Detecting Market Regime Changes for Trading

Dhruv Baid dbaid
Rohan Prasad rprasad2
Sarthak Vishnoi svishnoi
Uday Sharma udayshar

Pittsburgh Campus

Abstract

This project used machine learning to detect and predict changes in the market regime with the aim of improving existing trading strategies. Our approach used directional change (DC) indicators derived from the price series of the S&P 500 index. Our analysis had two parts: using a hidden Markov model on the training data to detect regimes retrospectively, and training three classifiers – naive Bayes (NBC), logistic regression (LR), and support vector machines (SVM) – on the labeled data. We evaluated the models based on the performance of a trading strategy that used the regime data and compared it to baseline strategies that didn't take regimes into account. We were able to improve the baseline trading strategies for all three models. The NBC performed the best.

1 Introduction

A regime change can be defined as a significant, and often abrupt, change in the behavior of financial markets. This change usually coincides with major changes in the global economy, policy, or regulation (*Regime Changes and Financial Markets*, 2011). Understanding and predicting regime changes are of great importance to all participants in the market as they can cause asset prices to behave erratically. In the event of a regime change, trading strategies based on assumptions on price dynamics need to be re-calibrated or even scrapped altogether (Tsang and Jun, 2018).

We can look at various statistical properties of prices when defining regimes. Past literature has found that the market has historically operated under one of two regimes – a stable low-volatility regime and a distressed high-volatility regime (Chen and Tsang, 2021). Hence, we chose to study regimes in the context of volatility.

The aim of our project is to detect and predict regime changes with the intent of improving existing trading strategies. We built upon the approach from the book "Detecting Regime Changes in Computational Finance"(Chen and Tsang, 2021). The authors used high-frequency data to train a hidden Markov model (HMM) to detect regimes retrospectively. They also used the output of the HMM to train a NBC to make real-time predictions on the prevalent market regime. Our work's main contribution is three-fold:

1. The authors used a regime-based strategy that was both mean-reverting and momentum-following. However, their baseline strategy was only mean-reverting. We used an additional momentum-based baseline strategy to get a better idea about the improvement offered by trading strategies which take regime information into account.
2. We used a more comprehensive and goal-focused approach to tuning the models – in particular, we tuned the model based on the validation set performance of the trading strategy. We cross-validated different values of the directional change (DC) threshold, different DC indicators, and different thresholds for classifiers to predict an abnormal regime.
3. We compared three classification algorithms – NBC, LR, and SVM – to determine which one performed best in terms of predicting the regimes.

We sought to answer the following research questions:

1. Can these regimes help us improve existing trading strategies?
2. Does any particular model perform better than the others in the context of the trading strategy?
3. What characteristics do these regimes show?

We used daily price data for the S&P 500 (^GSPC). Instead of using the traditional time series data, which are sampled at fixed intervals, we looked at DC data (Chen and Tsang, 2021). Under this event-based framework, we construct a subset of the price data that has an alternating sequence of uptrends and downtrends (Petrov et al., 2019). In DC, a trend is reversed only when the percentage price change crosses a threshold value (θ). Under this framework,

the data dictates when to sample the time points, providing an alternative way of looking at traditional time series data (Chen and Tsang, 2021). The advantage of this approach is that it allowed us to focus on significant market movements and filtered away a lot of the noise commonly present in financial data.

We used a subset of the DC data from the period 2005-01-01 to 2017-12-31 as the training set and from 2018-01-01 to 2019-12-31 as the validation set. We used a grid search on a set of parameters for creating the DC indicators (described in Section 2.2) for each classifier. For each set of parameters, we constructed the DC indicators, used the HMM to label the training data, and trained the classifiers on the training data. Following this, we predicted the regimes on the validation set using one of the three classifiers. The models were evaluated based on the performance of the trading strategy on the validation data – for each of the three models, we found the optimal set of parameters that maximized the profitability of the trading strategy.

We compared the three classifiers based on the validation set performance using the corresponding optimal parameters and picked the best one. We then made predictions on the test set from 2020-01-01 to 2022-12-31 using the best model. To validate the results, we looked at the characteristics of the predicted regimes using scatterplots of the DC indicators.

2 Methods

2.1 Data Collection and Preprocessing

We used Yahoo Finance (and its Python API) to obtain daily price data from 2005-01-01 to 2022-12-31 for the S&P 500. Since the authors used high-frequency data for their experiments, we tried to capture some of the intraday price movement in our model as well. Instead of only using the adjusted closing prices, we used both the opening and closing price and offset them by a factor of 12 hours on each date. Although the core trading session of the market is from 09:30 to 16:00 ("NYSE: NYSE Market Information", n.d.), we chose to use a 12-hour offset so that the opening and closing prices were equispaced. This helped double the number of data points we had.

2.2 Directional Change

Directional change is an event-based analysis of the time series. This method samples some points from the original time series, making the underlying assumption that the points which are not sampled do not add significant information about the market.

2.2.1 Directional Change Events

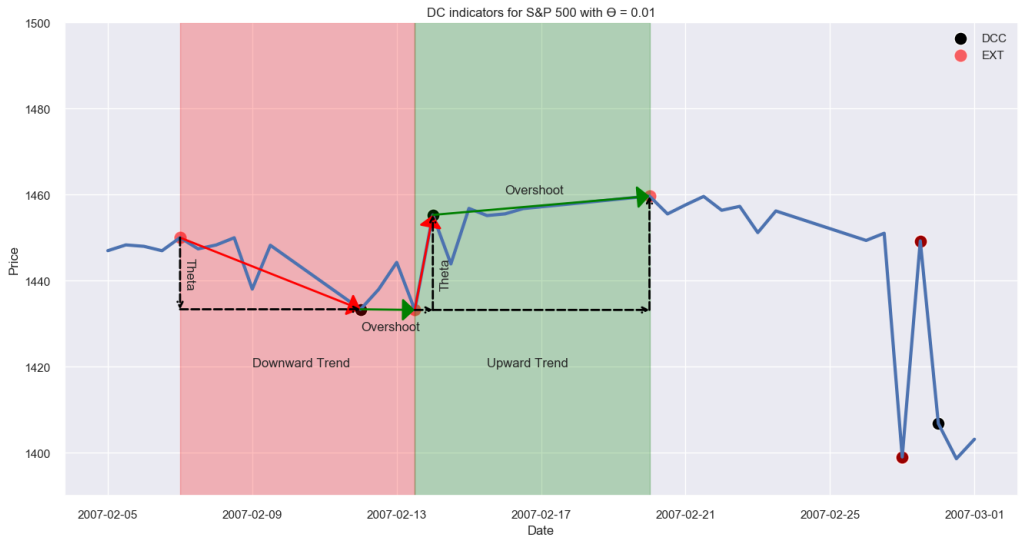

Figure 1: Market events under the DC framework.

The DC framework splits the market into 4 different events, as shown in Figure 1. A DC event starts at an extreme point (P_{EXT}) and ends at a directional change confirmation point (P_{DCC}). The hyperparameter θ is the threshold percentage price change at which we can define a DCC event (see (1), from Chen and Tsang, 2021):

$$P_t = P_{\text{DCC}} \iff \left| \frac{P_t - P_{\text{EXT}}}{P_{\text{EXT}}} \right| \geq \theta \tag{1}$$

We then transition to an overshoot event which continues until a reversal in trend happens. In the DC series, we sample the P_{EXT} and the P_{DCC} points from the entire time series.

2.2.2 Directional Change Indicators

The next step was to define three DC indicators as per Chen and Tsang, 2021:

1. **Total Price Movement (TMV)**, which measures the magnitude of price changes between extreme events:
$$\text{TMV}_{\text{EXT}}(n) \coloneqq \frac{P_{\text{EXT}}(n) - P_{\text{EXT}}(n-1)}{P_{\text{EXT}}(n-1) \times \theta} \tag{2}$$
2. **Time for Completion (T)**, which measures the time between extreme events:
$$T(n) = t_{\text{EXT}}(n) - t_{\text{EXT}}(n-1) \tag{3}$$
3. **Time-Adjusted Return (R)** combines the two indicators above and measures the change per unit time in the price between extreme events:
$$R(n) = \frac{\text{TMV}_{\text{EXT}}(n)}{T(n)} \times \theta \tag{4}$$

These indicators serve to quantify directional changes observed in the market and are the inputs to our HMM (Section 2.3). We used cross-validation (Section 2.6) to determine which of the indicators provided the most improved trading strategy over the control strategies.

2.3 Detecting Regimes Retrospectively – Hidden Markov Models

Hidden Markov models (HMM) are a class of statistical models that model a sequence of observations assumed to have been generated by an "underlying stochastic process that is not observable" (Rabiner and Juang, 1986). There are implicit underlying probability densities separately describing the state transitions, i.e., the probability that one state moves to the other (**A**) and the probability of observing a particular observation given an underlying state (**B**). The model assumes that the state sequence is a first-order Markov process – the probability of a particular state depends on only the previous state (Chen and Tsang, 2021). The DC series which we have created is based on the fact that the market undergoes changes among different events, as shown in Figure 1. This ensures that the next observation in the DC series only depends upon the current state, which is a weak condition implying that the regimes also follow a Markov process. Thus, using an HMM to model the regimes in the market is a fair assumption.

In our model, we assume that there are two underlying (hidden) states: an abnormal regime, characterized by high volatility (regime = 1), and a normal regime, characterized by low volatility (regime = 0). This is similar to the work done by Mahmoudi and Ghaneei, 2022 on the Canadian market. We observe a sequence of prices, which are then converted to different indicators (see Section 2.2) before being fed into the model. The model then attempts to classify each indicator data point into one of these two regimes.

The HMM model cannot differentiate between marking a regime 1 or 0. In other words, a label output of 1 could also have been 0 and vice versa. To train classifiers and run cross-validation, we need a unique pairing between HMM output labels and the regime characterization. This is done by standardizing the labels given by HMM. We calculate the total time the HMM predicts 1 and the time it predicts 0. We then flip the labels if, and only if, the market spends more time in label 1. This is done to ensure that the normal regime (label 0) happens for the majority of the time and the abnormal regime (label 1) happens less.

In its implementation, fitting a hidden Markov model involves using the Expectation-Maximization algorithm, which is prone to get stuck in a local minima for the loss function. In order to avoid this issue, we train the model 10 times, starting with different random initializations, and pick the one with the highest score (log-likelihood). This is the method used in the official documentation for the `hmmlearn` Python package ("Tutorial", n.d.).

The DC indicators are continuous, so we used a Gaussian distribution to model each of the two states. The Gaussian assumption seemed somewhat reasonable when we plotted a histogram of the training set indicators:

Through cross-validation, we selected an appropriate DC indicator and threshold to use. We then fit an HMM to the training data for that indicator and used the model to label the regimes of the training data.

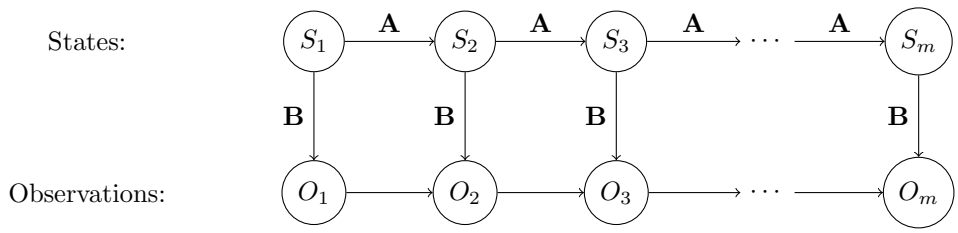

Figure 2: Hidden Markov model diagram. Based on Chen and Tsang, 2021. **O** are the observations, and **S** are the hidden states (regime 0 or 1) that the corresponding observations come from. m is the total number of observations.

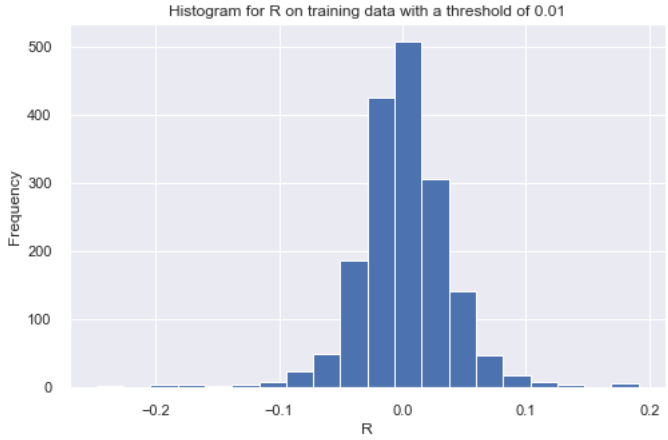

Figure 3: Histogram for the time-adjusted return (R) indicator (training set) for a threshold (θ) of 0.01.

2.4 Predicting Regimes – Machine Learning Algorithms

We trained each of the classifiers on the training data labeled by the HMM. These models could then be used to make predictions about the regime on the validation and test sets given the DC indicator as the covariate.

2.4.1 Naive Bayes Classifier

This algorithm is based on Bayes' theorem, which gives a formula for conditional probabilities. Since we have a single covariate X (the DC indicator), and our response variable $Y \in \{0, 1\}$ (regime) is binary, this classifier implements the following algorithm (Hastie et al., 2001):

$$\mathbb{P}\left[Y = y | X = x\right] = \frac{\mathbb{P}\left[X = x | Y = y\right] \mathbb{P}\left[Y = y\right]}{\mathbb{P}\left[X = x\right]}$$

We generated labels for each data point $X = x$ based on the following expression. ε is the threshold probability that we need from the classifier to classify the regime as abnormal.

$$\widehat{y} = \begin{cases} 0 & \text{if } \mathbb{P}\left[Y = 1 | X = x\right] \leq \varepsilon \\ 1 & \text{if } \mathbb{P}\left[Y = 1 | X = x\right] > \varepsilon \end{cases} \tag{5}$$

Since we only have one covariate, we did not need to worry about the independence assumption of the naive Bayes classifier. We also assumed Gaussian kernels for modeling the conditional probability in the model, which is revisited in Section 4.1.

2.4.2 Logistic Regression Classifier

This classifier models the log-probability of an event occurring (i.e., of the event $Y = 1$) as a linear function of the covariates (Hastie et al., 2001). Once again, since we have a single covariate, we can model the probability function as

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \tag{6}$$

The assumption that the log-odds is a linear combination of the covariates is a weak assumption, and reasonable in this case. We use ε in this model as well to change the threshold to predict the abnormal class.

2.4.3 Support Vector Machine Classifier

This classifier aims to construct a hyperplane in the feature space which maximizes the margin (the distance between differently-labeled points) (Hastie et al., 2001). We use predicted probabilities and the threshold hyperparameter ε to predict the probability of the abnormal region. The regimes are predicted using Equation (5).

This model has the fewest assumptions out of the three and focuses more on the points near the boundary. We include it in case our assumptions for the other models aren't reasonable. In this model, too, we define ε as the threshold probability to predict the abnormal regime.

2.5 Trading Strategy

We develop a novel way to develop our loss function for cross-validation. Extending the work by Chen and Tsang, 2021, we created three different trading strategies – two control strategies and a regime-dependent strategy. The first control strategy (CT1) is a basic mean-reverting strategy, while the second control strategy (CT2) is a basic momentum strategy. The regime-dependent strategy is mean-reverting during the low-volatility regimes and momentum-based during high-volatility regimes. We ensure that no look-ahead bias is present in any of the three strategies. These are described in detail in Table 1 and Appendix B.

Table 1: Trading strategies, based on Chen and Tsang, 2021. RCD refers to the regime change date.

Algorithms		Regime Dependent Strategy	Control Strategy – Mean Reversion (CT1)	Control Strategy – Momentum (CT2)
Normal regime (0) - low volatility	Open	Open **contrarian** position when |TMV| reaches 0.5	Open **contrarian** position when |TMV| reaches 0.5	Open **trend following** position when |TMV| reaches 0.5
	Close	Close at next DCC or **RCD**	Close at next DCC	Close at next DCC
Abnormal regime (1) - high volatility	Open	Open **trend following** position when |TMV| reaches 0.5	Same as in normal regime	Same as in normal regime
	Close	Close at next DCC or **RCD**		

2.6 Cross-Validation and Final Models

Combining the sequence above into a single pipeline, we determined that these were the hyperparameters that needed to be tuned with the following values:

Table 2: Cross-validation parameters.

Parameter	Values
θ	$\{0.01, 0.015, 0.02\}$
DC indicator	{'TMV', 'T', 'R'}
ε	$\{0.6, 0.65, 0.7, 0.75, 0.8\}$
Classifier	{naive Bayes, logistic regression, SVM}

Inspired by scikit-learn's GridSearchCV class, we developed our own class to iterate over a grid of the above-mentioned parameters and find the optimal set of parameters which maximizes the profit generated by the regime-based trading strategy. For a given set of parameters, our cross-validation pipeline consisted of the following steps:

- Use θ to generate the three DC indicators for each of the training, validation, and test sets.
- Fit a hidden Markov model on the DC indicator, one at a time, on the training set and get regime labels.
- Train the classifier, one at a time, on the labeled training set and make predictions on the validation set with the model threshold ε.
- Backtest the regime-dependent strategy on the validation set and get the profit, Sharpe ratio, and maximum drawdown (defined in Appendix Section A).

Note that as we made the assumption that the state process is first-order Markov (i.e., the current state depends only on the previous state), we did not need to worry about the time ordering of data while training and validating the models.

The optimal parameters for each classifier are given below, and the grid search results are available in Appendix C.

Table 3: Optimal parameters for the three classifiers.

Model	θ	ε	DC Indicator
naive Bayes	0.01	0.8	R
logistic regression	0.02	0.6	T
SVM	0.02	0.6	T

Based on these parameters, we trained each model and made predictions on the test set. We also ran the control strategy with these parameters on the test set to see if the regimes helped improve the results.

3 Results

3.1 Trading Strategy Performance

We compared three metrics on the test set – profit, Sharpe ratio, and maximum drawdown (MDD) – for the control and the regime-dependent trading strategies using each of the three models' classifications of the market's current regime. In all three cases, we found that the control strategies performed worse than the regime-based strategies on almost all metrics except MDD, indicating that our regime prediction algorithm was able to improve trading strategies. These results are summarized in Table 4 below.

Table 4: Test set (2020-01-01 to 2022-12-31) metrics for regime-based and control strategies across three classifiers.

Algorithm	Regime-Based			Mean-Reverting Control			Momentum Control		
	Profit	Sharpe	MDD	Profit	Sharpe	MDD	Profit	Sharpe	MDD
naive Bayes	1.0824	1.21	0.2506	0.3710	0.48	0.1543	0.3541	0.52	0.1546
logistic regression	0.7528	0.75	0.2138	0.4069	0.46	0.2702	0.5019	0.61	0.2587
SVM	0.7528	0.75	0.2138	0.4069	0.46	0.2702	0.5019	0.61	0.2587

Note that the control strategies for LR and SVM have the same metrics because we use the same threshold θ and DC indicator ('R') for these algorithms.

In Table 4 and Table 3, we see that the regime-based results for SVM and logistic regression are exactly the same. This is an interesting result, and we explore it further in Section 4.1.

The naive Bayes model performs the best out of the three models. We explore its results further in Section 3.2.

3.2 Regime Characteristic Plots

We look at the regimes predicted by the naive Bayes classifier using the optimal parameters. As the test set could be unusual due to the COVID-19 pandemic, we also look at the predictions on the validation set (although the model was tuned on this set, the predicted regimes can still help build intuition). See Figure 4. The plots are explored in greater detail in Section 4.2.

We also look at the characteristic plots of the regimes, as defined by Chen and Tsang, 2021. As we used the indicator R to make the regimes, we plot the other two indicators TMV and T, normalized to a 0-1 scale using min-max scaling. See Figure 5.

These results validate our initial assumption – the abnormal regime corresponds to high volatility. From the plot, we see that the points in regime 1 (abnormal) correspond to normalized T values that are very close to 0. This means that the price returns hit the up and down thresholds within a very short period of time, indicating high volatility.

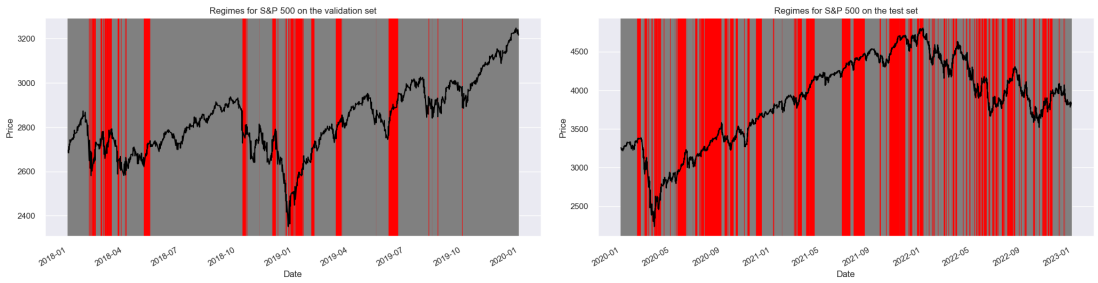

Figure 4: Predictions on the validation (L) and test (R) sets. Red is the abnormal regime, and gray is normal.

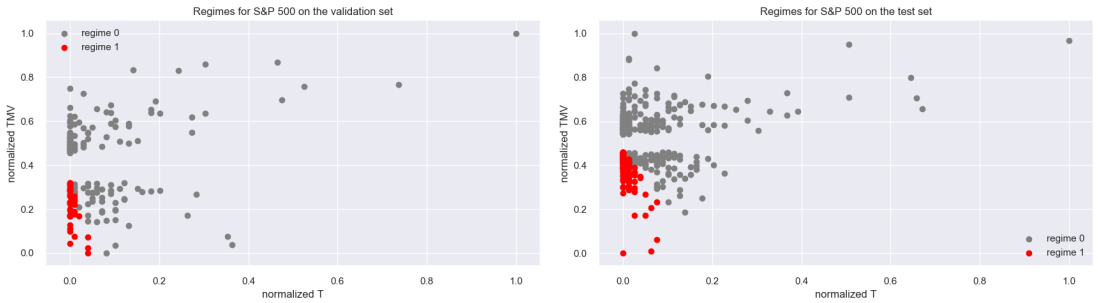

Figure 5: Regime characteristic plots on the validation (L) and test (R) sets.

4 Discussion

4.1 An Interesting Result

We found that the logistic regression and SVM models gave an identical performance on the test set, implying that they predicted the same regimes. As we only have a single covariate, it is possible that the two classifiers found the same decision boundary since both of them were trained on the same training set. We confirmed this by looking at the scatterplot of the points in the test set, colored according to the predicted regimes.

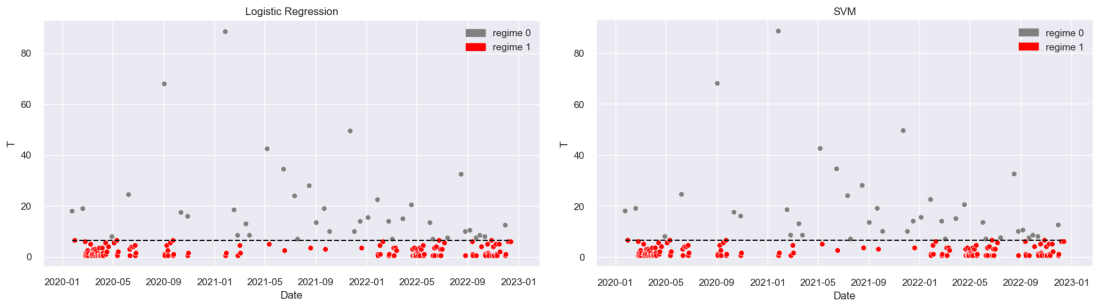

Figure 6: Scatterplot and decision boundary on the test set for logistic regression (L) and SVM (R).

This is not surprising since both of the models do not make strong assumptions about the data. Naive Bayes, on the other hand, assumes the distribution of the covariate given the class it belongs to. There are two interesting questions that we try to answer – why does naive Bayes perform better than the other two classifiers? Why does naive Bayes choose 'R' instead of 'T'?

We looked into the output of the HMM on the training set. After cross-validation, we found that LR and SVM performed better on the DC indicator 'T', and NBC performed better on the indicator 'R'. For each indicator, we plot a histogram of the indicator values separated by the labeled regime.

The histogram for 'R' looks Gaussian. On the other hand, the histogram for 'T' looks more like an exponential distribution rather than Gaussian. The data for 'R' has increased variance; however, the variance for both labels is similar. The data for 'T', on the other hand, has a very low variance for regime 1 while a very high variance for regime 0. This makes naive Bayes fit better on the 'R' indicator and not that much on the 'T' indicator. Naive Bayes

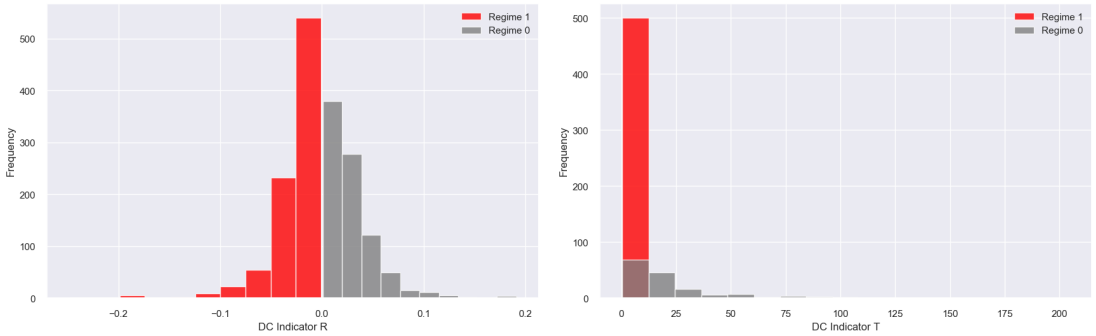

Figure 7: Histogram for DC indicators R (left) and T (right), separated by labeled regimes.

essentially has four parameters to learn, the two class means and variances while SVM and logistic regression have fewer parameters or are focused more on the geometry of the data. This makes naive Bayes more complex, thus fitting better to the data and giving better results with the trading strategy.

4.2 Conclusion

We get the following answers to the three research questions posed in the introduction:

1. **Can these regimes help us improve existing trading strategies?** Yes, the regimes can help in improving existing trading strategies in terms of profitability and Sharpe ratio. The trading strategy that used the regimes outperformed both the momentum and mean-reverting control strategies.
2. **Does any particular model perform better than the others in the context of the trading strategy?** The naive Bayes model outperforms the other two models. This is because we found that the DC indicator 'R' is normally distributed. The other two models do not make strong assumptions on the data and are hence not able to leverage this.
3. **What characteristics do these regimes show?** We found using the characteristic plots that these regimes correspond to periods of high volatility, as we had initially postulated. Our model is able to detect some of the obvious regime changes caused by macroeconomic events, like the oil crash of November 2018 (Egan, 2018), the COVID-19 pandemic in 2020 ("A timeline of covid-19 developments in 2020", 2021), and the Russian invasion of Ukraine in February 2022 ("The Russia-Ukraine crisis: What does it mean for markets?", 2022). It also detects some regimes that are not linked to obvious macroeconomic events providing valuable information for trading.

4.3 Limitations and Future Research

- It could be instructive to compare the regime classifications obtained using the DC indicators with those obtained using time series data (i.e., realized volatility of returns). According to Chen and Tsang, 2021, these two approaches are complementary, and finding a way to combine the two models' predictions (after passing the HMM output into any of the three classifiers) might lead to more accurate predictions.
- We can use a more general version of a hidden Markov model, which allows us to use multiple observations to predict the underlying hidden state based on the assumption that all input observation series are together Markov, but not independently.
- The methodology for standardizing the regime labels is not foolproof due to the assumption that the low volatility regime will show up for a large proportion of the time for *every* dataset. If we have a small dataset with enough points from the abnormal regime, we might end up flipping the correct labels.
- We can relax the assumption that there are only two underlying regimes in the market. We can try a model which partitions the market into high-vol, low-vol, and normal regimes. This will help us configure the trading strategy and limits according to different regimes, thus enabling us to get even better profits and a better drawdown.
- It could be interesting to look at the problem of regime detection qualitatively. We can combine the HMMs for different asset classes in a country – like yield curves, major stock and bond indices, FX, and so forth – and check how the regimes are modeled with all this data. We can combine these different models to qualitatively describe the market regimes in the country and draw some conclusions about the economy and other major political events that affected the financial markets.

References

Chen, J., & Tsang, E. P. (2021). *Detecting regime change in computational finance: Data science, machine learning and algorithmic trading* (First edition.). CRC Press, Taylor & Francis Group.

Egan, M. (2018). The great oil crash of 2018. https://www.cnn.com/2018/11/21/investing/oil-prices-trump-saudi-arabia/index.html

Fernando, J. (2022). Sharpe ratio definition. https://www.investopedia.com/terms/s/sharperatio.asp

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning*. Springer New York Inc.

Hayes, A. (2022). Maximum drawdown (mdd) definition. https://www.investopedia.com/terms/m/maximum-drawdown-mdd.asp

Mahmoudi, M., & Ghaneei, H. (2022). Detection of structural regimes and analyzing the impact of crude oil market on canadian stock market: Markov regime-switching approach. *Studies in Economics and Finance*.

Nyse: Nyse market information. (n.d.). https://www.nyse.com/markets/nyse-arca/market-info

Petrov, V., Golub, A., & Olsen, R. (2019). Instantaneous volatility seasonality of high-frequency markets in directional-change intrinsic time. *Journal of Risk and Financial Management*, *12*(2). https://doi.org/10.3390/jrfm12020054

Rabiner, L. R., & Juang, B. H. (1986). An introduction to hidden markov models. *IEEE ASSP Magazine*, *3*. https://doi.org/10.1109/MASSP.1986.1165342

Regime changes and financial markets. (2011). National Bureau of Economic Research.

The russia-ukraine crisis: What does it mean for markets? (2022). https://www.jpmorgan.com/insights/research/russia-ukraine-crisis-market-impact

A timeline of covid-19 developments in 2020. (2021). https://www.ajmc.com/view/a-timeline-of-covid19-developments-in-2020

Tsang, E., & Jun, C. (2018). Regime change detection using directional change indicators in the foreign exchange market to chart brexit. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *2*. https://doi.org/10.1109/TETCI.2017.2775235

Tutorial. (n.d.). https://hmmlearn.readthedocs.io/en/latest/tutorial.html#training-hmm-parameters-and-inferring-the-hidden-states

Appendix

A Metrics

The following are the metrics used to evaluate the different strategies:

- PNL: This is the cumulative profit or loss generated by the trading strategy and is calculated by taking the cumulative sum of the daily returns. This assumes that on each day, a new position is entered and the previous position is closed, with a fixed notional of \$1.
- Sharpe Ratio: This is used to measure the risk-adjusted return. Assuming a risk-free rate of zero, it is calculated by dividing the annualized return by the annualized volatility (Fernando, 2022).

  $$\text{Sharpe Ratio} = \frac{r_p}{\sigma_p}$$

  where r_p and σ_p are the annualized return and standard deviation of the portfolio, respectively.
- Maximum Drawdown (MDD): Theoretically, MDD is the maximum observed loss from a peak to a trough of a portfolio before a new peak is attained (Hayes, 2022). It is calculated as shown in Equation 7. We use a proxy for this, which finds the minimum subarray of the returns using Kadane's algorithm. This is given by Equation 8.

  $$\text{MDD} = \frac{\text{Trough Value} - \text{Peak Value}}{\text{Peak Value}} \tag{7}$$

  $$\text{MDD} = \min_{0 \le i \le j \le N} \sum_{k=i}^{j} ret_k \tag{8}$$

B Trading Strategy

The three trading strategies which we implemented don't have a look-ahead bias which is one of the main problems which drive high profits. In the trading strategy, we define $|\mathbf{TMV}|$ at all points of the time series by calculating it from the last extreme point (P_{EXT}). We assume a frictionless market and no transaction costs during trading. The two control strategies are defined in B.1 while the regime-based trading strategy is defined in B.2.

B.1 Control Strategies

We use two control strategies – one is momentum based, while the other is mean-reverting-based. In the momentum strategy, we take positions according to the market when $|\mathbf{TMV}|$ reaches a value of 0.5. We close this position at the next DCC event. In the mean reverting control strategy, we take a position opposite to the market according to the principle as defined above. None of the actions in the strategy depend upon the regime we are in or whether or not a regime change happens.

B.2 Regime-Dependent Strategy

In the regime-dependent strategy, we utilize the information about the market's current regime, which our two-class classifiers predict. We have different actions for the low-volatility and high-volatility regimes. We follow the same mean-reverting strategy as the control during the low vol times, but we close the position when a regime change is detected in addition to a DCC point. In the high-vol regime, we follow the market and trade on momentum. We open a trend-following position when the $|\mathbf{TMV}|$ reaches a value of 0.5. We close this position when a regime change is detected or when a DCC point is attained.

C Results

The following tables depict the cross-validation results from three models we tested: naive Bayes, logistic regression, and support vector machine. Rows corresponding to profit-maximizing sets of parameters are shaded in grey.

C.1 Naive Bayes

Table 5: Naive Bayes cross-validation results

Iteration	DC indicator	ε	θ	MDD	Profit	Sharpe
1	R	0.6000	0.0100	0.1306	0.1129	0.4522
2	R	0.6000	0.0150	0.1311	0.0908	0.3084
3	R	0.6000	0.0200	0.1431	0.0867	0.2944
4	R	0.6500	0.0100	0.0959	0.2301	0.6338
5	R	0.6500	0.0150	0.1014	0.2384	0.5882
6	R	0.6500	0.0200	0.1150	0.1805	0.4641
7	R	0.7000	0.0100	0.0959	0.2942	0.7512
8	R	0.7000	0.0150	0.1014	0.2581	0.6227
9	R	0.7000	0.0200	0.1150	0.1864	0.4742
10	R	0.7500	0.0100	0.0849	0.3053	0.7536
11	R	0.7500	0.0150	0.1014	0.2693	0.6308
12	R	0.7500	0.0200	0.1287	0.1361	0.3851
13	R	0.8000	0.0100	0.0849	0.3579	0.8304
14	R	0.8000	0.0150	0.1038	0.2130	0.5317
15	R	0.8000	0.0200	0.1187	0.1441	0.3992
16	TMV	0.6000	0.0100	0.1548	0.0566	0.2791
17	TMV	0.6000	0.0150	0.1606	0.0880	0.3511
18	TMV	0.6000	0.0200	0.1724	0.0711	0.2866
19	TMV	0.6500	0.0100	0.1548	0.0566	0.2791
20	TMV	0.6500	0.0150	0.1606	0.0880	0.3511
21	TMV	0.6500	0.0200	0.1724	0.0711	0.2866
22	TMV	0.7000	0.0100	0.1548	0.0566	0.2791
23	TMV	0.7000	0.0150	0.1606	0.0880	0.3511
24	TMV	0.7000	0.0200	0.1724	0.0711	0.2866
25	TMV	0.7500	0.0100	0.1548	0.0566	0.2791
26	TMV	0.7500	0.0150	0.1606	0.0880	0.3511
27	TMV	0.7500	0.0200	0.1724	0.0711	0.2866
28	TMV	0.8000	0.0100	0.1548	0.0566	0.2791
29	TMV	0.8000	0.0150	0.1606	0.0880	0.3511
30	TMV	0.8000	0.0200	0.1724	0.0711	0.2866
31	T	0.6000	0.0100	0.1497	0.0806	0.2926
32	T	0.6000	0.0150	0.1514	0.2048	0.5091
33	T	0.6000	0.0200	0.1732	0.2101	0.4956
34	T	0.6500	0.0100	0.1497	0.0806	0.2926
35	T	0.6500	0.0150	0.1514	0.2048	0.5091
36	T	0.6500	0.0200	0.1732	0.2101	0.4956
37	T	0.7000	0.0100	0.1497	0.0806	0.2926
38	T	0.7000	0.0150	0.1514	0.2048	0.5091
39	T	0.7000	0.0200	0.1732	0.1880	0.4603
40	T	0.7500	0.0100	0.1497	0.0806	0.2926
41	T	0.7500	0.0150	0.1514	0.2048	0.5091
42	T	0.7500	0.0200	0.1732	0.1880	0.4603
43	T	0.8000	0.0100	0.1497	0.0806	0.2926
44	T	0.8000	0.0150	0.1514	0.2048	0.5091
45	T	0.8000	0.0200	0.1732	0.1880	0.4603

C.2 Logistic Regression

Table 6: Logistic regression cross-validation results

Iteration	DC indicator	ε	θ	MDD	Profit	Sharpe
1	R	0.6000	0.0100	0.1548	0.0566	0.2791
2	R	0.6000	0.0150	0.1606	0.0880	0.3511
3	R	0.6000	0.0200	0.1724	0.0711	0.2866
4	R	0.6500	0.0100	0.1548	0.0566	0.2791
5	R	0.6500	0.0150	0.1606	0.0880	0.3511
6	R	0.6500	0.0200	0.1724	0.0711	0.2866
7	R	0.7000	0.0100	0.1548	0.0566	0.2791
8	R	0.7000	0.0150	0.1606	0.0880	0.3511
9	R	0.7000	0.0200	0.1724	0.0711	0.2866
10	R	0.7500	0.0100	0.1548	0.0566	0.2791
11	R	0.7500	0.0150	0.1606	0.0880	0.3511
12	R	0.7500	0.0200	0.1724	0.0711	0.2866
13	R	0.8000	0.0100	0.1548	0.0566	0.2791
14	R	0.8000	0.0150	0.1606	0.0880	0.3511
15	R	0.8000	0.0200	0.1724	0.0711	0.2866
16	TMV	0.6000	0.0100	0.1548	0.0566	0.2791
17	TMV	0.6000	0.0150	0.1606	0.0880	0.3511
18	TMV	0.6000	0.0200	0.1724	0.0711	0.2866
19	TMV	0.6500	0.0100	0.1548	0.0566	0.2791
20	TMV	0.6500	0.0150	0.1606	0.0880	0.3511
21	TMV	0.6500	0.0200	0.1724	0.0711	0.2866
22	TMV	0.7000	0.0100	0.1548	0.0566	0.2791
23	TMV	0.7000	0.0150	0.1606	0.0880	0.3511
24	TMV	0.7000	0.0200	0.1724	0.0711	0.2866
25	TMV	0.7500	0.0100	0.1548	0.0566	0.2791
26	TMV	0.7500	0.0150	0.1606	0.0880	0.3511
27	TMV	0.7500	0.0200	0.1724	0.0711	0.2866
28	TMV	0.8000	0.0100	0.1548	0.0566	0.2791
29	TMV	0.8000	0.0150	0.1606	0.0880	0.3511
30	TMV	0.8000	0.0200	0.1724	0.0711	0.2866
31	T	0.6000	0.0100	0.1497	0.0806	0.2926
32	T	0.6000	0.0150	0.1514	0.2077	0.5484
33	T	0.6000	0.0200	0.1732	0.2101	0.4956
34	T	0.6500	0.0100	0.1497	0.0806	0.2926
35	T	0.6500	0.0150	0.1514	0.2077	0.5484
36	T	0.6500	0.0200	0.1732	0.1880	0.4603
37	T	0.7000	0.0100	0.1497	0.0806	0.2926
38	T	0.7000	0.0150	0.1514	0.2077	0.5484
39	T	0.7000	0.0200	0.1732	0.1880	0.4603
40	T	0.7500	0.0100	0.1497	0.0806	0.2926
41	T	0.7500	0.0150	0.1514	0.2077	0.5484
42	T	0.7500	0.0200	0.1732	0.1054	0.3243
43	T	0.8000	0.0100	0.1497	0.0806	0.2926
44	T	0.8000	0.0150	0.1514	0.2048	0.5091
45	T	0.8000	0.0200	0.1732	0.1054	0.3243

C.3 Support Vector Machine

Table 7: SVM cross-validation results

Iteration	DC indicator	ε	θ	MDD	Profit	Sharpe
1	R	0.6000	0.0100	0.1548	0.0566	0.2791
2	R	0.6000	0.0150	0.1606	0.0880	0.3511
3	R	0.6000	0.0200	0.1724	0.0711	0.2866
4	R	0.6500	0.0100	0.1548	0.0566	0.2791
5	R	0.6500	0.0150	0.1606	0.0880	0.3511
6	R	0.6500	0.0200	0.1724	0.0711	0.2866
7	R	0.7000	0.0100	0.1548	0.0566	0.2791
8	R	0.7000	0.0150	0.1606	0.0880	0.3511
9	R	0.7000	0.0200	0.1724	0.0711	0.2866
10	R	0.7500	0.0100	0.1548	0.0566	0.2791
11	R	0.7500	0.0150	0.1606	0.0880	0.3511
12	R	0.7500	0.0200	0.1724	0.0711	0.2866
13	R	0.8000	0.0100	0.1548	0.0566	0.2791
14	R	0.8000	0.0150	0.1606	0.0880	0.3511
15	R	0.8000	0.0200	0.1724	0.0711	0.2866
16	TMV	0.6000	0.0100	0.1548	0.0566	0.2791
17	TMV	0.6000	0.0150	0.1606	0.0880	0.3511
18	TMV	0.6000	0.0200	0.1724	0.0711	0.2866
19	TMV	0.6500	0.0100	0.1548	0.0566	0.2791
20	TMV	0.6500	0.0150	0.1606	0.0880	0.3511
21	TMV	0.6500	0.0200	0.1724	0.0711	0.2866
22	TMV	0.7000	0.0100	0.1548	0.0566	0.2791
23	TMV	0.7000	0.0150	0.1606	0.0880	0.3511
24	TMV	0.7000	0.0200	0.1724	0.0711	0.2866
25	TMV	0.7500	0.0100	0.1548	0.0566	0.2791
26	TMV	0.7500	0.0150	0.1606	0.0880	0.3511
27	TMV	0.7500	0.0200	0.1724	0.0711	0.2866
28	TMV	0.8000	0.0100	0.1548	0.0566	0.2791
29	TMV	0.8000	0.0150	0.1606	0.0880	0.3511
30	TMV	0.8000	0.0200	0.1724	0.0711	0.2866
31	T	0.6000	0.0100	0.1497	0.0806	0.2926
32	T	0.6000	0.0150	0.1514	0.2077	0.5484
33	T	0.6000	0.0200	0.1732	0.2101	0.4956
34	T	0.6500	0.0100	0.1497	0.0806	0.2926
35	T	0.6500	0.0150	0.1514	0.2077	0.5484
36	T	0.6500	0.0200	0.1732	0.2101	0.4956
37	T	0.7000	0.0100	0.1497	0.0806	0.2926
38	T	0.7000	0.0150	0.1514	0.2077	0.5484
39	T	0.7000	0.0200	0.1732	0.1880	0.4603
40	T	0.7500	0.0100	0.1497	0.0806	0.2926
41	T	0.7500	0.0150	0.1514	0.2077	0.5484
42	T	0.7500	0.0200	0.1732	0.1880	0.4603
43	T	0.8000	0.0100	0.1497	0.0806	0.2926
44	T	0.8000	0.0150	0.1514	0.2048	0.5091
45	T	0.8000	0.0200	0.1732	0.1054	0.3243

D Code

We collaborated using GitHub. The entire codebase can be found at this link.