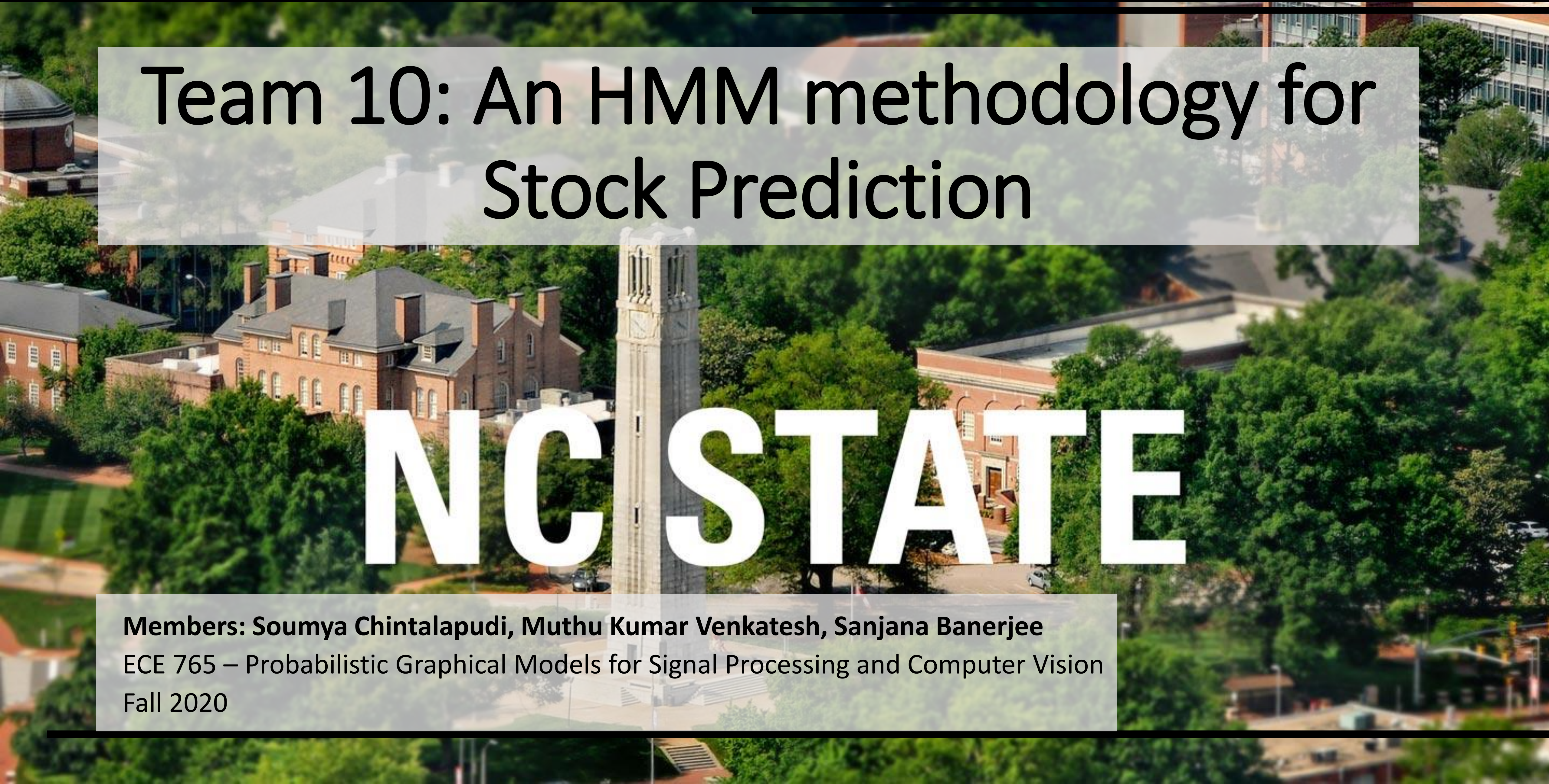
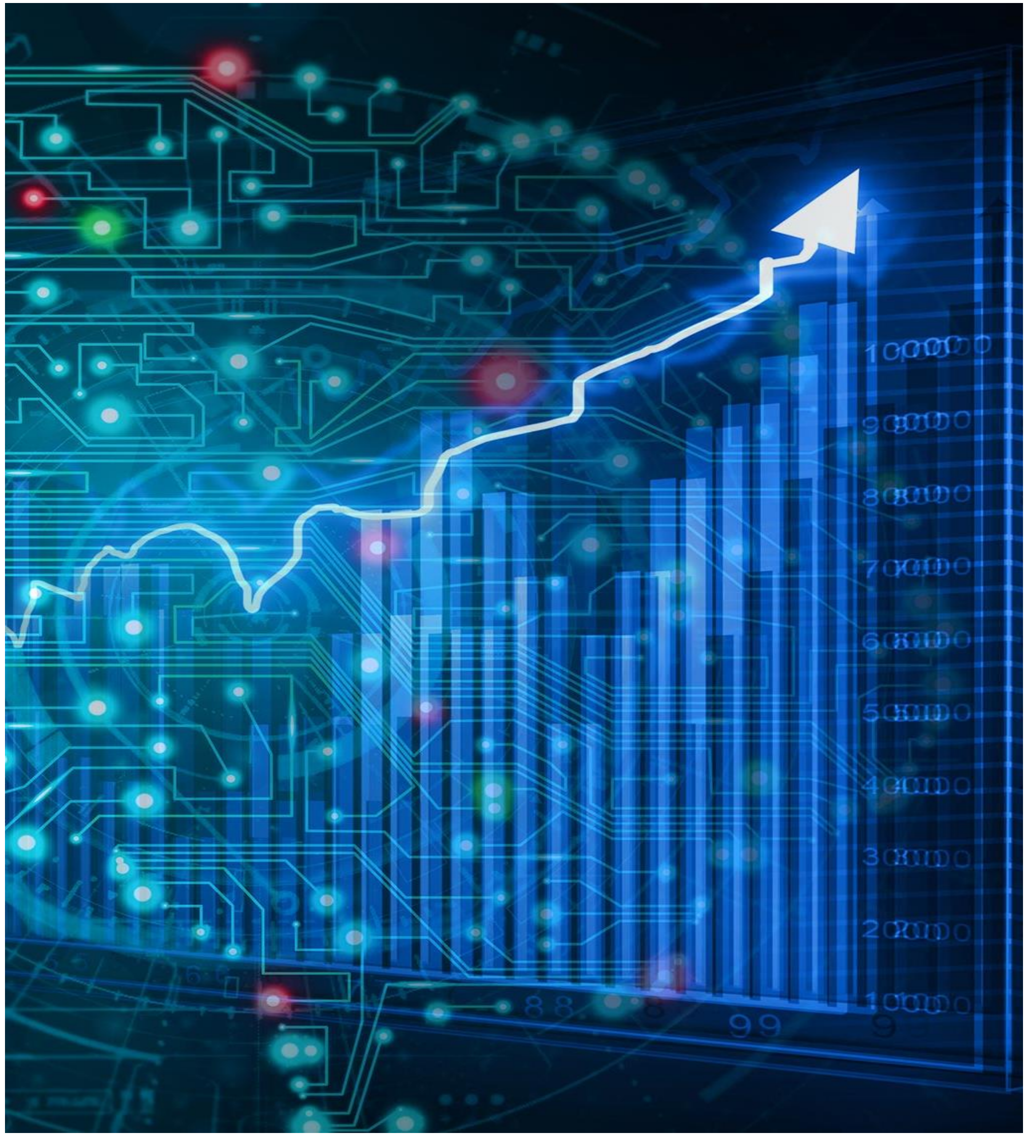
Team 10: An HMM methodology for Stock Prediction



Members: Soumya Chintalapudi, Muthu Kumar Venkatesh, Sanjana Banerjee

ECE 765 – Probabilistic Graphical Models for Signal Processing and Computer Vision

Fall 2020



Problem Statement

Understand Stock Prediction as a classical problem of non-stationary pattern recognition in machine learning.

Build a probabilistic model to extract important features of stocks from their historical prices to predict future trends.

Proposed Solution

HMM	Use Hidden Markov Model as a Probabilistic Model
Train	Train the model on Huge Stock Market Dataset
Form	Form a generative model
Optimize	Tune Hyperparameters, optimize model
Predict	Give predictions for next day's stock prices in terms of Open, Close, High and Low

Data

- Huge Stock Market Dataset is used for the project.
- It contains the full historical daily price and volume data for all US-based stocks and ETFs trading on the NYSE, NASDAQ, and NYSE MKT.
- The data (last updated 11/10/2017) is presented in CSV format as follows: Date, Open, High, Low, Close, Volume, OpenInt.
- We only make use of the Open, High, Low and Close labels.
- The model was trained on 4521 datapoints, which was split as 4421 in training set and 100 in test/prediction set.

```
Date,Open,High,Low,Close,Volume,OpenInt
2005-02-25,16.951,17.067,16.951,17.067,7003,0
2005-02-28,17.027,17.027,16.951,16.974,9145,0
2005-03-01,17.036,17.044,17.013,17.044,16347,0
2005-03-02,16.867,16.982,16.859,16.951,72413,0
2005-03-03,17.027,17.027,17.027,17.027,389,0
```

Reference: https://www.kaggle.com/borismarjanovic/price-volume-data-for-all-us-stocks-etfs

Hidden Markov Models

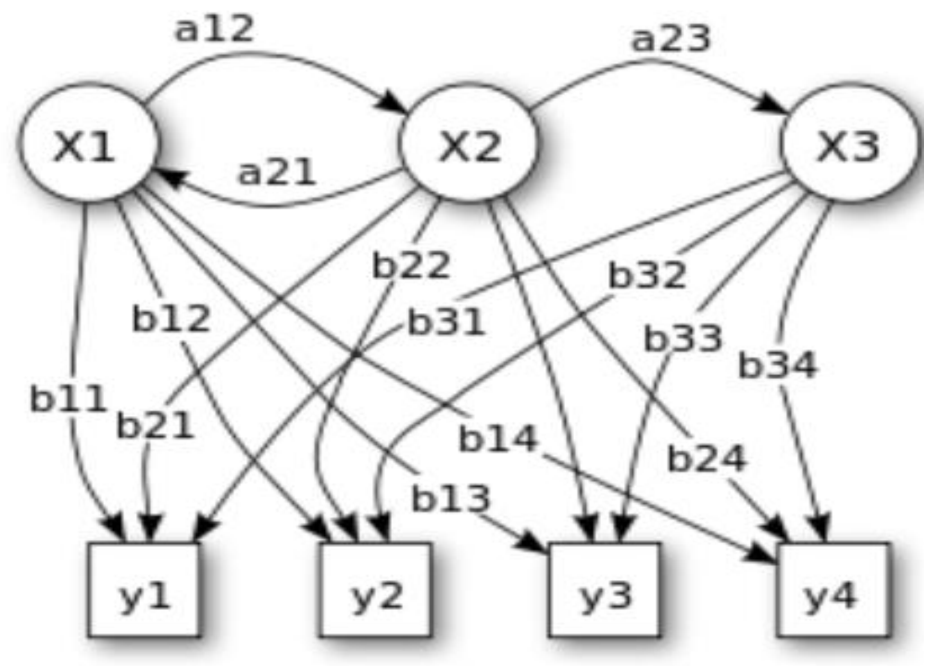


Figure 1. Probabilistic parameters of a hidden Markov model (example)
X — states
y — possible observations
a — state transition probabilities
b — output probabilities

- Strong probabilistic framework (Generative Model) for recognizing patterns in Stochastic Processes.
- Idea behind HMM- The likelihood of the observations depend on the states which are ‘hidden’ to the observer.
- The states keep changing as Markov Process with certain transition probability.
- The observations can be discrete or continuous, but the states are always discrete.
- The probability of observations given a state are determined by the emission probability.
- Emission probabilities can be PDF or PMF

Image Reference: https://www.google.com/url?sa=i&url=https%3A%2F%2Fmedium.com%2F%40postsanjay%2Fhidden-markov-models-simplified-c3f58728caab&psig=AOvVaw1ZlCXp0UlSt3E4MUBVY_WH&ust=1605376266537000&source=images&cd=vfe&ved=0CA0QjhxqFwoTCMDDv5WLgO0CFQAAAAAdAAAAABAx

Stocks as Hidden Markov Models

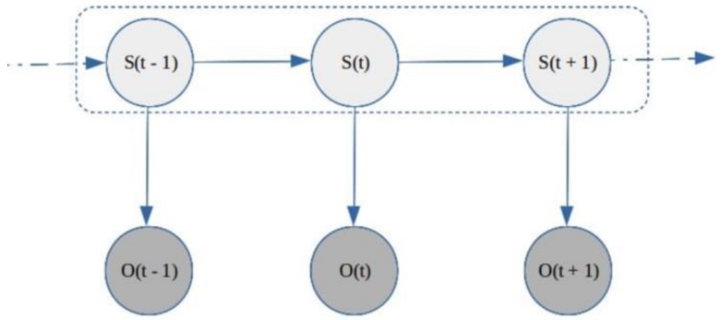
Underlying ‘hidden’ states drive the stock prices.

Investor can observe only the stock prices.

States and the transition probabilities are unknown.

Observations are in the form of a continuous vector- Open Price, Closing Price, High and Low.

Emission probabilities are PDF and assumed to be multivariate Gaussian distribution.



O_t= Observation on day t (daily close, daily open, daily high, daily low)
S_t= State on the day t

Questions to be answered for building HMM

Given the model, how likely it is to observe the given sequence of data?

FORWARD ALGORITHM

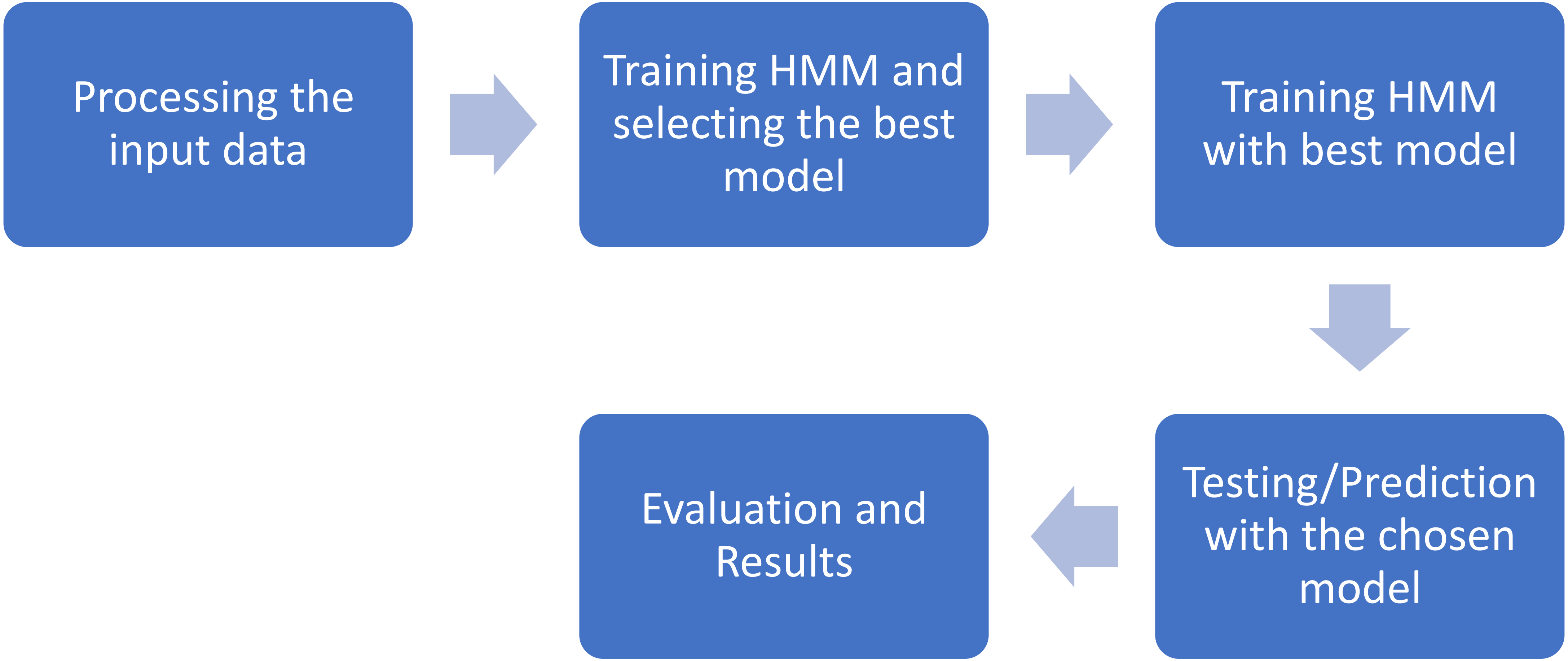
Given the model and observations, what is the best hidden state sequence?

VITERBI ALGORITHM

Given the observations, what are the optimal model parameters?

BAUM-WELCH ALGORITHM

Implementation



Input

- The input to the model is data of size NX4
- N – the training/test data size
- 4 columns corresponding to open, close, high, low values
- The data in decreasing order of time/date

```
[66.81  67.35  67.58  66.7  ]
[67.47  67.92  67.98  66.91 ]
[68.11  68.1   68.33  67.771]
[68.25  68.32  68.64  68.04 ]
[68.22  68.22  68.45  68.22 ]
```

Selecting the optimal model

- Obtain HMM for different number of states

```
model = hmm.GaussianHMM(n_components=states, covariance_type='full', tol=0.0001, n_iter=NUM_ITERS)
```

- Estimate the model parameters for training data

```
model.fit(dataset[NUM_TEST:,:])
```

- Optimal number of states(N=14) is chosen based on Bayesian Information Criterion(BIC)

BIC=$-2\log\left(P\left(O_{train/\lambda}\right)\right) - p.\log(T)$

(Where, p = number of model parameters, T = training data size, model = λ)

Prediction

- For Testing the model, we aim to predict the stock for next 100 days(data points)

Total data-	4521

Separation-	Test-100	Train-4421

- For prediction of first day, consider 100th sample as test and all the remaining data to obtain the model.

- Consider windows of size K=200 to obtain the HMM

Window1-	1:200	Remaining training

- Calculate its log likelihood given the model

Prediction

- We slide the window for training data to obtain multiple training dataset

Window 2-	2:201	Remaining training
Window3-	3:202	Remaining training

……and so on

- We calculate the log likelihood for each window

- Identify a subsequence with log likelihood closest to the first window

Prediction

- Calculate the differential price change between the latest day of the chosen window and its next day's price and add it 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 each prediction is made, it is included as part of train data for the subsequent predictions

- E.g. After 1st prediction

Test data=99	Train data=Remaining

Evaluation of model

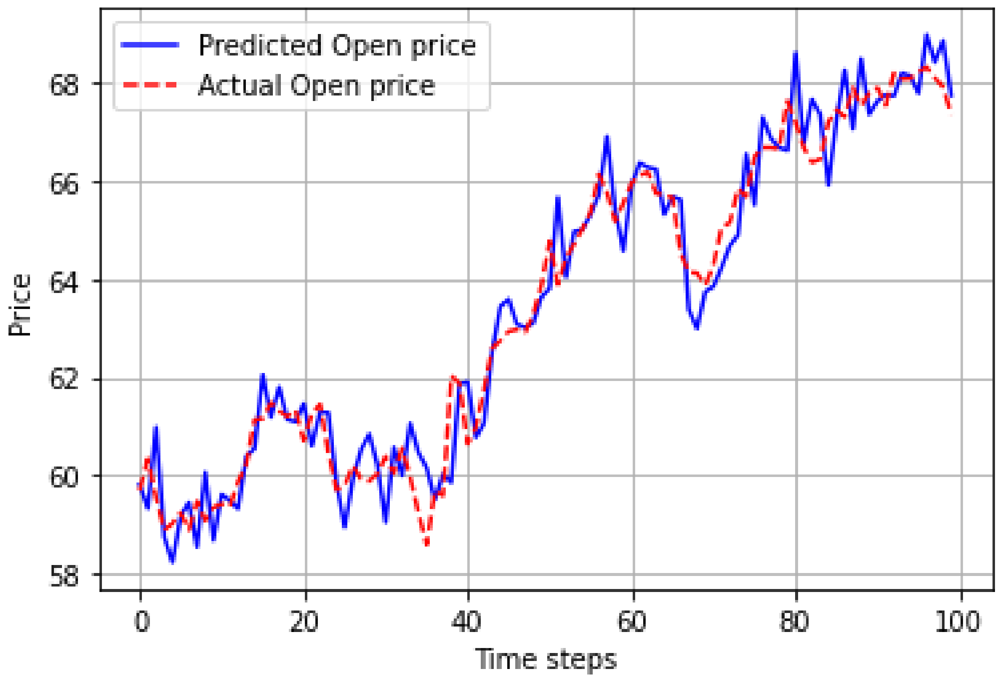
- The model is evaluated using the MAPE (Mean Absolute Percentage Error) between the predicted values and the ground Truth Test values.

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|Predicted(i)-True(i)|}{True(i)}$$

- N-number of test samples
- MAPE is calculated for each feature- Open, Close, High and Low separately

Results

CLOSE PRICE



OPEN PRICE

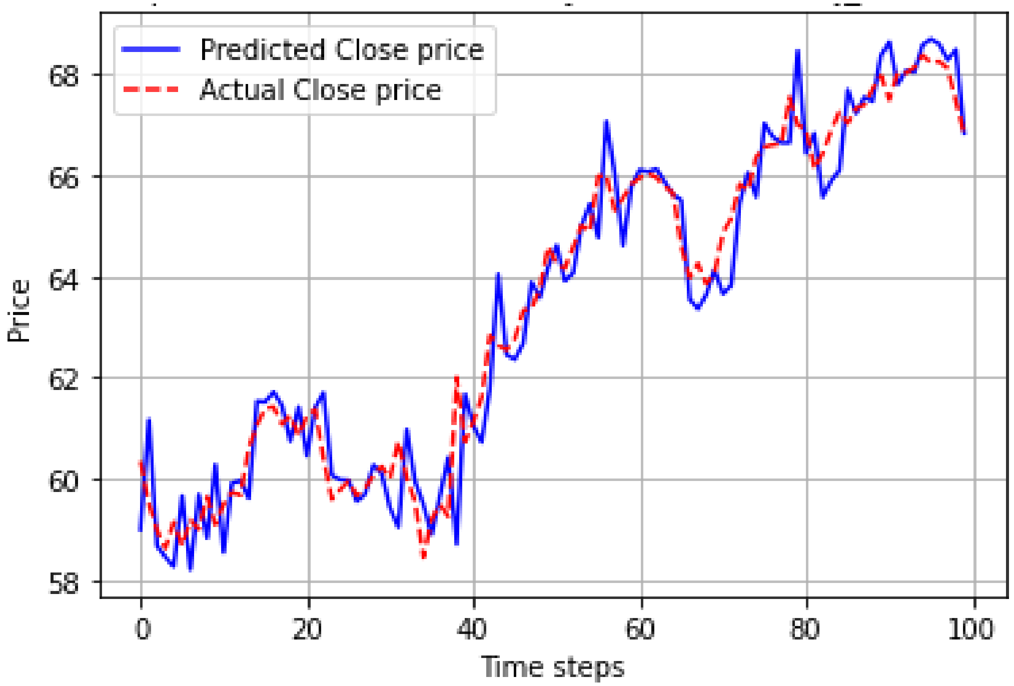
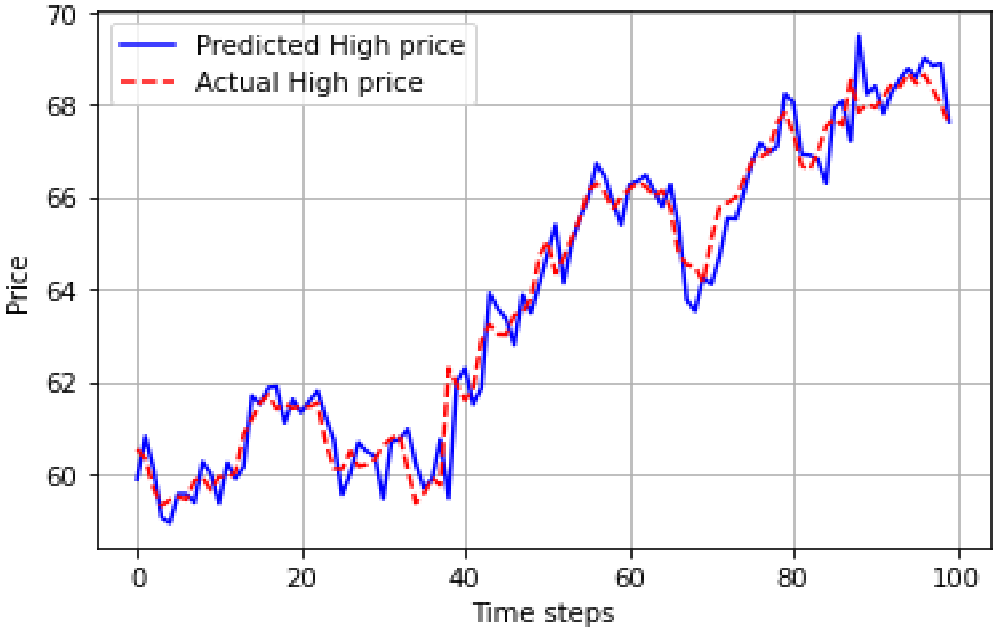


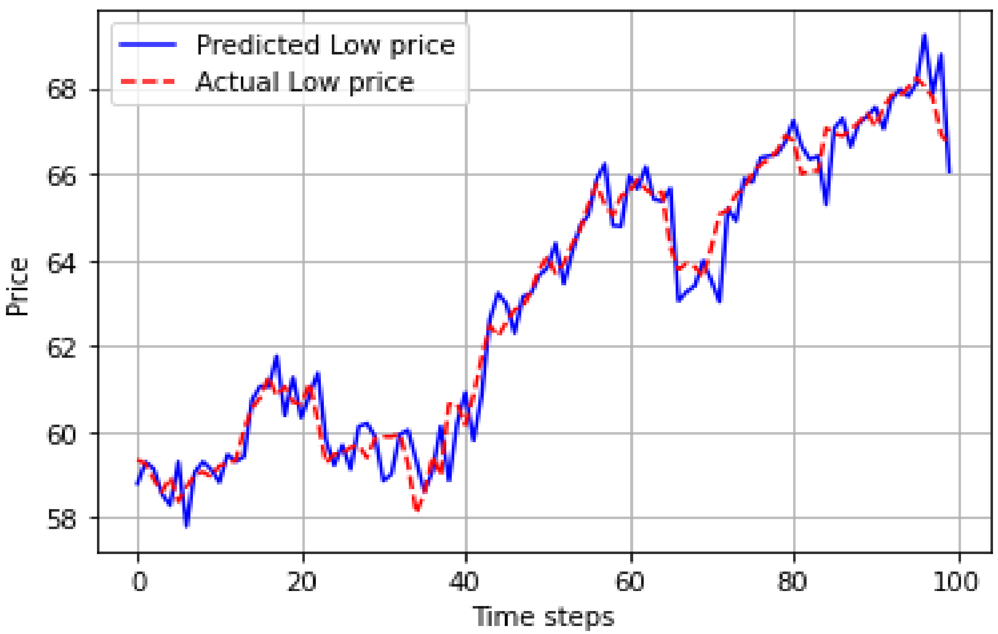
Fig: Plots showing Opening and Closing price prediction using HMM on the Huge Stock Market Dataset

Results

HIGH PRICE



LOW PRICE



	CLOSE	OPEN	HIGH	LOW
MAPE	0.0093	0.0092	0.0070	0.0077

MAPE values for four categories of the Historical Stock Price Dataset

Baseline

- Long Short Term Recurrent Neural Network (LSTM) used
- Popularly used on time-series data for prediction
- Capability to process the entire sequence of data
- Architecture comprises of the cell, input gate, output gate and forget gate.

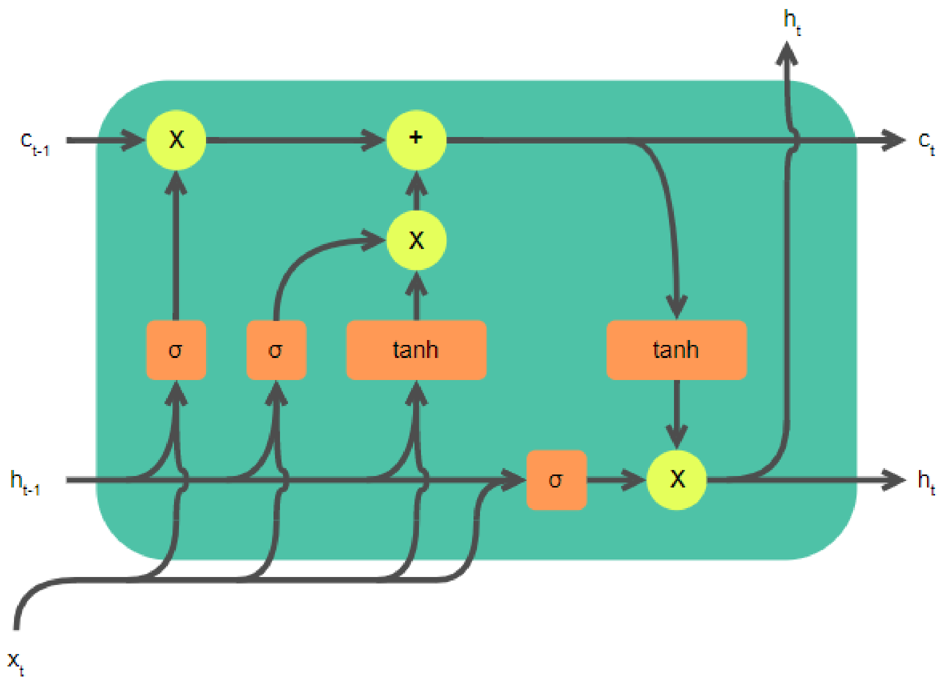


Fig: Architecture of a Long Short-Term Memory (LSTM) cell

Comparison with Baseline

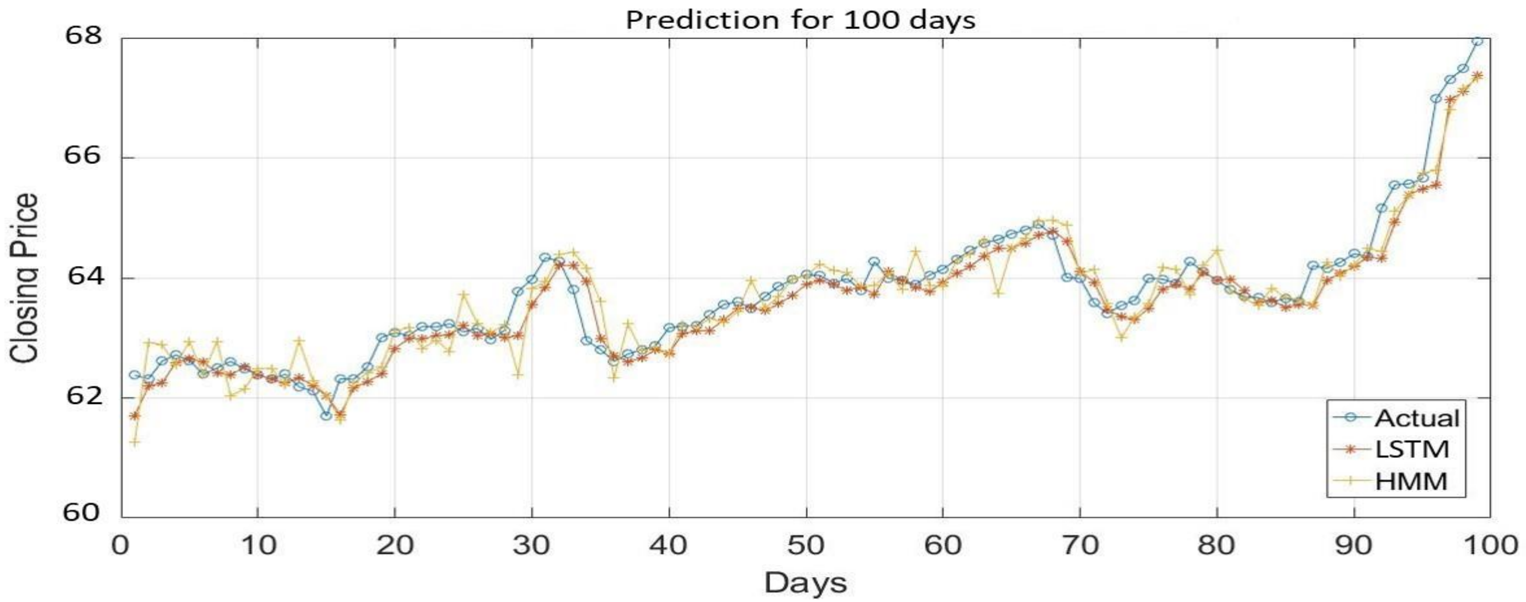


Fig: Plot showing comparison between ground truth data, HMM and LSTM predictions.

MODEL	CLOSE	OPEN	HIGH	LOW
HMM	0.0087	0.0091	0.0074	0.0085
LSTM	0.0062	0.0044	0.0057	0.0059

MAPE comparison of the 2 Models

Future Scope and Conclusion

- When compared to the Baseline: LSTM’s, HMM’s are suitable for capturing the dynamicity of the data while LSTM’s are suitable for stable data.
- They both have a similar performance when BIC is used for finding optimal states.
- Predictions can be made with only a certain accuracy and diverge when made for more than a day.
- Other parameters such as OpenInt would greatly increase the prediction accuracy of the HMM.
- We will try to incorporate this and search for other parameters as well.

THANK YOU!