Market Regime Capstone Project Spring 2022

ChienYueh(Oscar) Shih, Hanlu(Hannah) Xia

May 10, 2022

Executive Summary

Background: The term market regime refers to the collective behavior of traders, which is usually a cluster of persistent market conditions. Market regimes are vital to researchers due to their relation to the global economy and asset valuation. Forecasting market trends can be extremely useful for portfolio managers, especially when facing financial downturns or catastrophic black swan events. While statistical modeling is still the primary method used in building market-regime switching models, to make regime-switching prediction more interpretable, we considered an approach based on 9 machine learning methods.

Problem Statement:

1. What's the best time interval to predict Market Regime? Does the regime changing on a weekly or monthly basis?
2. Can feature selection help improve the result?
3. Which model would be the best model to use for regime detection problem?

Method: The principal component analysis is a proven useful technique in dimensional reduction and feature selection. The selection is made based on a set of 40 technical, fundamental and macroeconomic features. K-Means is used as the baseline model due to its simplicity. Other machine learning techniques used include classification,tree-based model, and hidden Markov model(HMM). K-means clustering and HMM are the only unsupervised learning algorithms used. The Hidden Markov model is a popular technique for detecting the underlying sequence of persistent states, which is a great fit for the market regime identification challenge. Other supervised models, such as Support-vector machines (SVM), gradient boosted decision trees (GBDT), and Random forest (RF), are robust supervised learning algorithms used in this research.

Approach: 3 datasets are used in this research: Monthly, Weekly, and Monthly with selected features. We calculated a target regime classification based on historical market returns. The datasets are all made stationary. A Grid Search technique is then applied to find out the best parameters of each model. The ROC and cross-validation scores are calculated for each supervised model. Finally, A backtesting strategy is applied to determine how well each model performed at real market regime classification.

Results:

1. Market Regime is more predictable on a monthly basis. A weekly time frame is more suitable for transient market change.
2. Feature selection does help increase the efficiency of modeling without the sacrifice of accuracy.
3. Hidden Markov Model and Linear Discriminant Analysis have the best performance in all 3 trials. Tree-based model, such as Random Forest and GBDT, also performed well at regime classification.

1. Data Description

1.1 Data Description

In this study, the data set we used can be grouped into 5 categories:

1. The open-source Federal Reserve Economic Data (FRED) ranges from 1990 to 2022 for both international and national datasets. These time series include information such as GDP growth, inflation, CPI, short and long-term interest rate, and unemployment rate. Only the quarterly data is available for the FRED.
2. The equity index S&P500 ranges from 1990 to 2022 for both weekly and monthly datasets.
3. The commodities index S&P Goldman Sachs Commodity Index ranges from 1970 to 2022 for both weekly and monthly datasets.
4. US Bond Yields for 1 month, 3 months, 5 years, 10 years and 30 years ranging from 1990 to 2022.
5. Chicago Board Options Exchange's CBOE Volatility Index (VIX), a popular measure of the stock market's expectation of volatility based on S&P 500 index options, ranges from 1990 to 2022.

Technical Indicators are also calculated for the above datasets. The major technical indicators we used are the return of each column, historical volatilities over the last 10 data points, and the first difference if necessary.

1.2 Data Preparation

Any missing values in between reporting dates were replaced using the latest observed percentage change in the data. The training data consists of all observations up to the last reported date in 2014. Testing data contains observations from the first reported date in 2015 and onward.

The raw datasets we have are all recorded in various date formats. For example, datasets all have different format and two types of date formats can appear in one file. All the datasets are converted into one date formats. Moreover, some datasets have huge variation. For example, S&P 500 historical data varies from 63.5 to 4766. All the data points are converted to natural log for modeling efficiency.

Potential outliers are cleaned using the approach of Brownlees and Gallo (2006). We will decide whether to retain or remove a price based on the following formula:

$$|p_{t_i} - \overline{p_{t_i}}(k)| < 2S_{t_i}(k) + \gamma \tag{1}$$

where $\overline{p_{t_i}}(k)$ is a k-day rolling mean and $S_{t_i}(k)$ refer to the standard deviation. The parameter $\gamma = 0.005$ p. The one greater than 2 standard deviations are potential outliers. We have to make sure whether they are outliers or they are representing bubbles or crashes of the market before removing them. The model can be fit with and without outliers to see the difference.

2. Literature Review

Following the Asian financial crisis, the dot-com bubble in the 1900s, and the subsequent crash, more and more research focus on detecting and forecasting financial crises. Generally speaking, past research approaches can be classified into two types: statistical models and machine learning approaches.

2.1 Statistical Approach

Messer (2021) provided methods for the detection of change points using a univariate sequence of independent random variables which are tested by statistical testing with a bivariate moving sum approach, which jointly quantifies the change in both expectation and variance[1]. Schatz (2020) introduced mathematical models of the financial market experiencing bubbles and crashes. The study developed a general stochastic model framework to describe the markets with increasingly positive returns (the build-up of a bubble) with a sudden reversal and an accumulation of large negative returns (the crash)[2]. Nystrup et.al (2016) proposed a non-parametric approach to detect the change points in VIX and S&P 500 without setting a fixed number of regimes and without assuming the distribution of the data[3]. For the extreme event measuring, Lattazi and Leonelli (2019) employed the extreme value mixture models with the Bayesian paradigm using the Markov Chain Monte Carlo (MCMC) machinery for generalized Pareto distribution to model the tail events[4]. Sornette et al. (2017) discuss the nature of volatility before and after the financial crisis[5].

2.2 Machine Learning Approach

Liu et al. (2021) tested various machine learning approaches, such as logistic regression, K-nearest neighbors, support vector machine, etc., to capture the complexity of financial crises given they have more flexibility in investigating the relationships between predictors and identifying nonlinear relationships[6]. The results showed that random forest, gradient boosting decision tree, and ensemble models performed the best in terms of building early warning models. Mizuno (2020) showed that the variation in market capitalization normalized by financial fundamentals that are estimated by Lasso regression could be beneficial to detect the dot-com bubble[7]. Akioyamen et al. (2021) implemented principal component analysis for dimensionality reduction and the k-means clustering was used to prepare for further supervised classification algorithms, such as Linear Discriminant Analysis and Adaptive Boosting Algorithm[8]. Benhamou et al. (2021) presented a gradient-boosting decision trees (GBDT) approach to predict large price drops in equity indexes from technical, fundamental, and macroeconomic features[9][10]. Chen et al. (2021) applied two classes of unsupervised learning methods: clustering and manifold learning for the reduction of dimensionality [11][12]. The volatility-based clustering succeeds in identifying periods of extreme market distress, such as the financial crisis in 2008 and the Covid-19 pandemic. McIndoe (2020) discussed the connection between market regimes and distributions of path signature and provide a metric space structure on the latter which allows for a clustering to be formulated. In other words, market conditions can be seen as the distributions on path signatures, and the market regimes can be thought of as the cluster of similar distributions[13]. Pharasi et al. (2020) presented clustering analysis over statistical properties of correlation matrices and classified different clusters, named as market states, based on the similarity of correlation structures[14].

2.3 Hidden Markov Model

The ability to estimate the transition probabilities between market regimes from observable factors such as macroeconomic data has brought lots of attention. Hidden Markov Model (HMM) has been highly used to build regime detection and forecast model with time-series data. Nystrup et al. (2015) construct 2-state HMM with time-varying behaviors of underlying process and calculated the transition probability and the model was tested with regime-based asset allocation due to it focused on the long-term investment horizon[15]. Nystrup et al. (2018) continued their study in 2015 and expand it to multiple asset classes, and a new, more intuitive way of inferring the hidden market states[16]. It is shown to be profitable compared to a diversified benchmark portfolio in a multi-asset universe. On the other hand, Oprisor and Kwon (2021) proposed a multi-period portfolio optimization framework to incorporate portfolio managers' views with the Black-Litterman model and use a regime-switching model to determine each view's confidence[17].

Going beyond typical HMM, Zheng et al. (2019) proposed a novel method of Markov regime-switching (MRS) model estimations by spectral clustering hidden Markov model (SV-HMM)[18]. SC-HMM predicts latent states and yields conditional distribution statistics without knowledge of the types of conditional distribution. Dal Pra (2021) introduced a regime-switching process driven by a standard first-order Markov chain. Markov switching regressions that imply time-varying slope coefficients associated with the predictors, and

Markov Switching vector autoregressions that also incorporate the dynamics of the predictor variables[19]. The regression model can also be used to determine the market-regime switching. It can be determined by some observable transition variables. Ciciretti (2021) detected the regime switches with the Vector Smooth Transition autoregressive model with lagged exogenous variables. The transition between market regimes is mirrored in correlation matrices, whose time-varying coefficients usually jump higher in the highly volatile regimes[20].

To the best of our knowledge, there are people trying to implement hybrid approach to regime detection which uses both unsupervised learning and supervised learning. A related work is by Akioyamen et al.(2021) where they implement supervised classification algorithms on clustering results. However, we believe that it is not reliable to use clustering results as our classification target. Therefore, we employ supervised classification algorithms on regime based on market rolling average return where regime changes only if it stays the same in two or more consecutive months. We implement statistical methods and also machine learning approaches to see which one performs the best on our strategy[8].

3. Methodology

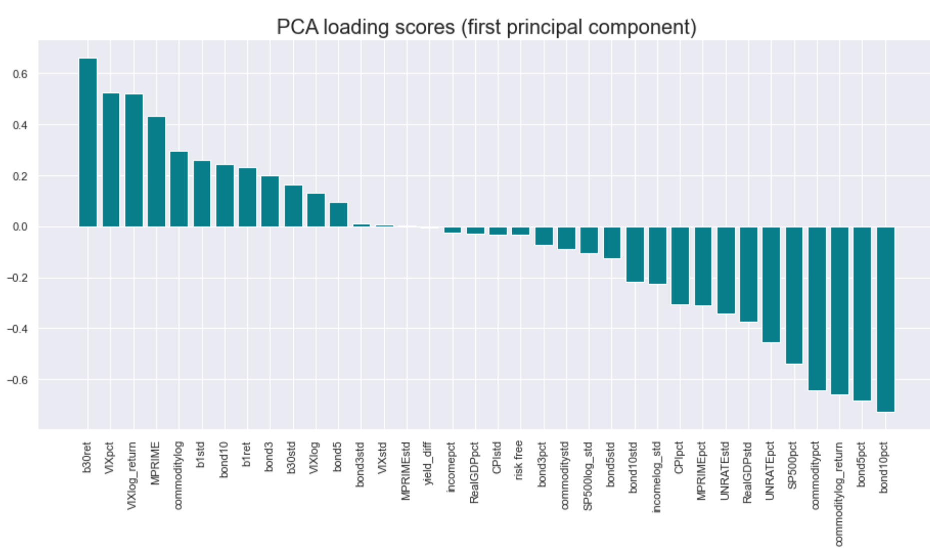
3.1 Principal Component Analysis

Principal Component Analysis, or PCA, is a dimension-reduction method that is often used to reduce the dimension of large data sets. PCA will transform a large set of variables into a smaller one but still keep most of the information from the original datasets. In this research, a smaller number of variables increases the simplicity of our models and reduces noise in datasets, thus can help improve both efficiency and accuracy. By ranking their eigenvalues, the desired after PCA dataset should capture the maximal variance from the original one. The first principal component represents the largest possible variance in the data set.

The resulting principal components are uncorrelated time-series observations. Let A be the data matrix of size the total number of time observations considered (U) times the total number of features in this work (T), so $A = U \times T$. Every data point r_{ij} of matrix A represents the percentage change of statistic j at the time i. This research will select components with at least a collectively of 90% of the variance present in the original data.

The PCA loading score of the first principal component is converted back to each feature. And Figure 1 shows the loading result of each feature. Features with importance above and below -0.2 will be selected into the feature importance detests.

Figure 1: PCA Loading Score of First Component



3.2 Major Machine learning Models

3.2.1 K-Means Clustering

Clustering is one of the unsupervised learning where it takes data and divides them into groups or clusters based on their similar properties. K-Means clustering is the algorithm we used in this capstone project where K is the number of clusters we want to divide our data into. This method randomly assigns K cluster centroids, then assigns each data point to the closest centroid. This model is used as an baseline model in this research due to its simplicity.

3.2.2 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a classifier with a linear decision boundary, generated by fitting class conditional densities to the data and using Bayes' rule. Despite its simplicity, LDA often produces robust, decent, and interpretable classification results. When tackling classification problems, LDA is often the benchmark before other more sophisticated ones are employed. A discriminant rule tries to divide the data into K disjoint regions (3 in our case) that represent all the classes. The allocation rule to separate data to j space is the maximum likelihood rule.

$$j = argmax_i f_i(x) \tag{2}$$

where f is the class-specific distribution

3.2.3 Adaptive Boosting Classifier

It is called adaptive boosting as the weights are re-assigned to each data point, with the higher weight assigned to incorrectly classified data point. In other words, the algorithm begins by fitting a classifier to the original dataset and then fits additional classifiers to the same dataset but with more weights on incorrectly classified data points such that the subsequent classifier focuses more on difficult data points.

3.2.4 Gradient Boosting Classifier

Just like the Adaptive boosting classifier, boosting in a classification problem refers to combining multiple ”weak” classifiers into a ”strong” classifier. In Gradient Boosting, each classifier tries to improve on its predecessor by reducing the errors. But the fascinating idea behind Gradient Boosting is that instead of fitting a predictor on the original dataset at each iteration, it fits a new classifier to the residual errors made by the previous predictor.

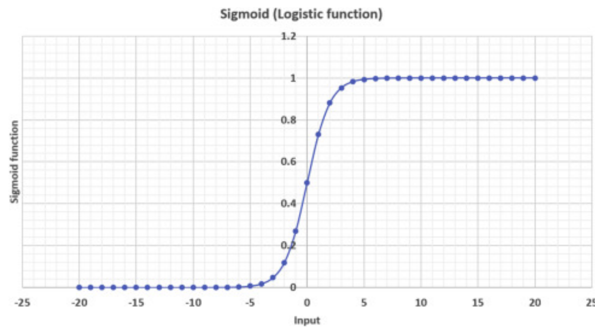
3.2.5 Logistic Regression

Logistic Regression is a linear regression but used for classification problem. It uses a logistic function defined as

$$\frac{1}{1+e^{-x}} \tag{3}$$

Equation (3) maps any number into a value between 0 and 1 and we can view the value as a probability to be in a specific class. If the curve goes to positive infinity, y predicted will become 1, whereas 0 if the curve goes to negative infinity. It can generalize to multinomial logistic regression which has a similar idea to binary logistic regression. That is, it is a model that is used to predict the probabilities of the different possible outcomes and assign the class with the highest probability.

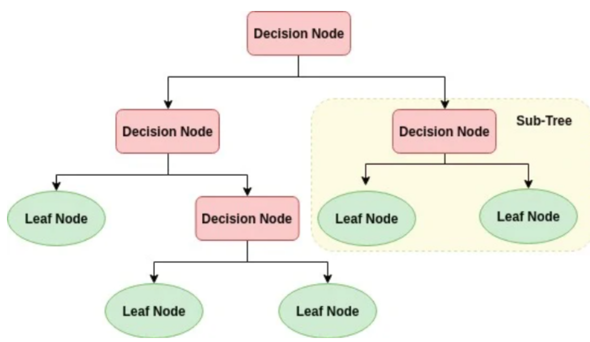
Figure 2: Logistic Regression Model



3.2.6 Decision Tree Classifier

The decision tree classifier creates the classification model by building a decision tree. Each node in the tree specifies a test on an attribute, and each branch descending from that node corresponds to one of the possible values for that attribute. Each leaf represents class labels, i.e. outcome, associated with the instance. Instances in the training set are classified by navigating them from the root of the tree down to a leaf, according to the outcome of the tests along the path.

Figure 3: Decision Tree Model



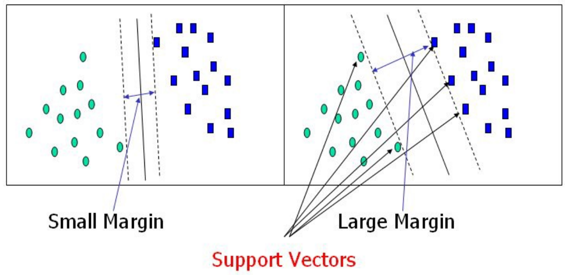
3.2.7 Random Forest Classifier

Random Forest Classifier is an ensemble method that collects multiple decision tree classifiers to increase the accuracy of an individual decision tree. Every individual decision tree is generated using a random selection of attributes at each node to determine the split. During classification, each tree votes, and the most popular class is returned.

3.2.8 Support Vector Classifier

Support vector machine algorithm tries to find a hyperplane in N-dimensional space (N — the number of features) that distinctly classifies the data points. Hyperplanes are decision boundaries that help classify the data points. Data points that fall on either side of the hyperplane can be attributed to different classes. Support vectors are data points that are closest to the hyperplane and influence the position and orientation of the hyperplane. The best hyperplane is the one that maximizes the margin of the classifier.

Figure 4: Support Vectors



3.2.9 Hidden Markov Model

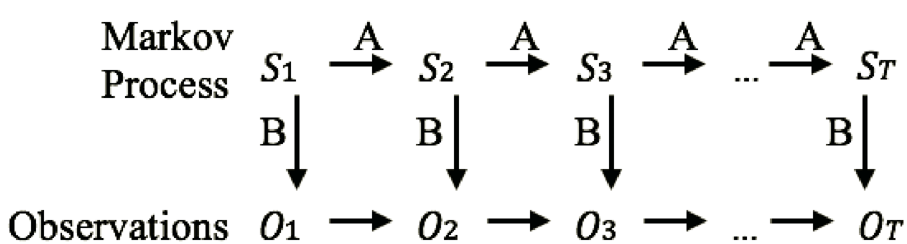
The theory of the Hidden Markov Model(HMM) was first introduced in the 1970s and it was then widely applied in areas such as economy, computational biology, and physical sciences. An HMM is a statistical model which relates a sequence of observations (public information) to a sequence of hidden states (market regime). The observation sequence is visible to the public, but the hidden state is implied and has to be calculated. Based on this nature, usually, no observable seasonal patterns can be found in the market cycle by using the HMM. In this market regime research, the goal of the HMM is to identify the underlying persistent state of the market regime by analyzing the public price sequence such as the S&P 500.

The HMM is built based on the Markov assumption. The Markov assumption describes the probability of one event only depending on its previous event in a sequence of events. The HMM assumes a first-order Markov chain, which means the probability of a particular state depends only on the previous state. The Markov assumption in HMM can be formulated as:

$$P(q_i = a|q_1 ... q_{i-1}) = P(q_i = a|q_{i-1}) \tag{4}$$

The general structure of an HMM is described in Figure 1 below. The model contains three parameters: the transition probability matrix A, a sequence of observation likelihoods B, and the initial probability distribution π_i. Each element a_{ij} in A represents the probability of moving from state i to state j. $B = b_i(O_t)$ and represents the probability of an observation O_t being generated from a state π_i. i is the probability that the Markov chain will start in state i.

Figure 5: Structure of HMM



In Figure 5, S_i is the hidden market regime and O_i is the market observation. A and B are the parameters of an HMM. The input to the machine learning algorithm would be an unlabelled sequence of observations O and the number of potential hidden states S. The model will find a sequence of S that maximally fits the observed data.

The algorithm will start with an initial estimation of the parameters of HMM, A and B. After giving each input data, the model computes the expected value of A and B. Those computed probability matrices are then used to re-estimate the new HMM parameters A and B. The algorithm will repeatedly carry out this process until it achieves convergence.

4. Exploratory Analysis

4.1 Regime Define

To perform supervised machine learning, we used the average return from several indexes (SP500, NASDAQ, Russell 2000, Wilshire 5000) to categorize market regimes into 3 states. The regime is smoothed with a quarterly rolling average and regime continuity.

Figure 6 shows the calculated regime for this project, where regime 0 can be described as the bad market (bear market), regime 1 is the normal growth state, and regime 2 is the high return market (super bull market). For example, the blue dots around 2002 and 2008 represents the early 2000s recession and the 2008 Great Recession. The green dots in 2020 represents the big market boom after COVID spread.

Figure 6: Calculated Monthly Regime as Reference

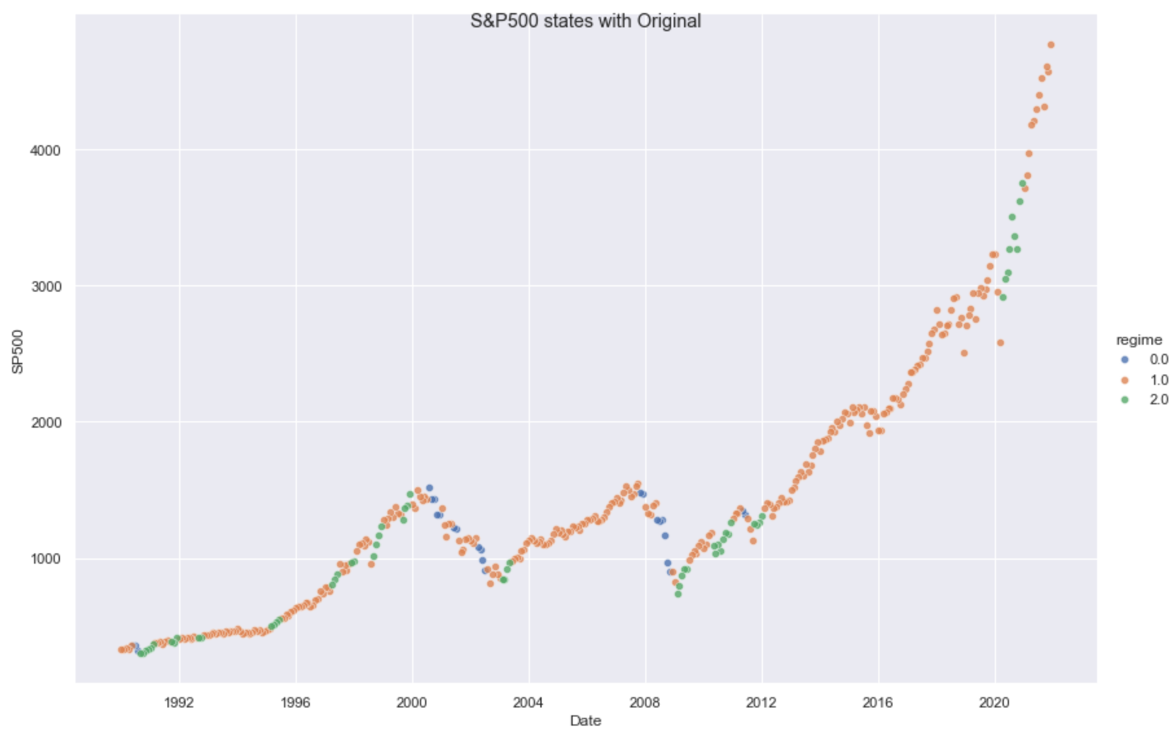


Table 1 will be used as a market regime reference in this project. It populated some major market crashes that happened recently. However, only data from 1990 to 2015 will be used as the training set. Major market crashes and expansion after 2015, such as the COVID crash and expansion, will be used as the testing case for each model to predict independently.

Table 1: Major Market Crash after 1990

Name	Time	Cause
Downturn of 2002	Oct 2002	Downturn in stock prices across countries
Financial crisis of 2007–2008	2007-2008	Failures of large financial institutions in US
2015–2016 stock market selloff	Aug 2015	The Dow Jones Fall
2020 stock market crash	Feb 2020	COVID-19 lock-downs

4.2 Stationary Analysis

The stationary study of variables was performed using the traditional unit root test - Augmented Dickey-Fuller test. Weekly, monthly, and monthly selected features were all tested before modeling, and the results are summarized in Table 2. These results show that at any condition, most bonds yield rates and bonds related aggregated values are non-stationary *(based on the level basis t-statistic and confidence level)* and stationary in the first difference.

Table 2: Stationary Analysis Result

Monthly		Weekly	
Feature	P-Value	Feature	P-Value
MPRIME	0.124	bond30	0.208
VIXlog	0.061	bond1	0.231
UNRATEpct	0.0578	bond10	0.760
bond10	0.839	bond5	0.660
bond5	0.675	bond3	0.097
bond3	0.121	logcommodity	0.442
commoditylog	0.533	yield diff	0.069
SP500logstd	0.184		
yield diff	0.109		
b30std	0.085		
b1ret	0.448		
b1std	0.055		
risk free	0.628		

4.3 Grid Search

Hyper-parameters are parameters that are not directly learned within estimators. In scikit-learn package, they are passed as arguments to the constructor of the estimator classes. For example, in the Support Vector Classifier, we need to manually tune C, kernel, and gamma. In scikit-learn GridSearchCV, it is possible and recommended to search the hyper-parameter space for the best cross-validation score. Any parameter provided when constructing an estimator may be optimized in this function. All supervised model in our research is tuned to their best performance hyper-parameters.

5. Prediction Result

The result section is divided into 3 sections: Monthly, Monthly with Selected Features, and Weekly. Only selected supervised and unsupervised models' prediction result is shown in this section. All other classifier outcomes are plotted in the appendix. Each subsection will cover the selected prediction result first and followed by a summary of supervised model performance.

Since HMM and K-means are unsupervised models, the state classification is made based on a different standard than the supervised model. For further analysis, the machine classified states need to be manually converted to the same standard as the supervised model. In addition, K-Means is used as a baseline model due to its simplicity. Its result will not be covered in this section, but in the backtesting section as a baseline model to compare with other models.

In this project, **ROC and Cross-Validation score are not the major metric to value a model's performance.** They are provided as a reference for their general performance. **The ability to capture market turns correctly in the testing set and the ability to outperform the market in backtesting are more important.**

5.1 Monthly

The Monthly result section includes the modeling result from the monthly data set without feature selection. Figure 7 shows the detailed classification result of HMM model. Each color represents a market state. Due to the nature of HMM algorithm, prediction is made based on the state switching indicator. Therefore, HMM is an ideal model for persistent state classification. In Figure 7, with only 4 outliers(1 purple dot and 3 red dots), most dots are clustered together showing a persistent state.

Figure 7: Monthly HMM States Cluster

Figure 8 is the monthly-based prediction presentation of the HMM model with each color representing one state. State 0 is the state that is experiencing rapid growth; state 1 indicates it has expected economic expansion; state 2 indicates the market is experiencing a major crash. Shown as the green dots, the HMM model successfully detected the big market crash in 2002, 2008, late 2015, and 2020. In addition, it also observed the rapid market expansion from 2010 to 2015 and after COVID.

Figure 8: Monthly HMM Prediction Results

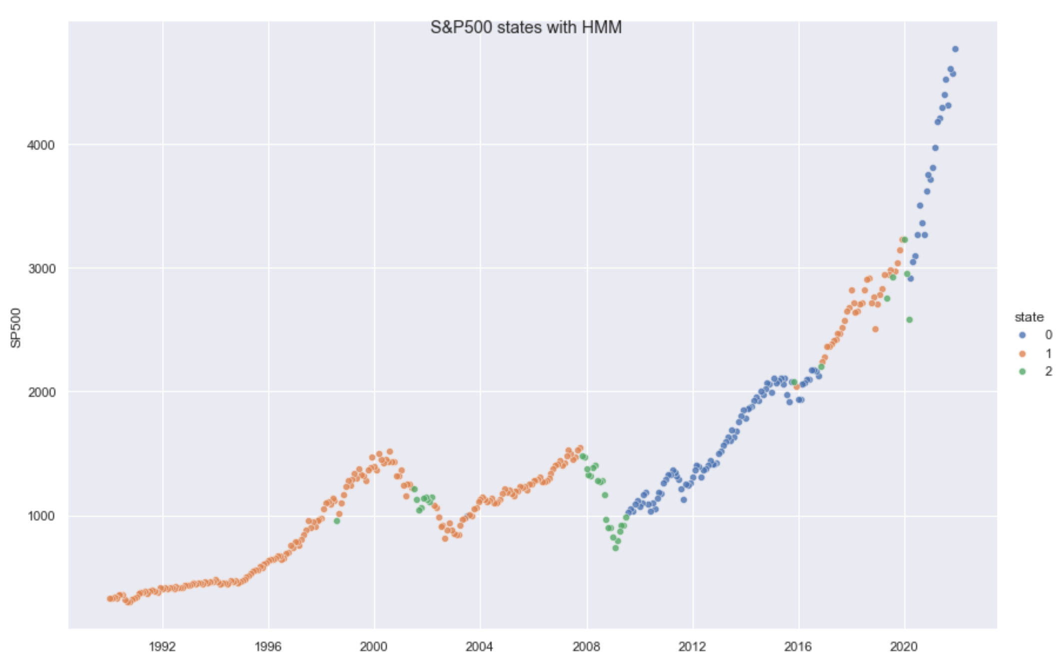


Figure 9 is the monthly-based prediction presentation of the Linear Discriminant Analysis(LDA) model with each color representing one state. Unlike HMM classification, state 0 is the state that is experiencing a major crash and state 2 indicates the market is experiencing rapid growth. Different from HMM, supervised models such as LDA tends to classify the state as more scattered and discontinued, which implies that each dot is an independent prediction. The LDA model successfully detected the market crash in 2002 and 2008. Moreover, as part of its testing, LDA captured the 2020 market crash and its afterward rapid expansion.

Figure 9: Monthly LDA Prediction Results

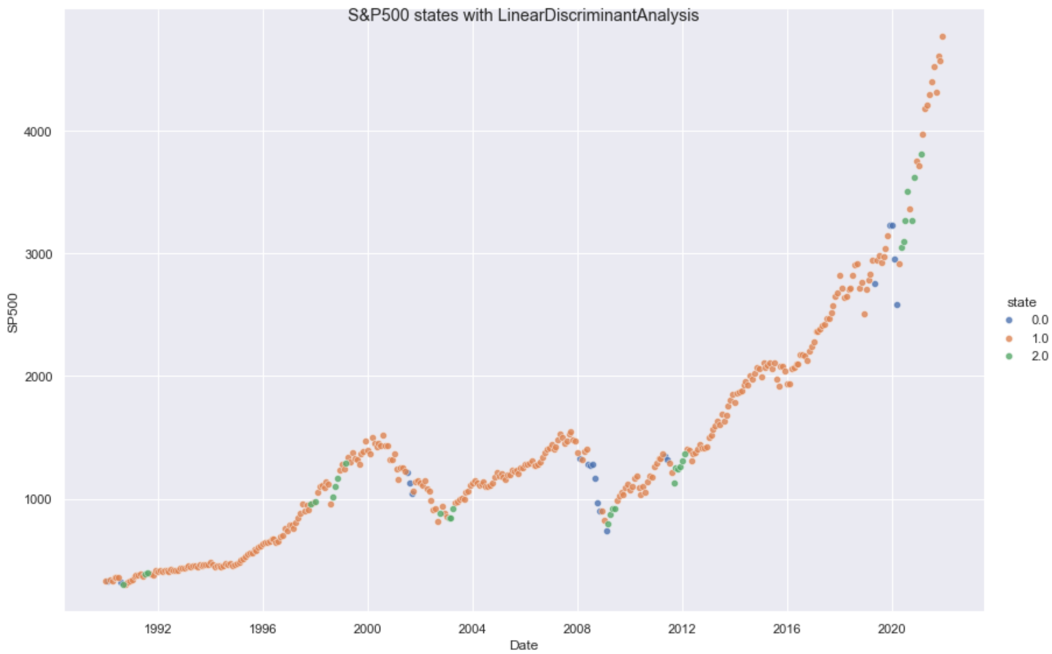


Table 3 represents the overall result of all monthly-based supervised models. Only 2 models failed to capture the market trends in 2020, which are the market crash in early 2020 and the rapid expansion afterward. The most important features in the monthly time frame are VIX standard deviation (VIXstd), CPI standard deviation (CPIstd), unemployment rate return (UNRATEpct), and bond-related features. Macroeconomic features do play an important role in monthly basis analysis.

Table 3: Monthly Prediction Result

Model Name	ROC	Cross Validation Score	Top Features	Capture 2020
LDA	0.867	0.788	commoditypct, commodityret, b1ret	Yes
LR	0.800	0.806	MPRIMEstd, bond5, bond3	Yes
SVM	0.563	0.792		No
DT	0.638	0.813	RealGDPstd, CPIstd, bond10	Yes
RF	1.000	0.827	VIXstd, CPIstd, UNRATEpct	Yes
Ada Boost	0.601	0.823	VIXstd	No
GBDT	1.000	0.818	VIXstd, UNRATEpct, SP500std	Yes

5.2 Monthly with selected features

In this research, the biggest benefit one can have in feature selection is the huge reduction in model training time. Figure 10 shows the detailed classification result of HMM model after feature selection. In Figure 7, we have more outliers in state 2 than the pure monthly prediction.

Figure 11 is the monthly based prediction presentation of the HMM model after feature selection. State 0 is the state that is experiencing rapid growth; state 1 indicates it has expected economic expansion; state 2 indicates the market is experiencing a major crash. The result in Figure 11 is the same as Figure 8. Feature selection has no impact on HMM.

Figure 10: Monthly with Selected Features HMM States Cluster

Figure 11: Monthly with Selected Features HMM Prediction Results

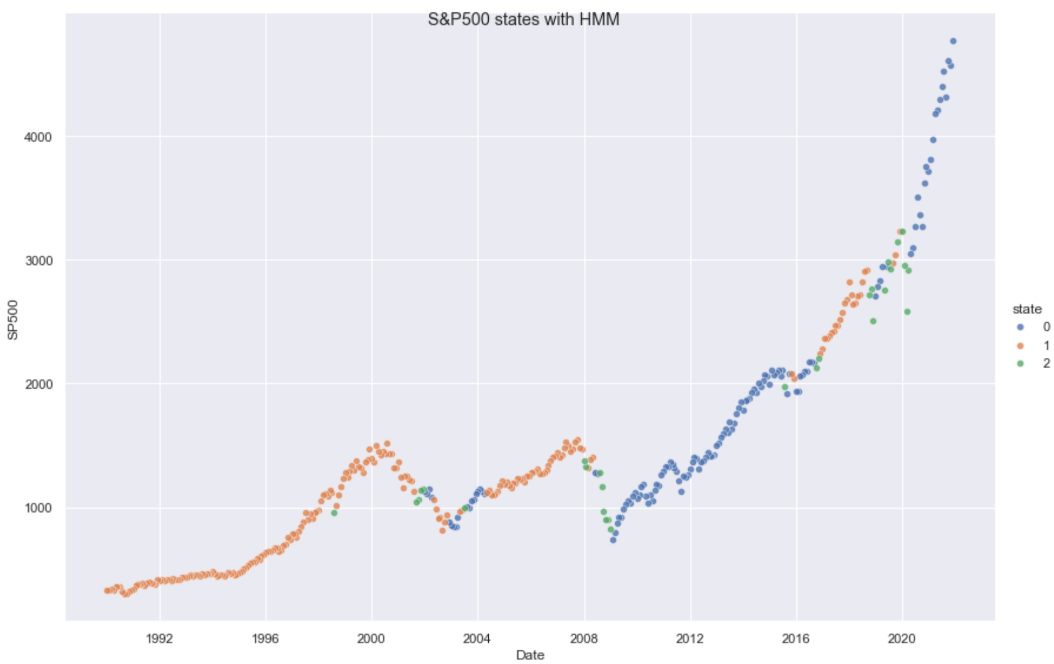


Figure 12 is the monthly based prediction presentation of the Linear Discriminant Analysis(LDA) model after feature selection. The LDA model also successfully detected the market crash in 2002 and 2008, followed by capturing the 2020 market crash and its afterward rapid expansion. Feature selection also has no impact in the supervised model such as LDA.

Figure 12: Monthly with Selected Features LDA Prediction Results

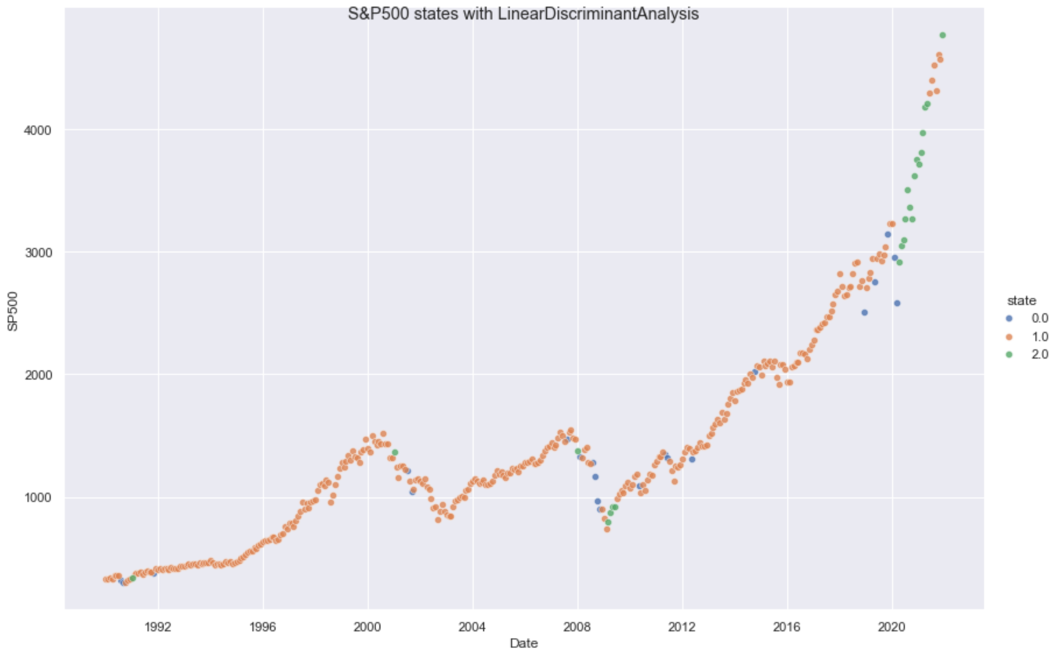


Table 4 represents the overall result of all monthly-based supervised models after feature selection. Comparing all feature monthly results, one more model failed to predict 2020 downturns and upturns. The ROC and Cross-Validation Score also dropped. This indicates that in this research feature selection will apply a small negative impact on our prediction. More features might lead to a more accurate model. The most important features in the feature selection test are real GDP standard deviation (RealGDPstd), unemployment rate return (UNRATEpct), and bond-related features. Macroeconomic features still played a dominant role in feature selection analysis.

Table 4: Monthly Selected Features Prediction Result

Model Name	ROC	Cross Validation Score	Top Features	Capture 2020
LDA	0.787	0.777	commoditypct, commodityret	Yes
LR	0.775	0.806	UNRATEpct, bond3, bond10	No
SVM	0.771	0.789		No
DT	0.628	0.806	RealGDPstd, bond10, b30std	Yes
RF	0.761	0.801	b30std, UNRATEpct, RealGDPstd	Yes
Ada Boost	0.601	0.797	UNRATEpct	No
GBDT	0.974	0.787	RealGDPstd, UNRATEpct, b1ret	Yes

5.3 Weekly

The weekly time frame is used to verify whether market regime-changing indication is more detectable at the weekly or monthly time frame. Figure 13 shows the detailed classification result of the weekly HMM model. In Figure 13, HMM classified state 1 as only one cluster. Unlike the other 2 trials, weekly HMM classified more market crashes and rapid expansion than any other model. This change is probably caused by the more prominent and volatile market change signals caused by the shorter time frame.

Figure 14 is the weekly-based prediction presentation of the HMM model. State 0 is the state that is experiencing rapid growth; state 1 indicates it has expected economic expansion; state 2 indicates the market is experiencing a major crash. Noticeably, unlike the other 2 monthly-based models, the Weekly model does predict most US market downturns, including minor crashes such as the Early 1990s recession, 1997 mini-crash, 2010 flash crash, and 2015 stock market selloff. This raises a concern that the weekly time frame might be a

Figure 13: Weekly HMM States Cluster

better candidate to detect minor market crashes. The state could be adjusted to only 2 states to capture the downturns accurately.

Figure 14: Weekly HMM Prediction Results

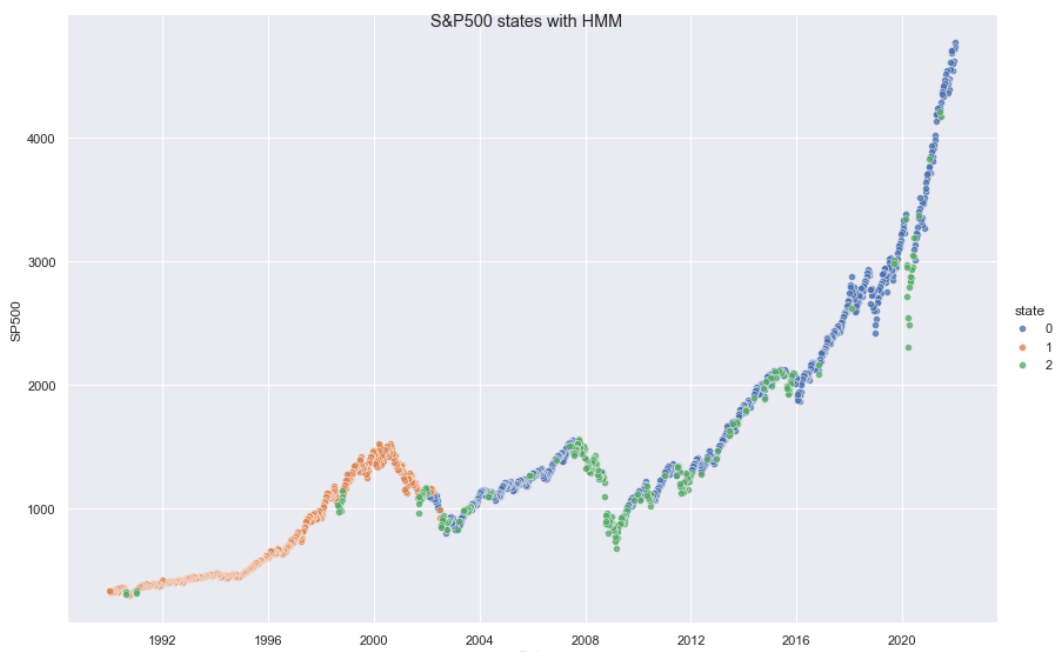


Figure 15 is the weekly-based prediction presentation of the LDA model. The LDA model also successfully detected the market crash in 2002 and 2008, followed by capturing the 2020 market crash and its afterward rapid expansion. However, the captured market change is very short.

In addition, LDA is the only supervised model that successfully captured the market change on the weekly basis. Combining results from HMM, on a weekly basis, it's better to classify the state into 2 states rather than 3. In a weekly time frame, models performed better at capturing transient market jumps, both short crashes, and growth. In addition, market-related features such as commodity and VIX return starts to outweigh macroeconomic features.

Figure 15: Weekly LDA Prediction Results

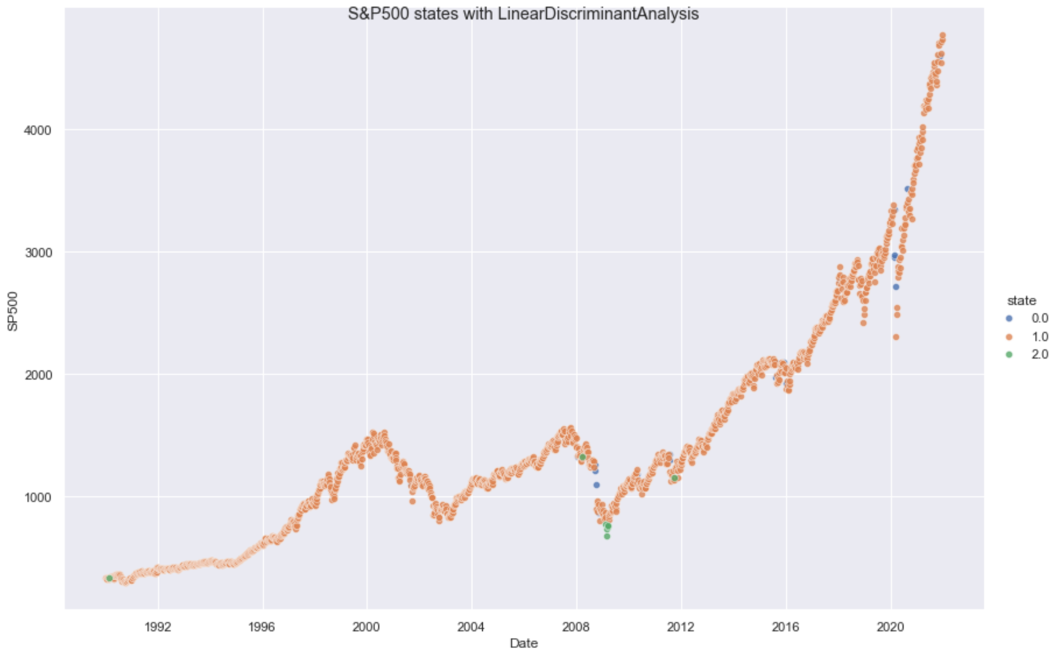


Table 5: Weekly Prediction Result

Model Name	ROC	Cross Validation Score	Top Features	Capture 2020
LDA	0.827	0.966	bond10pct, bond5pct, log return VIX	Yes
LR	0.775	0.985	log VIX	No
SVM	0.771	0.986		No
DT	0.690	0.983	commodityret, bond1, log commodity	No
RF	0.997	0.985	log return VIX, commoditystd, bond5pct	No
Ada Boost	0.918	0.985	VIXpct, log return VIX, commoditypct	No
GBDT	0.966	0.983	commoditypct, yield diff, log return VIX	No

6. Backtesting

Since cross-validation accuracy only determine how well the model can capture the calculated regime at each time step, this might not be an accurate representation of market regime-switching. For example, since most states fall into state 1, if a model predicts all data points as state 1, then the accuracy score can still be very high. ROC, confusing matrix, and F1 scores are all possible remedies to this issue. However, the best way to validate regime correctness is to design a backtesting strategy based on the predicted regime. If the predicted regime is a good representation of actual market changes, the return of the backtesting strategy should be much higher than the average market return.

We used SP500 cumulative return as the baseline for backtesting, which represents the return someone can get without any strategy. If our model predicted the regime correctly, the cumulative return should be higher than the cumulative SP500 return. For the simplicity of testing, the strategy is also based on the SP500 return.

Our strategy can be summarized in 3 conditions: (1) if the state equals 0, short the market, add negative current SP 500 return to the cumulative return; (2) if the state equals 1, follow the market, add positive current SP 500 return to the cumulative return; (3) if the state equals 2, long the market, add positive 1.5 times current SP 500 return to the cumulative return. Table 6 summarized the cumulative return result of backtesting. Each number represents the cumulative return of this model starting from 1990 to the end of the year 2021. For example, 2.95 means someone can get 290% of return if they follow the market since 1990. This also implies if a model classifies most of its state as state 1, the return should be very close to the market return baseline of

2.95. The higher the cumulative return, the better the model performance. Table 7 calculated the Sharp ratio of each model, which is derived from the backtesting return. The higher the Sharp ratio the better the model.

Table 6: Back Testing Cumulative Return Result

Model Name	Monthly	Monthly Selected Feature	Weekly
Market Baseline	2.95	2.95	2.95
Linear Discriminant Analysis	5.74	5.65	4.41
Logistic Regression	3.42	3.72	3.11
Support Vector Machine	5.59	3.16	3.63
Decision Tree	4.21	4.26	3.11
Random Forest	5.86	4.18	3.58
AdaBoost	4.44	4.31	3.81
Gradient Boosting	6.12	5.45	3.82
K-means	2.06	1.64	4.07
Hidden Markov Model	5.49	5.76	4.34
Total Good	5	3	2
Total Bad	1	2	2

Table 7: Sharp Ratio of Each Model

Model Name	Monthly	Monthly Selected Feature	Weekly
Linear Discriminant Analysis	1.797	1.975	1.195
Logistic Regression	0.965	0.866	0.046
Support Vector Machine	1.640	0.230	0.583
Decision Tree	1.068	1.225	0.046
Random Forest	1.740	1.188	0.624
AdaBoost	0.385	1.252	0.753
Gradient Boosting	1.784	1.863	0.572
K-means	-1.285	-0.022	0.893
Hidden Markov Model	1.791	1.870	1.070
Total Good	5	3	2
Total Bad	1	2	2

It's clear to say that **Monthly trial outperforms all other datasets.** The best model is colored green and the worst model is colored red. The bottom metric represents the number of good model and the number of bad model. Monthly trial has the highest overall return, with the lowest being K-means model. Feature selection do have a negative impact in overall performance, with the highest return decrease on the SVM and Random Forest model. **Surprisingly, Hidden Markov Model benefited the most from feature selection.** All other models still show a similar result with the result before feature selection. Although has the highest cross-validation score and ROC score, weekly trial has the worst performance among all three of them. This implies that the weekly models predict most states as state one, the state follows the market, and failed to detect market regime change. However, unsupervised model K-Means and Hidden Markov Model still have good performance on weekly time frame.

Figure 16, Figure 17, and Figure 18 represents the monthly, monthly feature selected, weekly over time back testing result. The X-axis is the date and the Y-axis is the cumulative return. In Figure 16, all model successfully predicted the regime before 2000. Random Forest, Linear Discriminant Analysis, GBDT, and HMM detected the bear market on 2002 and 2008 better than others. Their cumulative return lines separated from others at these 2 time points. Linear Discriminant Analysis predicted the 2008 recession the best as the orange line has a major return jump at 2008. HMM captured the short COVID recession and afterwards boom precisely as the blue line has a major jump at 2020. The worst prediction is the K-means model, the model clustered most data points before 2008 as state 1, and failed to recognize the market change during COVID time; the olive line stayed very close with the market based line before 2008 and experienced almost no return after 2015.

Figure 16: Monthly Cumulative Return

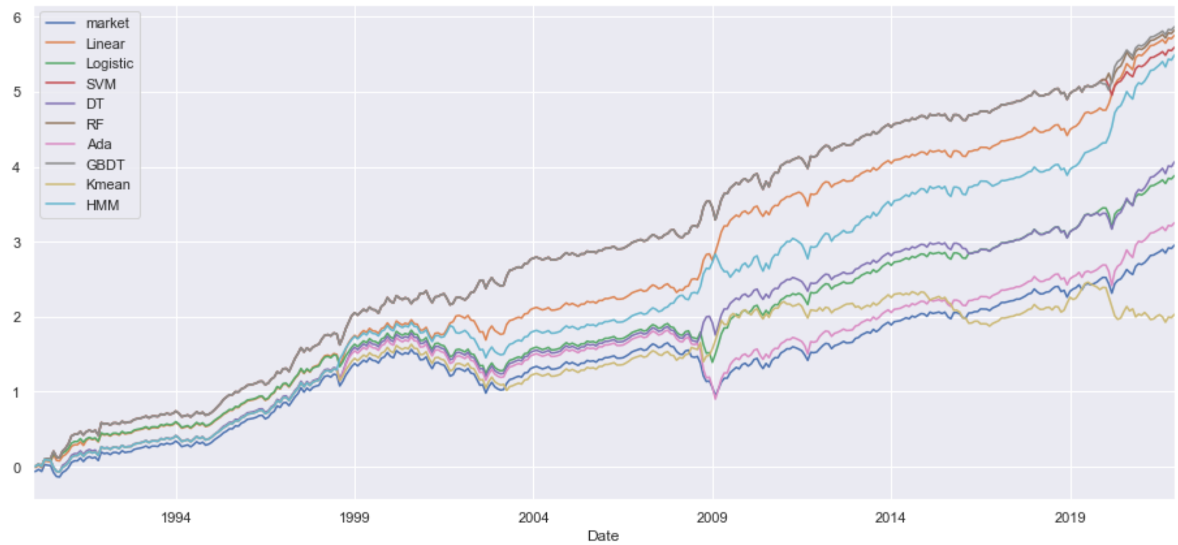


Figure 17 has similar trend with Figure 16. Linear Discriminant Analysis, GBDT, and HMM outperform all other models. Their cumulative return lines separated from others in 2008 and 2002. They predicted the 2008 recession the best as their line have a major return jump in 2008. HMM captured the short COVID recession and afterward boom precisely again as the blue line has a major jump at 2020. K-means model has almost the same trend after feature selection and an even worse return.

Figure 17: Feature Selection Monthly Cumulative Return

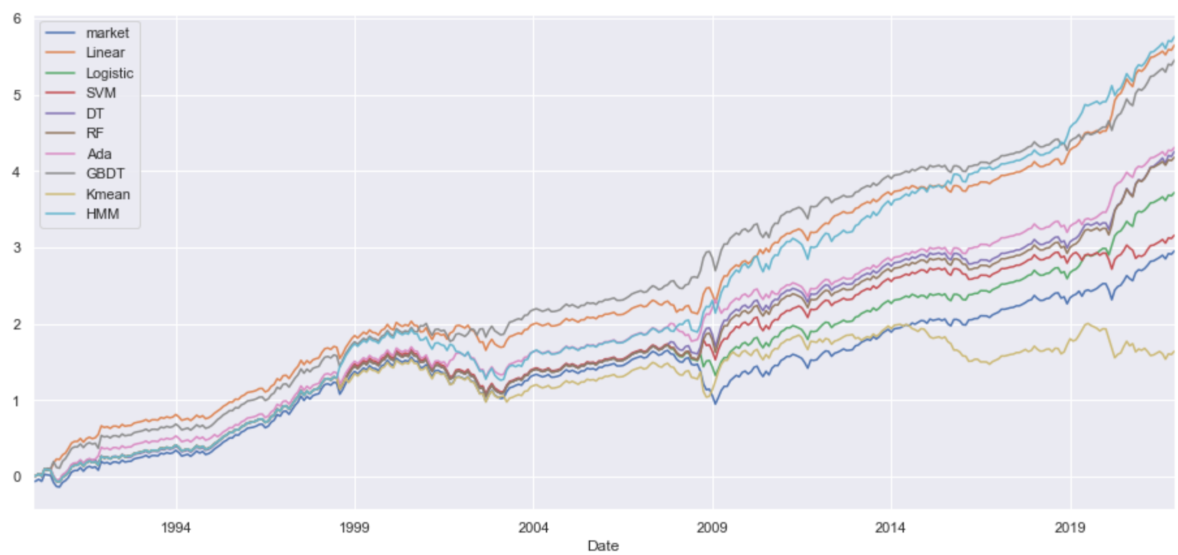


Figure 18 has the worst result among the 3 of them. Only Linear Discriminant Analysis and HMM generated an acceptable return. Linear Discriminant Analysis starts to separate from others in 2015 and experienced a major jump in 2020, which indicates that on a weekly basis Linear Discriminant Analysis failed to detect any recession. However, HMM experienced a big return decline in 2002, which implies HMM completely failed to detect the recession regime around 2002. HMM return starts to jump back in 2007 and surpass others after 2015. HMM still captured the short COVID recession and afterward boom precisely at the weekly time frame, and this implies HMM still has the highest cumulative return if we skip the 2002 recession. Surprisingly, the K-means model gained a positive return the first time. This might be caused by the 4 times larger data sets we have for the weekly time frame.

Figure 18: Weekly Cumulative Return

7. Conclusion

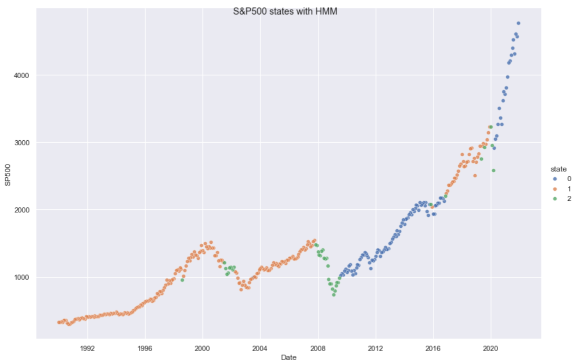
This analysis shows why Machine Learning techniques are extremely powerful tools in market regime detection. First, Machine Learning techniques are capable of providing evidence of significant relationships between public economic features and persistent market regimes. This paper explored 9 different models and discovered the best model for market regime detection are Hidden Markov Model and Linear Discriminant Analysis. Tree-based models such as Gradient Boosting Decision Tree are the second-best model. Second, the time frame analysis allows people to grasp additional information about when a market regime switch is going to occur. Market regime-switching is more likely to happen and easier to detect on a monthly basis. Lastly, feature selection can help improve the efficiency of modeling without the sacrifice of market regime classification accuracy. With only 20 features machine learning models can generate precise market regime classification.

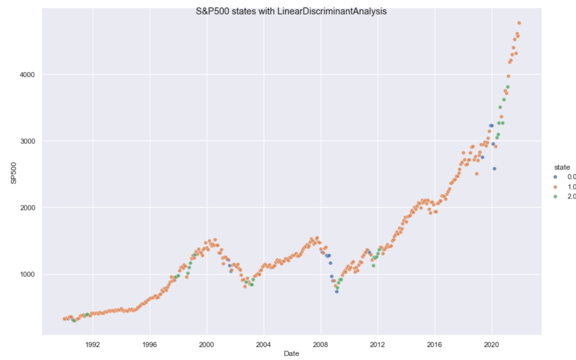
Investors can take advantage of the above knowledge to construct portfolios that profit from underlying market states. Overall, public economic information contains powerful predictive information that can not only be used to obtain profits but also helps people to become more conscious of increasingly concerning topics such as economic recessions.

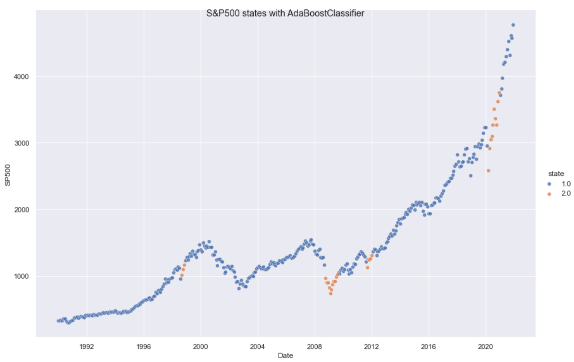
Appendix

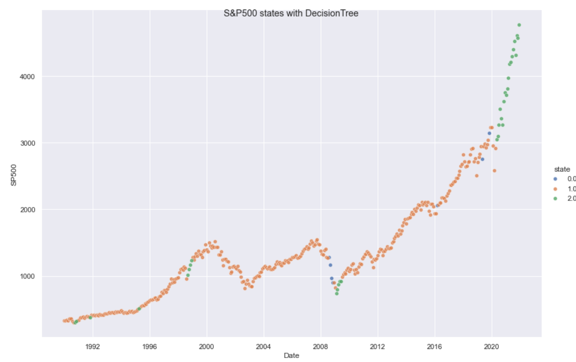
A Classification outcome of every approach

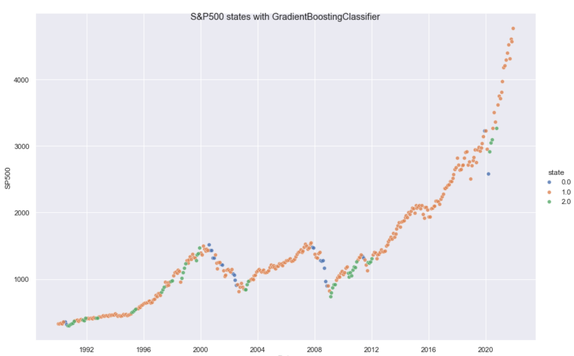
A.1 Original Monthly Dataset

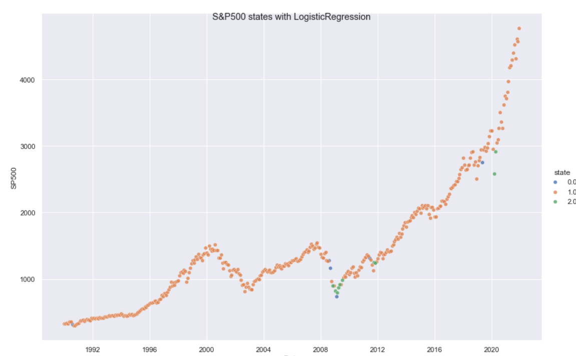


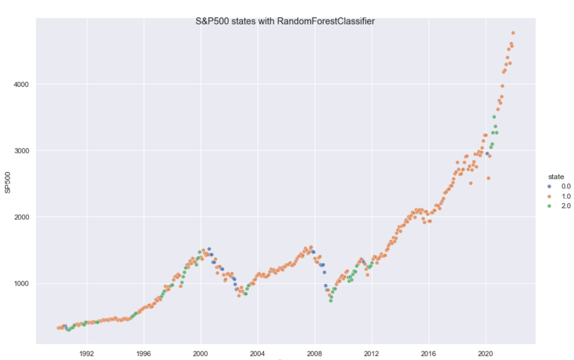


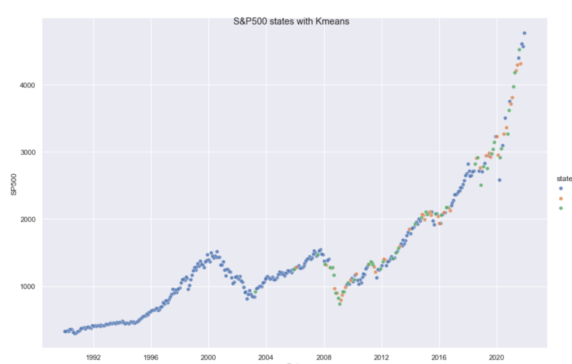


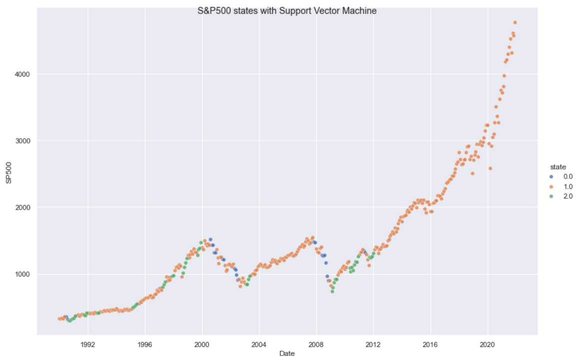




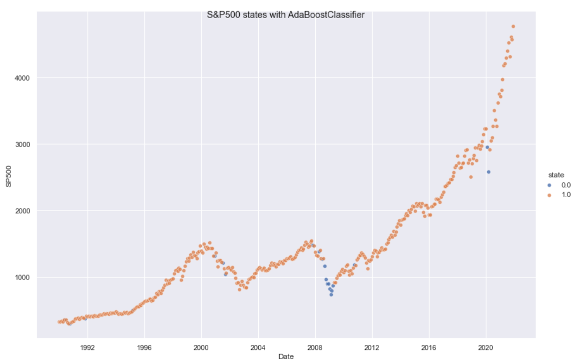


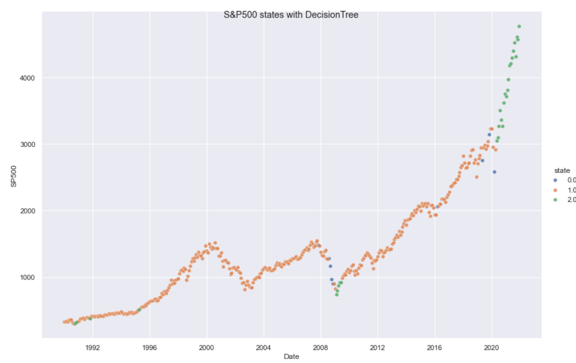


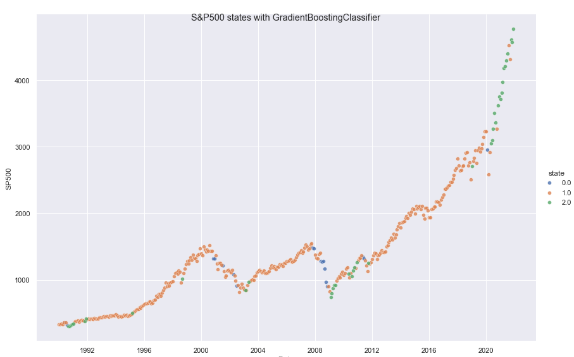


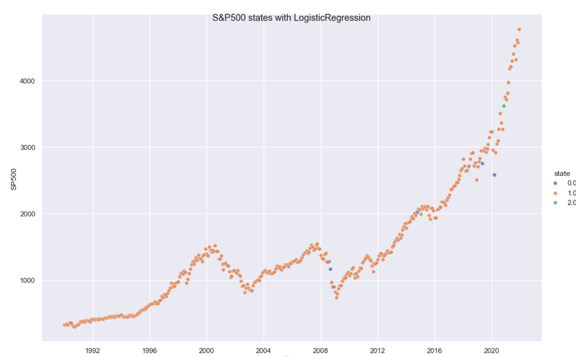


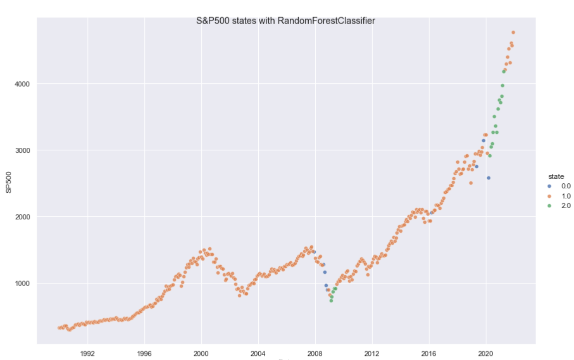
A.2 Monthly Dataset with Selected features

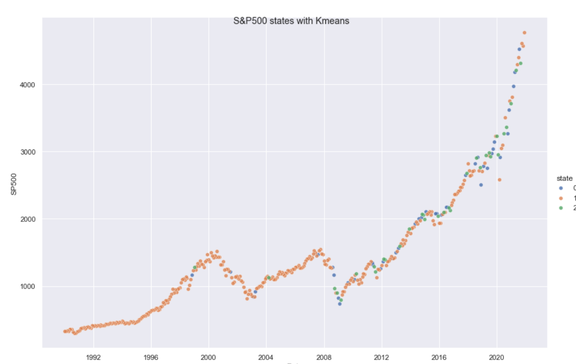


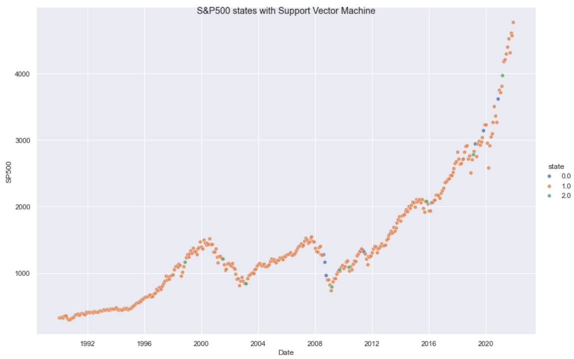


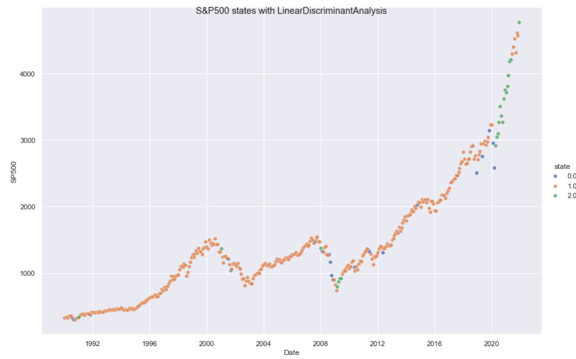


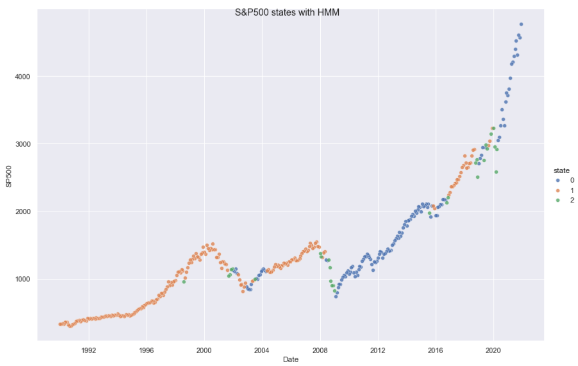




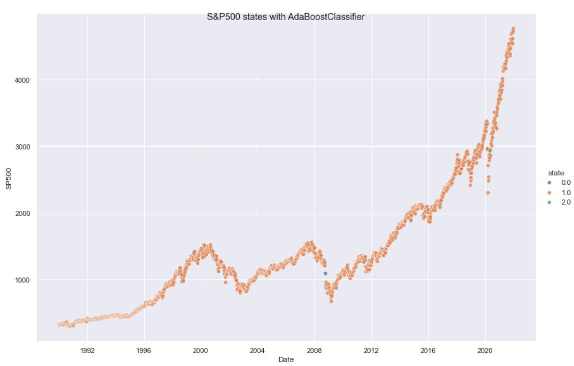


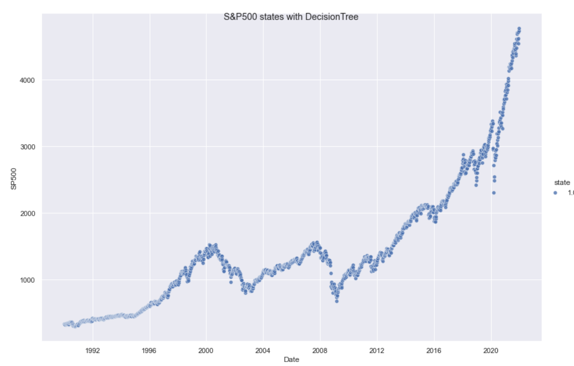


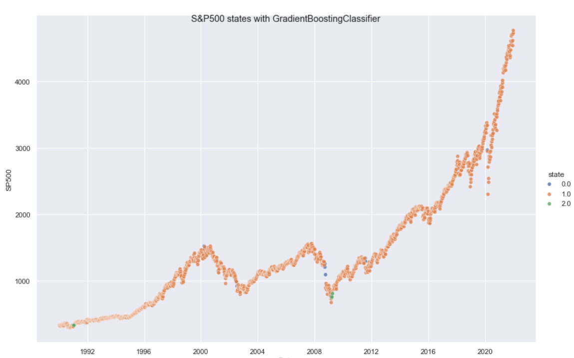


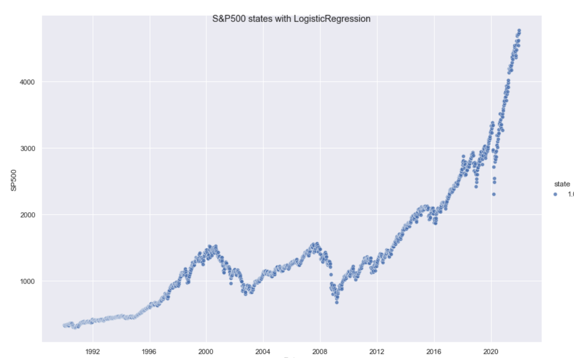


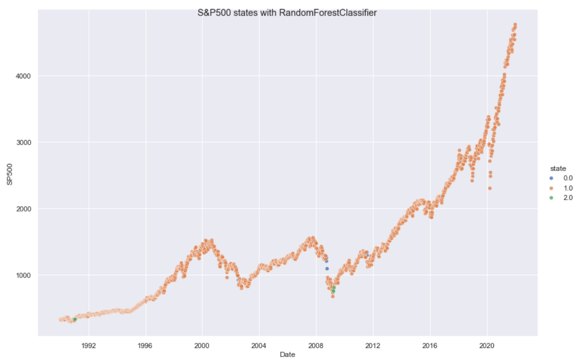
A.3 Weekly Dataset

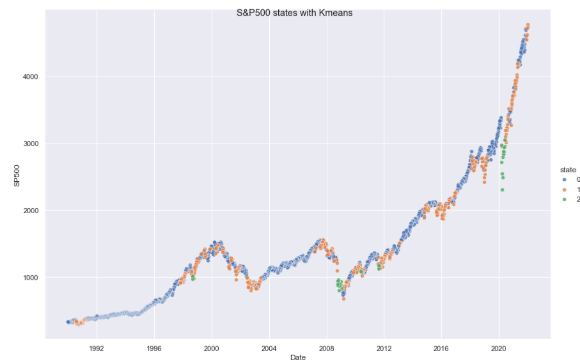


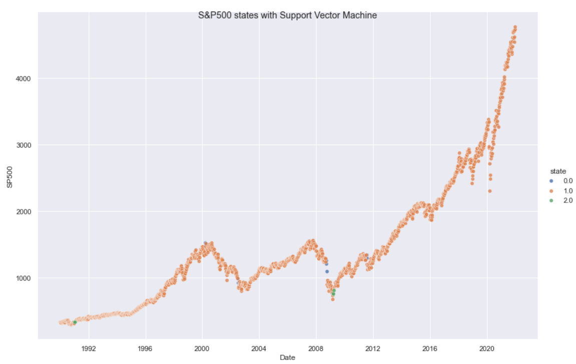


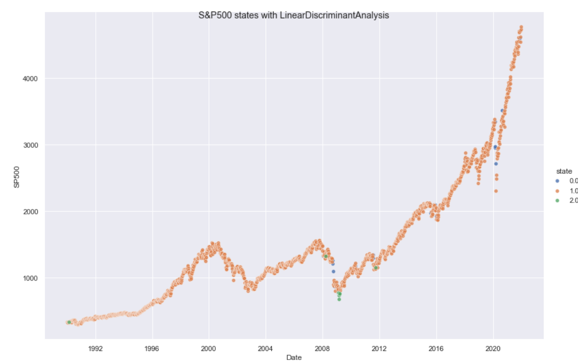


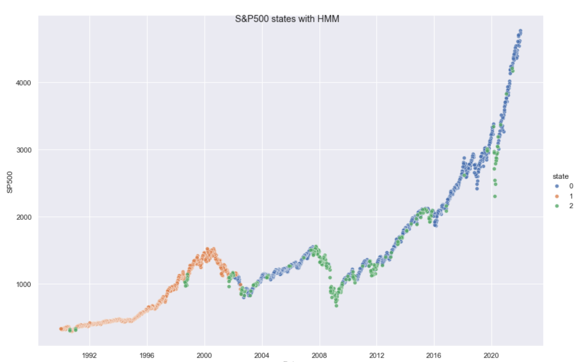



S&P500 states with RandomForestClassifier
SP500
4000
3000
2000
1000
1992
1996
2000
2004
2008
2012
2016
2020
Date
state
0.0
1.0
2.0


S&P500 states with Kmeans
SP500
4000
3000
2000
1000
1992
1996
2000
2004
2008
2012
2016
2020
Date
state
0
1
2


S&P500 states with Support Vector Machine
SP500
4000
3000
2000
1000
1992
1996
2000
2004
2008
2012
2016
2020
Date
state
0.0
1.0
2.0


S&P500 states with LinearDiscriminantAnalysis
SP500
4000
3000
2000
1000
1992
1996
2000
2004
2008
2012
2016
2020
Date
state
0.0
1.0
2.0


S&P500 states with HMM
SP500
4000
3000
2000
1000
1992
1996
2000
2004
2008
2012
2016
2020
Date
state
0
1
2

References

[1] Michael Messer. Bivariate change point detection: Joint detection of changes in expectation and variance. *Scandinavian Journal of Statistics*, 2021.

[2] Michael Schatz. Financial modeling of bubbles and crashes. *ETH Research Collection*, 2020.

[3] Peter Nystrup, Erik Lindström, and Henrik Madsen. Learning hidden markov models with persistent states by penalizing jumps. *Expert Systems With Applications*, 150, 2020.

[4] Jun Chen. Studying regime change using directional change. 2019.

[5] Georgi Smilyanov Didier Sornette, Peter Cauwels. Can we use volatility to diagnose financial bubbles? lessons from 40 historical bubbles. *Swiss Finance Institute*, N17-27, 2017.

[6] Lanbiao Liu, Chen Chen, and Bo Wang. Predicting financial crises with machine learning methods. *Journal of Forecasting*, 2020.

[7] Tsutomu Watanabe Takayuki Mizuno, Takaaki Ohnishi. Detecting stock market bubbles based on the cross-sectional dispersion of stock prices. *National Institute of Informatics*, 2019.

[8] Peter Akioyamen, Yi Zhou Tang, and Hussien Hussien. A hybrid learning approach to detecting regime switches in financial markets. *Proceedings of the First ACM International Conference on AI in Finance*, 2020.

[9] Eric Benhamou, Jean Jacques Ohana, David Saltiel, and Beatrice Guez3. Planning in financial markets in presence of spikes: using machine learning gbdt. *Université Paris-Dauphine Research Paper*, 2021.

[10] Eric Benhamou, Jean Jacques Ohana, David Saltiel, and Beatrice Guez. Explainable ai (xai) models applied to planning in financial markets. *Université Paris-Dauphine Research Paper*, 2021.

[11] James Ming Chen, Mobeen Ur Rehman, and Xuan Vinh Vo. Clustering commodity markets in space and time: Clarifying returns, volatility, and trading regimes through unsupervised machine learning. *Resources Policy*, 150, 2021.

[12] Jun Chen and Edward P K Tsang. Detecting regime change in computational finance data science, machine learning and algorithmic trading. *CRC Press*, 2020.

[13] Fatma Başoğlu Kabran and Kamil Demirberk Ünlü. A two-step machine learning approach to predict sp 500 bubbles. *Journal of Applied Statistics*, 48:2776–2794, 2020.

[14] A. Sinem Uysal and John M. Mulvey. A machine learning approach in regime-switching risk parity portfolios.

[15] HENRIK MADSEN PETER NYSTRUP, BO WILLIAM HANSEN and ERIK LINDSTRÖM. Regime-based versus static asset allocation: Letting the data speak. *The Journal of Portfolio Management*, Fall 2015, 2015.

[16] Henrik Olejasz Larsen Henrik Madsen Peter Nystrup, Bo William Hansen and Erik Lindström. Dynamic allocation or diversification: A regime-based approach to multiple assets. *The Journal of Portfolio Management*, Multi-Asset Special Issue 2018, 2018.

[17] Etienne Gael Tajeuna, Mohamed Bouguessa, and Shengrui Wang. Modeling regime shifts in multiple time series. *Environmental Science and Pollution Research*, 2021.

[18] Kai Zheng, Yuying Li1, and Weidong Xu2. Regime switching model estimation: spectral clustering hidden markov model. *Annals of Operations Research*, 303:297–319, 2019.

[19] Manuela Pedio Giulia Dal Pra, Massimo Guidolin and Fabiola Vasile. Regime shifts in excess stock return predictability: An out-of-sample portfolio analysis. *The Journal of Portfolio Management Finance*, 2018.

[20] Vito Ciciretti Andrea Bucci. Market regime detection via realized covariances: A comparison between unsupervised learning and nonlinear models. *Statistical Finance*, 2021.