Market States Classification and Prediction

Introduction

"Do not time the market" is common advice given to investors since perfectly predicting market movement is a challenging task. However, what if we can forecast the future state of the market?

In this paper, we shall define three latent market regimes, namely, bull market, static market, and bear market. Bull and bear markets represent the two extreme ends of the spectrum with the bull market representing a period of optimism, the bear market representing a period of pessimism, and the static market representing market conditions that are in between.

Starting with daily time series prices of Russell 3000, we employ an ensemble method that combines three classifiers to produce a unified, improved classification of market states. The classification labels along with additional carefully selected features are fed into our prediction model to produce market state labels for 2018-2021. Using the predicted labels along with our trading strategy, we were able to beat the buy-and-hold strategy.

Methodology and approach

To accurately forecast the future state of the market, we collected an exhaustive set of macroeconomic and technical data from various sources. We then cleaned and pre-processed the raw data for model fitting. Due to the temporal nature of financial data, we chose to split them into training set, validation set, and test set by time chronologically instead of random shuffling. In particular, the training set used for model fitting consisted of data from 2005 to 2014; the validation set used for hyperparameter tuning consisted of data from 2015 to 2017, and the test set used for model evaluation consisted of data from 2018 to 2021.

There is no universal agreement on how to define market states, so, before fitting the supervised prediction models to the training set, it is necessary to classify the training periods into corresponding market states. Specifically, an ensemble method that combined a mean-standard deviation model, a Markov switching model, and a k-means clustering model classified the market into different states. Based on the results, we classified the periods with relatively high daily returns but relatively low volatility as a bull, the periods with relatively low (even negative) daily returns but relatively high volatility as a bear, and other periods as static. It is worth mentioning that the method only labeled market states up to 2017 to avoid biasing prediction models.

After the training set was labeled, two prediction models were fitted. First, a random forest model using the entire data set was fitted. Second, a neural network model using a lower-dimensional representation of the data set was fitted. The output labels from these two models were then post-processed based on several conditions to form the final predicted market states. To evaluate and compare the performance of the two prediction models, we developed a trading strategy, backtested the predicted labels on the test set, and computed the cumulative returns.

Data and preprocessing

This section briefly summarizes the selected features and the reasoning to use them as predictors. The selection of time period as well as the methodology of handling missing data are also discussed in this section.

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, five groups of data are considered as predictors of future market states. Below is a list of data used and the corresponding rationales. Interest rate data are collected from Federal Reserve Economic Data. Other data is collected from Yahoo Finance.

- Interest rates: Daily T-bill rates data, Moody's seasoned Aaa corporate bond yield, Moody's seasoned Baa corporate bond yield. Interest rates are an 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, CNY/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.
- Major stock indices and VIX index: S&P 500, Dow Jones Industrial Average, Nasdaq, Russell 2000, FTSE 100, Nikkei 225, Hang Seng Index, VIX. The globalization of investment means that stock indices in different countries are more correlated than before. Changes in market states in one region can rapidly spread to another country. The examinations of momentum and reversion effects in previous literature also suggest the predicting ability of past stock index returns on future index performance. The VIX index is a strong indicator of volatility in the market.
- Commodities prices: WTI cushing crude oil spot price index, gold price. 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.

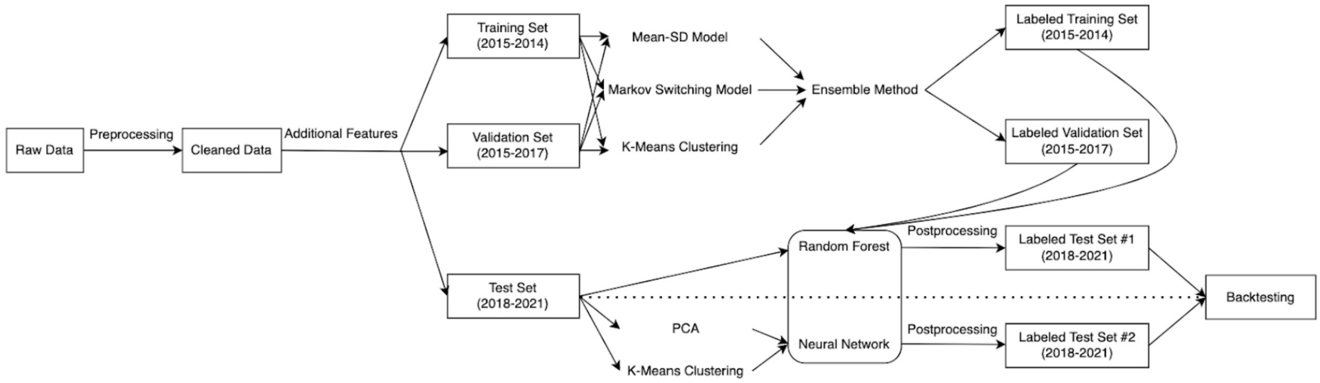

Figure 1: Flow chart of our overall methodology

- Prices of cyclical stocks: Apple, Microsoft, Exxon, General Electric, Johnson Johnson, Wells Fargo, Amazon, JPMorgan Chase, Walt Disney. Cyclical stocks usually have high betas with regard to the market return. These market movers can both reflect changes in market states and change market states.

Feature engineering techniques were used to create the final feature set. For example, daily percentage changes of gold price and oil price are calculated. The technical signals of the Russell 3000 itself is also calculated. A table of all features used to predict market states is included in the appendix.

Data from 2000 to 2014 are used as the training set, data from 2015 to 2017 serve as the validation set for model development, and the trading performance is tested on data from 2018 to 2021. This time frame captures the 2007 global financial crisis and guarantees that there is sufficient data for model fitting and development. Data before 2000 are not used to exclude out-of-date information.

When dealing with missing data, the geometric mean of previous and later available data is taken to fill the missing values. When there are consecutive missing values, they are dynamically filled using the same process from oldest date to newest date.

Market States Classification

To investigate the properties of different market states and predict future market states, the underlying trend of the stock market needs to be classified. In this paper, information about the Russell 3000 index including daily high, low, close, and open prices is used to determine different market states in history. In the past decades, there have been an extensive number of studies on market state classification based on stock return dynamics. Yet there are few formalized definitions and concepts that are widely accepted as the standard for determining market trends. Existing literature about division of market states significantly highlight bull and bear market state classification while only a small proportion of studies also include static market as the third state. Both threshold-based subjective methods and model-based objective models are used to approach this problem. For threshold-based methods, Frank and Jack (1977) proposed three alternative ways of defining market conditions through the comparison of past mean returns and past returns' standard deviations. For model-based objective methods, the Markov regime-switching model is widely used in this topic. However, most studies using the Markov regime-switching model divide the market into two states following Hamilton's (1989) work. In this report, we extend the application of the Markov regime-switching model to three market states classification and introduce the second method-based method using K-means clustering to identify states. The following part uses three different methods to classify the market into three states. To capture both subjective and objective information about the market, these three methods are individually applied and then combined in ensemble. The combined model shows the results of comprehensive states identifications. Finally, the results of all 3 individual models as well a final ensemble model are presented for comparison.

Mean-SD Model (Model 1)

Frank and Jack (1977) define substantial up and substantial down. They then classify the market as moving either substantial up or substantial down on monthly returns calculated on a rolling daily basis. Then, the standard deviations of monthly returns in the past 20 days are also calculated on the daily basis. To compare the monthly return and standard deviation, an arbitrarily chosen threshold parameter is needed. In this paper, both parameters for the substantial up and the substantial down are set 1.5. Specifically, we take the logarithmic return of Russell 3000 by its daily close price $r_t = log(\frac{y_t}{y_{t-1}})$ where y_t is the close price of Russell at date t. Then, monthly returns and standard deviations are calculated using the logarithmic returns. If today's monthly return is bigger than 1.5 times today's monthly return standard deviation, then the model identifies today as a bull. If the return is smaller than -1.5 times the standard deviation, then the model identifies today as a bear. Otherwise, it classifies today as a static market.

Markov Regime Switching Model (Model 2)

In this model, the daily logarithmic returns of Russell 3000 are used. Assume that the logarithmic returns r_t follows a first-order 3-states Markov chain, and for each state their means and variances are different. The model is as follows:

$$r_t = c_{S_t} + e_t$$

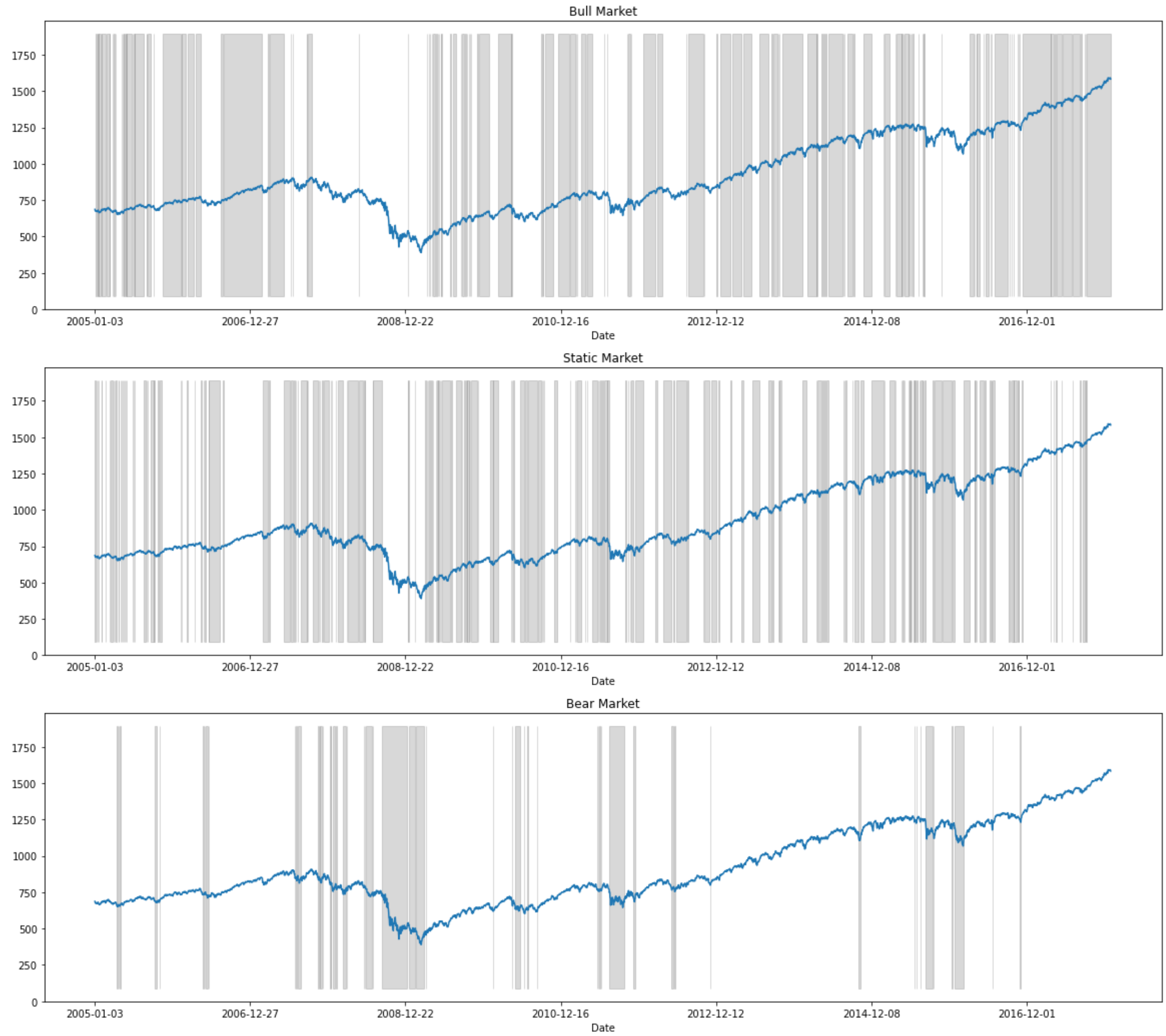

Figure 2: Classification of each market state in gray

$$e_t \sim i.i.d\, N(0, \sigma^2_{S_t})$$

S_t is the index of three different regimes, $S_t \in \{1, 2, 3\}$, c_{S_t} is the state depend mean, and e_t is the state dependent disturbance with mean 0 and state dependent variance $_{(}S_t)^2$. The probability of changing from one state to another state is governed by the transition matrix.

$$p_{ij} = P(S_t = j | S_{t-1} = i) \; for \, \forall i, j \in \{1, 2, 3\},$$

$$\sum_{j=1}^{3} p_{ij} = 1$$

Where p_{ij} is the probability that the market states will change from state i to state j. Under this specification, the transition matrix is as P:

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \tag{1}$$

The parameters in this model are estimated using Hamilton filtering method and maximum likelihood method. Table 1 shows the estimation results for this 3-states Markov regime switching model and the transition matrix.

Table 1: The parameter estimation results of the Markov regime-switching model

	State 1	State 2	State 3
C_{S_t}	0.001	-0.0002	-0.0017
$\sigma^2_{S_t}$	0.00003	0.0001	0.0008

Table 2: The transition probability matrix of the Markov regime-switching model

	State 1	State 2	State 3
State 1	$p_{11} = 0.97$	$p_{12} = 0.03$	$p_{13} = 0.00$
State 2	$p_{21} = 0.03$	$p_{22} = 0.96$	$p_{23} = 0.01$
State 3	$p_{31} = 0.00$	$p_{32} = 0.02$	$p_{33} = 0.98$

From the transition matrix, once the market enters a state (bull, bear, or static), there is more than a 96% chance that it will remain in the same state for the next day. Therefore, three of these states are persistent. To determine the market states, the smoothed marginal probabilities on each day of the three market states are calculated. Then we classify the market state as the state with the highest smoothed marginal probabilities.

K-means Clustering Model (Model 3)

Market classification can be regarded as a problem of clustering market conditions from the perspective of time. The characteristics of stock market is different every day with different market condition. Therefore, it is heuristic that these features can be used to cluster daily market dynamics. K-means is an unsupervised clustering algorithm that finds local optimum. For n sample points, they are clustered into K different classes. Each class has a center called centroid. For market classification, different days from 2005 to 2017 are classified into three clusters based on various features of Russell 3000. Then these three clusters are defined as bull, bear, and static market state based on state-wise mean returns and its t-value for significance test. Specifically, technical factors, different exchange rates, major stock indices, cyclical stock prices, commodities information and interest rates are put into the K-means model as features. The steps of K-means clustering algorithm are as follows. Consider different days $t_1, t_2, \dots, t_n$ with m features every day. First, three centroids of the three states are randomly chosen as k_1, k_2, k_3 where $k_i, i = 1, 2, 3$, are n dimension vector. Second, the square of distance between the features of each day and each centroid is calculated as $||S_i - k_n||_2^2$. Third, assign each day to the closest centroid based on the distance calculated as above. Fourth, for each group around a centroid, a new center can be calculated based on these days in a cluster as $\frac{1}{|C_n|}\sum_{i \in C_n} t_i$, where C_n is the group of days corresponding to k_n, $|C_n|$ is the number of days around k_n. Fifth, continue step 2 until the model reached convergence.

Ensemble Model

Ensemble model aims to combine the results given by the three models built above. For each day in the history, these three models will output three classification results. These three results vote for the market states. For each day $t_1, t_2, \dots, t_n$, assume the three models present results as $M1 = \{s_1^1, s_2^1, \dots, s_n^1\}$, $M2 = \{s_1^2, s_2^2, \dots, s_n^2\}$ and $M3 = \{s_1^3, s_2^3, \dots, s_n^3\}$. Then, the final result of classification $state_i$ of day i, $i = 1, 2, 3, \dots n$, are decided based on the following rule. If more than two models output the same state among $\{s_i^1, s_i^2, s_i^3\}$, then $state_i$ is set to be that state. If three models give three different states i.e., $s_i^1 \neq s_i^2 \neq s_i^3$, then set $state_i$ as static as this period shows high uncertainty and these three models cannot reach a consensus. Table 3 shows the break down of market returns for each individual model as well as the overall ensemble classification.

Classification Results

This model classification problem is unsupervised, so we cannot evaluate the model by standard metrics that are used to judge supervised methods. As a result, we compare statistics within each market regime to create a summary of our classification algorithm and check if these results are consistent with general qualities of each market states documented in classic financial studies. Table 3 compares three different models as well as the ensemble model based on the mean returns, standard deviations and t-values of returns in each state. Table 4 depicts many of the same statistics shown in Gonzalez (2005). From table 3, there are obvious trends that stand out, namely the average daily market return being significantly positive for bull markets, almost 0 in static markets, and significantly negative in bear markets. These results are consistent throughout all three models. The volatility in bull markets is the lowest with daily return standard deviation ranging from 0.53% to 0.69% while bear markets have the highest standard deviation with more than 2%. From table 4 and the ensemble part in table 3, we can check the features of the final classification results i.e. the labels used later in machine learning models. Based on statistics in table 4, our model only classifies 13.1% of days on the training set as bear market while the other two states contain 48.8% and 38.2% of days. The average daily return in bull markets is significantly positive with 8.59 t-value and significantly negative in bear markets with -2.76 t-value. Daily returns in static markets are almost 0 with 0.22 t-value. Another intuitive statistic is the percentage of days with positive return in each market state as the bull market has the strongest performance at 60.65%. The average duration before switching market states is longest in bull markets, with an average of almost 11 days, while statics market states only last about 6 days, on average.

Market States Prediction

Phase 1: Random Forest/Neural Network

Random Forest

Random forest is a common ensemble classification method that uses decision trees. Random forest extends the idea of bagging - where several independent decision trees are built in parallel and collectively classify the outcome. More formally, Breiman (2001) defines random forests as "a classifier consisting of a collection of tree-structured classifiers $h(x, \theta_k), k = 1, \dots$ where the θ_k are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x." Random forests sample a random set of features for each split in a tree and as a result, give more robust (lower variance) results than simple bagging techniques.

We train our random forest classifier on data prior to 2015 with the goal of a binary classification. We later use this binary classification to develop a trading strategy. We train the model with 500 trees (using gini impurity) and also employ balanced class weighting. Due to the imbalanced nature of our input, we are weighting the classes based on the inverse of the weighting in the training dataset - leading to balanced class weights. This weighting procedure prevents the model from biasing towards the more frequent class by using the weighting during the decision of splits. Two random forests are trained for bull and static states respectively. Take the bull market binary classification random forest as an example, the days with the bull state are labeled as 1 while the other states are both labeled as 0. The outputs of random forests are possibilities of being in the bull state. A threshold is set such that if the probability is bigger than the threshold, the corresponding day is predicted to be a bull state.

Neural Network

Instead of feeding the high dimensional data set into the neural network, Principal Component Analysis (PCA) and K-means clustering were employed to summarize the inputs. This would allow the network to converge faster to a robust model

Table 3: The statistics of returns in each state under four models

		Model 1	Model 2	Model 3	Ensemble
State 1	Count	1062	1729	1692	1596
	Mean	2.08%	0.10%	0.12%	0.14%
(Bull)	Std.	0.69%	0.53%	0.65%	0.63%
	t-value	9.79	7.42	7.61	8.59
State 2	Count	1752	1227	1013	1249
	Mean	0.06%	-0.03%	-0.002%	-0.007%
(Static)	Std.	1.14%	1.15%	1.08%	1.09%
	t-value	2.17	-0.79	-0.05	0.22
State 3	Count	458	316	567	427
	Mean	-0.53%	-0.16%	-0.21%	-0.33%
(Bear)	Std.	2.03%	2.91%	2.25%	2.49%
	t-value	-5.56	-0.97	-2.25	-2.76

Table 4: The statistics of ensemble classification by market regime

	Bull Market	Static Market	Bear Market
Average Duration (Days)	10.86 days	6.18 days	7.76 days
Average Daily Return (%)	0.14%	0.007%	-0.33%
Standard Deviation Daily Return (%)	0.63%	1.09%	2.49%
t-value	8.59	0.22	-2.76
Kurtosis of Daily Return	5.24	0.80	2.6
Percentage of days with positive returns	60.65%	49.92%	44.03%
Percentage of days in market state	48.8% (1596/3271)	38.2% (1248/3271)	13.1% (427/3271)

and take advantage of the clustering information provided by unsupervised learning.

i. PCA

Latent factor analysis is commonly used in economics and finance especially in the field of asset pricing. Chamberlain and Rothschild (1983) and Connor and Korajczyk (1986) applied PCA to find the aggregated factors that explains the economic relationship between the underlying features. Given a n by p matrix, X, with n observations in p dimensional space, PCA looks for a small number, say k, of dimensions that capture most of the variation along each direction. Consider approximating X by $XV_kV_k^T$, the projection of X onto the first k principal component directions. If $V_k = [v_1...v_k] \in R^{pk}$ is the matrix whose columns contain the first k principal component directions of X, then

$$\begin{aligned} XV_kV_k^T &= argmin_{rank(A)=k}||XA||_F^2 \\ &= argmin_{rank(A)=k}\sum_{i=1}^{n}\sum_{j=1}^{p}(X_{ij}A_{ij})^2\sum_{i,j}W_{ij}(y_i'^2+y_j'^2-2y_i'y_j') \end{aligned}$$

Since PCA seeks the direction that maximizes variance, standardization was performed prior to PCA to prevent variables with larger range to dominate over variables with smaller range leading to biased results. We used the first 10 principal components to reduce the input dimension down to 10.

ii. K-Means

K means is an unsupervised clustering technique. The algorithm works by optimizing the following objective function.

$$J = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}||x_n - \mu_k||^2$$

The method optimizes J over the cluster centers μ_k and the cluster assignments r_{nk}. One application of k-means clustering is labeling each data point with the cluster labels that the algorithm defines. The hope is that this makes downstream tasks easier by allowing models to gain succinct information on the nature of the inputs. For the neural network we used 5 cluster centers applied to the reduced data set since we have roughly 10 years of data and market cycles tend to be around 2 years. Therefore, the input size into the neural network is 11.

iii.ANN Artificial Neural Networks have become increasingly popular in the last few decades. We train binary classification Neural Networks for each market state instead of bear state. However, typical applications of neural networks in financial prediction require careful regularization considerations since overfitting is a model vulnerability due to the inherent low signal to noise ratio. The architecture of the network consists of 5 hidden linear layers with 200, 400, 1000, 400 , and 100 layers. Furthermore, a mix of leaky relu and tanh activation functions were employed. In order to prevent overfitting, each layer also had a dropout feature added on. Dropout is a popular regularization method for neural networks. During training, each node is crippled (set to zero) with probability p. Then the output of each layer is scaled by a factor of 1/(1-p). Therefore, during testing, the overall output of each layer is taken as is. Due to the random crippling of nodes during training, the network must learn more robust relationships between inputs and response and can not easily overfit. The number of nodes, activation functions, and dropout rates were chosen via cross validation. Similar to the random forest model, the possibilities of being in one specific state are given and a threshold is chosen to decide whether the state of the next day is the target state or not.

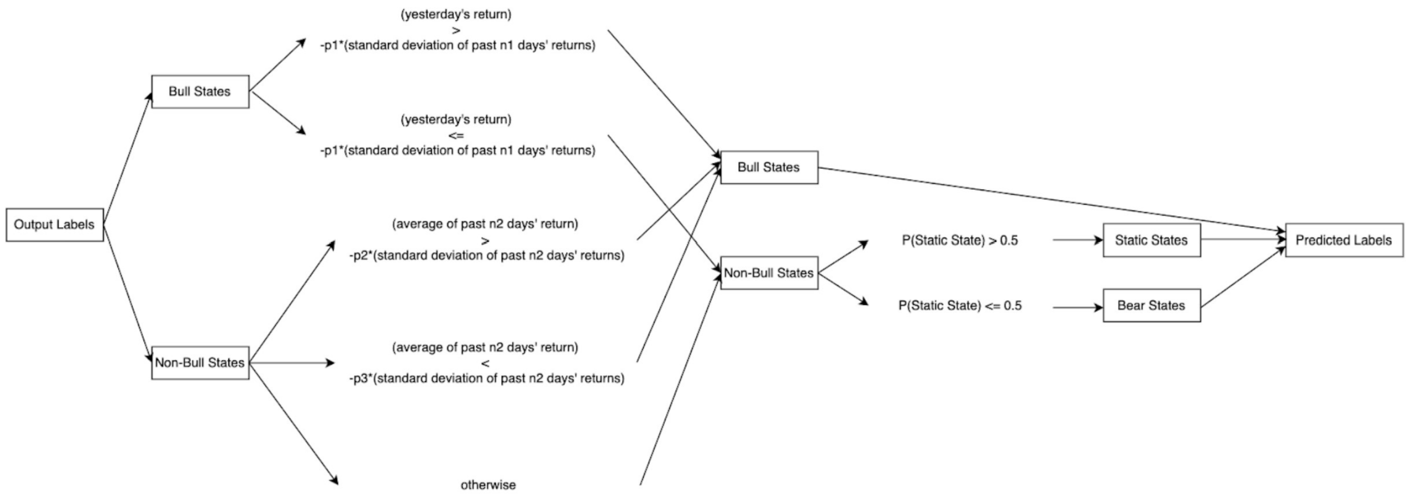

Figure 3: Flow chart of prediction post-processing

Phase 2: Post-Processing

It is noteworthy that it is relatively easy to label and classify market states of history in hindsight. Whereas, the task of prediction of future states is much trickier since the process is adapted. In other words, standing at time t, we can only observe what has happened up to time t and not t+1. Hence, the market information that is incorporated in the features used to make predictions is one day lag. As a result, the predictions given by machine learning models are likely to be lagged as compared to the true market states. This is common in time series prediction models.

In addition, in the real world when the market does not operate efficiently, there is likely a time lag even for traders and investors in realizing a change in market state. For instance, let us imagine that the market has turned from a bear market to a bull market and we notice that prices in the market are rising. Does it mean we are really out of a bear market or is it just noise? It might be too early to tell because we might need to see enough momentum in the upward movement to confirm that the market has indeed exited a bear market. Therefore, the initial recovery of a bear market could also be classified as a bear state instead of a bull state. However, an investor might wish to take a long position in a bear market rather than entering after the market recovers and take advantage of the low prices. Therefore, the results of the predictions generated by the machine learning models might not be appropriate for trading directly. Thus, an additional post-processing step is added to refine our labels for trading.

The method is developed with two main objectives in mind. First, we will want to label a bull state ahead of the actual change from a non-bull to a bull state to capitalize on the low index price. Second, we will wish to exit a bull market prematurely even before the bull market ends. The detailed processing steps are as follows.

1. To exit a bull market early by checking for early signs of bear or static market
    - If the model predicts a bull market tomorrow, compute the standard deviation, "SD1", of daily returns in the past n1 days.
    - If yesterday's return is smaller than the -p1 factor of SD1, the prediction of the bull is rejected and relabelled as non-bull.
2. To enter the market prematurely before the actual bull market, we look for signs of mean reversion by testing if there is enough upward momentum to justify a bull or if the prices have fallen too much by taking the following steps.
    - If the prediction is non-bull tomorrow, compute the mean return, MEAN2, of past n2 days and standard deviation, SD2, of daily returns in the past n2 days
    - If MEAN2 >-p2*SD2 or MEAN2 <=-p3*SD2, the non-bull prediction is rejected and relabeled as bull.

After classifying the market state as bull and non-bull, those that are identified as non-bull will further be divided into static and bull markets. Taking the binary output of static and non-static states from phase 1, threshold probability at 0.5 and label static state for output with probability greater than 0.5 and label bear otherwise. $p1, p2, p3, n1, n2$ are five hyper-parameters that need to be tuned for each model respectively based on the performance of backtesting on the validation period.

The reason why we do not use bear state binary classification models is that the number of bear states is much smaller than bull and static states, leading to severe unbalanced data and very poor model performance. Therefore, only bull and static models are used as these two models are more accurate and provide more reasonable prediction probabilities. A bear market is identified when neither of the other two states is likely to happen.

Results

Model Performance Evaluation

The Random Forest Classifier and Neural Network both predict market states in unique ways. After obtaining the raw prediction from machine learning models, the same post-processing methodology is applied to the outputs from both the random forest model and the neural network model. For the two machine learning models, we focus on the model performance on bull market binary classification. Random forest has accuracy 99% on the training set while only achieve 54% accuracy on the validation set. Neural Network has training accuracy 67.1% and validation accuracy 71.1%. Instead of using 0.5 as the threshold

Table 5: The statistics of random forest prediction results for 2018 to 2021

	Bull Market	Static Market	Bear Market
Average Duration	3.64 days	1.4 days	1.73 days
Average Daily Return	0.16%	0.20%	-0.15%
Standard Deviation Daily Return	1.17%	1.02%	1.72%
Kurtosis of Daily Return	14.8	-0.11	12.8
Percentage of days with positive returns	57.6%	61.9%	52.3%
Percentage of days in market state	67.1% (656/977)	2.15% (21/977)	30.7% (300/977)

Table 6: The statistics of neural network prediction results for 2018 to 2021

	Bull Market	Static Market	Bear Market
Average Duration	3.11 days	N/A	1.73 days
Average Daily Return	0.13%	N/A	-0.05%
Standard Deviation Daily Return	1.19%	N/A	1.63%
Kurtosis of Daily Return	14.5	N/A	14.3
Percentage of days with positive returns	56.8%	N/A	54.7%
Percentage of days in market state	64.3% (628/977)	0% (0/977)	35.7% (349/977)

of deciding whether the prediction is bull market state or non-bull market state, 0.4 is chosen in this paper based on the performance on the validation set. Besides, 5 parameters in the post-processing phase is also tuned on the validation set for both of the machine learning model raw outputs. For the random forest, $p1, p2, p3, n1, n2$ are set to be $0.1, 0.1, 0.8, 30, 60$. And for the neural network, $p1, p2, p3, n1, n3$ are set as $0.1, 0.1, 0.8, 30, 30$.

Table 5 and table 6 show some characteristics of returns in each market state based on two machine learning models and post processing phase. The neural network model predicts no data points in our test period as static. This is consistent with the behavior of the Random Forest model which only predicts 2.15% of data points as static. The overall market trend of the test time period was generally a strong upward trend (both models predict over 60% of days in bull market) except for the time of the COVID-19 crisis (in which case it was a volatile downward trend). The result of having very few static market regimes is consistent with this intuition.

When the random forest model predicted bull, the average daily return was 0.16% with a standard deviation of 1.17%. Similarly, when the neural network predicted bull, the average daily return was 0.13% with a standard deviation of 1.19%. We can compute the standard error of the average daily return given bull prediction by dividing the daily return standard deviation by the square root of the number of predicted bull says. Given this, the random forest average bull return estimator has a standard error of .046%. For the neural network the standard error of the average daily bull return is .047%. Therefore we have strong evidence that there will be a positive future return given a bull prediction from each model. Similarly the random forest's average daily bear return is -.15% with a standard error of .099% while the neural network's average daily bull return is -.05% with a standard error of .087%. One takeaway from these statistics is that in general the bear prediction is weaker across both models.

i. Identification of bear states

Over the period of 2018 to 2021, there are 2 major dips that we will classify as bear markets. We shall measure the performance of our predictors by scoring them on whether they were able to pick up the 2 significant drops as bear markets.

2018 December: There was a drop of about 16% as investors feared the tightening of monetary policy in a slowing economy amidst an intensifying trade war between the U.S. and China.

2020 March: Russells 3000 fell about 34% when the recent COVID-19 pandemic hit.

In both models, the above 2 periods were correctly identified as a bear market. The trading strategy implemented was straightforward: if the predicted state is a bull state, then we invest in the stock market; otherwise, we invest in the money market. The returns, volatility, Sharpe ratio, and Sortino ratio of such trading strategies, based on the post-processed random forest labels and post-processed neural networks labels, were then compared against the simple buy-and-hold strategy.

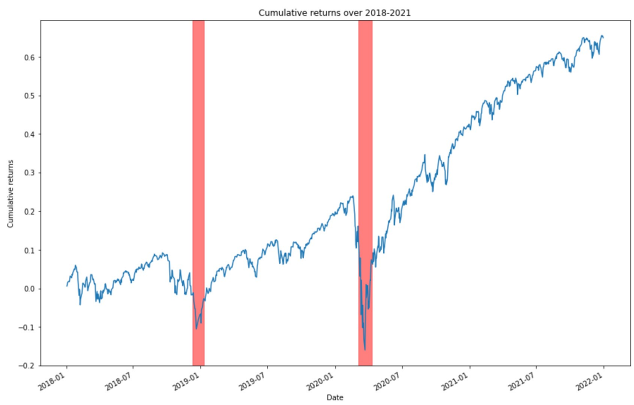

Figure 4: Bear market states during test period

ii. Performance of trading strategy

Both prediction models exhibit strong performance in periods of high volatility. Figures 7 (a) and (b) show the performance of the random forest and neural network model on the validation set (2015-2017). During the one period of high volatility in this time frame, both of the models quickly pick up on the strong bear market state and trade the signal well. During the periods of stable volatility, the strategies trade similar to the market return. Both of the strategies beat the market by much more in the test set time period than the validation set because of this trend. The sharp market crash in March 2020 due to COVID-19 was

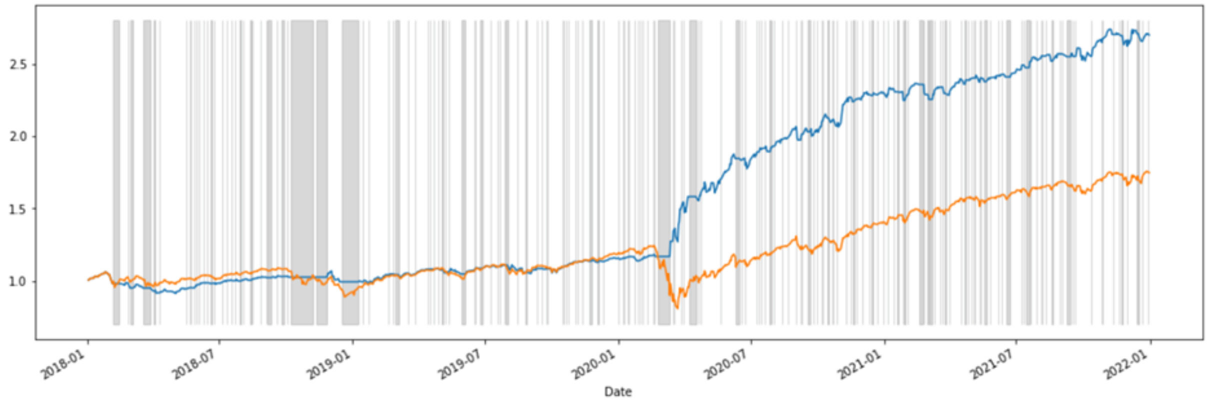

Figure 5: Random forest classification of bear states

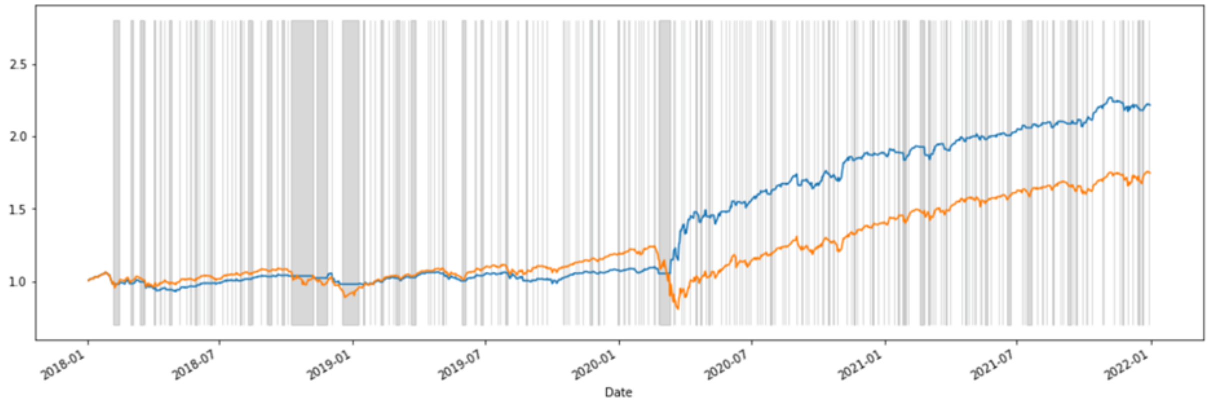

Figure 6: Neural network classification of bear states

(a) Random forest on validation set

(b) Neural network on validation set

(c) Random forest on test set

(d) Neural network on test set

Figure 7: Backtesting results on validation set and test set

favorable for both strategies leading to a much better return in the test period (2018-2021) shown in figures 7 (c) and (d). This means the prediction model captures such big drops very well. This can be further confirmed by figure 5 and figure 6 which show the bear markets predicted by two different models. At the beginning of the COVID-19 period when Russel 3000 had its biggest drop from 2018 to 2021, the prediction well captures the down trend showed by the gray areas. A unique observation of the random forest strategy is a lagged response to high volatility. Figure 7 shows the returns of the random forest model for the validation set and test set, respectively. Each time frame had one big downswing in the market, and the random forest strategy has a very brief period of constant return before detecting the sharp market downturn and trading against it. This idea is consistent with the validation accuracy seen for both models - where the random forest has much lower accuracy than the neural network. The neural network strategy is more robust to predicting market state quickly while the random forest strategy needs more data before detecting the change.

Table 7: Backtesting performance on validation set

	Random Forest	Neural Network	Market
Cumulative Return	0.3292	0.2689	0.2576
Annualized Return	0.1100	0.0899	0.0861
Annualized Volatility	0.0896	0.0834	0.1251
Annualized Sharpe	1.0254	0.8605	0.6175
Annualized Sortino	1.2163	0.9984	0.7937

Table 8: Backtesting performance on test set

	Random Forest	Neural Network	Market
Cumulative Return	1.0623	0.8622	0.5580
Annualized Return	0.2737	0.2222	0.1438
Annualized Sharpe	1.5170	1.1809	0.6187
Annualized Volatility	0.1510	0.1503	0.2181
Annualized Sortino	1.7975	1.3507	0.6755

Conclusion

In this paper, we did an in depth exploration on how market states can be first determined and then forecast using carefully engineered features and tuned machine learning models. In both classification and prediction steps, we obtained satisfying results that beat the plain buy-and-hold strategy for the period from 2018 to 2021.

A further extension to our report is to explore other learners that can be added to the ensemble to produce a stronger learner. For instance, Pier and Tomaso (2018) identified market states by minimizing the Mahalanobis distance of reference sparse precision matrix. In our dimension reduction step, an alternative to linear dimension reduction method, PCA, one could explore nonlinear dimension reduction techniques such as an autocoder which finds a low dimensional mapping of the input to itself. In addition to neural networks, one could explore using recurrent neural networks such as long short-term memory to capture the temporal properties of returns. Instead of random forests which train each tree independently, one could try gradient boosting where each tree is trained to correct the errors made by the previous tree. One possible extension could be to feed these predictions as features in asset pricing models to predict stock returns.

Although it is a difficult problem to predict exact market movements of the index, our report shows that being able to predict the state of the market can be an invaluable insight to an investor to subsequently develop strategies that can outperform the market.

References

Appel, Gerald and W Frederick Hitschler (1980). *Stock market trading systems*. Irwin Professional Pub.

Breiman, Leo (2001). "Random Forests." In: *Machine Learning* 45, pp. 5–32.

Connor G., Korajczyk R.A. (1986). "Performance measurement with the arbitrage pricing theory: a new framework for analysis." In: *J. Financ. Econ.* 15, 373–394.

Fabozzi, Frank J and Jack Clark Francis (1977). "Stability tests for alphas and betas over bull and bear market conditions." In: *The Journal of finance*. 32, 1093–1099.

Kole, Erik and Dick Dijk (2017). "How to Identify and Forecast Bull and Bear Markets?" In: *Journal of applied econometrics*. 32, 120–139.

Lambert, Donald R (1983). "Commodity channel index: Tool for trading cyclic trends." In: *Technical Analysis of Stocks & Commodities* 1, p. 47.

Liliana Gonzalez John G. Powell, Jing Shi Antony Wilson (2005). "Two centuries of bull and bear market cycles." In: *International Review of Economics and Finance* 14, 469–486.

Procacci, Pier Francesco and Tomaso Aste (2018). "Forecasting Market States." In: SSRN Electronic Journal.

Wilder, J Welles (1978). *New concepts in technical trading systems*. Trend Research.

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.

Appendix

Table of Features

Category	Feature Name	Description
Technical Signals	daily_return	Daily return of Russell 3000 of the last trading day
	r5, r20, r60, r120, r240	Return of Russell 3000 during the past 5, 20, 60, 120, 240 trading days
	vol5, vol20, vol60, vol120, vol240	Volatility of Russell 3000 during the past 5, 20, 60, 120, 240 trading days
	RSI5, RSI20, RSI60, RSI120, RSI240	Relative Strength Index (Wilder 1978) over the past 5, 20, 60, 120 and 240 trading days.
	DX60, DX120, DX240	Directional Movement (Wilder 1978) over the past 60, 120 and 240 trading days.
	CCI5, CCI20, CCI60, CCI120, CCI240	Commodity Channel Index (Lambert 1983) over the past 5, 20, 60, 120 and 240 trading days.
	corr5, corr20, corr60, corr120, corr240	Pearson product momentum correlation coefficient between Russell 3000 daily high and low using time window of 5, 20, 60, 120 and 240 trading days.
	MACD	Moving Average Convergence/Divergence (Appel and Hitschler 1980)
	RUAt0, RUAt1, RUAt2, RUAt3	Daily return of Russell 3000 of current day and 1, 2, and 3 trading days ago.
	EMA10, EMA20, EMA50, EMA200	Exponentially weighted moving average of daily returns of Russell 3000 for the past 10, 20, 50 and 200 trading days.
	R_Vol	Trading volume of iShares Russell 3000 ETF
Exchange Rates	USDJPY=X_Price, GBPUSD=X_Price, USDCAD=X_Price, USDCNY=X_Price, EURUSD=X_Price	Exchange rates between USD/JPY, GBP/USD, USD/CAD, USD/CNY and EUR/USD
Stock Indices	^HSI_Price, ^IXIC_Price, ^N225_Price, ^VIX_Price, SPY_Price	Index level of Hang Seng Index, Nasdaq Composite Index, Nikkei 225 Index, VIX Index and S&P 500 ETF
Cyclical Stock Prices	AAPL_Price, MSFT_Price, XOM_Price, GE_Price, JNJ_Price, WFC_Price, AMZN_Price, JPM_Price, DIS_Price	Stock prices of Apple, Microsoft, Exxon, General Electric, Johnson & Johnson, Wells Fargo, Amazon, JPMorgan Chase and Walt Disney
Commodities	Oil	Oil price
	Gld	Gold price
Interest Rates	DGS1MO, DGS3MO, DGS6MO, DGS1, DGS5, DGS10	Market yield on U.S. treasury securities at 1-month, 3-month, 6-month, 5-year and 10-year constant maturity
	CTB1MO, CTB3MO, CTB6MO, CTB1, CTB5, CTB10	Daily relative difference of corresponding DGS
	AAA, DBAA	Moody's seasoned Aaa and Baa corporate bond yield
	TE1, TE2, TE3, TE5, TE6	Term spread between DGS10/DGS1MO, DGS10/DGS3MO, DGS10/DGS6MO, DGS3MO/DGS1MO, DGS6MO/DGS1MO
	DE1, DE2, DE3, DE4, DE5	Default spread between DBAA and AAA, DGS10, DGS6MO, DGS3MO and DGS1MO