PREDICTING THE STOCK MARKET USING A HIDDEN MARKOV MODEL

Josh Jiang **MyungJin Lee** **Elishua Shumpert**

1 Introduction

Forecasting stock prices is a problem as old as stock markets themselves. The stock market, or any financial market, is volatile and very difficult to predict. The price of financial securities depend on a multitude of factors that can be economic, geopolitical, social, etc. To assist with predicting stock prices, traders have become increasingly dependent on using sophisticated statistical models that can detect signals from data. In this paper, we use a Hidden Markov Model (HMM) to forecast the price of the SPY ETF.

The SPDR S&P 500 ETF Trust (SPY) is an exchange traded fund (ETF) that tracks the S&P500. The S&P500 is a well known stock market index that tracks the performance of roughly 500 large companies listed on stock exchanges in the United States. Due to its coverage of large companies in every sector of the stock market, it is considered to be a de facto barometer for the health of the stock market as a whole. As such, SPY is a prime candidate for our investigation into stock price prediction.

Through its history, the stock market in the United States is known to transition between periods with their own unique sets of market conditions and sentiment. The two most common terms are known as bull markets – where market conditions are generally favorable for stocks, investor sentiment is high, and stock prices generally rise – and bear markets – where conditions and sentiment are bad and stock prices generally fall. To frame this phenomenon from a Bayesian perspective, the stock market exists in an unobservable state, the effect of which we see reflected in the changes in the stock market. Such a view lends itself to modeling the stock market as an HMM.

In section 2, we will elaborate on the data used in this analysis. Section 3 will go into the architecture of the HMM and how it will be used to predict the stock market. Section 4 will include the results of our analysis. Section 5 will be a discussion on the results of our analysis. [1]

2 Data

The data for this project is daily SPY data from 1/27/1994 to 3/25/2022 – for our analysis, we only used data from about the 4 most recent years. The data was accessed from Yahoo Finance[3]. The attributes of the dataset include the open price, high price, low price, close price, adjusted close price, and volume. In addition, we also engineered new features from the original data features that will be used in fitting the model. Specifically, we added the following features: `return`, `volume deviation`, `range`, and `variance`.

`Return`, for a given day, is the percent change from the day's close and yesterday's close. `Volume deviation` is defined as the ratio of the day's volume and the average volume in the past year. `Range` is the day's high minus the day's low divided by the day's low. `Variance` is the variance of the past ten day's returns. We defined and used these features because they are ways of representing the original data as stationary processes, which is convenient as it frees the model from its dependence on time.

2.1 Exploratory data analysis

For the exploratory analysis, we decided to analyze the behavior of the stock price features over the time-frame of the data from 1993 to current day. Figure 1 displays the effect of the stock price features: daily adjusted close prices, daily volume, difference in open and close prices, and difference of high and low prices over time. In terms of daily adjusted close prices, we notice several distinct changes in the stock market adjusted closing prices of S&P 500 from 1993 to current day. More specifically, we see four major bullish trends, or increasing trends, and three major bearish trends, or

[1] All of our analysis for this project can be found in our GitHub repo [2].

decreasing trends, in the adjusted close prices. The adjusted close prices increase overall from 1993 to 2000, 2003 to 2007, and from 2009 to 2020. On the other hand, the bearish trends in adjusted close prices roughly occurs from 2000 to 2003, 2007 to 2009 and a clear steep drop in 2020.

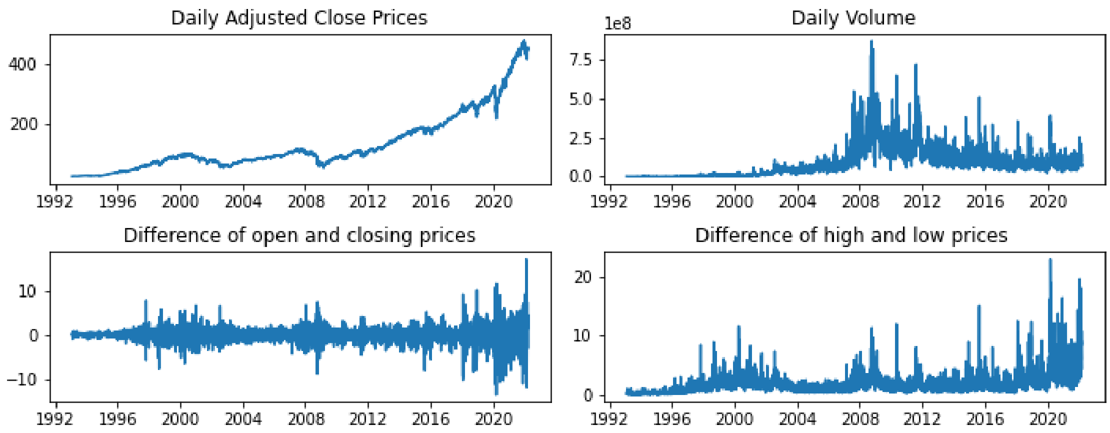

Figure 1: Panel plot showing daily adjusted close prices, daily volume, difference of open and close prices and difference of high and low prices of the S&P 500 stock index from 1993-2022.

We note that open prices, close prices, high prices, and low prices all show the same exact trends as adjusted close prices, therefore, we conclude that all of these stock price features are highly correlated which led to the decision of deriving new features from the original data. Thus, we decided to omit these variables except for adjusted close price which we plan to use for prediction in our models. In the other three plots from Figure 1, there are some differing patterns for volume, difference in open and closing prices and difference in high and low price over the time series. Looking at daily volume from the top right plot, the behavior in the number of shares traded is much more erratic and unstable over the time period. It is much harder to infer where the bullish and bearish trends are taking place based on the volume. One noticeable trend that can be seen is the sharp increase in number of shares traded from 2002 to 2008. We infer that this is mostly likely due to the dot-com bubble in the late 1990s with the massive growth in advancement of the Internet and information becoming more available due to this innovation. In the lower left panel of the plot, the pattern in the difference of open and close prices is much different than the daily volume in that the difference in prices is less volatile with certain spikes throughout time. These spikes roughly occur in 1997, 1998, 2000, 2008, 2015, 2018, 2020 and even during the current year. It is also apparent that the absolute difference in prices have been increasing significantly lately since 2020, thus, we can mostly likely attribute this to the Coronavirus pandemic that began during that time. Likewise, the same trends occur in the difference of high and low prices where the difference is mostly stable with some peaks occurring approximately around the same time periods as was noted for the difference in open and closing prices and the difference in high and low prices increasing on average more significantly from 2020 and on.

As mentioned earlier, these features have a dependence on time. In general, volume, price, and the difference in price goes up over time. For the model, it would be beneficial to remove this dependence. Figure 2 shows the features used from 2018-2022, roughly the time frame that was used in our analysis. The features exhibit patterns that are consistent with the trends mentioned earlier. In particular, we see a spike in volatility during March of 2020, corresponding to the onset of the COVID-19 pandemic. By fitting an HMM to this type of data, we hope that it will distinguish these patterns and make accurate predictions using it.

3 Analysis

For our analysis, we attempt to predict the stock market adjusted close prices of the S&P 500 stock index as accurately as possible. So, this is inherently a time series prediction problem. We used an HMM to tackle it. We also implemented a simple autoregressive integrated moving average (ARIMA) model to compare its performance with our more sophisticated model. Both models are evaluated by the mean squared error (MSE) on a test data set. The test data set is from February 1, 2021 to March 25, 2022.

3.1 Hidden Markov model

Hidden Markov models are a unique tool for modelling time series data. This suggests that HMMs may be suited for financial market prediction. They are used in many applications including speech recognition, DNA sequence analysis,

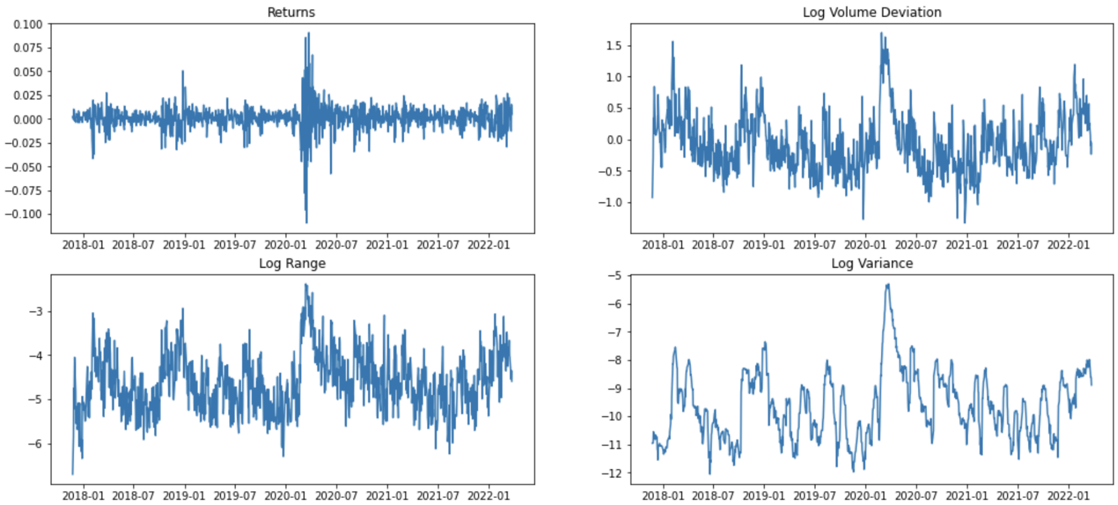

Figure 2: Panel plot showing daily return, log volume deviation, log range, and log variance from 2018-2022.

and hand written characters recognition [4]. Specifically, a Hidden Markov Model is a Bayesian network representing probability distributions over a sequence of observations.

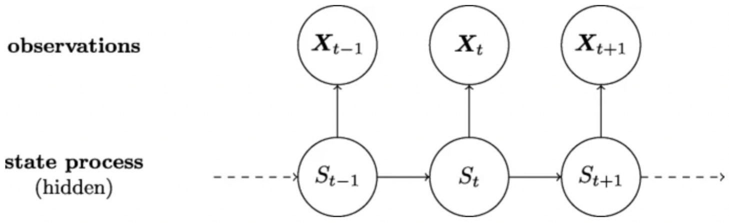

Figure 3: Basic structure of a Hidden Markov Model [5]

The fundamental assumption of an HMM is that the underlying process is a Markov Chain of unobservable states. Each state has their own unique distributions on what is observed – the data. More specifically, the observation X_t at time t was generated by some process whose state S_t is hidden from the observer. This process can be seen in figure 3. It assumes that the states of the hidden process follow the Markov property which requires that given the value of S_{t-1}, the current state S_t is independent of all the states prior to $t-1$. HMMs also assume that the hidden state variables are discrete.

The Hidden Markov model consists of the following: the number of hidden states, number of observation symbols, state transition probabilities, observation emission probability distribution that characterizes each state and the initial state distribution. The parameters of the HMM model is given by $\lambda = \{A, B, \pi\}$ where A is the transition matrix and the entry a_{ij} represents the transition probability from state i to state j, B represents the observation emission matrix and the entry $b_j(X_t)$ represents the probability of observing X_t at state j, and π is the prior probability distribution of being in a given state at the beginning of the experiment. We are using a Gaussian HMM, meaning the underlying emission distribution for each state is multivariate normal, $\mathcal{N}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, and p is the number of states where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are two more parameters that are included in λ. There are three fundamental problems for the HMMs:

1. Given observation data X and model parameters of λ, how do we compute probabilities of the observations $P(X|\lambda)$?
2. Given the observation data X and the model parameters of λ, how do we find the best hidden state sequence of X?
3. Given the observation data X, how do we estimate the model parameters that best explains the observed data?

The first and second problems can be solved by Viterbi algorithm and the Forward-Backward algorithm respectively. The last one is solved by an Expectation-Maximization algorithm known as the Baum-Welch algorithm. However, we will not go into the detail of these algorithms. For our analysis, we used the python package `hmmlearn`[1], which uses the algorithms mentioned to fit the HMM.

To predict the closing price of SPY for day $t + 1$, we first define a lag time l. The HMM is trained on the data from day $t - l$ to t and the return is predicted for the next day based on the predicted state for tomorrow. To find an optimal HMM, we treat the number of hidden states, n, and the number of lag days, l, for training as hyper-parameters to configure. We set the validation data set to contain the daily stock data from September 12, 2018 to January 30, 2021. We use a grid search of parameter values for each hyper-parameter and train the HMM using the parameter values that minimize the train mean squared error. Predictions of the adjusted close price for each day in the test set are computed using the following formula:

$$\widehat{\text{Adj Close Price}}_{t+1} = \text{Adj Close Price}_t \times (1 + \widehat{Return}_{t+1})$$

where $\widehat{\text{Adj Close Price}}_{t+1}$ and $\widehat{Return}_{t+1}$ are the predicted adjusted close price and predicted return and Adj Close Price_t is the adjusted close price of the previous day.

3.2 ARIMA benchmark

To investigate the effectiveness of the Bayesian approach, we consider the ARIMA model to compare with our HMM. AutoRegressive Integrated Moving Average, or ARIMA models are a class of statistical models that are often used to analyze time series. These models are extensively used for financial data.

Forecasting with an ARIMA model consists of 3 steps:

1. Check stationarity of data
2. Tune parameters p,d and q
3. Construct a rolling-forecast algorithm

We call a time series stationary when its statistical properties are constant over time, i.e. no observable trends and similar looking short-term patterns. If our data is not stationary, we may achieve stationarity through differencing the time series. The Augmented Dickey-Fuller test, or ADF test is a widely used method for checking stationarity. The ADF test tests the null hypothesis that there is a unit root in the time sample. If we can reject the null hypothesis, the time series can be recognized as stationary.

A non-seasonal ARIMA model is often called an ARIMA(p,d,q) model, indicating the parameters that are specified from the data. Each of the 3 parameters refer to: the number of autoregressive terms(p), the difference order(d), and the order of moving average(q). Although these parameters can be estimated through investigating the autocorrelation and partial autocorrelation in the data, we make use of the Auto ARIMA algorithm. Auto ARIMA is a variation of the Hyndman & Khandakar algorithm, which is a step-wise process including unit root tests and ARIMA fittings to minimize the Akaike Information Criterion and MSE.

The ARIMA model returns the single prediction of the day after the window. We design an algorithm to make predictions for a given period of time, by continuously "sliding" the training data window to the next day. We adopt the `lag` term to define the width of the window. With the returned predictions on `return`, we obtain the predicted adjusted close price, and calculate the MSE.

4 Results

The HMM and ARIMA models were both used to predict the next day for every day in the test set. The hyperparameters of the HMM were tuned using a training set and were chosen to minimize MSE between predicted and actual closing price for the next day. The parameters chosen were $(n, l) = (4, 900)$. Similarly, this was done for the ARIMA model and the resulting parameters were $(p, d, q) = (1, 0, 2)$.

	HMM	ARIMA
MSE	4810.08	5286.96

Table 1: Test MSE for HMM and ARIMA

Table 1 shows the MSE for both models on the test data. Of course, there is more to the models than just MSE. Figure 4 is a plot of the predicted closing prices for both models. Due to the auto-regressive nature of the ARIMA model, it tends to predict the same close for the next day as the current day. That is, the model predicts that the return for the following day will regress to the mean, which is close to 0. As such, the predicted close price of the ARIMA model is roughly the actual stock price shifted right by 1 day – this might be difficult to see in the actual figure. Meanwhile, the HMM does have the ability to predict a directional change based on whatever the underlying state is. Furthermore, since variance of returns is a feature that is used in the model, confidence intervals can be easily computed for the model. Though confidence intervals are also obtainable with the ARIMA model, it requires more computation than HMM, and thus we omit them here.

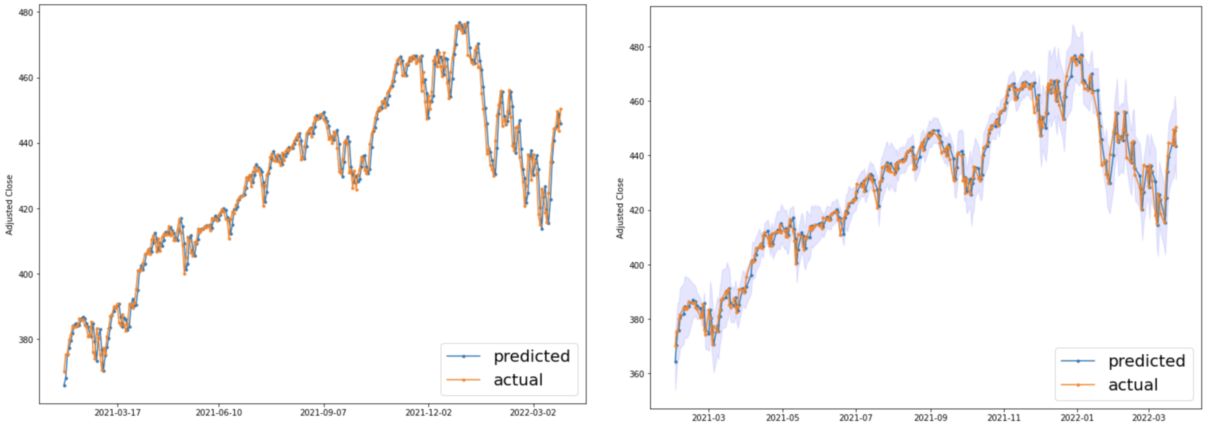

Figure 4: Comparison of ARIMA (left) and HMM (right) results. The HMM plot includes 95% confidence intervals.

4.1 Regime detection

For further analysis of the HMM, we also analyzed the individual states themselves. There is no particularly intuitive way to do this as each point in the test set was predicted using a different set of training data. Instead, we decide to simply train a HMM on the test set alone and analyze that model. Figure 5 show the predicted states for each day in the test data. The placement of the states allow for very intuitive interpretations of what they represent. Clearly, states 1 and 3 represent bullish market conditions that push stocks upward. Meanwhile, state 0 represents bearish market conditions, where the market is in a general downtrend. State 2 seems to be a mixed state that usually borders the bullish states from the bearish one.

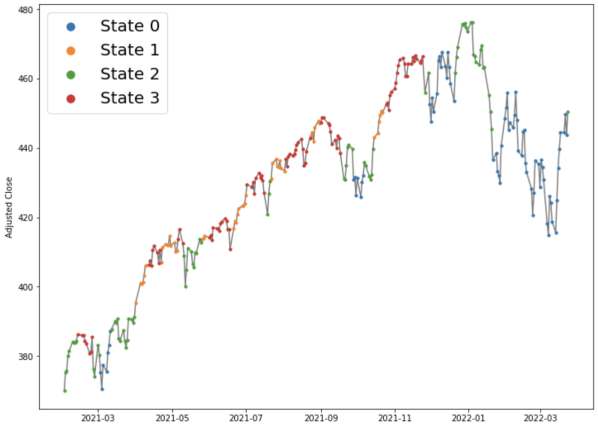

Figure 5: Predicted states of test data.

Investigating the states more deeply, we looked at the marginal distribution for each observable variable in each state. Keep in mind that the observable variables are modeled as a multivariate normal but in order to plot and analyze them, we looked at the marginals. Figure 6 contains plots of the marginal distributions of each feature variable and table 2 contains the mean of each variable/state combination.

These plots provide more color into the states themselves. For example, we see that the two bullish states, 1 and 3, share similar characteristics. In particular, they both have positive returns, below average volume, and relatively small ranges and variance compared to the other states. Meanwhile the bearish state has negative returns, above average volume, and the largest range and variance. The mixed state 2 also has positive return but with volume, range, and variance in

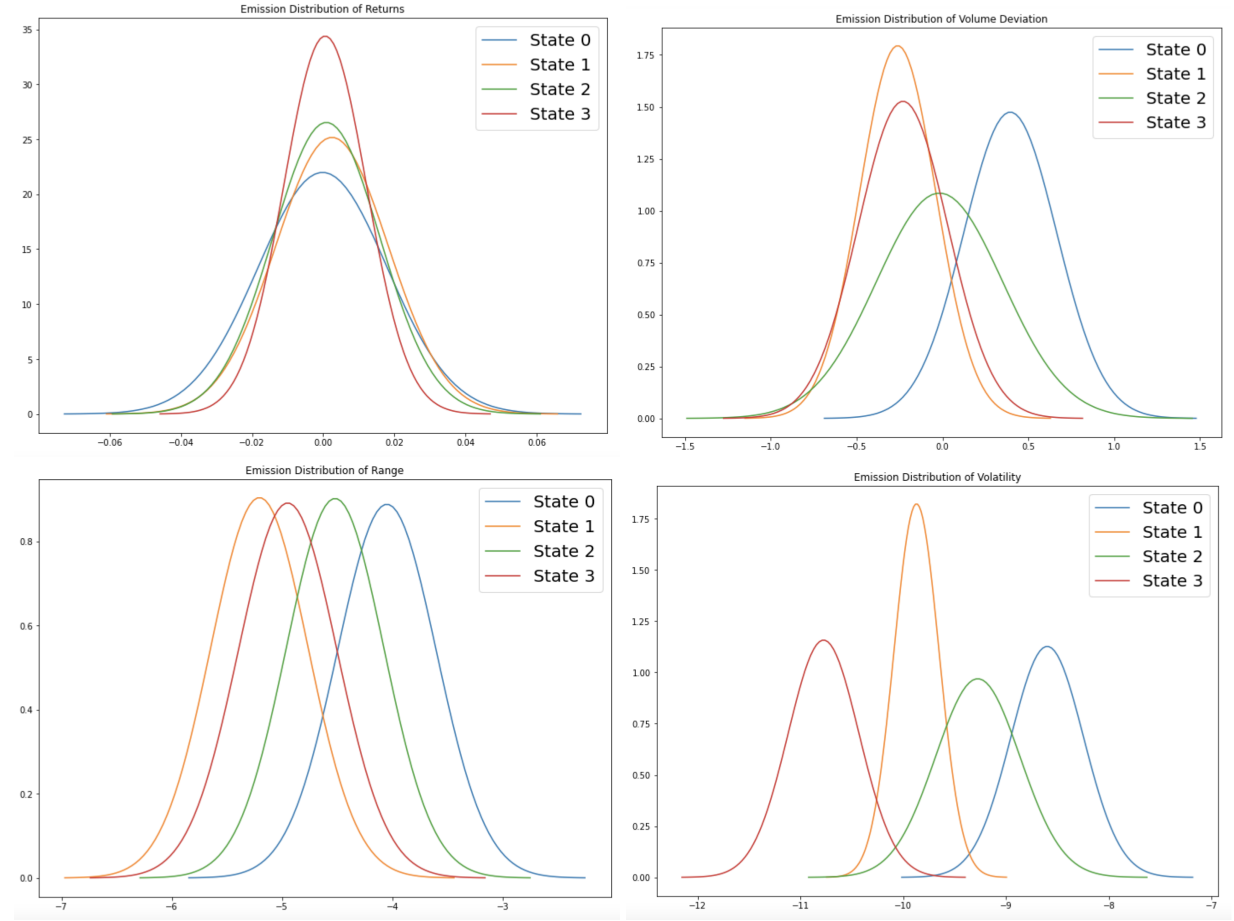

Figure 6: Marginal emission distribution of feature variables.

between the bullish and bearish states. These findings are consistent with the commonly accepted characteristics of bull and bear markets. In particular, bull markets are characterized to have positive returns and low volatility while bear markets have negative returns and high volatility.

	0	1	2	3
Return	-0.00015	0.00249	0.00091	0.00057
Log volume dev	0.3947	-0.2605	-0.0172	-0.2297
Log range	-4.049	-5.205	-4.519	-4.949
Log variance	-8.600	-9.872	-9.275	-10.776

Table 2: Means for each feature for each state.

5 Discussion

The results of our investigation shows positive signs in using HMMs to predict the stock market. At the very least, we were able to show that HMMs are significantly better than a naive predictor like the ARIMA model. The advantage is that HMMs are capable of predicting directional movement. If the HMM believes that the process is in a bearish state, it could predict a negative return for the next day and vice versa if it thinks it is in a bullish state. Meanwhile, naive predictors like ARIMA models tend to predict mean reversions in next day returns, which is ignoring a lot of signals that the HMM picks up. The results of our analysis show that analyzing volume and volatility – in the form of range and return variance – provides some signal in predicting stock price movement.

HMMs can be useful even when not used to predict stock prices. The ability for HMMs to predict an underlying state – regime detection – is powerful enough to be useful. HMM regime detection can be simply used to inform investors on the state of the market, which can influence their decision-making process. For example, if the model is showing that the model is currently in a bearish state, perhaps it is time to consider employing a conservative investment strategy.

Another advantage of HMMs is that its mechanistic structure allows for easy interpretability of the results. An analysis of the fitted states showed patterns that were consistent with prior knowledge of bull and bear market characteristics. The notion of the stock market being a hidden Markov process is also an intuitive abstraction due to people attributing general market conditions as reasons for price movement.

There are several assumptions about our HMM that does not necessarily hold up in reality. First and foremost is the fundamental Markov assumption. That is, the underlying states can be modeled as a Markov chain, with the next state depending only on the previous. While this property is convenient for many reasons, it does leave out a lot of historical information that could be useful. Such an assumption does not allow for the model to consider the "big picture" of the stock market but only what has happened in the immediate term. Our model also assumes that the emission distributions are normal. Stock market data tend to be heavy tailed with chances for extreme outliers. Using a normal distribution to approximate this kind of data is often untenable. As such, a heavier tailed distribution, like a t-distribution, for example, would be better suited for this problem.

The use of MSE as a metric for performance is convenient but not informative. Since the ultimate goal is to use HMMs for developing trading strategies, it would be interesting to simulate trading strategies using HMMs and using the return as a performance metric in the future.

References

[1] hmmlearn Website. https://hmmlearn.readthedocs.io/en/latest/index.html.

[2] Our GitHub Repo. https://github.com/zjiang2/551_final_project.

[3] yfinance Website. https://pypi.org/project/yfinance/.

[4] M. Hassan and B. Nath. Stock market forecasting using hidden markov model: A new approach. *5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*, 2005.

[5] V. Popov, A. Ellis-Robinson, and G. Humphris. Modelling reassurances of clinicians with hidden markov models. *BMC Medical Research Methodology*, 2019.