ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/351869777

Stock Price Prediction using Hidden Markov Models and understanding the nature of underlying Hidden States

Preprint · May 2021

CITATIONS
0

READS
1,799

4 authors, including:

Aman Verma
Birla Institute of Technology and Science Pilani
3 PUBLICATIONS **1** CITATION
SEE PROFILE

Snehil Gupta
Birla Institute of Technology and Science Pilani
2 PUBLICATIONS **1** CITATION
SEE PROFILE

Anirudh Singh Rana
Birla Institute of Technology and Science Pilani
58 PUBLICATIONS **567** CITATIONS
SEE PROFILE

All content following this page was uploaded by Aman Verma on 26 May 2021.

The user has requested enhancement of the downloaded file.

Stock Price Prediction using Hidden Markov Models

Kushank Maheshwari
Maths+CS
BITS PILANI
Pilani, India

Aman Verma
Maths+EEE
BITS PILANI
Pilani, India

Snehil Gupta
Maths+EEE
BITS pilani
Pilani, India

***Abstract*—The stock market is an important indicator that reflects the economic strengths and weaknesses of a country. Stock trading will have great returns if the economy is strongly growing, but it may have negative returns if the economics is depressing. An accurate stock price prediction is one significant key to be successful in stock trading. Although, the Stock markets are one of the most complex systems which are almost impossible to model in terms of dynamical equations. The main reason is that there are several uncertain parameters like economic conditions, the company's policy change, supply and demand between investors, etc. which drive the stock prices. These parameters are constantly varying which makes stock markets very volatile. Prediction of stock prices is a classical problem of non-stationary pattern recognition in Machine Learning and Mathematical Modelling. There has been a lot of research in predicting the behavior of stocks based on their historical performance using Artificial Intelligence and Machine Learning techniques like- Artificial Neural Networks, Fuzzy logic, and LSTM Neural Networks. One of the methods which are not as common as the above mentioned for analyzing the stock markets is Hidden Markov Models. Hence, we will be focusing on Hidden Markov Models in this project and compare its performance with the LSTM Neural Networks method and later we will give a visualisation for possible hidden states.**

***Index Terms*—Hidden Markov Model (HMM), LSTM Neural Networks, Hypothesis Testing, Transition Matrix Estimation.**

I. INTRODUCTION

Stock traders always wish to buy a stock at a low price and sell it at a higher price. However, when the best time is to buy or sell a stock is a challenging question. Stock investments can have a huge return or a big loss due to the high volatilities of stock prices. To tackle this problem, we present a method for predicting future stock prices using the Hidden Markov Model. Hidden Markov Models have a strong probabilistic framework for recognizing patterns in stochastic processes. They have been used for analyzing patterns in speech, handwriting, and gestures and are still extensively used in those areas. Later HMMs found success in analyzing a wide variety of DNA sequences. The observations that we are considering in this project aredaily open, close, high, and low prices for Amazon, Netflix, and Tesla's stock to build a prototype. In this project, we will consider our observations to be distributed as a multivariate Gaussian distribution. Many models were used in the past to predict stock prices such as the "exponential moving average" (EMA) or the "head and shoulders" methods. Recently, researchers have applied the hidden markov model for forecasting stock prices. Hassan and Nath[1] used fixed state HMMs for predicting some airline stocks by looking for a similar pattern in the past data. Nguyet Nguyen[2] extended the work of Hassan and Nath and used the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) to test the performance of the model for several hidden states and predict prices using base off historical stock prices and the chances of the likelihoods of the model of the most recent data set with the historical data set. We have implemented an HMM model similar to that implemented by Nguyet Nguyen. Our approach is different from their work in two modifications. The first difference is that we compare our predictions obtained by the HMM model with a LSTM model for performance measurement. The second difference is that we try to visualize the Hidden States of the model using Transition probability matrix estimation and Hypothesis testing.

II. INTRODUCTION TO HIDDEN MARKOV MODEL

A. Hidden Markov Model

The Hidden Markov model is a strong probabilistic framework or a Generative Model for recognizing patterns in Stochastic Processes. The idea behind using the hidden Markov model is that the likelihood of the observations depends on the states of the system which are 'hidden' to the observer. While the transition from one state to another is a Markov process with certain transition probability, the next state depends only on the present state, hence the name Hidden Markov model. States in a Hidden Markov Model are always discrete while the observations can either be discrete or continuous or both. For Ex. In the Stock Market, an investor can only observe the stock prices and the underlying states which are driving the stock prices are unknown. In general terms, HMM is a tool for analyzing non-stationary systems. The probability of observations given a state is determined by the emission probability. An HMM is specified by the components mentioned in Figure A.

Figure A: HMM Parameters

Q = q$_1 q_2 q_3 ... q_N$	A set of N states
A=a$_{11}...a_{ij}...a_{NN}$	A transition probability matrix A, each a$_{ij}$ representing the probability moving from state i to state j, s.t. $\sum_{j=1}^{N} a_{ij} = 1 \forall i$
O = o$_1 o_2 ... o_T$	A sequence of T observations, each one drawn from a vocabulary V = v$_1, v_2, ..., v_v$
B = b$_i(o_t)$	A sequence of observation likelihoods, also called emission probabilities each expressing the probability of an observation o$_t$ being generated from a state
π =multiplication of π_j	π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0. Also, \sum_{j=1}^{N} \pi_{ij} = 1$

B. Modelling Stock Markets as a Hidden Markov Model

It is observed that there are underlying 'Hidden' states that drive the stock prices in the stock market, to identify them we use a Hidden Markov Model where:
· These underlying 'Hidden states' are the states of our Hidden Markov Model.
Observations of the model are Stock prices that can be observed by the investor, they are continuous vectors comprising independent observations - Closing, Opening, High, and Low prices.
· State and Transition probabilities are unknown, though their optimal values are later found using Baum-Welch Algorithm.
· Probabilities of an observation given a state are modelled through a PDF and assumed to be from multivariate gaussian distribution.[Thus $b_i(O_t) = N(\mu_i, \sigma_i)$, where μ_i and σ_i are the mean and variance of the distribution corresponding to the state S_i]
Following are the notations used to define the Terminologies that are used to model Stock markets as Hidden Markov Model:
· Number of Transitions, T
· Latency, K
· Number of states, N ($S_t = (s_1, s_2, ..., s_N)$)
· Observation Sequence, O
· Initial state Probability, P_0
· State Transition Matrix, $A = [a_ij]$ where a_ij is the state transition probability from state s_i to s_j
· Observation Probability $\mu_i \sum_i i = 1, 2, ..., N$ where $\mu_i \sum_i$ are the mean and covariance matrix for Gaussian distribution for state i.
The Hidden Markov Model can be represented as $\lambda = (A, \mu, \sum, P_0)$.
Now, to use any Hidden Markov Model we generally have to answer the following 3 problems:
· (Problem1: Likelihood) Given the observation data and the model parameters, how to compute the observation likelihood?
· (Problem2: Decoding) Given the observation data and the model parameters, how to find the best corresponding hidden state sequence?
· (Problem3: Learning) Given the observation and set off in the HMM what are optimal HMM parameters?
To answer the first problem we use Forward Algorithm[3], the second problem can be solved using the Viterbi Algorithm[4] and for the third problem we use the Baum-Welch Algorithm[5].

1) Likelihood Computation: The Forward Algorithm: The forward algorithm computes the observation probability by summing over the probabilities of all possible hidden state paths that could generate the observation sequence, but it does so efficiently using Dynamic Programming. We define a helper joint probability function(Forward path probability) : $\alpha_t(j)$ which represents the probability of being in state j after seeing the first t observations, given the automaton λ.

$$\alpha_t(j) = P(o_1, o_2, ..., o_t, q_t = j|\lambda) \tag{1}$$

The Pseudo code 1 describes the algorithm.

1) **Initialization**:

$$\alpha_1(j) = \pi_j b_j(o_1) 1 \leq j \leq N \tag{2}$$

2) **Recursion**:

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t) 1 \leq j \leq N, 1 \leq t \leq T \tag{3}$$

3) **Termination**:

$$P(O|\lambda) = \sum_{j=1}^{N} \alpha_T(i) \tag{4}$$

Pseudo code 1: *Forward Algorithm*

2) HMM Training:The Baum-Welch Algorithm: Baum-Welch algorithm is an algorithm to calibrate parameters for the HMM given the observation data and the set of possible states in a HMM. Thus the algorithm allows us to train both the transition probabilities A and the emission probabilities B of the HMM. It is an E-M iterative algorithm, computing an initial estimate for the probabilities, then using those estimates to compute a better estimate, and so on, iteratively improving the probabilities that it learns. We define a helper joint probability function(Backward Path Probability): $\beta_t(i)$ is the probability of seeing the observations from time $t + 1$ to the end, given that we are in state i at time t (and given the automaton λ)

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, ..., o_T|q_t = i, \lambda) \tag{5}$$

The Pseudo code 2 describes the backward algorithm.

1) **Initialization**:

$$\beta_T(i) = 1, 1 \leq i \leq N \tag{6}$$

2) **Recursion**:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), 1 \leq i \leq N, 1 \leq t \leq T \tag{7}$$

3) **Termination**:

$$P(O|\lambda) = \sum_{j=1}^{N} \pi_j b_j(o_1) \beta_1(j) \tag{8}$$

Pseudo code 2: *Backward Algorithm*

Once we have forward and backward probabilities we can use them to compute estimates of transition probability $a_{i,j}$ and observation probability $b_i(o_t)$ from an observation sequence. We estimate $\hat{a}_{i,j}$ by a variant of simple maximum likelihood estimation:

$$\hat{a}_{i,j} = \frac{expected number of transitions from state i to state j}{expected number of transitions from state i} \tag{9}$$

Define, ξ_t as the probability of being in state i at time t and state j at time t+1 , given the observation sequence and of course the model:

$$\xi_t = P(q_t = i, q_{t+1} = j|O, \lambda) \tag{10}$$

To compute ξ_t from not-quite-ξ_t , we follow the laws of probability and divide by $P(O|\lambda)$, since

$$P(X|Y, Z) = \frac{P(X, Y|Z)}{P(Y|Z)} \tag{11}$$

Also since the probability of the observation given the model is simply the forward probability of the whole utterance (or alternatively, the backward probability of the whole utterance), therefore

$$P(O|\lambda) = \sum_{j=1}^{N} \alpha_t(j) \beta_t(j) \tag{12}$$

So, the final equation for ξ_t is:

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{j=1}^{N} \alpha_t(j) \beta_t(j)} \tag{13}$$

Thus,

$$\hat{a}_{i,j} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \sum_{k=1}^{N} \xi_t(i, k)} \tag{14}$$

Similarly, we estimate observation probability $\hat{b}_j(v_k)$[Probability of a given symbol v_k from the observation vocabulary V, given a state j] using:

$$\hat{b}_j(v_k) = \frac{expected\ no.\ of\ times\ in\ state\ j\ and\ symbol\ v_k}{expected\ number\ of\ times\ in\ state\ j} \tag{15}$$

Define, $\gamma_t(j)$ which is the probability of being in state j at time t as: $\gamma_t(j) = P(q_t = j|O, \lambda)$ Once again, we will compute this by including the observation sequence in the probability:

$$\gamma_t(j) = \frac{P(q_t = j, O|\lambda)}{P(O|\lambda)} \tag{16}$$

$$\gamma_t(j) = \frac{\alpha_t(j) \beta_t(j)}{P(O|\lambda)} \tag{17}$$

To compute b . For the numerator, we sum $\gamma_t(j)$ for all time steps t in which the observation o_t is the symbol v_k that we are interested in. For the denominator, we sum $\gamma_t(j)$ over all time steps t , therefore:

$$\hat{b}_j(v_k) = \frac{\sum_{t=1}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \tag{18}$$

The Algorithm 1 summarises the algorithm.

```
Algorithm 1: Baum-Welch Algorithm
initialize A and B
  iterate until convergence
  E Step
      $\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{\alpha_T(q_F)} \forall\ t\ and\ j$   (19)
      $\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\alpha_T(q_F)} \forall\ t, i\ and\ j$   (20)
  M Step
      $\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\sum_{k=1}^{N}\xi_t(i,k)}$   (21)
      $\hat{b}_j(v_k) = \frac{\sum_{t=1 s.t. O_t=v_k}^{T}\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)}$   (22)
```

Note: We have proved the Forward Algorithm and Baum-Welch algorithm for Single element observation space, the same can be generalised to multiple element observation space also.
For us to model stock markets as HMM we required only the Forward algorithm and the Baum-Welch algorithm as we were more interested in the price prediction, rather than finding the optimal sequence of hidden states affecting the prices. We used hmmlearn , an open source python library:
· To train the model i.e implement the baum-welch algorithm using hmmlearn.model.fit($Observationsequence$, $Numberofhiddenstates$).
· To calculate the likelihood of the observations i.e implement the forward algorithm using $hmmlearn.model.score(Observation_sequence)$.

III. Model Description

The Hidden Markov Model has been widely used in financial mathematics to predict stock prices. In this section we first describe how to use the Akaike information criterion(AIC) and the Bayesian information criterion(BIC) to test the HMM's performances with different numbers of states. Later we will then present how to use HMM to predict stock prices.

A. Corpus of Data Used

We choose three stocks that are actively trading in the stock market to examine our model: Amazon Inc. (AMZN), Netflix Inc. (NFLX), and Tesla Inc. (TSLA). These stocks were chosen with a viewpoint that they had some recent developments after the CoronaVirus pandemic, where Amazon suffered in its prices, whereas Netflix saw an increase in their stock prices and Tesla, due to their SpaceX programme was performing well during Mid months of 2020. This helped in catching some diverse and irregular variations in prices for our testing data, which helped us to better analyse and understand the behaviour of HMM. The daily stock prices (open, low, high, close) of these stocks and information of these companies can be found from finance.yahoo.com. We use daily historical prices of these stocks from June 01, 2010 to June 01, 2020 .

B. Selection of Optimal Number of Hidden States

Choosing the number of hidden states for the HMM is a crucial task. We use two common criteria: the AIC and the BIC to evaluate the performances of HMM with different numbers of states. In order to select a model with optimal number of states, we train a set of models by varying the number of states (N) from the state space G. We have considered the values of a number of states ranging from [2,15]. We then calculate these performance measure metrics for each of the models and choose the model which has the lowest value. The AIC and BIC are calculated using the following formulas respectively:
· $AIC = -2log(P(O_{train|\lambda}) - 2p$
· $BIC = -2log(P(O_{train|\lambda}) - 2log(T)$
$where, p = N^2 + 2N - 1$
In our project, we have used BIC as the model's performance measure to select the number of states in our target model.

C. Prediction of Stock Prices

In this section, we will use HMM to predict stock prices and compare the predictions with the real stock prices. We will forecast stock prices of Amazon, Netflix and Tesla using HMM with an optimal number of hidden states. First we introduce how to predict stock prices using HMM. The prediction process can be divided into three steps:
· Step 1: Calibrate HMM's parameters and calculate likelihood of the model for K previous observations.
· Step 2: Find a day in the past that has a similar likelihood for K previous observations to that of the recent day.
· Step 3: Use the difference of stock prices on the similar day in the past to predict future stock prices.
The main idea for predicting the next day's stock price is to calculate the log-likelihood of K previous observations and comparing it with the log-likelihood of all the previous sub-sequences of the same size by shifting the window by one day in the direction of past data. We then identify a day in the past whose log-likelihood of its K previous observations is the closest to the sub-sequence whose next day's price is to be predicted.

$j = argmin_i(|P(O_t, O_{t-1}, O_{t-2}, ..., O_{t-K}|\lambda)$
$- P(O_{t-i}, O_{t-i-1}, O_{t-i-2}, ..., O_{t-i-K}|\lambda)|)$
$where, i = 1, 2, ..., T/K$

We then calculate the differential price change from the identified day to its next day. This change is then added to the current day's price to get our next day's prediction.$O_{t+1} = O_t + (O_{t-j+1} - O_{t-j})$. After the prediction of the stock's close price for the time t+1 we update observation data O by moving it forward one day, to predict stock price for the day t+2. Day. The calibrated HMM's parameters in the previous prediction were used as the initial parameters for the next prediction. We repeat the three step prediction process for the next prediction and so on.

IV. Visualising Hidden States in HMM

After doing a Literature survey for Financial Markets we inferred that there's a possibility for Stock market to be considered as a Hidden Markov Model with broadly 3 hidden states:
· Bull (UP): Periods of time where prices generally are rising, due to the actors having optimistic hopes of the future.
· Bear (Down): Periods of time where prices generally are declining, due to the actors having a pessimistic view of the future.
· Stagnant: Periods of time where the market is characterized by neither a decline nor a rise in general prices.
To further strengthen our inference we estimated the Transition Matrix for a Markov chain with these states and then did a hypothesis testing for Markov Chains.

A. Estimation of Transition Matrix

For estimating the transition matrix we initially assumed the Stock Market to form a Markov Chain with 3 states as bull, bear and stagnant, thus forming a Transition matrix as shown in Table I,

Today/Tomorrow	Bear	Bull	Stagnant
Bear			
Bull			
Stagnant			

TABLE I
Transition Matrix

We form Transition matrix by taking the data and then using a variant of maximum likelihood estimation as present in [6]. The same is given by:

$$\hat{a}_{i,j} = \frac{expected\, number\, of\, transitions\, from\, state\, i\, to\, state\, j}{expected\, number\, of\, transitions\, from\, state\, i} \quad (23)$$

For this we were required to define how transitions were taking across different states. To tackle this problem we formulate 4 different models by varying how transitions are taking across different states.

B. Formulation of Models for transition matrix

To estimate the transition matrix, we have formulated 4 different models where we vary how transitions are taking place across different states. We have formulated the 4 different models based upon differences observed in successive days of the following two aspects:

1. Difference Pattern i.e. what parameters are considered to evaluate the difference parameter.
2. Difference Range i.e. what are the range of values corresponding to which the states: ‘Up’, ‘Stagnant’ and ‘Down’.
For the difference pattern, we have defined it using Open price and Close price in two different manners:
1. Difference for n^{th} day = Close Price for n^{th} day - Close Price for $(n-1)^{th}$ day
2. Difference for n^{th} day = Close Price for n^{th} day - Open Price for n^{th} day
For the difference range, we have modelled it in two ways:
1. Static: A fixed value to classify the difference values. This fixed value is fixed for each day of our data and was taken as 1.5 times the mean of difference value.
·a. Down: [Minimum Value of Difference, - Fixed Value]
·b. Stagnant: [- Fixed Value, Fixed Value]
·c. Up: [Fixed Value, Maximum Value of Difference]
2. Dynamic: A dynamic value to classify the difference values. This value changes for every data element and was formulated as:
$DynamicValue = |Change| * (OpenPrice)/2$ Where, $Change = (DifferenceValue)/(OpenPrice)$
·a. Down: [Minimum Value of Difference, -Dynamic Value]
·b. Stagnant: [- Dynamic Value, Dynamic Value]
·c. Up: [Dynamic Value, Maximum Value of Difference]

Thus the 4 models are structured as follows:
·Model 1:
Difference Pattern: Close Price for n^{th} day - Close Price for$(n-1)^{th}$ day
Difference Range: Static Method
·Model 2:
Difference Pattern: Close Price for n^{th} day - Open Price for n^{th} day
Difference Range: Static Method
·Model 3:
Difference Pattern: Close Price for n^{th} day - Close Price for $(n-1)^{th}$ day
Difference Range: Dynamic Method
·Model 4:
Difference Pattern: Close Price for n^{th} day - Open Price for n^{th} day
Difference Range: Dynamic Method

After formulation of models, we now see the hypothesis testing results on these models on different timeframes, and further analyse the results. If the probability is high for a particular model across different timeframes, we can infer that the parameters of that particular model show some dependency for HMM hidden states.

C. Hypothesis Testing

To verify and justify our assumption for the stock market forming a Markov chain with the Up, Down and Stagnate as state space, we perform Hypothesis testing on the 4 formulated models for Markov chain of order 1 for two different timeframes.
The reason we chose two different timeframes was to observe the behaviour of markov chains across shorter and longer time-intervals.

1) Basics of Hypothesis Testing: Hypothesis testing is used to assess the plausibility of a hypothesis by using sample data. The test provides evidence concerning the plausibility of the hypothesis, given the data. Statistical analysts test a hypothesis by measuring and examining a random sample of the population being analyzed. Here, the null hypothesis represents the realisation which we have observed from Markov Chain with a given transition. Under the null hypothesis, the statistic:

$$-2log\lambda = 2\sum_{j}\sum_{k} n_{jk}log\frac{n_{jk}}{n_{j}.p^{0}_{jk}} \tag{24}$$

has an asymptotic χ^2 distribution with m(m-1) degree of freedom, where lambda denotes likelihood ratio criteria for the null hypothesis and n_{jk} denotes the number of transitions from state j to state k as mentioned in [7]. After the calculation of the -2log(lambda), if the chi-square probability is very small the null hypothesis is rejected. Above mentioned testing will be used to test whether our transition matrix follows the first-order Markov chain .

2) Implementing Hypothesis Testing: The Hypothesis testing has been applied to the derived models to check the probability of them to follow a markovian pattern. The null hypothesis is that the observed realization comes from a Markov chain of order one. The test is applied to the matrix obtained in a time frame(Jan-June, March-August, May-October, 2010-2020, 2005-2015, 2000-2010) of a model which is also referred to as A-matrix. As ((-2)*log(lambda)) of the model, where lambda represents the likelihood ratio criteria, follows Chi-Square distribution. The following Table II shows the probability for each of the matrices using Chi-Squared Distribution. The likelihood ratio criteria are calculated using the equation given in the previous section for each matrix.

Model No./Time Range	January-June	March-August	May-October	2010-2020	2005-2015	2000-2010
1.	0.99802	0.99517	0.63591	0.99998	0.99941	0.99953
2.	0.34125	0.86178	0.48918	0.99997	0.99928	0.99928
3.	0.93563	0.97830	0.97830	0.99790	0.99854	0.99806
4.	0.41721	0.92237	0.47101	0.99849	0.99816	0.99732

TABLE II
RESULTS OF HYPOTHESIS TESTING

As mentioned in the previous section, if the calculated probability of a matrix is small then the assumed null hypothesis is rejected for that particular matrix of the model in the time frame.

3) Conclusions of Hypothesis Testing: The above results in Table 1 can be better explained by bifurcating them on the basis of duration of time frames and then later comparing

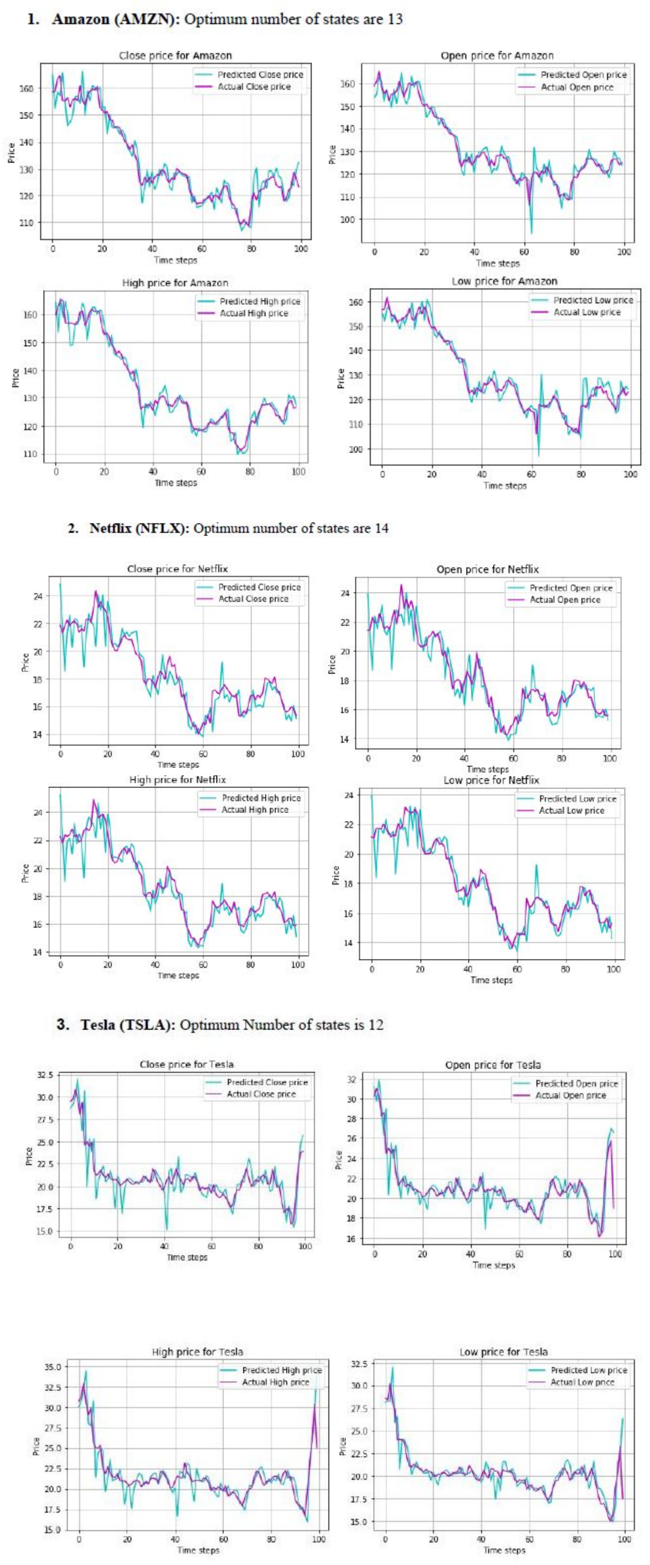

Fig. 1. HMM results in comparison to real data

them. First, we'll consider the 6 month timeframe predictions: We observe that some models are following the stated null hypothesis while the hypothesis is being rejected for others. Hence, the high variability of probability for different models over a 6 month timeframe does not give us the indication that in general the Stock markets follows a Markovian Chain for shorter time-intervals.

Now, we consider the results obtained from the 10 year timeframe prediction: We can observe that all the probabilities, irrespective of models yield a significantly higher result of around 99% . From this, we can conclude that for significantly longer durations of time interval, Stock markets in general follow a Markov chain. Therefore, we'll consider a 10 year time interval for predicting the future prices using Hidden Markovian Models. It can also be said here that, we get consistent probabilities in Model 3, all above 90%, across both shorter and longer duration of time stating that the Successive day difference of closing price for Difference pattern and dynamic behaviour of Difference range gives the best results and can be expected to be determining factors for transition across Up, Down and Stagnant state spaces. Also it provides a justification for considering the Up, Down and Stagnate State spaces as a possibility for Hidden States of a Stock Model modelled with HMM. Thus justifying our assumption that there are indeed hidden states that drive the Stock prices in a Stock market.

V. Implementation

In this project, the main objective was to determine the efficiency of HMMs in predicting the stock prices. We used hmmlearn , an open source python library to train the model and calculate the likelihood of the observations. The stocks that we selected are of Amazon Inc., Netflix Inc., and Tesla Inc. We used opening price, closing price, high and low as features for the past 2520 working days (approximately 10 years) when the market was open. We kept aside the recent 100 observations for testing and used the rest of the observations for training the model. We predicted the prices for the past 100 days, starting from the 100th day and then using its true observation to retune the model for predicting the 99th day and so on. Thus, every time we retune the model, the number of training samples will increase by one. We implemented HMM using an optimal number of hidden states that were calculated using BIC score. Figures given below show the stock price predictions for Amazon, Netflix and Tesla using HMM with corresponding optimal hidden states respectively along with their Optimum Number of states.

The Following plots correspond to Actual and predicted observational prices of respective stocks for 100 days (From 22 nd February 2020 to 01 st June 2020). It can be said after observing these plots that HMM is a fair model to predict the prices of stocks. We now consider another method called Long Term Short Memory(LSTM) which is a method from Recurrent Neural Networks to model the stocks and then compare them with our HMM predictions.

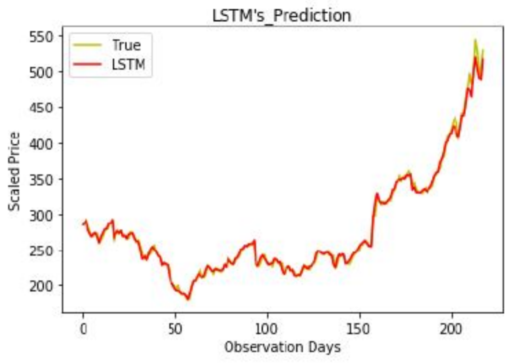

Fig. 2. LSTM Prediction Comparsion with real data

VI. Prediction of Stock Prices using Long Short Term Memory (LSTM)

The research and methods carried out so far yielded a near accurate result. In this section, we want to talk about yet another method of predicting stock prices using the concept of recurrent neural networks from machine learning. We use the LSTM (Long Short Term Memory) Recurrent Neural Network[8] to learn from historic price data and then predict future prices. Such networks have a short term memory capability and thus are able to learn from the past. The earlier used dataset of Amazon, Netflix and Tesla with the same timeframe of 2010-2020 will be used as a source of information for the network. Upon this dataset the model will be trained, evaluated and will attempt to predict whether the price of a particular stock will go up or not in the next 200 days. The objective is to compare the results obtained from state of the art techniques such as LSTM Neural Networks and then compare it with the predictions made by our HMM model. Steps Followed for the formulation of the LSTM Model:
1. Importing and Preprocessing the Data
2. Normalising (Transform features by scaling them to a given range) the Data
3. Splitting the entire dataframe into training, testing and validation dataset
4. Training the LSTM Model
5. Evaluating the predictions of the model.

The above plot shows the prediction of the LSTM model for the opening price of Tesla stock. The red line shows the LSTM prediction whereas the yellow one corresponds to the original values. It can be seen that the LSTM is able to predict very accurately, almost overlapping the original values. The results predicted by LSTM are much more accurate than the one predicted by HMM. Although HMM also gave a pretty-decent prediction. It was observed that predictions made using the LSTM-NN model was not found to be affected by drastic fluctuations in the stock price. However, the model implemented using HMM was found to be highly sensitive to the fluctuations in stock price.

VII. Conclusion

In this study, we have attempted to predict stock prices and understand the behaviour of the stock market when perceived in the form of a hidden markov chain. We also tried to visualize the Hidden States of the Stock market modelled through HMM using Transition probability matrix estimation and Hypothesis testing. We then compared our predictions obtained by the HMM model with a LSTM model for performance measurement. We had taken an initial assumption before applying the HMM model on the stock model that there exists hidden states driving stock prices, which we verified later during hypothesis testing. The testing also helped us to observe that stock market data with much longer duration is more likely to follow markovian properties than the short duration one. Therefore, we chose the longer duration for HMM training. Though in general, the observations will be greatly affected by the choice of the model i.e. the number of states in Hidden Markov Models, it did not make a significant difference when we tried to find the optimal states using BIC to find the best model. Both HMM and LSTM-NN give similar accuracy when the next one day prices are predicted. LSTM-NN captures the volatility of the stock prices better whereas HMM gives good predictions for stable markets. Therefore, LSTM-NN can work better for the stocks with high volatility and HMM can work better for stocks which are more stable. It is impossible to get a model that can 100% predict the price without any error, there are too many factors that can affect the stock prices. So, we cannot hope for a perfect model, but it is observed that the general trend of predicted price both with HMM and LSTM-NN is in line with the actual data, so the trader could have an indicator for reference, and use them to make wise-decisions

References

[1] Hassan, Md. R.; Nath, B.; Stock Market Forecasting Using Hidden Markov Models: A New approach. Proceeding of IEEE fifth International Conference on Intelligent Systems Design and Applications 2005.
[2] Nguyet Nguyen, Stock Price Prediction using Hidden Markov Model, Youngstown State University, Ohio, USA
[3] Baum, L. E.; Egon, J. A.; An inequality with applications to statistical estimation for probabilistic functions of Markov process and to a model for ecnogy. Bull. Amer. Meteorol. Soc 1967, 73, 360-363.
[4] Forney, G. D.; The Viterbi algorithm. IEEE 1973, 61, 268-278.
[5] Rabiner, L. R.; A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. IEEE 1989, 77.2, 257 - 286
[6] David Schön Myers, Lisa Wallin, Petter Wikström: An introduction to Markov chains and their applications within finance.
[7] Medhi, J.(2019). Stochastic Process (4th edition). Paperback
[8] Raghav Nandakumar, Uttam Raj K R, Vishal R, Y V Lokeswari:Stock Price Prediction Using Long Short Term Memory

View publication stats