Hidden Markov Model on Stock Price Prediction

Ian Quan

Supervisor: Omidali Aghabahaei Jazi

Department of Mathematical and Computational Sciences, University of Toronto Mississauga

August 20, 2024

Contents

1 Introduction

The prediction of the stock market, with its inherent volatility and potential for substantial financial gains, has long been a focus of sophisticated market participants such as institutional investors, hedge funds, and proprietary trading firms. These entities leverage advanced methods to make accurate predictions about future price movements and trends, aiming to gain a competitive edge and maximize their investment returns. Among the various approaches employed, Hidden Markov Models (HMMs) have emerged as a powerful tool for analyzing and predicting time series data, including stock prices.

Hidden Markov Models are statistical models that represent systems with hidden states through observable events. In essence, an HMM consists of a set of hidden states, transition probabilities between these states, and emission probabilities that link each state to specific observations. This framework has proven effective in a wide range of applications where temporal dynamics are crucial. In recent years, many strategies have been developed for algorithmic trading. Among these, Hidden Markov Models (HMMs) have emerged as powerful tools for the prediction of time series data, including stock market predictions. Hidden Markov Models are statistical models that represent systems with hidden states through observable events. In essence, an HMM is defined by a set of states, transition probabilities between these states, and emission probabilities that relate each state to a particular observation. This framework has proven to be highly effective in analyzing and predicting time-dependent events.

Several studies have explored the application of HMMs in stock market prediction. For example, HMMs have been used to model the daily opening, closing, high, and low prices of stocks, utilizing fractional changes in these quantities as features
[Catello et al.(2023)Catello, Ruggiero, Schiavone, and Valentino]. These models have shown promising results in predicting future stock prices, offering a probabilistic framework that captures the inherent uncertainty and dynamics of the market.

In this research report, we propose an approach to stock price prediction using Hidden Markov Models. Our method involves training a unique HMM for each stock using historical data from Yahoo Finance. We incorporate features such as daily fractional differences in stock values and the fractional deviation of intraday maximum and minimum values to generate accurate predictions. The goal is to develop an intelligent forecasting model that can assist investors in making informed investment decisions, ultimately enhancing their ability to navigate the complexities of the stock market.

This paper is organized as follows: in Section 2, we provide an overview of basic concepts in HMM, including some definitions and mathematical equations; Section 3 introduces our methodology; in Section 4, we describe our technical implementation, focusing on the prediction methodology. Section 5 presents the results. Section 6 discusses the limitations of the HMM approach. Finally, Section 7 concludes the paper by summarizing key findings and suggesting directions for future research.

2 Review of Basic Concepts

There are 5 components we need to define to run our HMM:

- **Hidden states** - The hidden states are are are not directly observable. The set of hidden states in an HMM is represented as:

$$S = \{S_1, S_2, \ldots, S_t\}$$

- **Set of observations** - The observations are related to the states but do not directly indicate the state. The set of possible observations generated by the states is:

$$O = \{O_1, O_2, \ldots, O_t\}$$

- **Transition matrix** - The transition matrix, A, contains the probabilities of moving from one state to another:

$$A = [a_{ij}] \quad \text{where} \quad a_{ij} = P(S_{t+1} = j \mid S_t = i)$$

  a_{ij} represents the probability of transitioning from state i to state j.

- **Emissions Matrix** - Also known as observation matrix, B, it contains the probabilities of observing each possible output from each state:

$$B = [b_j(k)] \quad \text{where} \quad b_j(k) = P(O_t = k \mid S_t = j)$$

  where $b_j(k)$ is the probability of observing the symbol k from state j.

- **Initial Probability Distribution** - The initial state probabilities, π, define the likelihood of the system starting in each state:

$$\pi = [\pi_i] \quad \text{where} \quad \pi_i = P(S_1 = i)$$

  where π_i is the probability that the Markov chain will start in state i.

2.1 Assumptions of HMM

Markov Assumption: The probability of moving to the next state depends only on the present state and not on how the present state was reached (memoryless property). Consider a sequence of state variables $S_1, S_2, ..., S_t$

$$P(S_t|S_1...S_{t-1}) = P(S_t|S_{t-1})$$

Output Independence: The probability of an output observation depends only on the state that produced the observation and it is independent of any other states or previous observations. Consider O_t is the observation at time t

$$P(O_t|S_1, O_1..., S_{t-1}, O_{t-1}, S_t) = P(O_t|S_t)$$

2.2 Likelihood Computation: The Forward Algorithm

The likelihood problem is to determine the probability of observing a specific sequence of observations given the model parameters(transition matrix A, emission matrix B and initial state probabilities π). This is typically solved using the Forward Algorithm, which efficiently computes the probabilities of observing a sequence given the model parameters.

Define the forward probability $\alpha_t(i)$ as the probability of observing the sequence up to time t and being in the state j at time t, given the model parameters:

$$\alpha_t(j) = P(o_1, o_2, \ldots, o_t, s_t = j \mid \lambda)$$

Initialization

The initialization of the forward probabilities at time 1 is given by:

$$\alpha_1(j) = \pi_j b_j(o_1)$$

where:

- π_j is the initial state probability of state j
- $b_j(o_1)$ is the probability of observing o_1 from state j.

Recursion

By the law of total probability, the recursive computation for subsequent times is given by:

$$\alpha_{t+1}(j) = \left(\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right) b_j(o_{t+1})$$

where:

- N is the number of states in the model such that $1 \leq j \leq N$.
- a_{ij} is the probability of transitioning from state i to state j.
- $b_j(o_{t+1})$ is the probability of observing o_{t+1} from state j.

Termination

The probability of observing the entire sequence $O = \{o_1, o_2, \ldots, o_T\}$ given the model λ is then computed by summing the forward probabilities at the final time step:

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

2.3 Decoding: The Viterbi Algorithm

The decoding problem in the context of Hidden Markov Models involves finding the most likely sequence of hidden states given an observed sequence of outputs. The **Viterbi Algorithm** is a dynamic programming algorithm specifically designed to solve the decoding problem efficiently. It identifies the most likely sequence of states that best explains a sequence of observations in an HMM. The algorithm works by building a path, state by state, which has the highest probability of leading to the observed sequence.

Initialization:

Begin by setting up the initial probabilities. Define $V[1, i]$ as the probability of i after observing the first observation and starting from the initial distribution. This is computed as:

$$V[1, i] = \pi_i b_i(o_1)$$

$$bt_1(j) = 0$$

where π_i is the initial probability of state i, and $b_i(o_1)$ is the emission probability of observing the first symbol in state i.

Recursion:

For each subsequent observation and each possible state, compute the probability of each path reaching state i at time t using the probabilities computed for time $t-1$. The probability $V[t, i]$ of the most likely path ending in state i at time t is given by:

$$V[t, i] = \max_{j}^{N}(V[t-1, j] \times a_{ji}) \times b_i(o_t)$$

Here, a_{ji} is the transition probability from state j to state i, and $b_i(o_t)$ is the emission probability of observing symbol o_t in state i.

Termination:

The probability of the most likely path that explains the entire observation sequence is obtained by taking the maximum over the last set of computed probabilities:

$$P^* = \max_i V[T, i]$$

where T is the total number of observations.

2.3.1 Path Backtracking:

To determine the most likely sequence of states, backtrack from the last observation. This involves tracing the path back from the state that has the highest probability at the final time step to the initial state, based on where each state probability came from at each step of the recursion.

2.4 Training: The Forward-Backward Algorithm

The Forward-Backward Algorithm is a fundamental method used in training Hidden Markov Models. The algorithm will let us train both the transition probabilities A and the emission probabilities B of the HMM. Here's an overview of the process:

1. The Forward Step

The Forward algorithm calculates the probability of the partial observation sequence up to a certain time step, given each state. This is done iteratively using the following steps:

Initialization:

$$\alpha_1(i) = \pi_i \cdot b_i(O_1)$$

where $\alpha_1(i)$ is the forward probability at time 1 for state i, π_i is the initial probability of state i, and $b_i(O_1)$ is the probability of observing O_1 in state i.

Induction:

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) \cdot a_{ij} \cdot b_j(O_t)$$

where a_{ij} is the transition probability from state i to state j, and $b_j(O_t)$ is the probability of observing O_t in state j.

Termination:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

where $P(O|\lambda)$ is the probability of the observation sequence O given the model λ, and T is the length of the observation sequence.

2. The Backward Step

The Backward algorithm calculates the probability of the partial observation sequence from a certain time step to the end, given each state. This is also done iteratively using the following steps:

Initialization:

$$\beta_T(i) = 1$$

where $\beta_T(i)$ is the backward probability at time T for state i.

Induction:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} \cdot b_j(O_{t+1}) \cdot \beta_{t+1}(j)$$

Termination:

$$P(O|\lambda) = \sum_{i=1}^{N} \pi_i \cdot b_i(O_1) \cdot \beta_1(i)$$

2.4.1 Combining Forward and Backward Probabilities

The Forward-Backward algorithm combines the forward and backward probabilities to compute the posterior probability of each state at each time step. This is given by:

$$\gamma_t(i) = \frac{\alpha_t(i) \cdot \beta_t(i)}{P(O|\lambda)}$$

where $\gamma_t(i)$ is the probability of i at time t given the observation sequence O.

Section 2 provides an overview of basic concepts in HMM, including some definitions and examples, Markov property, transition and emission matrices, fundamental problems in HMM including likelihood function, decoding, and learning, and some algorithms in HMM such as Viterbi, backward and forward algorithms. In section 3 introduces an application of HMM. In Section 4, we perform numerical study. Section 5 provides conclusions and closing remarks.

3 Methodology

In this study, we follow the prediction approach proposed by Nguyet Nguyen in the paper titled "Hidden Markov Model for Stock Trading" [Nguyen(2018)]. This methodology involves several key steps, beginning with the selection and preprocessing of historical stock price data, followed by the identification of the optimal number of hidden states for the HMMs. We then trained the models and evaluated their performance using various statistical criteria to ensure accuracy and robustness. The final model selection was based on a balance between model fit and complexity, guided by information criteria such as AIC, BIC, HQC, and CAIC.

3.1 Dataset

The dataset used in this research consists of historical stock price data for four companies: Apple Inc. (AAPL), NVIDIA Corp. (NVDA), Alphabet Inc. (GOOGL), and Microsoft Corp. (MSFT). The data spans the maximum available period for each stock and was retrieved using Yahoo Finance API [Finance(2024)].

For each stock, the data includes the following attributes:

- Open: The price at which the stock opened on a given trading day.
- High: The highest price the stock reached on a given trading day.
- Low: The lowest price the stock reached on a given trading day.
- Close: The price at which the stock closed on a given trading day.

The historical data was resampled to a monthly frequency to reduce noise and focus on longer-term trends. Any rows with missing or infinite values were removed to ensure the integrity of the dataset.

3.2 Data Prepossessing

The data prepossessing involved several key steps to ensure the integrity and suitability of the dataset for modeling with Hidden Markov Models (HMM). Initially, the OHLC (Open, High, Low, Close) prices were extracted from the raw stock data. Then, log returns were calculated for each price series to stabilize the variance and make the series more stationary, which is a crucial assumption for many time series models. The log returns are computed as the difference in the natural logarithm of consecutive prices, mathematically represented as

$$\text{Log Return} = log(\frac{P_1}{P_{t-1}})$$

where P_t and P_{t-1} are the prices at time t and $t-1$ respectively. This transformation helps in managing large fluctuations and normalizing the data.

Following the calculation of log returns, data cleaning was performed to remove any rows with infinite or NaN values, this step is critical as the presence of such values can significantly affect the performance and convergence of the HMM. After cleaning the data, the returns were standardized using the `StandardScaler` from the `sklearn` library. Standardization involves scaling the data to have a mean of zero and a standard deviation of one, which ensures that each feature contributes equally to the analysis and prevents features with larger scales from dominating the model.

3.3 Finding the Optimal Hidden State

This process involved exploring different configurations of hidden states to identify the optimal Hidden Markov Model (HMM) for each stock. The optimal number of hidden states must be chosen carefully since it directly affects the model's capacity to capture the underlying patterns and dynamics in stock market data.

We varied the number of hidden states from 2 to 10 and for each value, a Gaussian HMM was trained using the `hmmlearn` library. These models were trained with full covariance matrices, allowing for the modeling of the complete relationship between variables, and were iterated up to 1000 times to ensure convergence.

To evaluate and compare the performance of each model, several information criteria were calculated. These included the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), Hannan-Quinn Criterion (HQC), and Consistent Akaike Information Criterion (CAIC). These criteria are defined as follows:

- **Akaike Information Criterion (AIC)**:

$$\text{AIC} = -2\log(L) + 2k \tag{1}$$

- **Bayesian Information Criterion (BIC)**:

$$\text{BIC} = -2\log(L) + k\log(n) \tag{2}$$

where

- **Hannan-Quinn Criterion (HQC)**:

$$\text{HQC} = -2\log(L) + 2k\log(\log(n)) \tag{3}$$

- **Consistent Akaike Information Criterion (CAIC)**:

$$\text{CAIC} = -2\log(L) + k(\log(n) + 1) \tag{4}$$

where $\log(L)$ is the log-likelihood of the model, k is the number of parameters, and n is the number of samples.

For each stock, the models were evaluated using four statistical metrics—AIC, BIC, HQC, and CAIC—which balance model fit with complexity. The optimal number of hidden states was determined by selecting the model with the lowest scores across these criteria, ensuring accuracy without overfitting, thereby effectively capturing market dynamics without unnecessary complexity. The penalties for each metric differ: AIC imposes a linear penalty proportional to the number of parameters ($2k$), BIC scales this penalty with the logarithm of the sample size ($k\ln(n)$), HQC applies a slower-growing penalty using the logarithm of the sample size ($2k\ln(\ln(n))$), and CAIC further strengthens the penalty by adding an extra term to the BIC formula ($k(\ln(n)+1)$). These penalties ensure that the chosen models are well-balanced, avoiding excessive complexity while maintaining a good fit.

3.4 Results

The values of the performance metric are plotted in Figure 1, 2, 3 and 4, where a lower criterion value indicates a better model fit. The summary of optimal states for each stock is as follows:

- Amazon Inc. (AMZN): 6 hidden states
- NVIDIA Corporation (NVDA): 6 hidden states
- Apple Inc. (AAPL): 5 hidden states

The discrepancies in the optimal number of states across different criteria highlight the need for careful model selection based on the analysis objective. AIC often favors more complex models, potentially leading to overfitting, while BIC, HQC, and CAIC typically suggest simpler models to avoid this. Since our objective is to generalize the model to new data for prediction, we place more emphasis on the results of BIC, HQC, and CAIC when selecting the number of hidden states.

For AMZN and NVDA, BIC and HQC both recommend 6 hidden states, making it the preferred choice, as these criteria are conservative and generally reliable. For AAPL, BIC and CAIC both suggest 5 hidden states, so 5 is selected as the final model.

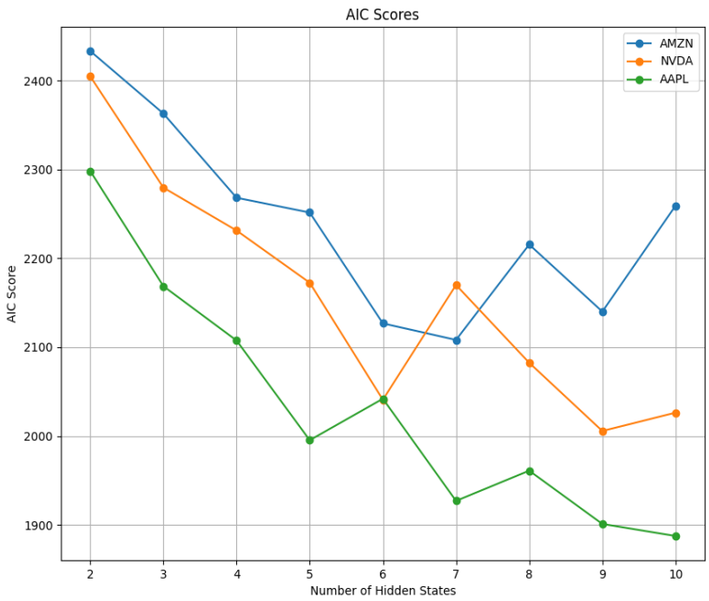

Figure 1: AIC scores

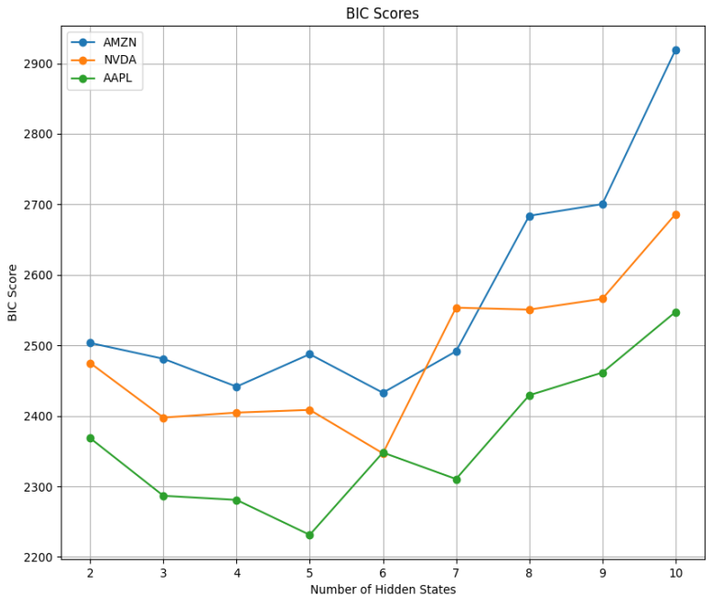

Figure 2: BIC scores

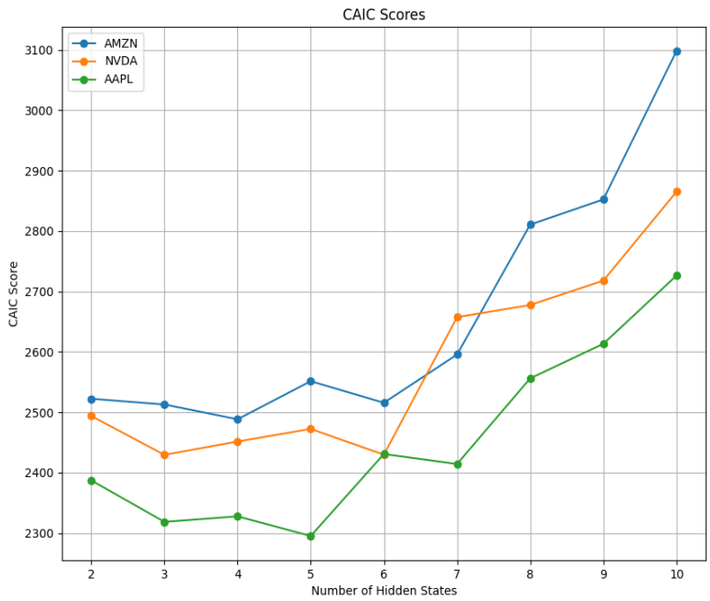

Figure 3: CAIC scores

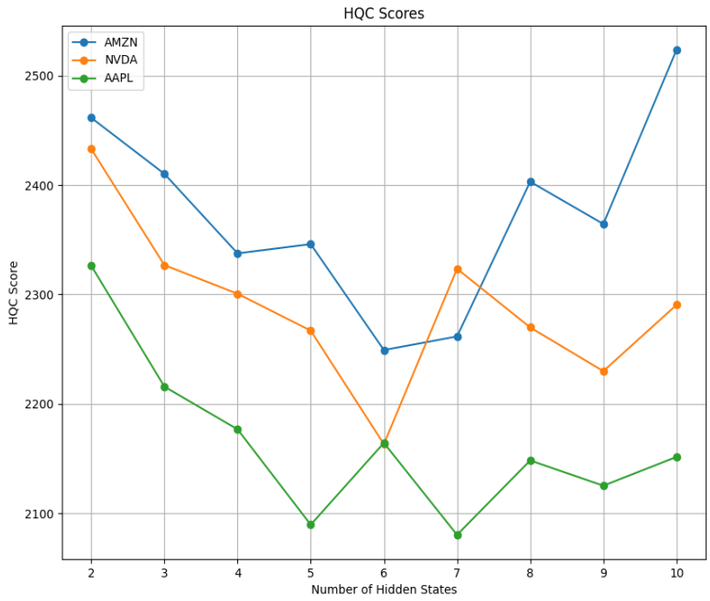

Figure 4: HQC scores

4 Implementation

4.1 Prediction Methodology

In this analysis, we employed a Hidden Markov Model (HMM) to predict future stock prices for three major stocks: Amazon (AMZN), Nvidia (NVDA), and Apple (AAPL). The primary objective was to forecast the prices of four key features—Open, High, Low, and Close—using historical data. The HMM was chosen due to its ability to model sequences of observations where the system being modeled is assumed to follow a Markov process with unobserved (hidden) states. This characteristic of HMMs makes them particularly well-suited for capturing the latent market conditions that influence stock prices.

4.2 Sliding Window Approach

To enhance the accuracy and robustness of our predictions, we adopted a sliding window approach. Specifically, we used a window of size $D = 96$ observations for training the model, and predictions were made for the subsequent $d = 96$ time steps. The sliding window method allowed us to iteratively update the model with new data, ensuring that the model remained relevant and adaptive to the latest market conditions.

This approach is particularly beneficial in financial time series forecasting, where the underlying data can exhibit non-stationarity, meaning that statistical properties such as mean and variance can change over time. By using a sliding window, we effectively captured the most recent market trends and reduced the risk of overfitting to outdated data.

4.3 Hidden Markov Model (HMM) Training

For each stock and feature, the HMM was trained using the following procedure:

- **Training Data:** At each time step, the model was trained using the past $D = 96$ observations from the historical data. This choice of window size was based on a balance between capturing enough historical information and avoiding overfitting to older, potentially less relevant data.
- **Model Initialization:** For the first iteration, a Gaussian HMM with an optimized number of hidden states was initialized and trained. The optimal number of hidden states was determined through cross-validation, balancing model complexity with predictive performance. For subsequent iterations, the model was re-initialized with parameters (start probabilities, transition matrix, and means) carried over from the previous iteration to ensure continuity and stability in the predictions. This transfer of parameters helped the model maintain consistency and leverage previously learned patterns.

The HMM can be described mathematically as follows:

$$P(X_t|S_t) = \mathcal{N}(X_t; \mu_{S_t}, \Sigma_{S_t})$$

Where:

- X_t represents the observed data at time t (e.g., stock prices).
- S_t represents the hidden state at time t.
- μ_{S_t} and Σ_{S_t} are the mean and covariance matrix of the Gaussian distribution associated with the hidden state S_t.

The parameters of the HMM are:

- **Start probabilities π:** This is the probability distribution over the initial state, representing the likelihood of the system starting in each hidden state.
- **Transition matrix A:** This matrix represents the probabilities of transitioning from one hidden state to another. It captures the dynamics of the market conditions and how they evolve over time.
- **Emission probabilities:** These are the likelihoods of the observed data given a particular hidden state, modeled as a Gaussian distribution. The emission probabilities link the hidden states to the observed stock prices.

4.4 Likelihood-Based Prediction

After training the HMM on the current window of data, we computed the likelihood of the training data under the fitted model. The likelihood function is given by:

$$\mathcal{L}(\theta|X_{1:T}) = P(X_{1:T}|\theta)$$

Where:

- θ represents the parameters of the HMM (start probabilities, transition matrix, means).
- $X_{1:T}$ represents the sequence of observed data up to time T.

The likelihood of the observed sequence given the model parameters provided a measure of how well the model explained the data. To predict future prices, we compared this likelihood to the likelihoods of historical windows to identify a past period where market conditions were most similar to the current period. The rationale behind this approach is that similar market conditions in the past may yield similar price movements in the future.

The predicted price at the next time step was calculated using the following formula:

$$\text{Predicted Price} = P_{T-1} + (P_{\min_t+1} - P_{\min_t}) \times \text{sign}\left(\mathcal{L}_{\text{original}} - \mathcal{L}_{\min}\right)$$

Where:

- P_{T-1} is the last observed price.
- $P_{\min_t+1}$ and $P_{\min_t}$ are the prices at the most similar historical time point.
- $\mathcal{L}_{\text{original}}$ and $\mathcal{L}_{\min}$ are the original and minimum likelihoods, respectively.

This method leverages historical similarities to predict future price movements, with the assumption that similar conditions will produce similar outcomes.

5 Result

5.1 Evaluation: Mean Absolute Percentage Error (MAPE)

To assess the accuracy of the predictions, we calculated the Mean Absolute Percentage Error (MAPE), a common metric for evaluating forecasting models. MAPE is defined as:

$$\text{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{A_i - P_i}{A_i} \right|$$

Where:

- A_i is the actual price at time i.
- P_i is the predicted price at time i.
- n is the number of predictions.

MAPE provides an intuitive measure of prediction accuracy, expressed as a percentage. It indicates the average percentage error between the predicted and actual values, with lower MAPE values signifying better predictive performance. In this study, the performance of the HMM for predicting the Open, High, Low, and Close prices of AMZN, NVDA, and AAPL is summarized in Table 1. The results demonstrate the model's ability to forecast stock prices with reasonable accuracy, though there is some variability across different stocks and features.

	AMZN	NVDA	AAPL
Open	8.65%	8.67%	8.24%
High	7.26%	8.45%	5.79%
Low	8.04%	8.59%	7.43%
Close	8.83%	9.04%	8.22%

Table 1: MAPE values for Amazon, Nvidia, and Apple.

5.2 Analysis

The results of the predictions for each stock and feature were plotted against the observed prices over the prediction window. Figures 5, 6, and 7 display the results for AMZN, NVDA, and AAPL, respectively. The plots reveal that the HMM model captures the general trends in stock prices reasonably well, though there are instances where the model's predictions deviate from actual prices. This could be attributed to sudden market shifts or events that were not captured in the historical data used for training.

Overall, the HMM model demonstrated a robust capability to predict stock prices, especially in markets with more stable and predictable conditions. However, further refinements, such as incorporating additional features or employing more sophisticated models, could potentially improve the prediction accuracy, particularly in more volatile market environments.

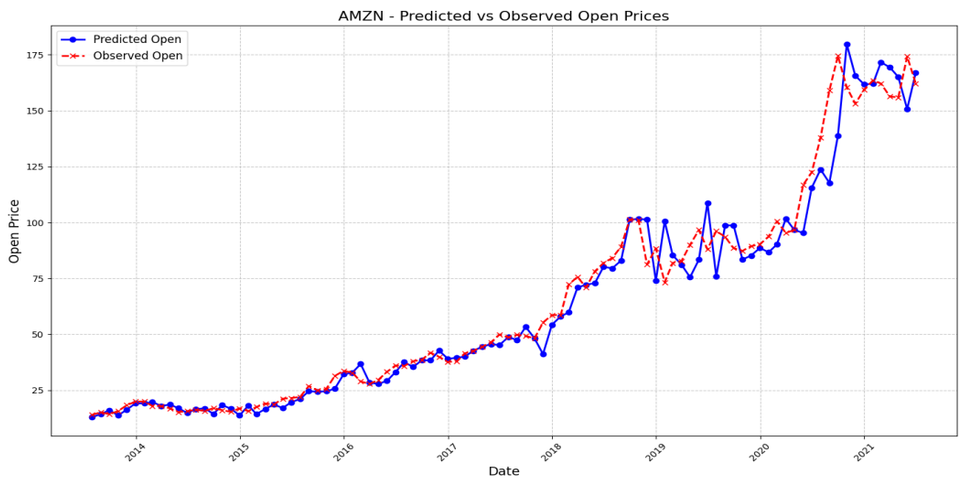

Figure 5: Predicted vs Observed Open Prices for Amazon Inc.

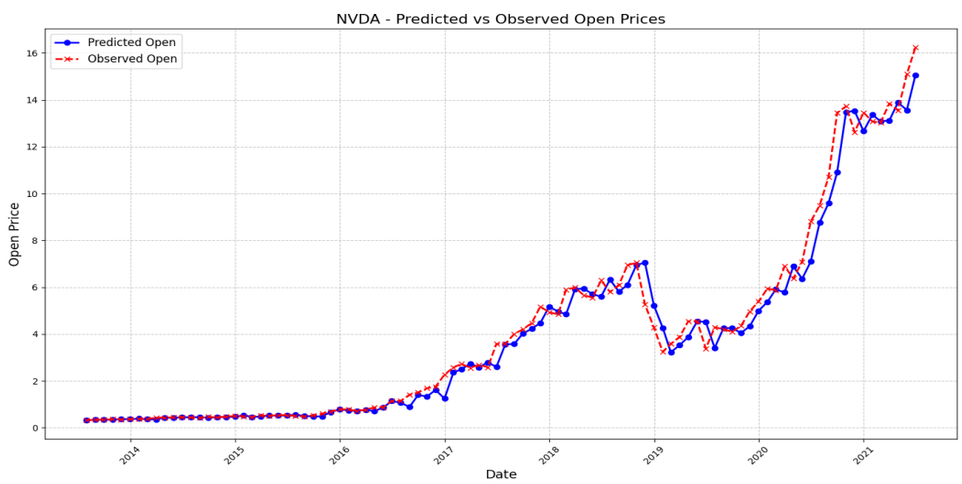

Figure 6: Predicted vs Observed Open Prices for NVIDIA Corporation

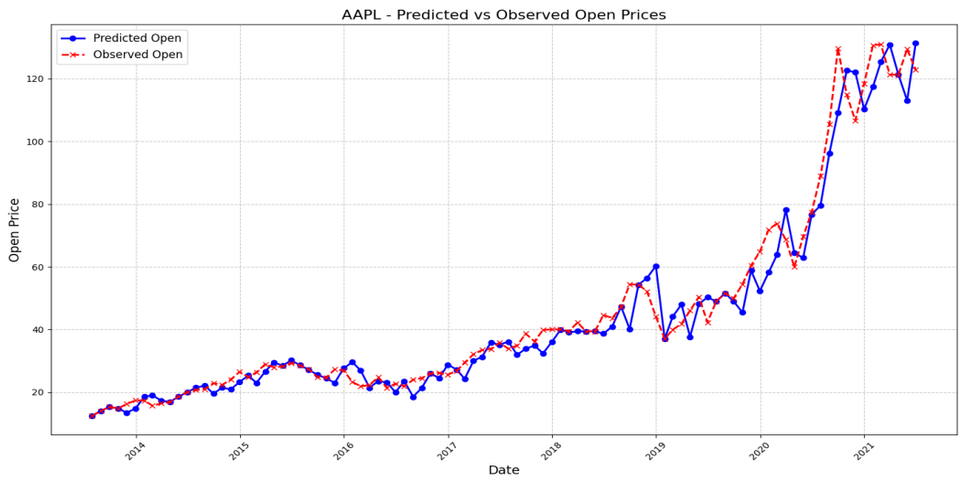

Figure 7: Predicted vs Observed Open Prices for Apple Inc.

6 Limitations

While Hidden Markov Models (HMMs) offer a robust framework for modeling time series data and have shown promise in stock price prediction, there are several limitations to this approach that must be acknowledged.

6.1 Assumption of Markov Property

One of the core assumptions of HMMs is the Markov property, which posits that the future state depends only on the current state and not on the sequence of events that preceded it. This "memoryless" assumption can be overly simplistic for financial markets, where historical data and long-term trends often play a significant role in determining future prices [Shumway and Stoffer(2011)]. This limitation may result in the model missing out on important patterns that extend beyond the immediate past.

6.2 Stationarity Assumption

HMMs implicitly assume that the statistical properties of the underlying process (such as transition probabilities between states) remain constant over time. However, financial markets are often characterized by non-stationarity, with changing dynamics due to evolving economic conditions, technological advancements, and policy changes. This assumption of stationarity may limit the model's ability to adapt to shifts in market regimes [Chang and Hu(2022)].

6.3 Limited Ability to Capture Extreme Events

Financial markets occasionally experience extreme events or "black swans," such as market crashes or booms, which are rare but have a profound impact on prices. HMMs, particularly those with Gaussian emissions, may struggle to capture these outliers or tail events accurately, as they are not designed to model heavy-tailed distributions. This can lead to significant prediction errors during periods of high volatility [Taleb(2008)].

7 Conclusions

In this study, we employed Hidden Markov Models to predict the stock prices of major companies such as Amazon (AMZN), Nvidia (NVDA), and Apple (AAPL). The HMM framework was chosen for its ability to model time series data with hidden states, capturing underlying market conditions that influence stock prices. Through a rigorous process of model training, sliding window techniques, and likelihood-based predictions, we demonstrated the potential of HMMs in forecasting stock prices with reasonable accuracy.

However, it is essential to recognize the limitations inherent in the HMM approach. The assumptions of the Markov property and stationarity and the challenges in capturing extreme market events all pose constraints on the model's predictive capabilities. Additionally, the reliance on historical data and the computational complexity of the model can limit its effectiveness in certain market environments.

Given these limitations, exploring alternative models or hybrid approaches may be beneficial. For instance, combining HMMs with other advanced techniques, such as Long Short-Term Memory (LSTM) networks, can address the issue of capturing long-term dependencies in time series data. LSTMs, a type of recurrent neural network, are particularly well-suited for modeling sequences where long-term contextual information is crucial. Additionally, incorporating regime-switching models, which can better handle the non-stationarity of financial time series by allowing for different market regimes, could improve prediction accuracy and robustness.

Another promising direction is the development of hybrid models that leverage the strengths of multiple approaches. For example, a hybrid model could use HMMs to identify underlying market regimes and LSTMs to predict price movements within each regime. Such models could potentially offer more accurate and resilient predictions by addressing the weaknesses of individual techniques.

In summary, while HMMs are valuable tools for modeling and predicting stock prices, their limitations must be carefully considered when interpreting results. Future research should focus on addressing these limitations through model enhancements or by integrating HMMs with other predictive techniques. By doing so, it may be possible to create more sophisticated models that provide better insights into the complex dynamics of financial markets.

7.1 Future Research Directions

Future research could explore several avenues to enhance the performance of HMMs in stock price prediction. These include:

- **Incorporating more complex emission distributions:** Instead of using Gaussian emissions, future models could explore non-Gaussian or mixture models that better capture the heavy tails and skewness often observed in financial data.
- **Hybrid modeling approaches:** As mentioned, combining HMMs with machine learning models like LSTMs or regime-switching models could provide a more holistic approach to time series forecasting.
- **Dynamic state adaptation:** Developing methods for dynamically adjusting the number of hidden states based on market conditions could improve model flexibility and accuracy.

In conclusion, while HMMs provide a solid foundation for stock price prediction, the incorporation of alternative models and enhancements is critical for advancing the field and improving the reliability of financial forecasting models.

References

[Catello et al.(2023)Catello, Ruggiero, Schiavone, and Valentino] Luigi Catello, Ludovica Ruggiero, Lucia Schiavone, and Mario Valentino. Hidden markov models for stock market prediction, 2023. URL https://arxiv.org/abs/2310.03775.

[Nguyen(2018)] Nguyet Nguyen. Hidden markov model for stock trading. *International Journal of Financial Studies*, 6(2), 2018. ISSN 2227-7072. doi: 10.3390/ijfs6020036. URL https://www.mdpi.com/2227-7072/6/2/36.

[Finance(2024)] Yahoo Finance. Historical data for appl, nvda and amzn, 2024. URL https://ca.finance.yahoo.com/quote/API/.

[Shumway and Stoffer(2011)] Robert Shumway and David Stoffer. *Time Series Analysis and Its Applications With R Examples*, volume 9. 01 2011. ISBN 978-1-4419-7864-6. doi: 10.1007/978-1-4419-7865-3.

[Chang and Hu(2022)] Qingqing Chang and Jincheng Hu. Application of hidden markov model in financial time series data. *Security and Communication Networks*, 2022:1–10, 04 2022. doi: 10.1155/2022/1465216.

[Taleb(2008)] Nassim Nicholas Taleb. *The Black Swan: The Impact of the Highly Improbable*. Random House, London, 1 edition, 2008. ISBN 1400063515.