation strategy employed by the majority of hedge funds, in which the capital gained from the short position is promptly invested in the long position. In order to achieve standardization, we set each trade to be a fixed amount of

$1, which means that the maximum investment in each pair is capped at $1, illustrated as:

$$
Position \begin{cases} \gamma \leq 1 \begin{cases} Long \begin{cases} Buy & \$1\ of\ B \\ Sell, & \$\gamma\ of\ A \end{cases} \\ Short \begin{cases} Buy & \$\gamma\ of\ A \\ Sell, & \$1\ of\ B \end{cases} \end{cases} \\ \gamma > 1, \begin{cases} Long \begin{cases} Buy & \$\frac{1}{\gamma}\ of\ B \\ Sell, & \$1\ of\ A \end{cases} \\ Short \begin{cases} Buy & \$1\ of\ A \\ Sell, & \$\frac{1}{\gamma}\ of\ B \end{cases} \end{cases} \end{cases}
$$

In this context, we choose to trade with the multiples of 100 times. All profits generated by a pair's transaction during the trading period are reinvested in that pair's subsequent trade as the trading process moves forward. Based on projections from Do and Faff, [36], this engine accounts for transaction costs for both assets in a pair, including fees (8 bps), market effect (20 bps), and short-selling restrictions (1% yearly). Slippage is set to 10% by default, and users can manually change any of the above parameters.
It is important to note that this study does not implement a stop-loss system under any circumstances. This means that a position is only exited if the pair converges or the trading period comes to an end.

3.2.3.5 Evaluation Metrics

The trading evaluation measures the strategy's ability to generate profits. It considers metrics such as annualized returns, net profits, and risk-adjusted returns, such as Sharpe Ratio, Calmar Ratio and Maximum Drawdown to assess the strategy's profitability relative to the risks taken:

1. Annualized Return calculates the average annual return generated by the trading strategy. It helps assess the strategy's performance over a longer time horizon.

2. Maximum Drawdown measures the largest drop in the value of the trading strategy from a peak to a trough over a specific period. It gives an indication of the strategy's risk and potential losses.

3. Sharpe Ratio assesses the risk-adjusted return of the trading strategy by considering the excess return earned per unit of risk taken. It takes into account

both the strategy's return and volatility. A higher Sharpe ratio indicates a better risk-adjusted performance, as it suggests that the investment or portfolio has generated higher returns per unit of risk taken.

$$SharpeRatio = \frac{Annualized\ Return - Risk - Free\ Rate}{Standard\ Deviation} \tag{3.4}$$

4. Calmar ratio is a risk-adjusted performance measure used in the financial industry to evaluate the return of an investment strategy relative to its maximum drawdown. A higher Calmar ratio indicates a better risk-adjusted performance, as it suggests that the strategy has generated higher returns relative to the risk taken.

$$CalmarRatio = \frac{Annualized\ Return}{Maximum\ Drawdown} \tag{3.5}$$

Chapter 4

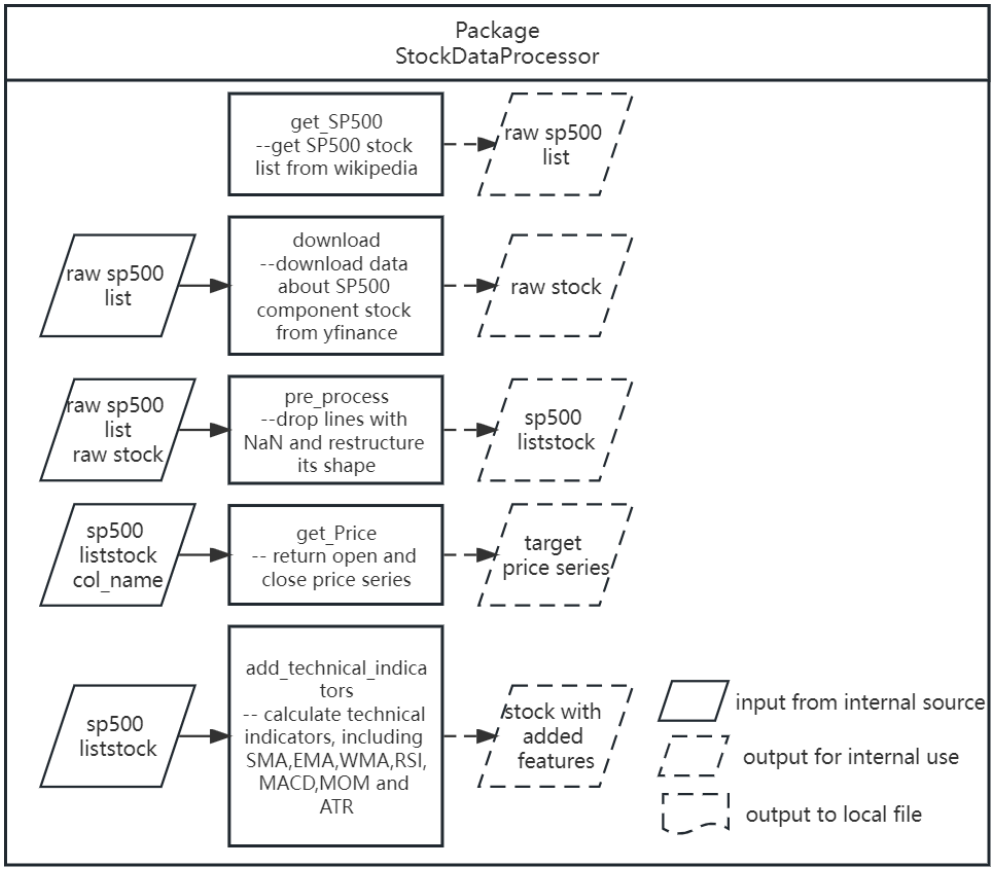
Empirical Study

The proposed simulation engine is compiled in 5 python package: StockDataProcessor.py, PortfolioConstruction.py, ConvergencePrediction.py and TradingSimulation.py. In the main.py, several functions are also given to execute several actions such as selecting pairs, predicting, simulating and comparison work. It also gives investors flexibility in the aspect of stock data features, model selection, and choice of related parameters.

4.1 Package StockDataProcessor

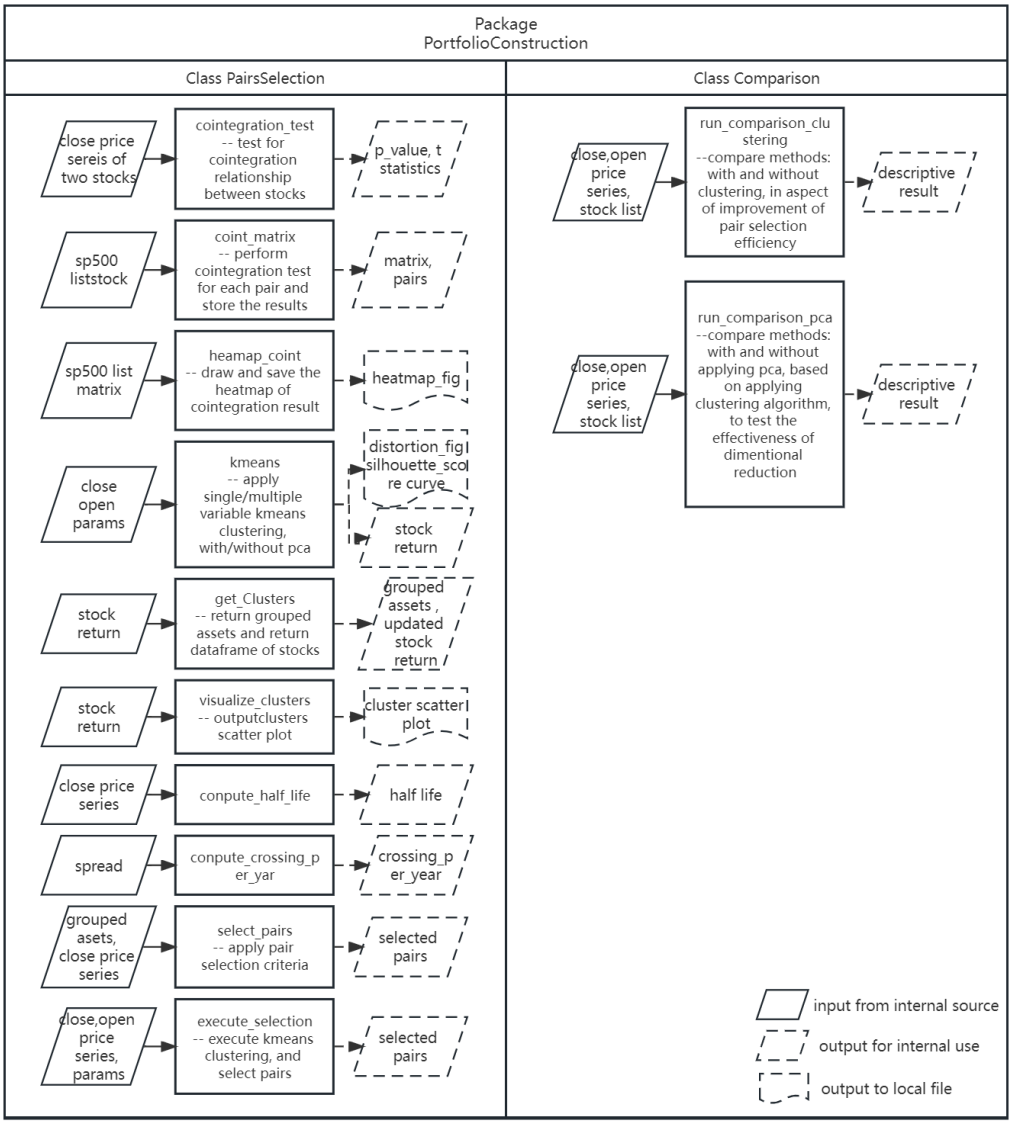
This class 4.1 mainly used for downloading and preprocessing data. It would be more efficient to download necessary data to local server and read the csv file into python, to avoid repetively downloading data through 'yfinance', which is not suggested, especially in strategy testing period. First, 'get_SP500' function is used to download the sp500 list through 'wikipedia' package, and then 'downloading' function download the relevent data of sp500 component stocks through 'yfinance' and the stock list received from 'get_SP500' function. Our sample period is from Janurary 2003 to December 2022. Generally, a larger sample size is preferred as it can help to reduce the risk of overfitting and improve the accuracy of the model. Afterwards, we conduct several pre-processing work, including drop the stocks which do not have full data within the periods that users assigned. Also, the 'get_Price' returns the price series of each stock, providing convenience for the following process. The technical indicators will also be calculated through 'ta-lib' package and be concated into the dataframe of stock data in this Class, through 'add_technical_indicators'.

Figure 4.1: Package StockDataProcessor



4.2 Package PortfolioConstruction

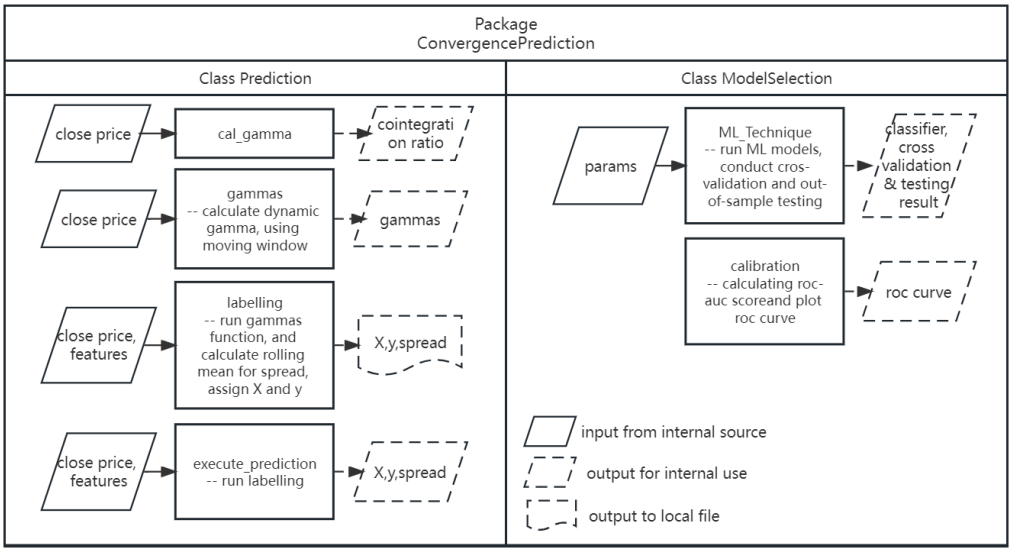
Figure 4.2: Package PortfolioConstuction



In PortfolioConstruction 4.2, there are two class: Class PairSelection and Class Comparison(PairSelection). First, Class PairSelection includes the execution of pair selection criteria and related calculation, such as cointegration test with its visualization B.1, k-means clustering, the application of PCA after normalizing the input data, and computation of half-life and crossing-per-year. Afterwards, Class Comparison inherits all the traits from PairSelection, which only includes one function 'run_comparison' to evaluate the effectiveness of clustering in pairs selection framework.

4.3 Package ConvergencePrediction

Figure 4.3: Package ConvergencePrediction

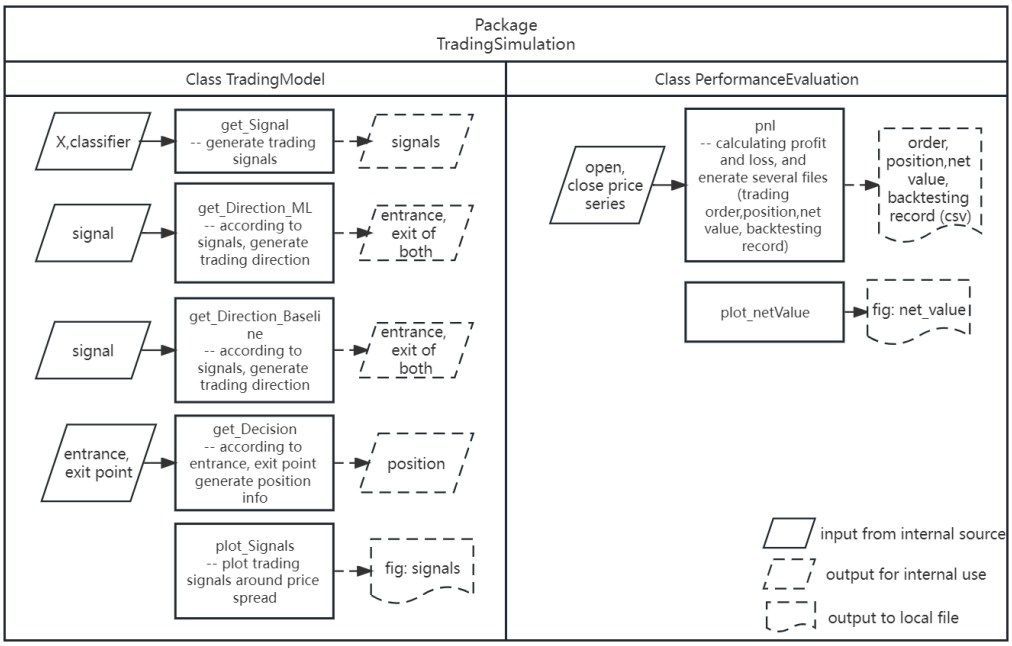


This package 4.3 is consist of two class: Prediction, ModelSelection. First, Class Prediction calculates gamma for the construction of price spread series of stock pairs and relevant dynamic thresholds. Additionally, it involvings function 'labelling', which mainly used for adding binary label according to dynamic z-score. If dynamic z-score is larger than one or smaller than -1, the label is assigned with 1, otherwise 0. The Class ModelSelection provides six machine learning binary prediction choice, performing cross-validation and calculating evaluation metrics for user's selection. According to user's input of ML model, it applies the classifier onto out-of-sample test set, and return the results of evaluation metrics. In this framework, we follow [37], using first 80% of the data for in-sample training and the rest 20% for out-of-sample testing without shuffling Finally, the function 'calibration' receives the testing results, calculates ROC-AUC score and plots ROC curve B.3 for each model, which will be saved to user's local file.

4.4 Package TradingSimulation

Finally, the Class TradingModel 4.4 generates the trading signal through y_pred and then get the trading direction. In this part, the Baseline model generates entry or exit signals according to the dynamic z-score, while ML-based model incorporates the re-

Figure 4.4: Package TradingSimulation



sult of prediction. The position record will also be generated according to the trading decision and it also provides plots of signals incurred. The package TradingSimulation also includes Class PerformanceEvaluation, which calculates profit and loss of each trading and save the files of net_value, trading order, trading positions and backtesting results to user's local server, as well as the visualization of trading signals presented in Figure B.5, where green representing open position, and red represent close position, and net value in Figure B.2, for their configuration with the real-time transaction

Chapter 5

Evaluation

With the packages described in Section 4, users are able to make use of the functions within each class and make informed decision. In this section, we mainly consider the evaluation of different pairs selection methods and trading models.
In Section 5.1, a summative table will be presented about comparing two methods: with/without grouping, and with/without PCA in the metrics of number of clusters, possible combinations and number of pairs selected. The effects of a validation set(during which the profitability of pairs selected will be evaluated) will be compared to check whether selecting pairs according to the returns obtained in validation set is helpful.
Subsequently, Section 5.2 will give information about comparison and evaluation of trading model, between baseline model (without applying any ML techniques), model with enhanced pair selection method, and model with enhanced forecasting and prediction techniques, in the metrics of profitability..

5.1 Evaluation of Pair Selection Methods

We begin by providing some pertinent data on the number of pairings discovered using each of the three pair-search strategies that are being contrasted at this time. As would be predicted, more pairs are chosen when there are no constraints placed on the search field since a wider range of stocks appear. On the other hand, there are less possible pair combinations when stocks are divided into ten partitions. Last but not least, k-means clustering significantly reduces the number of pair combinations that are conceivable.

5.1.1 Performance of Clustering

We propose studying the contributing price series to assess the clusters' integrity. As a result, we choose two clusters to represent the price series of the logarithmic stocks.

Table 5.1: Comparison - With/Without Applying kmeans clustering

Formation period	2003 - 2006	2004 - 2007	2005 - 2008	AVG	%
Without Grouping					
Number of Clusters	1	1	1		
Number of Total Combination	19701	19701	19701	19701	
Number of Pairs Selected	796	742	509	682	3.46
With Grouping on Single Variable					
Number of Clusters	10	10	10		
Number of Total Combination	2239	2437	2354	2343	
Number of Pairs Selected	114	168	134	139	5.92
With Grouping on Multiple Variable					
Number of Clusters	10	10	10		
Number of Total Combination	2789	2235	2359	2461	
Number of Pairs Selected	130	123	88	114	4.62

Figure B.4 visualize the scatter plot of stocks, with different color of dots representing different clusters. As we can see in Table 5.2, the number of combinations without clustering is significantly larger than the number of combinations with clustering, from about 20,000 to 2,000, which proves that clustering has reduced the number of pairs that need to be considered for cointegration testing. This reduction can lead to significant computational savings, especially when dealing with a large number of assets and the search space for potential pairs could be effectively reduced. Additionally, clustering helps in narrowing down the focus to assets within the same cluster, increasing the chances of finding co-integrated pairs among assets with similar characteristics. According to the average proportion of the number of pairs selected within the number of total combination, the percentage increased from 3.5 to around 6 with single-variable clustering, and to 4.6 with multiple-variable clustering, adding to its computational power efficiency. Therefore, it is reasonable to conclude that adding the functionality of clustering in pairs selection method is beneficial for reducing users' computational

complexity and adding to efficient decision making.

5.1.2 Performance of Dimensional Reduction

Table 5.2: Comparison - With/Without Applying PCA based on kmeans clustering with multiple dimensions

Formation period	2003 - 2006	2004 - 2007	2005 - 2008	AVG	%
With PCA					
Number of Clusters	10	10	10		
Number of Combination (Total Pairs)	2478	2271	2193	2314	
Number of Pairs Selected	129	110	95	111	4.81
Without PCA					
Number of Clusters	10	10	10		
Number of Combination (Total Pairs)	2545	2282	2506	2444	
Number of Pairs Selected	106	85	45	79	3.22

Table 5.2 gives information about the effectiveness of PCA. In this study, PCA is conducted on multiple-variable k-means clustering with close and open price series. The result shows that the efficiency of pairs selection is slightly improved. Although the total number of possible combination and number of pairs selected is not significant different, the average proportion of the number of pairs selected without PCA increased from 3.22 to 4.81 with PCA. Additionally, this study is limited to small dimension of close, open, high, low price series and volume, so it seems that the functionality of dimensionality reduction is critical and especially useful for users whose dataset with large dimension.

5.1.3 Effectiveness of Validation Period

In this part, we analyze the effects of validation period, which helps filter out pairs with positive returns and passing these pairs into the next testing period. We conducted analysis on three sets based on the Naive Bayes classifier, mentioned earlier in Section 3.3, with formation period in 2003-2006, 2004-2007, 2005-2008 and the corresponding validation period in 2006-2007, 2007-2008, 2008-2009, testing period in 2007-2008, 2008-2009, 2009-2010. It worth mentioning that pairs selected without applying pca,

into three different sets, since the cluster centroids of kmeans are random set in each trial.

As described in Table 5.3, pairs with NAN in validation or testing period means it did not open position or make transaction during that period. Among 15 pairs of 3 sets in total, during validation period, there are 6 pairs presenting negative annual return, 4 of these continues failing to make profit in the subsequent testing period. It seems that pairs with negative annual return in Validation Period continues have negative annual return in Testing period, although some negative pairs reverse its performance in the following period. However, limited to strict pairs selection criteria, only few pairs were selected in the formation period. Therefore, samples(stocks) should be extended and tested for its effectiveness to realize generalization. Thus, incorporating the validation period, which function as a filter, to rule out the pairs which might generate negative returns is worthwhile in helping users control loss while obtaining profit.

5.2 Evaluation of the Forecasting-based Trading Model

At this stage, we will examine how the forecasting-based trading model stacks up against the conventional model currently in use. Because of its proven effectiveness, we use the k-means clustering to choose partners in this expression. The last sector demonstrated that the number of pairs detected for the former is much lower because there are fewer active cointegrated stocks during this interval. Having fewer pairs is practical, despite the computationally intensive nature of training the forecasting models.

Additionally, we have confirmed that during the validation period, every model that has been built can outperform a naive implementation. In order to evaluate the forecasting-based trading model, we use the clustering model and introduce the validation period in this section. We analyze the performance obtained by the integration of these forecasting algorithms in the proposed trading model scheme.

It is remarkable that, under the situation that two methods have similar win rate, on average the standard deviation and drawdown of ML model is significantly smaller than that of Baseline Model, with the 3-year data at (0.2, 0.35) and 10-year data at (0.18, 0.33) respectively, which proves that ML-based Forecasting algorithm is effective in reducing loss and control risk, while obtaining reasonable profit.

Additionally, the testing set with 3-year Formation period shows the average annual return, sharpe ratio and calmar ratio shows its better profitability. However, in an-

Table 5.3: Evaluation of Effectiveness of Validation Period

	Set 1						Set 2						Set 3				
	Stock 1 Stock2	DE WY	DIS MAS	DIS WY	DOV SEE	DD PH	Stock 1 Stock2	ALK ROL	BIO DE	CAT J	BA FRT	BA VMC	Stock 1 Stock2	CMA HD	AMAT BALL	AMAT DXC	AMAT KLAC
RVTY																	
Validation Period	2006 - 2007						2007 - 2008						2008 - 2009				
	stddev	0.089	0.041	0.101	0.012	0.080		0.011	0.000	0.168	0.199	0.000		0.121	0.047	0.045	0.078
0.065																	
	sharpe	1.300	-0.535	-0.878	0.626	-2.316		-0.124	nan	0.347	0.244	nan		-0.193	-0.172	-0.545	-0.439
1.748																	
	calmar	2.436	-0.636	-0.564	0.709	-0.782		-0.137	nan	0.329	0.283	nan		-0.157	-0.187	-0.666	-0.343
2.986																	
	absReturn	0.104	-0.020	-0.080	0.007	-0.168		-0.001	0.000	0.053	0.044	0.000		-0.022	-0.007	-0.022	-0.032
0.104																	
	annualReturn	0.115	-0.022	-0.088	0.008	-0.185		-0.001	0.000	0.058	0.049	0.000		-0.023	-0.008	-0.024	-0.034
0.114																	
	maxDrawdown	0.047	0.034	0.157	0.011	0.237		0.010	0.000	0.178	0.171	0.000		0.149	0.043	0.037	0.100
0.038																	
	winRate	0.566	0.491	0.443	0.615	0.465		0.481	nan	0.460	0.456	nan		0.538	0.430	0.490	0.457
0.586																	
Testing Period	2007 - 2008						2008 - 2009						2009 - 2010				
	stddev	0.141	0.003	0.144	0.050	0.117		0.031	0.000	0.187	0.295	0.065		0.093	0.033	0.035	0.054
0.039																	
	sharpe	-0.528	0.117	-0.092	0.093	0.967		-0.362	nan	1.346	-0.936	1.795		-1.031	-0.145	0.508	-1.302
-0.860																	
	calmar	-0.327	0.189	-0.064	0.074	1.285		-0.472	nan	1.181	-0.828	13.626		-0.997	-0.180	0.591	-0.653
-0.531																	
	absReturn	-0.068	0.000	-0.012	0.004	0.103		-0.010	0.000	0.231	-0.253	0.106		-0.087	-0.004	0.016	-0.064
-0.030																	
	annualReturn	-0.075	0.000	-0.013	0.005	0.113		-0.011	0.000	0.252	-0.276	0.116		-0.096	-0.005	0.018	-0.070
-0.033																	
	maxDrawdown	0.228	0.002	0.207	0.063	0.088		0.024	0.000	0.213	0.333	0.009		0.096	0.026	0.030	0.107
0.063																	
	winRate	0.454	0.500	0.476	0.478	0.584		0.458	nan	0.561	0.489	0.700		0.486	0.510	0.545	0.456
0.466																	

Table 5.4: Comparison - Trading Model

	stddev(AVG)	sharpe(AVG)	calmar(AVG)	absReturn(AVG)	annualReturn(AVG)	maxDrawdown(AVG)	winRate(AVG)
3-year Formation Period							
ML model	0.2047	-0.0568	0.4898	0.0012	0.0014	0.2162	0.4932
Baseline Model	0.3586	-0.1349	0.3307	-0.0146	-0.0160	0.3934	0.4952
10-year Formation Period							
ML model	0.1812	0.0028	0.5347	0.0412	0.0453	0.1792	0.4903
Baseline Model	0.3265	0.0651	0.5010	0.0666	0.0732	0.3298	0.4974

other set with 10-year Formation Period, the ML-based model did not show significant advantages, which is more obvious in the terms of sharpe ratio, in which ML-based model is 8% higher than baseline model in 3-year formation period, while 6% lower in 10-year formation period. This probably because the old information not adds to, but rather deteriorate the forecasting algorithm. Therefore, introducing ML-based model is effective in enlarging profitable trading opportunities, by filling the arbitrage gap, where the traditional threshold calculated from spread fails to recognize.

Chapter 6

Conclusions and Future Work

6.1 Conclusion

In this proposed trading framework, we have outlined a comprehensive approach to capturing statistical arbitrage opportunities using a combination of clustering algorithm in pairs selection and advanced forecasting algorithms in trading model. By exploiting the relationship between stock pairs and their price spread, we aim to develop a robust trading strategy simulation engine that can adapt to changing market conditions and yield profitable outcomes.

First, we applied a k-means clustering algorithm to help reduce the complexity in stock pairs selection. After evaluation, we conclude that grouping stocks into clusters is effective to reduce the computational complexity in finding the candidate pairs, both the number of possible combinations and selected pairs can be largely reduced.

The trading model leverages the concept of the spread between the prices of stock A and stock B. The incorporation of the moving average of the price spread as a dynamic mean level introduces flexibility to adapt to evolving market trends and enables users to take appropriate short or long position. This approach aims to overcome the limitations of traditional fixed-threshold arbitrage methods by capturing changing market dynamics.

In trading signals identification part, we employ technical indicators as features. It allows the forecasting algorithms to learn and identify potential lead-lag relationships and correlations between these features and the price spread, enhancing the prediction accuracy.

6.2 Future Work

While the proposed trading framework presents a comprehensive strategy for pairs trading, several areas need further exploration and refinement First, the selection of features play a pivotal role in model performance. Exploring additional technical indicators or alternative data sources could enhance the predictive power of the models. Additionally, there are several hyperparameters in the proposed trading framework, such as the number of components in k-means clustering, the length of moving average window. Therefore, hyperparameter tuning for the selected classifiers can significantly impact their performance. Further optimization efforts can be undertaken to improve model accuracy and generalization. Lastely, incorporating news sentiment analysis and market event detection can provide more valuable insights to enhance prediction accuracy and decision-making.

Bibliography

[1] M. Ballings, D. Van den Poel, N. Hespeels, and R. Gryp, "Evaluating multiple classifiers for stock price direction prediction," *Expert systems with Applications*, vol. 42, no. 20, pp. 7046–7056, 2015.

[2] S. K. Sahu, A. Mokhade, and N. D. Bokde, "An overview of machine learning, deep learning, and reinforcement learning-based techniques in quantitative finance: Recent progress and challenges," *Applied Sciences*, vol. 13, no. 3, p. 1956, 2023.

[3] A. Pole, *Statistical arbitrage: algorithmic trading insights and techniques*. John Wiley & Sons, 2011.

[4] S. Hogan, R. Jarrow, M. Teo, and M. Warachka, "Testing market efficiency using statistical arbitrage with applications to momentum and value strategies," *Journal of Financial Economics*, vol. 73, pp. 525–565, 2003.

[5] O. Bondarenko, "Statistical arbitrage and securities prices," *Review of Financial Studies*, vol. 16, pp. 875–919, 2002.

[6] G. Vidyamurthy, *Pairs Trading: quantitative methods and analysis*, vol. 217. John Wiley & Sons, 2004.

[7] E. Gatev, W. N. Goetzmann, and K. G. Rouwenhorst, "Pairs trading: Performance of a relative-value arbitrage rule," *The Review of Financial Studies*, vol. 19, no. 3, pp. 797–827, 2006.

[8] R. J. Gundersen, "Statistical arbitrage: High frequency pairs trading," Master's thesis, 2014.

[9] C. Krauss, "Statistical arbitrage pairs trading strategies: Review and outlook," *Wiley-Blackwell: Journal of Economic Surveys*, 2017.

[10] R. Elliott, J. V. D. Hoek, and W. P. Malcolm, "Pairs trading," *Quantitative Finance*, vol. 5, pp. 271 – 276, 2005.

[11] M. J. Cummins and A. Bucca, "Quantitative spread trading on crude oil and refined products markets," *Quantitative Finance*, vol. 12, pp. 1857 – 1875, 2011.

[12] J. W. Jurek and H. Yang, "Dynamic portfolio selection in arbitrage," in *EFA 2006 Meetings Paper*, 2007.

[13] J. Liu and A. Timmermann, "Optimal convergence trade strategies," *Review of Financial Studies*, vol. 26, pp. 1048–1086, 2013.

[14] E. G. Galev, W. N. Goetzmann, and K. G. Rouwenhorst, "Pairs trading: Performance of a relative value arbitrage rule," *Capital Markets: Market Efficiency*, 1999.

[15] V. Kishore, "Optimizing pairs trading of us equities in a high frequency setting," 2012.

[16] A. S. Velayutham, D. Lukman, J. Chiu, and K. Modarresi, "High-frequency trading," tech. rep., Working Paper, Stanford University, 2010.

[17] S. M. Sarmento and N. C. G. Horta, "A machine learning based pairs trading investment strategy," 2021.

[18] B. Qian and K. Rasheed, "Hurst exponent and financial market predictability," in *IASTED conference on Financial Engineering and Applications*, pp. 203–209, Proceedings of the IASTED International Conference Cambridge, MA, 2004.

[19] E. Chan, *Algorithmic trading: winning strategies and their rationale*, vol. 625. John Wiley & Sons, 2013.

[20] Y. Lin, M. McCrae, and C. Gulati, "Loss protection in pairs trading through minimum profit bounds: A cointegration approach," *Adv. Decis. Sci.*, vol. 2006, pp. 73803:1–73803:14, 2006.

[21] C. L. Dunis, G. Giorgioni, J. Laws, and J. Rudy, "Statistical arbitrage and high-frequency data with an application to eurostoxx 50 equities," *Liverpool Business School, Working paper*, 2010.

[22] M. C. Chiu and H. Wong, "Dynamic cointegrated pairs trading: Mean-variance time-consistent strategies," *J. Comput. Appl. Math.*, vol. 290, pp. 516–534, 2015.

[23] M. Attari, A. S. Mello, and M. E. Ruckes, "Arbitraging arbitrageurs," *The Journal of Finance*, vol. 60, no. 5, pp. 2471–2511, 2005.

[24] B. Do and R. Faff, "Does simple pairs trading still work?," *Financial Analysts Journal*, vol. 66, no. 4, pp. 83–95, 2010.

[25] R. Kozhan and W. W. Tham, "Execution risk in high-frequency arbitrage," *Management Science*, vol. 58, no. 11, pp. 2131–2149, 2012.

[26] Y. Bao, Y. Lu, and J. Zhang, "Forecasting stock price by svms regression," in *Artificial Intelligence: Methodology, Systems, and Applications: 11th International Conference, AIMSA 2004, Varna, Bulgaria, September 2-4, 2004. Proceedings 11*, pp. 295–303, Springer, 2004.

[27] C. L. Dunis, J. Laws, and B. Evans, "Modelling and trading the gasoline crack spread: A non-linear story," *Derivatives Use, Trading & Regulation*, vol. 12, no. 1-2, pp. 126–145, 2006.

[28] C. L. Dunis, J. Laws, and B. Evans, "Modelling and trading the soybean-oil crush spread with recurrent and higher order networks: A comparative analysis," in *Artificial Higher Order Neural Networks for Economics and Business*, pp. 348–366, IGI Global, 2009.

[29] C. L. Dunis, J. Laws, P. W. Middleton, and A. Karathanasopoulos, "Trading and hedging the corn/ethanol crush spread using time-varying leverage and nonlinear models," *The European Journal of Finance*, vol. 21, no. 4, pp. 352–375, 2015.

[30] N. Huck, "Pairs selection and outranking: An application to the s&p 100 index," *European Journal of Operational Research*, vol. 196, no. 2, pp. 819–825, 2009.

[31] N. Huck, "Pairs trading and outranking: The multi-step-ahead forecasting case," *European Journal of Operational Research*, vol. 207, no. 3, pp. 1702–1716, 2010.

[32] T. D. Chaudhuri, I. Ghosh, and P. Singh, "Application of machine learning tools in predictive modeling of pairs trade in indian stock market.," *IUP Journal of Applied Finance*, vol. 23, no. 1, 2017.

[33] S. Guo and W. Long, "Pairs trading based on risk hedging: an empirical study of the gold spot and futures trading in china," in *Data Science: 6th International Conference, ICDS 2019, Ningbo, China, May 15–20, 2019, Revised Selected Papers 6*, pp. 488–497, Springer, 2020.

[34] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, pp. 281–297, Oakland, CA, USA, 1967.

[35] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping multidimensional data: Recent advances in clustering*, pp. 25–71, Springer, 2006.

[36] B. Do and R. Faff, "Are pairs trading profits robust to trading costs?," *Journal of Financial Research*, vol. 35, no. 2, pp. 261–287, 2012.

[37] B. D. Ripley, "Pattern recognition and neural networks," 1996.

Appendix A

Technical Indicators

1. Simple Moving Average (SMA):

   The SMA for a time series x over a period n is calculated as:

$$SMA_n(x) = \frac{1}{n}\sum_{i=1}^{n-1} x_{t-1} \tag{A.1}$$

2. Exponential Moving Average (EMA): The EMA for a time series x with smoothing factor α is calculated recursively as:

$$EMA_t(x) = \alpha\dot{x}_t + (1-\alpha)\dot{E}MA_{t-1}(x) \tag{A.2}$$

   where α is a smoothing factor between 0 and 1.

3. Weighted Moving Average (WMA): The WMA for a time series x with weights w_1, w_2, ..., w_n is calculated recursively as:

$$WMA_n(x) = \frac{\sum_{i=1}^{n} w_i\dot{x}_{t-i}}{\sum_{i=1}^{n} w_i} \tag{A.3}$$

4. Relative Strength Index (RSI): RSI is calculated as follows, where U is the average of gains over a specified period and D is the average of losses over the same period:

$$U = \frac{\sum_{i=1}^{n}(Close_i - Close_{i-1} for Close_i > Close_{i-1}}{n} \tag{A.4}$$

$$D = \frac{\sum_{i=1}^{n}(Close_{i-1} - Close_i for Close_i < Close_{i-1}}{n} \tag{A.5}$$

$$RSI = 100 - \frac{100}{1+\frac{U}{D}} \tag{A.6}$$

5. Moving Average Convergence Divergence (MACD): MACD is calculated as the difference between two EMAs:

$$MACD = EMA_a(x) - EMA_b(x) \tag{A.7}$$

where $EMA_A(x)$ is the EMA with a shorter period and $EMA_B(X)$ is the EMA with a longer period.

6. Signal Line (MACD Signal): The signal line for MACD is usually a 9-period EMA of MACD:

$$Signal\ Line = EMA_9(MACD) \tag{A.8}$$

7. MACD Histogram (MACD Hist): The MACD histogram is calculated as the difference between MACD and its signal line:

$$MACD_{Hist} = MACD - Signal\ Line \tag{A.9}$$

8. Momentum (MOM): The momentum for a time series x over a period is calcualted as the difference between the current price and the price n periods ago:

$$MOM_n(x) = x_t - x_{t-n} \tag{A.10}$$

9. Average True Range (ATR): The ATR for a time series with high (H), low (L) and close (C) prices over a period n is calculated as:

$$TR_n) = max(H_t - L_t, max(|H_t - C_{t-1}|, |L_t - C_{t-1})) \tag{A.11}$$

$$ATR_n = \frac{1}{n}\sum_{i=1}^{n} TR_i \tag{A.12}$$

Appendix B

Performance Figures

Figure B.1: Heatmap of cointegration

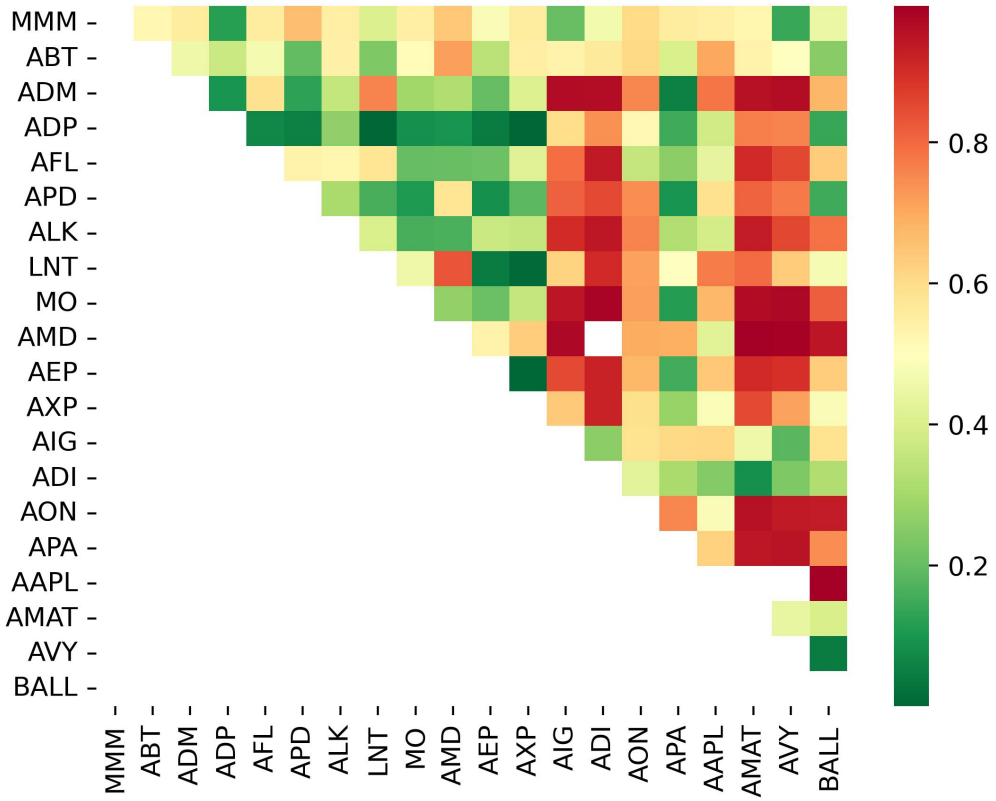


Figure B.2: Net value of pairs (AMAT&MAK) on ML-based model

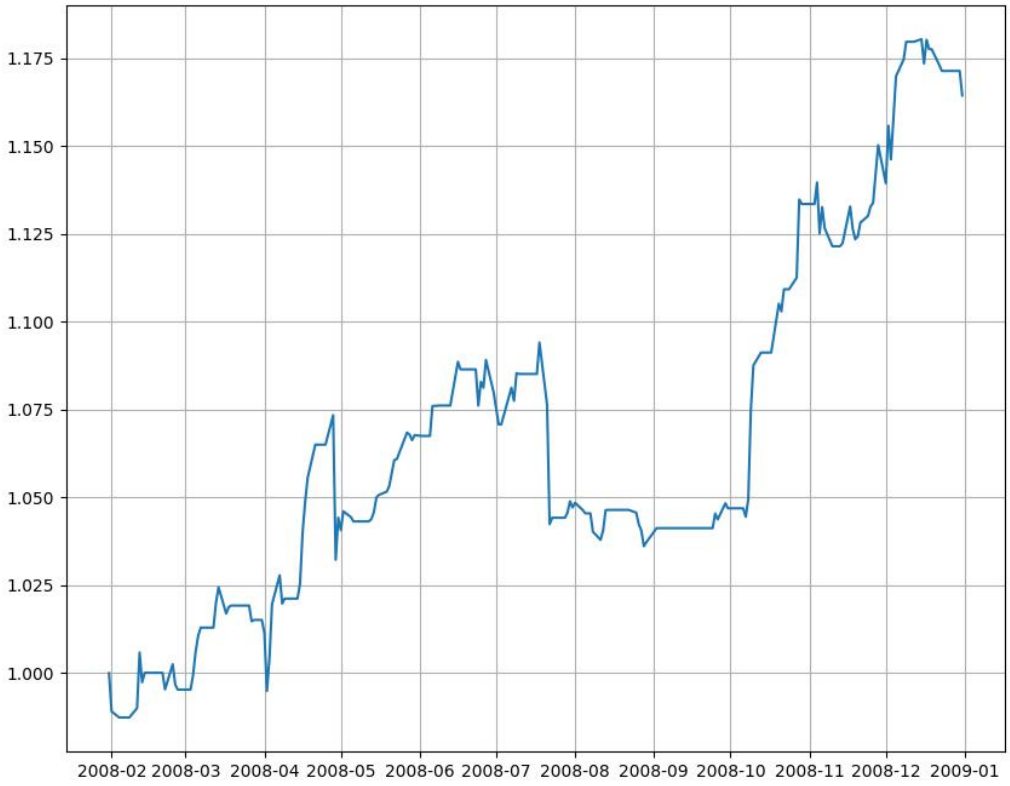


Figure B.3: ROC curve for Naive Bayes Classifier

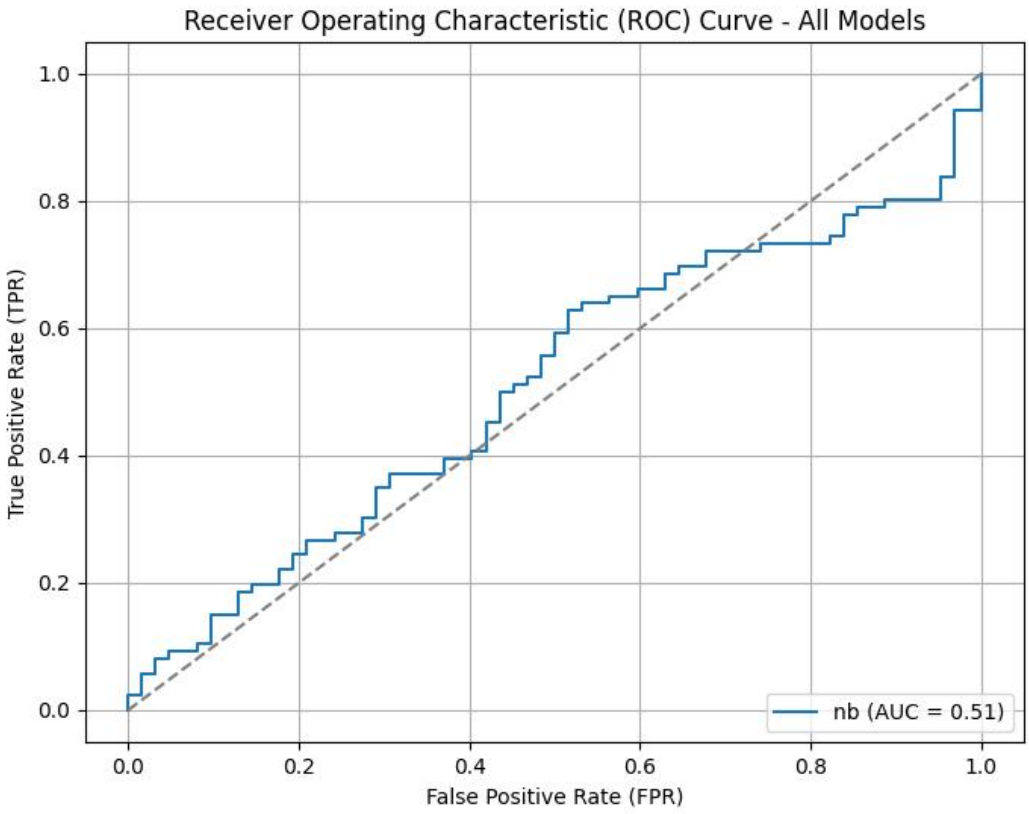


Figure B.4: Clusters Scatter Plot of Single-variable k-means algorithm

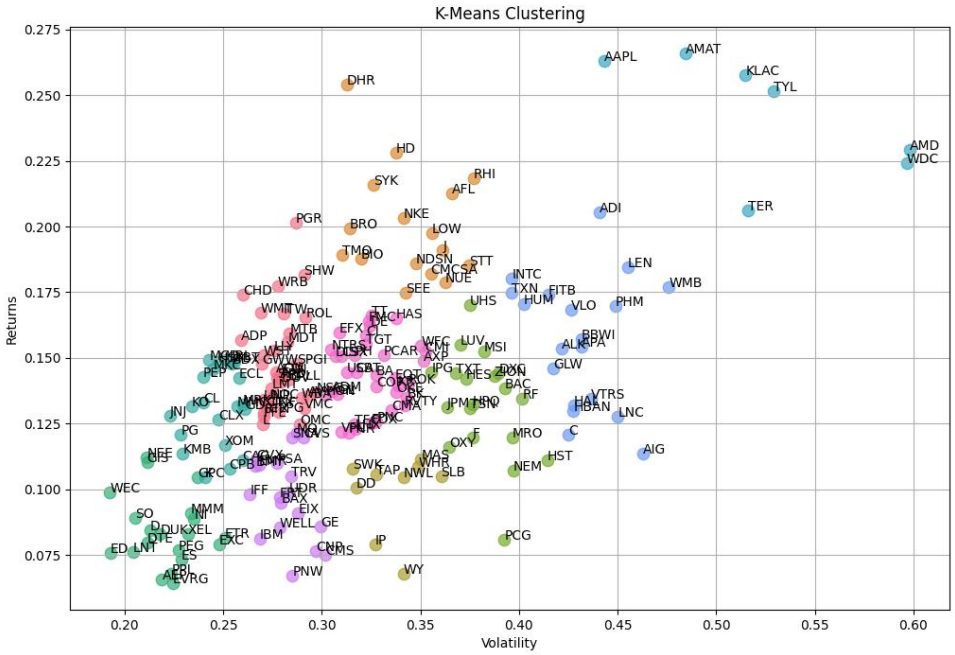


Figure B.5: Trading signals of pairs (ABT&ADI) on ML-based model