The Boy Who Cried Bear: Multi-Model Approach for Predicting Market State

Abstract: Enormous amounts of wealth are changing hands at every moment in the financial markets. Consequently, an accurate forecast of the future is a valuable piece of information for any investor. While one might generally believe that one cannot accurately predict the price of an individual security, being able to predict the "state" of the market would unambiguously give investors an advantage over others. The idea of predicting the market state has been widely accepted and discussed by academics and professionals. In this paper we review the popular definitions of Bull, Bear and Neutral market states and create a method to predict the "state" of the market by by combining an unsupervised learning method (HMM) with a supervised learning method (XGboost). We then deploy this hybrid model with a basic trading strategy on the Russell 3000 index and analyze its performance. Our aim is not to find the most profitable trading strategy, but to demonstrate its potential to be used as part of a risk management strategy.

Introduction

With the benefit of hindsight, it is very easy to classify Bull and Bear markets. Economists seem to agree on this simple recipe: find all the drawdowns larger than 20% and classify the peak to trough as Bear markets. The rest can be classified as a Bull market. A trading strategy that goes long in the Bull market and short in the Bear market - with this benefit of hindsight – would do very well. We call this strategy the "Oracle" strategy and view it as an upper bound of how well we can expect a Bull/Bear market prediction model to perform. A model should not call a market a bear too often or too seldom, compared to the Oracle, for it to be useful in practice.

Hidden Markov models (HMMs) are a natural choice to predict the regime (bear, bull or neutral) of the market. HMMs are unsupervised models whose main parameter, the number of regimes, can be set to three. These models can be fit to Russell 3000 returns and other macro variables. The three states can then be interpreted as Bull, Bear or Neutral. We find that HMM based strategies that go long in the Bull state, short in the Bear state and stay out of the market in the neutral state, perform better than long only, out of sample. However, the proportion of time that the HMM classifies the state as Bull in the 2018-2021 period is 70% while the "Oracle" classifies the market as a Bull 90% of the time. This model is unusually pessimistic and - like the boy who cried bear - stays out of the market too often.

There is no guarantee that the states of an HMM model reflect what most economists, with the benefit of hindsight, would refer to as a Bull or a Bear market. This potential disagreement motivated us to explore XGBoost, a popular supervised model. We train the model on the "Oracle" strategy. This model performs better than the Buy and Hold out of sample, but classified the market as a Bull 97% of the time. XGBoost trained on technical data is unusually optimistic.

Finally, we consolidate the two models using a simple voting strategy and find a happy medium between the two which is more closely aligned with the "Oracle".

1. Review of existing research

Financial media and sell-side research firms (S&P Global, Yardeni Research etc.) often use heuristic methods to classify market regimes. Traditionally, S&P500 performance has been used to study regimes instead of Russell 3000. For example, one simple heuristic is to classify the market as Bull when the S&P 500 Index cumulative returns are $> 20\%$ and Bear when they are $< 20\%$. Some firms even classify a Neutral market as periods when the S&P 500 Index traded within the range of -20% to $+5\%$[1].

Academic literature generally cite two broad methods for market regime classification - rule-based

[1]Source: https://www.raymondjames.com/neunuebelbarrantes/pdfs/history-of-market-corrections.pdf, https://www.yardeni.com/pub/sp500corrbeartables.pdf, https://russellinvestments.com/us/blog/bulls-vs-bears

methods and Hidden Markov (HMM)/Markov Switching (MS) models. Bry and Boschan (1971) algorithm constitutes a rule-based method of picking local extrema in economic activity data, based on a chosen time window. An important adaptation of this algorithm to financial data was a method presented by Pagan and Sossounov (PS)[1] and Lunde and Timmermann (LT)[2].

HMM's have been widely used in academia to analyze the behavior of the stock market and apply them to build quantitative trading strategies. Hassan and Nath (2005)[3] first used the HMM to study the "law" of stock market changes. Nguyen and Nguyet (2015)[4] used the HMM to predict the "hidden" state of observed data on the market and selected stocks based on the predicted state to form an optimal portfolio strategy. Similarly, Wang et al. (2020)[5] used the HMM to analyze regimes in S&P500 and built a regime-detection investing strategy that rotates between various factor models. They have been applied to Cryptocurrencies as well — Constandina et al.(2019)[6] used HMM's for predicting Cryptocurrency returns in the presence of state dynamics. A recent Two Sigma study used Gaussian Mixture models to analyze "hidden" market states and provide intuition behind them (ex-post analysis)[2].

Extreme Gradient Boosting (XGBoost) is another popular algorithm that has been widely used for the purpose of stock price prediction. Basak et al. (2019)[7] used decision trees and gradient boosting to analyze the direction of the stock market. Similarly, Yang et al. (2021)[8] used XGBoost and LightGBM to analyze and forecast price based on Jane Street's high frequency tick data. However, we have not yet found any research on XGBoost applied specifically to a regime detection problem.

2. Benchmark ("Oracle Strategy")

We first build the aforementioned benchmark "Oracle" model to test our algorithms against — this model represents the market regime classification over time using data from different public sources. We test a few of the widely-accepted rule-based methods based on the works of Pagan and Sossounov (2003) and Lunde and Timmermann (2004) along with a labelling strategy based on data from economists and investment research. Note that these do not classify a Neutral state.

We observe that there is a generally-accepted definition of Bull and Bear market states. They are the period from peak to trough or trough to peak that an index goes up or down a specific percentage. In Figure 1 below, we show the Russell 3000 performance overlaid with the Bull and Bear markets of the S&P 500 as categorized by the traditional economists cited above. The green shaded regions indicate a Bull market, and the red regions indicate a Bear market in the plot shown below.

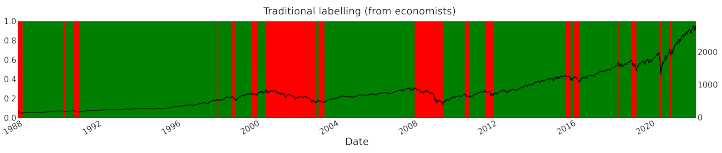


Figure 1. Regime classification by traditional economists

Following this methodology, we first classify the Russell 3000 index into the two states of Bull and Bear markets. We choose a peak-to-trough (or vice versa) percentage of 15% as our classifying condition, enabling any peak to trough (or vice versa) of 15% or greater to be defined as a Bear (Bull) state in retrospect. Intuitively, a Neutral state would seem to be the transitory period between the two well-defined states. More specifically, the week (5 market days) on either side of the state switch could be considered a Neutral state. We looked back across the Russell 3000 since its inception in 1987 classify the market using these rules ex-post. The "Oracle" strategy benchmark will be considered the upper bound on how well our model can predict the state of the market. The "Oracle" knows the "Oracle" market state and is fully invested in the market in a Bull state, in 3-month US Treasury Bill in a Neutral state, and short the market in a Bear state. The resulting "Oracle" regime map since the Russell 3000 inception and the Log Portfolio wealth of the "Oracle" versus Buy and Hold are shown below in Figure 2. Note that green denotes a Bull market period, red indicates a Bear market period and yellow denotes a "Neutral" market period.

3. Data

The first step in setting up our regime detection model strategy is to create a feature pipeline that

[2] https://www.twosigma.com/articles/a-machine-learning-approach-to-regime-modeling/

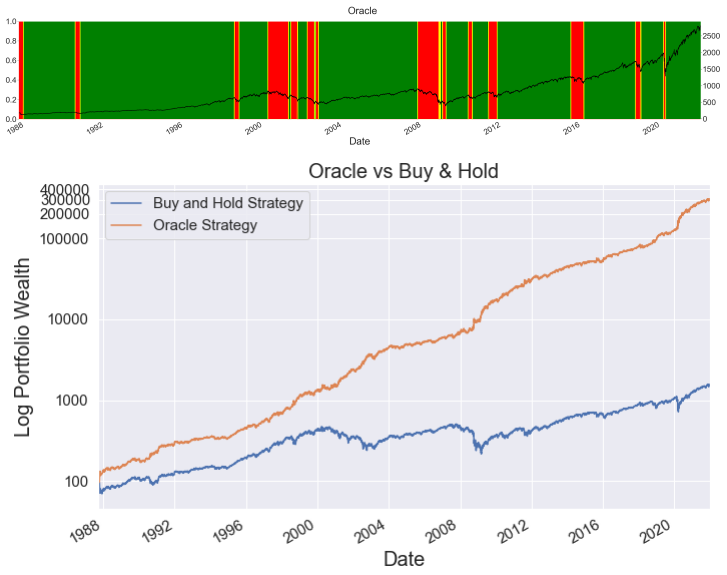


Figure 2. Oracle Regime Map and Performance

differentiates between the different market regimes and trains our model with data. The Russell 3000 index data was fetched from Yahoo Finance and the 3-month US Treasury Bill rate was sourced from FRED[3].

Exhibit 1 summarizes the features used for each model, with their historical data availability, data source, transformations employed. We decided to train our models using both technical indicators based on Russell 3000 historical price data and macroeconomic variables. All of the features have daily reporting frequency.

The correlation heatmaps in Figure 3 demonstrate that the macroeconomic features have very low or slightly negative correlations. The correlations between the technical features are high; this does not affect the tree-based models like XGBoost, however, since decision trees are, by nature, immune and robust to multicollinearity and correlation among independent variables.

3.1 Economic intuition for the features

Our hypothesis for the HMM is that an optimal set of key macroeconomic features should be able to explain sharp movements in the markets. We decide against using index returns since our implementation of HMM is unsupervised, akin to clustering, price changes that may not actually be due to an underlying regime change would cause the model to over fit the data. The HMM model uses macroeconomic features such as Oil price change [9], VIX[10] and Equity Put/Call Ratio to predict market states. We added a new predictive feature for identifying market turning points, a Trend Followers Index from Societe

[3] https://fred.stlouisfed.org/series/DTB3

Exhibit. 1. Feature summary

Category	Feature	Type	Source	History	Transformation	XGBoost	HMM
Technical	Daily Returns	Raw	Yahoo Finance	Sep. 1987–	—	X	
	Daily Volatility	Derived			—	X	
	20-day EWMA of Daily Returns	Derived			Decay span=20	X	
	50-day EWMA of Daily Returns	Derived			Decay span=50	X	
	Equity Put/Call ratio	Raw	CBOE, Alphalerts	Oct. 2003–	—	X	X
	Equity Put/Call ratio 1 day lag	Derived			1-day lag	X	
Macro	Oil WTI Spot Price	Raw	US EIA*	Jan. 1986–	Change in %		X
	VIX	Raw	Yahoo Finance	Jan. 1990–	Change in %		X
	SG Trend Index Returns	Raw	Societe Generale**	Dec. 1999–	—		X

* US Energy Information Administration

** https://wholesale.banking.societegenerale.com/en/prime-services-indices/

(a) Technical features **(b)** Macroeconomic features

Figure 3. Feature Correlation heatmaps

Generale designed to track the largest trend-following CTAs in the managed futures space.

We choose the training features for XGBoost based on their relevance to the label being predicted. This is because they are derived from the same price data that the "Oracle" label is derived from, except the "Oracle" label is determined in hindsight. The exception to this are Put/Call ratios, which are indicators of investor sentiment and are associated with short-term stock market returns. Including both the t and $t-1$ Put/Call ratios can give the algorithm insight into the most recent change in sentiment as well.

3.2 Feature Engineering

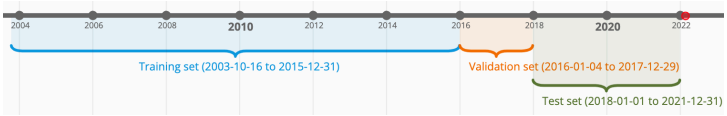
First, we sanitize our dataset to handle missing data:

- We remove null values from the Russell 3000 index, sourced from Yahoo Finance.
- For the 3-Month Treasury Bill rates, we found missing data for the federal holidays — we forward-fill missing data with last available data.
- We apply similar treatment to the VIX and WTI Oil Price data.

The feature transformations for the data are summarized in Exhibit 1. Most of the features have a long history, but the market Put/Call ratio only has available data from October 2003. Therefore, we limit our data available for training, validation and testing to start from then. Additionally, we shift labels by one day to avoid look-ahead bias in our predictions. We do not employ normalization techniques because both of our models are built to properly handle difference in scaling among features.

3.3 Train, validation, test split

We divide our dataset (Oct. 2003 — Dec. 2021) into training, validation and test sets. The training dataset includes roughly twelve years of data, the validation set includes two and the test set includes the final three years of data (Jan. 2018 — Dec. 2021).



4. Models

We decide to build two models, one supervised and one unsupervised, for regime prediction. A supervised model will better conform to the traditional definitions of Bull and Bear, while an unsupervised model will more organically identify market states with little outside influence, perhaps detecting what could be going on "beneath the surface". The supervised model we use is an XGBoost model with technical indicators as features, trained on the "Oracle" labels derived from market performance, as described earlier. The unsupervised model is based on the HMM and uses macroeconomic features and Put/Call ratios.

We opt not to use a technical rule-based model (like PS and LT referenced in Section 1) in our ensemble. Even though rule-based algorithms are easy to understand and implement, they suffer from delayed state identification due to "information lag" (lag between the moment of regime switch and its detection) — which is a major practical disadvantage. This information lag even hurts the performance of less flexible supervised models such as a Multinomial

Logistic Regression and a Support Vector Classifier. HMM/MS models are less intuitive, but produce immediate results.[11]

4.1 XGBoost

XGBoost is a tree-based ensemble method which relies upon the Gradient Descent algorithm to minimize residual errors as it builds subsequent models. It is more suitable for small and medium-sized data sets than Neural Networks, which would be the go-to choice for supervised learning in large, unstructured data. A boosting algorithm takes the predictors, considered weak learners, and converts them into strong learners. Subsequent trees give extra weight to points incorrectly predicted by earlier predictors. The boosting algorithm will run for M boosting iterations and at termination it will take a weighted vote from the predictors.

The steps for a boosting iteration are as follows:

1. Let F_m predict a labeled variable $y(F(x) = \hat{y})$.
2. F_m will be associated with a residual $f_m(x) = y - F_m$ that is fit to the residuals in the previous step.
3. Combine F_m and $f_m(x)$ to give $F_{m+1}(F_{m+1} = F_m + f_m(x))$.

F_{m+1} is the boosted version of F_m and will have a lower loss function value than the previous model. New trees are constructed in this fashion for M iterations.

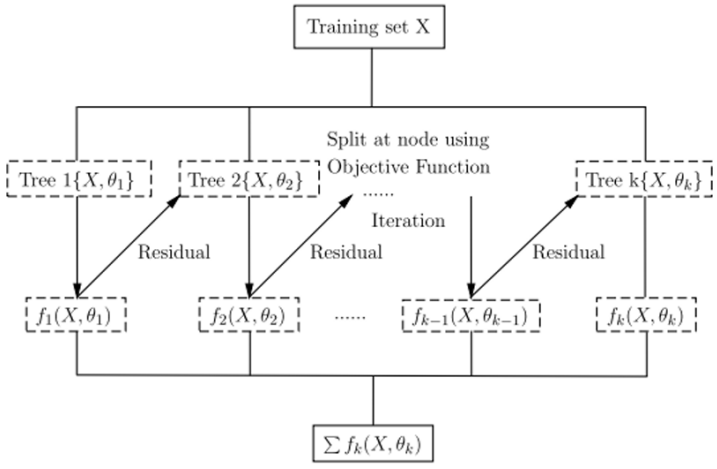


Figure 4. Extreme gradient boosting (XGB)

XGBoost self-regularizes by introducing L1 and/or L2 penalties and "tree pruning" to prevent overfitting. Three important parameter choices found via random search using the the train and validation sets were the learning rate (.001), n-estimators (15) and max depth (10). These low parameter values also help to prevent overfitting. We conclude that technical indicators worked best; the intuitive reasoning for this is already stated. XGBoost directly evaluates feature importance, shown below in Figure 5.

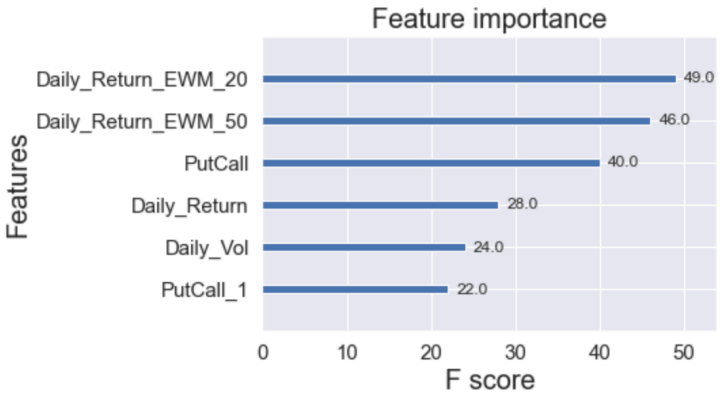


Figure 5. XGBoost Feature Importance

4.2 Hidden Markov Model (HMM)

Simply put, a HMM is an unsupervised model that can capture the "hidden" states of underlying observation data. In our case, we aim to use HMM in an unsupervised fashion to achieve a result akin to clustering. The states are represented as nodes in the

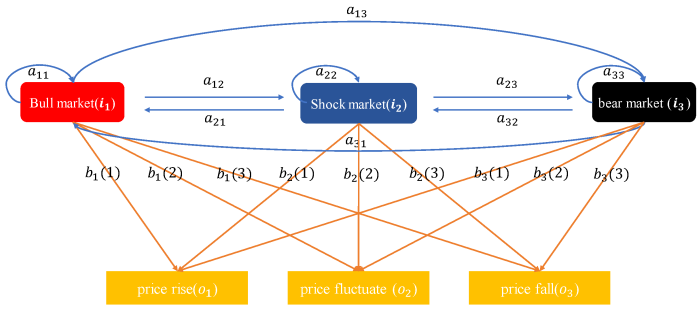


Figure 6. HMM: Shock refers to the neutral state

Figure 6, and the transitions, with their probabilities, as edges. The transitions are probabilities: the values of arcs leaving a given state must sum to 1.

There are three main steps to train a HMM[4] –

1. Estimate the probability of occurrence for the set of observations with a recursive forward algorithm[12] that computes the forward probabilities of ending in a state given the prior observation sequence.

[4] https://web.stanford.edu/~jurafsky/slp3/A.pdf

2. Determine the most optimal sequence of "hidden" states for the HMM with the Viterbi algorithm[13] that recursively computes the most probable path through a sequence of states.
3. Find the optimal set of parameters with the Expectation Maximization (EM) algorithm to calibrate, or to move from initial estimates of the parameters to more probable estimates using maximum likelihood. Specifically, we use the Baum-Welch algorithm[14], a special case of EM.

We choose to model the probability distributions of the HMM observations as Gaussian – we make use of the hmmlearn open source library[5] in Python for our model. There are two primary parameters for the model:

- Number of hidden states: set to 3.
- Covariance type: set to "full" (assumes features may be correlated) instead of the default "diagonal" (assumes features are independent). The full matrix correctly captures cross-correlation between features.

We then train the model with the training set comprised of macroeconomic features and check its performance on the validation and the test sets.

5. Model Evaluation Criteria

5.1 Classification Accuracy Metrics

Our first model evaluation criteria is forecast accuracy; we compare the proportion of regime states in each model to the Oracle. Overall accuracy is not preferred because the Bull state is much more frequent than the other two. Instead, a true skill statistic known as Kuiper's Score (KS)[6] is a better metric for our analysis, as shown below. Kuiper's Score for regime forecasting rewards models for correctly predicting bull markets and penalizes for incorrectly "crying Bear", also known as false alarms. TP, FP, TN, and FN stand for number of true positives, false positives, true negatives, and false negatives, respectively. This method helps to penalize models which "cry bear" falsely.

$$KS = \frac{Bull_{TP}}{Bull_{TP} + Bull_{FN}} - \frac{Bear_{FP}}{Bear_{FP} + Bear_{TP}}$$

[5] https://hmmlearn.readthedocs.io/en/latest/

[6] https://cawcr.gov.au/projects/verification/#Methods_for_multi-category_forecasts

5.2 Profitability, Risk, and Feasibility

We compare the different models based on their performance in a given investment strategy. We first outline how each component and portfolio returns are calculated. Let us denote the starting portfolio wealth as w_0 and the market state on a particular day t as $state_t$ taking values as –

$$state_t = \begin{cases} 1 & \text{if Bull,} \\ 0 & \text{if Neutral,} \\ -1 & \text{if Bear.} \end{cases}$$

We use adjusted closing prices for equity returns to account for any corporate actions. 3-Month Treasury Bill rates are annualized rates of return denoted in percentages with an Actual/360 day count convention. Our regime-switching strategy portfolio goes long the Russell 3000 if the market state is Bull, goes short the Russell 3000 if Bear and is out of the market, invested in 3-Month Treasury Bills (3M T-Bills) if the state is Neutral. The "Buy-and-Hold" portfolio simply mimics an investor who is long the Russell 3000 index for the full period.

$$\text{Equity Returns: } r_t^{eq} = \frac{p_t}{p_{t-1}} - 1$$

$$\text{3M T-Bill Returns: } r_t^{tbill} = \frac{r_t^{tbill_a nn}}{360 \times 100}$$

$$\text{Portfolio Returns: } r_t^{port} = \log\Big[1 + r_t^{tbill} - \Big(|state_t| \times r_t^{tbill}\Big) + \Big(state_t \times r_t^{eq}\Big)\Big]$$

$$\text{Terminal wealth: } w^{port} = w_0 \exp\Big(\sum_{t=1}^{T} r_t^{port}\Big)$$

We use a diverse set of performance and risk metrics to provide a holistic view of trading strategy performance comparison across models. First we include annualized and cumulative returns, annualized risk, Sharpe Ratio[7], and Sortino ratio. Next, we include a set of unique metrics to quantify how well the regime switching strategy handles market downturns. We select Maximum Drawdown, gain to pain ratio (ratio of net returns to the losses incurred in getting those returns) and Ulcer index.[8].

[7] https://alo.mit.edu/wp-content/uploads/2017/06/The-Statistics-of-Sharpe-Ratios.pdf

[8] https://www.investopedia.com/terms/u/ulcerindex.asp

The performance metrics are calculated using an open source Python library, QuantStats[9], that performs portfolio profiling with in-depth analytics and risk metrics.

Note that, even though we assume frictionless markets (no transaction and short selling costs), number of trades (back and forth switching between asset classes) can be assumed to be directly proportional to transaction costs. It is presented here as a naive heuristic proxy to gauge the impact of costs.

6. Results and Model Evaluation

We use the criteria described in the last section to first evaluate each of our models in the out-of-sample validation set. Using a validation set avoids the common pitfall of data-mining the test set. If we choose a model by evaluating performance on the test set ("data snooping" or "data peeking"), the test set is no longer truly out-of-sample[10].

6.1 Out-of-sample validation performance

Exhibit. 2. Proportion of Time in a Market State: Validation Set

State	Oracle	XGB	HMM
Bull	93.4%	98.0%	57.8%
Neutral	2.0%	0.0%	33.3%
Bear	4.6%	2.0%	8.9%
Kuiper's Score	1	.55	.39

After training our two models, we predict on the out-of-sample validation set. Neither model, on its own, is able to beat the "Buy-and-Hold" strategy return, but both strategies minimize Maximum Drawdown considerably. The HMM achieves a superior Sharpe Ratio. Additionally, the HMM appears to be much more active than the "Oracle" strategy, picking up many signals outside of our "Oracle" state. However, the strategy is a bit noisy and one does not truly understand why it picks the state it does. The HMM also predicts far more Neutral states than the "Oracle" label. We can indeed understand the XGBoost model and observe its feature importance in the table below. Its prediction appears to be close to our "Oracle" state, but investment managers may wish to have more insight during a Bull market than XGBoost gives, which motivates our decision to combine models.

6.2 Voting Algorithm and Combination Reasoning

As discussed earlier, the two models have been trained on different datasets due to their different underlying natures. Combining the models should produce a forecast with a better outcome and less noise. We decide upon an simple, intuitive, voting-based logical gate combination. If both models agree on the state, we keep that state. If one model predicts Neutral while the other predicts Bull (or Bear), the result is Bull (or Bear). If one model predicts Bear while the other predicts Bull, the result is Neutral.

6.3 Composite Model Validation Performance

The relevant metrics and ratios in the strategy risk/return profile improve dramatically when the two models' predictions are combined through a custom logical gate (Exhibit 3). In addition, the regime proportions and accuracy scores (shown in Figure 7 and Exhibit 4), as well as trading performance (Figure 8) become more closely aligned with the "Oracle" strategy.

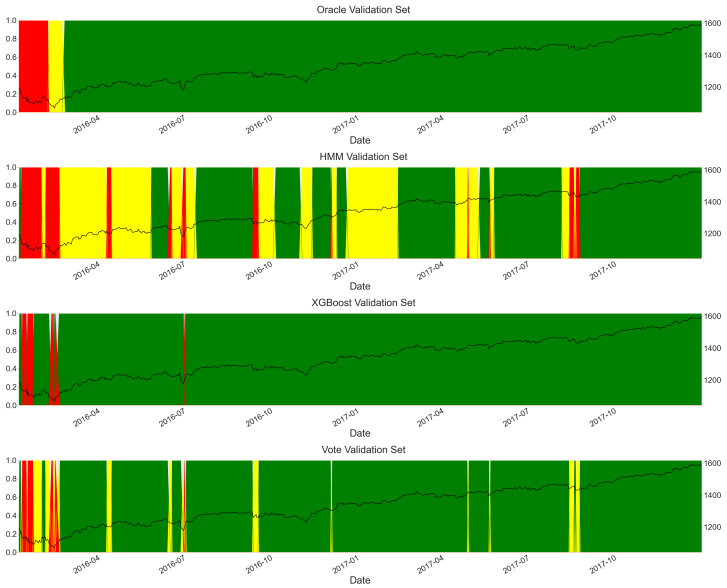


Figure 7. Regime Plots on the validation set

6.4 Out-of-sample Test Performance

We first observe the day-ahead state forecasts of our three forecasting algorithms compared to the "Oracle" state in our test set.

Both the XGBoost model and the HMM comfortably outperform "Buy-and-Hold" here, as there was

[9] https://github.com/ranaroussi/quantstats

[10] Source: https://machinelearningmastery.com/difference-test-validation-datasets/

Exhibit. 3. Performance metrics: Validation set

	Oracle	Buy & Hold	XGB	HMM	Vote
Annualized Return	14.76%	10.51%	9.84%	10.31%	14.13%
Cumulative Return	48.59%	33.09%	30.98%	32.61%	46.28%
Annualized Risk	10.29%	10.76%	10.77%	8.17%	9.21%
Sharpe Ratio	1.95	1.36	1.28	1.74	2.08
Maximum Drawdown	-5.98%	-10.25%	-7.66%	-4.6%	-5.37%
Sortino Ratio	2.98	1.96	1.83	2.82	3.23
Gain/Pain Ratio	0.45	0.3	0.28	0.52	0.5
Ulcer Index	0.01	0.02	0.02	0.02	0.01
Number of Trades	3	1	11	35	33

Exhibit. 4. Proportion of Time in a Market State: Validation Set compared to Vote Strategy

State	Oracle	XGB	HMM	Vote
Bull	93.4%	98.0%	57.8%	91.0%
Neutral	2.0%	0.0%	33.3%	2.0%
Bear	4.6%	2.0%	8.9%	7.0%
Kuiper's Score	1	.55	.39	.58

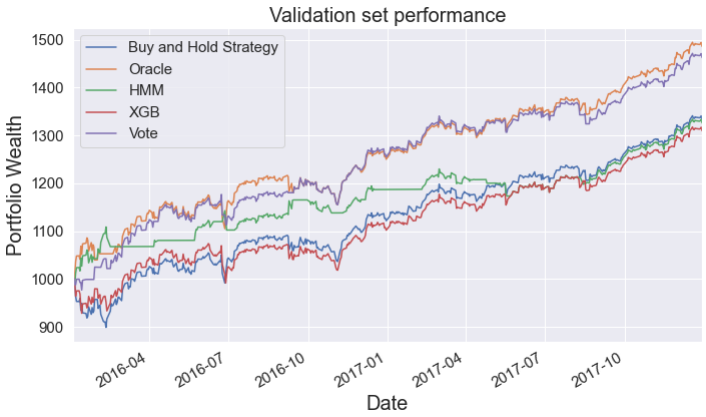


Figure 8. Validation set portfolio performance chart

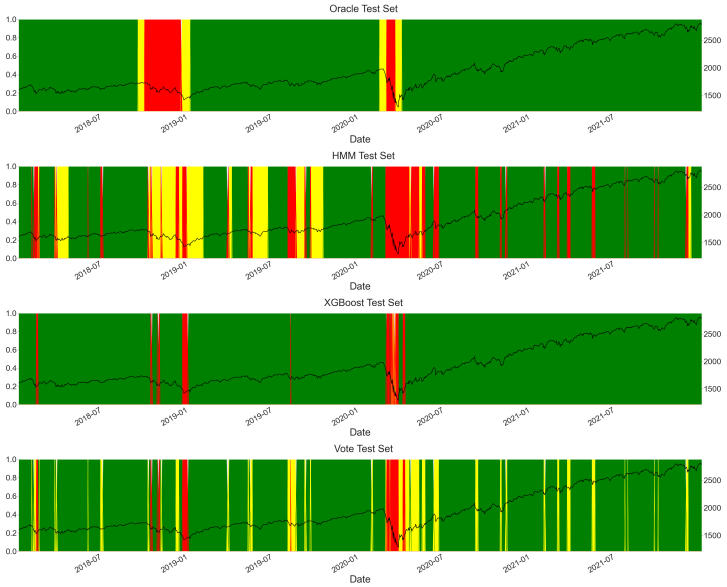


Figure 9. Regime Plots on the test set

Exhibit. 5. Proportion of Time in a Market State: Test Set

State	Oracle	XGB	HMM	Vote
Bull	89.1%	96.5%	70.3%	85.0%
Neutral	6.8%	0.3%	15.1%	11.5%
Bear	4.1%	3.2%	14.6%	3.5%
Kuiper's Score	1	.36	.17	.42

more volatility, and thus more trading opportunities from regime switching, in this period. The regime plots (Figure 9) and table of proportions of time in each state (Exhibit 5) illustrate that the composite voting strategy more closely captures the "Oracle" state than each individual strategy. In addition, the composite voting strategy has a higher Kuiper's Score than the other algorithms, on both validation and test sets. (Exhibit 5)

As shown in Figure 10 and Exhibit 6, the composite voting strategy exhibits much more consistent returns than either strategy on its own; the validation and test set results closely align. The combined model has lower return than the HMM, and only slightly higher return than the XGBoost, but it has better risk/return metrics such as Sharpe Ratio, Ulcer Index, Gain/Pain Ratio and Sortino Ratio. The Maximum Drawdown is 13.3% versus the 24% and 20% drawdowns of the standalone strategies.

7. Conclusion

Our research adds to the important, but still relatively underdeveloped area of financial market regime identification methodology. We document the robustness of each model's capability to "time" the market through the lens of a simple trading strategy — both individual component models showed strong predic-

Exhibit. 6. Performance metrics: Test set

	Oracle	Buy & Hold	XGB	HMM	Vote
Annualized Return	26.23%	10.12%	19.39%	20.56%	19.44%
Cumulative Return	285.19%	73.7%	178.98%	195.19%	179.65%
Annualized Risk	19.74%	21.68%	20.8%	20.23%	18.27%
Sharpe Ratio	1.83	0.74	1.35	1.45	1.51
Maximum Drawdown	-9.97%	-35.09%	-19.88%	-24.21%	-13.27%
Sortino Ratio	2.93	1.01	2.14	2.22	2.35
Gain/Pain Ratio	0.46	0.17	0.32	0.41	0.4
Ulcer Index	0.03	0.07	0.06	0.06	0.03
Number of Trades	9	1	27	77	86

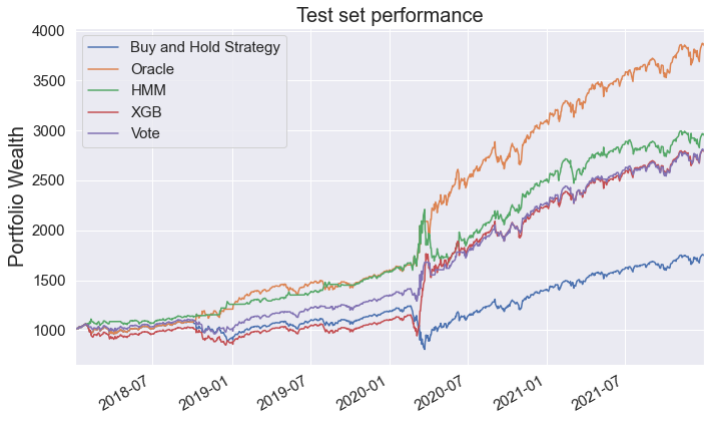


Figure 10. Test set portfolio performance chart

tive power at identifying actual bear markets. However, they disagreed largely in predicting market rebounds and Bull market corrections. This skewed behaviour in the model predictions motivated us to aggregate the predictive power of both models into a composite indicator.

We observe that the voting strategy's profitability, lower risk measures (volatility and drawdowns) and forecasting accuracy persists through both the more sleepy validation set and the more volatile test set. This persistence gives us confidence that the composite algorithm is providing a useful indicator of the future state of the market.

Most encouragingly, in the test set, we successfully eliminated every false Bear market post the Covid-crash from the HMM model by grounding it with the XGBoost supervised algorithm in our Vote model. The "Oracle" classified the state as a Bear market 4.1% of days, while the Voting model predicted 3.5%. The neutral state in the Vote model was 11.5% versus the "Oracle" proportion of 6.8%, but as seen in the regime plots the Vote model was capturing draw-downs during the Bull market, helping to reduce overall strategy risk. We find the reduction in "Crying Bear" as a consequence of our combination to be a powerful result which should inspire confidence in its practical usage and reliability.

References

[1] Adrian R. Pagan and Kirill A. Sossounov. A simple framework for analysing bull and bear markets. Journal of Applied Econometrics, 18(1):23–46, 2003.

[2] Asger Lunde and Allan Timmermann. Duration dependence in stock prices: An analysis of bull and bear markets. Journal of Business & Economic Statistics, 22(3):253–273, 2004.

[3] Md Rafiul Hassan and Baikunth Nath. Stock market forecasting using hidden markov model: a new approach. In 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), pages 192–196. IEEE, 2005.

[4] Nguyet Nguyen and Dung Nguyen. Hidden markov model for stock selection. Risks, 3(4):455–473, 2015.

[5] Matthew Wang, Yi-Hong Lin, and Ilya Mikhelson. Regime-Switching Factor Investing with Hidden Markov Models. JRFM, 13(12):1–15, December 2020.

[6] Constandina Koki, Stefanos Leonardos, and Georgios Piliouras. Do cryptocurrency prices camouflage latent economic effects? a bayesian hidden markov approach. In Multidisciplinary Digital Publishing Institute Proceedings, volume 28, page 5, 2019.

[7] Snehanshu Saha Luckyson Khaidem Sudeepa Roy Dey Suryoday Basak, Saibal Kar. Predicting

the direction of stock market prices using tree-based classifiers. The North American Journal of Economics and Finance, 47:552–567, 2019.

[8] Peikun Wang Xu Jiali Yue Yang, Yang Wu. Stock price prediction based on xgboost and lightgbm. E3S Web Conf. 275 01040 (2021).

[9] Mohammadreza Mahmoudi and Hana Ghaneei. Detection of structural regimes and analyzing the impact of crude oil market on canadian stock market: Markov regime-switching approach. 2021.

[10] Peter Nystrup, Bo William Hansen, Henrik Madsen, and Erik Lindström. Detecting change points in VIX and S&P 500: A new approach to dynamic asset allocation. Journal of Asset Management, 17(5):361–374, September 2016.

[11] Travis Berge and Òscar Jordà. A chronology of turning points in economic activity: Spain, 1850–2011. SERIEs: Journal of the Spanish Economic Association, 4(1):1–34, March 2013.

[12] Leonard E. Baum and John A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. Bulletin of the American Mathematical Society, 73:360–363, 1967.

[13] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE transactions on Information Theory, 13(2):260–269, 1967.

[14] Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. The annals of mathematical statistics, 41(1):164–171, 1970.