STOCK PRICE PREDICTION USING TIME SERIES MODELS
STOCK PRICE PREDICTION USING TIME SERIES MODELS

Dissertation submitted in partial fulfillment of the requirements for the degree of

MSc Data Analytics

At Dublin Business School

Shruti Chouksey

Supervisor: Ms. Terri Hoare

MSc Data Analytics 2018 - 19
DECLARATION

I, Shruti Chouksey, declare that this research is my original work and that it has never been presented to any institution or university for the award of Degree or Diploma. In addition, I have referenced correctly all literature and sources used in this work and this work is fully compliant with the Dublin Business School’s academic honesty policy.
ACKNOWLEDGMENTS

I would like to express my special thanks of gratitude to Ms. Terri Hoare, my teacher and research supervisor for her patient guidance from the beginning with constructive and valuable suggestions provided throughout the planning and improvement of my dissertation. She also helped me in doing a lot of Research, and her ability to give her time generously has been particularly valued.

Secondly, I would like to thank my family and friends for their support and encouragement throughout my course.
ABSTRACT

This Thesis titled- “Stock price forecasting using time series models” focused on the comparison of the performance of time series models to predict the stock price for 5 banks. Forecasting and stock price analysis is important in finance and economics. Time series forecasting can be applied on any set of variables that change over time. For stocks or share prices, time series forecasting is common to track the price movement of the security over time. There is considerable past research work available on time series forecasting. In this thesis, a comparative study of time series forecasting using 3 models ARIMA (autoregressive integrated moving average), PROPHET and KERAS with LSTM (Long Short Term Memory) models has been explored. Historical stock price data was obtained from the National Stock Exchange (NSE) and used to build these models for comparative purposes. The results obtained reveal that all 3 models have strong potential for prediction and forecasting on the sourced historical data samples. All of the models performed better on larger data samples with LSTM best able to forecast seasonality.
# Table of Contents

DECLARATION .................................................................................................................. 3

ACKNOWLEDGMENTS ....................................................................................................... 4

ABSTRACT .......................................................................................................................... 5

List of Important Abbreviations ....................................................................................... 9

1 INTRODUCTION ........................................................................................................... 10

1.1 Terms and Definitions .............................................................................................. 10

1.2 Dissertation Roadmap .............................................................................................. 12

2 LITERATURE REVIEW ............................................................................................... 13

2.1 Business Importance ............................................................................................... 13

2.2 Scope ....................................................................................................................... 13

2.3 Previous Research Analysis .................................................................................... 13

3 RESEARCH AIMS AND OBJECTIVES ..................................................................... 16

3.1 Forecasting Methods ............................................................................................... 16

3.2 Proposed Approach ................................................................................................. 16

4 METHODOLOGY .......................................................................................................... 17

4.1 Sample Data Preparation ....................................................................................... 17

Time Series Features Selection .................................................................................... 18

4.2 CRISP-DM Design ................................................................................................ 18

4.3 Time Series Components ....................................................................................... 19

4.4 Procedures and Functions ....................................................................................... 20

4.4.1 ARIMA Model .................................................................................................. 20

4.4.2 PROPHET Forecasting Model ......................................................................... 22

4.4.3 KERAS with LSTM model ............................................................................ 24

4.5 Software and Packages .......................................................................................... 26

5 DATA ANALYSIS ......................................................................................................... 28

5.1 Exploratory Data Analysis ...................................................................................... 28

5.1.1 Sample Data Analysis ...................................................................................... 28

5.1.2 RMSE value analysis ...................................................................................... 30

5.1.3 20% Test Forecasting Analysis ..................................................................... 32

5.1.4 10% Forecasting Analysis ............................................................................. 33

6 RESULTS AND DISCUSSION ..................................................................................... 34

6.1 Result Review .......................................................................................................... 34

6.1.1 ARIMA (Autoregressive Moving Integrated Average model) ......................... 34
List of Tables and Figures

Fig 1.1: Axis Bank stock price from NSE Index
Fig 1.2: Dissertation roadmap
Table 4.1: Least RMSE value
Fig 4.2: CRISP-DM Methodology
Fig 4.3: CRISP-DM Methodology like methodology used
Fig 4.4 Time Series Components
Fig 4.5 Time Series Components Analysis
Fig 4.6 ARIMA model (Time Series)
Table 4.7 Least AIC value for ICICI bank
Fig 4.8 Graphical representation of least AIC values (acf, pacf)
Fig 4.9 Prophet forecasting flow
Table 4.10 Predicted values after applying Prophet Model
Fig 4.11 RNN transformation using hidden state
Fig 4.12 LSTM cell diagram
Table 4.13 Future predicted values after applying KERAS package with LSTM
Fig 4.14 Rstudio console
Fig 5.1 Graphical representation of ICICI bank stock closing price index
Fig 5.2 Graphical representation of SBI bank stock closing price index
Fig 5.3 Non stationary data for ICICI bank
Fig 5.4 Graphical representation after applying differencing method
Fig 5.5 RMSE values analysis for AXIS bank for all 3 models
Fig 5.6 RMSE values analysis for HDFC bank for all 3 models
Fig 5.7 RMSE values analysis for ICICI bank for all 3 models
Fig 5.8 RMSE values analysis for KOTAK bank for all 3 models
Fig 5.9 RMSE values analysis for SBI bank for all 3 models
Fig 5.10 SBI bank stock price forecasting for 20% test dataset
Fig 5.11 ICICI bank stock price forecasting for 20% test dataset
Fig 5.12 SBI bank stock price forecasting for 10% test dataset
Fig 5.13 ICICI bank stock price forecasting for 10% test dataset
Table 6.1 Least AIC value for all 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)
Table 6.2 Accuracy (RMSE) values for all 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)
Fig 6.3 Graphical representation of RMSE values for all 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)
Table 6.4 Predicted values for 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)
Table 6.5 Accuracy (RMSE) values for all 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)
Fig 6.6 Graphical representation of RMSE values for all 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)
Table 6.7 Predicted values for 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)
Table 6.5 Accuracy (RMSE) values for all 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)
Fig 6.6 Graphical representation of RMSE values for all 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)
Table 6.7 Predicted values for 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)
List of Important Abbreviations

➢ ACF- Autocorrelation function
➢ PACF- Partial autocorrelation function
➢ AR - Autoregressive
➢ ARIMA – Autoregressive Integrated Moving Average
➢ ARMA- Autoregressive moving average model
➢ LSTM- Long Short Term Memory
➢ RMSE- Root Mean Square Error
➢ AIC- Akaike Information Criteria
➢ ts- Time series
➢ RNN- Recurrent Neural Networks
➢ SVM- Support Vector MAchine
➢ ANN- Artificial Neural Networks
➢ CNN- Convolutional Neural Network
➢ CRISP-DM - Cross-Industry Standard Process for Data Mining
➢ EDA – Exploratory data Analysis
➢ NSE- National Stock Exchange
➢ BSE- Bombay Stock Exchange
➢ NYSE- New York Stock Exchange
➢ CSV- Comma-separated values
➢ HTML- Hypertext Markup Language
➢ PDF- Portable Document Format
1 INTRODUCTION

“Prediction is a very difficult art, especially when it involves the future” - Neils Bohr (Nobel Laureate Physicist).

“Forecasting is the typical process which helps in making statements whose actual outcomes yet to be observed or have not yet been observed” Wikipedia.

1.1 Terms and Definitions

What is time series forecasting?

A time series is a sequence of data points that are listed in time order. Time series analysis is combinations of different methods which help in analyzing time series data so as to get extricate meaningful statistics. Time series forecasting is a methodology that helps the model to predict future values using previously observed values.

Brockwell, P.J., Davis, R.A. and Calder, M.V., 2002, and Granger, C.W.J. and Newbold, P., 2014 stated that time series is a set of observations of N number of elements each of which recorded at a particular time (t). A time series is a sequence of values recorded in order by the time parameter. Such as Economic Forecasting, Sales Forecasting, Stock Market Analysis, Yield Projections, Process and Quality Control, Inventory Studies, Workload Projections, Utility Studies, Census Analysis, medical, meteorology (rainfall, temperature), astronomy and many more. Most of the classical statistical techniques and methods are not relevant in time series studies, so the new techniques Time Series analysis have been devised.

Time series forecasting (Granger, C.W.J. and Newbold, P., 2014) is a technique of drawing inference and making a prediction of the future. These techniques required to set up some initial hypothetical probability models, estimate parameters, check for the goodness of fit to the data and possibly to use the fitted model to enhance the understanding of the mechanism generating the series. Once a satisfactory model has been developed, it may be used in a variety of ways depending on the particular field of application. There are many methods used to model and forecast time series. The application type and preference decide the selection of the appropriate technique.

Basically, (Montgomery, D.C., Jennings, C.L. and Kulahci, M., 2015) there need to distinguish between a forecast and predicted value of $y_t$ that was made at some previous time period, and a fitted value of $y_t$ that has resulted from estimating the parameters in a time series model to historical data. The forecast error that results from a forecast of $y_t$ that was made at time period $t - \tau$ is the lead $-\tau$ forecast error.

$$e_t(\tau) = y_t - \hat{y}_t(t - \tau)$$

For example, the lead $-1$ forecast error is

$$e_t(1) = y_t - \hat{y}_t(t - 1)$$

The difference between the observation $y_t$ and the value obtained by fitting a time series model to the data, or a fitted value $\hat{y}_t$ defined earlier, is called a residual, and is denoted by

$$e_t = y_t - \hat{y}_t$$
**What is stock?**

A stock (Akhilesh Ganti, Mar 2019) (also known as "shares" or "equity) is a kind of security that implies proportionate proprietorship in the issuing organization. This qualifies the investor for that extent of the company's advantages and profit. Stocks are purchased and sold dominantly on stock trades, however there can be private deals also, and the establishment of about each portfolio.

These transactions have to conform to government regulations which are meant to protect investors from fraudulent practices. Historically, they have outperformed most other investments over the long run. These investments can be purchased from most online stock brokers.

An analysis (Lo, A.W. and Wang, J., 2001) begins with $I$ investors indexed by $i = 1, ..., I$ and $J$ stocks indexed by $j = 1, ..., J$. There is always an assumption that all the stocks are risky and non-redundant. For each stock $j$, let $N_jt$ be its total number of shares outstanding, $D_{jt}$ its dividend, and $P_{jt}$ its ex-dividend price at date $t$.

For notational convenience and without loss of generality, assume throughout that the total number of shares outstanding for each stock is constant over time, i.e. $N_{jt} = N_j, j = 1, ..., J$.

For each investor $i$, let $S_{jt}^i$ denote the number of shares of stock $j$ he holds at date $t$. Let

$$ P_t \equiv [P_{1t}, ..., P_{Jt}]^T $$

which denote the vector of stock prices and shares held in a given portfolio, where $A_t$ denotes the transpose of a vector or matrix $A$.

Finally, denote by $V_{jt}$ the total number of shares of security $j$ traded at time $t$, i.e., share volume, hence

$$ V_{jt} = \frac{1}{2} \sum_{i=1}^{I} |S_{jt}^i - S_{jt-1}^i| $$

Where the coefficient $\frac{1}{2}$ corrects for the double counting when summing the shares traded over all investors.

**Time series analysis for stock price trends:**

Time series can be applied on any set of variables that change over time. For stocks or share price, time series is common to track the price of a security over time. This can be tracked over the short term, such as the price of a security from time of open to close of a business day, or closing price of a daily business day or maybe last day of every month over the course of last 15-20 years. The seasonal trend and flow is the highlight of the stock market.

This can be useful to see how a company capital, security or any related economic variable changes over time. It can also be used to examine how the changes associated with the chosen data point compare to shifts in other variables over the same time period.

For example, suppose analyzing a time series of daily closing stock prices for Axis bank stock from National Stock Exchange Index India over a period of ten years. Obtain a list of all the closing prices for the stock from each business day for the past ten years and list them in chronological order. This would be ten years all closing price time series for the stock.
The fig 1.1 shows the graphical presentation of stock price over the period of more than ten years.

![Image of stock price graph]

Fig. 1.1: Axis Bank stock price from NSE Index

1.2 Dissertation Roadmap
This thesis covered forecasting models, the algorithms used within the model and other optimization techniques used for better performance and accuracy for time series dataset. The various performance evaluation parameters applied for automatic selection of proper input variables and model-dependent variables and optimizing the model parameters simultaneously to compare the results with those of other various methods, which show the effectiveness of the proposed approach for the seasonal time series.

The proposed method is a model-independent approach. Here main focused was on predicting price forecasting for any particular bank using the daily closing price and identified trends and patterns in the data using different algorithms. For this RNN (Recurrent Neural Networks) approach LSTM (Long Short Term Memory) was used for the stock price prediction of NSE (National Stock Exchange) listed banks (HDFC, AXIS, ICICI, KOTAK, and SBI) and compare their performance. Also applied different algorithms- ARIMA model, PROPHET algorithm, and KERAS with LSTM (Long Short Term Memory) model for predicting future values on a short term basis and performance of the models was quantified using RMSE error.

![Dissertation roadmap]

Fig. 1.2: Dissertation roadmap
2 LITERATURE REVIEW

2.1 Business Importance
Stock market or equity market has an overpowering impact in today’s economy. A rise or fall in the share price has an important role in determining the investor’s gain. Stock market forecasting is always very tricky. No one can see the future—the world is inherently uncertain and surprising things will happen. Even if it is known what’s going to happen, however, you might not know how markets will respond.

Investment in the stock market is regarded as high risks and high gains and so attracts a large number of investors and economists. However, information regarding a stock is normally incomplete, complex, uncertain and vague, making it a challenge to predict the future economic performance. People invest in the stock market based on some analysis.

2.2 Scope
Forecasting can be defined as the prediction by analyzing the historical data for many areas including business and industry, economics, environmental science, and finance. Forecasting of time series data provides the organization with useful information that is necessary for making important decisions.

Many of the forecasting’s involves the analysis of time. A time series of data for forecasting can be defined as a chronological sequence of observations for a selected variable (stock price). It can either be univariate or multivariate. Univariate data includes information about only one particular stock whereas multivariate data includes stock prices of more than one company for various instances of time.

Also, the analysis of patterns helps in identifying the best-performing companies for a specified period. This makes time series analysis and forecasting an important area of research.

Stock prices are more informative when the information it contains has less social value. Stock market prediction is always uncertain rather an analysis of time series data helps in identifying patterns, trends and periods or cycles existing in the data. It is shown that there is a fundamental tension between the informativeness of stock prices and the effectiveness of corporate governance, which limits the disciplining role of stock prices.

2.3 Previous Research Analysis
The ARIMA model is one of the most widely used techniques for analyzing time series data points or to predict future data points. One research study of 2004 (Ariyo, A.A., Adewumi, A.O. and Ayo, C.K., 2014) has compared ARIMA models with Support Vector Machine and proposed a hybrid methodology that exploits the unique strength of the ARIMA model and the SVM model in forecasting stock prices problems. They have used real data sets from stock prices to examine the forecasting accuracy of the proposed model.

The hybrid model proposed is composed of the ARIMA component and the SVMs component. Thus, the model can model linear and nonlinear patterns with improved overall forecasting performance. The presented model is believed to greatly improve the prediction performance of the single ARIMA model in forecasting stock prices. This study demonstrated that a simple combination of the two best individual models does not necessarily produce the best results. Therefore, the structured selection of optimal parameters of the model is of great interest.
A comprehensive study of A. Victor Devadoss and T. Antony Alphonse Ligori, 2013 was done on the common parameters to design a neural network for forecasting economic time series data. The study identifies Artificial Neural Networks are a highly flexible function that can map any nonlinear function and are being used widely in many different fields of business and industry. One of the major application areas of ANN is forecasting.

They implemented nonlinear technique ANN to forecast the stock prices of BSE (Bombay Stock Exchange) stocks and concluded that predicted results of ANN was successful with better accuracies which clearly ensured that ANN is suitable and perform better for forecasting. A clear observation says that in order to improve the performance of the network macroeconomic factors and technical analysis indicators may be used as input variables. They also used Multilayer feedforward network with back propagation algorithm for forecasting purpose which has multiple layers in between and data flows in one direction from input to output layer.

Research paper of Ariyo, A.A., Adewumi, A.O. and Ayo, C.K., 2014 presented the extensive process of designing time series models for stock price prediction using the ARIMA model. They have used the New York Stock Exchange (NYSE) and Nigeria Stock Exchange (NSE) data for research. They used Eviews analytical software and picked up data stock data from two different countries. The data composed of four elements, namely: open price, low price, high price, and close price respectively. For research, the closing price was chosen to represent the price of the index to be predicted. The closing price was chosen because it reflects all the activities of the index in a trading day. Analyzing their results revealed that the ARIMA model has short term potential for prediction. As per this study, ARIMA models can compete reasonably well with emerging forecasting techniques in short-term prediction.

X. Ding, Y. Zhang, T. Liu, and J. Duan, 2015 demonstrated that deep learning method was useful for event-driven stock market prediction by using deep convolution neural network model with the combined influence of long-term events and short-term events on stock price movements, proposing a neural tensor network for learning event embeddings. They focused on automatically learn embeddings for structured events which shows more fundamental relations between events, even if the same object or event is not shared between events. Various studies have found that financial news may have an impact on share price hence they used structured events to represent news, which represents objects of the event. Their experimental result represents that event embedding is useful for the task of stock price prediction. However, the event embeddings based methods give more accurate performance than event-based methods and deep convolution neural network might help in capturing long term influence of news event than standard feedforward neural network.

There are models that are too simple to catch all of the available information. On the other hand there are methods with more parameters employed in order to cope with more demanding underlying patterns; unfortunately, while optimizing all these parameters usually these complex methods end up actually over-fitting the actual data. So, this approach aims to help the models capture the data. This is achieved by breaking the data down into several simpler series, each one of which captures part of the information included in the original series. As a result of this process, simpler models can adapt to these simpler series.

Menon, V.K., Vasireddy, N.C., Jami, S.A., Pedamallu, V.T.N., Sureshkumar, V. and Soman, K.P., 2016 analyzed NSE (National Stock Exchange) stocks using linear models (AR and ARMA). The study identifies that AR predictions are positive whereas ARMA has an enhancement trend. Their prediction worked well for almost all of the stock despite in case of stock splits.
Another Insight they have gained was AR predictions were positive at best while ARMA had an exaggeration trend. So if the prediction is a rise, then the AR value will better quantize that and if it’s a fall then it can bank on the ARMA forecast value. So the conclusion is if a prediction is a rise then AR value will be more accurate otherwise can rely on ARMA forecast value if a prediction is a fall.

Hiransha, M., Gopalakrishnan, E.A., Menon, V.K., and Soman, K.P., 2018 experimented stock price prediction using three deep learning models (RNN, LSTM, CNN) and they concluded that the CNN (Convolutional neural network) performs better and gives more accurate results than other two models. The reason is that CNN doesn’t depend on any previous information for prediction. CNN helps the model to understand patterns and able to capture trends occurring in the current window since CNN follows the current window for prediction. To compare the error percentage they used linear model (ARIMA) for forecasting. They also observed that changes occurring in the stock market may not follow the same cycle every time and the existence of the trend may be unlike sector wise. Such an analysis of trends and cycles will surely help investors to gain more profit.

LSTM (C Olah, C., 2015) (Long Short Term Memory) is the most popular RNN (Recurrent Neural Networks) approach and they are networks with loops in them, allowing information to persist. These loops make recurrent neural networks seem kind of strange. A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor.

KERAS (Choi, K., Joo, D. and Kim, J., 2017) used deep learning in audio informatics research. For KERAS, preprocessing data frequently involves bunches of time and effort. It was difficult when managing audio data than images or texts because of its vast size and overwhelming interpreting calculation. Hence they utilized a generator. To actualize a generator that loads the data they included a KAPRE layer at the input side of the KERAS model. Their main focus behind using KAPRE is to perform audio preprocessing in the KERAS layers.

Forecasting at scale analyzed by Taylor SJ, Letham B. 2017 where they focused on many iterations of forecasting and they implemented it on Facebook data. They used Facebook data so they applied the PROPHET method which was developed by Facebook. They also used the simple modular regression model which provides the best performance using default parameters. Their main focus was on the flagging forecast for manual check and measure or tracking forecast accuracy. Finally, they concluded that simple and adjustable models help to give better performance with a large amount or a variety of data.

Basically, there are three categories of layer: input, hidden and output. The input layer takes the raw input data as its input; the hidden layers take the output from the previous layer as its input and the output from the output layer are seen as the results of the ANN (Artificial Neural Network). Hidden layers are titled hidden as one does not see or know their inputs or outputs. They are often characterized as feature detectors (Shachmurove, 2002). Generally, it is accepted to have n input perceptron for n inputs however there is no recognized optimum number of hidden perceptron or layers and the number of output perceptron depends on the number of outputs necessary (Kartalopoulos, 1996, Hastie et al., 2001).

“No indication is provided as to the optimal number of nodes per layer. There is no formal method to determine this optimal number; typically one uses trial and error.” [Kartalopoulos, 1996]
3 RESEARCH AIMS AND OBJECTIVES

Time series forecasting includes creating and utilizing a predictive model by analyzing the historical data which has a relationship between observations. The decision which is made after applying models, directly impact each and every step of project including the evaluation of forecast models to the fundamental difficulties of the forecast research. Stock price forecasting is the demonstration of attempting to decide the future estimation for any companies stock.

Research Question- Comparison of neural network and deep learning algorithms with traditional statistical learning approaches for stock price forecasting by analyzing NSE listed banks stock price.

Aim- Assist Indian banks with decisions relating to stock price forecasting using time series analysis.

Objective- To analyze and compare different deep learning and statistical learning algorithms trained on 15-20 years of bank data for 5 banks to predict the estimated price of stocks. The objectives are:
1) Model specification (or model identification);
2) Model fitting (or model estimation);
3) Model evaluation (or model accuracy assessment).

3.1 Forecasting Methods
1) Autoregressive integrated moving average (ARIMA) methods
2) KERAS with LSTM (Long Short Term Memory) model
3) PROPHET Algorithm

3.2 Proposed Approach
The CRISP-DM methodology (Devi, B.U., Sundar, D. and Alli, P., 2013) is a 6 steps (Business understanding, Data understanding, and Data preparation, Modeling, Evaluation and Deployment) process for identifying, selecting, and assessing conditional mean models for discrete, Univariate time series data. For this dissertation proposed steps are:

Step 1: Business Data Understanding
1) Business objectives
2) Business importance
3) Sample data collection
4) Describe and explore the data

Step 2: Data Preparation
1) Transform the data to stabilize the attributes
2) Integrate and format data

Step 3: Data Modeling
1) Examine the data to identify potential models
2) Estimation and testing
3) Predict and forecast

Step 4: Data Evaluation
1) Evaluate results
2) Compare models
4 METHODOLOGY

Moving from machine learning to time-series forecasting is a radical change. It is a challenging, yet enriching experience that helped in understanding in a better way how machine learning can be applied to forecast the stock price. The objective of a predictive model was to estimate the value of an unknown variable. A time series has time \( t \) as an independent variable and a target dependent variable \( y \). The output of the model is the predicted value for \( y \) at time \( t(y) \).

Time series models- ARIMA, PROPHET, and KERAS with LSTM were used for stock price prediction. Sample data collected from Yahoo Finance in the CSV (Comma-separated values) format. CRISP-DM methodology concept was applied to design stock price data transformation and load in the Rstudio for model evaluation. All 5 models were applied for 5 banks (HDFC, ICICI, SBI, KOTAK, and AXIS) stock price data to compare the time series algorithms.

4.1 Sample Data Preparation

The data set consists of the stock price for National Stock Exchange listed 5 Indian corporate banks for the period of 20 years. It includes information like date, stock close price, number of shares and market capital. For this work selected companies from banking sectors for study which are HDFC Bank, AXIS Bank, Kotak Bank, SBI Bank and ICICI Bank. These banks were identified by the help of NIFTY 50 index. The data for these banks were extracted from the available data on Yahoo Finance in the .csv format and was subjected to preprocessing to obtain the stock price using the R programming language. Read the data in R to and then converted it into time series (ts) format.

```r

> Train_Axis=ts(Axis$Monthly_Avg_Share_Price,start=c(1999,01), frequency = 12)
```

The work was based on a sliding window approach for a short term future prediction. To make the data consistency converted the dataset into a monthly average. In order to get more accurate result divided dataset for all 5 banks into trainset and test set in order of trainset- 90%, 80%, 70%, 60% and test set- 10%, 20%, 30%, 40% and then used for loop to calculate the accuracy. Below is the example of all 3 models applied in R.

```r
> Axis_fit=Arima(Train_Axis,order = c(Least_AIC$p,Least_AIC$d,Least_AIC$q))

> M=prophet(train)

> future <- make_future_dataframe(M, periods = N-n,freq = 'm')

> model <- keras_model_sequential()

> model%>%
layer_lstm(units, batch_input_shape = c(batch_size, X_shape2, X_shape3), stateful= TRUE)%>%
layer_dense(units = 1)
```

So result look like below:

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Test</th>
<th>RMSE Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARIMA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>146 (60%)</td>
<td>97 (40%)</td>
<td>223.8319</td>
<td></td>
</tr>
<tr>
<td>170 (70%)</td>
<td>75 (30%)</td>
<td>216.0632</td>
<td></td>
</tr>
<tr>
<td>194 (80%)</td>
<td>49 (20%)</td>
<td>84.51927</td>
<td></td>
</tr>
<tr>
<td>219 (90%)</td>
<td>24 (10%)</td>
<td>86.21248</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.1: Least RMSE value
Time Series Features Selection
In machine learning features are created either manually or automatically, creating features is one of the most important tasks in applied machine learning which is not there in time series forecasting. This does not mean that features are completely off limits. Instead, they should be used with care because of the following reasons:

- A pure time series model may have similar or even better performance than one using features.
- If the features are predictable then they have some patterns that can help in building a forecast model for each of them. However, keeping in mind that using predicted values as features will propagate the error to the target variable, which may cause higher errors or produce biased forecasts.

4.2 CRISP-DM Design
CRISP-DM (Wirth, R. and Hipp, J., 2000) (Zazzaro, Gaetano & Romano, Gianpaolo & Mercogliano, Paola 2017) (CRoss Industry Standard Process for Data Mining) method is very first and an open standard process that describes a structured approach to plan a data mining project. The CRISP-DM process mainly focuses on making large data mining projects that are more reliable, faster and manageable but less costly.

As a methodology, it includes descriptions of the typical phases of a project, the tasks involved with each phase, and an explanation of the relationships between these tasks and as a process model, CRISP-DM provides an overview of the data mining life cycle.

The life cycle model consists of six phases with arrows indicating the most important and frequent dependencies between phases. The CRISP-DM model is flexible and can be customized easily. For example, if an organization aims to detect money laundering, it is likely that it will sift through large amounts of data without a specific modeling goal. Instead of modeling, work will focus on data exploration and visualization to uncover suspicious patterns in financial data. CRISP-DM allows creating a data mining model that fits particular needs.

In such a situation, the modeling, evaluation, and deployment phases might be less relevant than the data understanding and preparation phases. However, it is still important to consider some of the questions raised during these later phases for long-term planning and future data mining goals.
For this thesis deployment phase is not relevant so the methodology used is like CRISM-DM not complete CRISP-DM.

Fig. 4.3: CRISP-DM Methodology like methodology used

4.3 Time Series Components
The factors that are responsible for bringing about changes in a time series, also known as the components of time series, are as follows:

- General Trends
- Seasonal Movements
- Cyclical Movements
- Irregular Fluctuations

Fig. 4.4 Time Series Components

**Trend:** A trend exists when a series increases, decreases or remains at a constant level with respect to time.

**Seasonality:** This property of a time series refers particular periodical patterns that repeat at a constant frequency. Time series models include seasonal variables as dummy features, using binary variables to avoid correlation between features.

**Cycles:** Cycles are seasons that do not occur at a fixed rate. Also, do not repeat at regular time intervals and may occur even if the frequency is 1.
Fig. 4.5 Time Series Components Analysis

Time series components are highly important to analyze variable of interest in order to understand its behavior, what patterns it has, and to be able to choose and fit an appropriate time-series model.

4.4 Procedures and Functions

For this thesis, applied models on training dataset to predict values for the test dataset. The performance evaluation parameter RMSE applied for accuracy measurement after trainset and test set data selection of input and model dependent variables. For seasonal time series, optimized models parameters simultaneously to compare and discuss the result. 3 models are chosen dependent on the basis of the research done on time series forecasting are as below:

4.4.1 ARIMA Model

Since, it is basic to recognize a model (Devi, B.U., Sundar, D. and Alli, P., 2013) to demonstrate the pattern with sufficient data for the financial specialist to settle on a choice. It prescribes that ARIMA is an algorithmic way to deal with change the arrangement is superior to anticipating straightforwardly, and furthermore it gives increasingly exact outcomes. This methodology has excluded any testing criteria for model estimation.

The ARIMA approach was popularized by Box and Jenkins (Devi, B.U., Sundar, D. and Alli, P., 2013), and ARIMA models are often referred to as Box-Jenkins models. The general transfer function model employed by the ARIMA procedure was discussed by Box and Tiao in 1975.

ARIMA model (Auto-Regressive Integrated Moving Average) is a class of statistical models for analyzing and forecasting time series data. It explicitly caters to a suite of standard structures in time series data. The AR part of ARIMA indicates that the evolving variable of interest is regressed on its prior values. The MA part indicates that the regression error is actually a linear combination of error terms whose values occurred contemporaneously and at various times in the past.

Non-seasonal ARIMA models are generally denoted ARIMA (p, d, q) where parameters p, d, and q are non-negative integers, p is the order (number of time lags) of the autoregressive model, d is the degree of differencing (the number of times the data have had past values subtracted), and q is the order of the moving-average model. Seasonal ARIMA models are usually denoted ARIMA(p, d, q) (P, D, Q)m, where m refers to the number of periods in each season, and the uppercase P, D, Q refers to the autoregressive, differencing, and moving average terms for the seasonal part of the ARIMA model.

ARIMA algorithm in three steps:
Step 1: Model identification
Step 2: Model estimation
Step 3: Forecasting

ARIMA MODEL

A non seasonal ARIMA model is classified as an "ARIMA(p,d,q)" model, where:

* p is the number of autoregressive terms,
* d is the number of non seasonal differences needed for stationarity, and
* q is the number of lagged forecast errors in the prediction equation.

*Stationary Series: A stationary series has no trend, its variations around its mean have a constant amplitude. A non stationary series is made stationary by differencing.

Fig. 4.6 ARIMA model (Time Series)

In an ARIMA model (Pai, P.F. and Lin, C.S., 2005), the future value of a variable is supposed to be a linear combination of past values and past errors. In general, ARIMA model is denoted by ARMA (p, q). The form of the ARMA (p, q) model is,

\[ y_t = C + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \theta_q \epsilon_{t-q} \]

Where \( \epsilon_t \) is an uncorrelated innovation process with mean zero and \( y_t \) is the actual value and \( \epsilon_t \) is the random error at time \( t \), \( \phi_1 \) and \( \phi_2 \) are the coefficients, \( p \) and \( q \) are integers that are often referred to as autoregressive and moving average polynomials.

For example, the ARIMA \((1,0,1)\) model can be represented as follows

\[ y_t = \theta_0 + \phi_1 y_{t-1} + \epsilon_t - \theta_1 \epsilon_{t-1} \]

Here, applied for loop using \((p,d,q)\) values between 0 to 2 to find least AIC value which later helped to find best the RMSE value as an outcome.

```r
> modelAIC <- data.frame()
> for (d in 0:1)
> { + for (p in 0:2)
> + { + for (q in 0:2)
> + { + Axis_fit=Arima(Train_Axis,order = c(p,d,q),method="ML")
> + AIC=AIC(Axis_fit)
> + modelAIC <- rbind(modelAIC, data.frame(p,d,q,AIC))
> + } + } + }

After applying for loop, found 54 values in total and 18 combinations for \((p, d, q)\) and 18 AIC values for each bank shown in below figure. Out of 18 AIC values the least AIC values’ \((p, d, q)\)
value is (0, 1, 1) (highlighted in yellow). Fig 4.8 represented the graphical visualization for least (p, d, q) which is (0, 1, 1) (acf and partial acf).

<table>
<thead>
<tr>
<th>p</th>
<th>d</th>
<th>q</th>
<th>AIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2378.656</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2444.571</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1984.136</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1650.315</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1648.608</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1645.329</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1649.188</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1648.482</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1650.446</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1633.775</td>
</tr>
<tr>
<td>**0</td>
<td>1</td>
<td>1</td>
<td><strong>1632.266</strong></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1632.844</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1632.839</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1632.033</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1633.877</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1633.758</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1633.903</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1643.785</td>
</tr>
</tbody>
</table>

Table 4.7 Least AIC value for ICICI bank

Fig 4.8 Graphical representation of least AIC values (acf, pacf)

4.4.2 PROPHET Forecasting Model

Prophet (Taylor SJ, Letham B. 2017) is a methodology for forecasting time series data dependent on an added substance model where non-direct patterns are fit with yearly, week by week, and daily trends, in addition to occasion impacts. It works best with time series that have solid occasional impacts and a few periods of authentic data. Prophet is vigorous to missing data and moves in the pattern, and commonly handles anomalies well.

Fig. 4.9 Prophet forecasting flow

Prophet is an additive model with the following components:
\[ y(t) = g(t) + s(t) + h(t) + \epsilon_t \]

Where, \( g(t) \) model trend, which describes long-term increase or decrease in the data. Prophet incorporates two trend models, a saturating growth model, and a piecewise linear model, depending on the type of forecasting problem.

\( s(t) \) Model seasonality with Fourier series, which describes how data is affected by seasonal factors such as the time of the year

\( h(t) \) Model the effects of holidays or large events that impact business time series

\( \epsilon_t \) Represents irreducible error term of any idiosyncratic changes which are not accommodated by the model

Prophet (Taylor SJ, Letham B. 2017) naturally assesses estimate execution and banners issues that warrant manual intercession. One of the most straightforward evaluation methods is to set a benchmark with some basic forecasting techniques. It is helpful to contrast shortsighted and propelled forecasting techniques with decide if extra execution can be picked up by utilizing an increasingly unpredictable model.

The Prophet (Taylor SJ, Letham B. 2017) forecast can predict both the week after week and yearly seasonality’s, and dissimilar to the baselines, does not overcompensate to the occasion plunge in the principal year. In the principal forecast, the Prophet model has somewhat over fit the yearly regularity allowed just a single year of information.

In the Prophet model (Taylor SJ, Letham B. 2017) one of the important specification which says there are a few spots where analysts can modify the model to apply their ability and outer information and it is not required to have any comprehension of the fundamental statistics. Some of the key specification to highlight below:

**Capacities:** Experts can use any outer information from any source for the absolute market estimate and then resourceful information as learning can be applied legitimately by determining limits.

**Changepoints:** Known dates of change points, such as dates of product changes, can easily be specified.

**Holidays and seasonality:** Experts’ experience involvement help with which occasions sway development in which districts and they can legitimately use as input to store relevant occasion dates and the relevant time sizes of regularity or seasonality.

**Smoothing parameters:** Hereby changing the value of \( \tau \), this can be chosen from inside scope of increasingly worldwide or locally smooth models. The seasonality and holiday smoothing parameters \( (\sigma, \nu) \) help model to amount of the historical seasonal variety which will be expected in the future.

In order to get the best predicted values used PROPHET package in R with trainset (60%, 70%, 80%, and 90%) and test set (40%, 30%, 20% and 10%). Below is the R code:

```r
>i=0
>M=prophet(train)
>future <- make_future_dataframe(M, periods = N-n.freq = 'm')
>FC= predict(M, future)
>Pred_value=list(0)
```
```r
> Actual_value=list(0)
> for (x in c(0.6,0.7,0.8,0.9))
+ { 
+ N = nrow(df)
+ n = round(N *x)
+ train = df[1:n, ]
+ test  = df[(n+1):N, ]
+ M= prophet(train)
+ future<- make_future_dataframe(M, periods = N-n,freq = 'm')
+ FC= predict(M, future)
+ }
```

Below table shows the result of the above query which represented predicted values for test dataset (40%, 30%, 20%, and 10%) for PROPHET model. Similarly, result tables have been generated for all 5 banks after applying PROPHET model.

<table>
<thead>
<tr>
<th>Pred values _40</th>
<th>Pred values _30</th>
<th>Pred values _20</th>
<th>Pred values _10</th>
</tr>
</thead>
<tbody>
<tr>
<td>237.6298468</td>
<td>275.7105767</td>
<td>341.2668104</td>
<td>544.2299788</td>
</tr>
<tr>
<td>245.1902297</td>
<td>279.5206584</td>
<td>351.1294229</td>
<td>553.8058583</td>
</tr>
<tr>
<td>253.7697587</td>
<td>279.7811739</td>
<td>357.7469229</td>
<td>557.1304658</td>
</tr>
<tr>
<td>255.5605151</td>
<td>279.1948909</td>
<td>361.0561604</td>
<td>564.0741909</td>
</tr>
<tr>
<td>260.4592625</td>
<td>283.7491245</td>
<td>365.3562378</td>
<td>565.788838</td>
</tr>
<tr>
<td>261.1650298</td>
<td>285.2513247</td>
<td>360.3097042</td>
<td>569.4520383</td>
</tr>
<tr>
<td>265.5374135</td>
<td>288.2537366</td>
<td>363.1235096</td>
<td>572.6965856</td>
</tr>
<tr>
<td>270.9613008</td>
<td>293.8515743</td>
<td>368.8047127</td>
<td>573.9355997</td>
</tr>
<tr>
<td>273.5315588</td>
<td>294.7924521</td>
<td>376.3092254</td>
<td>577.4671241</td>
</tr>
<tr>
<td>274.4772095</td>
<td>295.6082705</td>
<td>377.4894395</td>
<td>584.576187</td>
</tr>
</tbody>
</table>

Table 4.10 Predicted values after applying Prophet Model

### 4.4.3 KERAS with LSTM model

The LSTM (Long short-term memory) (Gulli, A. and Pal, S., 2017) is a variant of RNN that is capable of learning long term dependencies. LSTMs were first proposed by Hochreiter and Schmidhuber and refined by many other researchers. They work well on a large variety of problems and are the most widely used type of RNN. In deep learning, while preprocessing data using KERAS (Choi, K., Joo, D. and Kim, J., 2017) frequently involves bunches of time and effort.

The Simple RNN (Gulli, A. and Pal, S., 2017) uses the hidden state from the previous time step and the current input in a tanh layer to implement recurrence. LSTMs also implement recurrence in a similar way, but instead of a single tanh layer, there are four layers interacting in a very specific way.

The following diagram (4.11) illustrated the transformations that are applied to the hidden state at time step \( t \) where \( c \) is cell state (top line) represents the internal memory of the unit and \( i, f, o, \) and \( g \) are gates for a hidden state (bottom line):
The LSTM (Long short-term memory) (Gulli, A. and Pal, S., 2017) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. The core data structure of KERAS is a model, a way to organize layers. The simplest type of model is the Sequential model, a linear stack of layers. For more complex architectures the KERAS functional API can be used which allow building arbitrary graphs of layers.

![LSTM cell diagram](image)

**Fig. 4.12 LSTM cell diagram**

The mathematics of the LSTM cell looks like this:

So LSTM cell takes the previous memory state $C_{t-1}$ and does element-wise multiplication with forget gate ($f$)

$$C_t = C_{t-1} \times f_t$$

If forget gate value is 0 then previous memory state is completely forgotten.

If forget gate value is 1 then previous memory state is completely passed to the cell (where $f$ gate gives values between 0 and 1).

Now with current memory state $C_t$, we calculate new memory state from input state and C layer.

$$C_t = C_t + (I_t \times C_{t-1})$$

$C_t$ = Current memory state at time step t and it gets passed to next time step.

Finally, the output will be based on cell state $C_t$ but will be a filtered version. So need to apply $Tanh$ to $C_t$ then do element-wise multiplication with the output gate $O$, Which will be current hidden state $H_t$

$$H_t = Tanh (C_t)$$

Then pass these two $C_t$ and $H_t$ to the next time step and repeat the same process.
In order to get best future predicted values used KERAS package with LSTM for trainset (60%, 70%, 80%, and 90%) and test set (40%, 30%, 20% and 10%).

```r
> model <- keras_model_sequential()
> model$>
+ layer_lstm(units, batch_input_shape = c(batch_size, X_shape2, X_shape3), s+ tateful= TRUE)$>
+ layer_dense(units = 1)

# Compile
> model$>
+ loss = 'mean_squared_error',
+ optimizer = optimizer_adam( lr= 0.02, decay = 1e-6 ),
+ metrics = c('accuracy') )

# Fit the model
> Epochs = 50
> for(i in 1:Epochs )
+ {
+ model$>
+ fit(x_train, y_train, epochs=1, batch_size=batch_size, verbose=1, + shuffle=FALSE)
+ model$>
+ reset_states()
+ }
```

Below table shows the result of the above query which represented predicted values for test dataset (40%, 30%, 20%, and 10%) for KERAS with LSTM model. Similarly, result tables have been generated for all 5 banks after applying KERAS with LSTM model.

<table>
<thead>
<tr>
<th>Pred values</th>
<th>40</th>
<th>Pred values</th>
<th>30</th>
<th>Pred values</th>
<th>20</th>
<th>Pred values</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.13 Future predicted values after applying KERAS package with LSTM

4.5 Software and Packages

R is a programming language and free software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing.

R is an integrated suite of software facilities for data manipulation, calculation, and graphical display. It is a well-developed, simple and effective programming language which includes conditionals, loops; user defined recursive functions and input, and output facilities. R programming is a strong foundation in functional programming. The ideas of functional programming are well suited which help in solving many of the challenges for data analysis. R provides a powerful and flexible toolkit which allows writing concise yet descriptive code.

Contrasted with other programming languages, the R language, in general, is increasingly focused on results rather than procedures. Information of programming designing accepted procedures is inconsistent. R is very useful assets for imparting outcomes. R packages make it simple to use or deliver HTML or pdf reports or create interactive websites.
Fig 4.14 Rstudio console

**Packages and Libraries:**
```r
install.packages('prophet')
install.packages('rsample')
install.packages('tensorflow')
library(sqldf)
library(tidyverse)
library(keras)
library(forecast)
library(prophet)
library(data.table)
library(dplyr)
library(ggplot2)
```

**Number of Files:** 5 csv files (AXISBANK.csv, ICICIBANK.csv, HDFCBANK.csv, KOTAKBANK.csv, SBIBANK.csv)

**Number of records:** Each csv file contains about 20 years’ stock price data which has approx 5000 records each.
5 DATA ANALYSIS

5.1 Exploratory Data Analysis
Exploratory data analysis (EDA) is a systematic way to explore the data using transformation and visualization. EDA is an iterative cycle and it’s not a process with any set of rules however some basic steps to be applied that help to manage data in a systematic way:

Step 1: Generate questions about data.

Step 2: Search for answers by visualizing, transforming, and modeling the data.

Step 3: Use what is to learn to refine questions and if required then generate new questions.

5.1.1 Sample Data Analysis
After sample data preparation the next task in modeling was sample data analysis and visualization. Fig (5.1) and (5.2) representing the monthly average share price for ICICI and SBI bank. Both graphs were a representation of stock closing price index for almost 20-25 years from 2002 to 2019 for ICICI bank and 1996 to 2019 for SBI bank. X-axis represented years-months and y-axis represented an average of each month stock closing price.

There were up and down trends observed many times during the last 20 years. The stock price has experienced continuous growth since 2002. There was a sharp fall observed between 2008-2009, similarly, in Dec-2016, a sharp fall was observed. There are multiple sharp peaks observed for multiple years for ICICI bank, which could be a sign of good growth or a stock bubble.

Fig 5.1 Graphical representation of ICICI bank stock closing price index

The SBI bank graph was a representation of stock closing price index for almost 25 years from 1996 to 2019. X-axis represented years-months and y-axis represented an average of each month stock closing price. There were up and down trends observed many times during the last 25 years. The stock price has experienced very slow growth since 1996. After 2007 SBI bank stock price started growing usual than expected then there was a sharp fall during 2010. There are multiple sharp peaks observed for multiple years, which could be a sign of good growth or a stock bubble.
Fig 5.2 Graphical representation of SBI bank stock closing price index

Until now data was non-stationary and it is required to have stationary data for time series forecasting because it makes future prediction easier with stationary series data. For stationary time series, it is required to have the mean and variance constant over time. The perfect way to check whether mean and variance are constant or not is plotting which helps to find the correct difference for time series. Below fig shows non-stationary data.

Fig 5.3 Non stationary data for ICICI bank

To make data stationary differencing method is used which helps data to transform from non-stationary to stationary time series. Validation of the assumptions has been done using graphical visualization. The difference between the current time and previous time period is first differencing value. After applying differencing method time series looked like below which is stationary data.

Fig 5.4 Graphical representation after applying differencing method
5.1.2 RMSE value analysis

Root Mean Square Error (RMSE) calculates the error between population values predicted by a model and the values observed. RMSE is a proportion of accuracy which helps to analyze forecasting errors of various models for a specific dataset since it is scale-subordinate.

Once the sample data analysis was completed then the next step was an evaluation of the models. Here, 3 models were applied which were ARIMA, PROPHET, and KERAS with LSTM. Models result has been analyzed by RMSE value. Fig (5.5) (5.6) (5.7) (5.8) show the RMSE values for all 3 models for all 5 banks.

Since there were 4 train dataset (60%, 70%, 80%, 90%) and test dataset (40%, 30%, 20%, 10%) it got easier to find out best performance for all 3 models. The table represented for Axis bank (Fig 5.5). The ARIMA model was showing the lowest RMSE value for 20% test data evaluation and higher for 40% test data prediction. Similarly, PROPHET model has shown the lowest RMSE value for 10% test data evaluation and higher for 30% test data prediction. And KERAS with LSTM model has shown the lowest RMSE value for 40% test data evaluation and higher for 20% test data prediction. (Lowest RMSE value highlighted with red color)

Fig 5.5 RMSE values analysis for AXIS bank for all 3 models

The table represented for HDFC bank (Fig 5.6). The ARIMA model was showing the lowest RMSE value for 10% test data evaluation and higher for 40% test data prediction. Similarly, PROPHET model has shown the lowest RMSE value for 10% test data evaluation and higher for 40% test data prediction. And KERAS with LSTM model has shown the lowest RMSE value for 20% test data evaluation and higher for 30% test data prediction. (Lowest RMSE value highlighted with red color)

Fig 5.6 RMSE values analysis for HDFC bank for all 3 models
The table represented for ICICI bank (Fig 5.7). The ARIMA model was showing the lowest RMSE value for 10% test data evaluation and higher for 40% test data prediction. Similarly, PROPHET model has shown the lowest RMSE value for 20% test data evaluation and higher for 30% test data prediction. And KERAS with LSTM model has shown the lowest RMSE value for 10% test data evaluation and higher for 40% test data prediction. (Lowest RMSE value highlighted with red color)

![Fig 5.7 RMSE values analysis for ICICI bank for all 3 models](image)

The table represented for KOTAK bank (Fig 5.8). The ARIMA model was showing the lowest RMSE value for 10% test data evaluation and higher for 30% test data prediction. Similarly, PROPHET model has shown the lowest RMSE value for 10% test data evaluation and higher for 40% test data prediction. And KERAS with LSTM model has shown the lowest RMSE value for 40% test data evaluation and higher for 10% test data prediction. (Lowest RMSE value highlighted with red color)

![Fig 5.8 RMSE values analysis for KOTAK bank for all 3 models](image)

The table represented for SBI bank (Fig 5.9). The ARIMA model was showing the lowest RMSE value for 10% test data evaluation and higher for 40% test data prediction. Similarly, PROPHET model has shown the lowest RMSE value for 10% test data evaluation and higher for 40% test data prediction. And KERAS with LSTM model has shown the lowest RMSE value for 30% test data evaluation and higher for 10% test data prediction. (Lowest RMSE value highlighted with red color)

![Fig 5.9 RMSE values analysis for SBI bank for all 3 models](image)
After analyzing RMSE values it was found that most of the models have given the best performance for 20% and 10% test dataset. Hence for the next step which is forecasting, therefore, performed forecasting analysis on 20% and 10% test dataset. Out of 5 banks discussed 2 banks SBI and ICICI bank.

Fig (5.10) (5.11) was shown forecasting for 20% test dataset for SBI and ICICI bank. All 3 models have been drawn in the graph. Red line represents actual forecasting, the blue dotted line represents the PROPHET model, the green line represents the LSTM model and the black line represents the ARIMA model. PROPHET model had shown straight line for forecasting for 20% test dataset whereas LSTM and ARIMA models had shown some trends and patterns for both the banks.
5.1.4 10% Forecasting Analysis

Fig (5.12) (5.13) was shown forecasting for 10% test dataset for SBI and ICICI bank. All 3 models have been drawn in the graph. Red line represents actual forecasting, the blue dotted line represents the PROPHET model, the green line represents the LSTM model and the black line represents the ARIMA model. ARIMA model had shown straight line for forecasting for 10% test dataset whereas LSTM and PROPHET models had shown some trends and patterns for both the banks.

The summarized outcome of this thesis is as below:

1) The performance of each algorithm varies with a different sample dataset.

2) The monthly average accuracy needs to be calculated for each model to predict accurately for all the banks.

3) RMSE is used for measurement as it is a measure which is easy to interpret as well as precise.

4) Backtesting proved to be an effective technique to test the accuracy of Time series prediction models for all datasets.
6 RESULTS AND DISCUSSION

6.1 Result Review
The ARIMA model has been picked for a single time series ought not to change that much and pursue some pattern for order selection. The pattern segment of PROPHET was important to the point that it delivered very nearly a straight line that’s the reason why it didn't diminish auspiciously. The LSTM prediction depended on a lot of last qualities, along these lines less inclined to change because of seasonality and the current pattern.

There were 3 methods (ARIMA, PHOPHET, KERAS with LSTM) applied for 5 Indian banks (AXIS, HDFC, ICICI, KOTAK, and SBI). The performance evaluation parameter RMSE applied after trainset and test set data selection of input and model dependent variables. For seasonal time series, optimized all 3 model parameters simultaneously to compare and discuss the result. Let’s review results for all 3 models.

6.1.1 ARIMA (Autoregressive Moving Integrated Average model)
The ARIMA model had been implemented for 5 Indian banks using train dataset and least AIC (Akaike Information Criteria) value has been calculated to find the accuracy. Least AIC value is highlighted in yellow in the below figure. Train dataset has been divided in such a way (60%, 70%, 80%, and 90%) so that the best result can be evaluated. Accuracy (RMSE) was calculated by the error between population values predicted by a model and the values observed.

<table>
<thead>
<tr>
<th>p</th>
<th>d</th>
<th>q</th>
<th>AIC</th>
<th>p</th>
<th>d</th>
<th>q</th>
<th>AIC</th>
<th>p</th>
<th>d</th>
<th>q</th>
<th>AIC</th>
<th>p</th>
<th>d</th>
<th>q</th>
<th>AIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3267.372</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4333.727</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2136.636</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3114.492</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2964.869</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3696.836</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2144.671</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2893.927</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2723.835</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2679.396</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1984.136</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2598.299</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2187.802</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2672.85</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1650.815</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1995.413</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2190.759</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2622.38</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1648.808</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1951.562</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2122.293</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2617.071</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1649.329</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1952.455</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2119.469</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2546.133</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1654.185</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1956.69</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>2131.456</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>2654.517</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1650.483</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1952.492</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2132.373</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2617.689</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1650.446</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1948.622</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2199.393</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2562.062</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1633.775</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1975.757</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2012.222</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2601.781</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1632.286</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1951.907</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2108.769</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>2603.789</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1632.851</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1982.817</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2101.23</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2507.591</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1632.595</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1966.804</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2015.225</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2603.618</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1632.033</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1932.958</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2015.451</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2586.175</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1632.377</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1934.764</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2103.257</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2607.347</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1633.753</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1932.698</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2105.256</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2602.16</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1632.983</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1934.627</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2106.887</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2583.96</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1635.785</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1936.543</td>
</tr>
</tbody>
</table>

Table 6.1 Least AIC value for all 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)

Here, applied for loop in R programming language using (p, d, q) values between 0 to 2 to find the least AIC value which later helped to find the best RMSE value as an outcome. As per for result, least AIC value has been measured which are for AXIS bank least AIC values are (1, 1, 0), for HDFC bank least AIC values are (2, 1, 2), for ICICI bank is least AIC values are (0, 1, 1), KOTAK bank is (0, 1, 1) and SBI bank is (1, 1, 2). To evaluate accuracy (RMSE) value applied the ARIMA model using these (p, d, q) values.

Accuracy has also been calculated using 4 sets of train dataset and test dataset. Below fig has shown the RMSE value for all 5 banks for 4 sets. Least RMSE value for AXIS bank is 84.52 (80% trainset and 20% test set), HDFC bank is 511.34 (90% trainset and 10% test set), ICICI bank is 36.44 (90% trainset and 10% test set), KOTAK bank is 207.68 (90% trainset and 10% test set) and SBI bank is 25.70 (90% trainset and 10% test set).
Table 6.2 Accuracy (RMSE) values for all 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)

Below fig has shown the graphical representation of RMSE values for all 5 banks where least RMSE value can easily be identified after applying the ARIMA model.

Fig 6.3 Graphical representation of RMSE values for all 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)

Below fig has shown the predicted values for all 5 banks after applying the ARIMA model for 4 trainset and test set.

Table 6.4 Predicted values for 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)
6.1.2 PROPHET
The PROPHET model had been implemented for 5 Indian banks using train dataset and least RMSE value has been calculated to find the accuracy. Train dataset has been divided in such a way (60%, 70%, 80%, and 90%) so that the best result can be evaluated. Accuracy (RMSE) has been calculated by the error between population values predicted by a model and the values observed.

Accuracy has also been calculated using 4 sets of train dataset and test dataset. Below fig has shown the RMSE value for all 5 banks for 4 sets. Least RMSE value for AXIS bank is 57.20 (90% trainset and 10% test set), HDFC bank is 465.50 (90% trainset and 10% test set), ICICI bank is 36.70 (80% trainset and 20% test set), KOTAK bank is 169.30 (90% trainset and 10% test set) and SBI bank is 27.46 (90% trainset and 10% test set).

<table>
<thead>
<tr>
<th>Train</th>
<th>Test</th>
<th>RMSE Values</th>
<th>Train</th>
<th>Test</th>
<th>RMSE Values</th>
<th>Train</th>
<th>Test</th>
<th>RMSE Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>146 (60%)</td>
<td>97 (40%)</td>
<td>85.4952291</td>
<td>167 (60%)</td>
<td>112 (40%)</td>
<td>687.9130796</td>
<td>121 (60%)</td>
<td>80 (40%)</td>
<td>47.59108218</td>
</tr>
<tr>
<td>170 (70%)</td>
<td>73 (30%)</td>
<td>139.4255647</td>
<td>195 (70%)</td>
<td>84 (30%)</td>
<td>589.8793149</td>
<td>141 (70%)</td>
<td>60 (30%)</td>
<td>62.01539325</td>
</tr>
<tr>
<td>194 (80%)</td>
<td>49 (20%)</td>
<td>132.0441463</td>
<td>223 (80%)</td>
<td>56 (20%)</td>
<td>574.7622151</td>
<td>161 (80%)</td>
<td>40 (20%)</td>
<td>36.72500746</td>
</tr>
<tr>
<td>219 (90%)</td>
<td>24 (10%)</td>
<td>57.1710411</td>
<td>251 (90%)</td>
<td>28 (10%)</td>
<td>465.5499221</td>
<td>181 (90%)</td>
<td>20 (10%)</td>
<td>41.16325737</td>
</tr>
</tbody>
</table>

Table 6.5 Accuracy (RMSE) values for all 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)

Below fig has shown the graphical representation of RMSE values for all 5 banks where least RMSE value can easily be identified after applying the PROPHET model.
Below fig has shown the predicted values for all 5 banks after applying the PROPHET model for 4 trainset and test set.

<table>
<thead>
<tr>
<th>Pred values 20</th>
<th>Pred values 20</th>
<th>Pred values 20</th>
<th>Pred values 20</th>
<th>Pred values 20</th>
<th>Pred values 20</th>
<th>Pred values 20</th>
<th>Pred values 20</th>
</tr>
</thead>
<tbody>
<tr>
<td>217.6929486</td>
<td>275.7105767</td>
<td>341.2684104</td>
<td>544.2299769</td>
<td>303.4165298</td>
<td>501.6341194</td>
<td>752.2662429</td>
<td>1226.2684022</td>
</tr>
<tr>
<td>253.1902297</td>
<td>279.5203934</td>
<td>851.1285229</td>
<td>533.8053834</td>
<td>305.6760217</td>
<td>512.2853207</td>
<td>765.4680022</td>
<td>1215.8297018</td>
</tr>
<tr>
<td>248.8911287</td>
<td>279.7811739</td>
<td>357.7692129</td>
<td>557.1904608</td>
<td>305.3644931</td>
<td>509.3767879</td>
<td>777.6067187</td>
<td>1241.5630527</td>
</tr>
<tr>
<td>255.2921311</td>
<td>279.1848905</td>
<td>831.2685164</td>
<td>594.0435302</td>
<td>301.8887775</td>
<td>510.8518940</td>
<td>784.5669816</td>
<td>1257.7236464</td>
</tr>
<tr>
<td>265.4928262</td>
<td>285.7451245</td>
<td>365.2681378</td>
<td>565.7688693</td>
<td>309.7548031</td>
<td>527.6253555</td>
<td>791.1778999</td>
<td>1274.3042503</td>
</tr>
<tr>
<td>261.8580291</td>
<td>285.3513247</td>
<td>360.3959462</td>
<td>599.4720321</td>
<td>317.6950282</td>
<td>539.9712398</td>
<td>796.0679312</td>
<td>1290.7527384</td>
</tr>
<tr>
<td>265.5734115</td>
<td>288.2537346</td>
<td>363.1253906</td>
<td>572.4985905</td>
<td>319.608061</td>
<td>544.3222994</td>
<td>802.0021799</td>
<td>1300.1031071</td>
</tr>
<tr>
<td>270.9013008</td>
<td>291.8527417</td>
<td>368.8047127</td>
<td>573.9153997</td>
<td>322.0928464</td>
<td>579.4207025</td>
<td>799.8129838</td>
<td>1317.3943412</td>
</tr>
<tr>
<td>273.5335588</td>
<td>294.7304521</td>
<td>376.3029254</td>
<td>577.4671281</td>
<td>325.8222981</td>
<td>580.2901253</td>
<td>813.575464</td>
<td>1327.5494949</td>
</tr>
<tr>
<td>274.4772695</td>
<td>295.6081871</td>
<td>377.4843935</td>
<td>594.7011115</td>
<td>328.8337181</td>
<td>599.6378931</td>
<td>822.6149846</td>
<td>1346.0412500</td>
</tr>
</tbody>
</table>

Table 6.7 Predicted values for 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)

6.1.3 KERAS with LSTM (Long short-term memory)
The KERAS with LSTM model had been implemented for 5 Indian banks using train dataset and least RMSE value has been calculated to find the accuracy. Train dataset has been divided in such a way (60%, 70%, 80%, and 90%) so that the best result can be evaluated. Accuracy (RMSE) has been calculated by the error between population values predicted by a model and the values observed.

Accuracy has also been calculated using 4 sets of train dataset and test dataset. Below fig has shown the RMSE value for all 5 banks for 4 sets. Least RMSE value for AXIS bank is 6.31 (60% trainset and 40% test set), HDFC bank is 10.27 (80% trainset and 20% test set), ICICI bank is 3.18 (90% trainset and 10% test set), KOTAK bank is 6.42 (60% trainset and 40% test set) and SBI bank is 3.86 (70% trainset and 30% test set).

<table>
<thead>
<tr>
<th>Train</th>
<th>Test</th>
<th>RMSE Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>145 (60%)</td>
<td>97 (40%)</td>
<td>6.309496112</td>
</tr>
<tr>
<td>170 (70%)</td>
<td>73 (30%)</td>
<td>7.627314237</td>
</tr>
<tr>
<td>194 (80%)</td>
<td>49 (20%)</td>
<td>12.53491778</td>
</tr>
<tr>
<td>219 (90%)</td>
<td>24 (10%)</td>
<td>8.117764154</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Train</th>
<th>Test</th>
<th>RMSE Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>167 (60%)</td>
<td>112 (40%)</td>
<td>15.92855438</td>
</tr>
<tr>
<td>195 (70%)</td>
<td>84 (30%)</td>
<td>18.06708961</td>
</tr>
<tr>
<td>223 (80%)</td>
<td>80 (20%)</td>
<td>10.27311767</td>
</tr>
<tr>
<td>251 (90%)</td>
<td>28 (10%)</td>
<td>15.84698072</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Train</th>
<th>Test</th>
<th>RMSE Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>121 (60%)</td>
<td>80 (40%)</td>
<td>141 (70%)</td>
</tr>
<tr>
<td>161 (80%)</td>
<td>60 (30%)</td>
<td>6.423424443</td>
</tr>
<tr>
<td>181 (90%)</td>
<td>20 (10%)</td>
<td>3.18659373</td>
</tr>
</tbody>
</table>

Table 6.5 Accuracy (RMSE) values for all 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)
Below fig has shown the graphical representation of RMSE values for all 5 banks where least RMSE value can easily be identified after applying the KERAS with LSTM model.

![Graphical representation of RMSE values for all 5 banks](image)

Fig 6.6 Graphical representation of RMSE values for all 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)

Below fig has shown the predicted values for all 5 banks after applying the KERAS with LSTM model for 4 trainset and test set.

![Predicted values for 5 banks](image)

Table 6.7 Predicted values for 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)

<table>
<thead>
<tr>
<th>Predicted values</th>
<th>Predicted values</th>
<th>Predicted values</th>
<th>Predicted values</th>
<th>Predicted values</th>
<th>Predicted values</th>
<th>Predicted values</th>
<th>Predicted values</th>
</tr>
</thead>
<tbody>
<tr>
<td>259.72365</td>
<td>259.72365</td>
<td>259.72365</td>
<td>259.72365</td>
<td>259.72365</td>
<td>259.72365</td>
<td>259.72365</td>
<td>259.72365</td>
</tr>
</tbody>
</table>

6.2 Comparison of Algorithms
In order to compare the results of statistical algorithms, for Axis bank ARIMA model performed well with trainset 80% and test set 20%, whereas PROPHET performed well with trainset 90% and test set 10% followed by LSTM with trainset 60% and test set 40%. Overall LSTM model performed better for all 5 banks.

Almost all 3 models performed better with larger training datasets with the exception LSTM model. ARIMA and PROPHET model performed best with 80% and 90% train dataset whereas LSTM had no trend. LSTM performed differently with each training dataset for all 5 banks.
7 CONCLUSION

This study has presented the extensive procedure of ARIMA model, PROPHET model, and KERAS with LSTM model for stock price prediction. The examinations of these 3 models uncovered that stock data set of all five banks have distinguished by each algorithm. The test results acquired with LSTM model showed the capability to predict stock prices satisfactorily on a short-term basis. This could direct speculators at stock price to settle on gainful investment decisions. With the outcomes acquired LSTM model can contend sensibly well with other methods in short-term prediction.

Overall, PROPHET has the ability to work with large frequency values. The PROPHET model has been picked for a single time series which pursued some pattern for order selection. The ARIMA has a large amount of established time series techniques. The pattern segment of ARIMA was important to the point that it delivered very nearly a straight line. The LSTM prediction depended on a lot of attributes, along these lines less inclined to change because of seasonality and the current pattern. As opposed to that, the PROPHET model has worked admirably demonstrating as an added substance framework, discovering and showing seasonality. With regards to arranging future outstanding workloads, it has been observed that one could now do fine-tuning of the model. At last, keeping in mind that as of now there has been a decent pattern model which is prominent and it was not really dependably need complex machine learning calculations for determining models.

ARIMA and PROPHET generated a consistent model when there was a strong seasonal pattern in the data, whereas the LSTM model was able to identify the type of seasonality in the data and decompose the time series based on the type of seasonality and accurately predicted for all 5 banks. When comparing the statistical models and neural network, seasonal data is better handled by LSTM model than ARIMA and PROPHET, which suggests that LSTM was able to predict data with a strong pattern.

While comparing the RMSE for every model for each bank, it has been observed that the performance of statistical methods differ from Recurrent Neural Network (RNN) method because RNN method is more suitable than statistical models in predicting the stock market returns, which needs to be adapted in a case for stock price prediction.

Future Research Work

The future work of this study will be extending the number of algorithms used for stock price forecasting. The findings from this study can be used for stock price forecasting as below:

- The prediction of each bank’s stock price must be analyzed to identify if there is any trend or seasonality in the data running on larger training datasets.
- Based on the trend or seasonality, a set of traditional statistical and neural network algorithm must be implemented to identify the best algorithm for stock price prediction.
- The back-testing technique must be implemented to evaluate the performance of each of the algorithm.
- The computed error terms can be expressed in RMSE for the ease of interpretation of a business user.
- The best algorithms for each stock price can be identified based on the lowest RMSE value, that algorithm must be used to predict the stock price forecasting and the successful prediction of a stock price might end up with significant profit.
8 PLAGIARISM AND REFERENCES


X. Ding, Y. Zhang, T. Liu, and J. Duan, 2015. Deep learning for event-driven stock prediction.


9 APPENDICES
This document will guide you through the contents of the Artifacts and the necessary steps to implement the R code for dissertation project titled “STOCK PRICE PREDICTION USING TIME SERIES MODELS”.

9.1 Contents of the Artifacts

9.1.1 Datasets
- DataPreparation.R: This R code pre-process the data for the requirement before modeling
- AXISBANK.csv: The pre-processed data file for AXIS bank.
- HDFCBANK.csv: The pre-processed data file for HDFC bank.
- ICICIBANK.csv: The pre-processed data file for ICICI bank.
- KOTAKBANK.csv: The pre-processed data file for KOTAK bank.
- SBIBANK.csv: The pre-processed data file for SBI bank.
- Axis_Monthly_Avg_Share.csv: Monthly average data file for AXIS bank.
- HDFC_Monthly_Avg_Share.csv: Monthly average data file for HDFC bank.
- ICICI_Monthly_Avg_Share.csv: Monthly average data file for ICICI bank.
- Kotak_Monthly_Avg_Share.csv: Monthly average data file for KOTAK bank.
- SBI_Monthly_Avg_Share.csv: Monthly average data file for SBI bank.

9.2 Model Execution

9.2.1 ARIMA
- Arima_AXIS.R: This R code split the data into Trainset and test set and ARIMA model execution for AXIS bank.
- Arima_HDFC.R: This R code split the data into Trainset and test set and ARIMA model execution for HDFC bank.
- Arima_ICICI.R: This R code split the data into Trainset and test set and ARIMA model execution for ICICI bank.
- Arima_KOTAK.R: This R code split the data into Trainset and test set and ARIMA model execution for KOTAK bank.
- Arima_SBI.R: This R code split the data into Trainset and test set and ARIMA model execution for SBI bank.

9.2.2 PROPHET
- Prophet_AXIS.R: This R code split the data into Trainset and test set and PROPHET model execution for AXIS bank.
- Prophet_HDFC.R: This R code split the data into Trainset and test set and PROPHET model execution for HDFC bank.
- Prophet_ICICI.R: This R code split the data into Trainset and test set and PROPHET model execution for ICICI bank.
- Prophet_KOTAK.R: This R code split the data into Trainset and test set and PROPHET model execution for KOTAK bank.
- Prophet_SBI.R: This R code split the data into Trainset and test set and PROPHET model execution for SBI bank.
9.2.3 KERAS with LSTM
- LSTM_AXIS.R: This R code split the data into Trainset and test set and KERAS with LSTM model execution for AXIS bank.
- LSTM_HDFC.R: This R code split the data into Trainset and test set and KERAS with LSTM model execution for HDFC bank.
- LSTM_ICICI.R: This R code split the data into Trainset and test set and KERAS with LSTM model execution for ICICI bank.
- LSTM_KOTAK.R: This R code split the data into Trainset and test set and KERAS with LSTM model execution for KOTAK bank.
- LSTM_SBI.R: This R code split the data into Trainset and test set and KERAS with LSTM model execution for SBI bank.

9.2.4 Model Result
- graphs.R: This R file contains R code for graphs including all 3 models for all 5 banks.
- Results_ARIMA.xlsx: This file contains result for all 5 banks including AIC values, RMSE values, trainset and test set % vise data count, and predicted values.
- Results_Prophet.xlsx: This file contains result for all 5 banks including RMSE values; trainset and test set % vise data count, and predicted values.
- Results_LSTM.xlsx: This file contains result for all 5 banks including RMSE values; trainset and test set % vise data count, and predicted values.
- Combined result RMSE.xlsx: This file contains combined result for RMSE values for all 5 banks.