Detection and Application of Market Regime Using K-Means, HMM and Deep Learning

Shunyao Xu, Josie Yang, Helen Zhang, Ye Zhou

NEW YORK UNIVERSITY
BANK OF AMERICA

2023

Abstract

Detection and Application of Market Regime Using K-Means, HMM and Deep Learning

Shunyao Xu, Josie Yang, Helen Zhang, Ye Zhou

Our study innovatively combines the clustering capabilities of the Gaussian Hidden Markov Model (Gaussian HMM) and the K-means method with advanced machine learning techniques to predict market behavior. Initially, the K-means method segmented the market into three sections: bull, bear, and neutral, which the Gaussian HMM further refined for machine learning. Utilizing XGBoost, AdaBoost, and deep learning models like LSTM and GRU, we captured time-related market patterns using a 30-day rolling window. Early testing informed a simplified approach to prevent overfitting, ensuring our models were robust yet adaptable. XGBoost effectively analyzed average market returns, while LSTM and GRU excelled at assessing market volatility. Our adaptive investment strategies, informed by these insights, outperformed both static strategies and the S&P 500 benchmark, underscoring the potential of integrating diverse machine learning methods for market trend analysis.

Key Words: Market Regime Detection, K-Means, Gaussian Hidden Markov Model, Deep Learning Hybrid Approach, Dynamic Portfolio Construction, Asset Allocation

Table of Contents

Introduction

As global financial markets grow in complexity, the identification and understanding of distinct market regimes have become critical for effective risk management, portfolio optimization, and strategic decision-making. Market regimes, representing different phases in market conditions such as bull, bear, or sideways markets, can substantially impact the performance of investment strategies. Traditional methods often struggle to adapt to these regime shifts, necessitating the application of advanced machine learning techniques for more dynamic and flexible models.

Machine learning offers a suite of powerful tools to tackle these challenges, including clustering and classification techniques. In the realm of unsupervised learning, clustering algorithms like K-means can help identify and group similar periods of market behavior, effectively partitioning historical data into distinct regimes. By observing commonalities within clusters, investors can gain insights into prevailing market conditions, thus informing their strategy under similar future circumstances.

On the other hand, Hidden Markov Models (HMMs) provide a unique blend of unsupervised and supervised learning approaches for market regime detection. These probabilistic models hypothesize that the market at any given point in time is in one of a few hidden states or regimes, and transitions between these states follow probabilistic rules. The beauty of HMMs lies in their ability to capture the temporal dependencies and inherent volatility of financial markets, making them well-suited for regime identification.

Moreover, supervised learning models such as XGBoost, AdaBoost, LSTM or GRU, can also play a critical role when we have pre-labeled regimes. They can be trained to classify different market states based on historical data, allowing these models to predict future market regimes.

By leveraging these machine learning techniques, we can classify and predict market regimes more accurately and robustly. This not only enhances our understanding of market dynamics but also empowers us to design more effective, regime-adaptive investment strategies. In the subsequent sections, we will delve into the technical intricacies of K-means clustering, Hidden Markov Models, and other machine learning techniques, and their application in the classification and detection of market regimes.

Literature Review

Identification of market regimes is an unsupervised process, and there is a wide range of techniques that can help determine the current market state based on historical market data. Among all these techniques, two mainstream methods in finance are widely used to detect market regimes: Hidden Markov Models (HMM) under statistical approaches, and a series of unsupervised machine learning models.

Among the unsupervised learning models, the Gaussian Mixture Model (GMM) is one of the unsupervised machine-learning-based methods used to determine market regimes (Botte & Bao, 2021). The GMM will fit various Gaussian distributions to capture different parts of the asset's return distribution in its time series data. Each of those distributions would have its properties, like means and volatilities, and be recognized as a particular kind of regime. Other studies have used the k-means clustering model to classify the regimes that existed on the market. The classical k-means clustering algorithm was applied as part of the hybrid learning framework by Peter Akioyamen, etc. (Akioyamen, Tang, & Hussien, 2021) to detect points in time that share underlying common characteristics, grouping the time series data into distinct regimes. Based on this, Blanka Horvath, etc. (Horvath, Issa, & Muguruza, 2021) proposed a modified version of the classical k-means clustering algorithm - the Wasserstein k-means algorithm - which uses the Wasserstein Distance as a metric to classify segments of the historical evolution of market returns into distinct regimes. The results proved that the WK-means approach is more robust than classical k-means clustering in the sense that it does not depend on the modeling assumptions of the underlying time series.

The hidden Markov Model (HMM) is a statistical model that is used to describe the probabilistic relationship between a sequence of observations and a sequence of hidden states. Since Hamilton proposed employing a regime-switching model to identify economic cycles using the GNP series (Hamilton, 1989), HMMs have been widely employed in finance to create regime-based models. HMMs simultaneously capture skewness, kurtosis, and time-varying correlations from financial return series, among other characteristics, as noted by Andrew Ang and Allan Timmermann (Ang & Timmermann, 2011). They also offer accurate approximations for processes for which the underlying model is unknown (Nystrup, Madsen, & Lindström, 2015). Based on HMMs, some scholars have raised a refined model called the hidden semi-Markov model (HSMM). By enabling any probability distribution to be used for the state duration time, the hidden semi-Markov model (HSMM) generalizes the HMM. The HSMM is therefore better adapted to capture the movements of a financial market. However, there are still few econometric applications of HSMMs to financial time series data. In the study of HSMMs, Jan Bulla and Ingo Bulla fitted two HSMMs to daily return series from 18 Pan-European sector indices, and the results showed a significantly improved fit of the autocorrelation function (Bulla & Bulla, 2006). In later years, Zhenya Liu and Shixuan Wang also employed the HSMM to detect various regimes in the stock market (Liu & Wang, 2017). It is a three-state (bull market, bear market, and sideways trending market) hidden semi-Markov model (HSMM) used to explain the time-varying distribution of the Chinese stock market returns since 2005.

Few experiments exist using hybrid machine learning methods for regime detection, and the paper by Akioyamen, etc. (Akioyamen, Tang, & Hussien, 2021) demonstrates the effectiveness of using unsupervised learning for finding the optimal regime number and supervised methods in classifying market regimes. However, in the clustering step, no attempt was made to use a hierarchical or density-based approach, and there was not enough improvement in k-means for the features of the time-series data. In the part of classification with supervised learning, some traditional models such as LDA, QDA, and logistic regression have been tried. Benhamou, etc. (Benhamou, Guez, Ohana & Saltiel, 2021) also tested the method of gradient boosting decision trees to classify markets into two regimes, normal and crisis, and a deep learning approach is used for comparison. In the data of stock index futures, these methods consistently show more than 85% accuracy, but we hope to expand to more asset varieties. The applicability test of regime testing methods is also an important part. Many studies have examined the profitability of regime-based asset allocation (RBAA). Nystrup et al. (Nystrup et al., 2017) compared the performance of portfolios under RBAA with static 60/40 long-only benchmark portfolios in a diversified asset universe. The empirical results show that HMM-based regime-based asset allocation is more profitable. We will also apply a similar comparison to test the profitability of our regime clustering and classification approach in asset allocation. We also plan to compare the combined performance of mainstream HMM-based methods and hybrid machine learning methods.

Data

In this study, data are needed both for the market regime classification process and for the investment strategy exploration process. For the market regime classification process, we focused on data that contains specific features of different market regimes. The datasets we selected span a long period of time to contain as many market regimes as possible. The datasets are also liquid, constantly available, and easy to access. For the investment strategy exploration process, we selected relative industries and assets to form our portfolio after we investigated the correlations and patterns of the market regimes.

2.1 Data Selection

For the market regime classification process, we used 25 years of daily data spanning from January 1st, 1998, to December 31, 2022. A wide range of macroeconomic and technical features have been used to predict future market trends in past literature. For example, Zhong and Enke (2017) use past returns of major stock indices, T-bills rates, risky interest rates, term and default spreads, exchange rates between major currencies, and trading volumes of major stock indices to predict the price movement direction of S&P 500 ETF. In our research, four groups of data are considered as predictors of future market states. Below is a list of data used and the corresponding rationales. Corresponding datasets are retrieved from Yahoo Finance (https://finance.yahoo.com/) and Bloomberg.

- **Interest rates:** 1Y Treasury yield, 5Y Treasury yield, 10Y Treasury yield, 30Y Treasury yield, 10Y-3M Treasury yield spread, 10Y-2Y Treasury yield spread, 3M LIBOR rate. Interest rates are economic variables that are most correlated with the status of economic activities. On one hand, interest rates can reflect the monetary policy of an economy, which is critical to changes of market states; on the other hand, demand, and supply of funds in the market also influence interest rates. Thus, past interest rates can have strong predicting effects on market states.
- **Exchange rates:** EUR/USD, JPY/USD, GBP/USD. The globalization of capital markets makes exchange rates strongly correlated with states of economies. A strong economy is highly correlated with a strong performance of the currency of this economy. Thus, exchange rates between the United States and other major economies can be used as indicators of the market states in the United States.
- **Stock indices and VIX index:** S&P 500 and VIX. The S&P 500 is often considered the best representation of the U.S. stock market and is used as a benchmark for the broader market's performance. Its movements can provide insights into the overall health and direction of the U.S. equity market. The VIX index is a strong indicator of volatility in the market.
- **Commodities prices:** WTI Crude Oil Futures Contract, S&P GSCI gold index. Oil prices have substantial influences on different sectors of the economy, including manufacturing, transportation, and utilities. Fluctuations in oil prices can lead to changing market states. Gold price is also a good indicator of sentiments in the market because gold is an asset for risk-aversion.

2.2 Data Processing

Data quality plays a pivotal role in shaping prediction accuracy, making data preprocessing an indispensable step. Given the minimal presence of missing values in our dataset, we opted to eliminate null entries to maintain data integrity. While price indices provide valuable insights, daily returns offer a more granular and accurate depiction of market states. Consequently, we transformed each price index into its corresponding daily returns. Additionally, we also composed technical indicators like MA10 (10-day moving average of S&P 500 index return) and MA50 (50-day moving average of the S&P 500 index return). The MA smooths out daily volatility, reducing noise and providing a more accurate representation of the underlying trend, it helps in discerning sustained movements, making them essential for trend identification. We used both a price return and a price volatility with rolling window size equal to one year (252 days). To optimize the performance of machine learning algorithms, we employed various feature engineering techniques to curate a refined feature set. Moreover, as the heatmap shown below, we assessed the correlation among all variables to ensure there was no pronounced interdependence, guaranteeing that our dataset maintains its robustness and avoids potential multicollinearity issues.

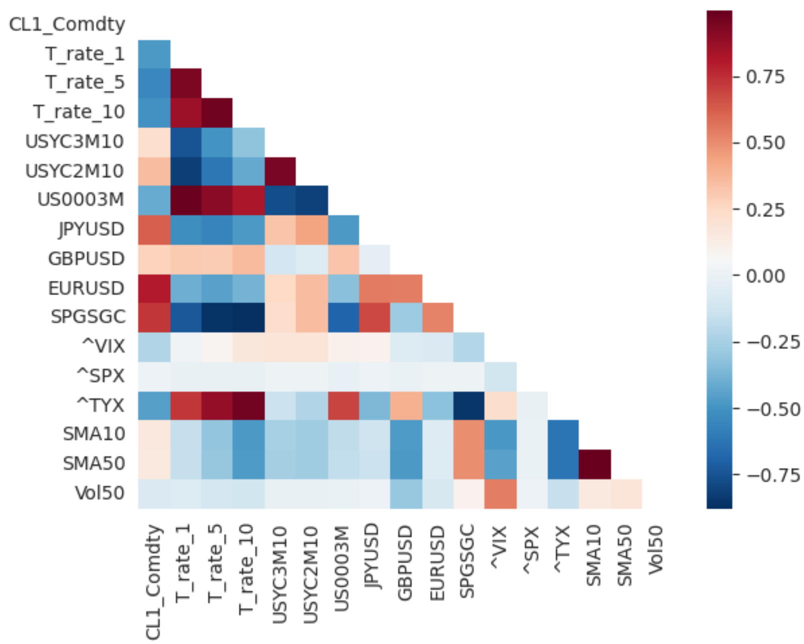

Figure 2a: The Correlation between all datasets

Methodologies

3.1 K-Means

K-means clustering involves partitioning n observations into K clusters by utilizing vector quantization. The primary objective is to assign each observation to the cluster whose mean or centroid is closest, acting as a representative prototype. This process seeks to minimize the total squared distances between data points and their respective cluster centroids. As a result, it produces clusters that exhibit both internal uniformity and distinctiveness from one another.

Given a training set $x^{(1)}, \ldots, x^{(m)}$, the clustering problem involves organizing the data into coherent "clusters." Each data point $x^{(i)}$ is represented as a feature vector in R^n, and there are no corresponding labels $y^{(i)}$, defining this as an unsupervised learning task. The objective is to determine k centroids and assign a label $c^{(i)}$ to each datapoint. The procedure for the k-means clustering algorithm is outlined below:

1. Initialize **cluster centroids** $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$ randomly.
2. Repeat until convergence: {

   For every i, set
   $$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$
   For each j, set
   $$\mu_j := \frac{\sum_{i=1}^m 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)} = j\}}.$$
   }

3.2 Gaussian Hidden Markov Model

A Gaussian Hidden Markov Model (GHMM) is a probabilistic model that combines the concepts of Hidden Markov Models (HMMs) and Gaussian distributions. It is used to model time series data where the underlying states are hidden, and each observed data point is assumed to be generated from a Gaussian distribution associated with the current hidden state.

The formal structure of an HMM is shown below, which can be conceptualized as a chain of mixture models. The "hidden" states are denoted by S_t and the observed values are denoted by O_t , where t is a specific point in time. ∅ represents the output model defining $P(s_i)$, and θ is the transition matrix specifying $P(s_{t-1})$.

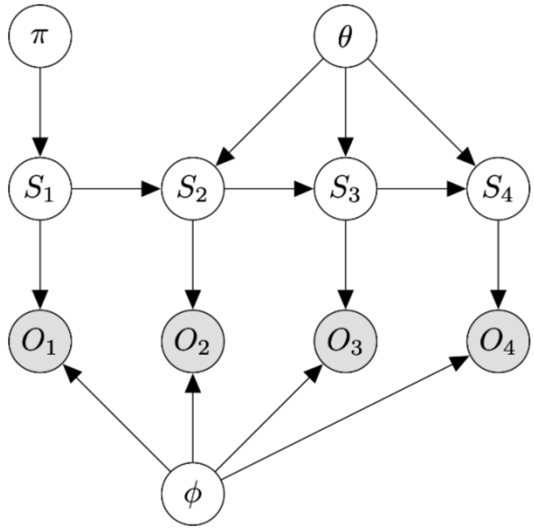

Figure 3a: Graphical model for an HMM with T=4 timesteps

Specifically, Gaussian Hidden Markov Model includes the following components:

- States: There are a finite number of hidden states $S = \{s_1, s_2, \dots, s_N\}$.
- Observations: At each time step t, an observation o_t is emitted from a Gaussian distribution associated with the current state.
- State Transition Probabilities: The probability of transitioning from state s_i to state s_j is given by A_{ij}.

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{bmatrix}$$

- Initial State Probabilities: The probability of starting in state s_i is given by π_i.

$$\pi = [\pi_1 \;\; \pi_2 \quad \dots \quad \pi_N]$$

- Emission Probabilities: The emission probability of observing o_t given that the system is in state s_i is described by a Gaussian distribution with mean μ_i and covariance matrix Σ_i.

$$P(s_i) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_i|}} exp\ exp \left(-\frac{1}{2} (o_t - \mu_i)^T \Sigma_i^{-1} (o_t - \mu_i) \right)$$

3.3 Machine Learning Models

3.3.1 LSTM

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture that addresses the challenges of learning long-range dependencies in sequential data.

LSTMs are designed to capture and retain information over extended time intervals, making them well-suited for tasks like natural language processing and time series prediction.

At the heart of an LSTM cell is its ability to control information flow through three key gates: the input gate, the forget gate, and the output gate. These gates regulate the flow of information into and out of the cell, allowing LSTMs to selectively update and maintain memory.

The input gate (i_t), also controlled by a sigmoid function, determines which new information should be added to the cell state.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$$

The forget gate (f_t), governed by the sigmoid activation function, decides what information from the previous cell state should be discarded.

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right)$$

A tanh activation function calculates a candidate vector of new values that could be added to the cell state.

$$\widetilde{C_t} = tanh\ tanh\ (W_C \cdot [h_{t-1}, x_t] + b_C)$$

The combination of the input and forget gates, along with the candidate vector, helps update the cell state.

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C_t}$$

The output gate, determined by another sigmoid activation, controls the output of the LSTM cell. It decides which parts of the cell state will be exposed as the hidden state (h_t) at the current time step.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

The final hidden state (h_t) contains the information that the LSTM cell has deemed relevant for the current time step.

$$h_t = o_t \odot tanh\ tanh\ (C_t)$$

LSTMs demonstrate remarkable proficiency in capturing both short-term and long-term dependencies in sequences. This is attributed to their gating mechanisms, which enable the selective preservation and updating of information. This ability to manage memory and information flow makes LSTMs a powerful choice for tasks requiring context and temporal understanding.

3.3.2 GRU

A Gated Recurrent Unit (GRU) is a type of recurrent neural network (RNN) architecture that addresses some of the limitations of traditional RNNs, such as the vanishing gradient problem and the difficulty in learning long-range dependencies in sequential data. GRUs were introduced as a variation of the more complex Long Short-Term Memory (LSTM) networks, offering similar capabilities with a simpler structure.

At its core, a GRU is designed to capture and remember relevant information from past time steps while also allowing the network to forget irrelevant or outdated information. This is achieved through two key mechanisms: the update gate and the reset gate. These gates control the flow of information through the GRU cell, enabling it to selectively update its hidden state based on the current input and the previous hidden state.

A typical GRU unit consists of the following components:

- Update Gate (z_t): This gate determines how much of the previous hidden state should be mixed with the candidate activation for the current time step.
- Reset Gate (r_t): This gate decides how much of the previous hidden state should be forgotten, making way for new information.
- Candidate Activation ($\widetilde{h_t}$): This is the new candidate activation that combines information from the current input and the previous hidden state.
- New Hidden State (h_t): The final hidden state is a combination of the previous hidden state and the candidate activation, controlled by the update gate.

The equations for a single GRU unit can be described as follows:

$$z_t = sigmoid\left(W_z * x_t + U_z * h_{(t-1)}\right)$$

$$r_t = sigmoid\left(W_r * x_t + U_r * h_{(t-1)}\right)$$

$$\tilde{h}t = tanh\left(W_h * x_t + U_h * \left(r_t * h(t-1)\right)\right)$$

$$h_t = (1 - z_t) * h_{(t-1)} + z_t * \widetilde{h_t}$$

In these formulas, x_t represents the input at time step t, $h_{(t-1)}$ is the previous hidden state, and W, U are weight matrices associated with the different gates and candidate activation. The sigmoid and tanh functions introduce non-linearity and control the flow of information through the gates.

The GRU's ability to update and reset information based on gating mechanisms makes it well-suited for capturing sequential patterns and dependencies in data.

3.3.3 XGBoost

XGBoost (Extreme Gradient Boosting) is a popular machine learning algorithm known for its effectiveness in handling a wide range of predictive modeling tasks. It belongs to the ensemble learning family and is based on the concept of gradient boosting. XGBoost combines the strengths of decision trees and gradient boosting to create a highly accurate and robust predictive model.

XGBoost constructs an ensemble of decision trees sequentially, with each new tree aiming to correct the errors made by the previous ones. The final prediction is the sum of predictions from all individual trees, weighted by a learning rate. The XGBoost algorithm minimizes a loss function that quantifies the difference between predicted values and actual target values, using a combination of the following components:

- Objective Function (L): The objective function defines the loss to be minimized during training. For regression tasks, it often uses the mean squared error (MSE), while for classification tasks, it may use the logistic loss or softmax loss, depending on the number of classes. The objective function typically comprises a data loss term and a regularization term to prevent overfitting.

$$L = \sum_{i=1}^{n} l(y_i, \widehat{y_i}) + \sum_{k=1}^{K} \Omega(f_k)$$

Here, $l(y_i, \widehat{y_i})$ represents the data loss for the i-th example, $\widehat{y_i}$ is the predicted value, $\Omega(f_k)$is the regularization term for the k-th tree, and K is the total number of trees.

- Decision Trees: Each decision tree in XGBoost captures different patterns and residuals in the data. XGBoost uses a modified version of the gradient boosting algorithm to build these trees. The trees are shallow and constrained to prevent overfitting. Each leaf node of a tree contains a score that contributes to the final prediction.
- Gradient Boosting Updates: XGBoost employs gradient boosting to iteratively improve the ensemble of trees. It computes the negative gradient of the loss function with respect to the current ensemble's predictions and fits a new tree to this residual error.
- Regularization $\left(\Omega(f_k)\right)$: Regularization terms are added to the objective function to control the complexity of the individual trees. These terms prevent overfitting by penalizing overly complex trees.

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda|w|^2$$

Here, T represents the number of leaf nodes in the k-th tree, γ controls the depth of the tree, λ controls the L2 regularization on leaf scores, and w are the scores at the leaf nodes.

By optimizing the objective function through a combination of data-driven and regularization terms, XGBoost creates a strong predictive model that is robust, interpretable, and capable of handling complex relationships within the data.

3.3.4 AdaBoost

AdaBoost is a popular ensemble learning algorithm that focuses on improving the performance of weak learners (typically simple models like decision stumps) by combining their predictions. It assigns different weights to training samples in each iteration, with higher weights on misclassified samples. This allows the algorithm to pay more attention to difficult examples, thus creating a strong classifier from the ensemble of weak learners.

The mathematical formulation of AdaBoost includes the following:

- Weighted Error (ϵ_t) In each iteration, AdaBoost calculates the weighted error of the current weak learner on the training data. The weight of each sample is adjusted based on whether it was correctly classified by the current learner.

$$\epsilon_t = \frac{\sum_{i=1}^{N} w_{i,t} \cdot I\left(y_i \neq h_t(x_i)\right)}{\sum_{i=1}^{N} w_{i,t}}$$

  Here, N is the number of training samples, $w_{i,t}$ is the weight of the i-th sample at iteration t, $h_t(x_i)$ is the prediction of the weak learner at iteration t for the i-th sample x_i, and $I(\cdot)$ is the indicator function.
- Classifier Weight (α_t) AdaBoost calculates the weight of the current weak learner based on its error rate. A classifier with a lower error rate will have a higher weight in the ensemble.

$$\alpha_t = \frac{1}{2} ln\, ln\, \left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

- Sample Weights Update ($w_{i,t+1}$): After each iteration, AdaBoost updates the sample weights, assigning higher weights to misclassified samples, so that the next weak learner focuses more on these samples.

$$w_{i,t+1} = w_{i,t} \cdot exp\, exp\, \left(-\alpha_t y_i h_t(x_i)\right)$$

- Final Strong Classifier $\left(H(x)\right)$: The final strong classifier is a weighted combination of the weak learners' predictions, where the weights are determined by the classifier weights α_t.

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

The AdaBoost algorithm continues to iterate through these steps for a predefined number of iterations or until a certain threshold is reached. The final strong classifier combines the predictions of the weak learners with their corresponding weights, effectively creating a powerful ensemble model that excels at handling complex classification problems.

Market Regime Detection

In this paper, we used K-means and Gaussian Hidden Markov Model to separately detect the two-state and three-state market regimes. The details are shown in the following.

4.1 K-means Method

4.1.1 3-state K-means

At first, we used the K-means method to detect the market regimes since it is simple. We tried to use the Elbow method and the Silhouette Coefficient to find the best regime number respectively. The basic data for the K-means method is VIX and SPX, besides that, we also composed some technical indicators like MA10 (10-day moving average of the return of S&P 500 index) and MA50 (50-day moving average of the S&P 500 index). We tried to detect the regime from the beginning of 1998 to the end of 2022. The results are as follows:

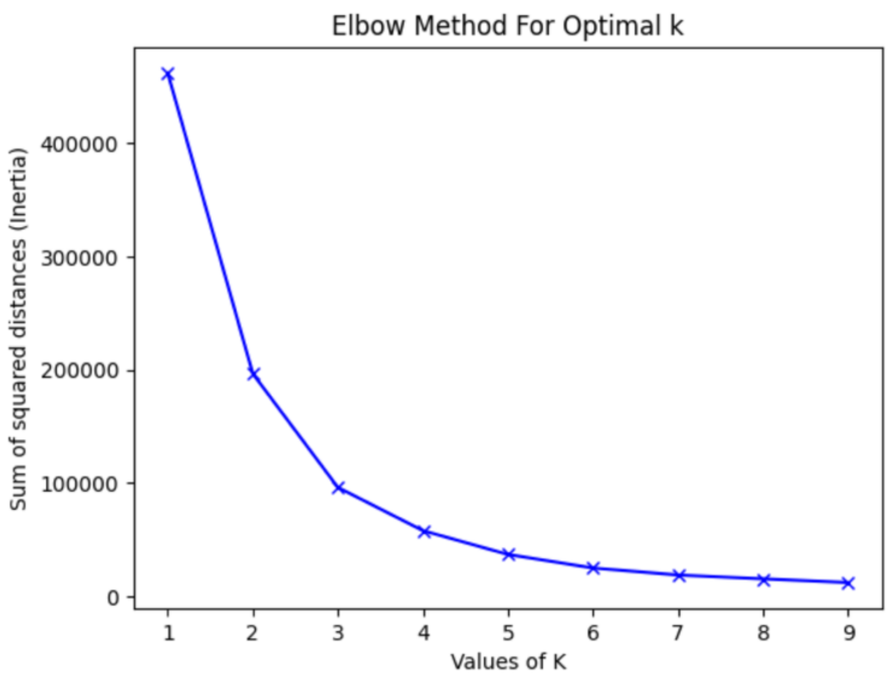

Figure 4a: The Elbow Method for Optimal k

From the Elbow method plot, we can find that as the number of clusters increases, the variance within each cluster tends to decrease. This is because when there are more clusters, each one of them will have fewer constituent instances, and they will be closer to the centroids of their respective clusters. However, the decrease in this intra-cluster variance reduces at a certain point, thus making the additional cluster redundant. This point, where the rate of decrease sharply changes, is called the 'elbow'. The picture shows that the elbow for the optimal k is 3.

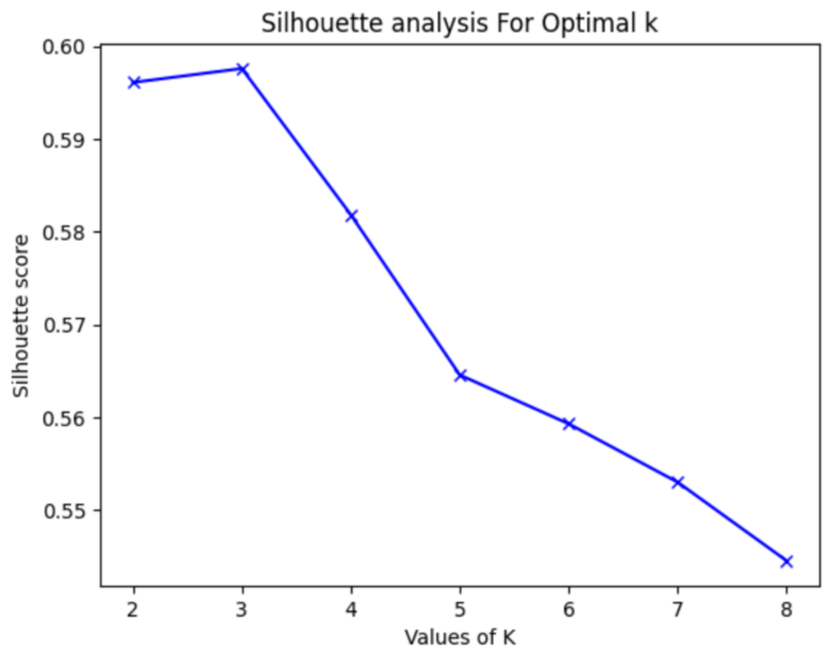

Figure 4b: The Silhouette analysis for Optimal k

As for the plot above, the Silhouette coefficient provides a succinct graphical representation of how well each object has been classified. It considers both the cohesion (how close objects are to other objects in the same cluster) and the separation (how far objects are from objects in other clusters). A value close to 1 implies that the data point is well matched to its own cluster, so by evaluating the average silhouette coefficient for various numbers of clusters, we can find that the optimal k is 3. Combining the above two results, we found the best regime number is 3.

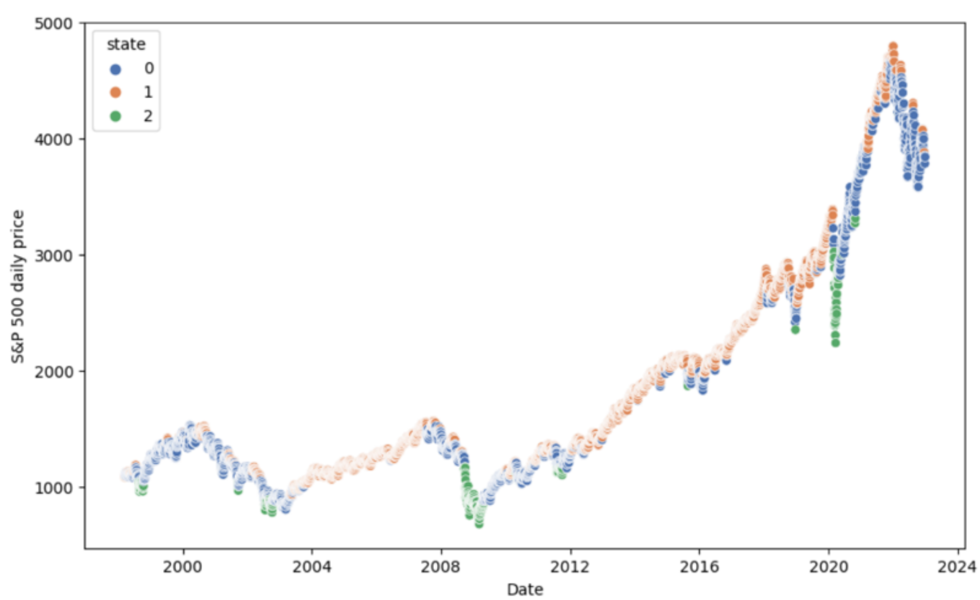

Figure 4c: Market Regimes of 3-state K-means

Upon applying K-means clustering to our dataset, we identified not just two (bull and bear), but three distinct market regimes. This result suggests a more intricate market structure than traditionally understood. The explanation of three markets are as follows:

Neutral Market (State 0): This is the third regime that our clustering algorithm possibly identified. In a sideways market, prices of securities remain within a tight range for an extended period. This market neither classifies as a bull nor a bear market but can be seen as a 'resting phase'.

Bull Market (State 1): This is a phase where market prices are rising or are expected to rise. The length and robustness of a bull market are often driven by a strong economy, low unemployment, and high investor confidence.

Bear Market (State 2): Characterized by a prolonged period of falling stock prices, typically by 20% or more from recent highs. Economic slowdowns, rising unemployment, and declining investor confidence often accompany bear markets.

The identification of three market regimes instead of the traditional two suggests that the market, at times, might not be distinctly bullish or bearish but can be in a transitional or indecisive state. Distinguishing three distinct market regimes—bull, bear, and neutral—presents a richer, more comprehensive view of market dynamics than the conventional binary division of just bull and bear phases. This threefold classification not only captures periods of clear upward or downward momentum but also acknowledges the often-overlooked phases of market consolidation or indecision. Such a sideways or neutral market can be indicative of a transitional period where external factors, investor sentiment, and economic indicators are mixed, leading to a lack of a clear trend. By identifying this third regime, investors gain a crucial tool for navigating these murky waters. It allows for the development and deployment of specific strategies tailored for uncertain conditions, potentially maximizing returns and minimizing risks. This nuanced approach, therefore, equips investors with a more holistic understanding, ensuring that they are not caught off guard by market shifts and can adjust their portfolios proactively in response to the intricate dance of market forces.

4.1.1 3-state K-means

Although three regimes provide a more detailed landscape of market behavior, considering that the traditional two-regime model offers a simple and broad overview of market conditions, we also had a try on it. From the picture below, we can obviously find that state 0 is the bull market and state 1 is the bear market, which provides a high-level view of market conditions.

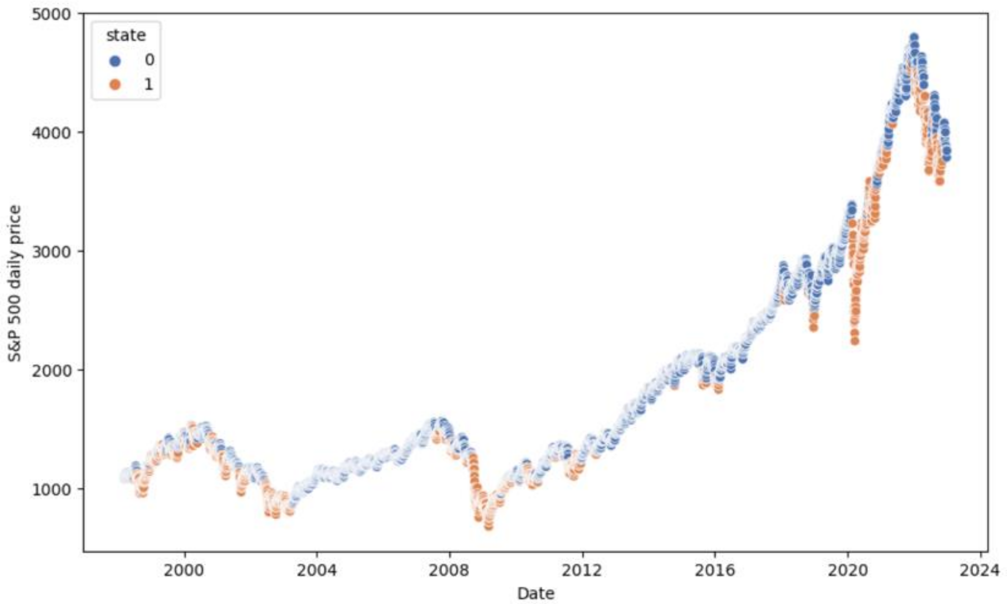

Figure 4d: Market Regimes of 2-state K-means

Given the limited features in the dataset, the use of the K-means clustering method doesn't offer a robust representation of the underlying patterns. K-means, as a clustering approach, relies heavily on the features provided to discern and create distinct groupings. When we work with a limited feature set, the resulting clusters might not capture the true complexities and nuances of the data.

Interestingly, as we increased the number of features to provide a more comprehensive view, the performance of K-means did not improve as expected. This may be attributed to inherent limitations of the K-means algorithm itself. When the feature space becomes too vast, K-means might struggle to find clear centroids, especially if the data does not naturally segregate into distinct clusters. This can lead to overlapping or poorly defined clusters that don't offer meaningful insights.

Given these challenges, we decided not to utilize the market regimes classified by K-means for deeper machine learning training or back testing. Instead, the K-means clustering was employed as a preliminary exploration, providing a basic categorization without delving into more complex analytical methods.

4.2 Gaussian Hidden Markov Model

4.2.1 3-state Gaussian HMM based on Moving Average Return

We assume there are three different regimes on the market based on asset's price: Bull market, Bear market and Neutral market. Under this section, we used price return with rolling window size equals to one year (252 days). We also highlighted three largest U.S. stock crashes since 1998 in the regime-result graph in order to see whether our trained Gaussian HMM model can capture the market crisis precisely.

The underlying features input into the HMM model is the annual price return of 1-year Treasury yield, 5-year Treasury yield, 10-year Treasury yield, 30-year Treasury yield, 10 year-3 month Treasury yield spread, 10 year-2 year Treasury yield spread, 3-month LIBOR rate, JPY/USD exchange rate, GBP/USD exchange rate, EUR/USD exchange rate, S&P 500 index, S&P GSCI gold index and crude oil index from 1998-01-01 to 2022-12-31.

Figure 4e: Market Regimes of 3-state 252-day Rolling Gaussian HMM based on Return

Based on the market regime plot above, we can see that State 0 refers to the Bull market, State 1 refers to the Bear market and State 2 refers to the Neutral market. The 3-state Gaussian HMM model can grasp most of financial crises happened through history and describe the different market states rather clear and correct: when the S&P 500 stock index goes up, model shows the regime as Bull market; when the S&P 500 stock index draws down, model shows the regime as Bear market; and when the S&P 500 stock index fluctuates, model shows the regime as Neutral market. In conclusion, the 3-state Gaussian HMM model with 252-day rolling window size based on asset's return performs well in detecting and classifying market regimes.

4.2.2 3-state Gaussian HMM based on Moving Average Volatility

We assume there are three different regimes on the market based on asset's volatility: High-volatility market, Low-volatility market, and Medium-volatility market. Under this section, we used asset price's volatility plus one volatility index with rolling window size equals to one year

(252 days). We also highlighted three largest U.S. stock crashes since 1998 in the regime-result graph to see whether our trained Gaussian HMM model can capture the market crisis precisely.

The underlying features input into the HMM model is the annual price volatility of crude oil, gold, S&P 500 stock index, 30-year Treasury yield and VIX index itself from 1998-01-01 to 2022-12-31.

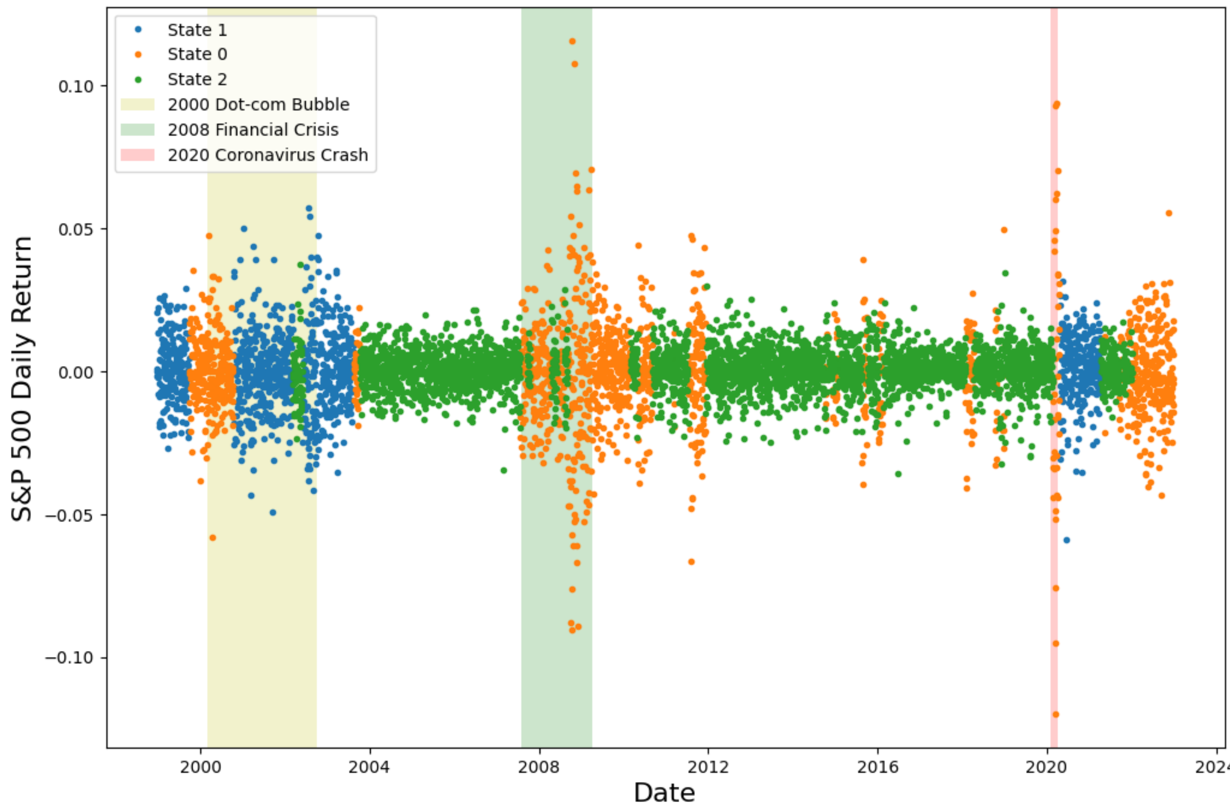

Figure 4f: Market Regimes of 3-state 252-day Rolling Gaussian HMM based on Volatility

Based on the market regime plot above, we can see that State 0 refers to the High-volatility market, State 1 refers to the Medium-volatility market and State 2 refers to the Low-volatility market. The 3-state Gaussian HMM model captured historical crises on the stock market precisely, as the regime shows high market volatility (State 0) when crisis events happen. Also, the low volatility period of the S&P 500 stock index is successfully detected as Low-volatility market (State 2). In conclusion, the 3-state Gaussian HMM model with 252-day rolling window size based on asset’s volatility performs well in detecting and classifying market regimes.

Backtest Result

5.1 Mean-Variance Optimization

Portfolio management has always sought ways to maximize returns while minimizing risk. In this quest, Harry Markowitz's Modern Portfolio Theory (MPT) introduced the concept of mean-variance optimization as a pivotal breakthrough in the 1950s.

At its core, mean-variance optimization focuses on two main aspects:

- **Portfolio Expected Return (Mean)**: The anticipated return of a portfolio, calculated as a weighted average of the expected returns of its individual assets.

$$E(R_p) = \sum_{i=1}^{n} \omega_i E(R_i)$$

where ω_i denotes the weight of asset i in the portfolio and $E(R_i)$ is the expected return.

- **Portfolio Risk (Variance)**: The variability of the portfolio returns, calculated using the variances and covariances of its assets.

$$\sigma_p^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \omega_i \omega_j \sigma_i \sigma_j \rho_{ij}$$

where σ_i and σ_j are the standard deviations of returns for assets i and j, and ρ_{ij} is their correlation coefficient.

Given a set of assets and their expected returns, variances, and covariances, the objective is to find the portfolio weights that maximize portfolio return while minimizing portfolio risk. The mathematics formula is:

$$min\, \lambda \boldsymbol{\omega^T \Sigma \omega} - \boldsymbol{\omega^T R_p}$$

$$s.t. \boldsymbol{\omega^T 1} = 1$$

Where:

$\boldsymbol{\omega^T R_p}$ is the expected return of the portfolio.
$\boldsymbol{\omega^T \Sigma \omega}$ is the portfolio variance.
λ is the risk aversion coefficient. As λ increases, the investor becomes more risk-averse, and the portfolio variance contributes more to the objective function.

5.2 Dynamic Optimal Portfolio Construction

5.2.1 Optimal Portfolio Construction under Bull/Bear Regimes

To construct our optimal portfolio, we chose the diversified underlying assets as below:

Table 5a: Underlying Assets in the Portfolio

Index	**Full Name**	**Description**
MSCI	Morgan Stanley Capital International	captures large and mid-cap representation across 23 Developed Markets (DM) countries
EEM	iShares MSCI Emerging Markets ETF	captures large and mid-cap representation across 24 Emerging Markets (EM) countries
GOVT	iShares U.S. Treasury Bond ETF	tracks U.S. Treasury bonds across various maturities
LQD	iShares iBoxx $ Investment Grade Corporate Bond ETF	tracks U.S. investment-grade corporate bonds
HYG	iShares iBoxx $ High Yield Corporate Bond ETF	tracks U.S. high-yield (junk) corporate bonds
GLD	SPDR Gold Trust	tracks the price of gold; it offers exposure to gold without owning the physical metal
WTI	West Texas Intermediate	A grade of crude oil used as a benchmark in oil pricing, primarily in the U.S
DJCI	Dow Jones Commodity Index	tracks a diversified group of commodities; used as a benchmark for commodity markets

First, we calculated each underlying asset's expected return under bull/ bear/ neutral regimes. The graph below shows the result of expected returns of portfolio assets grouped by different market states. State 0 stands for Bull market, State 1 stands for Bear market and State 2 stands for Neutral market. From the graph, we can see that most asset classes under the bull market regime have positive expected returns and most asset classes under the bear market regime have negative expected returns, which accord with our intuitive feelings.

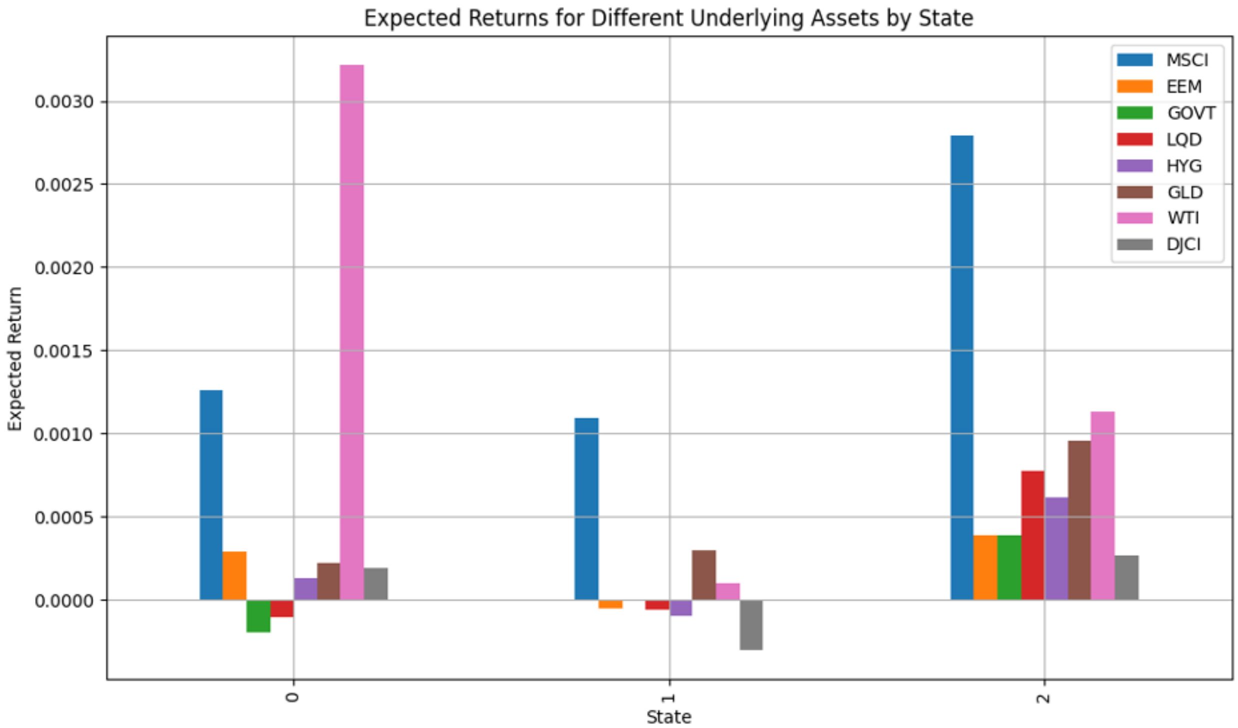

Figure 5a: Expected Returns of Different Underlying Assets by Market State
(0 for bull, 1 for bear, and 2 for neutral)

Then, we calculated the Covariance and Correlation Matrix of each asset in the portfolio under different market regimes. The graphs below show the correlation heatmaps of each market state. From the heatmaps, we can see the correlation between asset classes under bear market is mostly higher than that under bull market. In bear markets, assets tend to move more closely together due to widespread caution and a "sell-off" mentality. Investors often rush to exit riskier positions, regardless of individual asset fundamentals, driven by broader economic fears and a desire for liquidity. This collective move to safety can cause various assets to decline simultaneously, leading to higher correlation. Conversely, in bull markets, individual asset performance is often driven more by specific company or sector news, resulting in lower correlations.

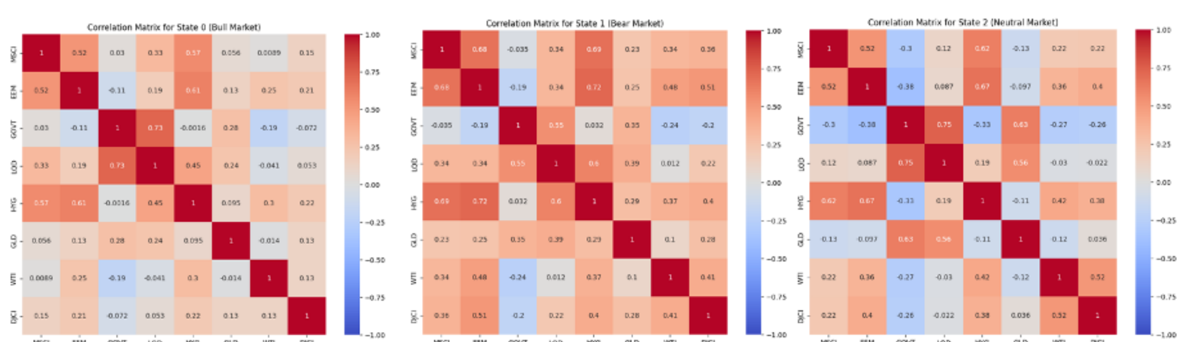

Figure 5b: Correlation Heatmaps under Different Market States

Next, we used the principle of Portfolio Mean-Variance Optimization to calculate portfolio's optimal weight under each state, constructing the dynamic portfolio by adjusting its assets' weights as the market regime switches. The graph below shows optimal weight of each asset in the dynamic portfolio. State 0 stands for Bull market, State 1 stands for Bear market and State 2 stands for Neutral market. From the graph, we can see the regime-adjusted portfolio has a large portion of stock and crude oil under the bull market, and a large portion of gold under bear market. This can be viewed as very reasonable results. In bull markets, driven by economic optimism and growth, stocks and crude oil are favored: stocks benefit from rising corporate profits and consumer spending, while crude oil sees increased demand due to heightened industrial activity. In contrast, during bear markets, characterized by economic uncertainty and pessimism, investors gravitate towards gold. Historically seen as a "safe haven," gold retains its value amidst market declines, offering stability and acting as a hedge against potential inflation or currency devaluation. This portfolio composition reflects traditional strategies, optimizing growth during booms and preserving value during downturns.

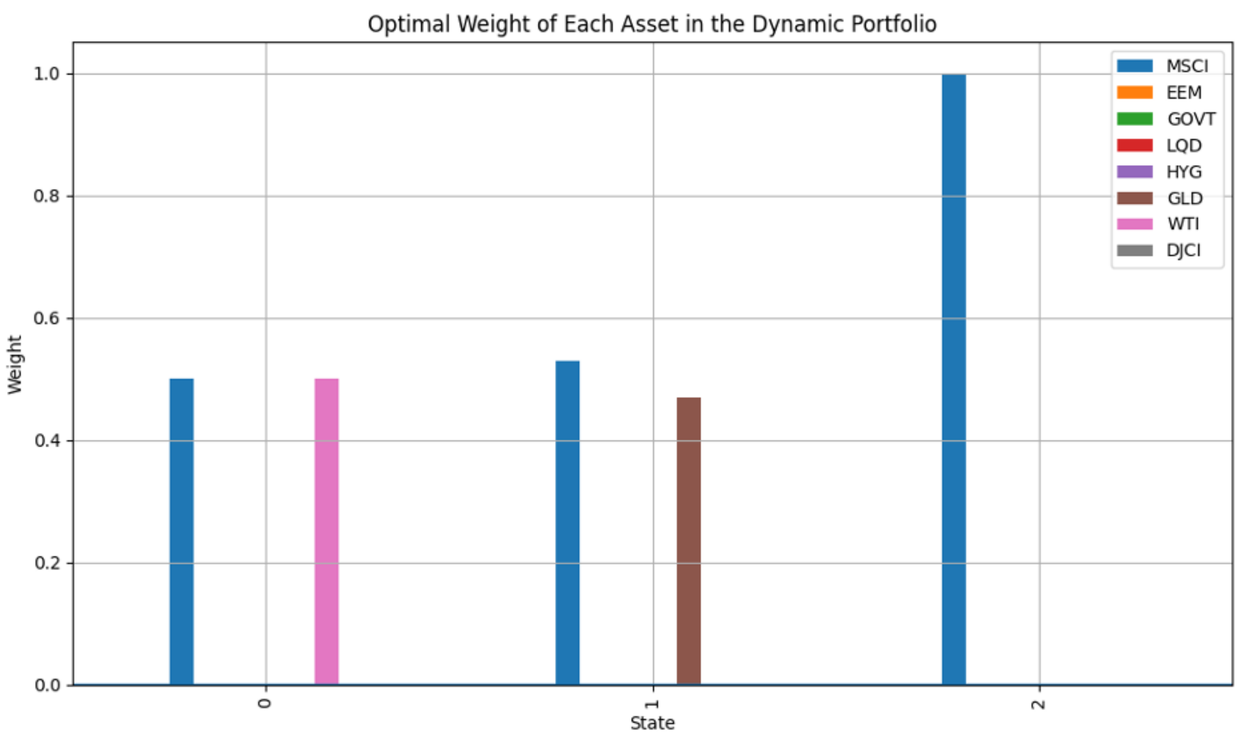

Figure 5c: Expected Returns of Different Underlying Assets by Market State
(0 for bull, 1 for bear, and 2 for neutral)

Finally, we measured the back test performance by comparing the portfolio's cumulative returns between dynamic portfolio (regime-adjusted), static portfolio, and S&P 500 benchmark by plotting the graph of cumulative return and calculating several evaluation metrics.

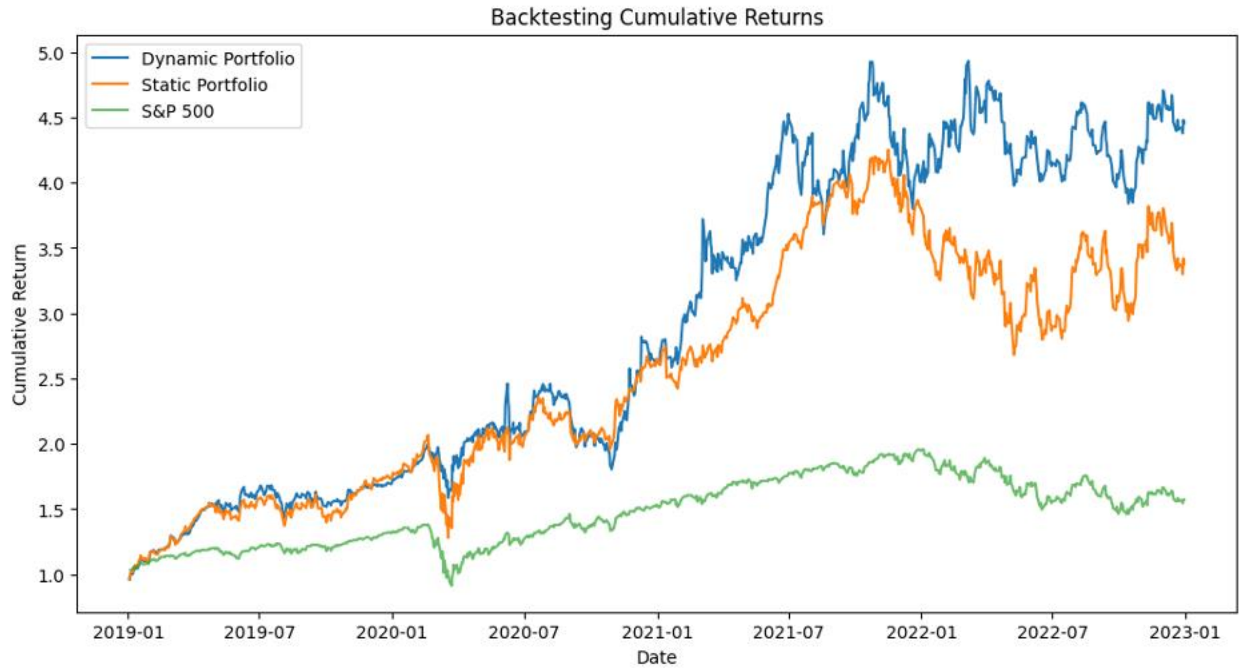

Figure 5d: Result of Backtesting Cumulative Returns

The graph shows our regime-adjusted dynamic portfolio has similar performance with static portfolio before 2021 and out-performs static portfolio after that. And both the dynamic and the static portfolio beats the performance of the S&P 500 benchmark.

Table 5b: Result of Backtesting Return Metrics

Portfolio	**Annual Return**	**Annual Volatility**	**Sharpe Ratio**	**Sortino Ratio**	**Max Drawdown**
Dynamic	0.453	0.340	1.214	1.877	-0.267
Static	0.356	0.368	0.858	1.198	-0.380
S&P 500	0.120	0.229	0.348	0.420	-0.339

The metrics table underscores the superior performance of the dynamic portfolio in most scenarios, with the highest Sharpe Ratio and Sortino Ratio. By recalibrating asset weights based on changing market conditions, investors can achieve the highest annual returns with reduced risk. This approach not only offers the most significant excess return but also minimizes potential maximum drawdowns. This result validates the effectiveness of using HMM for market regime classification.

5.2.2 Optimal Portfolio Construction under Volatility Regimes

For the portfolio construction under volatility regimes, the choice of seven underlying assets is the same as the portfolio construction under bull/bear regimes.

First, we calculated each underlying asset's expected return under different volatility regimes. The graph below shows the result of expected returns of portfolio assets grouped by different market states. State 0 stands for High-volatility market, State 1 stands for Medium-volatility market and State 2 stands for Low-volatility market. From the graph, we can see that most asset classes under high volatility market regime have negative expected returns, which conforms to the objective fact because the high volatility indicates that the market is under stress.

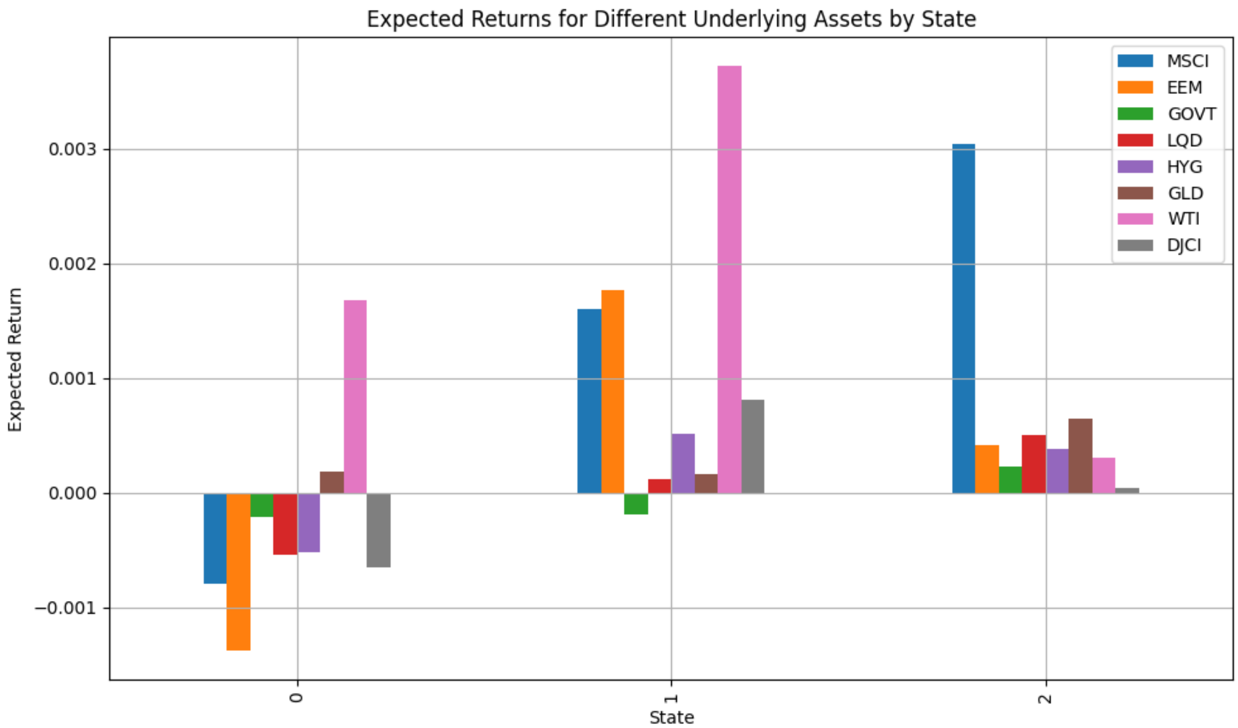

Figure 5e: Expected Returns of Different Underlying Assets by Market State
(0 for high-volatility, 1 for medium-volatility, and 2 for low-volatility)

Then, we calculated the Covariance and Correlation Matrix of each asset in the portfolio under different market regimes. The graphs below show the correlation heatmaps of each market state. From the heatmaps, we can see the correlation between asset class increase largely. This phenomenon conforms to the fact that there are increasing correlations between assets under stressed markets (in this case, the high-volatility market). When the market gets rocky, many investors start to act in similar ways. They might pull their money out of riskier assets and move it to safer places all at the same time. Also, global markets are closely linked, so a big drop in one place can cause ripples everywhere else. Plus, when everyone's watching and reacting to the same big news or events, it makes different investments move in similar ways. This makes the relationship between different investments more alike during these tense times, which is what we mean by "increasing correlations".

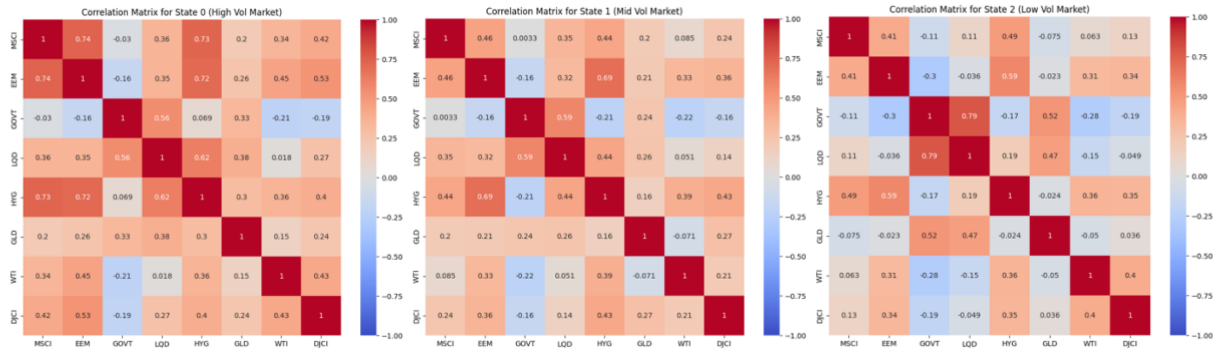

Figure 5f: Correlation Heatmaps under Different Market States

Next, we used the principle of Portfolio Mean-Variance Optimization to calculate portfolio's optimal weight under each state, constructing the dynamic portfolio by adjusting its assets' weights as the market regime switches. The graph below shows optimal weight of each asset in the dynamic portfolio. State 0 stands for High-volatility market, State 1 stands for Medium-volatility market and State 2 stands for Low-volatility market. From the graph, we can see the regime-adjusted portfolio has a large portion of gold under high-volatility market, and some trade-off between emerging market stocks and crude oil under medium-volatility market. This can be viewed as very reasonable results. In volatile markets, investors typically seek assets that can act as safe havens to protect their wealth, and gold has traditionally served this role, being viewed as a stable store of value. Thus, in the regime-adjusted portfolio, a significant allocation to gold during high-volatility periods reflects this risk-averse behavior. Meanwhile, in medium-volatility scenarios, the portfolio showcases a balance between emerging market stocks and crude oil. Emerging markets offer growth potential but come with higher risk, while crude oil can serve as both a hedge and a speculative asset, depending on market dynamics. The trade-off between these two suggests an effort to capture growth opportunities (from emerging markets) while maintaining some level of hedging or diversification with commodities like oil.

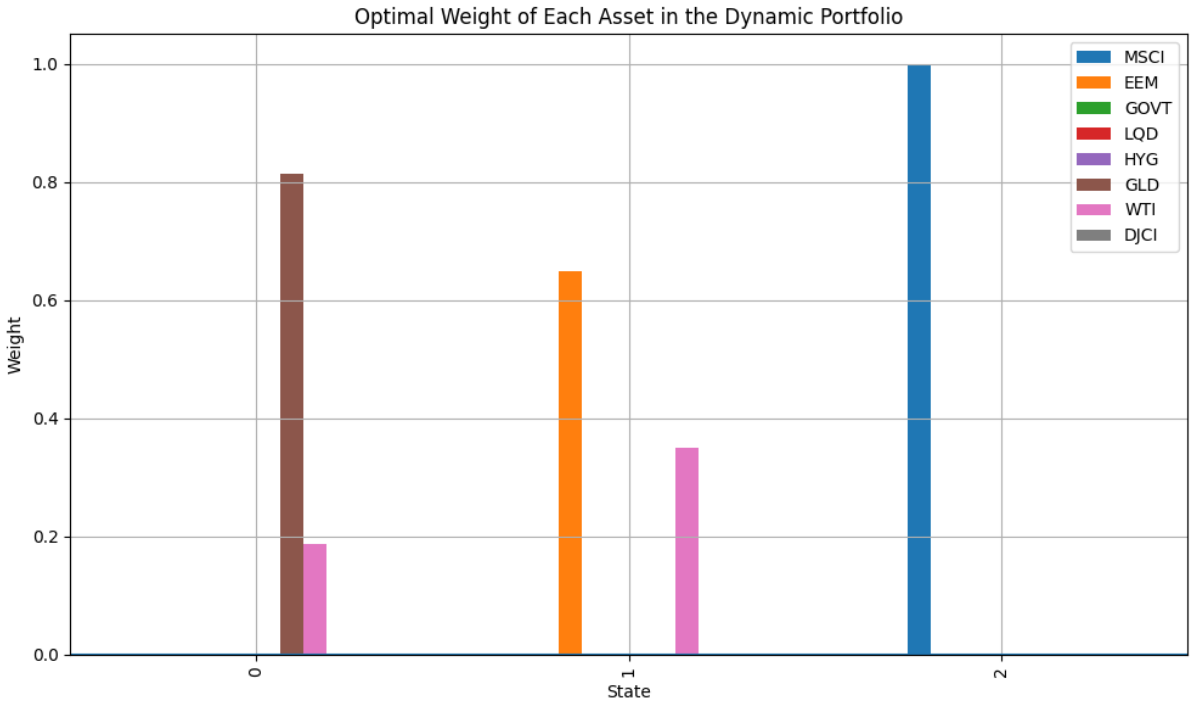

Figure 5g: Optimal Weight of Dynamic Portfolio under Different Market States
(0 for high-volatility, 1 for medium-volatility, and 2 for low-volatility)

Finally, we measured the back test performance by comparing the portfolio's cumulative returns between dynamic portfolio (regime-adjusted), static portfolio, and S&P 500 benchmark by plotting the graph of cumulative return and calculating several evaluation metrics.

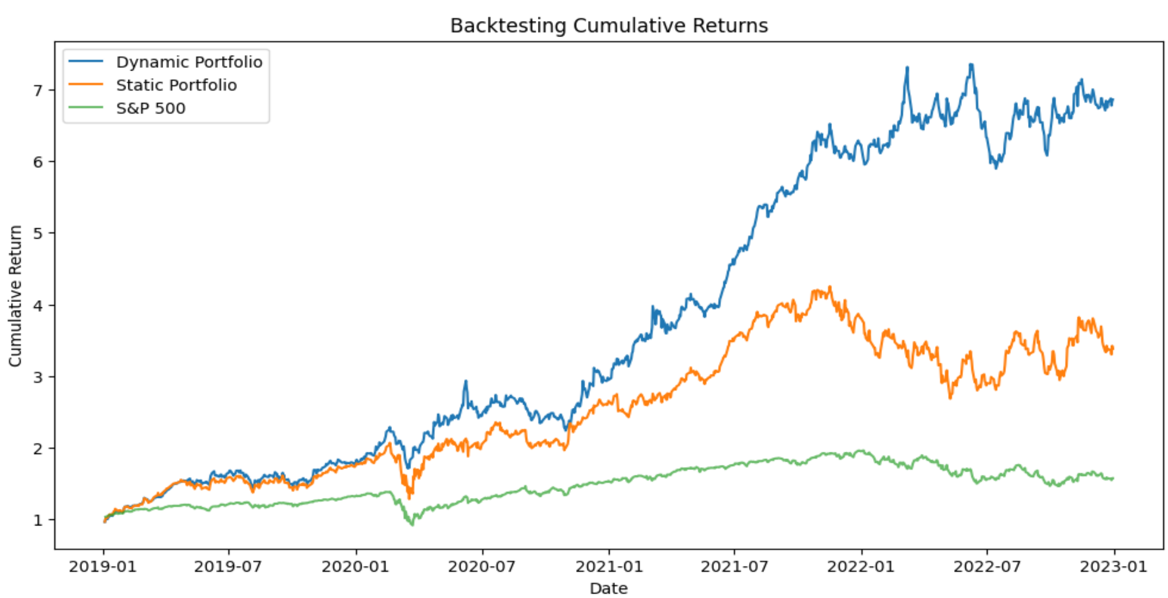

Figure 5h: Result of Backtesting Cumulative Returns

The graph shows our regime-adjusted dynamic portfolio has the highest cumulative returns almost among the whole time period and beats the performance of both static portfolio and S&P 500 benchmark. And the static portfolio beats the benchmark.

Table 5c: Result of Backtesting Return Metrics

Portfolio	**Annual Return**	**Annual Volatility**	**Sharpe Ratio**	**Sortino Ratio**	**Max Drawdown**
Dynamic	0.619	0.283	2.047	3.201	-0.253
Static	0.356	0.368	0.858	1.198	-0.380
S&P 500	0.120	0.229	0.348	0.420	-0.339

The metrics table further illustrates that the result of a dynamic portfolio is ideal in most cases. Adjusting the asset's weight as market regime switches makes investors realize the maximum annual return while bearing the lower risk, also gives investors the highest excess return and the minimum amount of max drawdown.

Market Regime Prediction and Backtest with Hybrid Approach

The goal of the hybrid approach is to leverage the clustering results obtained from the Gaussian Hidden Markov Model (GHMM) as labels to enhance the training process of a machine learning model. The Gaussian Hidden Markov Model has provided a sophisticated and probabilistic way to segment the data into different clusters based on their underlying patterns. Instead of relying solely on traditional labeling methods, which might be prone to biases or limitations, we aim to utilize these identified clusters as a novel form of ground truth for the machine learning task. By treating the cluster assignments as labels, we trained the machine learning model to understand and generalize the intricate relationships between data points within each cluster. This approach has the potential to improve the robustness and accuracy of the resulting machine learning model, as it learns directly from the latent structure revealed by the GHMM. The utilization of unsupervised clustering results as labels in supervised learning is an innovative strategy that holds promise for enhancing the performance of my machine learning model and uncovering deeper insights within the data.

In our study, we employed a diverse range of methodologies to tackle the task at hand. Specifically, we harnessed the power of tree-based algorithms, encompassing both XGBoost and AdaBoost. These ensemble methods excel in capturing intricate relationships within the data and are renowned for their robustness and predictive capabilities.

Furthermore, we delved into the realm of deep learning by incorporating techniques such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks. These advanced neural architectures are tailored to handle sequential and time-series data, making them a natural fit for our complex problem. By leveraging the sequential dependencies present in our dataset, LSTM and GRU models exhibit the potential to unearth nuanced patterns that might elude conventional methods.

In our approach, we selected the decision tree as the foundational classifier for our AdaBoost ensemble. To fine-tune the performance for XGBoost and AdaBoost, we employed the GridSearchCV technique to explore and identify the optimal hyperparameters, ensuring the decision tree's effectiveness in enhancing the ensemble's predictive capacity.

As for the realm of deep learning, we designed a meticulously crafted layer flow for both our LSTM and GRU models. With a rolling window size of 30, we harnessed the temporal dependencies within our dataset to capture intricate patterns. The process of hyperparameter tuning was entrusted to the RandomizedSearchCV method, which enabled a more comprehensive search within a limited time frame.

We only used one layer of LSTM/GRU and intentionally refrained from inserting an additional dense layer prior to the final softmax activation. This strategic decision was informed by our rough experimentation, as we discovered that such an inclusion could lead to significant overfitting issues during our initial model iterations. By eliminating this layer, we aimed to strike a delicate balance between model complexity and generalization.

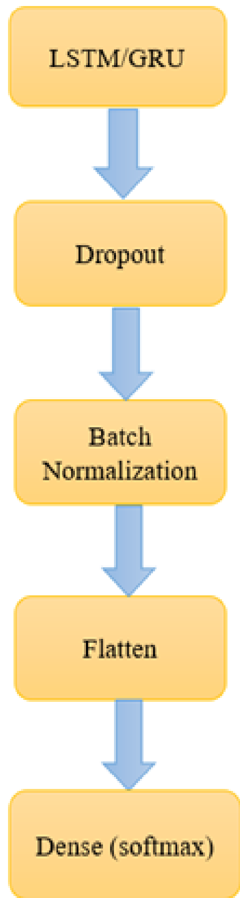

Figure 6a: Deep learning flowchart

6.1 Under 3-state HMM based on Moving Average Return

Under the Bull/Neutral/Bear pattern in regime detection, we obtained the following accuracies on the test set shown in Table 6a.

Table 6a: Performance of different Machine Learning models

Model	Accuracy
XGBoost	0.8890
AdaBoost	0.7044
LSTM	0.5287
GRU	0.6192

Remarkably, XGBoost and AdaBoost surpassed the performance of deep learning models in terms of accuracy. We proceeded to employ the labels predicted by XGBoost on data post-dating January 1, 2019, to conduct a backtest on the underlying assets detailed in Section 4. The following graph is the expected returns for different underlying assets by predicted states.

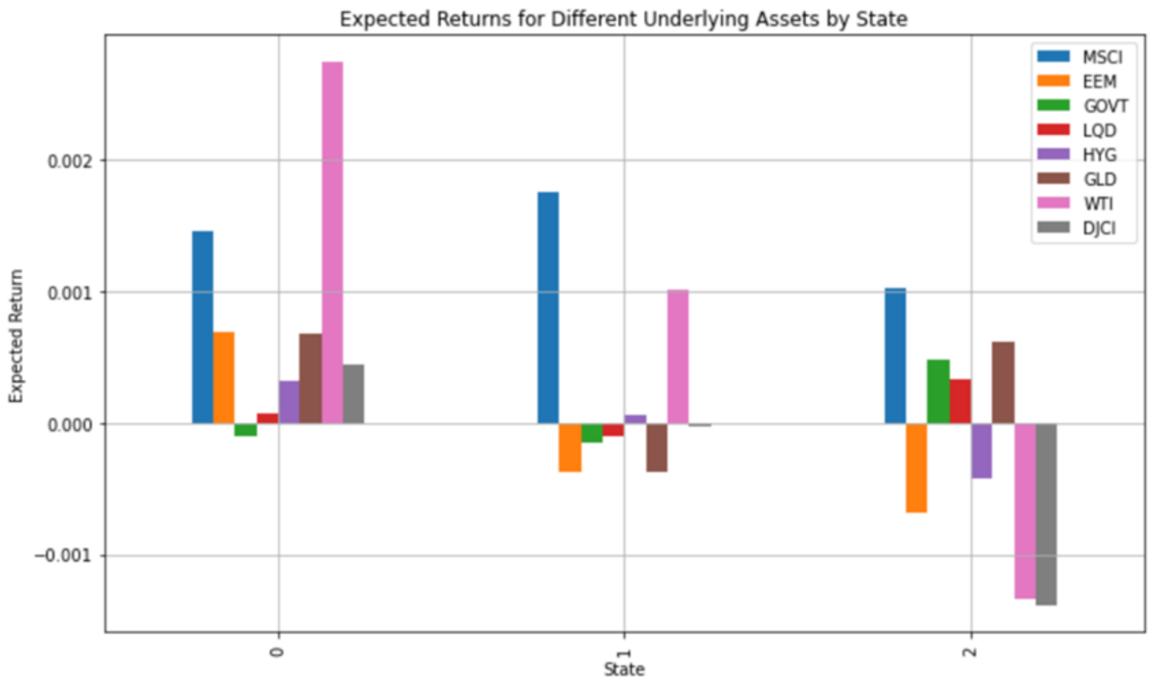

Figure 6b: Expected returns for different underlying assets by predicted states
(0 for bull, 1 for neutral, and 2 for bear)

We still used mean-variance optimization to find the optimal weights for each state.

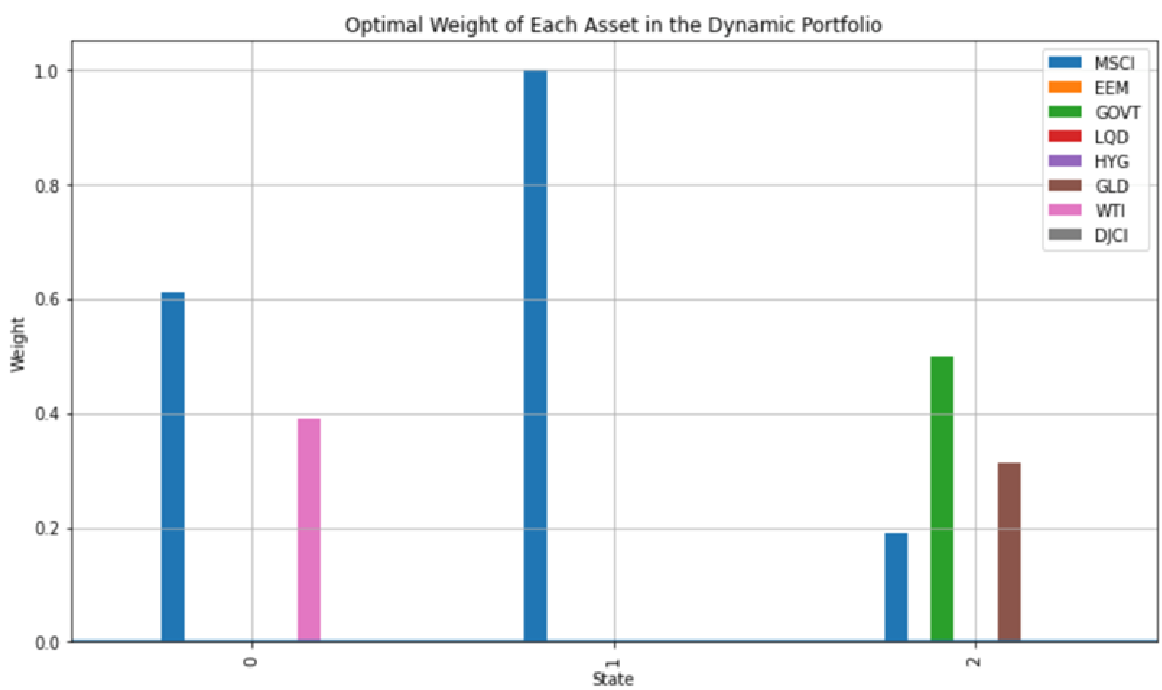

Figure 6c: Optimal weight of each asset in the dynamic portfolio based on XGBoost model
(0 for bull, 1 for neutral, and 2 for bear)

Government bonds are considered relatively low-risk investment instruments, especially when holding bonds issued by stable governments. In bear markets, investors typically seek safe-haven assets, and government bonds are regarded as a secure refuge. The returns of government bonds are often correlated with lower market volatility and reduced risk sentiment. Holding government bonds can provide a steady income stream while protecting investment portfolios from the intense fluctuations of the stock market.

Gold is regarded as a safe-haven asset and often performs well in unstable economic and market conditions. During bear markets, due to economic downturns and market instability, investors may seek to hedge against stock market risks, with gold often being favored in such scenarios. The price of gold is typically linked to expectations of economic recession and inflation, making it an asset that can retain value during periods of market volatility.

The back test performance is given in the cumulative return graph in Figure 6d and the evaluation metrics in Table 6b.

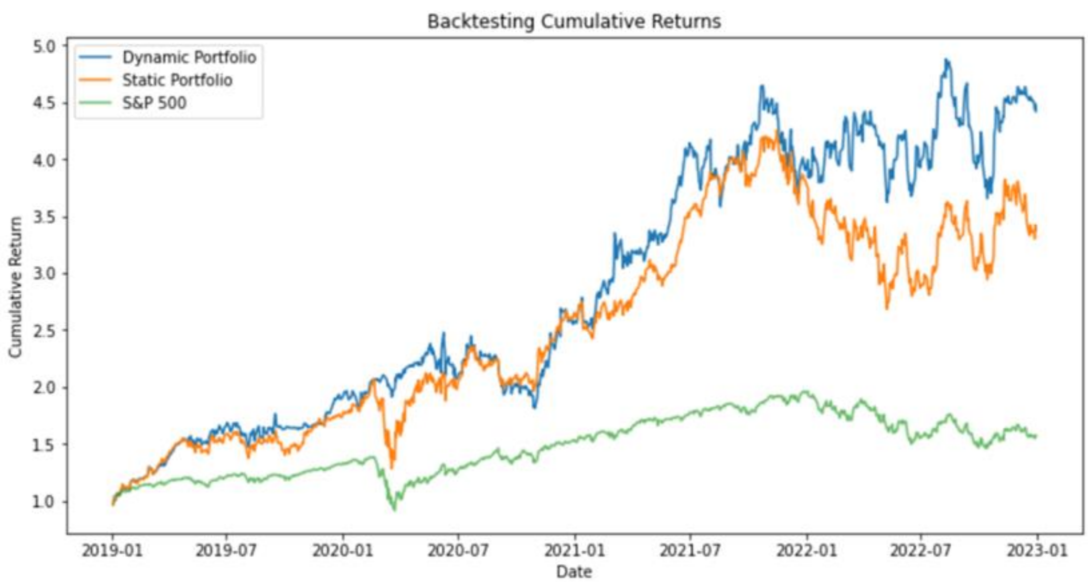

Figure 6d: Backtesting cumulative returns

Table 6b: Result of Backtesting Return Metrics

Portfolio	Annual Return	Annual Volatility	Sharpe Ratio	Sortino Ratio	Max Drawdown
Dynamic	0.450	0.316	1.297	2.006	-0.269
Static	0.356	0.368	0.858	1.198	-0.380
S&P 500	0.120	0.229	0.348	0.420	-0.339

6.2 Under 3-state HMM based on Moving Average Volatility

Next, we tried the same hybrid approach on clustering results based on market volatility. Table 6c displayed the accuracy statistics of each model.

Table 6c: Performance of different Machine Learning models

Model	Accuracy
XGBoost	0.8890
AdaBoost	0.7044
LSTM	0.5287
GRU	0.6192

Reversely, LSTM and GRU outperformed boosting methods. LSTM and GRU are designed to capture sequential and temporal patterns in data, which are commonly found in time series. The empirical result reflected that they can effectively model long-range dependencies and capture complex relationships across different time steps, allowing them to capture volatility patterns that may span multiple periods. LSTM and GRU also have memory mechanisms that enable them to retain and update information over long sequences. This is essential for capturing and remembering volatility patterns that may have occurred at various time scales in the past.

We chose GRU to predict the labels on the test set. Figure 6e showed the expected returns for different underlying assets by predicted states.

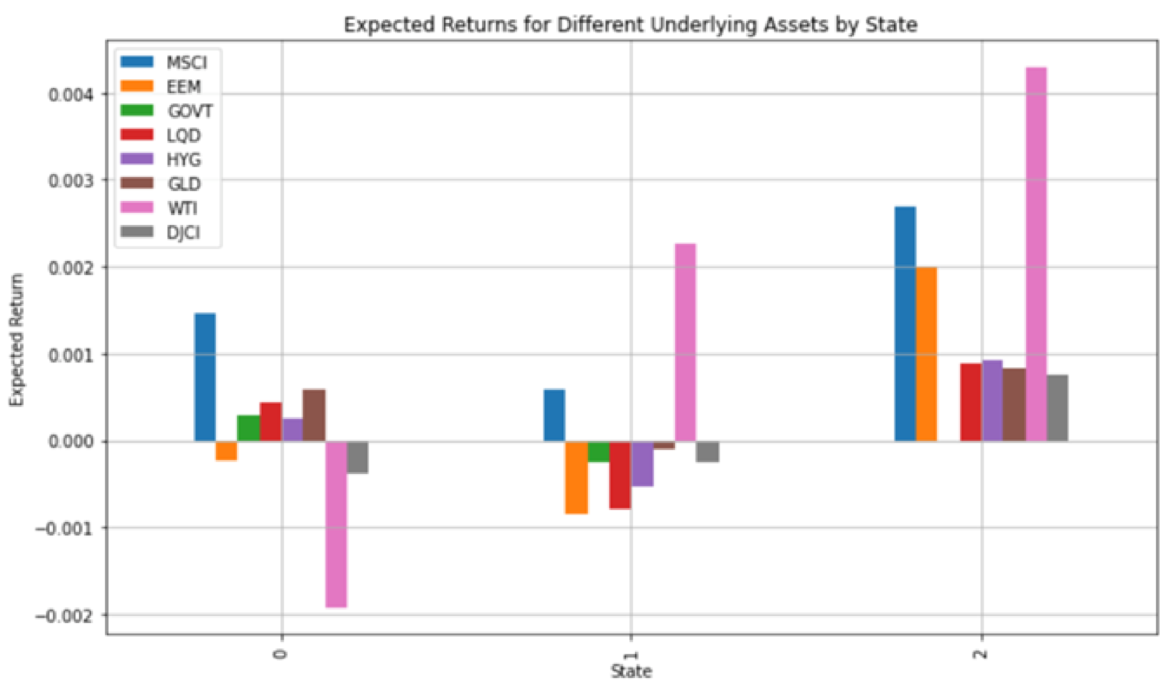

Figure 6e: Expected returns for different underlying assets by predicted states.
(0 for low-vol, 1 for medium-vol, and 2 for high-vol)

The optimal weights indicated by mean-variance optimization are plotted in Figure 6f.

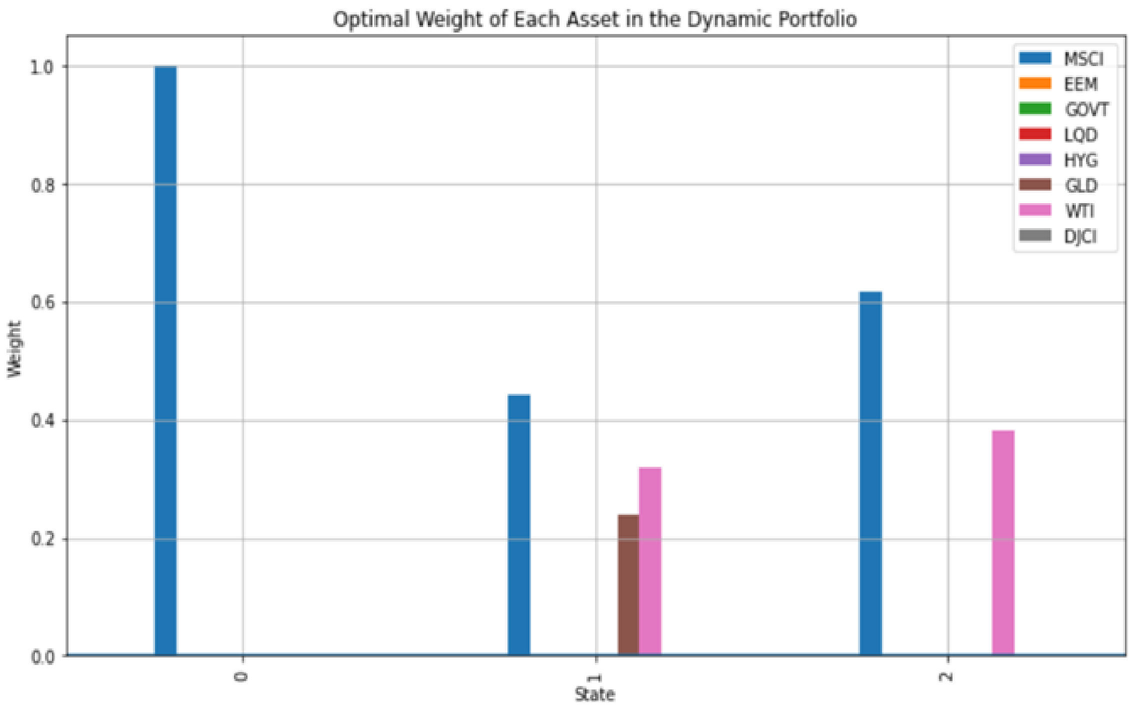

Figure 6f: Optimal weight of each asset in the dynamic portfolio based on GRU model.
(0 for low-vol, 1 for medium-vol, and 2 for high-vol)

For low-volatility regimes, developed stock markets are generally considered relatively stable compared to other asset classes like commodities. Therefore, allocating all investments to the developed stock market reflects a conservative approach. The expectation is that the stock market, while potentially offering moderate returns, will also have lower volatility and a reduced risk of significant losses.

In a medium-volatility market, the investment strategy introduces some diversification by allocating weights to gold and crude oil. Gold is considered a safe-haven asset that can act as a hedge against market uncertainties, while WTI crude oil is influenced by supply-demand dynamics and can benefit from economic growth. By including both gold and crude oil, the portfolio aims to balance risk and return, leveraging the potential benefits of different asset classes in a moderately volatile environment.

When market volatility is high, the focus shifts towards more aggressive strategies to potentially capitalize on larger market swings. In this scenario, the allocation concentrates on two relatively high-risk assets: the stock market and crude oil. This allocation strategy seeks to capture potential returns from market volatility while acknowledging the associated risks.

The back test performance is given in the cumulative return graph in Figure 6g and the evaluation metrics in Table 6d.

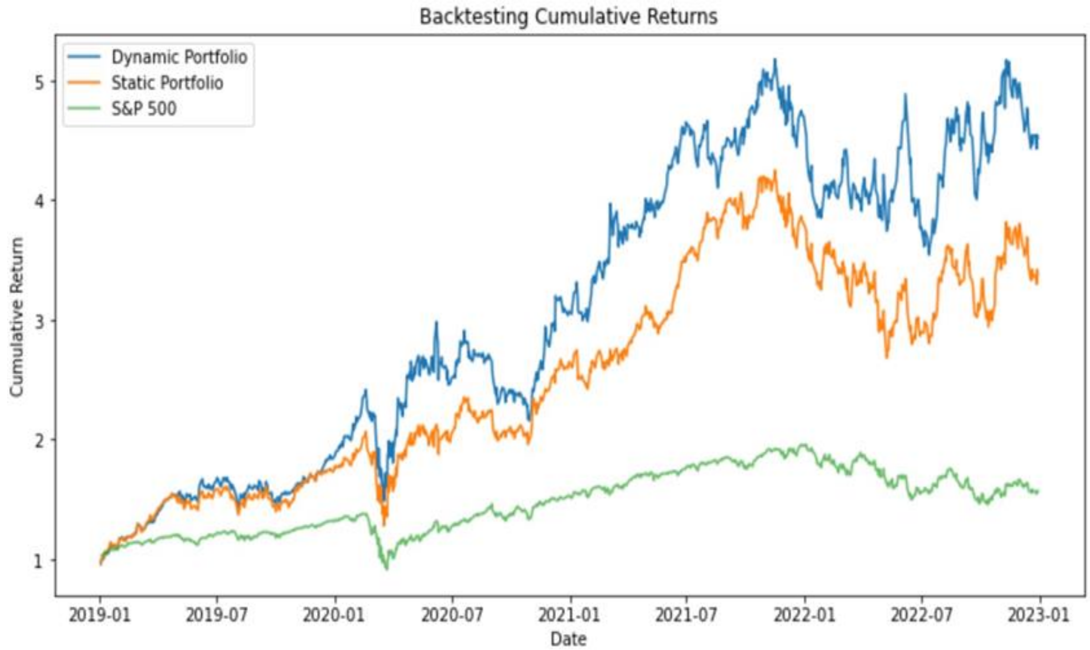

Figure 6g: Backtesting cumulative returns

Table 6c: Result of Backtesting Return Metrics

Portfolio	**Annual Return**	**Annual Volatility**	**Sharpe Ratio**	**Sortino Ratio**	**Max Drawdown**
Dynamic	0.459	0.384	1.089	1.590	-0.385
Static	0.356	0.368	0.858	1.198	-0.380
S&P 500	0.120	0.229	0.348	0.420	-0.339

The dynamic portfolio consistently outperforms both the static portfolio and the S&P500 in terms of returns. However, it comes with the drawback of experiencing a higher maximum drawdown.

Conclusion

In this study, we employed both K-means and HMM techniques to identify market regimes and utilized four machine learning models to predict these regimes. Both clustering methods distinctly highlighted market states, offering insightful analysis of underlying market characteristics. The evidence leans towards a preference for a three-state classification (bull, bear, neutral) using both methods, as this approach more accurately mirrors real-world market conditions and aligns more closely with actual regime shifts in terms of expected returns and volatility. Among our predictive models, XGBoost stands out for its superior accuracy when considering moving average returns, while GRU and LSTM excel when based on Moving Average Volatility. With the regimes identified, we devised portfolio strategies grounded in these states (dynamic) and conducted back testing using real-time data. The results showed that a dynamic portfolio, influenced by market states, surpassed the performance of a static portfolio strategy and the benchmark S&P 500. This superior performance underscores the efficacy of our market regime detection techniques, resonating with previous research that highlights the capability of HMM and machine learning tools in capturing market dynamics.

Future Study

Expanding on our foundational research, several avenues beckon for deeper exploration. Firstly, in terms of clustering methodologies, while K-means and HMM have laid down a robust framework, other techniques like DBSCAN or Gaussian mixture models could be employed to validate or even enhance our regime classifications.

Additionally, while our machine learning models like XGBoost, GRU, and LSTM showed promise, there's potential in probing more intricate architectures, especially Transformer-based and hybrid models, which might offer nuanced insights into market dynamics. Moreover, our feature set, which played a pivotal role in regime predictions, can be augmented with alternative data sources, such as sentiment analysis from news or social media, and other macroeconomic indicators, providing a more holistic view of the market's undercurrents.

Lastly, refining our back testing framework to account for real-world constraints, like transaction costs and taxes, could make our strategies more pragmatic and actionable.

References

[1] Akioyamen, P., Tang, Y. Z., & Hussien, H. (2021). A Hybrid Learning Approach to Detecting. Regime Switches in Financial Markets.

[2] Ang, A., & Timmermann, A. (2011). Regime Changes and Financial Markets.

[3] Benhamou, E., Ohana, J.-J., Saltiel, D., Guez, B. (2021). Explainable AI (XAI) models applied to planning in Financial Markets.

[4] Botte, A., & Bao, D. (2021). A Machine Learning Approach to Regime Modeling.

[5] Bulla, J., & Bulla, I. (2006). Stylized facts of financial time series and hidden semi-Markov models.

[6] Hamilton, J. D. (1989). A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle.

[7] Horvath, B., Issa, Z., & Muguruza, A. (2021). Clustering Market Regimes using the Wasserstein Distance.

[8] Liu, Z., & Wang, S. (2017). Decoding Chinese stock market returns: Three-state hidden semi-Markov model.

[9] Nystrup, P., Hansen, B. W., Larsen, H. O., Madsen, H., & Lindström, E. (2017).

[10] Dynamic allocation or diversification: a regime-based approach to multiple assets. The Journal of Portfolio Management, 44(2), 62–73.

[11] Nystrup, P., Madsen, H., & Lindström, E. (2015). Stylised facts of financial time series and hidden Markov models in continuous time. Quantitative Finance.

[12] Zhong, X. and D. Enke (2017). "A comprehensive cluster and classification mining procedure for daily stock market return forecasting." In: Neurocomputing 267, pp. 152–168.