****

**PERFORMANCE EVALUATION AND ANALYSIS OF MACHINE LEARNING ALGORITHMS IN STOCK PRICE PREDICTION**

Gaurab Shrestha

230188112

Submitted in Accordance with the Requirements for the Degree of

Master of Science (Level 7)

York St John University

**Department of Computer Science**

Sangita Pokharel

Date of Submission: 28th January 2025

**ABSTRACT**

Forecasting stock price is a difficult yet highly sought-after endeavour in financial markets because of its potential for substantial financial rewards and its influence on investment strategies and economic planning. The growing intricacy of financial markets and the abundance of large datasets have rendered machine learning a crucial tool for stock price predictions. This research explores the efficacy of various machine learning techniques in forecasting stock prices, concentrating on their precision and computational efficiency. Utilizing two datasets—Google stock prices and Nabil Bank stock prices—the study assesses different machine learning models including Linear Regression, Support Vector Machines, Random Forest, XGBoost, Long Short-Term Memory, Convolutional Neural Networks, a hybrid CNN-LSTM approach, alongside the statistical SARIMAX and FB Prophet models. Basic ML models like LR and SVM consistently outperformed more complex algorithms, especially on the volatile and noisy Nabil Bank dataset, where advanced machine learning techniques struggled with issues of overfitting and generalization. The investigation also emphasizes the effect of training methodologies, noting that the train\_test\_split() method from scikit-learn produced superior outcomes compared to training on the initial 80% of data, as the former offers a more balanced representation. Although deep learning models are proficient in capturing non-linear and temporal dynamics, their effectiveness is significantly influenced by the size, quality of the dataset, and meticulous tuning. This study highlights the necessity of matching model selection with the characteristics of the dataset and the strategies used for training. It concludes that simpler algorithms frequently yield better results in scenarios involving limited or noisy data, and suggests that incorporating external elements like news sentiment or social media trends could improve prediction models in the future.

***Keywords: Stock Price, machine learning, NEPSE, SVM, CNN, LSTM.***

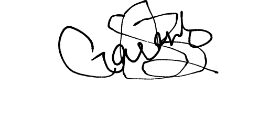
**DECLARATION**

I certify that this dissertation is my own original work, and that any language from other sources is enclosed in quotation marks, with proper acknowledgment given for the use of others’ language, ideas, expressions, or writings.

I affirm that I have not plagiarized from any source, nor engaged in any form of copying, including outsourcing any part of this work (such as unauthorized proof-reading), nor have I fabricated, altered, or exaggerated data in the completion of this dissertation.

I declare that this dissertation presents original research that has not been submitted for any other degree at any educational institution.

I state that the inclusion of AI in this dissertation does not compromise the academic integrity of the work and adheres to the regulations established by York SJ University. Any help provided by Artificial Intelligence (AI) tools, including but not limited to research, composition, or editing, has been recognized and utilized following ethical standards.



Gaurab Shrestha

28th January 2025

**ACKNOWLEDGEMENT**

I would like to convey my heartfelt appreciation to everyone who has offered their invaluable support and assistance in the preparation of this report. This work would not have been achievable without their contributions.

I am truly thankful to my supervisor, Sangita Pokharel from the Department of Computer Science, for her expert guidance, unwavering encouragement, and willingness to devote time from her busy schedule for the progress reviews of my dissertation. Her constant inspiration has enabled me to complete this project on time and meet its objectives.

I also wish to express my sincere gratitude to Prashant Bikram Shah, of the Department of Computer Science, for his ongoing motivation and support. Additionally, I would like to convey my heartfelt appreciation to all the faculty members for their encouragement and assistance.

The unwavering support from my classmates has been crucial, as they have motivated and assisted me directly or indirectly in the completion of this work. Their constructive feedback and encouragement have been essential to my success in this dissertation project.

Finally, I want to thank everyone who played a role, either directly or indirectly, in contributing to this report. The help and encouragement I received from various individuals have been vital in bringing this project to fruition.

I appreciate all of you for your support and for being a part of this journey.



Gaurab Shrestha

230188112

gaurab.shrestha@yorksj.ac.uk

**Table of Contents**

[**1.** **INTRODUCTION** 11](#_Toc183987995)

[**1.1** **Background:** 11](#_Toc183987996)

[**1.2** **Problem Statement and Motivation:** 13](#_Toc183987997)

[**1.3** **Objectives:** 14](#_Toc183987998)

[**1.4** **Research Questions and Hypothesis:** 14](#_Toc183987999)

[**1.5** **Dissertation Structure:** 15](#_Toc183988000)

[**2.** **LITERATURE REVIEW** 16](#_Toc183988001)

[**2.1** **Traditional Methods and Their Limitations** 16](#_Toc183988002)

[**2.2** **Machine Learning Approach in Stock Price Prediction:** 16](#_Toc183988003)

[**2.2.1** **Basic Machine Learning Approaches for Prediction** 17](#_Toc183988004)

[**2.2.2** **Deep Learning Approaches along with other ML Algorithms** 17](#_Toc183988005)

[**2.2.3** **Hybrid Models and Ensemble Approaches** 19](#_Toc183988006)

[**2.2.4** **Performance Metrics for Evaluation** 20](#_Toc183988007)

[**2.3** **Research Gap:** 20](#_Toc183988008)

[**3.** **Methodology** 21](#_Toc183988009)

[**3.1** **Project Overview** 21](#_Toc183988010)

[**3.2** **System Design and Methodology Overview** 22](#_Toc183988011)

[**3.2.1** **Project Flowchart** 22](#_Toc183988012)

[**3.3** **Data Understanding** 23](#_Toc183988013)

[**3.3.1** **Data Collection** 24](#_Toc183988014)

[**3.3.2** **Data Wrangling (Data Preprocessing and Feature Engineering)** 26](#_Toc183988015)

[**3.3.3** **Exploratory Data Analysis (EDA)** 30](#_Toc183988016)

[**3.3.4** **Data Splitting** 37](#_Toc183988017)

[**3.4** **Model Building** 38](#_Toc183988018)

[**3.4.1** **Basic Machine Learning Regression Algorithms** 38](#_Toc183988019)

[**3.4.2** **Ensemble Learning Algorithms** 41](#_Toc183988020)

[**3.4.3** **Time Series Forecasting Algorithms** 43](#_Toc183988021)

[**3.4.4** **Deep Learning Algorithms** 46](#_Toc183988022)

[**3.5** **Hyperparameter Tuning** 50](#_Toc183988023)

[**3.5.1** **Tuning Parameters Used in Different Algorithms** 51](#_Toc183988024)

[**3.6** **Performance Evaluation Metrics** 51](#_Toc183988025)

[**3.6.1** **Mean Squared Error (MSE)** 51](#_Toc183988026)

[**3.6.2** **Root Mean Square Error (RMSE)** 52](#_Toc183988027)

[**3.6.3** **Mean Absolute Error (MAE)** 52](#_Toc183988028)

[**3.6.4** **Computational Time (Training Time)** 53](#_Toc183988029)

[**3.7** **Project Implementation** 53](#_Toc183988030)

[**3.7.1** **Implementation Details** 53](#_Toc183988031)

[**3.7.2** **Implementation Challenges** 54](#_Toc183988032)

[**4.** **Results, Analysis and Discussion** 55](#_Toc183988033)

[**4.1** **Results** 55](#_Toc183988034)

[**4.1.1** **Result from experiment on Google (GOOGL) Stock Price Dataset** 55](#_Toc183988035)

[**4.1.2** **Result from experiment on Nabil Bank (NABIL) Stock Price Dataset** 63](#_Toc183988036)

[**4.2** **Analysis** 70](#_Toc183988037)

[**4.2.1** **Performance of Different Models** 70](#_Toc183988038)

[**4.2.2** **Impact of Dataset Characteristics** 71](#_Toc183988039)

[**4.2.3** **Effect of Training Methodology** 71](#_Toc183988040)

[**4.2.4** **Why Deep Learning Underperformed** 71](#_Toc183988041)

[**4.2.5** **Effect of Stock Price Trends** 71](#_Toc183988042)

[**4.3** **Discussion** 72](#_Toc183988043)

[**5.** **Conclusion** 73](#_Toc183988044)

[**5.1** **Summary of the Work** 73](#_Toc183988045)

[**5.1.1** **Overview** 73](#_Toc183988046)

[**5.1.2** **Linkage to Objectives** 73](#_Toc183988047)

[**5.2** **Reflection** 74](#_Toc183988048)

[**5.3** **Future Work** 74](#_Toc183988049)

[**5.3.1** **Research Limitations** 74](#_Toc183988050)

[**5.3.2** **Recommendations for Future Research and Improvements** 75](#_Toc183988051)

[References 77](#_Toc183988052)

**List of Tables**

[Table 1: Overview of the Dataset 28](#_Toc183988053)

[Table 2: Performance Metrics of Different Algorithms in Google Stock Price Dataset 56](#_Toc183988054)

[Table 3: Performance Metrics of Different Algorithms in Nabil Bank Stock Price Dataset 63](#_Toc183988055)

**List of Figures**

[Figure 1: Candlestick Representation of Stock Price (Hopman, 2021) 11](#_Toc183988056)

[Figure 2: Stock Movement Prediction Methods Efficacy (Kumar, 2021) 12](#_Toc183988057)

[Figure 3: Workflow of Stock Price Prediction Model using Machine Learning (Kumbure, et al., 2022) 21](#_Toc183988058)

[Figure 4: System Design of Stock Price Prediction 22](#_Toc183988059)

[Figure 5: Model Training Flowchart of Stock Price Prediction 23](#_Toc183988060)

[Figure 6: Code to Download Google Stock Price Data from Yahoo Finance 24](#_Toc183988061)

[Figure 7: Importing and Setting Up Selenium Driver 25](#_Toc183988062)

[Figure 8: Code for Scrapping Nabil Bank Stock Price Data 25](#_Toc183988063)

[Figure 9: Descriptive Statistics of Google Dataset 27](#_Toc183988064)

[Figure 10: Descriptive Statistics of Nabil Bank 27](#_Toc183988065)

[Figure 11: Dropping and Renaming the Column Names 28](#_Toc183988066)

[Figure 12: Boxplot of Google Stock Price Dataset for Outliers Detection 29](#_Toc183988067)

[Figure 13: Boxplot of Nabil Stock Price Dataset for Outliers Detection 30](#_Toc183988068)

[Figure 14: Changing dtypes of Columns of Nabil Dataset 30](#_Toc183988069)

[Figure 15: Google Stock Price Trend over Time 31](#_Toc183988070)

[Figure 16: Nabil Stock Price Trend over Time 31](#_Toc183988071)

[Figure 17: Probability Distribution of Close Price of Google and Nabil respectively 32](#_Toc183988072)

[Figure 18: Google Historical Annual Stock Price Data 32](#_Toc183988073)

[Figure 19: Nabil Historical Annual Stock Price Data 33](#_Toc183988074)

[Figure 20: Google Stock High and Low-Price Comparison 33](#_Toc183988075)

[Figure 21: Nabil Stock High and Low-Price Comparison 33](#_Toc183988076)

[Figure 22: Top 20 Highest Share Volumes Traded in a Day 34](#_Toc183988077)

[Figure 23: Daily Volume Average of the Stock 34](#_Toc183988078)

[Figure 24: Google Stock Price with 20, 50, 100-Day Moving Average 35](#_Toc183988079)

[Figure 25: Nabil Stock Price with 20, 50, 100-Day Moving Average 35](#_Toc183988080)

[Figure 26: Google Stock Price: Candlestick Representation 36](#_Toc183988081)

[Figure 27: Nabil Stock Price: Candlestick Representation 36](#_Toc183988082)

[Figure 28: Google Stock Price Variable Correlation 37](#_Toc183988083)

[Figure 29: Nabil Stock Price Variable Correlation 37](#_Toc183988084)

[Figure 30: Different Train-Test Dataset Splitting Methods 38](#_Toc183988085)

[Figure 31: Algorithms Used for Predicting Stock Price 38](#_Toc183988086)

[Figure 32: Graph Showing Simple Linear Regression and Multiple Linear Regression (Uddin, 2023) 39](#_Toc183988087)

[Figure 33: Support Vector Regressor Graph (Otten, 2024) 40](#_Toc183988088)

[Figure 34: Single Decision Tree of Random Forest 42](#_Toc183988089)

[Figure 35: Flowchart showing Boosting Ensemble Learning (XGBoost) 43](#_Toc183988090)

[Figure 36: FB Prophet Logo (Ankit, 2022) 45](#_Toc183988091)

[Figure 37: Layers of CNN 47](#_Toc183988092)

[Figure 38: Components of LSTM (Calzone, 2022) 49](#_Toc183988093)

[Figure 39: Performance Comparison of Different Model in Google Stock Price Data (train\_test\_split vs first 80% train) 57](#_Toc183988094)

[Figure 40: Training Time for Different Machine Learning Models in Google Stock Price Dataset 58](#_Toc183988095)

[Figure 41: Result from Linear Regression (train\_test\_split() vs first 80% train) 59](#_Toc183988096)

[Figure 42: Result from SVM (train\_test\_split() vs first 80% train) 59](#_Toc183988097)

[Figure 43: Result from Random Forest (train\_test\_split() vs first 80% train) 60](#_Toc183988098)

[Figure 44: Result from XGBoost (train\_test\_split() vs first 80% train) 60](#_Toc183988099)

[Figure 45: Result from SARIMAX (train\_test\_split() vs first 80% train) 61](#_Toc183988100)

[Figure 46: Result from FB Prophet 61](#_Toc183988101)

[Figure 47: Result from CNN (train\_test\_split() vs first 80% train) 62](#_Toc183988102)

[Figure 48: Result from LSTM (train\_test\_split() vs first 80% train) 62](#_Toc183988103)

[Figure 49: Result from CNN-LSTM (train\_test\_split() vs first 80% train) 63](#_Toc183988104)

[Figure 50: Performance Comparison of Different Model in Nabil Stock Price Data (train\_test\_split vs first 80% train) 65](#_Toc183988105)

[Figure 51: Training Time for Different Machine Learning Models in Nabil Bank Stock Price Dataset 65](#_Toc183988106)

[Figure 52: Result from Linear Regression (train\_test\_split() vs first 80% train) 66](#_Toc183988107)

[Figure 53: Result from SVM (train\_test\_split() vs first 80% train) 66](#_Toc183988108)

[Figure 54: Result from Random Forest (train\_test\_split() vs first 80% train) 67](#_Toc183988109)

[Figure 55: Result from XGBoost (train\_test\_split() vs first 80% train) 67](#_Toc183988110)

[Figure 56: Result from SARIMAX (train\_test\_split() vs first 80% train) 68](#_Toc183988111)

[Figure 57: Result from FB Prophet 68](#_Toc183988112)

[Figure 58: Result from CNN (train\_test\_split() vs first 80% train) 69](#_Toc183988113)

[Figure 59: Result from LSTM (train\_test\_split() vs first 80% train) 69](#_Toc183988114)

[Figure 60: Result from CNN-LSTM (train\_test\_split() vs first 80% train) 70](#_Toc183988115)

**List of Abbreviations**

# **INTRODUCTION**

## **Background:**

According to (Statista, 2024), the total global market capitalization of all stock markets exceeded $111 trillion till 2023. This figure represents the overall worth of publicly listed companies around the globe. The U.S. stock market, bolstered by exchanges such as the New York Stock Exchange and Nasdaq, makes up almost 46.2% of the total value of the world's stock markets. In contrast, China's Shanghai Stock Exchange and Shenzhen Stock Exchange, Japan's Tokyo Stock Exchange, and the London Stock Exchange are significant contributors to global stock market capitalization.

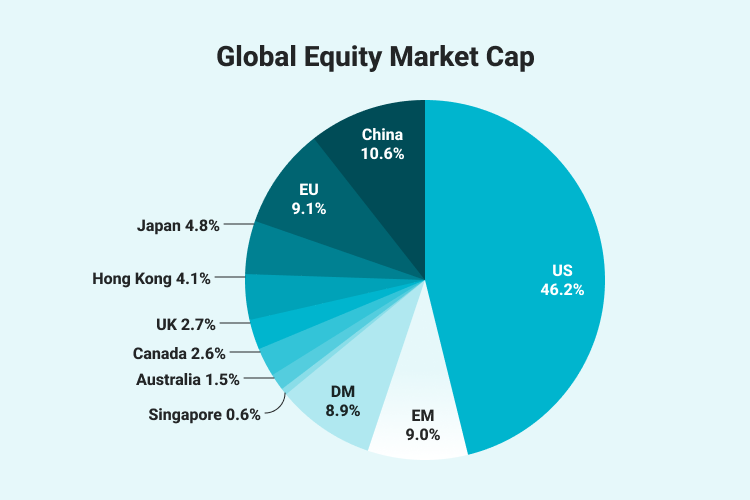


Figure 1: Global Equity Market Cap (Sahm Investor Education, 2023)

In an article by (Bajracharya, 2019), he highlighted that there is a growing fascination with stocks and shares, as people from various backgrounds repeatedly inquire, “Which shares do you suggest purchasing in the Nepalese Stock market at this moment?” In the late 1800s, Charles Dow, a co-founder of the Wall Street Journal and Dow Jones & Company, became one of the pioneers in forecasting stock prices. He developed Dow Theory, which suggests that stock prices move in trends that can be predicted by analysing patterns of market highs and lows. So basically, a stock is simply a capital raised by any company through the issue and subscription of shares whose value is dynamic throughout the time whereas the stock market is the combined platform of several markets and exchangers where various investors sell and purchase according to the availability of those stocks (Tretina, 2024). In basic terms, the stock market is where individuals can purchase and trade shares of companies, along with currencies, equities, derivatives, and more. A nation's financial market plays a crucial role in determining its overall condition, helping economists and analysts assess the country's economic health (Kabir, et al., 2023). Well, with the growth of broad financial market, nowadays, people are more concerned about buying and selling stocks, trading in forex and so on. But, with less knowledge and idea about the stock market, most of the people are losing their money. So, there comes the concept of stock price prediction. Stock market investors have long been developing strategies to predict stock prices and maximize their profits. The forecasting of future stock price movements has captured the interest of researchers for many years.

Over time, people have relied on traditional methods, such as fundamental and technical analysis, to predict stock prices. Fundamental analysis examines factors like a company's intrinsic value found in annual and quarterly reports, financial ratios, government policies, and the impact of national and international events, as well as political and economic developments. Key intrinsic values often analysed include Earnings Per Share (EPS), Price-to-Book Value (PBV), Book Value (BV), Price-to-Earnings (P/E) Ratio, recent bonus percentages, and total market capitalization. It actually considers the value of the company and generally helps to anticipate the value of its assets and the profits it can generate ultimately making an impact on the supply and demand of the stock. But fundamental analysis can’t anticipate what might happen in the short term. Technical analysis emphasizes trading volume, patterns, charts, and trends, along with daily price movements and historical price data. Its main focus is on price movements, offering insights into the supply and demand forces that drive stock prices. This method is primarily used for short-term predictions. One of the major benefits of using technical analysis over fundamental analysis is that the stocks can be analysed quickly and identify the price targets. Some people also buy and sell the stocks according to the market sentiment which include herd behaviour, media stories, rumours, recommendations. But sometimes, due to some reason, some stocks might create hype in the market that will affect in longer or shorter period of time.

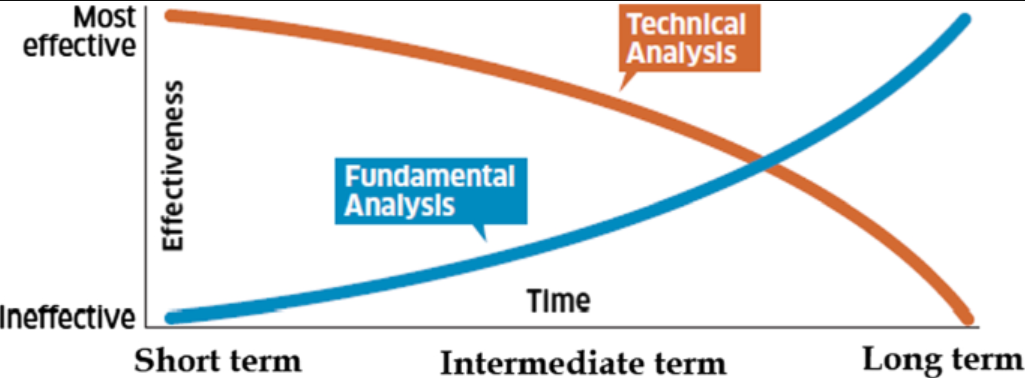


Figure 2: Stock Movement Prediction Methods Efficacy (Kumar, 2021)

With the progress in technology and computing, individuals are forecasting stock prices using machine learning and historical data, which can be seen as a sophisticated form of technical analysis. Machine learning, which involves the development of computer systems capable of learning and adapting without explicit instructions, uses algorithms and statistical models to analyse data patterns and make predictions (Crabtree, 2023). Historical stock market data often exhibit trends and patterns, and by applying machine learning algorithms, these can be identified and leveraged to predict future stock prices and their movements.

## **Problem Statement and Motivation:**

Data is quickly becoming the most valuable resource in the world, and sometimes compared with gold for its significance, it holds numerous insights that can be derived from them. With the world being data literate and knowing the importance of the data, there has always been increasing curiosity on how the historical data can be used in stock market prediction to get a precise interpretation. The stock market prediction should be accurate, efficient, and robust and must account for all the variables that might affect the stock’s value and performance.

Over time, researchers have developed numerous machine learning and deep learning algorithms to address various types of problems and handle different kinds of data. People have modelled and experimented a lot for stock price prediction from simple machine learning linear regression algorithm to complex deep learning hybrid models. Researcher are trying and updating the algorithms according to the prediction performance and computational efficiency in stock price. Simple linear regression is easy to implement and interpret and can somehow predict the stock price near to its actual value but not that accurate. Support vector machine might be effective in high-dimensional spaces and with non-linear boundaries using kernel functions but it requires careful selection of kernel and regularization parameters. Random forest algorithm reduces overfitting the data but it’s harder to interpret. Neural networks like CNN, LSTM, etc. can model highly complex and non-linear relationships including time series but requires significant computational power and extensive training data. Furthermore, stock prices are influenced by time, and to effectively analyse such time-dependent data, time series analysis has become a commonly employed technique. The Autoregressive Integrated Moving Average (ARIMA) model, introduced by (Box, et al., 2015), is one of the most well-known approaches for modelling time series data and is widely applied in real-world scenarios. However, as a linear model, ARIMA does not account for volatility or fluctuations in the underlying variance of stock prices. This limitation renders ARIMA ineffective for forecasting stock price data, particularly in the presence of fluctuating volatilities. In recent times, LSTM models have gained popularity for predicting stock prices.

Stock market indices are highly volatile, which affects investor confidence. Stock prices are considered dynamic and prone to quick changes due to the nature of the financial sector and a mix of known factors (such as the previous day's closing price and P/E ratio) and unknown variables (like election results and rumours). Machine learning methods used for stock price prediction analyse historical data to estimate the likelihood of future events or forecast trends. This process involves detecting patterns in the data, combining both current and past information, and applying predictive models that best match the data. To achieve the best performance, these models are trained using various algorithms and fine-tuned with parameters, including hyperparameters.

Many efforts have been made to predict stock prices using Machine Learning, with numerous researchers developing different models for estimation. However, predictions made by machine learning models are generally less reliable than those made by humans. Despite this, these models can act as augmented intelligence, assisting investors in making informed decisions about which stocks to invest in, potentially leading to higher returns and reduced losses (Kumar, 2021).

Recently, the rise of machine learning and deep learning has greatly improved the accuracy of models used for predicting stock prices. However, with the introduction of various machine learning and deep learning algorithms, their performance and prediction accuracy remain uncertain. Therefore, a comprehensive study is needed to evaluate and analyse the performance of different machine learning algorithms in terms of prediction accuracy and computational efficiency/time. Additionally, it is important to examine how factors such as the characteristics of the stock dataset, data splitting methods, and stock trends affect the performance of these algorithms.

## **Objectives:**

Even with significant progress, numerous obstacles remain in employing machine learning for forecasting stock prices. These challenges include selecting appropriate algorithms, feature engineering, ensuring data quality, and evaluating model performance. Furthermore, there is a need for detailed comparative studies to assess the effectiveness of machine learning models under various market conditions and investment timeframes.

This dissertation explores the feasibility of predicting stock prices using historical data combined with different machine learning and deep learning algorithms. It also evaluates and analyses these algorithms in terms of their prediction accuracy and computational efficiency.

The primary objective of this dissertation is to:

“**Evaluate and analyse the performance of machine learning algorithms in stock price prediction”.**

Some of the objectives of the dissertation are listed below:

* Evaluate the effectiveness of various machine learning techniques in predicting stock prices.
* Compare the accuracy and reliability of machine learning models with traditional statistical methods.
* Identify the most effective machine learning model for stock price forecasting.
* Assess the impact of the data used for training and evaluating the machine learning models.

## **Research Questions and Hypothesis:**

This dissertation seeks to answer:

1. How effectively and accurately can different machine learning algorithms and models predict future stock price trends?
2. Which algorithm yields the highest accuracy and requires the least computational time for stock price prediction?
3. What impact will the data utilized for training and evaluating the machine learning and deep learning models have?

Also, this dissertation seeks to test these three hypotheses:

1. **Prediction Accuracy Hypothesis:** Different machine learning algorithms exhibit varying levels of accuracy when predicting stock prices.

* **Null Hypothesis (H0):** There is no significant difference in the prediction accuracy among different machine learning algorithms.
* **Alternative Hypothesis (H1):** There is a significant difference in the prediction accuracy among different machine learning algorithms.

1. **Model Efficiency Hypothesis:** Some machine learning algorithms are more computationally efficient than others in stock price prediction.

* **Null Hypothesis (H0):** All machine learning algorithms have similar computational efficiency when used for stock price prediction.
* **Alternative Hypothesis (H1):** There is a significant difference in the computational efficiency of different machine learning algorithms.

1. **Data Effect Hypothesis:** The data used for training and testing machine learning models in stock price prediction has a significant impact on the results.

* **Null Hypothesis (H0):** There is no significant effect of data in the prediction accuracy of machine learning algorithms.
* **Alternative Hypothesis (H1):** There is significant effect of data in the prediction accuracy of machine learning algorithms.

## **Dissertation Structure:**

The remainder of this dissertation is organized as follows:

**Chapter 1. Introduction** - which gives the general overview of the research with background, problem statement, objectives, hypothesis, etc.

**Chapter 2. Literature Review** - provides a comprehensive review of recent research and practices on stock price prediction methodologies using different machine learning methodologies.

**Chapter 3. Methodology** - outlines the data collection and pre-processing procedures with detailed the research methodology, including algorithm selection, model development, and evaluation metrics.

**Chapter 4. Results, Analysis and Discussions** - presents the empirical results, findings, implications and limitations of the comparative analysis.

**Chapter 5. Conclusions and Future Works** - summarizes the main contributions and proposes areas for future research.

# **LITERATURE REVIEW**

The use of machine learning (ML) algorithms for stock price prediction has gained significant attention in both academic and financial sectors. Predicting stock prices is a complex task influenced by factors such as market sentiment, economic conditions, and global events. Over the years, numerous researchers have proposed solutions to address these challenges. While traditional methods like fundamental and technical analysis have been widely used for financial forecasting, they often fail to capture the non-linear patterns in stock market data. Machine learning models, however, offer the advantage of learning from large datasets and uncovering complex patterns. This literature review examines existing research on the effectiveness of various machine learning algorithms in stock price prediction and outlines the methodologies used to assess and compare their performance.

## **Traditional Methods and Their Limitations**

According to (Shrestha, 2021), there are lots of researches that found Fundamental Analysis as the most widely adopted approach for predicting the stock price where investor focuses on information such as intrinsic value of the corporate published in the annual report, quarterly reports, financial ratios, governmental policies, national and international events, political and economic development. The history of fundamental analysis can be traced to the 1934 book Security Analysis by Benjamin Graham and David Dodd (Graham & Dodd, 1934) which established the intellectual foundation for what would later become known as value investing. Numerous studies have been conducted to formalize and expand upon the stock selection principles outlined in the book.

There is also another approach, Technical Analysis which is almost different than the previous one. This involves the use of charts, technical indicators and historical price patterns to predict future price movements. According to (Kabir, et al., 2023), while traditional methods of forecasting stock price like fundamental and technical analysis can provide valuable insights, they may overlook short-term maker dynamics and external events, limiting their predictive capabilities. Also, according to (Fama, 1970) in his paper regarding theoretical and empirical literature on the *efficient market hypothesis*, found that all new information will be reflected in asset prices immediately without delay. So, changes in future prices have nothing to do with past and present information. From their perspective, predicting future asset price is considered impossible. On the other hand, many studies try to prove *effective market hypothesis* experimentally and empirical evidence shows that the stock market can be predictable in some ways.

## **Machine Learning Approach in Stock Price Prediction:**

In recent years, the application of machine learning for stock price prediction has gained significant attention. Extensive research has been conducted in this area, with many studies demonstrating that machine learning methods can effectively predict stock prices using historical data.

### **Basic Machine Learning Approaches for Prediction**

(Pathak & Pathak, 2020) conducted research on stock price prediction using basic machine learning algorithms and published a paper titled "Study of Machine Learning Algorithms for Stock Market Prediction." They utilized algorithms such as Random Forest, Support Vector Machine, KNN, and Logistic Regression, evaluating their performance using metrics like accuracy, recall, precision, and F-score. Their experiment revealed that Random Forest achieved the highest accuracy at 80.7%, followed by Logistic Regression with an accuracy of 78.6%.

Similarly, (Kabir, et al., 2023) presented a research paper on stock price prediction using machine learning, where basic algorithms such as Linear Regression, Support Vector Machine, Decision Tree, and Random Forest were utilized. They studied the behavior of these algorithms and analysed the impact of different features on prediction accuracy. The results highlighted the potential of machine learning in the financial sector and emphasized the significance of feature selection and engineering.

The author from (Shrestha, 2021) discussed stock price prediction for NEPSE (Nepal Stock Exchange) using machine learning algorithms such as Multi-Layer Perceptron, Support Vector Regressor, and Multiple Linear Regression. The research also identified several unpredicted factors, including foreign exchange rates, the base interest rate set by **Nepal Rastra Bank**, and the prices of commodities like gold, silver, and petroleum products, which influence stock market trends. The author concluded that while these unpredicted factors are correlated with stock prices, their significance is minimal, and found that Multiple Linear Regression outperformed all other models in every area.

(Khanna, et al., 2022) studied four prediction models—LSTM, XG Boost, SVM, and Random Forest—in their paper, evaluating the accuracy and F1 scores for each model. The research found that using continuous data with 10 technical indicators produced the best performance with the Random Forest classifier, achieving the highest accuracy of 84.89% and the highest F1 score of 89.33%. The experiment also provided insights into why the Naïve Bayes classification model is not suitable for stock price prediction.

### **Deep Learning Approaches along with other ML Algorithms**

The research work by (Mehtab & Sen, 2019) proposed a hybrid approach for predicting stock price movement using machine learning, deep learning, and natural language processing. To predict price movement patterns, various classification techniques were employed, while regression models were used to predict the actual closing price. The study focused on the NIFTY 50 index values from the National Stock Exchange (NSE) of India, collecting daily price data over three years. LSTM-based deep learning models were developed to predict stock closing prices and compared with the prediction accuracies of other machine learning algorithms. Additionally, the model was enhanced by integrating a sentiment analysis module using Twitter data to link public sentiment to market trends. The proposed scheme was tested using cross-validation with Self-Organizing Fuzzy Neural Networks (SOFNN). In a follow-up study (Sen, et al., 2021), the same stock was analysed using four deep learning-based regression models built on Long Short-Term Memory (LSTM) networks, employing a novel walk-forward validation approach. The results clearly showed that the LSTM-based univariate model, which used one-week prior data to predict the next week's open value of the NIFTY index, was the most accurate model.

In a study conducted by (Rizvi & Khalid, 2024), the authors explored stock market analysis using computer models to predict stock price changes, focusing specifically on the AAPL dataset from Apple Inc. They compared various models—Multi-Layered Perceptron (MLP), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU)—to provide a comprehensive performance analysis of the deep learning approach. The study concluded that LSTM demonstrated the best predictive capability by capturing complex patterns due to its ability to handle long-term dependencies. In contrast, the MLP model struggled as it could not capture the data's intricate patterns. The findings showed that the LSTM model achieved the highest accuracy when the number of epochs was set to 50, with lower MAE, MSE, and RMSE values at this epoch count.

The research done by (Saud & Shankya, 2019) examined two major commercial banks listed on the Nepal Stock Exchange (NEPSE) and compared the stock price prediction performance of the GRU model with widely used gradient descent optimization techniques: Momentum, RMSProp, and ADAM. The study concluded that GRU with ADAM provided a more accurate and consistent approach for predicting stock prices. The authors used MAPE as the performance indicator, with the best MAPE achieved by GRU with ADAM at 1.15 for EBL and 2.13 for HBL stock data. GRU with RMSProp predicted the stock prices for EBL and HBL with MAPE values of 2.27 and 3.08, respectively, while GRU with Momentum resulted in MAPE values of 2.90 for EBL and 4.59 for HBL.

The paper by (K, et al., 2021) compared the effectiveness of various Artificial Neural Networks, such as LSTM, RNN, SVM, and other deep learning and machine learning algorithms, for stock price prediction, particularly in the context of long-term investing. The study presented a comparative analysis of stock price prediction using both fundamental and technical analysis alongside machine learning techniques. The paper concluded that for an effective market strategy in predicting stock prices, both fundamental and technical analysis should be integrated, as this approach helps investors make informed long-term investment decisions.

In the research study by (Huang, et al., 2021), the authors analysed 22 years of quarterly stock financial data and investigated three machine learning algorithms: Feed-forward Neural Network (FNN), Random Forest (RF), and Adaptive Neural Fuzzy Inference System (ANFIS) for stock prediction based on fundamental analysis. Additionally, RF-based feature selection and bootstrap aggregation were applied to enhance model performance and combine the predictions from different models. The study concluded that the RF model provided the best prediction results, and feature selection improved the performance of both FNN and ANFIS. The paper found that the aggregated model outperformed all baseline models, demonstrating that machine learning models can assist fundamental analysts in making informed stock investment decisions.

### **Hybrid Models and Ensemble Approaches**

(Chen, 2020) in his research paper “Using Machine Learning Algorithms on Prediction of Stock Price”, three machine learning algorithms—Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), and Support Vector Regressor (SVR)—were compared. The paper also introduced a hybrid model combining CNN and LSTM (CNN-LSTM) and applied it to four stock datasets: Apple, Mastercard, Ford, and ExxonMobil. The study concluded that a machine learning model trained on one stock dataset could be used to forecast other stock prices, with only a slight reduction in accuracy. The paper summarized the model accuracies using mean absolute percentage error (MAPE), showing that combining CNN and LSTM improved accuracy compared to LSTM alone. Additionally, the SVR model was found to predict stock prices with the highest accuracy, likely due to the more comprehensive set of features used in the SVR model. The paper also included the kurtosis and skewness metrics of the financial data. Kurtosis measures the "tailedness" of the return distribution, indicating the presence of outliers and extreme returns (Kenton, 2024), while skewness measures the asymmetry of the return distribution around its mean, reflecting whether data points are concentrated on one side of the mean (Taylor, 2024).

The authors from (Sonkavde, et al., 2023) discussed forecasting stock market prices using various machine learning and deep learning models, presenting a systematic review, performance analysis, and implications. They provided an overview of the machine learning and deep learning models applied in the financial sector, along with a generic framework for stock price prediction and classification. Additionally, they implemented an ensemble hybrid model (stacked ensemble model) combining Random Forest, XG-Boost, and LSTM to forecast the stock prices of TAINIWALCM and AGROPHOS, and performed a comparative analysis with other popular models. The authors evaluated the models using Root Mean Square Error (RMSE) and R-squared (R²) scores and found that the hybrid model (stacked ensemble model) achieved the lowest RMSE and the highest R² score, making it the best model for stock price prediction. For the stacked ensemble model, they used Mean Squared Error (MSE) as the loss function, the ADAM optimizer, and a maximum of 50 epochs.

The paper presented by (A, et al., 2023), also illustrates that a CNN-LSTM Neural Network can effectively track the patterns and characteristics of stock data. It proposed that by extracting features from stock data and transforming them into tensors, it is possible to obtain these features and subsequently feed them into LSTM neural networks to identify patterns, thus enabling predictions of the stock market over a specified timeframe. The paper outlined the attributes of a custom CNN-LSTM model and tested its performance, revealing a high accuracy even when trained on real-time stock market data. To evaluate the performance of various machine learning algorithms, the Mean Square Error (MSE) score was utilized, with the CNN-LSTM model achieving an MSE score of 0.035, while the LSTM and XG-Boost models recorded MSE scores of 0.045 and 0.047, respectively.

### **Performance Metrics for Evaluation**

Multiple studies have emphasized the need for proper evaluation metrics to assess machine learning models in stock price prediction. (Guresen, et al., 2011) examine various metrics such as Mean Absolute Error (MAE), mean squared error (MSE), root mean squared error (RMSE), and R-squared (R²) to evaluate predictive accuracy. As mentioned in the previous section, (Pathak & Pathak, 2020) employed recall, precision, and F1 score to assess predictive accuracy. (A, et al., 2023) utilized mean square error (MSE) as a performance metric, while (Khanna, et al., 2022) focused on predictive accuracy and the F1 score. Other researchers also adopted MAE, RMSE, among others, as performance metrics to gauge predictive accuracy.

## **Research Gap:**

Following the advent of machine learning for forecasting stock prices, a majority of researchers concentrated on enhancing specific machine learning algorithms. They thoroughly explored various methods for predicting stock prices and proposed numerous advanced predictive models. The research papers that were studied for the literature review mostly focused on building models and calculating the prediction accuracy. Most of them compared few of the algorithms in same category based on the prediction accuracy which are not enough to understand the behaviour of the algorithm in predicting the stock price in terms of accuracy and computational time. Various researchers utilized different performance evaluation metrics, which were insufficient for comparing and analysing the effectiveness and prediction accuracy of various machine learning algorithms in forecasting stock prices.

To address the research gap mentioned, we plan to develop various machine learning models across several categories: basic machine learning algorithms, ensemble learning algorithms, time series forecasting algorithms, and deep learning algorithms, including hybrid models. For basic machine learning, we will use Linear Regression and Support Vector Machine. In the ensemble learning category, we will implement Random Forest Regressor and the XGBoost algorithm. ARIMA will be chosen for time series forecasting, while the deep learning models will include CNN, Long Short-Term Memory (LSTM), and a hybrid CNN-LSTM model. We intend to evaluate and compare these models based on prediction accuracy and computational time, using consistent performance evaluation metrics to assess their effectiveness.

# **Methodology**

## **Project Overview**

Forecasting stock prices is a challenging task due to the inherent volatility and complexity of financial markets. With the continued advancement of machine learning (ML), there is increasing interest in utilizing these algorithms to predict stock price trends by analyzing historical data and identifying patterns those traditional methods may miss. This project aims to systematically evaluate the performance of various machine learning algorithms in stock price prediction, comparing their accuracy, reliability, and practical application for both short-term and long-term forecasts.

The research will explore a range of algorithms—from basic models like linear regression and support vector machines (SVM) to more advanced techniques such as convolutional neural networks (CNN) and long short-term memory (LSTM) networks—using stock market data. By examining historical stock prices, the project will assess how effectively each model can predict future stock movements.

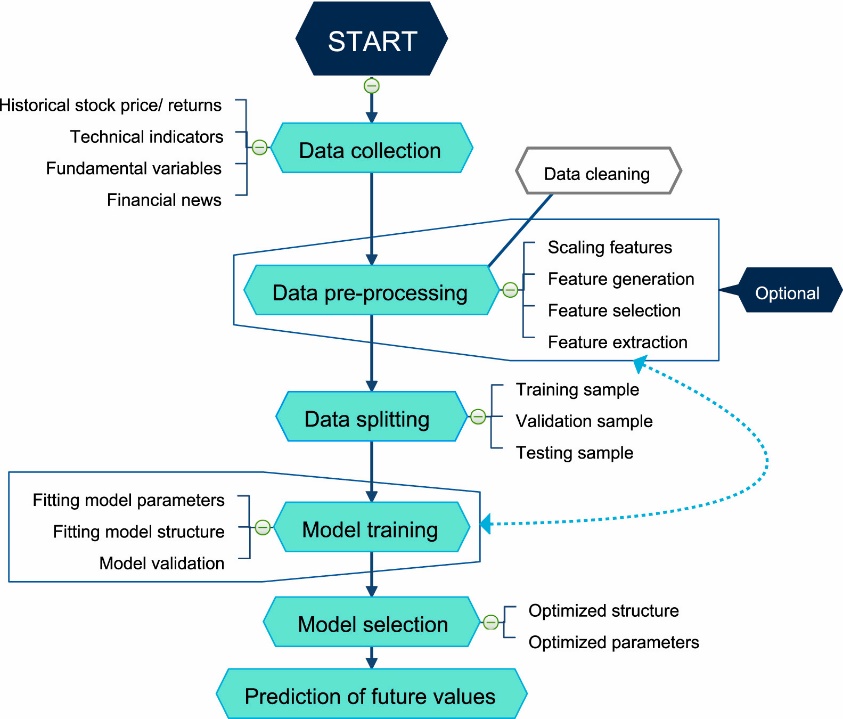


Figure 3: Workflow of Stock Price Prediction Model using Machine Learning (Kumbure, et al., 2022)

A thorough comparison is carried out using performance metrics such as Mean Absolute Error, Mean Squared Error, Root Mean Squared Error, and training time data to assess the predictive accuracy of these machine learning algorithms. Additionally, factors like computational efficiency, training duration, and model interpretability will be crucial in determining which machine learning algorithms are most effective for forecasting stock prices under various conditions. This performance evaluation provides valuable insights into the strengths and limitations of different ML models, contributing to the future improvement and application of predictive algorithms in financial markets. In the end, the results will provide important recommendations for traders, investors, and data scientists engaged in this domain.

## **System Design and Methodology Overview**

The system design for this study **“Performance Evaluation and Analysis of Machine Learning Algorithms in Stock Price Prediction”** emphasizes the development of a thorough framework for gathering data, training models, assessing their performance, and contrasting various algorithms. The objective is to not just forecast stock prices but also to evaluate the effectiveness of different machine learning algorithms based on prediction accuracy, performance metrics, and computational efficiency, offering a comprehensive insight into their advantages and disadvantages in this particular field.

Designing a machine learning-based system for stock price prediction involves several key stages, including data acquisition, processing, model training, evaluation, and deployment. The following diagram illustrates the primary components in the typical architecture of such a system.

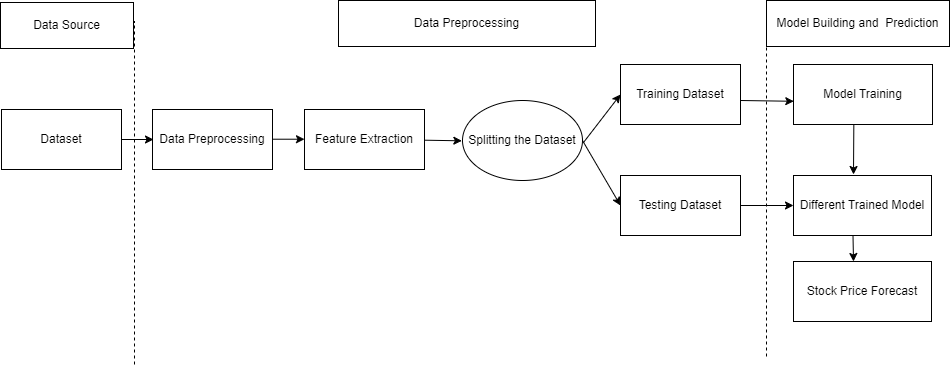


Figure 4: System Design of Stock Price Prediction

Our research project is developed and implemented following this system design. Overall system design of our project consists of data collection, data processing, model building, testing and predicting the future stock price.

### **Project Flowchart**

Below figure shows the project methodology and flow of our research project. Our research started with the problem understanding with review of existing architecture of the system shown in literature review section. We collected the data required for the project from different sources. We performed the required data pre-processing and feature extraction. Then the model is trained with the training data for the stock price prediction.

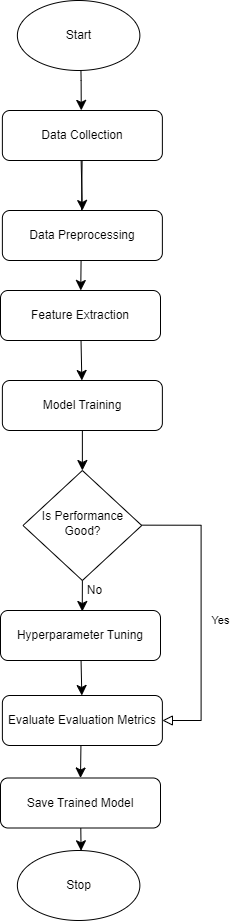


Figure 5: Model Training Flowchart of Stock Price Prediction

## **Data Understanding**

According to (Ronaghan, 2019), quality data is fundamental to any machine learning and data science engagement to get actionable insights where appropriate data must be sourced and cleansed. Understanding data is a vital phase in the machine learning workflow that encompasses gathering, examining, and interpreting data to derive insights that guide the creation of predictive models. This stage is important since the quality, relevance, and attributes of the data greatly affect the effectiveness of machine learning algorithms. For our research project, we adhered to the subsequent steps for data comprehension:

### **Data Collection**

For this research project, we used two different datasets: google.csv (Google Stock Price Dataset) and nabil.csv (Nabil Bank Stock Price Dataset). These datasets are from different stock market of the world.

1. **Google Stock Price Dataset**

We took Google Stock Price Dataset (google.csv) from Yahoo Finance website which can be found on the link ([Yahoo Finance](file:///C:\Users\gauri\AppData\Roaming\Microsoft\Word\(https:\finance.yahoo.com\quote\GOOG\history\%3fperiod1=1093996800&period2=17162679)). Google's parent company, Alphabet Inc., is registered on the NASDAQ stock exchange in the United States. Alphabet has two types of shares that are publicly traded: GOOGL (common shares with voting rights) and GOOG (shares with no voting rights). Both GOOGL and GOOG are listed on NASDAQ under their respective ticker symbols. NASDAQ is a major stock exchange known for listing technology companies, including other giants like Apple, Microsoft, and Amazon. For our research, we used GOOGL share price which is a class A share with voting rights. The share prices are in US dollar ($). It contains the Google Stock Price Data from 19th August 2004 to 25th September 2024. This dataset has 7 columns: **Date, Open, High, Low, Close, Adj. Close, Volume.** Due to the availability of high-quality data with large historical dataset, globally recognized company with strong financials which is followed widely by investors, analysts and institutions, we chose this dataset for our research project. The dataset can be downloaded from Yahoo Finance website or can be downloaded from its python library called ‘**yfinance**’.

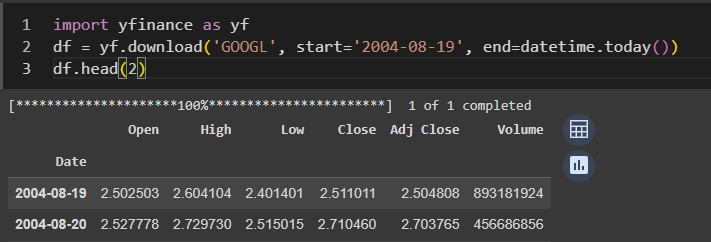


Figure 6: Code to Download Google Stock Price Data from Yahoo Finance

1. **Nabil Bank Stock Price Dataset**

Nabil Bank Stock Price Dataset (nabil.csv) is taken from Share Sansar website which can be found on the link ([Share Sansar](https://www.sharesansar.com/company/nabil)). Nabil Bank is one of the largest commercial banks in Nepal, and its stock is widely followed by investors due to its stable performance and regular dividends. Nabil Bank Stock is registered under Nepal Stock Exchange (NEPSE) which is an only stock exchange in Nepal. The dataset contains the data from 20th March 2011 to 25th September 2024. There are 8 columns in the dataset: **Date, Open, High, Low, Ltp, Change, Qty, Turnover**. The share prices are in Nepalese currency (NPR).

The Nepalese stock dataset along with Nabil Bank Stock Price dataset is not easily available in the internet and there was no dataset in the proper format that we needed. As there was no any downloadable dataset file in the website, the dataset was scrapped from Share Sansar website. We got the approval from Share Sansar to scrape the data from the website as part of ethical considerations.

For data scrapping, we used **Selenium** because it can be utilized to extract dynamic information from websites that need user engagement, such as completing forms, pressing buttons, or moving between pages. Figure below shows the complete setup for web data scrapping using selenium. The required libraries along with selenium web driver was imported and the driver instance was created which will handle the browser and the requests. We used VS code as it was easier for us to write, debug and navigate the code.



Figure 7: Importing and Setting Up Selenium Driver

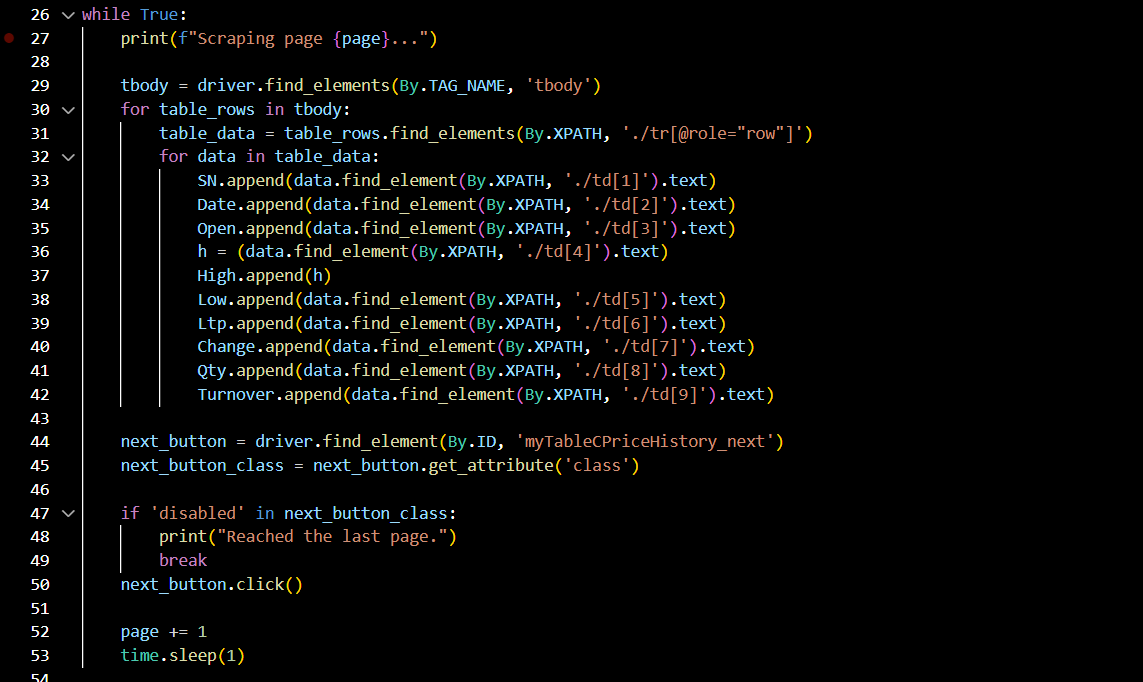


Figure 8: Code for Scrapping Nabil Bank Stock Price Data

The scrapped data was stored in CSV format. The complete code for data scrapping can be found in the GitHub repository with the link ([webscrapping-using-beautifulsoap-selenium](https://github.com/GaurabStha/webscrapping-using-beautifulsoap-selenium/tree/main/Nabil_bank_stock_price_scrapping)). The dataset comes with specific usage rights that impose restrictions on commercial use and public redistribution, and it is intended for development and research purposes.

### **Data Wrangling (Data Preprocessing and Feature Engineering)**

As part of data preprocessing and feature engineering, we explored the foundational aspects of both the dataset that included the process of loading, understanding its overall dimensions and gaining insights into variables at our disposal. We used Google Colaboratory for data preprocessing, feature engineering, EDA and model building.

#### **Overview of the Datasets**

This section contains the summary of the dataset. Talking about the data dimension of both datasets, there are 7 variables (columns) and 5060 observations (data or rows) in Google dataset where in Nabil dataset, there are 8 variables and 3083 observations.

The 7 variables of Google dataset are:

1. **Date:** This column indicates the calendar date on which the stock data is recorded.
2. **Open:** This column indicates the initial price of the stock for a given trading session.
3. **High:** The high price refers to the maximum value at which the stock was traded during a particular session, reflecting the highest price achieved that day.
4. **Low:** The low price denotes the minimum value the stock reached during the session, marking the lowest price for that day.
5. **Close:** The closing price shows the final price at which the stock was traded at the end of the session, representing its last value before the market closed.
6. **Adj Close**: The adjusted closing price accounts for corporate actions such as dividends, stock splits, and new stock offerings, which may affect the stock’s price but are not directly related to its performance. The adjusted close is often used to assess the stock’s performance over time.
7. **Volume:** Volume reflects the total number of shares traded during a specific period. It indicates the level of market activity and liquidity for the stock. Higher volume often suggests greater investor interest, while lower volume may indicate less active trading.

For the Nabil dataset, the first 4 columns are same as Google dataset. The rest of the columns of Nabil dataset are:

1. **Ltp:** It is the last transaction price at the end of the trading session which is same as the Close column of Google dataset.
2. **% Change:** It’s the percentage difference between opening price and closing price.
3. **Qty:** It is same as the volume column of Google dataset which is the total number of shares traded in a specific time period.
4. **Turnover:** It is the total value of shares traded rather than the number of shares.

#### **Descriptive Statistics**

We provided the comprehensive summary of each variable in the dataset, encompassing key statistical measures such as length, class, mode, minimum value, median, maximum value, mean and quartile values. This thorough examination allowed us to grasp the range of our variables and identify potential outliers and missing values. Through this analysis, we were able to affirm the absence of any missing values in our dataset.

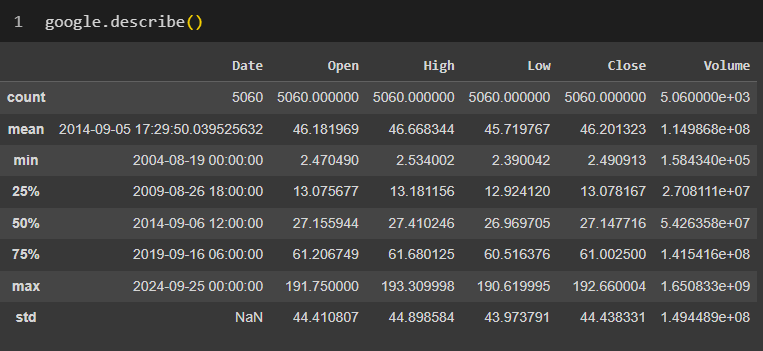


Figure 9: Descriptive Statistics of Google Dataset

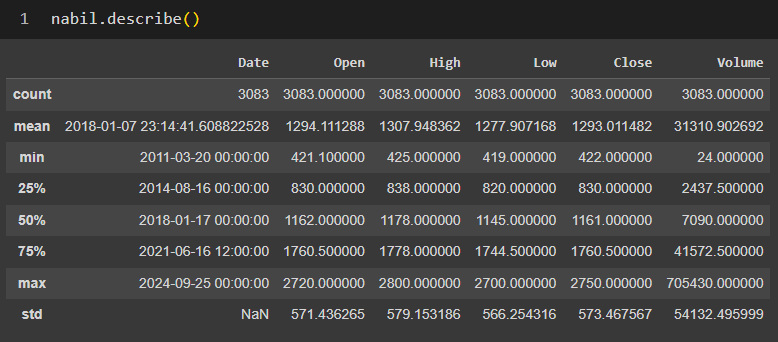


Figure 10: Descriptive Statistics of Nabil Bank

#### **Data Cleaning**

We undertook the transformation of our data into structured formats or types that facilitate convenient data processing. This step is crucial for effective data cleaning and ensures optimal preparation for subsequent analyses.

1. **Irrelevant Features**

As we had to perform the exploratory data analysis on both the dataset, we need to have same columns so we dropped the ‘Adj Close’ column from Google dataset as it was almost same as ‘Close’ column. Also, we dropped the ‘% Change’ and ‘Turnover’ columns from the Nabil dataset. We also changed the column name of Nabil dataset from ‘Ltp’ to ‘Close’ and ‘Qty’ to ‘Volume’. The figure below shows the code to change the column names and drop the unwanted columns.

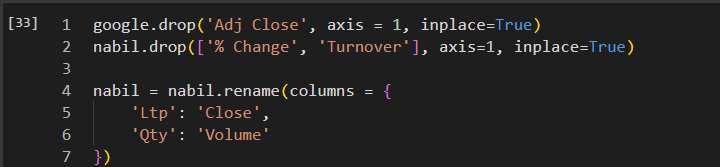


Figure 11: Dropping and Renaming the Column Names

1. **Missing values**

Missing value refer to the absence of data in a specific variable or observation within a dataset. The overall overview of both the dataset is presented in table:

Table 1: Overview of the Dataset

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Columns | Description | Count | | Missing Values | |
| Stock Price Dataset | | **Google** | **Nabil** | **Google** | **Nabil** |
| Date | Date on which the stock is documented | 5060 | 3083 | 0 | 0 |
| Open | Initial listed price of the stock for a trading day | 5060 | 3083 | 0 | 0 |
| High | Peak price of the stock exchanged during a particular trading day | 5060 | 3083 | 0 | 0 |
| Low | Lowest price of the stock exchanged during a specific trading day | 5060 | 3083 | 0 | 0 |
| Close | Final trading price of the stock at the conclusion of a trading session | 5060 | 3083 | 0 | 0 |
| Volume | Total volume of shares exchanged within a certain timeframe | 5060 | 3083 | 0 | 0 |

From above Table 1, we were sure that there are no any missing values in both the dataset.

1. **Outliers**

In this outlier subsection, we addressed unusual data points that deviate significantly from the majority of our dataset. Identifying and understanding these outliers is crucial as they can impact statistical analyses and interpretations. While looking at the data with boxplots, we found some unusual points (outliers) that stand out in our time series analysis. Though we see these outliers, our current knowledge does not cover specific methods to deal with them. Therefore, for this report, we have decided to keep the outliers as they are.

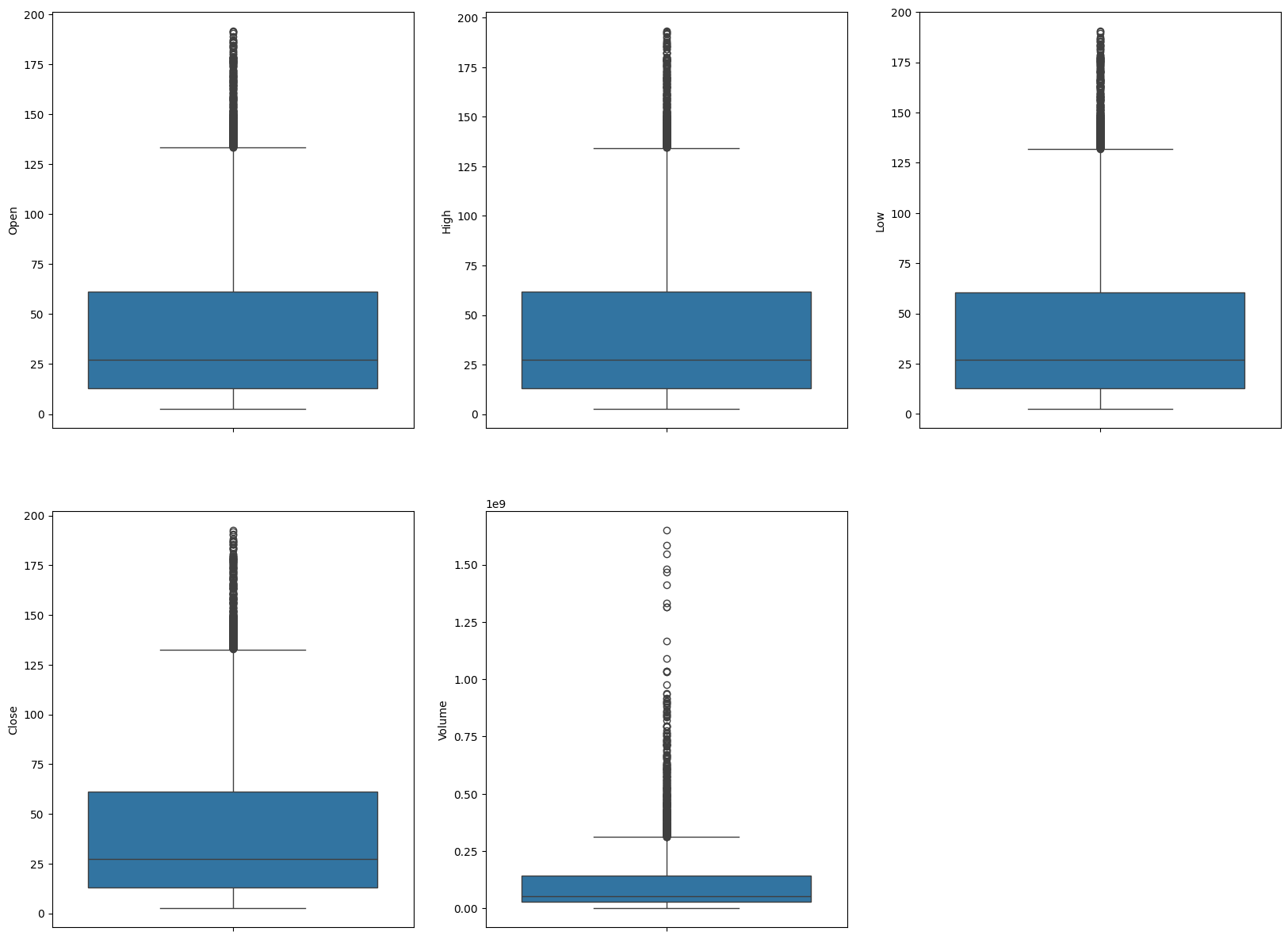


Figure 12: Boxplot of Google Stock Price Dataset for Outliers Detection

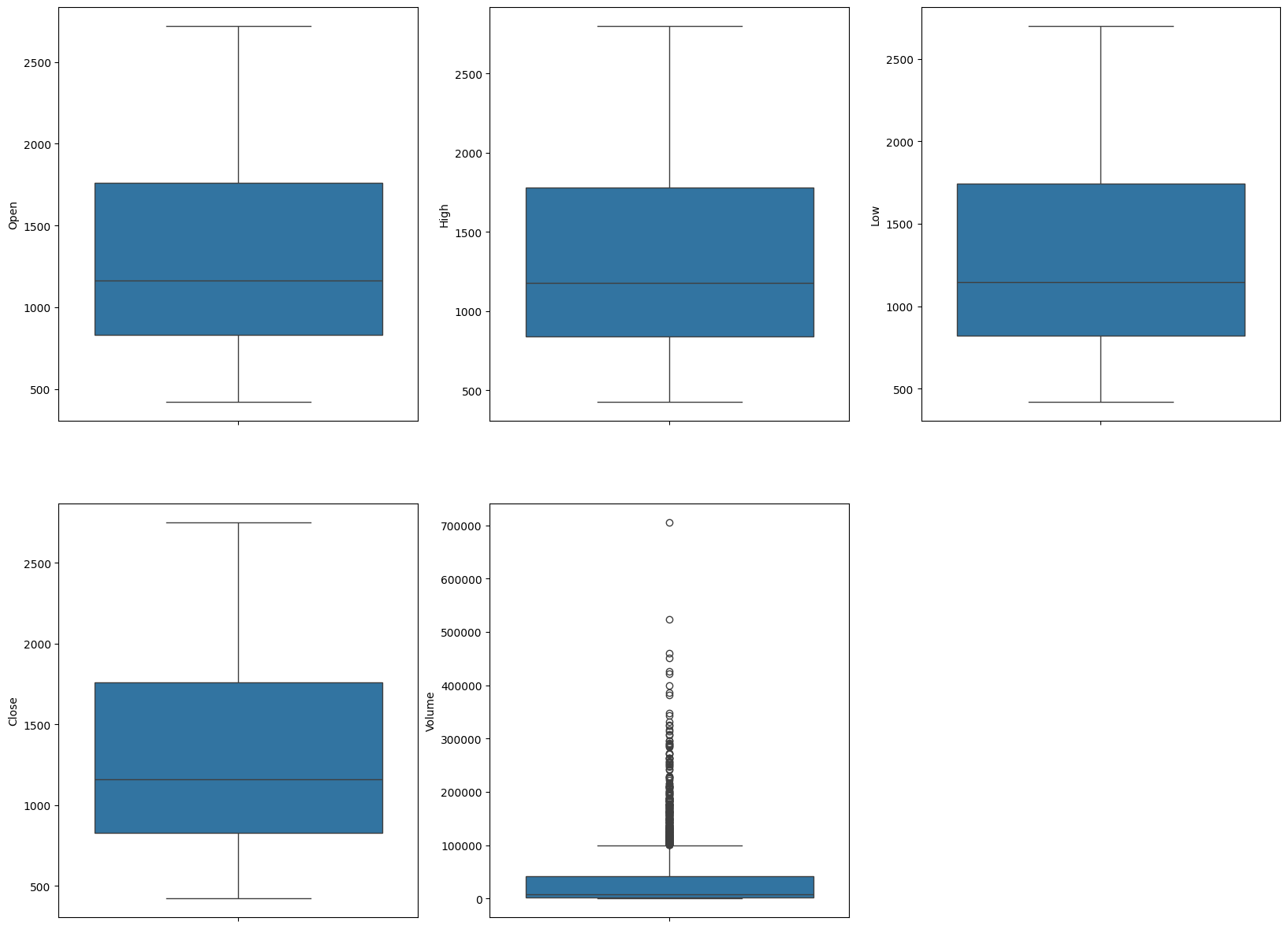


Figure 13: Boxplot of Nabil Stock Price Dataset for Outliers Detection

#### **Data Transformation**

Here we transformed our data into shapes or types that make data processing convenient. In Nabil dataset, all the data types (dtypes) of the columns were object so we changed the data types (dtypes) of each column.

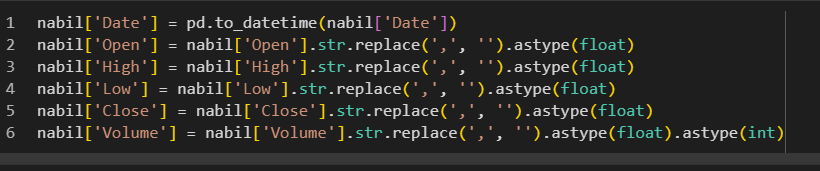


Figure 14: Changing dtypes of Columns of Nabil Dataset

### **Exploratory Data Analysis (EDA)**

In this section, we employed visual representations and responses to address and uncover the significant patterns and insights within our dataset, fostering a comprehensive understanding of its dynamics.

1. **Stock Price over Time**

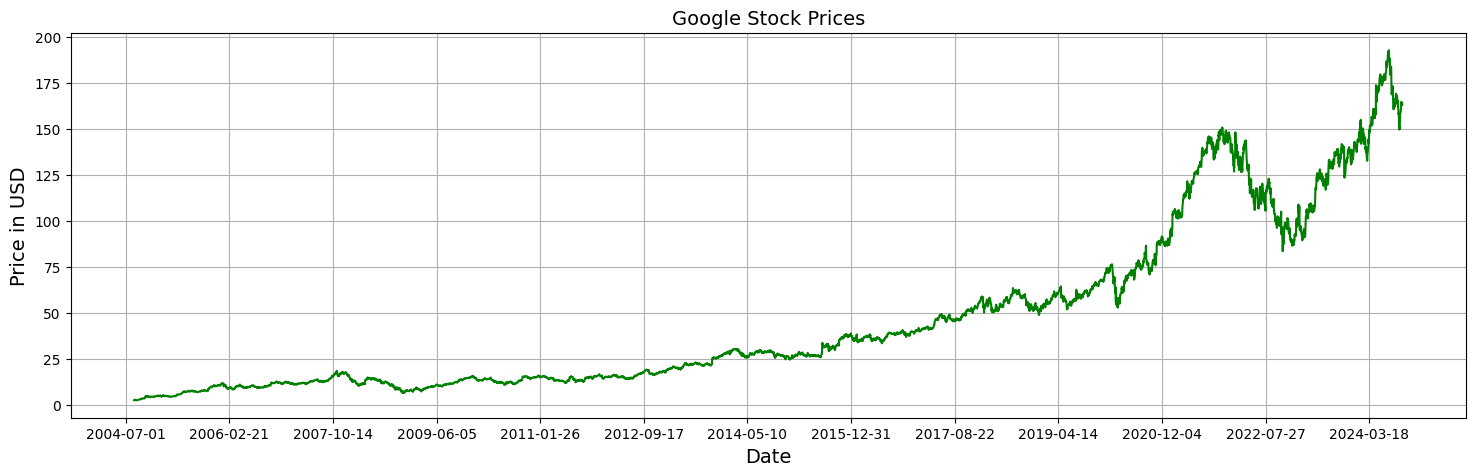
****

Figure 15: Google Stock Price Trend over Time

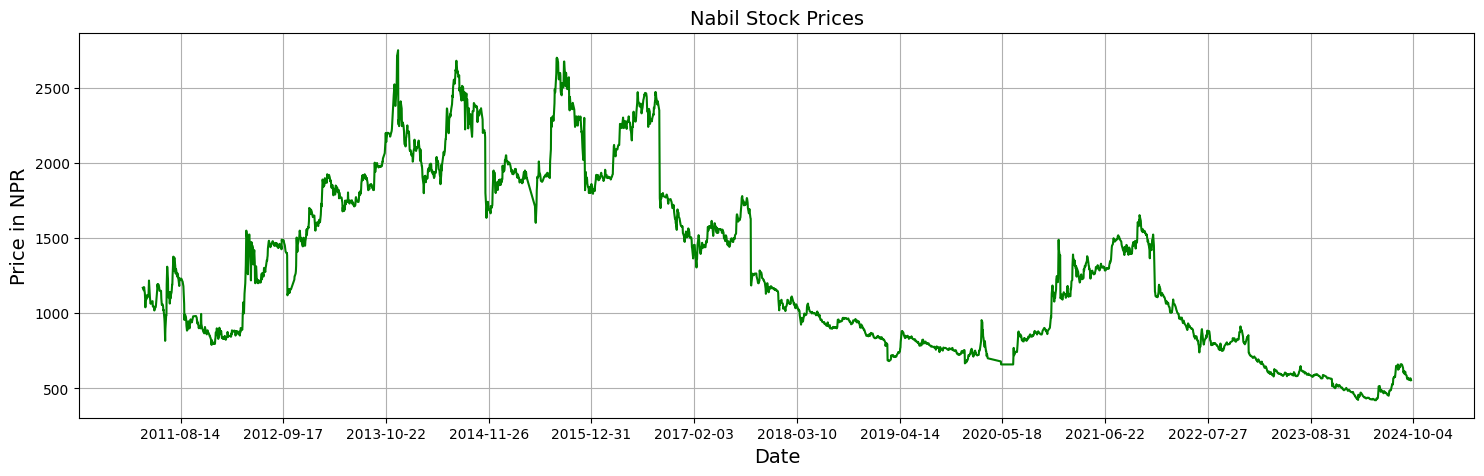


Figure 16: Nabil Stock Price Trend over Time

From the above Google Stock Price line graph (Figure 16), a consistent upward trajectory over the years with a notable peak in mid-2024 is observed, marking the highest closing price. However, a substantial decline is observed from the commencement of 2022 until 2023. Interestingly, the stock demonstrates a renewed upward trend from the early months of 2023, showcasing dynamic market movements and suggesting a resilient growth pattern amid fluctuations.

From the Figure 17 of Nabil Stock Price trend graph, it is observed that the stock price was initially high which touched the highest closing price in late 2013. The price was gradually declining which observed its lowest closing price at the mid of 2024.

1. **Probability Distribution of Close Price of Stock**

We utilized a probability distribution to illustrate the closing price data in our dataset. The graph above demonstrates the rising mean and standard deviation, suggesting that our series is not stationary.

The following graph presents the probability distribution of the closing price for both stock datasets.

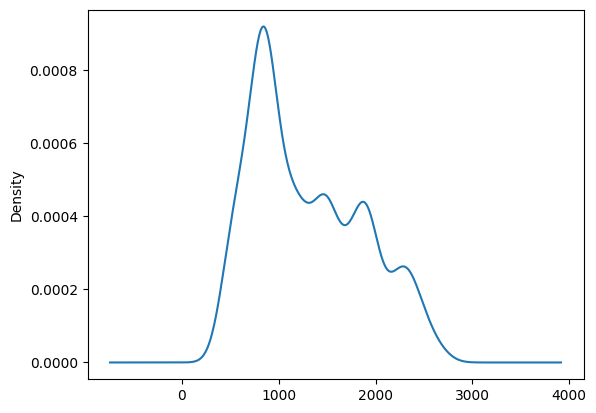
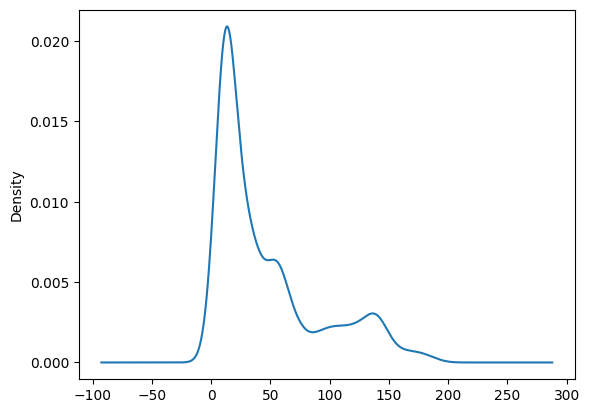


Figure 17: Probability Distribution of Close Price of Google and Nabil respectively

1. **Historical Annual Price**

The figures below provide a comprehensive overview of Google and Nabil historical annual stock price data, featuring key metrics for each year with average stock price, stock open price, stock high price, stock low price and close price with annual change.

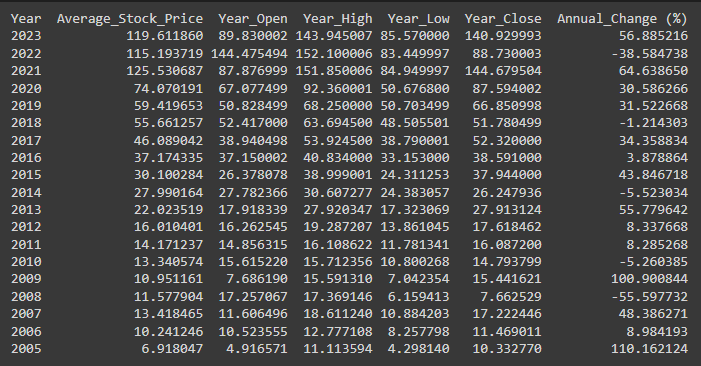
****

Figure 18: Google Historical Annual Stock Price Data



Figure 19: Nabil Historical Annual Stock Price Data

1. **Stock Price High vs Low Comparison**

The figures below show the high and low stock price of each day of Google stock and Nabil stock. From the graph, we can understand the range of the stock price that had been in a day.

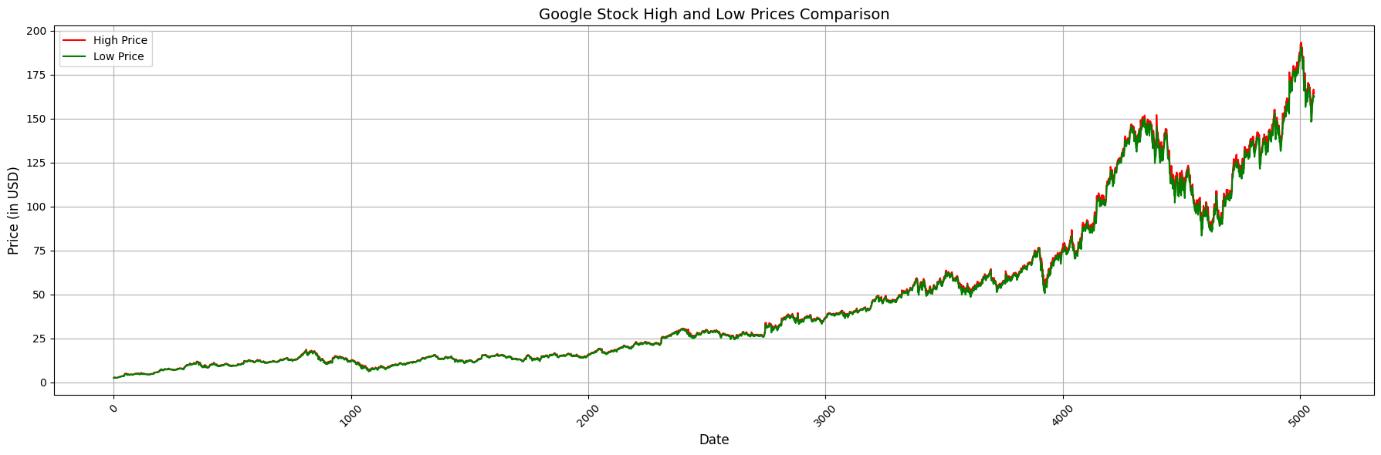
****

Figure 20: Google Stock High and Low-Price Comparison



Figure 21: Nabil Stock High and Low-Price Comparison

1. **Top 20 Highest Share Volumes Traded in a Day**

On 2006 January 20, Google had its highest share volumes traded in a day that was around 1.6 billion of shares whereas on 2021 November 3, Nabil had its highest share volume traded which was 705K of shares. The below graphs show the top 20 day when both stocks had their highest share volume traded.

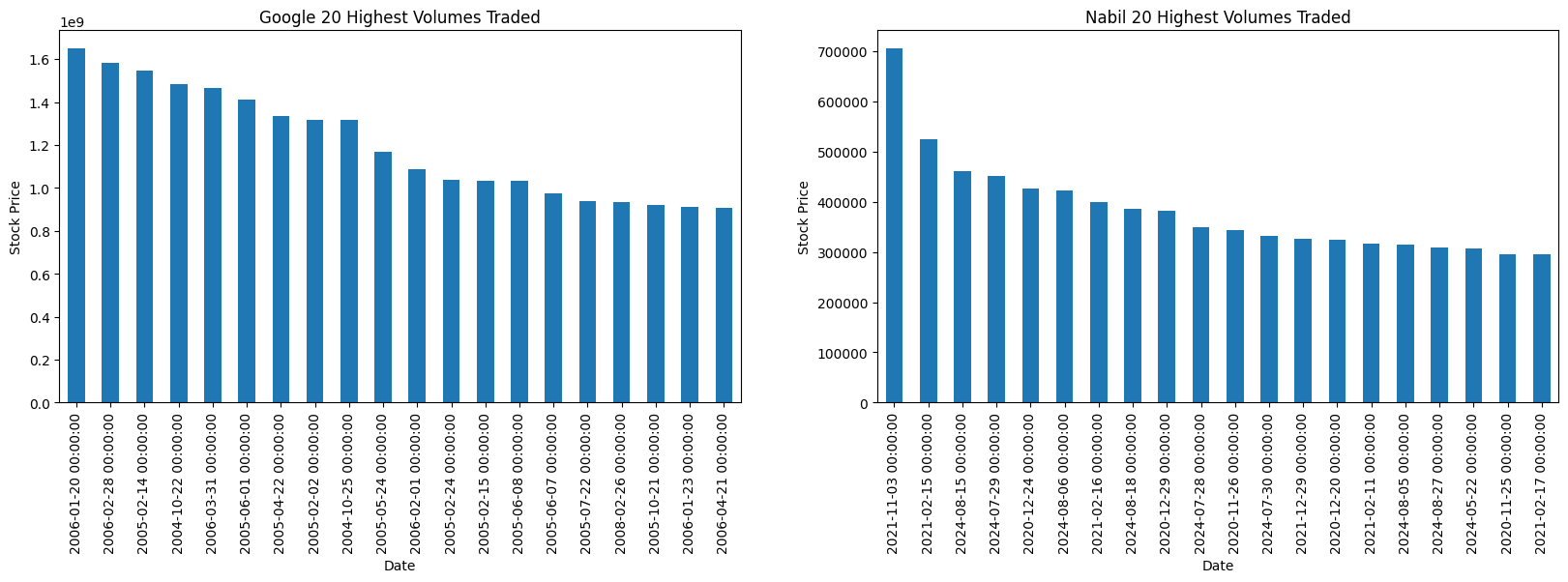
****

Figure 22: Top 20 Highest Share Volumes Traded in a Day

1. **Daily Volume Average**

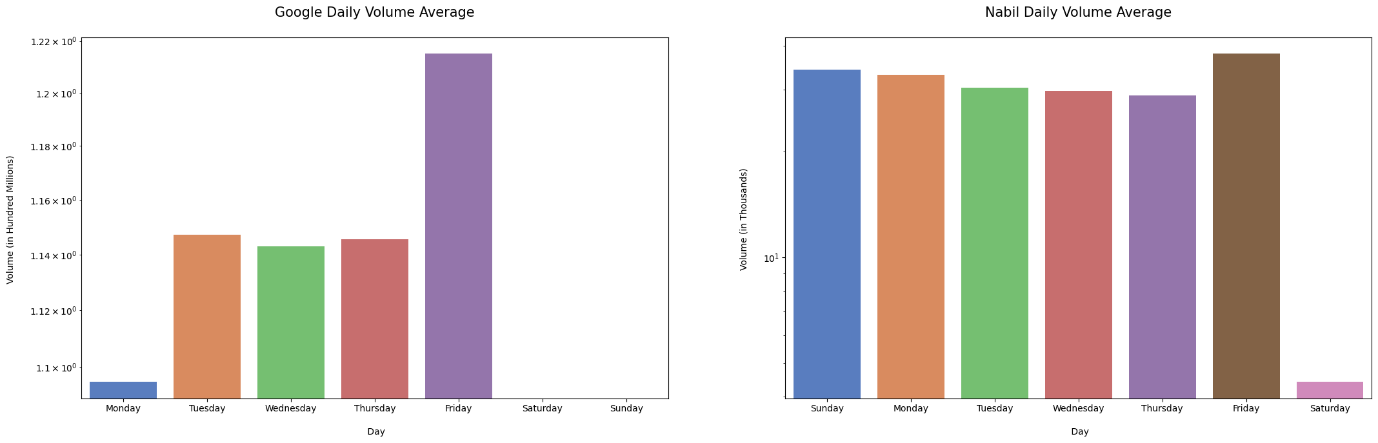
****

Figure 23: Daily Volume Average of the Stock

The analysis of the plot reveals that Friday stands out with the highest volume average, indicating increased market activity, possibly influenced by end-of-week trading strategies. In contrast, Monday records the lowest volume average, suggesting a subdued start to the trading week whereas the remaining days exhibit relatively similar average volume levels for Google. For Nabil, all most all weekdays it exhibits similar average volume levels. But Friday records the highest volume average according to other days.

1. **Simple Moving Average Days**

In finance, a moving average (MA) is a frequently used stock indicator in technical analysis. Its primary purpose is to smooth out price fluctuations by generating an average price that is continuously updated (Fernando, 2024).

The analysis of the plots reveals that the moving average changes over 20, 50, and 100 days exhibit a strong similarity to the close price plot across all observed days. This suggests that the moving averages effectively capture the underlying trend and volatility of Google’s and Nabil’s stock price over various time horizons.

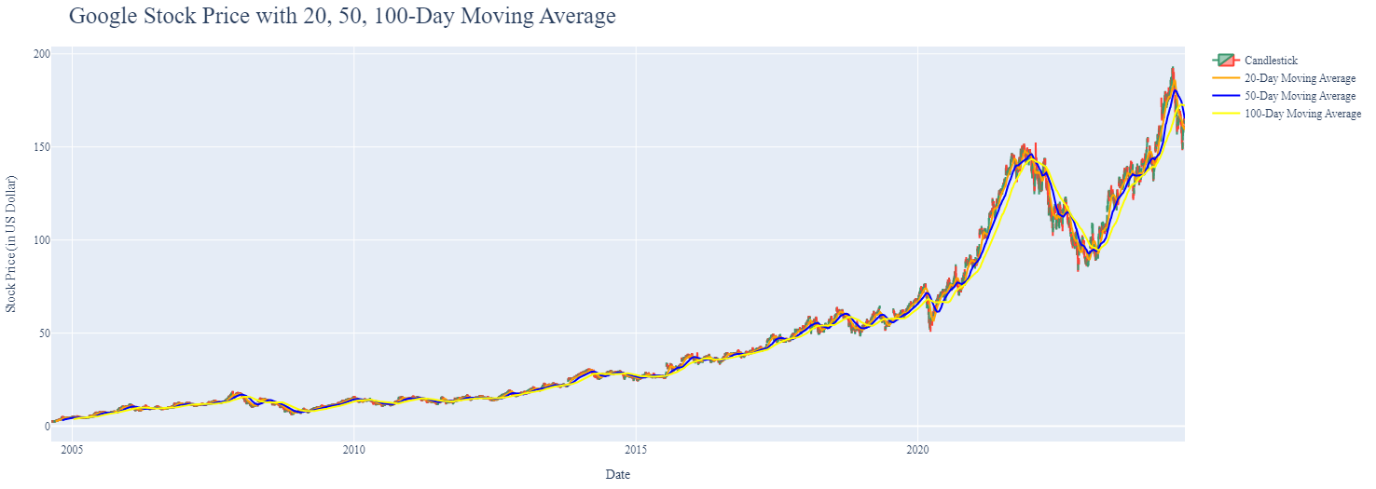
****

Figure 24: Google Stock Price with 20, 50, 100-Day Moving Average

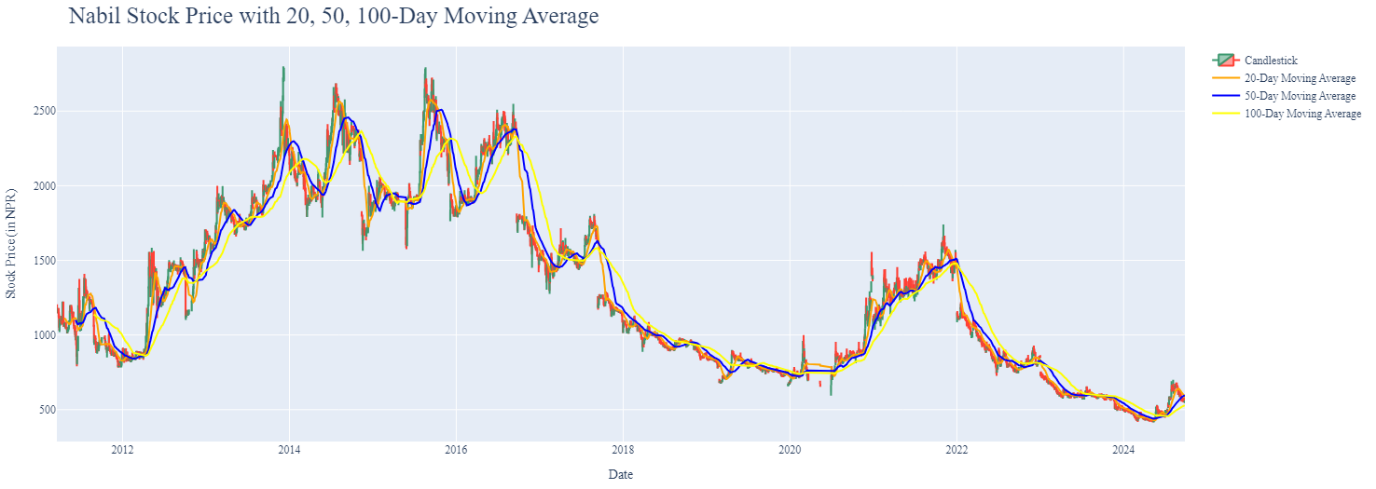


Figure 25: Nabil Stock Price with 20, 50, 100-Day Moving Average

1. **Candlestick Representation**

The images (Figure 25 & Figure 26) depict a candlestick chart representing Google's and Nabil’s stock market performance over a period of time (from early 2023 to late 2023). The x-axis shows the date range, while the y-axis represents the stock price in USD and NPR.

Our first plot reveals a consistent upward trajectory, aligning with the broader trend observed in the stock market. Noteworthy instances include a substantial reduction in price range around February and the end of October, indicative of potential market stability or specific influencing factors during those periods. The presence of a cluster of green candles in late March and mid-May suggests bullish market sentiment, corresponding to periods of increased closing prices compared to openings. Conversely, pronounced occurrences of red candles at the beginning of February, the end of July, and the end of October signal bearish tendencies.

Similarly, in the second plot, the overall trend of Nabil Bank's stock price is downward. Starting at around 850 NPR in January 2023, the stock price has consistently decreased, reaching below 500 NPR by the end of the year. This chart provides an analysis of Nabil Bank's stock with a bearish pattern and decreasing value over the year, often used for technical analysis by traders to assess price movements and potential entry/exit points.



Figure 26: Google Stock Price: Candlestick Representation



Figure 27: Nabil Stock Price: Candlestick Representation

1. **Variable Correlation**

The analysis of the relationships between stock market factors, such as opening, closing, highest and lowest prices, and trading volume, provides insight into the strength and direction of their interconnections. The correlation coefficient, which ranges from -1 to 1, can indicate a perfect positive correlation (where an increase in one variable leads to a corresponding increase in another), a perfect negative correlation (where an increase in one variable causes a decrease in another), or no linear correlation (suggesting a weak or non-existent relationship between the variables).

In the case of Google's stock market data, the correlation analysis shows a perfect positive relationship (correlation coefficient of 1) between the open, close, high, and low prices, demonstrating a strong linear association. On the other hand, volume demonstrates a moderate negative correlation (approximately -0.47) with the rest of the variables. The correlation analysis of Nabil Bank's stock prices shows a perfect positive relationship between the open, high, low, and close prices, highlighting a strong connection between these variables. However, volume exhibits a negative correlation (around -0.28) with the other variables, suggesting that as the price-related metrics rise, trading volume tends to slightly decline.

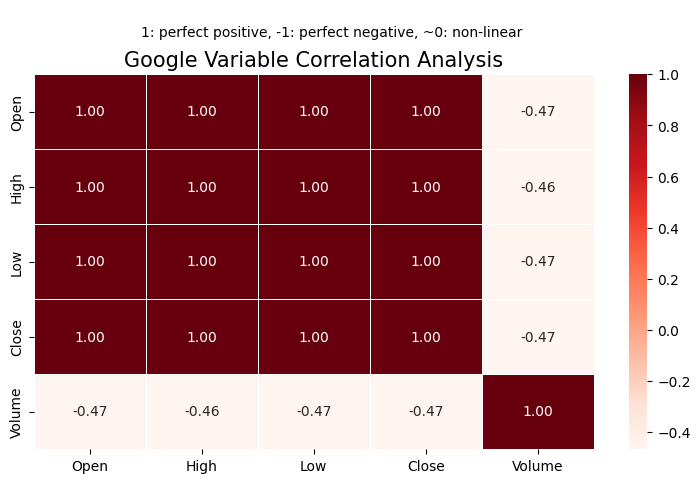


Figure 28: Google Stock Price Variable Correlation

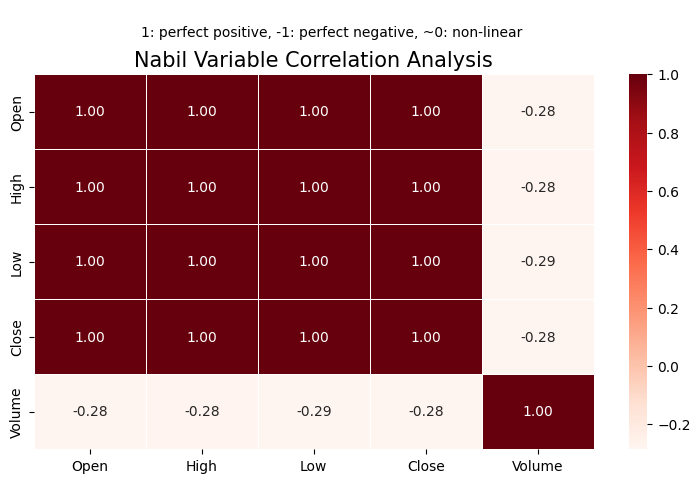


Figure 29: Nabil Stock Price Variable Correlation

The complete code for explanatory data analysis (EDA) can be found on project’s GitHub repository [Performance Evaluation and Analysis of Machine Learning Algorithms in Stock Price Prediction](https://github.com/GaurabStha/PERFORMANCE-EVALUATION-AND-ANALYSIS-OF-MACHINE-LEARNING-ALGORITHMS-IN-STOCK-PRICE-PREDICTION).

### **Data Splitting**

To train and evaluate the machine learning models, it is important to split the dataset into training and testing subsets. In this case, 80% of the data was used for training the model, while the remaining 20% was reserved for testing the model's performance. The datasets are separated using two different methods: the first one used is from **scikit learn library – (train\_test\_split) where we took test size of 0.2 that is 20% and random state of 42**. The second one that we used is splitting **first 80% data as train set and last 20% data as test set.**

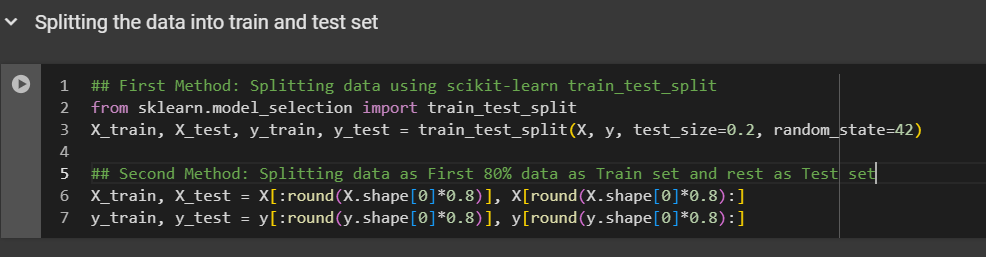


Figure 30: Different Train-Test Dataset Splitting Methods

## **Model Building**

After processing the data and engineering the features, the refined dataset was used to build, train, and evaluate the model. A range of machine learning and deep learning algorithms were applied in the model construction, which are detailed in the following sections:

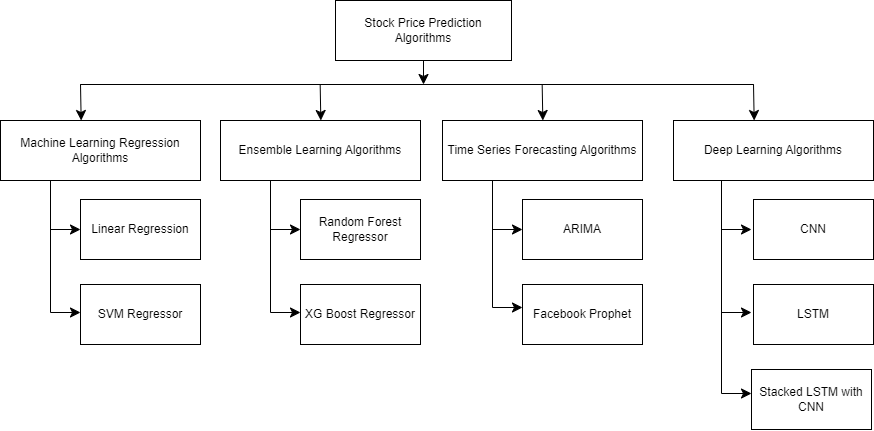


Figure 31: Algorithms Used for Predicting Stock Price

### **Basic Machine Learning Regression Algorithms**

Machine learning regression algorithms are employed to predict continuous numerical values based on input features. For the basic machine learning algorithms, we utilized two algorithms: Linear Regression and Support Vector Machine (SVM).

1. **Linear Regression**

Linear regression is one of the simplest and most widely used algorithms for predictive modelling in both statistics and machine learning. It establishes a relationship between a dependent variable and one or more independent variables by fitting a linear equation to the data (Kavita, 2024). In this project, we implemented multiple linear regression, an extension of simple linear regression that allows for the modelling of a dependent variable influenced by multiple independent variables (features). Its objective is to analyse the relationship between a single continuous outcome variable and two or more predictor variables.

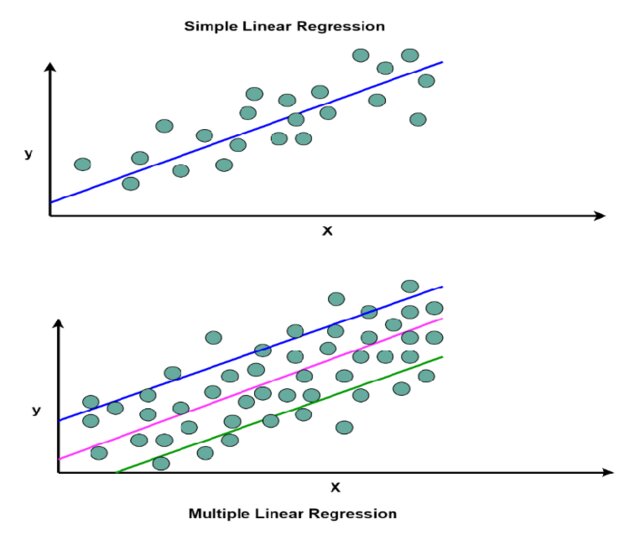


Figure 32: Graph Showing Simple Linear Regression and Multiple Linear Regression (Uddin, 2023)

The mathematical expression of Multiple Linear Regression is as follows:

*Y = β0 + β1​X1​ + β2X2 ​+…………………. + βnXn + ϵ*

Where:

* *Y* is the dependent variable that we are trying to predict.
* *X* represents the independent variable (feature).
* *β0* is the y-intercept, which indicates the predicted value of Y when X = 0.
* *Β1* is the slope of the line, representing how much Y changes for a one-unit increase in X.
* *ϵ* is the error term, which accounts for the difference between the predicted and actual values.
* *n* is the total number of observations.

1. **Support Vector Machine (SVM)**

Support Vector Machines (SVMs) were introduced by Vladimir Vapnik (Vapnik, 1999) and other researchers in the 1960s. The core concept of the SVM model is to create a linear classifier that identifies the optimal decision boundary among multiple possibilities. SVMs have gained popularity due to their strong performance, efficiency with medium-sized datasets, and ability to work well with CPU-based systems (Alakh, 2024). "Support vectors" refer to the data points closest to the decision boundary. SVMs are widely used in tasks such as classification, recognition, regression, and time series analysis. The main goal of SVM is to construct a hyperplane as a decision surface that maximizes the margin between positive and negative data points.

Support Vector Regression (SVR) is a variant of the Support Vector Machine (SVM) algorithm, tailored for regression problems. While SVM mainly focuses on classification tasks, SVR utilizes comparable concepts to forecast continuous values, such as stock prices (Otten, 2024). The objective of SVR is to identify a function that diverges from the actual observed values by a minimal margin (regulated by a parameter), while remaining as flat as feasible.

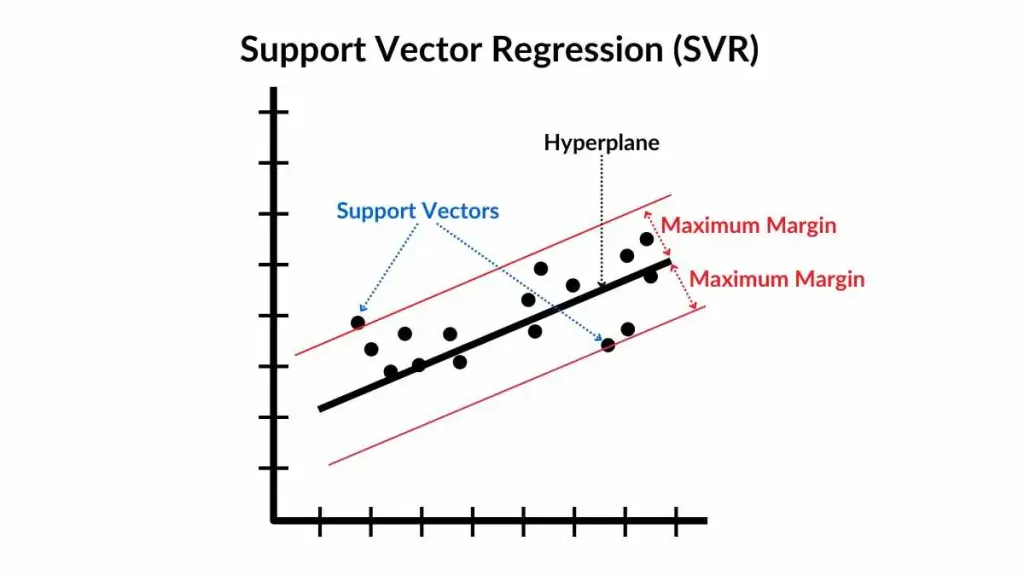
****

Figure 33: Support Vector Regressor Graph (Otten, 2024)

When the data exhibits non-linearity (which frequently occurs in stock price prediction), Support Vector Regression (SVR) can manage this by converting the input data into a higher-dimensional space through the use of a kernel trick. This kernel trick enables SVR to execute non-linear regression by applying a kernel function to the input data.

The common kernels include:

* **Linear kernel:** For linear relationships.
* **Polynomial kernel:** For polynomial relationships.
* **Radial Basis Function (RBF) kernel:** For more complex, non-linear relationships.

The transformation is performed implicitly using the kernel function, avoiding the need to compute the mapping explicitly. The regression model in the kernel space becomes:

***F(x) = i -*** α***i\*)K(xi, x) + b***

Where:

* ***i = n*** is number of observations.
* ***K(xi, x)*** is the kernel function that computes the similarity between the points ***xi and x.***
* ***αi and* αi\*** are Lagrange multipliers from the dual formulation of the optimization problem.

Support Vector Regression aims to fit a regression line within a defined margin (epsilon-tube), minimizing the errors beyond this margin and creating a smooth predictive model. With the ability to use kernel functions, SVR is flexible enough to handle both linear and non-linear relationships, making it a powerful tool for tasks like stock price prediction, where non-linear patterns are often observed.

### **Ensemble Learning Algorithms**

Ensemble learning algorithms are a set of machine learning techniques that combine the predictions of multiple individual models to achieve more accurate and reliable outcomes than any single model could offer (Brownlee, 2021). The core idea is that by merging the results of different models, the limitations of one model can be compensated for by the strengths of others, leading to improved generalization and performance. All ensemble methods are grounded in the Weak Law of Large Numbers (WLLN), which states that if each model in the ensemble has slightly above 50% accuracy and operates independently, increasing the number of models will make the overall accuracy of the combined prediction approach 100% through majority voting. In this discussion, we will focus on one bagging method, the Random Forest Regressor, and one boosting method, the XG-Boost Regressor.

1. **Random Forest Regression**

Random Forest Regression is an ensemble learning method that employs decision trees to tackle regression tasks by building multiple decision trees and combining their predictions (Baheshti, 2022). The fundamental concept behind Random Forest is that a collection of diverse, weak learners (decision trees) can collaborate to produce a strong and accurate predictive model. This approach involves averaging the outputs of individual decision trees, which reduces overfitting and improves the model's ability to generalize.

It is an ensemble learning approach known as bagging, where multiple models (typically of the same kind) are trained on different random samples of the dataset. The outcomes from all these models are then aggregated, usually by averaging (for regression) or voting (for classification), to enhance overall accuracy and lessen overfitting.

So, for a given input ***x***, the predicted value from each tree ***Ti(x)*** is averaged to get the final prediction:

***ŷ = i(x)***

Where:

* ***ŷ*** is the final prediction
* ***Ti(x)*** is the prediction from the ***i-th*** tree
* ***N*** is the total number of trees in the forest

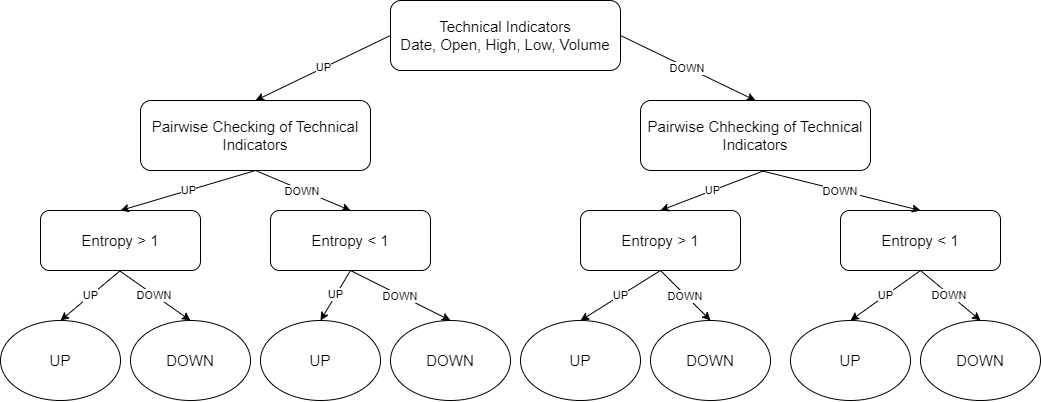


Figure 34: Single Decision Tree of Random Forest

1. **XGBoost Regression**

XGBoost (Extreme Gradient Boosting) is an advanced and highly efficient variant of gradient boosting, designed for improved speed and performance. It builds a sequence of decision trees, where each subsequent tree is trained to correct the errors made by the previous ones. For regression tasks, XGBoost focuses on minimizing the difference between predicted and actual values (the residuals) by iteratively fitting additional trees to these residuals (Brownlee, 2021).

XGBoost operates under the gradient boosting framework, which creates models in a sequential manner by introducing new trees that address the errors from earlier trees. The performance of the model enhances with each iteration by minimizing a loss function (such as Mean Squared Error for regression) through gradient descent methods. XGBoost constructs the model as an additive sequence of decision trees. If the prediction at the ***t-th*** step is denoted by ***ŷt***, the model at step ***t+1*** adds a new decision tree ***ft+1(x)*** to improve the previous predictions:

***ŷt+1(x) = ŷt(x) + ft+1(x)***

The idea is that each new tree ***ft+1(x)*** is trained to minimize the residuals from the previous prediction.

Below figure demonstrate the algorithm:

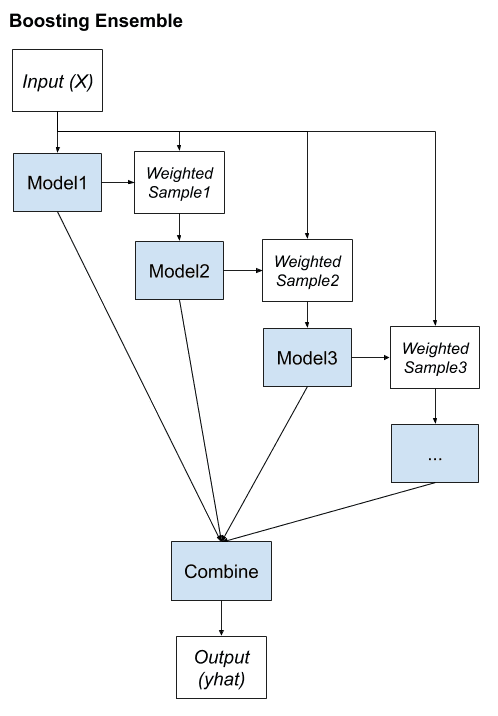


Figure 35: Flowchart showing Boosting Ensemble Learning (XGBoost)

The model is refined through gradient descent applied to a loss function, incorporating regularization to mitigate overfitting. Through the integration of numerous trees and the addition of randomness, XGBoost attains excellent predictive performance while maintaining resilience against overfitting, which makes it ideal for intricate tasks such as predicting stock prices.

### **Time Series Forecasting Algorithms**

Time series forecasting methods are utilized to estimate future outcomes based on historical data points, considering the temporal relationships within the data. These methods are frequently employed in fields such as finance, economics, meteorology, and supply chain management. ARIMA and FB Prophet were employed as time series forecasting techniques to develop the model.

1. **ARIMA**

ARIMA (Auto-Regressive Integrated Moving Average) is a widely used statistical method for forecasting time series data, combining three core components: autoregression (AR), integration (I), and moving average (MA) (Hayes, 2024). This technique is a form of regression analysis designed to explore the relationship between a dependent variable and various influencing factors. The primary objective of ARIMA is to predict future trends in securities or financial markets by analyzing the fluctuations in a time series' values rather than focusing on the exact values themselves. However, ARIMA can only handle a single feature, so incorporating SARIMAX (Seasonal Auto-Regressive Integrated Moving Average with exogenous variables) allows for the inclusion of additional features, such as Open, High, Low, and Volume, which enhance the stock price prediction model. SARIMAX models are particularly useful when seasonal patterns and external factors affect the target variable, such as the Close price in this context. Different components in ARIMA are:

1. **Auto-Regressive (AR) component:**

The autoregressive (AR) component of the model represents how a current observation depends on several preceding ones (previous time points). The parameter 𝑝 defines how many past observations are included in the model. The AR model can be represented as:

***Yt = ϕ1Yt-1 + ϕ2Yt-2 + ………. + ϕpYt-p + ϵt***

Where:

* ***Yt***  is the observed value at time ***t.***
* ***Φ1, ϕ2,….,*** ***ϕp*** are the autoregressive coefficients.
* ***ϵt*** is a white noise error term.

This equation shows that the current value ***Yt*** is influenced by the previous ***p*** values in the series.

1. **Integrated (I) component:**

The integrated component of the model is concerned with converting the time series into a stationary form, which is crucial for ARIMA modelling. A stationary time series has a constant mean and variance over time. The integration process involves differencing the data, represented by the parameter 𝑑, which indicates how many times the data must be differenced to achieve stationarity. Differencing a time series involves subtracting the previous value from the current one:

***Y’t = Yt – Yt – 1***

If the series is still non-stationary after the first difference, further differencing may be applied (e.g., second differencing) until the series becomes stationary.

1. **Moving Average (MA) component:**

The moving average component of the model captures the relationship between an observation and the residuals from a moving average model applied to prior observations (Saadeddin, 2024). The MA part is characterized by the parameter 𝑞, indicating the count of lagged forecast errors that are incorporated in the model. The MA model can be represented as:

***Yt = θ0 + θ1 ϵt-1 +*** ***θ2 ϵt-2 + …... + θq ϵt-q + ϵt***

Where:

* ***θ0*** is a constant term.
* ***Θ1, θ2, θt-q*** are the moving average coefficients.
* ***ϵt-1, ϵt-2, ϵt-q*** are the white noise error terms from previous time points.

So, combining these terms, the general form of the ARIMA model can be represented as:

***Yt = ϕ1Yt-1 + ϕ2Yt-2 + ………. + ϕpYt-p + θ0 + θ1 ϵt-1 +*** ***θ2 ϵt-2 + …... + θq ϵt-q + ϵt***

Where:

* ***p*** is the number of autoregression terms.
* ***d*** is the number of times the series has been differenced.
* ***q*** is the number of moving average terms.

1. **FB Prophet**

FB Prophet is a forecasting tool that is open-source and was created by Facebook's Core Data Science team, aimed at analysing time series data with significant seasonal trends and multiple years of historical data. It's especially effective for business predictions because of its user-friendly nature and resilience to missing values and anomalies. Prophet offers a straightforward method for modelling seasonal patterns and includes the impact of holidays, making it applicable for a range of uses such as predicting stock prices, forecasting sales, and analysing website traffic (Melanie, 2024).



Figure 36: FB Prophet Logo (Ankit, 2022)

FB Prophet operates on an additive framework that integrates three essential elements:

1. **Trend component:**

The trend component represents the general direction of the time series as it progresses over time. Prophet employs piecewise linear or logistic growth curves to depict trends.

The linear trend can be expressed as:

***g(t) = g0 + kt***

Where:

* ***g(t)*** is the value of the trend at time ***t.***
* ***g0*** is the initial value of the trend at ***t = 0.***
* ***K*** is the growth rate (slope) of the trend.

The logistic trend i.e., data is bounded can be represented as:

***g(t) =***

Where:

* ***C*** is the carrying capacity (the maximum value the time series can approach).
* ***m*** is the reflection point where the growth rate changes.

1. **Seasonal component:**

The seasonal component captures periodic fluctuations in the data. Prophet models’ seasonality using Fourier series to approximate seasonal effects over different periods (e.g., daily, weekly, yearly).

The seasonal effect is modeled as:

***s(t) =***

Where:

* ***s(t)*** is the seasonal effect at time ***t.***
* ***an and bn*** are the Fourier coefficients for each frequency component ***n.***
* ***P*** is the period of the seasonality.
* ***N*** is the number of Fourier terms included in the model to capture the seasonal pattern’s complexity.

1. **Holiday component:**

FB Prophet enables users to incorporate significant events or holidays that may influence the time series, enhancing the model further. Users specify the holiday impacts, and the model integrates extra components to account for these effects.

The holiday effect can be modeled as:

***h(t) =***

Where:

* ***h(t)*** is the holiday effect at time ***t.***
* ***Ij(t)*** is an indicator function that is 1 if holiday ***j*** occurs at time ***t*** and ***0*** otherwise.
* ***δj*** is the impact of holiday ***j.***

Combining these components, the overall model can be expressed as:

***Y(t) = g(t) + s(t) + h(t) + ϵt***

Where:

* ***Y(t)*** is the predicted value at time t
* ***g(t)*** is the trend component.
* ***s(t)*** is the seasonal component.
* ***h(t)*** is the holiday effect.
* ***ϵt*** is the error term, assumed to be normally distributed with mean zero.

FB Prophet enables users to incorporate special events or holidays that could impact the time series, which enhances the model. Users specify the holiday effects, and the model integrates additional components for these effects.

### **Deep Learning Algorithms**

Deep learning algorithms belong to the category of machine learning and utilize neural networks with multiple layers (hence the term "deep") to capture intricate patterns within large datasets. We employed CNN, LSTM, and a combination of CNN and LSTM (stacked CNN-LSTM) to create the predictive model.

1. **CNN**

Convolutional Neural Networks (CNNs), originally designed for tasks in computer vision, have also proven effective for processing sequential data, such as stock prices. The fundamental idea of CNNs is to identify local patterns within the data using convolutional processes. CNNs primarily target the detection of temporal patterns that are essential for forecasting time series values while providing strong generalization and resilience to noise. Their capability to extract abstract features and reveal hidden non-linear relationships in the data, all without the need for human input, makes CNNs a compelling alternative to conventional feature-based methods (Gunduz, et al., 2017).

Stock prices are time series data that exhibit patterns over time, such as trends, cycles, and volatility shifts. CNNs can capture these local patterns (such as a short-term trend or specific behaviour over a period) through the use of filters or kernels in the convolution layer. This capability makes CNNs suitable for identifying useful features that can help predict future price movements.

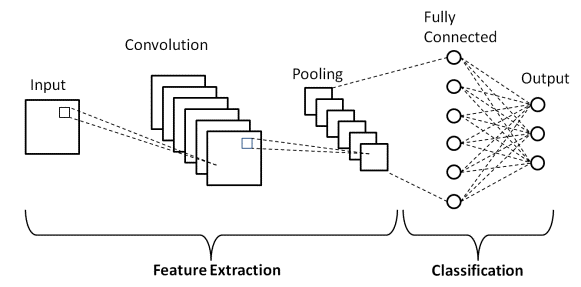


Figure 37: Layers of CNN

The CNN is composed of five layers: the Input layer, Convolutional layer, Pooling layer, Fully Connected layer, and Output layer. The crucial layers within a CNN are the hidden layers, which are situated between the input and output layers.

1. **Convolutional Layer:**

The convolutional layer employs several filters (or kernels) to detect local features from the input data. Each filter is a small matrix (such as one-dimensional for time series) that moves along the time axis to identify pattern. The main mathematical operation involved is convolution:

***Zi, j =***

Where:

* ***Wm*** is the weight of the filter at position ***m.***
* ***Xi+m-1, j*** is the input at position ***i + m – 1*** for feature ***j.***
* ***b*** is the bias term.
* ***Zi, j*** is the result after applying the filter at position ***i.***

1. **Pooling Layer:**

A pooling layer is commonly used after the convolutional layer to reduce the size of the feature maps, improving the network's efficiency. Max pooling is a commonly used pooling technique that selects the highest value from a defined window of values:

***Pi, j = max (0, Zi: i + f - 1, j)***

Where:

* ***F*** is the pooling window size.
* ***Zi, j*** is the output from the convolution layer.
* ***Pi, j*** is the pooled layer.

1. **Fully Connected Layer:**

In the fully connected layer, each neuron is connected to every element of the flattened vector. This layer combines the features detected by the convolutional layers to produce the final prediction. The output is calculated as:

***y = σ(Wfc ⋅ P + bfc)***

Where:

* ***Wfc*** is the weight matrix of the fully connected layer.
* ***P*** is the pooled output.
* ***Bfc*** is the bias term.
* ***Σ*** is the activation function.

CNNs are useful for stock price prediction because they can capture localized patterns over time, such as trends or volatility patterns, and they are computationally efficient due to weight sharing and parameter reduction in the convolution layers. When used with other models (e.g., LSTM or GRU), CNNs can help extract meaningful features from time series data for more accurate predictions.

1. **LSTM**

LSTM (Long Short-Term Memory) is a type of Recurrent Neural Network (RNN) designed to handle and learn long-term dependencies in sequential data. Unlike traditional RNNs, which face challenges in retaining information over long sequences due to problems like vanishing or exploding gradients, LSTMs feature a specialized structure that allows them to selectively retain or discard information over time. This capability makes LSTMs particularly effective in time series prediction tasks, such as stock price forecasting, as they excel at identifying both short-term and long-term patterns in sequential data.

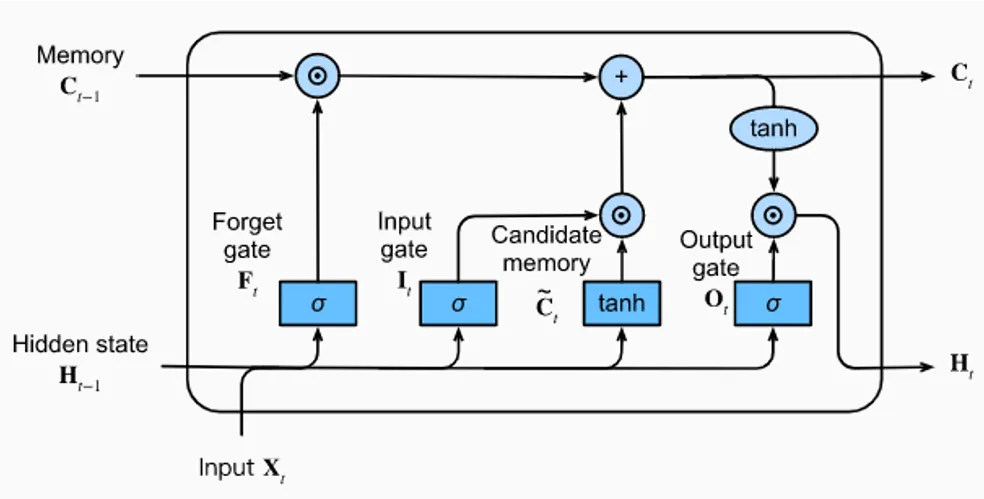


Figure 38: Components of LSTM (Calzone, 2022)

The LSTM structure have mainly four components:

1. **Forget Gate *ft***

The forget gate controls which parts of the previous cell state (Ct-1) should be kept or discarded. Using a sigmoid activation function, it generates an output between 0 and 1, indicating the degree to which each element of the previous cell state should be remembered or forgotten.

***ft = σ (Wf ⋅ [ht-1, xt] + bf)***

Where:

* ***Ft*** is the forget gate,
* ***Wf*** is the weight.
* ***ht-1*** is the previous hidden state.
* ***xt*** is the input at the current time step.
* ***σ*** is the sigmoid activation function.
* ***bf*** is the bias term.

1. **Input Gate *if***

The input gate controls the amount of new information to be incorporated into the cell state. It uses a sigmoid activation function to determine which values should be updated, allowing the network to adjust its memory based on the current input.

***it = σ (Wi*** ***⋅ [ht-1, xt] + bi)***

***C̃t = tanh(Wc⋅[ht-1, xt] + bc)***

Where:

* ***it***
* ***Wi***
* ***C̃t***
* ***Wc*** is the weight matrix for generating the candidate cell state.
* ***tanh*** is the hyperbolic tangent function that ensures the values of ***C̃t*** lie between -1 and 1
* ***bi and bc*** are bias.

1. **Update Cell State *Ct***

The cell state ***Ct*** is updates by combining the previous state ***Ct-1*** controlled by the forget gate and the new candidate state ***C̃t*** controlled by the input gate.

***Ct = ft ⋅ Ct-1 + it ⋅  C̃t***

Where:

* ***ft ⋅ Ct-1*** represents the retained old cell state.
* ***it ⋅  C̃t*** represents the added new information.

1. **Output Gate *ot***

The output gate determines the hidden state ***ht***, which is used for the next time step and serves as the output of the current cell.

***ot = σ (W0 ⋅ [ht-1, xt] + b0)***

***ht = ot ⋅ tanh(Ct)***

Where:

* ***ot*** is the output gate.
* ***ht*** is the hidden state, which represents the output for the current time step.
* ***W0*** is the weight matrix for the output gate.
* ***Tanh(Ct)*** is the non-linear transformation of the updated cell state

1. **Stacked CNN-LSTM**

A Stacked CNN-LSTM architecture integrates Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks sequentially to derive both spatial and temporal characteristics from stock price information. This model utilizes the CNN's capacity to identify local patterns (such as short-term price fluctuations or trends) and the LSTM’s capability in modelling long-term dependencies within time series data.

Stock prices are sequential data that also exhibit localized patterns, such as short-term price movements or volatility spikes. CNNs can capture these localized patterns efficiently, while LSTMs capture long-term temporal dependencies. By stacking CNN layers on top of LSTM layers, the model can first extract features using CNNs and then learn long-term dependencies with LSTMs. This hybrid approach is effective for predicting stock prices because it incorporates both local feature extraction and temporal sequence learning.

## **Hyperparameter Tuning**

Hyperparameter tuning refers to the process of selecting the optimal hyperparameters for a machine learning model (Jordan, 2017). Without tuning, a model may not perform at its best, as poor hyperparameters can lead to suboptimal outcomes, underfitting, or overfitting. The goal of hyperparameter tuning is to strike a balance between model complexity and its generalization capability. Common methods for hyperparameter tuning include Grid Search, Random Search, and others.

For our research project, we used Grid Search for basic machine learning algorithms as this method try every possible combination of hyperparameters from a specified grid which is exhaustive and guaranteed to find the best combination from the grid. But it is computationally expensive that scales poorly with large grid. But for deep learning algorithms, we are using random search which will try random combination of hyperparameters from the specified grid.

### **Tuning Parameters Used in Different Algorithms**

## **Performance Evaluation Metrics**

In this research project, our goal was to evaluate and analyse the performance of machine learning models in predicting stock prices. To accomplish this, we calculated performance evaluation metrics for each machine learning model. These metrics are essential in assessing the effectiveness and accuracy of machine learning (ML) and deep learning (DL) models. The key performance evaluation metrics used in our study to assess the machine learning models are summarized below.

### **Mean Squared Error (MSE)**

The Mean Squared Error (MSE) measures the average of the squared differences between predicted values and actual results. By squaring the differences, it applies a greater penalty to larger mistakes compared to smaller ones, which makes this metric responsive to significant prediction errors (Practicus AI, 2019). It is the most straightforward and widely used loss function.

Formula for **MSE**:

**2**

Where:

* N is the number of data points
* is the actual value
* is the predicted value

If we have lower MSE, it is said to be better model performance. Squaring the error makes MSE more sensitive to large errors (outliers), making it suitable when large deviations are undesirable. MSE penalizes large errors more than small errors helping to avoid models that make significant mistakes. The disadvantage of this is if our model makes a particularly poor prediction, the squaring in the function amplifies the error. However, in many real-world scenarios, we aren’t overly concerned with these outliers and are more focused on developing a balanced model that performs adequately for the majority of cases. Also, it is difficult to interpret because it’s not in the same units as the target variable.

### **Root Mean Square Error (RMSE)**

The Root Mean Squared Error (RMSE) is the square root of the MSE, allowing the error metric to be interpreted in the same units as the target variable. It retains the benefits of MSE (penalizing large errors) but provides results that are easier to interpret (Olumide, 2023).

To calculate the RMSE, we can root the MSE or use the formula,

**2**

Where:

* N is the number of data points
* is the actual value
* is the predicted value

The model is said to be performing better of the RMSE value is lower. It is easier to understand and preferred compared to MSE because it is in the same units as the output. It is also sensitive to outliers. It penalizes large error similarly to MSE but with an interpretable scale.

### **Mean Absolute Error (MAE)**

The Mean Absolute Error (MAE) measures the average absolute difference between predicted outcomes and actual results. It provides a simple metric in the same unit as the target variable, making it easy to understand.

Formula for MAE,

Where:

* N is the number of data points
* is the actual value
* is the predicted value

Lower MAE indicates better model performance. It is simple to calculate and understand. It treats all errors equally since it doesn’t square the differences, unlike MSE. MAE is robust to outliers compared to MSE because outliers aren’t squared. It is directly interpretable in terms of the unit of the target variable but it lacks for large errors, so might underestimate models with a few significant errors.

### **Computational Time (Training Time)**

Computational Time refers to the total time taken by an algorithm or model to perform a specific task, which can include training, validation, testing, and inference (prediction). Understanding computational time is essential for evaluating the efficiency and scalability of machine learning models, especially in production environments where speed and resource management are critical (Qayyum, 2022).

The time it takes to compute, especially during training, is a vital metric for evaluating performance in machine learning, especially as models grow more complex and datasets expand. It indicates the duration needed to train a model with a specific dataset. Grasping training time is important for various reasons, such as managing resources, choosing models, and the practicality of implementing models in real-life scenarios. Although performance metrics and accuracy are critical, understanding the duration required to train a model offers valuable insights into its practicality and feasibility.

## **Project Implementation**

Our research project “**Performance Evaluation and Analysis of Machine Learning Algorithms in Stock Price Prediction**” is focused on predicting stock price with different machine learning algorithms and comparing, evaluating and analysing the performance of these algorithms in predicting the stock price. So, when implementing a machine learning project for stock price prediction, it’s essential to follow a structured approach that ensures accurate performance evaluation.

### **Implementation Details**

For the project, after all the project requirements were analysed, the project was set to develop with system designed along with developing project methodology and flowchart. Firstly, the dataset for the project was collected from different website or the data was scrapped from the different website. The datasets were imported in the integrated development environment (IDE) using Pandas library. The data were then pre-processed and feature engineered with the process like removing irrelevant features, removing outlier data, fulfilling missing values, transforming the data into required formats. The data were scaled with minmax scaler which helps ensure that machine learning models perform optimally by eliminating issues arising from differences in feature scales. It improves model convergence, provides regularization, supports distance-based metrics, and enhances the interpretability of results, making it an essential step in numerous machine learning workflows. Following data pre-processing and feature engineering, the dataset is divided into a training set and a testing set for training and evaluating the machine learning model. We utilized 80% of the dataset for model training and reserved 20% for testing the machine learning model.

After processing the data, various machine learning and deep learning techniques, such as Linear Regression, SVM, Random Forest, XG Boost, LSTM, and CNN, were utilized to create predictive models for future data. We implemented these algorithms using the scikit-learn and TensorFlow libraries in Python. To enhance performance, we fine-tuned the hyperparameters of each algorithm, which aids in achieving optimal prediction accuracy. For the assessment and analysis of the machine learning and deep learning models, we measured MSE, RMSE, MAE, and computational time, particularly focusing on training duration.

The complete code, setup, and implementation can be found on project’s GitHub repository [Performance Evaluation and Analysis of Machine Learning Algorithms in Stock Price Prediction](https://github.com/GaurabStha/PERFORMANCE-EVALUATION-AND-ANALYSIS-OF-MACHINE-LEARNING-ALGORITHMS-IN-STOCK-PRICE-PREDICTION).

### **Implementation Challenges**

Implementing machine learning models for stock price prediction presents several challenges from data collection to model’s performance evaluation. We used two stock price datasets: **Google Stock Price dataset** and **Nabil Bank Stock Price dataset.** Google stock price dataset can be found anywhere from different websites like Kaggle, Yahoo Finance, etc. but the Nabil stock price dataset is hard to find in downloadable form in any of the websites in a proper format. So, we had to scrape the data from the dedicated website that required the legal approval. Also, the data scrapping process was too lengthy due to the time required for the data to be scrapped. Stock prices are inherently volatile, non-stationary, and influenced by many external factors, making it difficult for models to generalize well from historical data to future events. Common issues include data leakage, overfitting to noisy historical data, and dealing with limited historical information, all of which can result in misleading performance metrics. Additionally, selecting appropriate evaluation metrics is crucial. Standard metrics like MSE or RMSE might not reflect true performance in financial applications, where predicting trends or direction is more critical.

Another significant challenge is the availability of computational resources, particularly when working with complex algorithms such as CNN, LSTM and stacked models (CNN-LSTM) of these algorithms. These algorithms require substantial computing power and long training times due to their high computational demands. As the dataset grows and the number of parameters increases, the complexity of these models may become unmanageable with limited computing resources.

# **Results, Analysis and Discussion**

This section outlines the experiments conducted, their outcomes, and the results produced by the model developed for our research. We employed two datasets for our analysis: 1. Google Stock Price dataset 2. Nabil Bank Stock Price dataset. These datasets are utilized to create prediction models by applying various machine learning and deep learning techniques. A range of performance evaluation metrics is implemented to evaluate the effectiveness of these algorithms.

## **Results**

For the study, all machine learning and deep learning algorithms except FB Prophet were trained with dataset having same features, same scaling process (standard scalar) and same amount of data. FB Prophet algorithm takes two columns from the dataset: Date and Close price columns. And to train the model with **FB Prophet**, we used all price data except last 30 days price data which is used for testing the model.

Time series algorithm, **ARIMA** can only take one feature for training so we took the extended form of ARIMA- **SARIMAX**. SARIMAX can take multiple features which take order parameters that contains 3 values as **p** (Order of Auto Regressive), **d** (number of times the series is differentiated), **q** (order of moving average) and seasonal parameters which is alike order parameter but have seasonal component. This order is to be determined in order to forecast values with SARIMAX. For this, we used ‘**pmdarima’** which is a python library for automating hyperparameter tuning and simplifying the forecasting pipeline. For the deep learning models, we employed a standard batch size of **32** and ran for **50 epochs**. The batch size indicates the amount of data points evaluated before the model adjusts its weights, while epochs signify a full iteration over the whole training dataset throughout the training period.

After conducting experiments on two datasets using nine different types of machine learning, deep learning, and time series algorithms, it was discovered that nearly all models achieved a prediction accuracy score of 99%. While accuracy is a clear and intuitive metric, it is often insufficient, particularly in complex, imbalanced, or domain-specific scenarios. Performance evaluation metrics offer a more comprehensive and detailed insight into model performance, addressing subtleties that accuracy alone may overlook (Jain, 2024). Consequently, MSE, RMSE, and MAE were utilized to assess the performance of the machine learning and deep learning algorithms in this study. The results of the experiments were organized to compare the performance evaluation metrics and computational efficiency (training time) of various algorithms.

As, two methods were employed to split the data into train set and test set, we calculated performance evaluation metrics for each but we calculated training time only for the model which is trained with data that is split with **scikit-learn’s** **model\_selection** library **train\_test\_split()** method. The results and graphs from the experiment with two datasets are discussed below.

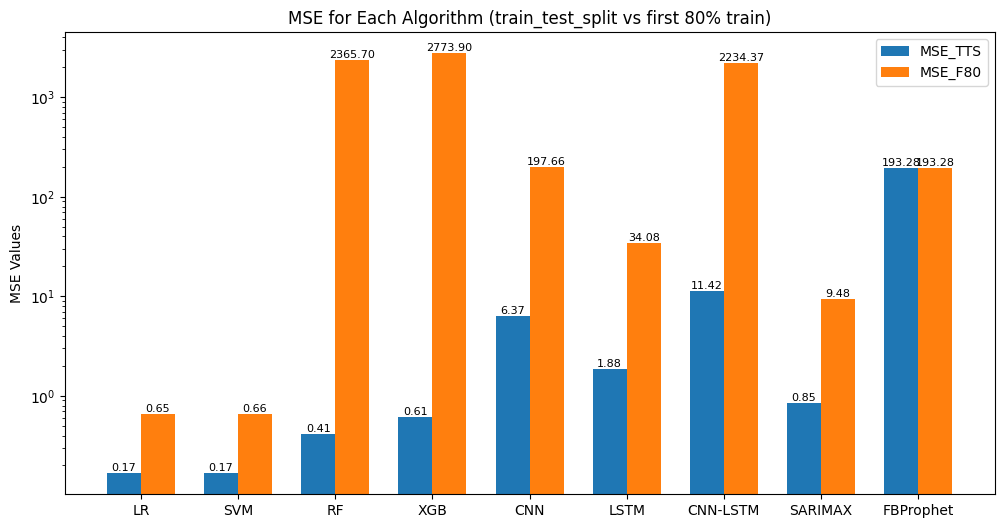
### **Result from experiment on Google (GOOGL) Stock Price Dataset**

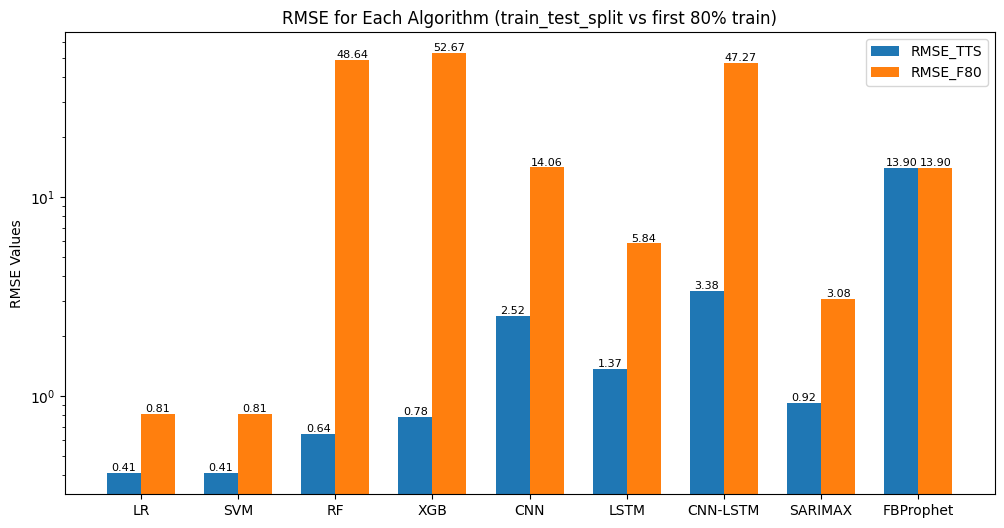
After experimenting on Google stock price dataset, we got the following performance from different algorithms that is presented in table 2.

Table 2: Performance Metrics of Different Algorithms in Google Stock Price Dataset

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm | Mean Squared Error (MSE) | | Root Mean Squared Error (RMSE) | | Mean Absolute Error (MAE) | | Training Time (sec) |
|  | **Train Test Split** | **First 80% Train** | **Train Test Split** | **First 80% Train** | **Train Test Split** | **First 80% Train** | **Train Test Split** |
| LR | 0.168 | 0.653 | 0.410 | 0.808 | 0.218 | 0.626 | 0.023 |
| SVM | 0.167 | 0.660 | 0.409 | 0.812 | 0.217 | 0.630 | 1.283 |
| RF | 0.415 | 2365.704 | 0.644 | 48.638 | 0.325 | 41.600 | 2.320 |
| XGB | 0.614 | 2773.901 | 0.784 | 52.667 | 0.374 | 46.262 | 2.679 |
| CNN | 6.372 | 197.659 | 2.524 | 14.059 | 1.747 | 13.114 | 131.116 |
| LSTM | 1.879 | 34.076 | 1.371 | 5.837 | 0.741 | 4.745 | 1934.98 |
| CNN-LSTM | 11.416 | 2234.370 | 3.378 | 47.269 | 2.853 | 40.717 | 201.006 |
| SARIMAX | 0.846 | 9.485 | 0.920 | 3.079 | 0.768 | 2.689 | 30.902 |
| FB-Prophet | 193.277 | | 13.902 | | 10.923 | | 6.552 |

From the table, performance of different machine learning, deep learning and time series forecasting algorithms based on their MSE, RMSE, MAE and training time has been observed.





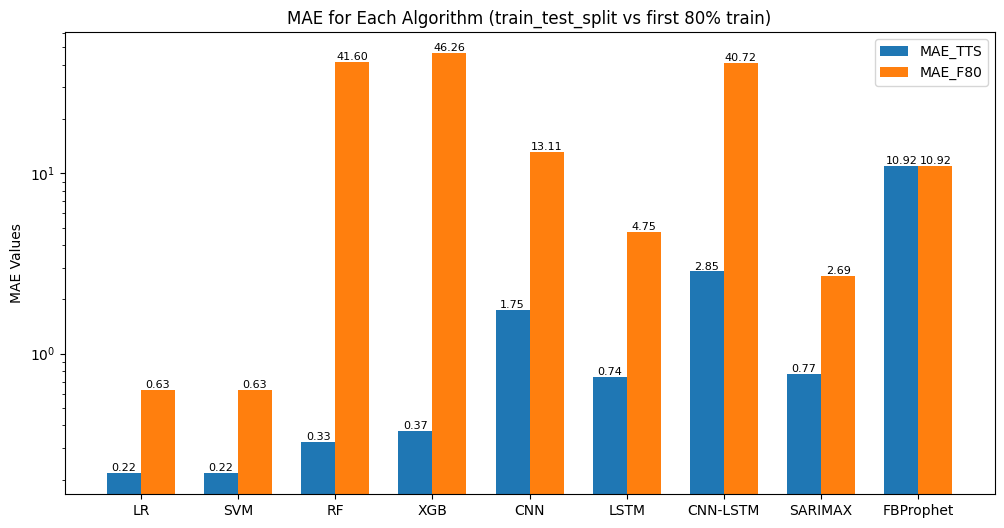


Figure 39: Performance Comparison of Different Model in Google Stock Price Data (train\_test\_split vs first 80% train)

In the above bar charts, bar with blue colors are for the machine learning models trained with training data from **train\_test\_split()** method and bar with orange colors are for the machine learning models trained with first 80% of the dataset. Based on their MSE, RMSE, and MAE, in basic machine learning algorithms, Linear Regression and Support Vector Machines are performing well with lowest MSE (0.17), RMSE (0.41) and MAE (0.22) trained with training data from **train\_test\_split0**. In deep learning models, LSTM outperformed all other algorithms with MSE 1.88, RMSE 1.37 and MAE 0.74 with training data from **train\_test\_split0**. Looking at the models trained with **first 80% data**, all performance evaluation metrics are bit higher than the models trained with training data from **train\_test\_split0**. However, Linear Regression and Support Vector Machine performed well than other algorithms. Looking at the ensemble learning algorithms, they are performing well when trained with training data from **train\_test\_split()** method but when trained with first 80% data, they are not performing well as they have the highest MSE, RMSE and MAE. In time series forecasting algorithms, SARIMAX is performing better than FB Prophet with MSE 0.85, RMSE 0.92 and MAE 0.77 trained with training data from **train\_test\_split().**

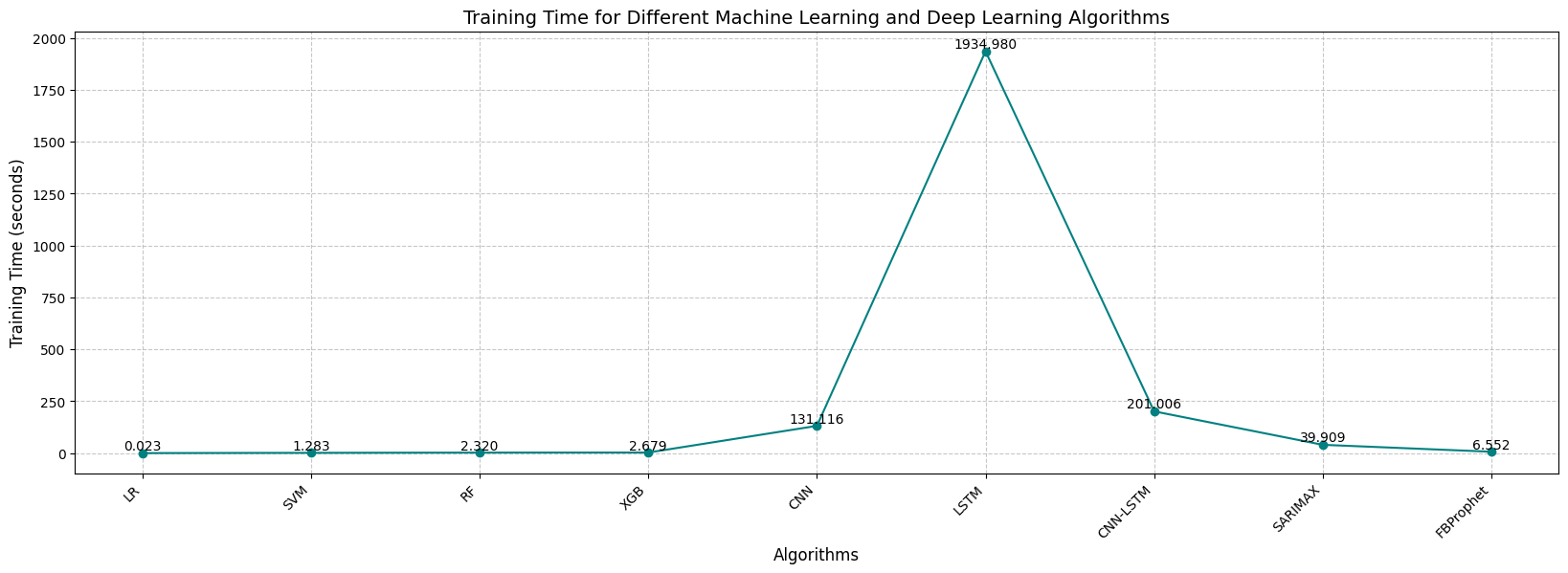


Figure 40: Training Time for Different Machine Learning Models in Google Stock Price Dataset

The figure above illustrates the training times required for each algorithm, highlighting the relative computational efficiency of various methods. The line graph represents different models along the horizontal axis while the vertical axis indicates the training time in seconds. Among the algorithms, linear regression, being the simplest, has the shortest training duration at just 0.023 seconds. In contrast, LSTM shows the longest training time at 1934.98 seconds, followed closely by the stacked hybrid model CNN-LSTM, which takes 201.006 seconds to train. This extended training time indicates the presence of larger parameter spaces and more complex computations that these models necessitate, often resulting in improved accuracy but also requiring more computational resources. Basic machine learning and ensemble learning algorithms exhibit significantly shorter training durations compared to deep learning algorithms, which tend to be more efficient in terms of training time due to their simpler nature and lower processing demands.

The line graphs below compare Google's actual closing stock prices to the predicted closing prices generated by various machine learning, deep learning, and time series forecasting algorithms. For each algorithm, the first graph displays the results from models trained using the **train\_test\_split()** method, while the second graph presents outcomes from models trained on the **first 80% of the data**.

1. **Linear Regression**

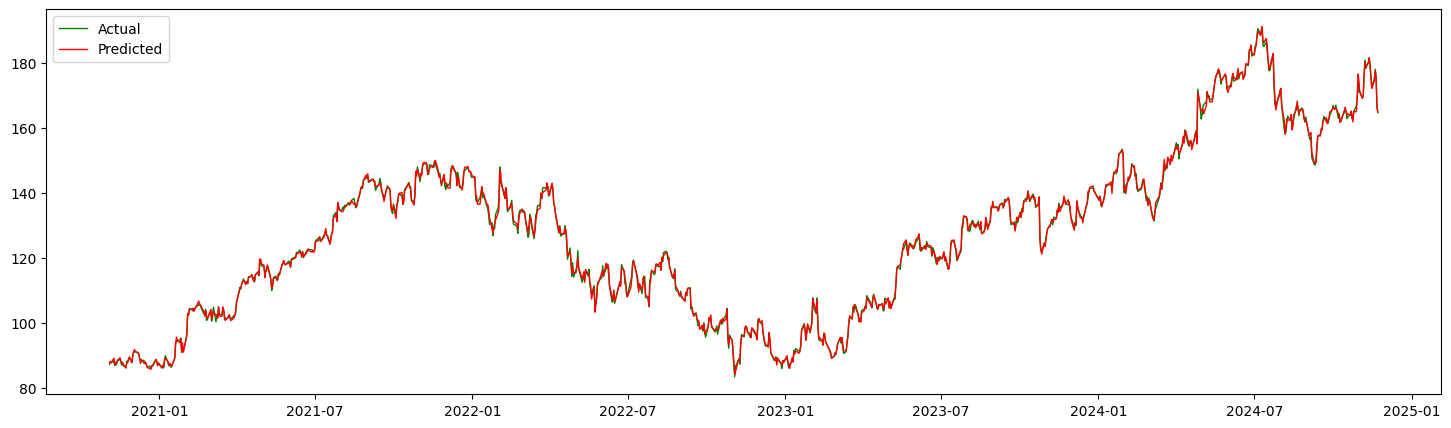
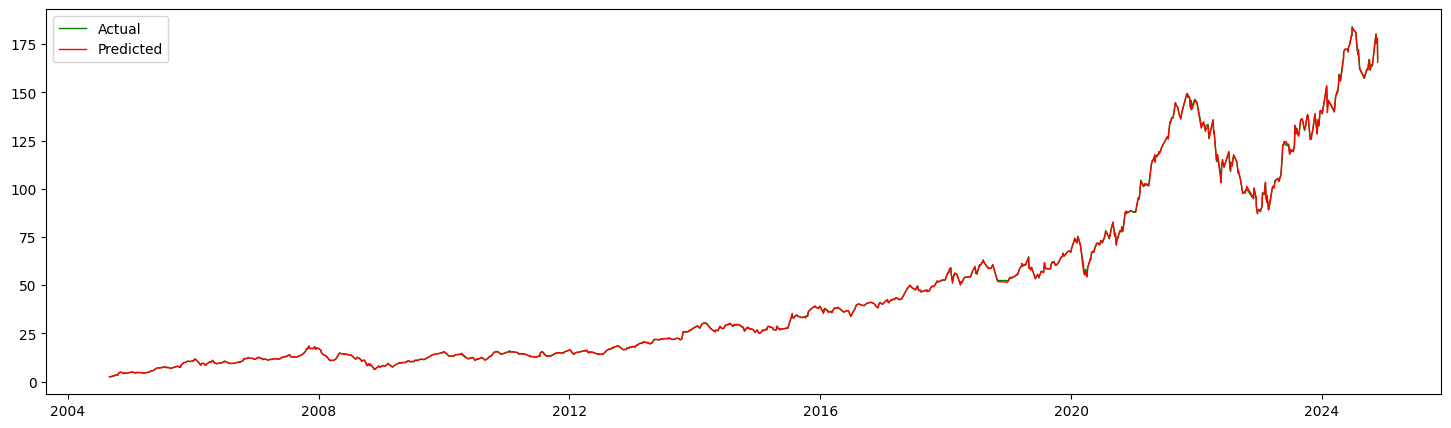
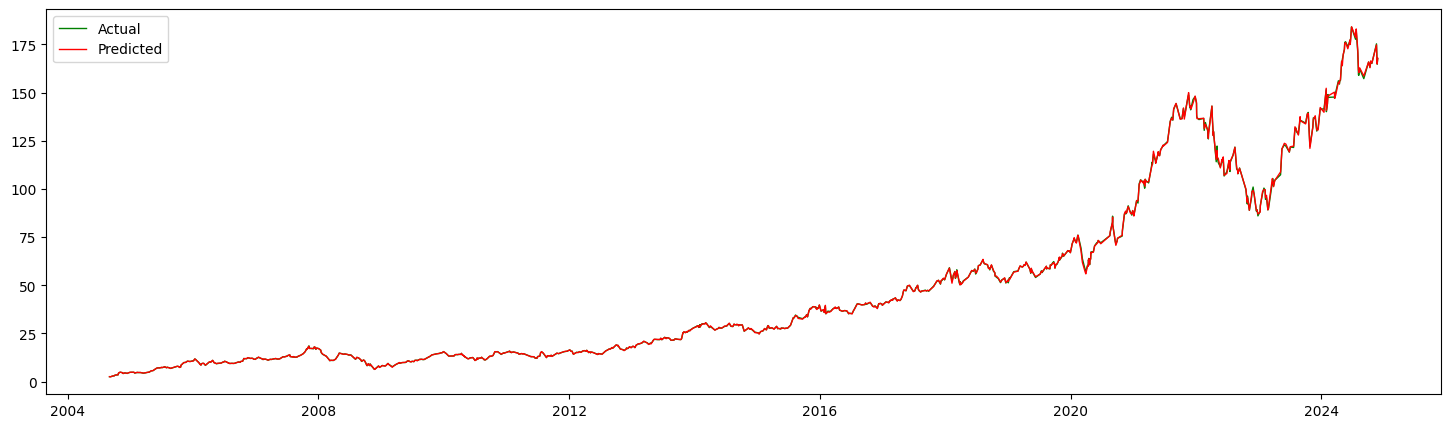
****

Figure 41: Result from Linear Regression (train\_test\_split() vs first 80% train)

1. **Support Vector Machine**

****

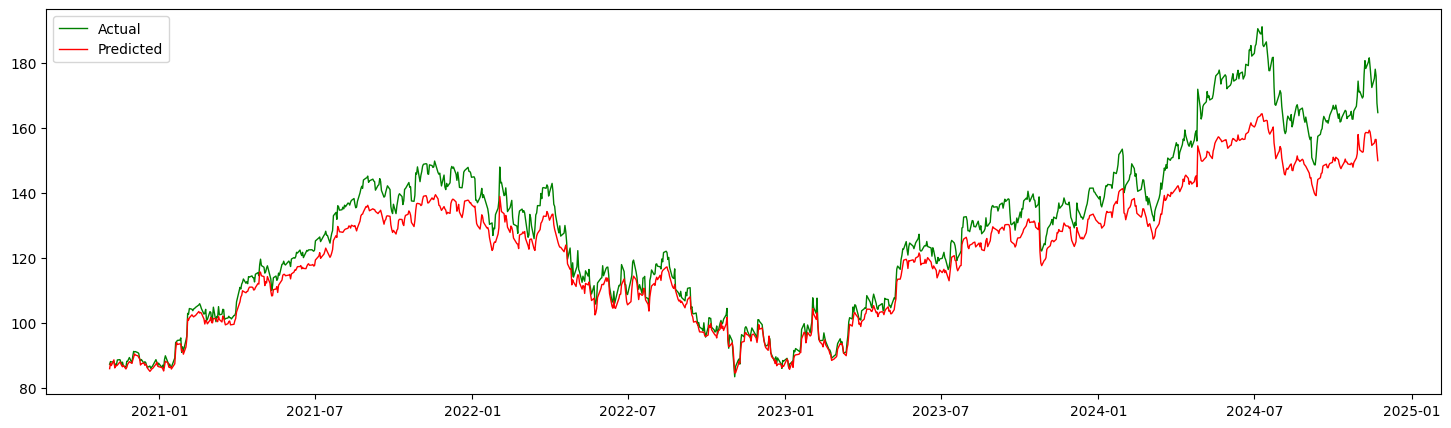
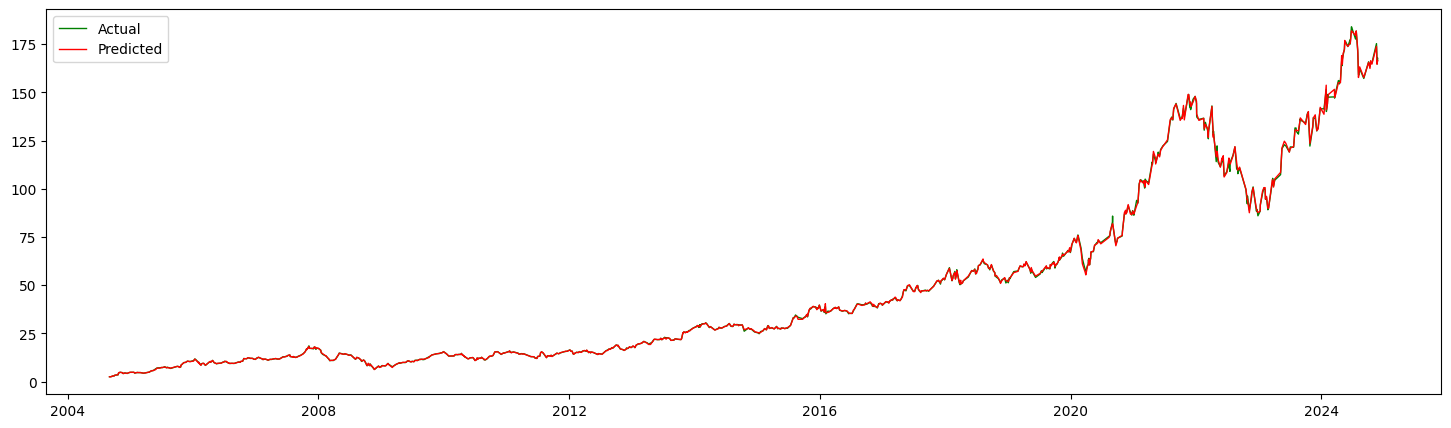
****

Figure 42: Result from SVM (train\_test\_split() vs first 80% train)

1. **Random Forest**

****

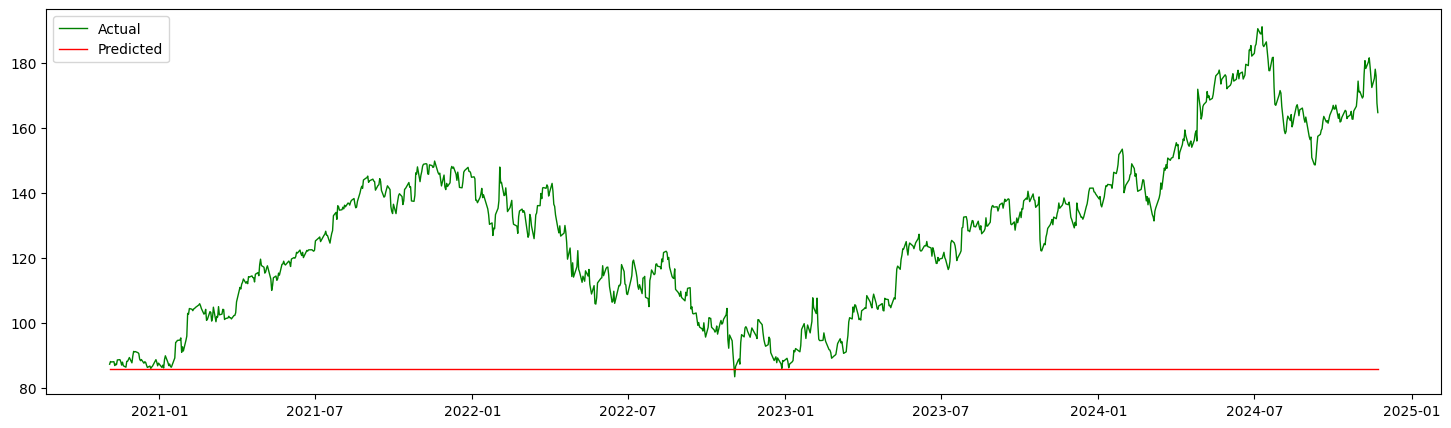
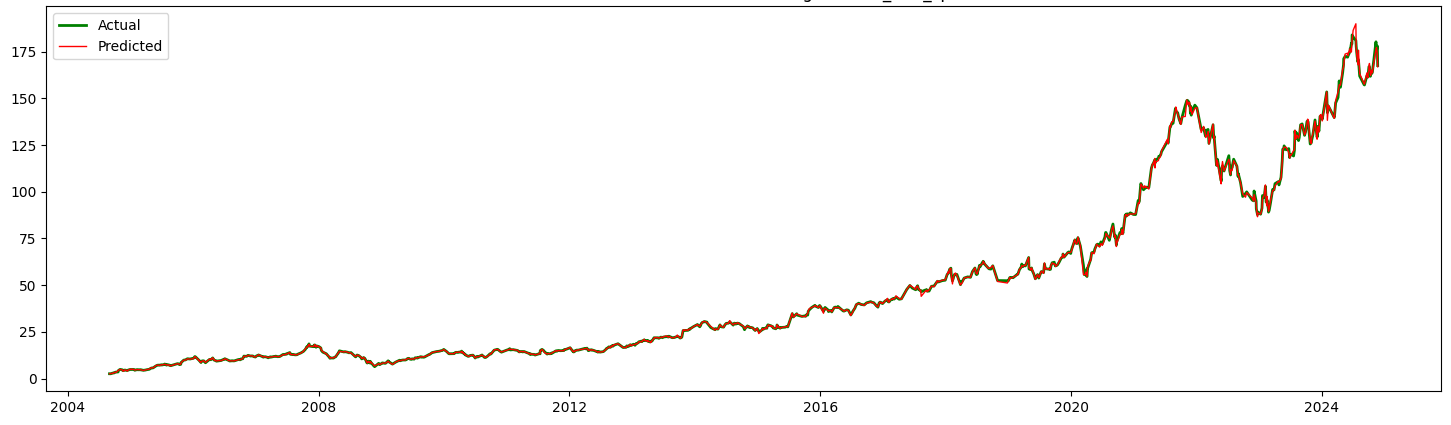
****

Figure 43: Result from Random Forest (train\_test\_split() vs first 80% train)

1. **XG Boost**

****

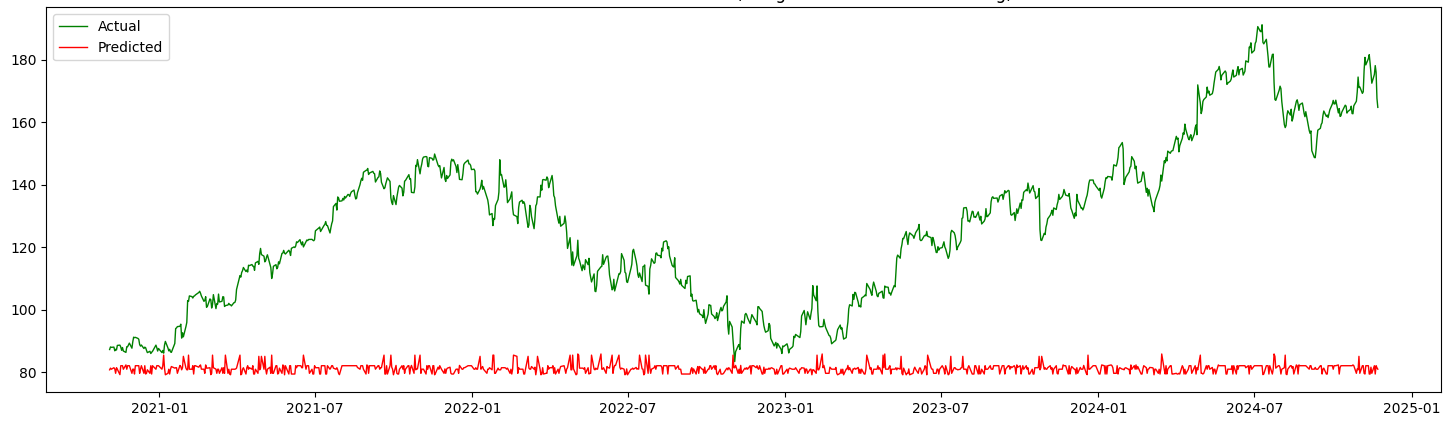
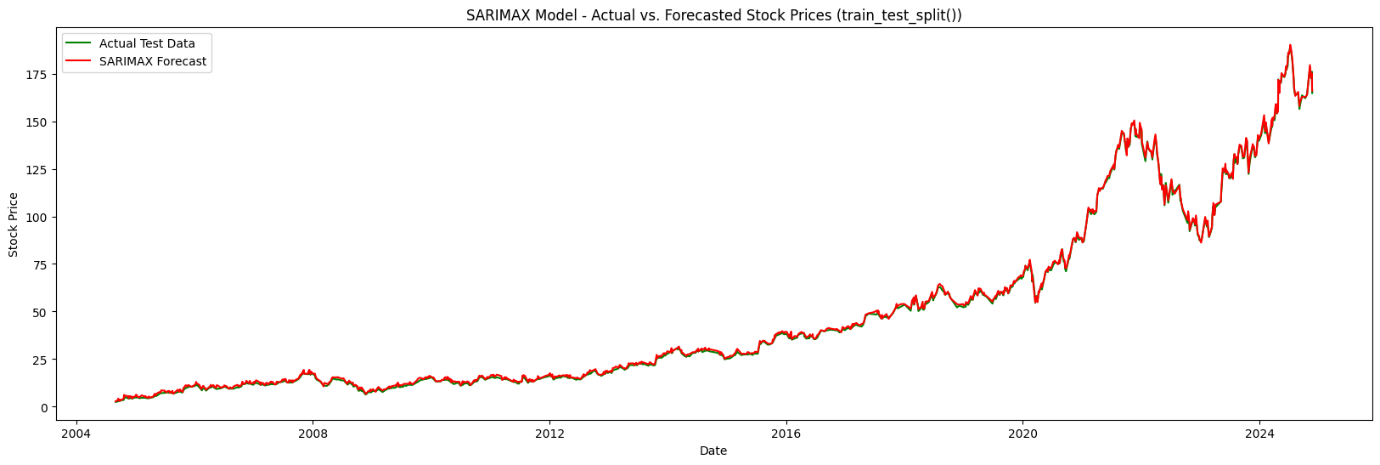
****

Figure 44: Result from XGBoost (train\_test\_split() vs first 80% train)

1. **SARIMAX**

****

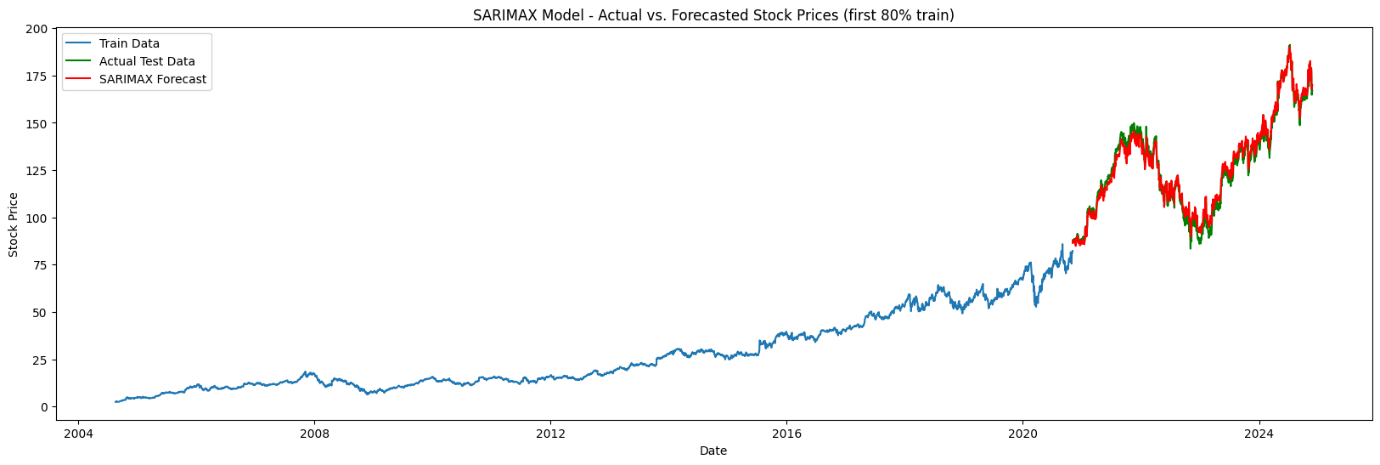
****

Figure 45: Result from SARIMAX (train\_test\_split() vs first 80% train)

1. **FB Prophet**

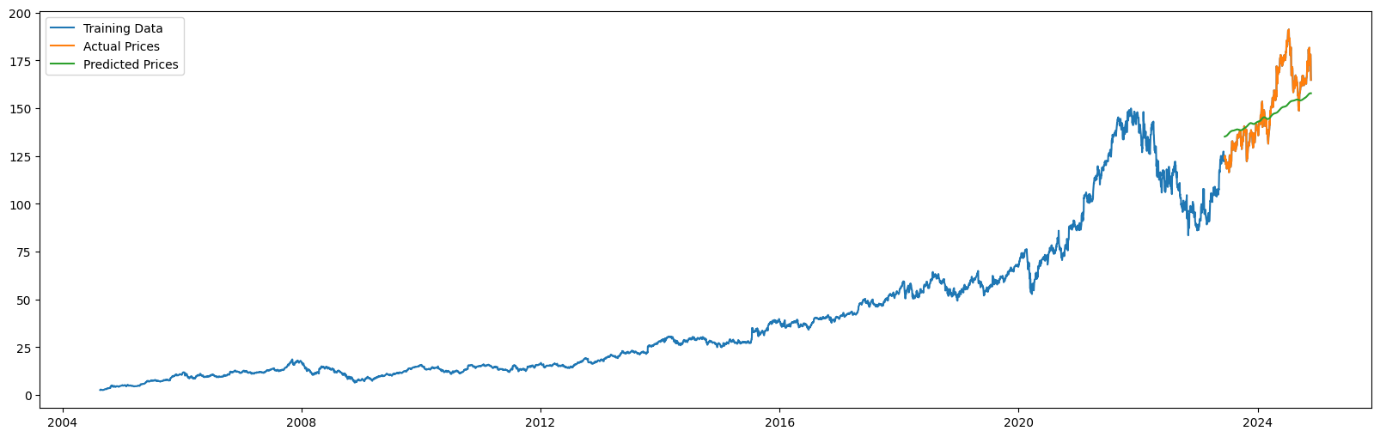
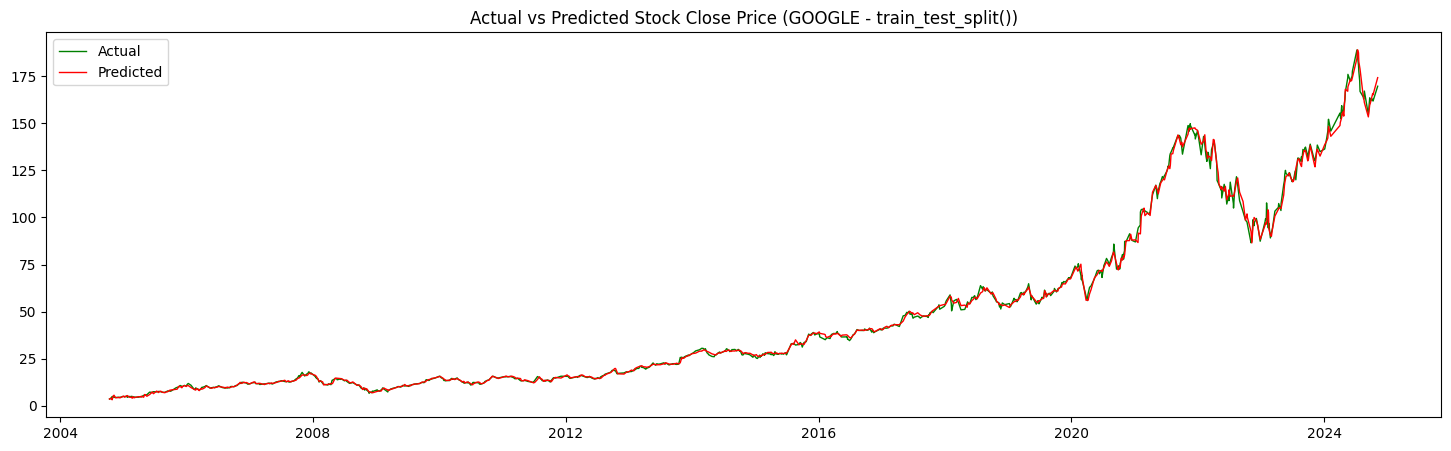
****

Figure 46: Result from FB Prophet

1. **CNN**

****

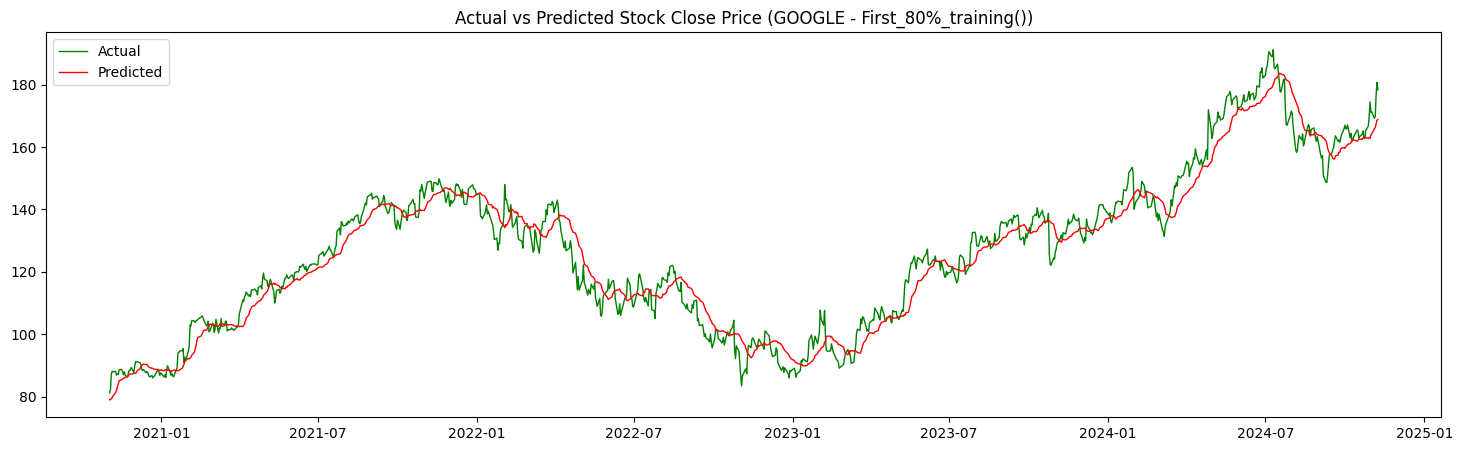
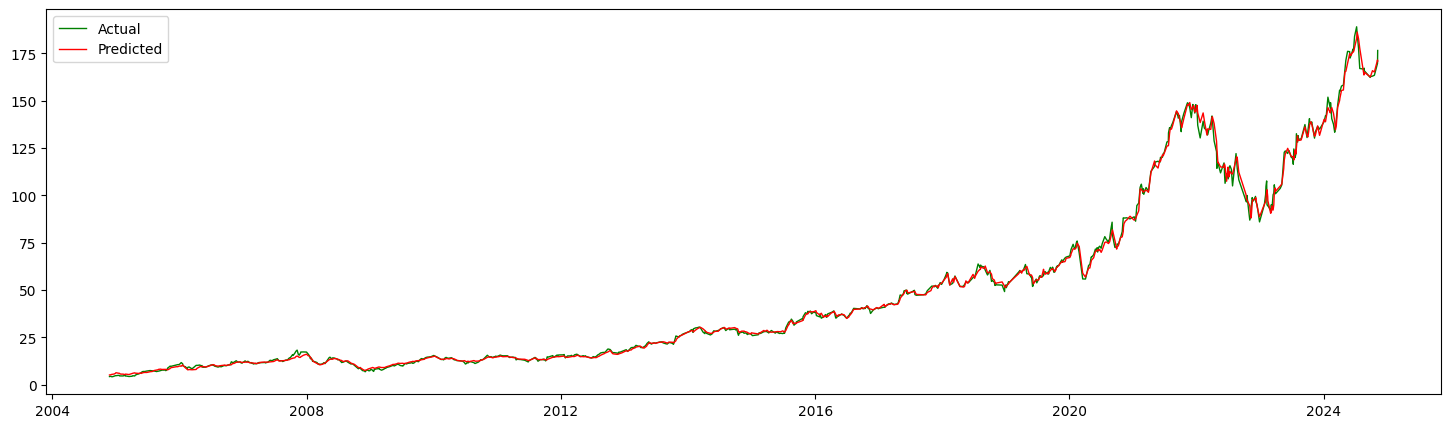
****

Figure 47: Result from CNN (train\_test\_split() vs first 80% train)

1. **LSTM**

****

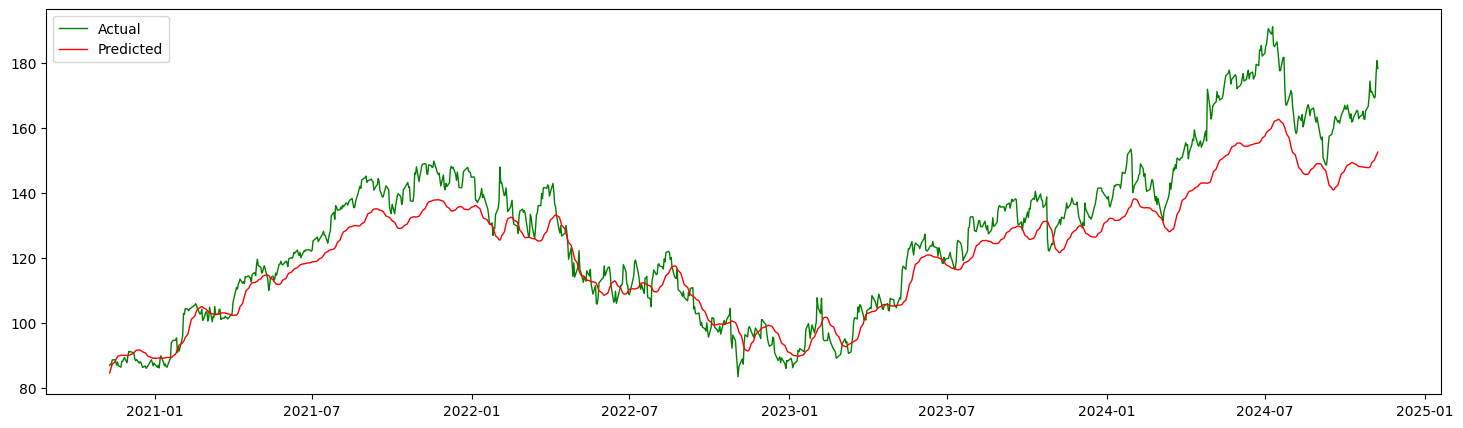
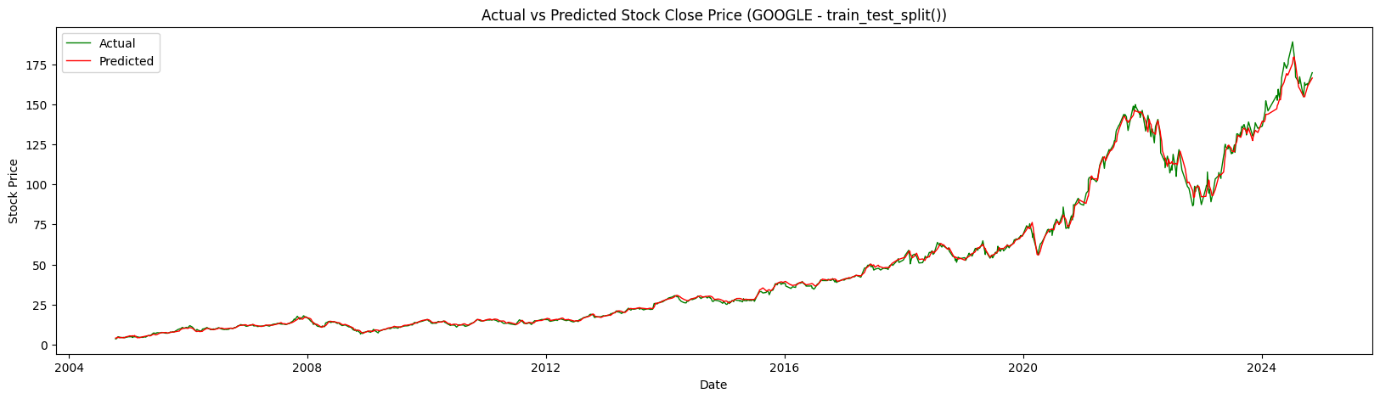
****

Figure 48: Result from LSTM (train\_test\_split() vs first 80% train)

1. **CNN-LSTM**

****

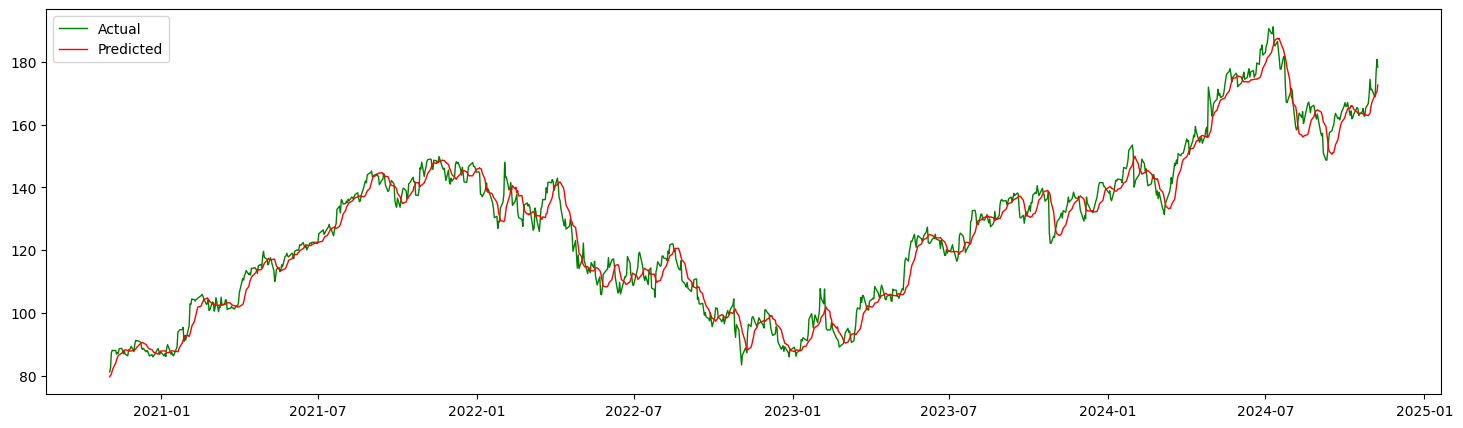
****

Figure 49: Result from CNN-LSTM (train\_test\_split() vs first 80% train)

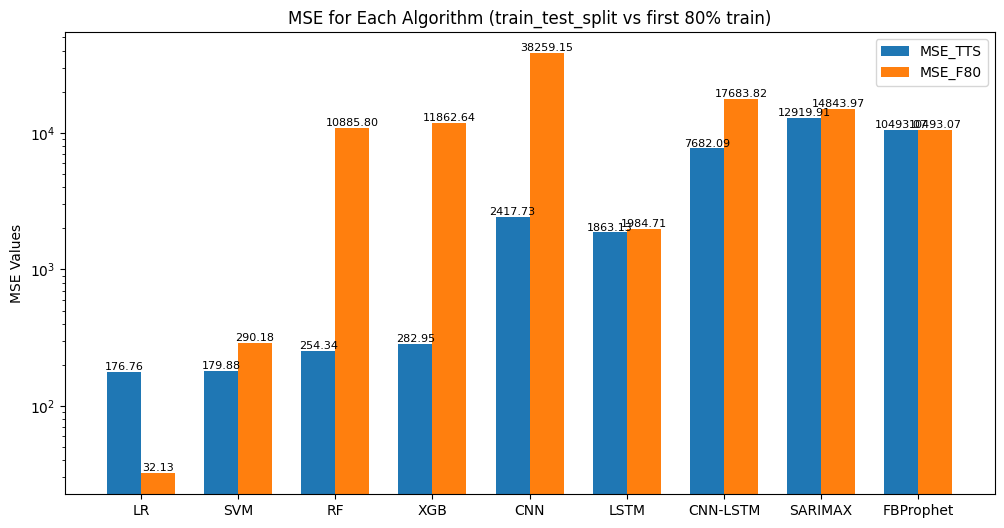
### **Result from experiment on Nabil Bank (NABIL) Stock Price Dataset**

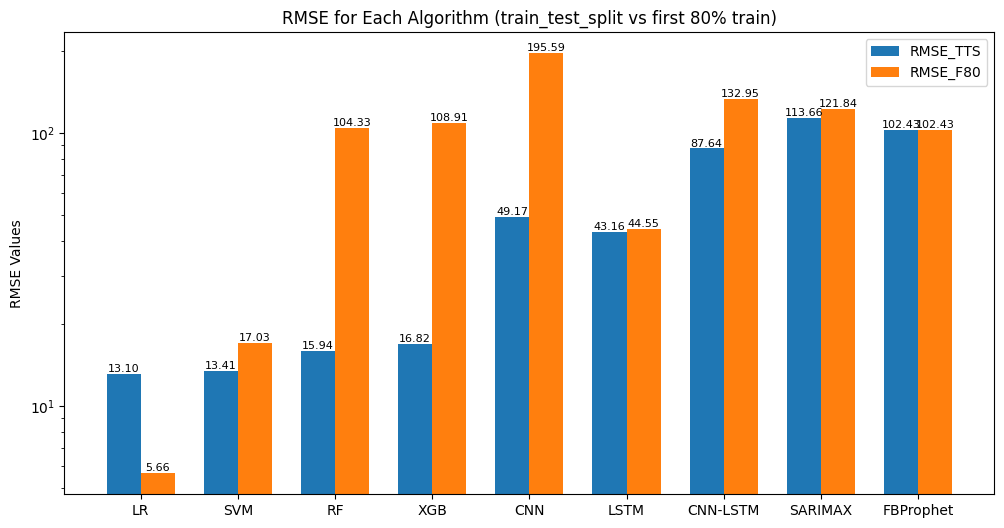
The following performance results are obtained from experiment performed on Nabil Bank (NABIL) stock price dataset with different machine learning, deep learning and time series forecasting.

Table 3: Performance Metrics of Different Algorithms in Nabil Bank Stock Price Dataset

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm | Mean Squared Error (MSE) | | Root Mean Squared Error (RMSE) | | Mean Absolute Error (MAE) | | Training Time (sec) |
|  | **Train Test Split** | **First 80% Train** | **Train Test Split** | **First 80% Train** | **Train Test Split** | **First 80% Train** | **Train Test Split** |
| LR | 176.76 | 32.13 | 13.10 | 5.66 | 8.76 | 4.17 | 0.031 |
| SVM | 179.88 | 290.18 | 13.411 | 17.03 | 8.89 | 12.22 | 1.001 |
| RF | 254.34 | 10885.80 | 15.94 | 104.33 | 9.84 | 71.79 | 2.255 |
| XGB | 282.95 | 11862.64 | 16.82 | 108.91 | 10.66 | 76.59 | 1.234 |
| CNN | 2417.73 | 38259.15 | 49.17 | 195.59 | 34.01 | 150.12 | 91.938 |
| LSTM | 1863.13 | 1984.71 | 43.16 | 44.55 | 24.66 | 39.87 | 966.430 |
| CNN-LSTM | 7682.09 | 17683.82 | 87.64 | 132.95 | 58.81 | 111.23 | 135.355 |
| SARIMAX | 12919.91 | 14843.97 | 113.66 | 121.84 | 99.69 | 104.39 | 93.043 |
| FB-Prophet | 10493.07 | | 102.43 | | 86.13 | | 6.603 |

The above tabular data is presented in below graphs where in each bar graphs, performance evaluation metrics MSE, RMSE and MAE of different machine learning models trained with training data from **train\_test\_split()** method and first 80% data are compared.





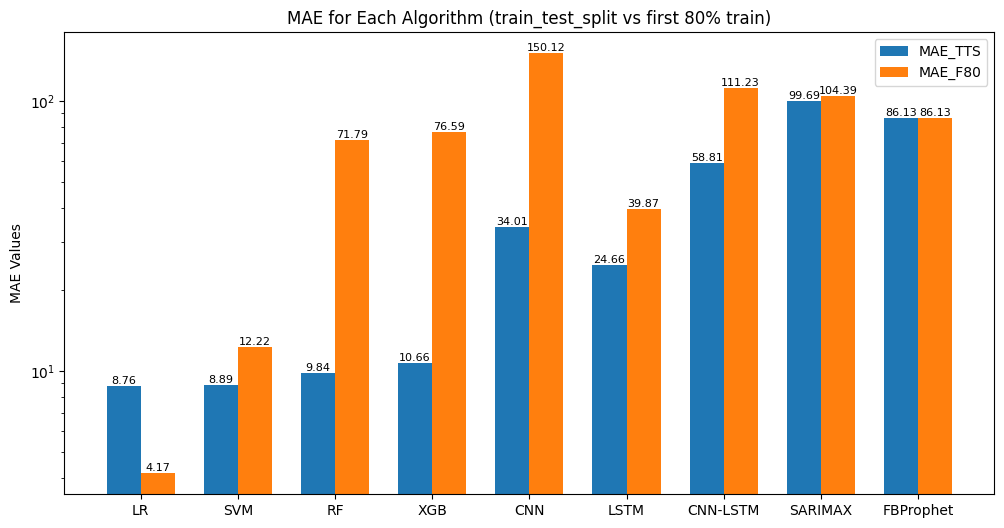


Figure 50: Performance Comparison of Different Model in Nabil Stock Price Data (train\_test\_split vs first 80% train)

Like in Google stock price prediction model, model trained with training data from **train\_test\_split()** method has outperformed model trained with first 80% data except Linear Regression model which has lower MSE, RMSE and MAE with the model trained with first 80% data.

But if we observe all the performance evaluation metrics of the different machine learning models trained with Nabil Bank stock price, the models are not performing well in this dataset. Although, basic machine learning algorithms have outperformed all other algorithms. In deep learning algorithms, LSTM performed best than other.

The machine learning models trained with Nabil Bank stock data have same kind of training time as the models trained with Google stock price data. LSTM took the longest time to train whereas Linear Regression took shortest time to train as it is the most basic algorithm among the nine algorithms.

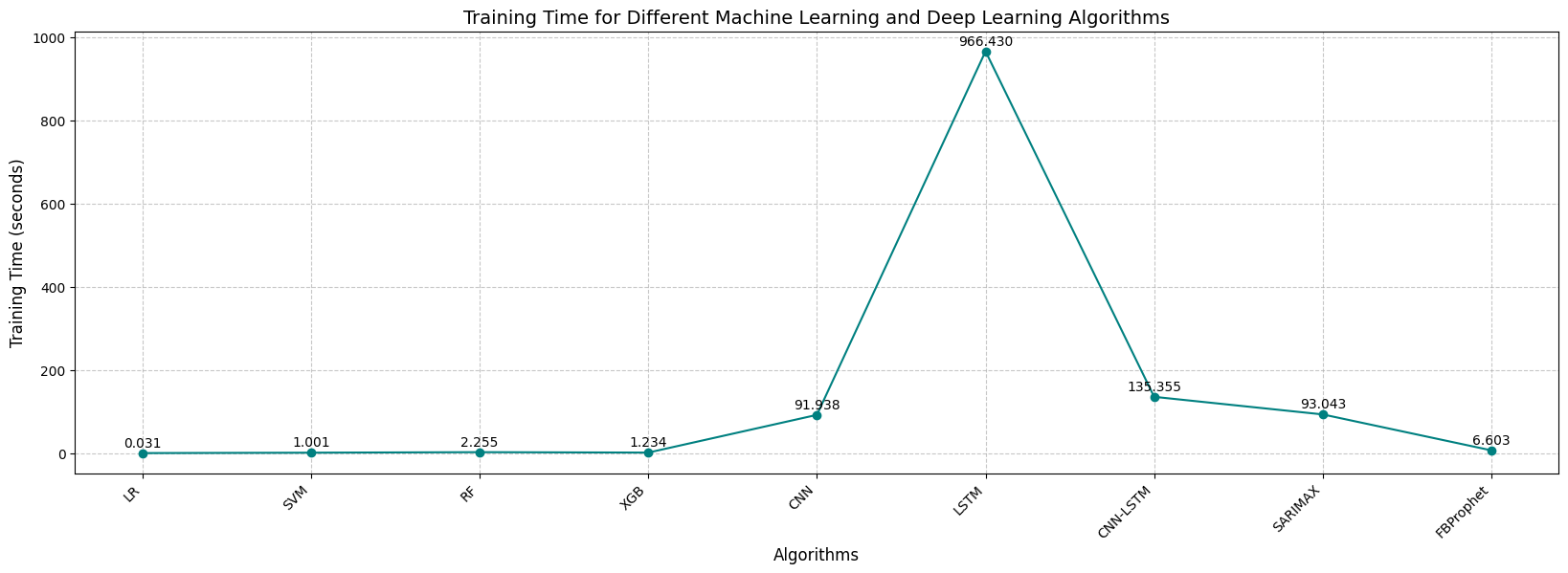


Figure 51: Training Time for Different Machine Learning Models in Nabil Bank Stock Price Dataset

The line graphs below show the Google’s Stock Actual Close Price vs Predicted Close Price obtained from different machine learning, deep learning and time series forecasting algorithms. For each algorithms, the first graph is obtained with the model trained with the training data from **train\_test\_split()** and the second graph is obtained from the model trained with first 80% data as a training dataset.

1. **Linear Regression**

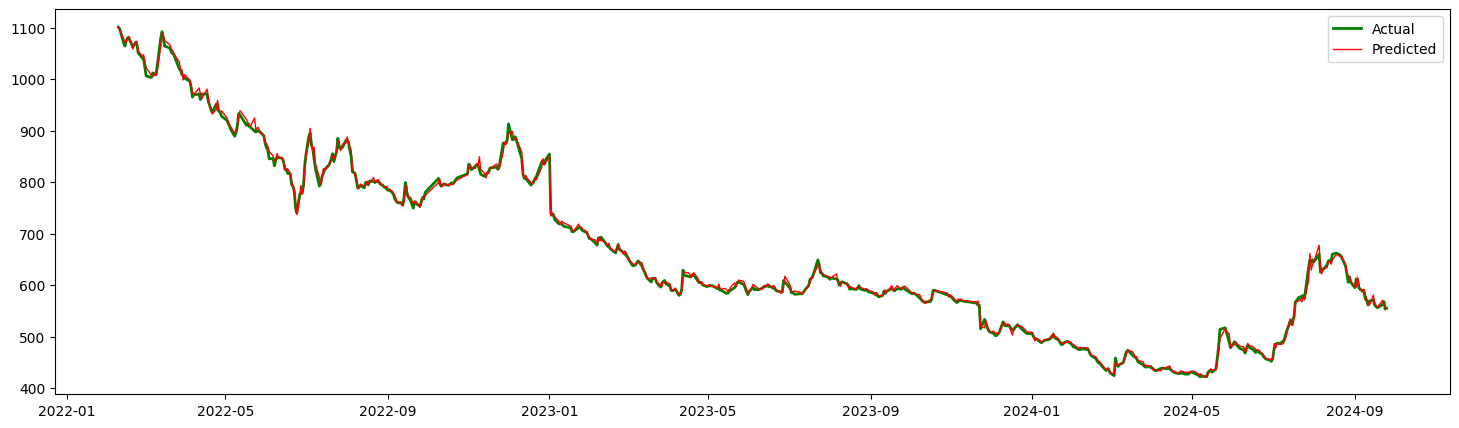
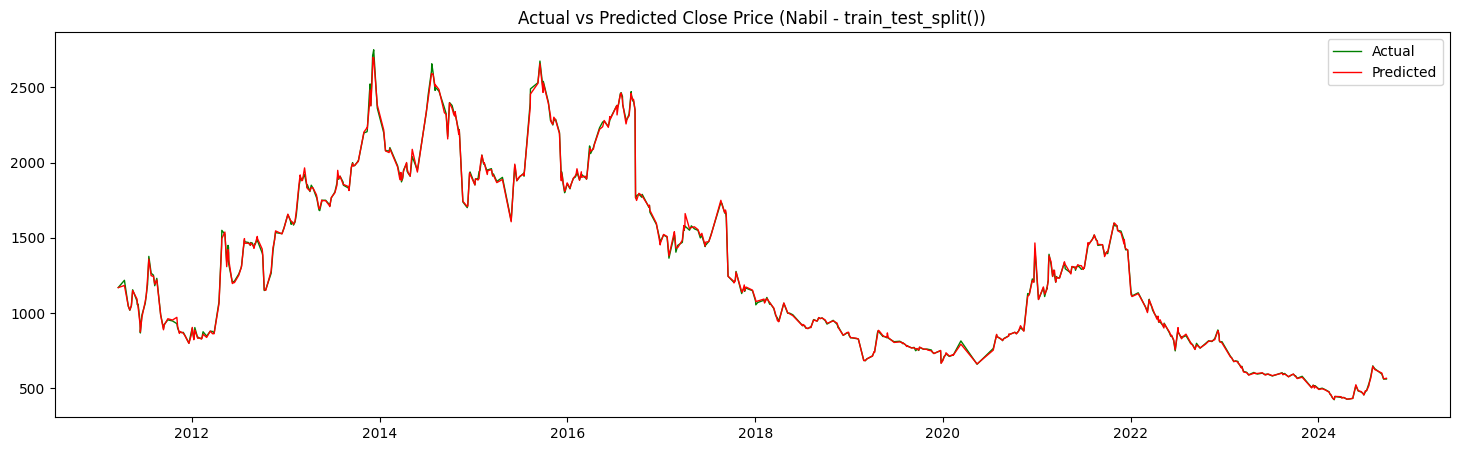
****

Figure 52: Result from Linear Regression (train\_test\_split() vs first 80% train)

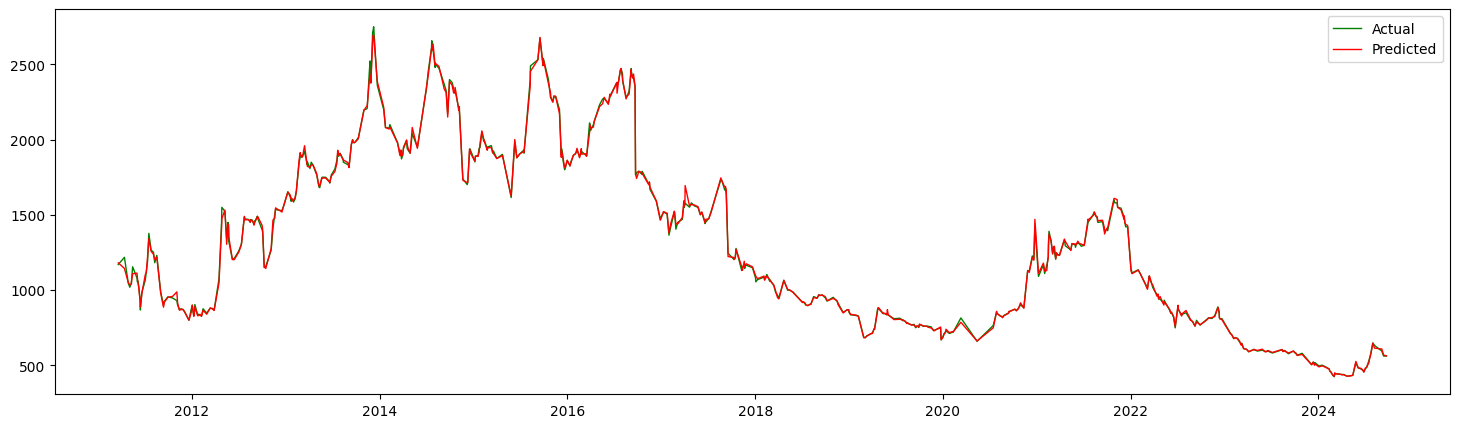
1. **Support Vector Machine**

****

****

Figure 53: Result from SVM (train\_test\_split() vs first 80% train)

1. **Random Forest**

****

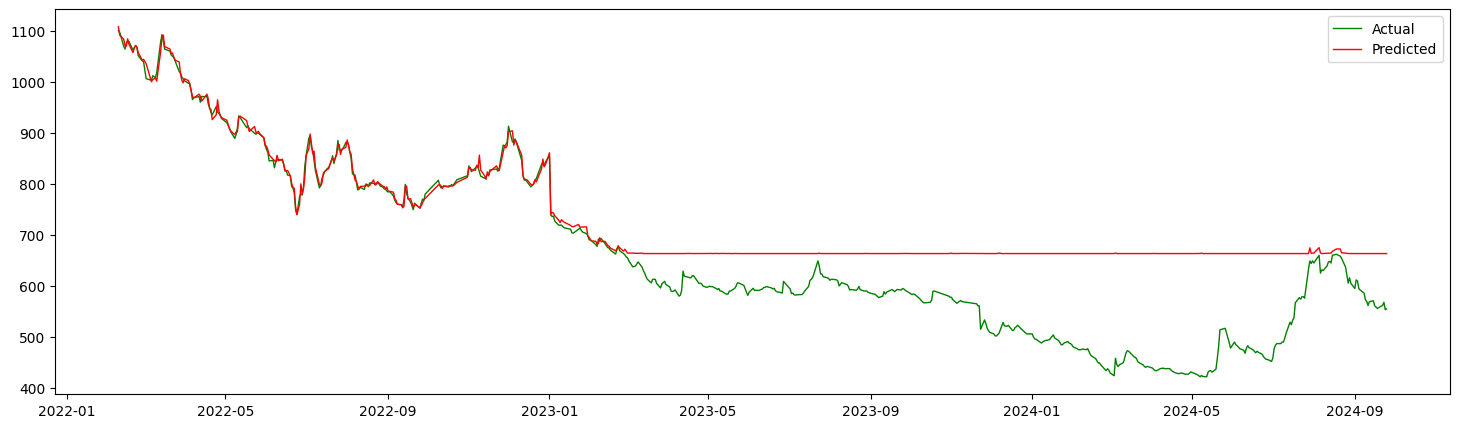
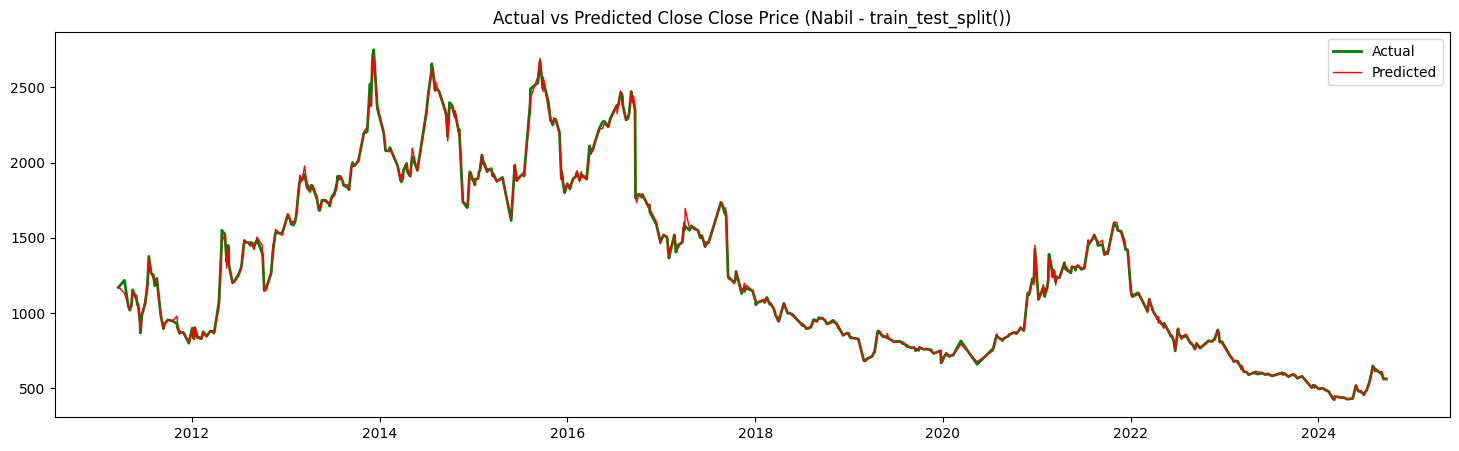
****

Figure 54: Result from Random Forest (train\_test\_split() vs first 80% train)

1. **XG Boost**

****

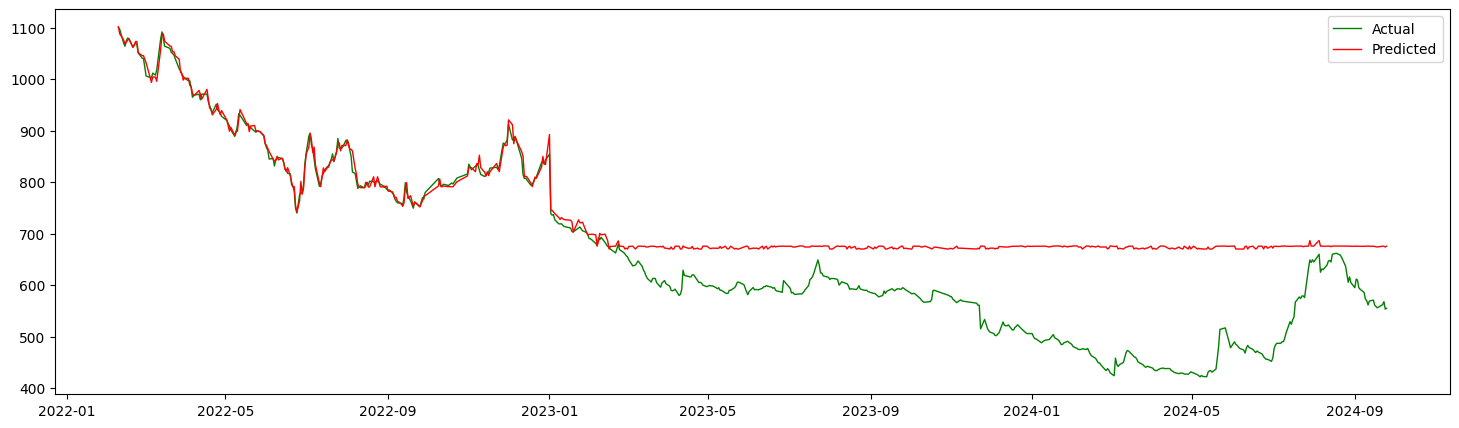
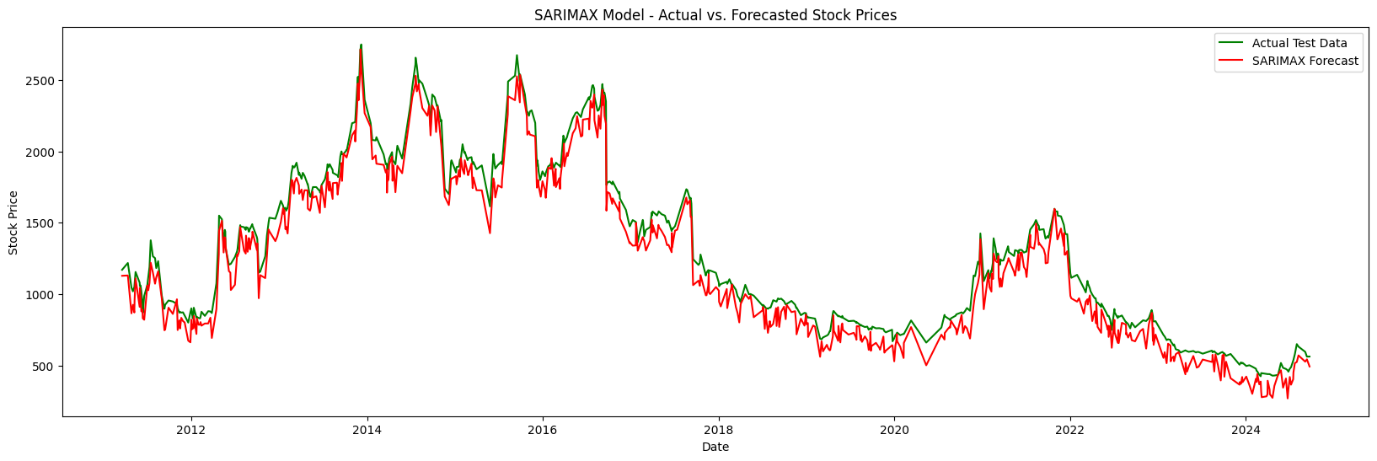
****

Figure 55: Result from XGBoost (train\_test\_split() vs first 80% train)

1. **SARIMAX**

****

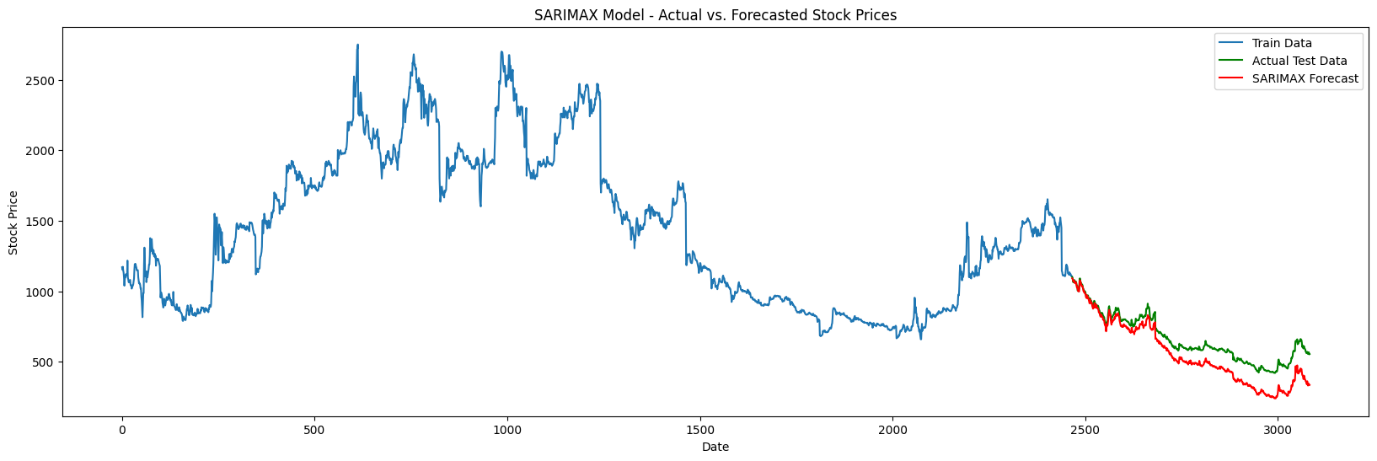
****

Figure 56: Result from SARIMAX (train\_test\_split() vs first 80% train)

1. **FB Prophet**

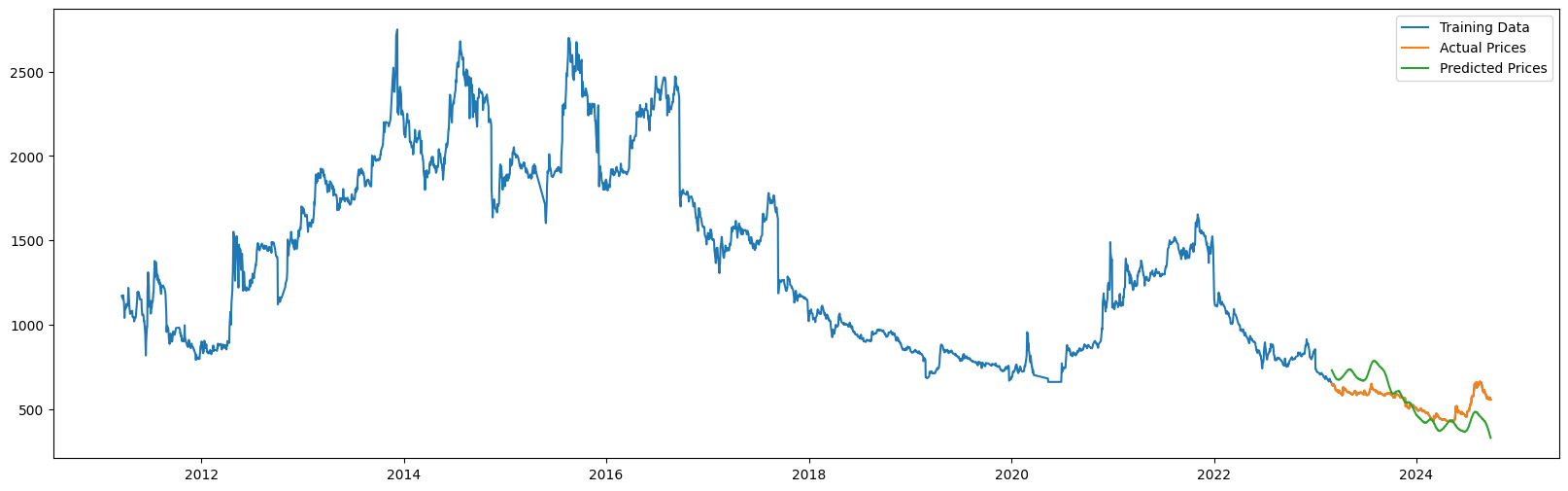
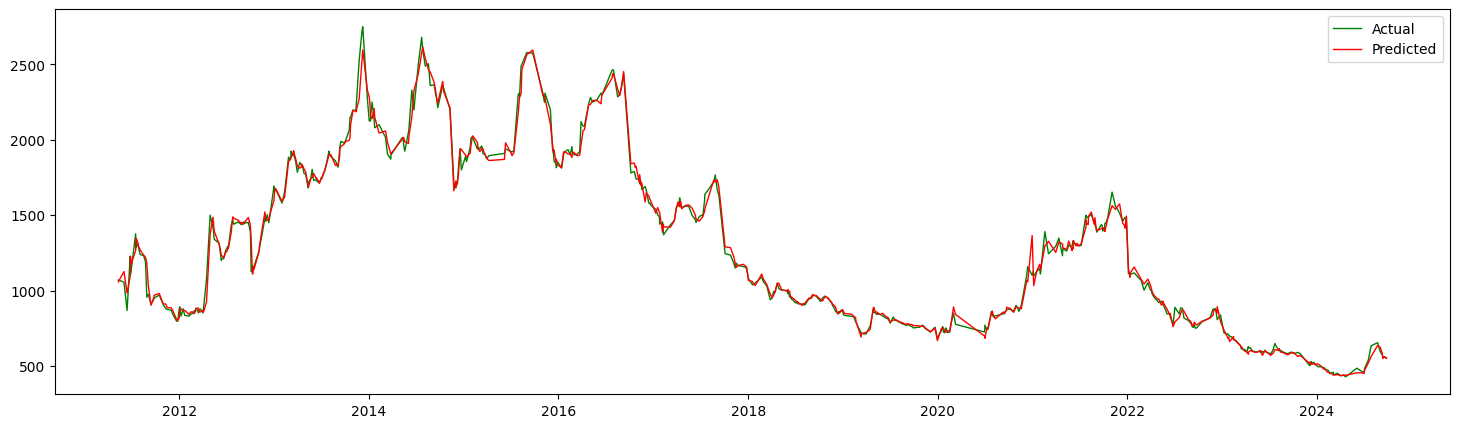
****

Figure 57: Result from FB Prophet

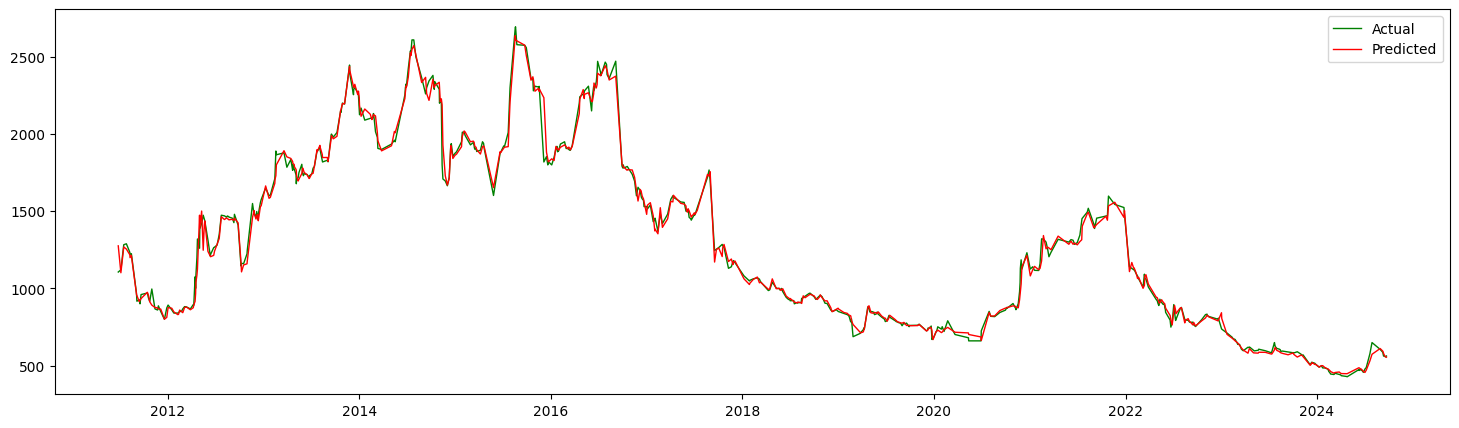
1. **CNN**

****

****

Figure 58: Result from CNN (train\_test\_split() vs first 80% train)

1. **LSTM**

****

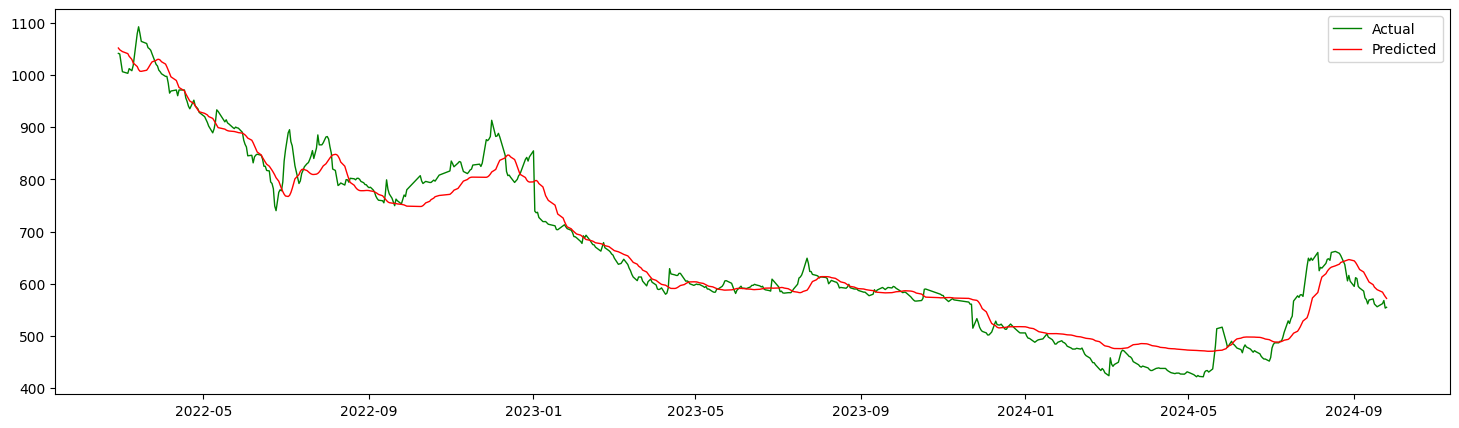
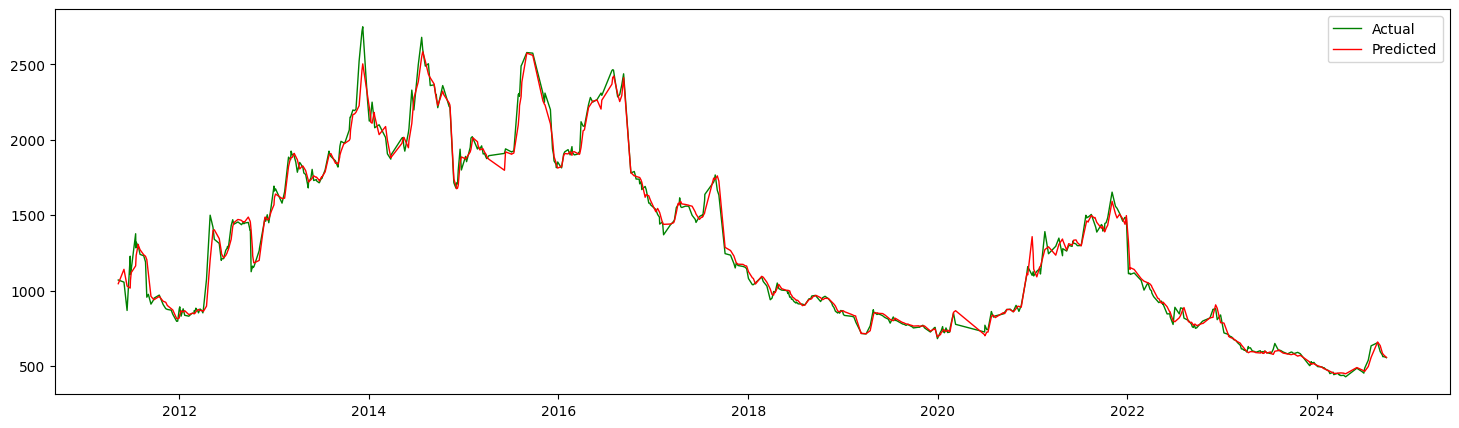
****

Figure 59: Result from LSTM (train\_test\_split() vs first 80% train)

1. **CNN-LSTM**

****

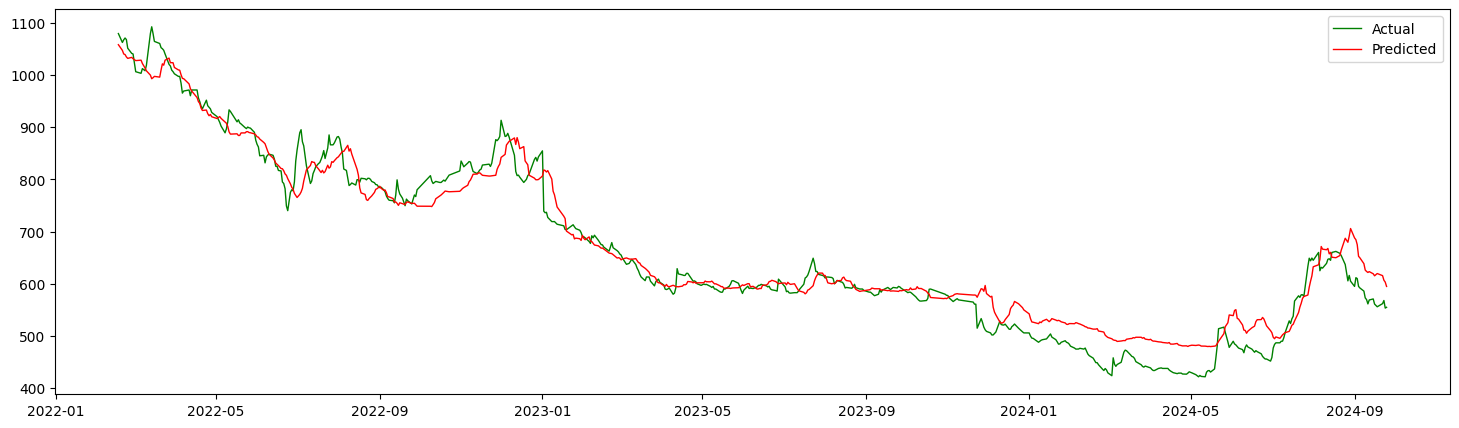
****

Figure 60: Result from CNN-LSTM (train\_test\_split() vs first 80% train)

## **Analysis**

The findings and the accompanying graph show that the research analysis underscores several key aspects regarding the efficacy of machine learning algorithms in predicting stock prices using two distinct datasets: Google stock prices, which reflect international markets, and Nabil Bank stock prices, sourced from Nepal. The outcomes provide intriguing insights into how both machine learning and deep learning algorithms function when utilized on diverse datasets and training approaches.

### **Performance of Different Models**

Fundamental machine learning methods, such as Linear Regression and Support Vector Machines, consistently outperformed more complex models like Random Forest, XGBoost, Convolutional Neural Networks, and Long Short-Term Memory networks. Both datasets showed that LR and SVM achieved lower values for Mean Squared Error, Root Mean Squared Error, and Mean Absolute Error, while also demonstrating faster training times. On the other hand, ensemble techniques (RF and XGB) and deep learning models (CNN, LSTM, and CNN-LSTM) encountered difficulties, particularly when trained with the first 80% of the dataset.

This disparity arises because simpler models like LR and SVM can effectively handle small, noisy datasets by capturing the basic linear or nonlinear relationships without overfitting. In contrast, deep learning models and ensemble methods are sensitive to data quality, requiring large amounts of well-structured and representative data to fully leverage their strengths. Moreover, the results indicate that ensemble learning methods, when trained with the first 80% of the data, fail to generalize effectively to the test set, likely due to their inability to capture temporal dependencies inherent in stock prices. These methods excel in datasets with richer features or structured patterns, which are not as evident in the Nabil Bank dataset.

### **Impact of Dataset Characteristics**

The fundamental differences in the characteristics of the Google and Nabil Bank datasets significantly influence model performance. The Google stock prices exhibit a more stable and upward trend over time, providing a clearer pattern for the models to learn. This stability likely allows simpler models to fit the data well without overfitting or underfitting. On the contrary, the Nabil Bank stock prices are highly volatile, with frequent and sharp fluctuations. This volatility introduces noise that confuses models, especially deep learning models, which rely on finding consistent temporal or spatial patterns. As a result, even advanced models struggle to perform well on this dataset, producing high error metrics.

### **Effect of Training Methodology**

The methodology used for training is crucial in influencing the performance of a model. The findings indicate that the train-test split method produced superior results compared to using the initial 80% of the dataset for training. This is especially apparent for deep learning models and ensemble techniques. Utilizing the train-test split method allows for random allocation of data, which likely provides a more representative sampling for both training and testing, thereby minimizing overfitting and enhancing the model's generalization capabilities. Conversely, training on the first 80% of the data adheres to a rigid chronological structure, which can cause models to have difficulty generalizing if the test dataset includes patterns or trends that weren't present in the training set.

The train-test split method performed particularly well with deep learning models like LSTM and CNN, which are designed to capture sequential dependencies. The temporal aspect of stock prices is better preserved when using this approach, enabling these models to learn effectively. However, training with the first 80% of the data exposed the limitations of ensemble learning methods (e.g., RF and XGB) as these models are not inherently designed to handle time-series data.

### **Why Deep Learning Underperformed**

While deep learning models such as LSTM are theoretically ideal for predicting time-series data due to their capability to capture temporal relationships, they underperformed in this analysis. Several reasons played a role in this result. Firstly, deep learning models typically need a large amount of training data to learn effectively, and the datasets utilized in this study might not have been sufficiently large to leverage their full potential. Secondly, these models tend to overfit, particularly when dealing with noisy or highly volatile data like Nabil Bank stock prices. This overfitting is apparent from the elevated error metrics observed in the CNN and LSTM models when they were applied to this dataset. Lastly, the intricate nature of these models requires meticulous tuning of hyperparameters and preprocessing; if not properly optimized, this can result in subpar performance.

### **Effect of Stock Price Trends**

The trends of the stock prices have a profound impact on model building and training. The Google dataset's steady upward trend provided a smoother and more predictable pattern for models to learn, allowing even simpler algorithms to perform well. In contrast, the erratic fluctuations in the Nabil Bank dataset added complexity, requiring models to balance fitting to the observed data while avoiding overfitting to noise. Deep learning models, which rely heavily on trend patterns, struggled with the inconsistencies in the Nabil Bank dataset, leading to significantly poorer performance compared to simpler algorithms.

## **Discussion**

The findings of this research provide several important insights into the effectiveness of machine learning algorithms for stock price prediction. One significant observation is that simpler approaches, including Linear Regression and Support Vector Machines, achieved better results than more complex methods such as Random Forest, XGBoost, and deep learning techniques (CNN, LSTM, CNN-LSTM) in both datasets analyzed. This result is unexpected, given the theoretical advantages associated with deep learning and ensemble techniques for challenging tasks like stock price forecasting. The better performance of LR and SVM can be linked to their straightforward nature and their capacity to generalize well in situations where the data is limited and noisy. In contrast, more sophisticated models are more likely to overfit smaller or highly volatile datasets, which negatively affected their predictive accuracy.

One of the key factors influencing the model performances is the difference in characteristics between the two datasets. The Google dataset exhibited a relatively smooth and upward trend, making it easier for models to learn and predict. Conversely, the Nabil Bank dataset displayed significant volatility and irregular fluctuations, introducing complexity that even advanced models struggled to handle. Deep learning models like LSTM and CNN, while designed to capture sequential dependencies, failed to perform effectively on the Nabil Bank dataset due to its noise and lack of consistent trends. Additionally, ensemble methods like RF and XGB showed poor performance when trained on the first 80% of data, likely because these methods are not inherently suited for time-series forecasting and could not effectively capture temporal dependencies.

Another important observation relates to the impact of the training methodology. The train-test split method consistently outperformed the first 80% training approach across most models, particularly for deep learning techniques. The randomness in the train-test split method provided a more balanced and representative sample of the dataset, enabling models to generalize better to unseen data. Training on the first 80% of the data, however, posed challenges for most models, as it exposed them to limited patterns and trends, reducing their ability to predict future fluctuations effectively. These findings underscore the importance of selecting a training methodology and model that align with the characteristics and challenges of the dataset. While advanced algorithms have the potential to excel in stock price prediction, they require extensive tuning, sufficient data, and a structured approach to fully leverage their capabilities.

# **Conclusion**

## **Summary of the Work**

### **Overview**

In the field of stock price forecasting, assessing and analyzing machine learning algorithms is vital for obtaining reliable and actionable outcomes. This research examined how effective various algorithms, such as Linear Regression, Support Vector Machines, Random Forest, XGBoost, Long Short-Term Memory, Convolutional Neural Networks, and hybrid models like CNN-LSTM, are at predicting stock prices using historical data. The findings indicated that simpler models, like LR and SVM, outperformed more intricate models, including RF, XGB, and deep learning techniques (CNN, LSTM), particularly on the volatile Nabil Bank dataset. These straightforward models exhibited superior generalization and required fewer computational resources, while more complex models faced issues with overfitting and struggled to effectively identify non-linear patterns in the dataset. Importantly, the train-test split method produced better performance compared to the first 80% training approach, especially for deep learning models, as it created a more representative sample for training purposes. While conventional machine learning techniques provided interpretability and efficiency, advanced models like LSTMs and CNNs demonstrated promise but necessitate larger datasets and meticulous tuning to prevent overfitting. This study underscores the significance of aligning model choice with the characteristics of the dataset and training methods, highlighting that simpler model such as LR and SVM can be more effective in situations involving noisy, limited data. Additionally, it points out that stock market prediction is inherently intricate and affected by variables beyond historical data, suggesting that subsequent research could investigate the incorporation of external elements, such as news sentiment and social media trends, to improve prediction precision.

### **Linkage to Objectives**

The objective of this research was to evaluate and analyze the effectiveness of various machine learning algorithms in predicting stock prices. The findings directly address this objective by comparing the performance of different models, including Linear Regression, Support Vector Machine, Random Forest, XGBoost, and deep learning models (LSTM, CNN, CNN-LSTM), using two datasets: Google stock prices and Nabil Bank stock prices. The study provided valuable insights into the predictive accuracy, reliability, and computational efficiency of these algorithms, successfully achieving the goal of assessing the performance of different machine learning models.

Additionally, the study assessed the effectiveness of machine learning models in comparison to traditional statistical methods, such as SARIMAX, highlighting the strengths and weaknesses of each approach. The findings indicated that simpler models like Linear Regression and Support Vector Machine outperformed more complex algorithms, such as deep learning, in certain scenarios, thereby fulfilling the goal of comparing machine learning techniques with traditional statistical methods. Furthermore, the research identified the most effective machine learning model for predicting stock prices, with LR and SVM demonstrating better generalization, especially on datasets with high volatility. This outcome achieved the goal of determining the most efficient machine learning model for stock price forecasting.

Finally, the study examined the effect of different data training strategies, such as the train-test split method versus using the first 80% of the data for training, and how these strategies influenced model performance. The research concluded that the train-test split method provided more balanced and representative training data, which led to better generalization and predictive accuracy, thus fulfilling the objective of identifying the effect of data used for training and testing machine learning models.

## **Reflection**

This study has delivered important insights regarding the use of machine learning algorithms for predicting stock prices, outperforming conventional methods in terms of adaptability and effectiveness in fluctuating market conditions. The study emphasized the importance of strong evaluation frameworks and high-quality data for assessing the dependability of predictive models. Although LSTM networks proved to be a promising method, it is crucial to recognize the complex and uncertain nature of financial markets. The precision of predictions can be greatly affected by elements such as economic indicators, geopolitical situations, and investor sentiment, which may not be adequately represented by historical data alone.

A constraint of this research is the dependence on historical records, which might not sufficiently account for unexpected occurrences or abrupt changes in the market. Subsequent investigations could examine the merging of real-time news sentiment evaluation and social media information to enhance predictive precision. Furthermore, utilizing more advanced deep learning methods, including attention mechanisms and transformer architectures, could significantly improve the effectiveness of predictive models. It's crucial to emphasize that, although machine learning provides powerful tools for forecasting stock prices, it ought not to be the sole basis for making investment choices. A cautious strategy involves integrating quantitative analysis with qualitative insights derived from fundamental and technical assessments. By grasping the fundamental factors influencing stock price fluctuations and recognizing the constraints of machine learning models, investors can arrive at better-informed choices.

## **Future Work**

### **Research Limitations**

Although research has advanced considerably in investigating machine learning algorithms for predicting stock prices, it is crucial to recognize various limitations that can influence the models' accuracy and reliability:

1. **Market Volatility and Uncertainty:**

* **Unpredictable Events:** Sudden market shocks, economic crises, or geopolitical events can significantly impact stock prices, making accurate predictions challenging.
* **Time-Varying Patterns:** Stock market trends and patterns can change over time, requiring continuous model adaptation.

1. **Model Complexity and Overfitting**

* **Overfitting Risk:** Complex models with numerous parameters are more susceptible to overfitting the training data, which results in poor generalization performance on unseen data.
* **Computational Cost:** Advanced deep learning models can be resource-intensive to train and deploy, requiring significant computational power and time.

1. **Ethical Considerations and Responsible AI:**

* **Fairness and Bias:** Models may unintentionally reinforce biases embedded in the training data, resulting in biased or unfair outcomes.
* **Transparency and Interpretability:** Black-box models can be difficult to interpret, making it challenging to understand their decision-making process.

1. **Real-world Application and Practical Constraints:**

* **Transaction Costs and Market Impact:** Real-world trading involves transaction costs, slippage, and market impact, which can affect the profitability of trading strategies.
* **Regulatory and Compliance:** Adhering to regulatory requirements and compliance standards can impose additional constraints on investment strategies.

### **Recommendations for Future Research and Improvements**

1. **Incorporating External Factors:**

* **News Sentiment Analysis:** Analyze news articles and social media sentiment to capture market sentiment and its impact on stock prices.
* **Economic Indicators:** Incorporate macroeconomic indicators like GDP growth, interest rates, and inflation rates to enhance predictive models.

1. **Advanced Deep Learning Techniques:**

* **Attention Mechanisms:** Utilize attention mechanisms to focus on important parts of the input sequence, improving the model's ability to capture long-term dependencies.
* **Transformer Models:** Investigate the application of transformer models, known for their success in tasks related to natural language processing and computer vision, in identifying intricate patterns within time series data.

1. **Hybrid Models:**

* **Combining Technical and Fundamental Analysis:** Integrate technical indicators (such as moving averages and RSI) with fundamental aspects (like earnings and dividends) to develop hybrid models.

1. **Real-time Prediction and Adaption:**

* **Online Learning:** Develop models that can adjust to evolving market conditions by continuously updating and learning from new data.
* **Real-time Data Streams:** Process real-time data streams to make timely predictions and adjust strategies accordingly.

# **References**

A, A., R, R., S, V. R. & Bagde, A. M., 2023. Predicting Stock Market Time-Series Data using CNN-LSTM Neural Network Model. *TechRxiv.*

Agarwal, N. B. & Yadav, D. K., 2024. A Comprehensive Analysis of Cassical Machine Learning and Modern Deep Learning Methodologies. *International Journal of Engineering Research & Technology (IJERT),* 13(5).

Alakh, 2024. *Support Vector Regression Tutorial for Machine Learning, Analytics Vidhya.* [Online]   
Available at: https://analyticsvidhya.com/blog/2020/03/support-vector-regression-tutorial-for-machine-learning/  
[Accessed 2 October 2024].

Ankit, 2022. *Generate Quick and Accurate Time Series Forecasts using Facebook’s Prophet (with Python & R codes), Analytics Vidhya.* [Online]   
Available at: https://www.analyticsvidhya.com/blog/2018/05/generate-accurate-forecasts-facebook-prophet-python-r/  
[Accessed 2 October 2024].

Baheshti, N., 2022. *Random Forest Regression: A Basic Explanation and Use Case in 7 Minutes, Medium.* [Online]   
Available at: https://towardsdatascience.com/random-forest-regression-5f605132d19d  
[Accessed 2 October 2024].

Bajracharya, B. B., 2019. *Nepal's Safest Stock in a Volatile Share Market.* [Online]   
Available at: https://www.smartfamily.com.np/finance/nepals-safest-stock-in-a-volatile-share-market  
[Accessed 20 July 2027].

Biswal, A., 2024. *Stock Market Prediction using Machine Learning in 2024, Simplilearn.* [Online]   
Available at: https://www.simplilearn.com/tutorials/machine-learning-tutorial/stock-price-prediction-using-machine-learning  
[Accessed 23 September 2024].

Box, G. E. P., Jenkins, G. M., Reinsel, G. C. & Ljung, G. M., 2015. *Time Series Analysis: Forecasting and Control.* 5th ed. New York: Wiley.

Brownlee, J., 2021. *A Gentle Introduction to Ensemble Learning Algorithms, Machine Learning Mastery.* [Online]   
Available at: https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/  
[Accessed 1 October 2024].

Calzone, O., 2022. *An Intuitive Explanation of LSTM.* [Online]   
Available at: https://medium.com/@ottaviocalzone/an-intuitive-explanation-of-lstm-a035eb6ab42c  
[Accessed 3 October 2024].

Chen, L.-P., 2020. Using Machine Learning Algorithms on Prediction of Stock Price. *Journal of Modeling and Optimization,* 12(2), pp. 84-99.

Codecademy Team, 2024. *What is Scikit-Learn?.* [Online]   
Available at: https://www.codecademy.com/article/scikit-learn  
[Accessed 24 September 2024].

Crabtree, M., 2023. *What is Machine Learning? Definition, Types, Tools & More.* [Online]   
Available at: https://www.datacamp.com/blog/what-is-machine-learning?dc\_referrer=https%3A%2F%2Fwww.google.com%2F  
[Accessed 25 July 2024].

Fama, E. F., 1970. Effective Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance,* 25(2), pp. 383-417.

Fernando, J., 2024. *Moving Average (MA): Purpose, Uses, Formula, and Examples.* [Online]   
Available at: https://www.investopedia.com/terms/m/movingaverage.asp  
[Accessed 1 October 2024].

GeeksforGeeks, 2024. *Introduction to Matplotlib.* [Online]   
Available at: https://www.geeksforgeeks.org/python-introduction-matplotlib/  
[Accessed 24 September 2024].

Graham, B. & Dodd, D., 1934. *Security Analysis.* 1st ed. s.l.:McGraw-Hill Book Co..

Gunduz, H., Yaslan, Y. & Cataltepe, Z., 2017. Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations. *Knowledge-Based Systems,* Volume 137, pp. 138-148.

Guresen, E., Kayakutlu, G. & Daim, T. U., 2011. Using artificial neural network models in stock market index prediction. *Expert Systems with Applications,* 38(8), pp. 10389-10397.

Hayes, A., 2024. *Autoregressive Integrated Moving Average (ARIMA) Prediction Model.* [Online]   
Available at: https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp  
[Accessed 2 October 2024].

Hopman, M., 2021. *Stock and Crypto Market Values, Unsplash.* [Online]   
Available at: https://unsplash.com/photos/red-and-blue-light-streaks-fiXLQXAhCfk  
[Accessed 2 October 2024].

Huang, Y., Capretz, L. F. & Ho, D., 2021. Machine Learning for Stock Prediction Based on Fundamental Analysis. *IEEE Symposium Series on Computational Intelligenc,* pp. 01-10.

Jain, A., 2024. *A Comprehensive Guide to Performance Metrics in Machine Learning.* [Online]   
Available at: https://medium.com/@abhishekjainindore24/a-comprehensive-guide-to-performance-metrics-in-machine-learning-4ae5bd8208ce#:~:text=Performance%20metrics%20play%20a%20crucial,its%20performance%20across%20various%20tasks.  
[Accessed 6 November 2024].

Jordan, J., 2017. *Hyperparameter tuning for machine learning models..* [Online]   
Available at: https://www.jeremyjordan.me/hyperparameter-tuning/  
[Accessed 2 October 2024].

Kabir, M. H., Sobur, A. & Amin, M. R., 2023. Stock Price Prediction Using The Machine Learning. *International Journal of Creative Research Thoughts (IJCRT),* 11(7), pp. 946-950.

Kavita, 2024. *Linear Regression: A Comprehensive Guide, Analytics Vidhya.* [Online]   
Available at: https://www.analyticsvidhya.com/blog/2021/10/everything-you-need-to-know-about-linear-regression/  
[Accessed 1 October 2024].

Kenton, W., 2024. *Excess Kurtosis: Definition, Types, Example: Investopedia.* [Online]   
Available at: https://www.investopedia.com/terms/e/excesskurtosis.asp  
[Accessed 25 July 2024].

Khanna, M. et al., 2022. Performance Evaluation of Machine Learning Algorithms for Stock Price and Stock Index Movement Prediction Using Trend Deterministic Data Prediction. *International Journal of Applied Metaheuristic Computing,* 13(1), pp. 1-30.

K, P. et al., 2021. Comparative Study: Stock Price Prediction Using Fundamental and Technical Analysis. *2021 IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC),* pp. 1-4.

Kumar, A., 2021. *Stock Price Prediction using Machine Learning Techniques, Analytics Yogi.* [Online]   
Available at: https://vitalflux.com/popular-machine-learning-techniques-for-stock-price-movement-prediction/  
[Accessed 23 September 2024].

Kumar, R., 2022. *What is Numpy and How it works? An Overview and Its Use Cases? - DevOps School.* [Online]   
Available at: https://www.devopsschool.com/blog/what-is-numpy-and-how-it-works-an-overview-and-its-use-cases-2/  
[Accessed 24 September 2024].

Kumbure, M. M., Lohrmann, C., Luukka, P. & Porras, J., 2022. Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications,* 197(2022), pp. 1-41.

Lawrence, S. J., 2023. *What is LSTM? - Introduction to Long Short-Term Memory.* [Online]   
Available at: https://www.scaler.com/topics/deep-learning/lstm/  
[Accessed 03 October 2024].

Matsuo, K., 2022. *Selenium Automation Testing - A Complete Guide for 2024, headspin.* [Online]   
Available at: https://www.headspin.io/blog/selenium-testing-a-complete-guide  
[Accessed 26 September 2024].

Mehtab, S. & Sen, J., 2019. A Robust Predictive Model for Stock Price Prediction Using Deep Learning and Natural Language Processing. *Proceedings of the 2019 International Conference on Business Analytics and Intelligence (ICBAI 2019),* 1 December.

Melanie, 2024. *Facebook Prophet : All you need to know, DataScientest.* [Online]   
Available at: https://datascientest.com/en/facebook-prophet-all-you-need-to-know  
[Accessed 2 October 2024].

Olumide, S., 2023. *Root Mean Square Error (RMSE) In AI: What You Need To Know, Arize.* [Online]   
Available at: https://arize.com/blog-course/root-mean-square-error-rmse-what-you-need-to-know/  
[Accessed 1 October 2024].

Otten, N. V., 2024. *Support Vector Regression (SVR) Simplified & How To Tutorial In Python, Spot Intelligence.* [Online]   
Available at: https://spotintelligence.com/2024/05/08/support-vector-regression-svr/  
[Accessed 2 October 2024].

Pathak, A. & Pathak, S., 2020. Study of Machine Learning Algorithms for Stock Market Prediction. *International Journal of Engineering Research & Technology (IJERT),* 9(06), pp. 295-300.

Practicus AI, 2019. *Understanding the 3 most common loss functions for Machine Learning Regression, Medium.* [Online]   
Available at: https://towardsdatascience.com/understanding-the-3-most-common-loss-functions-for-machine-learning-regression-23e0ef3e14d3  
[Accessed 01 October 2024].

Qayyum, R., 2022. *Computational Complexity of Machine Learning Algorithms, Medium.* [Online]   
Available at: https://rafayqayyum.medium.com/computational-complexity-of-machine-learning-algorithms-254c275de84  
[Accessed 1 October 2024].

Raju, S. S. et al., 2023. A Three-Dimesional Approach for Stock Prediction Using AI/ML Algorithms: A Review & Comparision. *2023 4th International Conference on Innovative Trends in Information Technology (ICITIIT),* pp. 1-6.

Rizvi, D. R. & Khalid, M., 2024. Performance Analysis of Stocks Using Deep Learning Models. *5th International Conference on Innovative Data Communication Technologies and Application - ICIDCA,* 233(2024), pp. 753-762.

Ronaghan, S., 2019. *Data Understanding for Machine Learning: Assessment & Exploration, Medium.* [Online]   
Available at: https://towardsdatascience.com/data-understanding-for-machine-learning-assessment-exploration-aca1aadc1cb6  
[Accessed 24 September 2024].

Saadeddin, Z., 2024. *ARIMA for Time Series Forecasting: A Complete Guide, DataCamp.* [Online]   
Available at: https://www.datacamp.com/tutorial/arima  
[Accessed 2 October 2024].

Saud, A. S. & Shankya, S., 2019. Analysis of Gradient Descent Optimization Techniques with Gated Recurrent Unit for Stock Price Prediction: A Case Study on Banking Sector of Nepal Stock Exchange. *Journal of Institute of Science and Technology,* 24(2), pp. 17-21.

Sen, J., Mehtab, S. & Dutta, A., 2021. Stock Price Prediction Using Machine Learning and LSTM-Based Deep Learning Models. *TechRxiv (Powered by: IEEE).*

Shrestha, P., 2021. *Application of Machine Learning and Deep Learning Techniques for Nepal Stock Market Prediction.* [Online]   
Available at: https://www.researchgate.net/publication/353634645\_Application\_of\_Machine\_Learning\_and\_Deep\_Learning\_Techniques\_for\_Nepal\_stock\_market\_price\_prediction  
[Accessed 20 July 2024].

Sonkavde, G. et al., 2023. Forecasting Stock Market Prices Using Machine Learning and Deep Learning Models: A Systemic Review, Performance Analysis and Discussion of Implications. *International Journal of Financial Studies,* 11(3), p. 22.

Statista, 2024. *Total market capitalizations of companies listed on stock exchanges worldwide from 2013 to 2023.* [Online]   
Available at: https://www.statista.com/statistics/274490/global-value-of-share-holdings-since-2000/  
[Accessed 15 August 2024].

Taylor, S., 2024. *Skewness, Corporate Finance Institute.* [Online]   
Available at: https://corporatefinanceinstitute.com/resources/data-science/skewness/#:~:text=However%2C%20skewed%20data%20will%20increase,large%20losses%20on%20the%20investment.  
[Accessed 25 July 2024].

Tretina, K., 2024. *What Is The Stock Market? How Does It Work?.* [Online]   
Available at: https://www.forbes.com/advisor/in/investing/what-is-stock-market/  
[Accessed 24 July 2024].

Uddin, M. R., 2023. *COVID-19 Pandemic Data Analysis and Prediction Using Machine-Learning Algorithms, Research Gate.* [Online]   
Available at: https://www.researchgate.net/publication/374010617\_COVID-19\_Pandemic\_Data\_Analysis\_and\_Prediction\_Using\_Machine-Learning\_Algorithms  
[Accessed 01 October 2024].

Vapnik, V. N., 1999. *The Nature of Statistical Learning Theory.* s.l.:Springer.

Willems, K., 2022. *Pandas Tutorial: DataFrames in Python, datacamp.* [Online]   
Available at: https://www.datacamp.com/tutorial/pandas-tutorial-dataframe-python  
[Accessed 24 September 2024].