Comparative study among ARIMA, SARIMA & XGBoost for prediction of NIFTY IT index

Mayuri Kishor Bendale

Supervisor: Mr. Ahmed Makki

Dublin Business School

This presentation is submitted for the degree of

Masters of Science in Business Analytics

January 2024

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Mayuri Kishor Bendale

January 2023

Acknowledgement

My greatest gratitude goes to Mr. Ahmed Makki, his expertise, constant encouragement, and counsel helped me finish the project proposal. I would especially like to express my gratitude to the teachers at the Dublin Business School who have assisted me in expanding my knowledge of business analytics and improving my skills. More than anybody else, my family has played a key role in helping me complete this ambition. I want to express my gratitude to my parents for their love and support in whatever I do. They are the best role models to imitate.

Abstract

Nifty IT index of India stock market is one of the most important yet neglected index when it comes to research and prediction. Prediction of Nifty IT index has benefits would be able to provide foresight and informed decision making to investors, traders, policy makers, etc. as Nifty IT represents IT sector of India. This research is a comparative study between three time series prediction algorithms viz. ARIMA, SARIMA and XGBoost for the most precise forecasting of Nifty IT index. The dataset used for this study has the Nifty IT index data of last 6 years. This time frame covers the dramatic historic moments such as covid-19 pandemic, Russia-Ukraine war, India-Canada tensions and the drastic changes in prices during these events. Three models were hyper parameter tuned and then compared on the basis of three metrices- MSE, RMSE and MAE. Out of the three, SARIMA models seem to have outperformed both ARIMA and XGBoost and hence the conclusion of the study is SARIMA is the most precise algorithm to use for prediction of Nifty IT index out of three.

Contents

List of Figures

List of Tables

List of Important Abbreviation

- ARIMA: Auto Regressive Integrated Moving Average
- SARIMA: Seasonal Auto Regressive Integrated Moving Average
- XGBoost: Extreme Gradient Boosting
- MSE: Mean Squared Error
- RMSE: Root Mean Squared Error
- MAE: Mean Absolute Error
- EDA: Exploratory Data Analysis

1 Introduction

Stock market is a complex and ever-changing mechanism that mirrors the actions and anticipations of traders, businesses, and investors. It is impacted by a number of things, including news, opinions, political developments, and economic situations. Accurately predicting the movement of stock prices is a difficult but worthwhile endeavor since it allows investors to maximize their profits and make well-informed decisions.

Stock market prediction provides a stable foundation for the financial system and benefits a wide range of stakeholders. Using projections of future stock patterns can be quite beneficial for individual investors. Equipped with this insight, they are able to make informed financial choices and strategically time buy and sell orders in order to maximize profits and minimize losses. Furthermore, by using these forecasts, individual investors can improve their risk management techniques and proactively modify their portfolios in reaction to expected market swings. The advantages go beyond investment portfolio optimization for institutional investors and portfolio managers. These organizations are able to strategically deploy resources by anticipating market moves and modifying asset allocations to take advantage of these trends and reduce risks related to particular industries or asset classes. This helps accomplish institutional goals more successfully by matching investing strategies with specific objectives and assisting in exceeding benchmark indexes. NiftyIT index is one of the most well-known and extensively utilized stock market indices in India; it monitors performance of information technology (IT) industry. 10 organisations that are registered on National Stock Exchange (NSE) make up NiftyIT index, which represents roughly 75% of IT industry's overall market capitalization. Among well-known businesses included in NiftyIT index are Tech Mahindra, Infosys, Wipro, Tata Consultancy Services (TCS), and HCL Technologies. IT industry's growth and future potential are reflected in NiftyIT index, which serves as a benchmark.

Because of its unmatched ability to fully understand complex, non-linear correlations present in financial data, machine learning is the most effective method for predicting moves in stock market. at contrast to conventional techniques such as technical or fundamental analysis, machine learning is highly effective at capturing subtle relationships that exist between stock prices and a variety of characteristics, such as news, sentiment, historical data, and indicators. Its strength is in its capacity to learn from large datasets, identify trends, and adjust to changing market conditions—all without need for clear guidelines or preconceived notions. Through the use of advanced algorithms, machine learning gives players in market ability to navigate ever-changing and intricate world of financial markets. It also offers a versatile and strong tool for improving the prediction accuracy of stock market forecasting.

In this research, the movement of NiftyIT index is predicted using machine learning. six steps of data science methodology—business understanding, data collecting, data preparation, data

exploration, data modelling, and data evaluation—have been followed. NiftyIT index has been predicted using three machine learning models: XGBoost, SARIMA, and ARIMA. Two time series models that account for the seasonality and temporal dependence of stock prices are ARIMA and SARIMA. A strong learner is created by combining several weak learners using gradient boosting XGBoost model. Mean squared error (MSE) is a statistic that has been used to evaluate performance of the machine learning models. Target variable's actual and predicted values are compared, and mean squared difference (MSE) is calculated. A model's fit to the data is better when MSE is lower; a greater MSE suggests a worse match. Regression models use mean square error (MSE) to predict a continuous variable. Squared errors for each observation are added together, then the total is divided by total number of observations to get mean square error (MSE). (Day, 2023)

This research work's primary goal is to present a thorough and comparative review of machine learning models for stock market prediction, with an emphasis on NiftyIT index. It is hoped that this study would add to the body of knowledge already in existence and offer insightful information to traders, researchers, and investors.

1.1 Background to the study

One of the most vibrant and significant economic areas in India is information technology (IT) industry, which supports employment, innovation, and national development. The National Stock Exchange of India's (NSE) Nifty IT index is a benchmark stock market index that monitors IT sector's performance. 10 of the top IT businesses, involved in software development, hardware, IT infrastructure, etc., make up index. For investors, analysts, and policymakers alike, Nifty IT index is a valuable tool since it provides insight into actions and patterns of the Indian IT sector. (Bajaj Finance, 2023)

On the other hand, there is a gap in the field of research and practical application of Nifty IT index study and prediction attempts. Analysts may see possibilities and risks, policymakers may create sensible laws and regulations, and investors can make well-informed judgements by forecasting Nifty IT index's moves. So, this study aims to clarify the movements of Nifty IT index and determine which of the three widely used algorithms—ARIMA, SARIMA, and XGBoost—is the most effective in predicting it.

1.2 Problem Statement

There is a compelling problem with the current research gap concerning the implementation of machine learning techniques for predicting NiftyIT index in the Indian stock market. In spite of NiftyIT index's growing significance, there is an obvious absence of thorough research on the application of machine learning algorithms to produce precise forecasts. The lack of such research

is a problem for analysts, investors, and other stakeholders looking for advanced techniques to fully understand and predict the movements of NiftyIT index. In order to close this gap, this project will conduct a methodical analysis of the performance of different machine learning models, ultimately determining which model is most suited for forecasting NiftyIT index. The study will help improve forecasting in the context of Indian stock markets, offering insightful information to decision-makers and promoting a better comprehension of the dynamics affecting NiftyIT index.

1.3 Research Aim

The aim of this research is to employ machine learning models to systematically identify the optimal predictive model for forecasting NiftyIT index in the Indian stock market.

1.4 Objective of the study

Given NiftyIT stock index's vital role in the Indian stock market, the proposed research seeks to address and achieve a number of important goals. Its primary goal is to build a fundamental knowledge of NiftyIT index's behaviour by thoroughly analysing its past performance and trends. The study then seeks to investigate the several elements—market indicators, economic variables, sentiment analysis—that impact NiftyIT index. By doing this, it hopes to identify the complex linkages and relationships that influence the variations in NiftyIT index. In addition, the study aims to utilise advanced algorithms using machine learning for predicting future movements of NiftyIT index, ultimately determining the most efficient predictive model. By fulfilling these goals, the research hopes to offer insightful information to stakeholders, analysts, and investors, advancing a more sophisticated comprehension of the dynamics pertaining to NiftyIT index within the framework of the Indian stock market. The following lists the study's aims:

1. Analyze the historical performance of NiftyIT index to establish the understanding of its behavior.
2. Train the machine learning models based on the past data of NiftyIT Index.
3. Identify the best suited machine learning algorithm for the prediction of NiftyIT Index.
4. Implement the practical time series forecasting model in order to address the problem.
5. Find the best set of hyperparameters for machine learning models.
6. Implement the visualizations for the easy understanding of results.

1.5 Research Questions

The study aims to pursue following research questions by performing the comparative study methodology:

1. Which model will outperform out of ARIMA, SARIMA and XGBoost to forecast Nifty IT index.
2. What are the optimal hyperparameter configurations for each model, and how does the sensitivity of these models to hyperparameters impact their forecasting accuracy for NiftyIT index?

1.6 Research Motivation

The idea for this study came from a critical observation in the field of financial research: there is a big gap in the study of stock market measures that directly represent important parts of the Indian economy. Specifically, Information Technology (IT) sector, which is a key part of India's economic strength, doesn't get the prediction research it needs. At the center of this industry representation is NiftyIT index, which is a key measure of how well IT companies traded on the National Stock Exchange (NSE) are doing. The study's inspiration came from the observation that, despite the fact that IT industry plays a key role in India's economic growth, there is a noticeable lack of research on how to accurately predict NiftyIT index.

The primary objective is to address this gap and enhance the common comprehension of the dynamic relationship between IT industry and the Indian stock market. In an ever-changing technology environment, precise and timely predictions of NiftyIT index are crucial for a range of stakeholders, such as investors, financial analysts, and regulators. The objective of this study is to meet this urgent need by using sophisticated prediction models such as ARIMA, SARIMA, and XGBoost. The primary objective is to improve the prediction capabilities of NiftyIT index and provide practical insights that may guide investment strategies, risk management choices, and policy considerations about the dynamic and significant IT industry in India.

1.7 Research Rationale:

The reason for this research project is that there is a major gap in the present state of financial research when it comes to showing and predicting stock market indices for Information Technology (IT) sector in India. In terms of economic progress and global competition, Information Technology business is one of the most important parts of the country's economy. NiftyIT index, which is meant to reflect the success of IT companies traded on the National Stock

Exchange (NSE), has not gotten enough attention when it comes to systematic forecast analysis, even though it is very important.

The idea behind this study came from the realization that people who have a stake in NiftyIT index, like investors, financial experts, and lawmakers, need to know a lot about it. Since IT industry is still a big part of India's economic growth, being able to correctly guess NiftyIT index's trends and moves is very important for making smart business decisions. By filling in this study gap, we hope to give important information about the complicated workings of IT industry in the Indian stock market. This will help people who use accurate assessments to make investment choices and policy decisions in the long run.

The decision to use advanced forecasting models like ARIMA, SARIMA, and XGBoost comes from the desire to use cutting-edge methods to make estimates more accurate and reliable. The point of this study isn't just to fill in a gap in knowledge; it's also to give stakeholders the useful and practical information they need to effectively manage the complicated Indian stock market, especially in IT sector. The study's goals are to add something useful to the body of research, improve the ability to predict the future, and help people make better decisions about NiftyIT index and the Indian IT business as a whole.

1.8 Research Significance

The importance of this research project is based on its ability to fill in a major knowledge gap and make important contributions to both scholarly study and real-world financial decision-making. India's economic growth depends on Information Technology (IT) sector, and NiftyIT measure is a key sign of how healthy that sector is. However, there hasn't been a lot of study done on predicting NiftyIT index. This makes it hard to understand how this important part of the Indian stock market works.

It becomes clear how important this study is when it can give useful information to many various organizations. Predictions that come true are very important for investors, financial experts, and politicians to make smart choices. This study aims to improve NiftyIT index's ability to predict the future by using advanced forecasting models such as ARIMA, SARIMA, and XGBoost. The study's results will give owners more accurate information to help them make smart investment choices, handle their portfolios, and lower their risk in the fast-paced IT field.

This is also important for academics because it adds to the body of research on methods for predicting the stock market, especially when it comes to sector-specific statistics. As technology becomes more important in countries around the world, this study's results may not only affect the Indian stock market, but also show how prediction modeling can be used to understand the differences between sectors and help make decisions in other global markets.

Basically, this study is important because it could fill in a significant need in financial research, give useful information to stakeholders, and add important information to the academic

community. This will help us learn more about the complicated relationship between India's IT industry and its stock market.

2 Literature Review

The method of developing a “stock price prediction model using the ARIMA (Autoregressive Integrated Moving Average) model” is presented in detail in paper "Stock Price Prediction Using ARIMA Model" by Ayodele A. Adebiyi, Aderemi O. Adewumi, and Charles K. Ayo. To create their prediction model, authors used publicly available stock data from the Nigeria Stock Exchange (NSE) and the New York Stock Exchange (NYSE). The study highlights importance of stock price prediction in finance and economics and how it has inspired scientists to create increasingly accurate prediction models throughout time. This paper examines the ARIMA model, which is well-known for its effectiveness and robustness in financial time series forecasting, particularly for short-term prediction. Results showed that the ARIMA model can compete well with current methods for stock price prediction and has a great deal of potential for short-term prediction. The article adds that prediction will remain an interesting area of research, giving organizations and people the ability to choose wisely when to invest and create practical plans for their present and future undertakings. The present study's effectiveness with ARIMA model highlights the model's potential for use in stock price prediction. (Ayodele Ariyo Adebiyi, 2014)

In order to predict the closing price of NIFTY 50 index, two deep learning techniques—Long Short-Term Memory (LSTM) and Backward Elimination LSTM (BE-LSTM)—are compared in paper "Forecasting of NIFTY 50 Index Price by Using Backward Elimination with an LSTM Model" by Syed Hasan Jafar, Shakeb Akhtar, Hani El-Chaarani, Parvez Alam Khan, and Ruaa Binsaddig. The authors forecasted the closing price using 15 years' worth of daily data that they had acquired from Bloomberg, taking into account factors like date, high, open, low, close volume, and 14-period relative strength index (RSI). The findings demonstrated that, with a 95% prediction accuracy, the BE-LSTM model beat LSTM model in predicting the price of NIFTY 50 index over ensuing 30 days. The research concludes that suggested model greatly enhanced NIFTY 50 index price predicted, highlighting BE-LSTM model's potential for stock market prediction. (Syed Hasan Jafar, 2023)

A deep learning-based model and a thorough customization of feature engineering are presented in the paper "Short-term stock market price trend prediction using a comprehensive deep learning system" by Jingyi Shen & M. Omair Shafiq in order to forecast stock market price trends. The authors gathered two years' worth of data from Chinese stock market and put forth a solution that consists of pre-processing stock market dataset, applying various feature engineering approaches, and integrating a deep learning-based system that is specifically designed to anticipate stock market price trends. The study highlights importance of stock price prediction in finance and economics and how it has inspired scientists to create increasingly accurate prediction models throughout time. The authors found that their suggested method performs better than others because of thorough feature engineering they built after conducting extensive assessments on popular machine learning models. The method predicts stock market trends with an overall excellent level of accuracy. This work adds to stock analysis research community in both

financial and technical sectors with its thorough design and evaluation of prediction term lengths, feature engineering, and data pre-processing techniques. (Jingyi Shen, 2020)

A technique for predicting stock prices using XGboost algorithm model is presented in Yifan Zhang's paper, "Stock Price Prediction Method Based on XGboost Algorithm." Based on the daily time series features of stocks, author forecasts stock prices using the XGboost algorithm model, whose parameters are managed by GridSearchCV search algorithm. The study highlights how important stock price prediction is to finance and economics and how it has inspired scholars to create increasingly accurate prediction models throughout time. The model results show that by applying a computer algorithm to analyze vast amounts of data related to stock prices, model can better capture high-frequency time series fluctuation trend of stock while controlling for overfitting and underfitting, leading to more accurate prediction results regarding stock price. The study concludes that using algorithms to estimate price of financial assets is more than just data analysis; it has a sound scientific foundation that is reinforced by current trend in behavioral finance. The XGboost algorithm's performance in this study highlights its potential for use in stock price prediction. (Zhang, 2023)

The ARIMA and LSTM models are used in paper "Stock Price Prediction Under Anomalous circumstances" by Jinlong Ruan and Wei Wu to forecast stock prices in unexpected circumstances. The authors constructed a typical ARIMA model using consecutive stock prices, looking for locations where forecasts considerably differed from actual values to discover outliers. At level of individual stocks, industry level, and general market level, they trained LSTM and ARIMA models, respectively. Additionally, authors included sentiment analysis in models in form of sentiment ratings, which are derived from Reddit comments regarding particular companies. The study highlights how unpredictable and complicated stock market could turn out to be, particularly in 2020 because of several local and worldwide "black swans," such as COVID-19 pandemic. In a single week from March 9 to 16, U.S. stock market threw circuit breaker three times—a record in history. Additionally, stock values of individual firms fell at rates that no prior forecasting model could have anticipated. With an average prediction accuracy of 98%, models can be utilized to enhance current prediction techniques. The study concludes that there were insufficient models available to accurately forecast how stock prices would fluctuate in event of catastrophic or extremely rare occurrences. (Ruan, et al., 2022)

2.1 Ways of predicting the stock market:

One can guess what the stock market prices will do in a number of ways. These are some of the most popular ways to do it:

1. Fundamental Analysis: This method looks at a company's financial records, business trends, and global factors to figure out what company is really worth. It is then checked to see if the stock is overvalued or economical by comparing its true value to its present market price. (Aditya Birla Capital, 2021)

2. Technical Analysis: This approach looks at past market data, such as price and volume, to find patterns and trends that can be used to predict how prices will move in future. Technical experts use charts and other tools to find trend lines, levels of support and resistance, and other clues that help them make smart buying decisions. (Union College, n.d.)
3. Quantitative Analysis: This method uses statistical methods and mathematical models to look at market data and make predictions about how prices will change in future. A lot of the time, machine learning techniques like decision trees, artificial neural networks, and support vector regression are used in quantitative analysis. (Union College, n.d.)
4. Momentum Trading: When this approach is used, one buys stocks that have been doing well and sell stocks that have not been doing well. Following the idea behind momentum investing is that stocks that have done well in past are likely to keep doing well, while stocks that have not done well in the past are likely to keep doing poorly. (TRISTAN YATES, 2022)

Patterns and trends can be found by using famous method of technical analysis, which predicts how prices will move in future by looking at past market data such as price and volume. Machine learning (ML) can make technical analysis better because it can automatically find patterns and trends in very large datasets.

2.2 Benefits of using machine learning algorithms:

1. The accuracy is better because machine learning systems can look at a lot of data and find trends that humans would miss. This could lead to more correct predictions about how prices will move in the future.
2. Faster Analysis: Machine learning systems can look at large files a lot faster than people can. Traders can use this right now to help them make better decisions.
3. Reduce bias: Human researchers may be influenced by their biases when they're looking at data. Computer programs that use machine learning don't care about these kinds of flaws, so they can give more objective information.
4. The power to change: As new data comes in, machine learning systems can change their predictions. Because of this, they are perfect for studying markets that are always changing.
5. Finding complicated Patterns: Algorithms that use machine learning can find complicated trends in data that human researchers might miss. People who trade can use this information to find buying/selling chances and make smarter decisions.

2.3 Significance of NiftyIT index:

NiftyIT index is a standard for stock market that measures how well the Indian IT industry is doing. There is a group of top IT companies that make up this index. They are all listed on National Stock Exchange of India (NSE). Because the free-float market capitalization weighted approach is used to make index, the weight of each stock in it is based on its free-float market capitalization. Nifty IT measure is a well-known one in Indian stock market. It lets investors keep an eye on how IT sector of India is doing.

2.4 Advantages of using 'Close' price:

Using the "Close" price to predict stock prices over time has a number of benefits. The "close" price of a stock is last day's deal price for that company. The "Close" price is helpful for time series predictions in the following ways:

1. Accessibility: Traders can easily get to "Close" price. It is available in the public domain without any processing on row data.
2. Reflects Sentiment of the Market: The "Close" price shows how market sentiment is, at the end of trading day. It takes in account all the news, changes, and possible effects that could have on stock price during the day.
3. Stable: The "Close" price is a reliable way to see how well a stock is doing. In comparison, it is less volatile than other signs like "Open" or "High" prices.
4. Used in Technical Analysis: The "close" price is often used in technical analysis to find patterns and trends in how stock prices move. With the help of charts and other tools, technical analysts can find trend lines, levels of support and resistance, and other clues that can help them make smart trade decisions.

2.5 Choice of algorithms – ARIMA, SARIMA, XGBoost

Prominent machine learning techniques including XGBoost, SARIMA, and ARIMA are utilized for stock price time series forecasting. Several factors make these algorithms superior to alternative machine learning methods, including:

2.5.1 ARIMA

A class of linear models called ARIMA, or Autoregressive Integrated Moving Average, forecasts future values using historical data. Technical analysis frequently uses ARIMA to predict future

security prices. It is especially helpful for short-term forecasting and can be applied to data anomalies, trends, and seasonality modeling. Because ARIMA is simple to use and may yield precise predictions for a variety of data sets, it is a popular option for time series forecasting. (ADAM HAYES, 2023)

2.5.2 SARIMA

Data seasonality trends are taken into consideration by Seasonal Autoregressive Integrated Moving Average (SARIMA), an extension of ARIMA. SARIMA is especially helpful for cycle-containing complicated data space forecasting. For a variety of data sets, this robust method can produce precise predictions. (Aayush Bajaj, 2023)

2.5.3 XGBoost

Time series forecasting makes extensive use of the machine learning technique Extreme Gradient Boosting (XGBoost). Strong algorithms like XGBoost can manage large, multidimensional, complex data sets. It is very helpful for long-term forecasting and can be applied to data anomalies, trends, and seasonality modeling. Because it can produce precise forecasts for a variety of data sets, XGBoost is a well-liked option for time series forecasting.

In conclusion, because they are simple to use, robust against large amounts of data, and capable of producing precise forecasts for a variety of data sets, ARIMA, SARIMA, and XGBoost are well-liked machine learning algorithms for time series forecasting of stock prices.

3 Research Methodology

In recent years, it has become more and more common to make stock market predictions using historical data and time series forecasting machine learning algorithms. A method called time series forecasting makes predictions about a time series' future values based on past data. The stock market uses it extensively to forecast future stock prices. Time series forecasting can be accomplished using a variety of machine learning techniques, for example- Extreme Gradient Boosting (XGBoost), Long Short-Term Memory (LSTM), Seasonal AutoRegressive Integrated Moving Average (SARIMA), and AutoRegressive Integrated Moving Average (ARIMA). These algorithms are able to identify patterns in past data and forecast future stock market trends based on those patterns. The accuracy with which future stock prices can be predicted by algorithms depends on the model's complexity and quality of past data.

It is important to note that while these algorithms can be useful for predicting future stock prices, they are not perfect. This is due to the fact that a wide range of variables, including company performance, political developments, and overall economic conditions, have an impact on stock market and it's difficult to forecast just based on previous data. Consequently, organizations and researchers work constantly to improve the precision and decrease error of these algorithms. To increase precision of stock market forecasts, new algorithms are being created and research in this field is also underway. The proposed study is a step in the same direction.

Since Nifty IT index provides investors with a diverse basket of IT sector firms, examining index is more advantageous than analyzing individual stocks. This is because investing in individual stocks carries a higher risk. Ten IT-related stocks listed on National Stock Exchange (NSE) of India make up Nifty IT index. The index is intended to give investors a standard by which to compare the performance of Indian stock market's IT sector. The free-float market capitalization weighted approach is used to create the index.

For above reason, Nifty IT index of National Stock Exchange is being used and studied in this work. In order to study Nifty IT index of National Stock Exchange, the data science lifecycle has been followed. This lifecycle gives systematic step-by-step approach of addressing problem.

The lifecycle offers an organized method for handling analytical challenges, which facilitates better comprehension of the issue and selection of best course of action. It also ensures that the appropriate data is used for analysis and that the appropriate models are created to address problem, which helps to increase the analysis's accuracy. By using lifecycle in an organization, one can do informed decisions based on the predictions.

3.1 System Flowchart

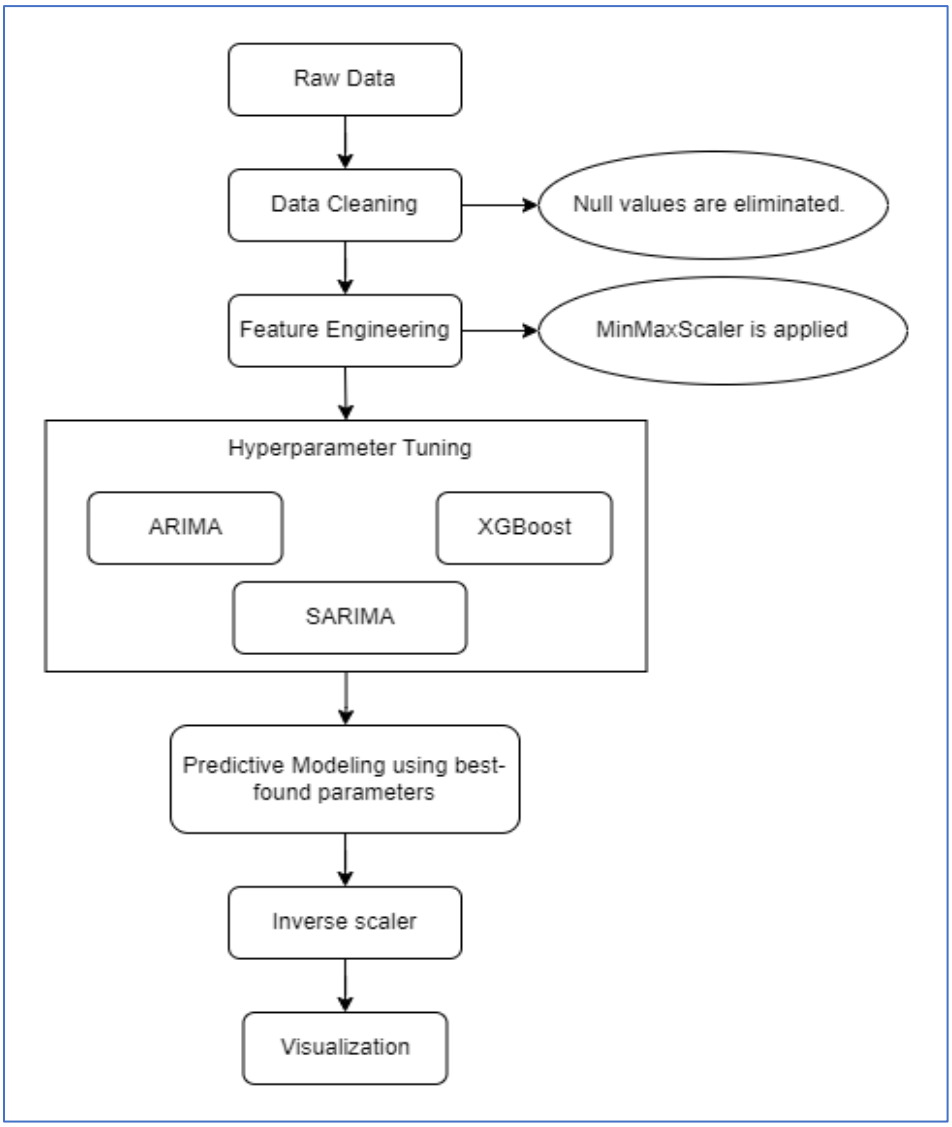

Figure 3-1 Flowchart of the implemented system

In above diagram, a well-organized way to look at and predict Nifty IT stock index can be seen. The work is based on dataset from Nifty IT stock index that is stored in a CSV file. During the cleaning process, any null or missing numbers are taken out of data. This step is necessary to make sure the data is correct and reliable before it can be used for more research.

The process then moves on to feature engineering, where MinMaxScaler is used to make the dataset normal. By setting numbers in the dataset to a range between 0 and 1, this process makes sure that they all have same scale. This is especially important when working with market index data, since the amounts of points can be very different.

After designing the features, next step is to tune the hyperparameters. This is process of fine-tuning algorithms like ARIMA, SARIMA, and XGBoost to find the best settings that accurately predict market trends for Nifty IT stock. These models are often used in time series forecasting, which is related to predicting stock indexes.

After prediction model has been run with the best-found parameters, an inverted scaler is used to change result back to its original scale. This has to be done because the data was already scaled down during feature engineering step. Lastly, the visualization process turns data into a picture that makes them easier to understand and interpret.

3.2 Data Science Lifecycle

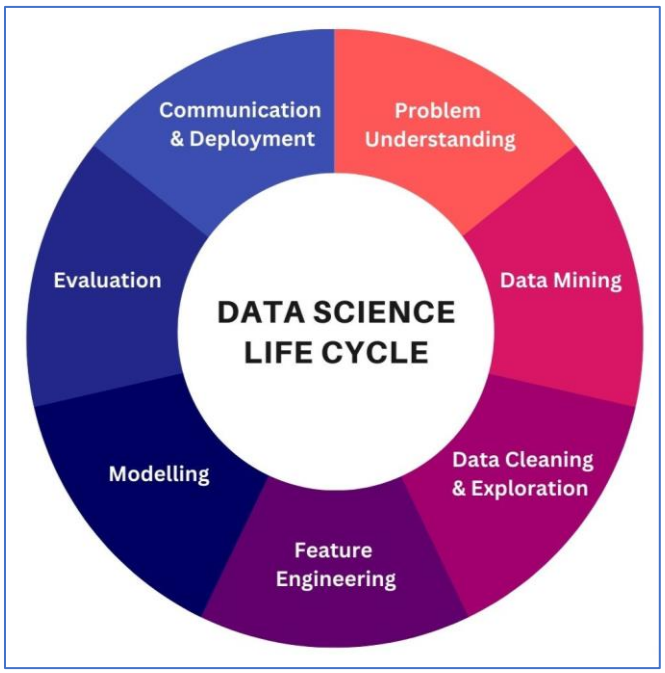

Figure 3-2 Data Science Life Cycle

Using a variety of methods and resources to examine past data, spot trends, and formulate well-informed forecasts, Data Science can be applied to forecast changes in the stock market. For implementation of time series forecasting of Indian stock market index Nifty IT, the implementation goes through steps of data science lifecycle. Firstly, the problem is well defined. The research questions are fixated. In next phase, data mining, data is downloaded from Yahoo finance. The stock market index data is available on Yahoo Finance in daily, weekly, monthly fashion. The project is implemented using data in weekly fashion. For each row there are Date, Open, High, Low, Close, Adjacent Close, etc. values are available. Date serves as index to the dataset. Each occurrence in dataset is representation of index price varying throughout week. Later, data cleaning is performed by deleting the columns that are not required. (HAYES, 2021) For that reason, other columns such as Open, High, Low, Adjacent close are deleted and only

Close column is focused. After deleting columns, rows are cleaned by deleting rows which contained null value. For the feature engineering, all values of close are converted within 0 to 1 magnitude. This means the lowest close value is considered as zero and highest close value is considered as 1. The values lying between highest and lowest are resized to relatively fit between 0 and 1. By doing this, data is scaled onto same level. This project experiments with three popular and robust time series forecasting algorithms viz, ARIMA, SARIMA, XGBoost. These algorithms are first tested on the smaller set of same dataset. They are hyperparameter tuned to find the best parameters suitable for dataset. After tunning the parameters, models are trained with full dataset. Then they are tested by making forecast and its precision is checked using Root Mean Square Error (RMSE).

3.2.1 Problem Understanding

This is one of most important steps in data science and machine learning. During problem understanding, the exact requirements of research are identified. Research questions are used as blueprint for requirements extractions. In the case of this project, further contributing to field of stock market predictions and time series forecasting in general goal.

More specifically, programs are written to identify differences in forecasting performances variety of forecasting algorithms. From literature review it was identified that ARIMA, SARIMA, XGBoost would be the best fit for prediction of stock market analysis.

Nifty IT index of Indian stock market is taken as input for the forecasting algorithms. Nifty IT index consists of top 10 IT companies of India. In other words, this index gives over all bullish and bearish movements of IT sector in India.

3.2.2 Data Acquisition

Nifty IT Index data consists of top 10 IT companies of India hence it can be considered as one of the best Indexes to represent IT sector of India. This list includes companies such as (Trading Hat, 2023)Nifty IT Index data is downloaded in CSV format from Yahoo Finance (Yahoo Finance, n.d.). Yahoo provides option to download data in the CSV format.

The downloaded dataset consists of weekly data of Nifty IT index from date 25/10/2017 to 23/10/2023. That is past 6 years of data. Total number of rows are 1475. This file consisted following columns- Date, Open, High, Low, Close and Adjacent Close.

The data is inconsistent at a few places. The rows where Close price is missing, those are deleted from the dataset.

3.2.3 Data Cleaning

The original dataset had entries like Date, Open, High, Low, Close and Adj Close. During data science cleaning stage, Volume and Adj Close were taken out and to make sure the data was correct, null numbers were also carefully taken out. After being cleaned, dataset now focuses on important financial measures i.e. Date, Open, High, Low and Close. This was done by getting rid of useless data and making the dataset better for further research.

Following table shows remaining columns and its datatypes:

Column Name	Datatype
Date	Datetime
Open	Float
High	Float
Low	Float
Close	Float

Table 3-1 Column datatypes

Following screenshot shows the head of imported dataset into program for processing. Head method of dataframe returns the first 5 rows of imported dataset.

```
                 Open          High           Low         Close
Date
2017-10-25  10829.79981  10986.50000  10799.34961  10925.45020
2017-10-26  10915.65039  10945.45020  10836.00000  10883.04981
2017-10-27  10901.29981  10911.65039  10831.25000  10862.04981
2017-10-30  10892.54981  10913.90039  10842.95020  10854.29981
2017-10-31  10793.09961  10870.54981  10781.00000  10837.90039
```

Figure 3-3 Dataset content

3.2.4 Feature Engineering

SKlearn's MinMaxScaler was only used on 'Close' feature during feature engineering. The dataset's relative links were kept when the MinMaxScaler changed "Close" numbers to a scaled range of 0 to 1. The normalization method makes sure that feature scores are always the same. This keeps any one characteristic, like "Close," from having too much of an effect on later studies. As a consequence of use of MinMaxScaler, dataset is now prepared for machine learning algorithms that are sensitive to varying feature sizes. This, in turn, leads to modeling results that are more dependable and accurate.

The MinMaxScaler class is imported from the SKlearn library and then implemented as follows:

```
scaler = MinMaxScaler() # default=(0, 1)
numerical = ['Close']
df[numerical] = scaler.fit_transform(df[numerical])
print(df.head())
```

Figure 3-4 Scaler Implementation

Following screenshot shows the effect of MinMaxScaler on the Close column of the dataset:

Date	Open	High	Low	Close
2017-10-25	10829.79981	10986.50000	10799.34961	0.003788
2017-10-26	10915.65039	10945.45020	10836.00000	0.002303
2017-10-27	10901.29981	10911.65039	10831.25000	0.001567
2017-10-30	10892.54981	10913.90039	10842.95020	0.001296
2017-10-31	10793.09961	10870.54981	10781.00000	0.000721

Figure 3-5 Effect of MinMaxScaler on Close

3.2.5 Model Implementation

The dataset subset was analysed using a variety of time series forecasting models, including XGBoost, SARIMA (Seasonal ARIMA), and ARIMA (AutoRegressive Integrated Moving Average), during Module Evaluation phase. For every model, different combinations of hyperparameters were methodically used to investigate a wide range of options. Finding ideal combination of characteristics that produced the most exact and accurate predictions was goal. The Root Mean Squared Error (RMSE), a metric that measures the variations between predicted and observed values and provides a quantitative indicator of model's predictive accuracy, was the evaluation criterion used for model performance assessment. This all-encompassing method made sure that every possible model configuration was thoroughly explored, which eventually resulted in the selection of best forecasting model.

3.2.5.1 ARIMA (AutoRegressive Integrated Moving Average)

A time series forecasting technique called the ARIMA (AutoRegressive Integrated Moving Average) model combines moving average, differencing, and autoregressive components. It forecasts future values by using historical observations (AR), correcting for stationarity (I), and modelling short-term volatility (MA). Its scientific effectiveness is in capturing temporal patterns, making data stationary, and accounting for short-term variations. It is represented as ARIMA (p, d, q), where "p" is autoregression order, "d" is differencing degree, and "q" is moving average order. By optimising these components, it can make accurate predictions in a variety of temporal datasets. (Visual Design, 2022)

$$y_t^* = \Delta^d y_t$$

$$y_t^* = \mu + \underbrace{\sum_{i=1}^{p} \phi_i y_{t-i}^*}_{\text{AR}} + \underbrace{\sum_{i=1}^{q} \theta_i \epsilon_{t-i}}_{\text{MA}} + \epsilon_t$$

Expression of an ARIMA(p,d,q) model. We denote **y** the series of observations, **ϕ** and **θ** the parameters for the AR and MA parts of the model respectively, **ϵ** the prediction errors, **μ** the intercept parameter, and **Δ** the differencing operator

Equation 3-1 ARIMA Equation

3.2.5.2 SARIMA (Seasonal ARIMA)

An advanced method for time series forecasting that builds on ARIMA by adding seasonality is called the Seasonal AutoRegressive Integrated Moving Average (SARIMA) model. It combines seasonal counterparts with moving average, differencing, and autoregressive components. Symbolised as SARIMA (p, d, q) (P, D, Q, s), it comprises seasonal parameters (P, D, Q) and a seasonal interval "s." SARIMA improves prediction accuracy in time-dependent data by addressing seasonality-induced trends. (Visual Design, 2022)

$$(1-\phi_1 B)\,(1-\Phi_1 B^4)(1-B)(1-B^4)y_t = (1+\theta_1 B)\,(1+\Theta_1 B^4)\varepsilon_t.$$

Equation 3-2 SARIMA Equation

3.2.5.3 XGBoost

As a scientific way to make predictions as exact as possible, XGBoost is a strong regression model. It's a group activity for learning. XGBoost creates a chain of decision trees that fix each other's mistakes one by one by using regularization and gradient boosting notions. It stops things from fitting too well by lowering a loss function and adding control through regularization terms to make things more complicated. XGBoost can handle missing numbers, links that are hard to understand, and very big data sets well thanks to its gradient-based optimization method. When it comes to machine learning and predictive modeling, this regression method is strong and useful because it can balance model complexity, reduce bias, and raise variance.

$$L(y_i, p_i) = \frac{1}{2}(y_i - p_i)^2$$

Equation 3-3 XGBoost Equation

3.2.6 Hyperparameter Tunning:

"Hyperparameter tuning" is the methodical process of finding best set of hyperparameters for a computing model. In contrast to learning from data, hyperparameters are outside settings that are set before training. For example, learning rates, tree levels in various methods, and the power of regularization are some examples. Hyperparameter tuning is process of figuring out which set of hyperparameters should be changed to get the best results from a model. Nowadays, more complex methods like Bayesian optimization or tactics like grid search or random search are used to do this. Before finding setup that improves performance metrics on a validation set, the model needs to be trained and tested with different sets of hyperparameters. Adjusting hyperparameters correctly can make a model much more accurate and useful for new data.

An approach to setting hyperparameters known as "grid search" carefully examines a set grid of hyperparameter values. Within the grid, there is every possible pair of models hyperparameter values. The model is trained and tested using cross-validation for each combo. The best performance test results are used to choose set of hyperparameters. Many hyperparameters and a wide range of possible values can make grid search very computationally expensive, but it ensures a full exploration of the hyperparameter space. Unlike grid search, random search picks random combinations of hyperparameters from a set search area. Picking a set number of

combinations at random, training and testing model for each, and then finding the best set of hyperparameters is what random search does instead of carefully looking at every possible combination. Random search is very helpful when search space is big because it efficiently explores by focusing on hyperparameter pairs that are more likely to be ideal. Using randomness instead of grid search can help find high-performing combinations faster, especially when certain hyperparameters have a more noticeable effect on model performance. (Hestisholihah, 2023) (Linmarsirait, 2023)

While hyperparameters were being tuned, grid search was the main optimization method used in this project. To look into different combos, Python's for loops were used to step through a grid of set hyperparameter values. For every round, a different set of hyperparameters were used to train machine learning model, and cross-validation was used to see how well it did. Different numbers for things like learning rates, regularization strengths, and tree levels were shown on the grid, which covered hyperparameter space.

The for loops, which went through every possible combination, made it possible to do a full study of how the model behaved across intended hyperparameter grid. This long search wanted to find the setup that gave best performance measure. It made sure that the hyperparameter space was thoroughly and carefully looked into. Since number of possible hyperparameters and steps goes up rapidly, it's important to think about how much it will cost to run this method. Grid search with for loops gives openness and clarity in hyperparameter improvements, even though it is hard to compute. This helps choose the best model setup for this project.

3.2.6.1 MSE (Mean Squared Error):

The Mean Squared Error (MSE) is a widely used metric in regression issues, such as time series forecasting, to assess how accurate forecasts are. It measures mean squared difference between the actual and expected values. The MSE calculation is:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Equation 3-4 MSE Equation

When n is number of observations, yi denotes the actual value at time i, and yi denotes what was predicted at the same time i. When differences between expected and actual values are squared, larger errors are highlighted, making mean square error (MSE) sensitive to both small and large deviations.

The reason that MSE works so well for time series forecasting evaluation is that it restricts forecast errors based on how big of an inaccuracy they are. Model-state error (MSE) offers a complete measure of how effectively a model reproduces the observed values across time in time series data, where catching trends and patterns is critical. Squaring errors also increases the MSE's sensitivity to outliers, a crucial feature in time series analysis since extreme values can greatly affect how well predictions turn out. The square root of MSE (RMSE), on other hand, is frequently employed for a more comprehensible scale because it tends to amplify the impact of outliers and is expressed in squared units of original data.

The following set of hyperparameters are used in ARIMA:

```
# Hyperparameter tuning - Grid Search Approach
p_values = [0,2,4]
d_values = [1,2,3]
q_values = [1,2,3]
```

Figure 3-6 ARIMA Hyperparameter ranges

The following table shows the combinations of hyperparameters supplied to ARIMA model and MSE (Mean Square Error) yielded by each combination:

p	d	q	Mean Square Error (MSE)
0	1	1	0.000645
0	1	2	0.000648
2	1	3	0.000649
4	1	1	0.000649
2	2	2	0.000653
2	1	1	0.000657
0	2	1	0.000659
0	2	2	0.000661
4	2	1	0.000663
2	1	2	0.000664

Table 3-2 Hyperparameters against MSE

The following set of hyperparameters are used in SARIMA:

```
#HyperParameter tuning
seasonal=[16,52]  #Seasonality
p_params = [0,1]  #0 to 5
d_params = [1]  #1 or 2
q_params = [0,1]  #0 to 5
P_params = [1,2]  # 0 to 2
D_params = [0]  # 0 or 1
Q_params = [0,1]  # 0 to 2
m_params = seasonal
```

Figure 3-7 SARIMA Hyperparameters ranges

The following table shows the combinations of hyperparameters supplied to SARIMA model and MSE (Mean Square Error) yielded by each combination:

p	d	q	P	D	Q	m	MSE
0	1	1	1	0	0	52	0.045922
1	1	1	1	0	0	52	0.046446
0	1	1	1	0	0	16	0.04762
1	1	1	1	0	0	16	0.048194
0	1	1	1	0	1	16	0.049955
0	1	1	2	0	0	52	0.049996
1	1	1	2	0	0	52	0.050781
0	1	1	2	0	1	52	0.05083
1	1	1	1	0	1	16	0.050932
1	1	1	2	0	1	52	0.052321

Table 3-3 SARIMA parameters against MSE

As it can be seen in the above table, when (p,d,q) is (0,1,1) and (P,D,Q,m) is (1,0,0,52) seems to be giving the smallest SME value.
The following set of hyperparameters are used in XGBoost:

```
##HyperParameter tuning - Grid Search approach
Max_depth = [4,10,15]
Gamma = [0,0.25,0.5,1]
N_estimator = [2,5,10]
```

Figure 3-8 XGBoost Hyperparameters

The following table shows the combinations of hyperparameters supplied to XGBoost model and MSE (Mean Square Error) yielded by each combination:

Max Depth	Gamma	N-estimator	MSE
4	0.5	2	0.001108
4	0	2	0.001547
4	0.25	2	0.001561
10	0	2	0.001824
15	0	2	0.001888
4	1	2	0.003352

Table 3-4 XGboost parameter against MSE

As it can be seen in the above table, when Gamma 0.5, Max Depth 4, N-estimator 2 seems to be giving the smallest SME value.

3.2.7 Predictive Modeling

Three different types of While the prediction modeling is going on, ARIMA, SARIMA, and XGBoost are used to look at Nifty IT dataset. Next, you should look at the "Close" column, which displays Nifty IT index's final numbers. Closing prices are important for predicting future trends and changes in IT index, which is why this choice was made on purpose.

The "Close" column is then standardized with MinMaxScaler, a common step in machine learning preparation that sets the data's range, usually between 0 and 1. It stops any one model from doing better than the others because of differences in input feature scale by making sure that all models are trained on data with the same scaling.

After cleaning, TrainTestSplit method is used to split the dataset into training and testing sets. This split is necessary for checking how well models work on data that hasn't been tried yet so that you can get a good measurement of their ability to generalize. The next step is to teach each model (ARIMA, SARIMA, and XGBoost) how to find patterns and trends in the data using past "Close" values from the training set.

After learned models are used on the test set, estimates are made. It is possible to assess and understand model results by converting the expected values back to their original scale using opposite of the MinMaxScaler. For useful comparisons to be possible and to understand how accurate models are expected to be when used with the original information, this step is needed.

Using reversed scaler, you can clearly see the difference between correct "Close" numbers from the test set and what was predicted. There is a structured and methodical way to do predictive modeling and evaluation with Nifty IT dataset that uses this all-encompassing process. It includes model training, prediction, inverse scaling, selected feature selection, and normalization.

3.2.8 Evaluation

As seen in predictive modeling phase, all three algorithms ARIMA, SARIMA, XGBoost are made to work on the actual full-sized dataset by providing best-found parameters during hyper parameter tuning phase. The values that are predicted by these algorithms are then stored into the an excel file to further process on it. Further processing involves visualization etc. Next step into process is to evaluate the predictions made by each of algorithms to see how close their predictions are from the actual Nifty IT prices.

In beginning of the process, MinMaxScaler was applied on raw data to transform the values of ‘Close’ into scale of 0 to 1. These values now need to be again converted back to original values using Inverse Scaler function in python. Following piece of code converts the 0 to 1 scaled value to original scaled values.

```
#Inverse MinMaxScaler

predictions = np.array(predictions).reshape(-1, 1)
inverse_predictions = scaler.inverse_transform(predictions)

test = np.array(test).reshape(-1, 1)
inverse_test = scaler.inverse_transform(test)
```

Figure 3-9 Inverse MinMaxScaler to get original scale

It can be seen in the above screenshot that, both the test and predicted values are converted back to original scale. Once the values are scaled back to original scale, they can be visualized using the line chart.

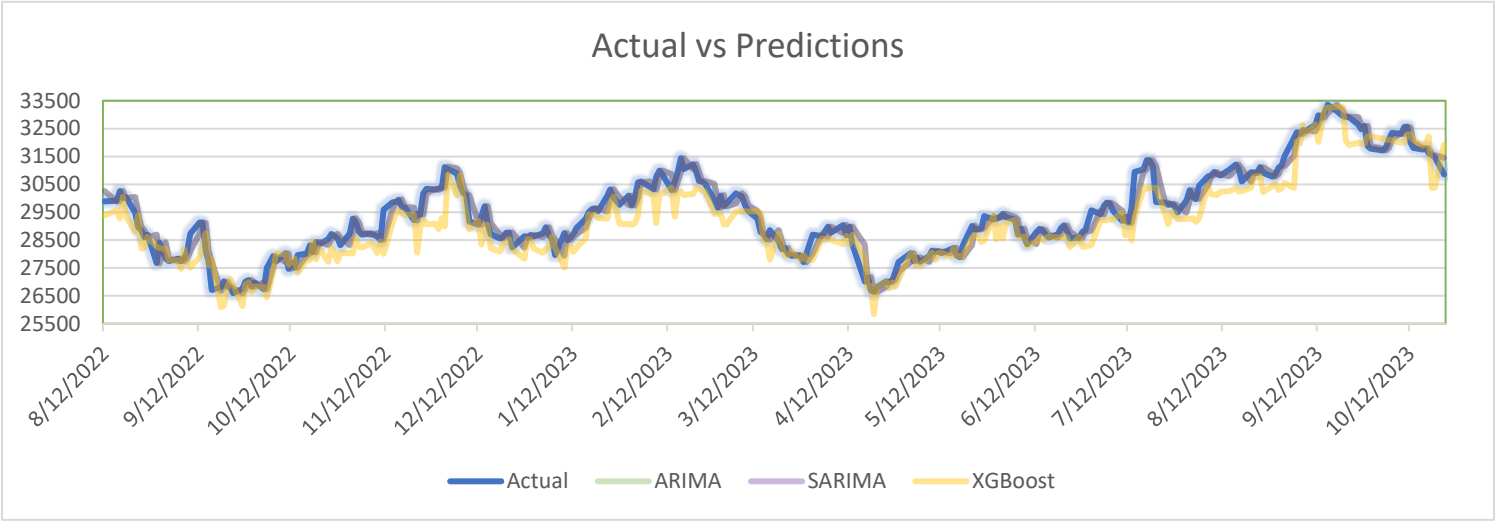

Figure 3-10 Line plot of actual prices vs predicted prices by all models

Above graph shows comparison between the actual prices and the prices predicted by the ARIMA, SARIMA and XGBoost models executed with the best-fitted parameters. The graph consists of four-line plots. Each line represents actual prices, prices predicted by ARIMA, prices predicted by SARIMA and prices predicted by XGBoost, respectively. Prices predicted by ARIMA, SARIMA and XGBoost are in green, purple and yellow semitransparent color. Whereas actual prices are denoted by solid blue color. Date is there on the X-axis and price is on the Y-axis. Date range starts at 8/12/2022 and ends at 10/12/2023. Prices range from 25500 to 33500. The line plot of predicted values is purposely kept as semitransparent because the lines of predicted prices overlap the actual price line. So, in order to understand the movement of actual price, the line plot for it is in solid color and rest of the plots are in semitransparent.

On the first glance at the graph, it could appear that graph only has three lines out of four. The reason behind this is, ARIMA and SARIMA predictions are almost similar with slight deviation at every point. Hence on graph with 250+ data points it looks like ARIMA and SARIMA exactly overlaps one another. On the other hand, XGBoost can be seen performing worst with the predictions.

Chart below shows the zoomed in version of the above graph. It is only has plotting of ARIMA and SARIMA predictions. The zoomed in version is shown in order to see the minute difference between the two.

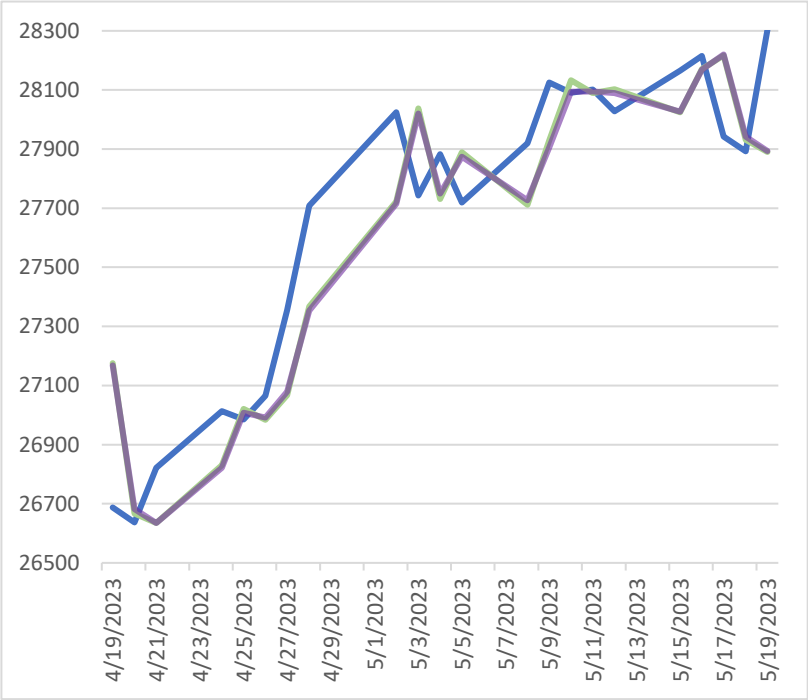

Figure 3-11 Zoomed in price displacement of ARIMA and SARIMA

Same as the previous graph, blue represents actual prices, green represents ARIMA and SARIMA is denoted by purple color. In above graph, at some places green line (ARIMA) can be seen peeking out at some points. For examples, at the first occurrence actual prices valued above 28100, ARIMA line also crossed the 28100 mark but SARIMA line can be seen below the 28100 price.

Following the line plot, precision of the models is calculated using the statistical analysis called as MSE (Mean Squared Error) and RSME (Root Mean Squared Error). MSE and RSME are calculated for the actual and predicted values of each of the models, as shown in the tables below:

MSE	
ARIMA	117191.2
SARIMA	117049
XGBoost	374975.6

Table 3-5 MSE of all models

Above table shows the MSE values for each of the models. It can be seen that the SARIMA returned the lowest MSE meaning the highest precision in terms of prediction. ARIMA performed second best and XGBoost retuned highest MSE meaning it is worst fit for this particular problem.

RMSE	
ARIMA	342.332
SARIMA	342.1242
XGBoost	612.3525

Table 3-6 RSME of all the models

After MSE, RMSE is calculated in order to scale the MSE values close to actual error values in scale. In the above table, ARIMA and SARIMA both models seem to be performing with similar precision. Difference between them is fractional. However, SARIMA is better between them. XGBoost is tuned out to be twice erroneous than ARIMA and SARIMA models.

MAE	
ARIMA	254.7458
SARIMA	254.4746
XGBoost	483.5492

Table 3-7 MAE of all models

After RMSE, MAE has been calculated. MAE stands for mean absolute error. This is another method of calculating the error in predictions. With MAE as well, SARIMA is out-performing ARIMA and XGBoost. It is out performing ARIMA by marginal difference but it is twice as precise as XGBoost.

4 Conclusion

Predicting the stock market price movements is a quite challenging task. Using machine learning algorithms for this reason is quintessential. Apart from being the tedious and repetative task, machine learning algorithms can help to find hidden patterns, relationships, etc within the data. When looked at time series forecasting models, they make predictions based on the historic data.

Nifty IT index of Indian stock market consists of top ten IT companies of India and around 75% volume of stocks traded in IT industry are consisting of companies that are part of Nifty IT index. So, Nifty IT index incidentally also represents IT industry of India. The dataset chosen for the implementation was starting from year 2017 in order to train the models on the historic events such as Covid-19 pandemic during year 2020, emergence of Russia Ukraine war in 2022, raised tensions between India and Canada in 2023. The models also accommodate these historic events while training.

Once the data was acquired from Yahoo finance, it was then subsequently cleaned, feature engineering was done on the data. After this, three time series forecasting algorithms are chosen for the implementation of this project viz ARIMA, SARIMA and XGBoost. A subset of dataset was extracted and train-tested on each of the algorithms to find best performing hyperparameters. After finding the hyperparameters, they are supplied to respective models to make prediction using the full dataset.

For the evaluation of best algorithm out of three, the predictions made by these algorithms are compared with the actual values. MSE, RSME and MAE matrices are used for calculating the error in the prediction for each of the models. Upon comparison it was found out that out of three models, SARIMA fits best for the forecasting of Nifty IT index.

5 Future work

The approach used for the implementation was to focus on prediction of 'Close' price by taking into account the historic data of only itself. The problem with this approach is there is only one input variable to the algorithm. For more precise forecasting, it is recommended to include other factors as well in the process of training the algorithm and predicting. These inputs can include the factors that affects the price movement of stocks. Here are some examples of the factors- news sentiments can be analyzed to provide it as an input, sentiments of twitter trends can be analyzed, fundamental analysis of company can be provided as input, etc. Adding such factors could probably put out better and more precise prediction results.

6 Bibliography

TRISTAN YATES, 2022. *4 Ways to Predict Market Performance.* [Online] Available at: https://www.investopedia.com/articles/07/mean_reversion_martingale.asp [Accessed December 2023].

Aayush Bajaj, 2023. *ARIMA & SARIMA: Real-World Time Series Forecasting.* [Online] Available at: https://neptune.ai/blog/arima-sarima-real-world-time-series-forecasting-guide [Accessed December 2023].

ADAM HAYES, 2023. *Autoregressive Integrated Moving Average (ARIMA) Prediction Model.* [Online]
Available at: https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp
[Accessed December 2023].

Aditya Birla Capital, 2021. *Prediction of Stock Price Movements: 4 Strategies.* [Online] Available at: https://www.adityabirlacapital.com/abc-of-money/4-strategies-in-prediction-of-stock-price-movements
[Accessed December 2023].

Ayodele Ariyo Adebiyi, A. A. C. A., 2014. *Stock price prediction using the ARIMA model.* Cambridge University, United Kingdom, Research Gate.

Bajaj Finance, 2023. *Nifty IT - Everything you need to know.* [Online] Available at: https://www.bajajfinservsecurities.in/blog/nifty-it-overview/ [Accessed December 2023].

Day, U., 2023. *Understanding Mean Squared Error (MSE) in Regression Models.* [Online] Available at: https://medium.com/@wl8380/understanding-mean-squared-error-mse-in-regression-models-9ade100c9627
[Accessed December 2023].

HAYES, A., 2021. *What Is Closing Price? Definition, How It's Used, and Example.* [Online] Available at: https://www.investopedia.com/terms/c/closingprice.asp

Hestisholihah, 2023. *Hyperparameter Tuning Showdown: Grid Search vs. Random Search — Which is the Ultimate Winner?.* [Online] Available at: https://medium.com/@hestisholihah01/hyperparameter-tuning-showdown-grid-search-vs-random-search-which-is-the-ultimate-winner-5927b322e54d
[Accessed December 2023].

Jingyi Shen, M. O. S., 2020. Short-term stock market price trend prediction using a comprehensive deep learning system. *Journal of Big Data,* Issue 7, p. 66.

Linmarsirait, 2023. *Hyperparameter Tuning in Machine Learning Using Bayesian Optimization.* [Online]
Available at: https://medium.com/@linmarsirait2/hyperparameter-tuning-in-machine-learning-using-bayesian-optimization-8ee522ef6d99
[Accessed December 2023].

Ruan, J., Wu, W. & Luo, J., 2022. Stock Price Prediction Under Anomalous Circumstances.

Syed Hasan Jafar, S. A. ,. E.-C. P. A. K. a. R. B., 2023. Forecasting of NIFTY 50 Index Price by Using Backward Elimination with an LSTM Model. *Journal of Risk and Financial Management,* Volume 16(10), p. 423.

Trading Hat, 2023. *Nifty IT Stocks List 2023 with weightage.* [Online]
Available at: https://www.tradinghat.com/nifty-it-stocks/

Union College, n.d. *Methods of Stock Market Prediction.* [Online]
Available at: https://muse.union.edu/2019capstone-hladikl/methods-of-stock-market-prediction-2/
[Accessed December 2023].

Visual Design, 2022. *Time Series Analysis - ARMA, ARIMA, SARIMA.* [Online]
Available at: https://www.visual-design.net/post/time-series-analysis-arma-arima-sarima
[Accessed December 2023].

Yahoo Finance, n.d. *NIFTY IT (^CNXIT) History.* [Online]
Available at: https://finance.yahoo.com/quote/%5ECNXIT/history

Zhang, Y., 2023. Stock Price Prediction Method Based. *ICBBEM 2022,* p. 595–603.