University of Essex

School of Mathematics, Statistics
and Actuarial Science

MA981 DISSERTATION

Retail Sales Time Series Forecasting using Statistical Learning, Machine Learning and Deep Learning

Nagadithya Bathala
2310244

Supervisor: Gao Tao

September 17, 2024

Colchester

Abstract

This dissertation study aimed to present a comprehensive analysis and implementation of time series forecasting using advanced time series analysis to predict future sales of Corporacion Favorita, a large Ecuadorian-based grocery retailer. Stores in the Retail Industry must plan their organizational environment carefully to optimize the costs and maximize the revenue. Accurate sales forecasting is crucial in the retail environment as it helps in inventory management, revenue optimization, resource allocation, financial planning, and customer satisfaction. This study begins with exploratory data analysis (EDA) to understand the underlying patterns in sales data such as seasonality, cycles, trends, and the impact of external factors (holidays). Feature engineering was carried out to extract the key features in the data that help to improve the model performance.

This research focuses on developing and comparing various time series forecasting models including traditional time series models such as Autoregressive Integrated Moving Average (ARIMA), Seasonal Autoregressive Integrated Moving Average (SARIMA) along with machine learning algorithms such as Random Forest (RF), Extreme Gradient Boosting Machine (XGB), Support Vector Machine (SVM), Facebook Prophet (Fb-Prophet) and deep learning algorithm Long Short-Term Memory (LSTM) networks. A recursive time series forecasting technique is used to generate multi-step forecasts which helps to predict sales over extended periods. This study also addresses the impact of seasonality, trends, and external factors on sales prediction. The proposed model's performance was evaluated using evaluation metrics such as Root Mean Squared Error (RMSE), coefficient of determination R-squared (R2), and Mean Absolute Percentage Error (MAPE). The results show the ability of implemented models to capture complex patterns in sales data and the importance of feature engineering and external factors in enhancing model performance that helps superstores optimize operations and decision-making processes. Among the models used to forecast future sales, XGB and SVM outperformed other models with low RMSE error and high R2 score highlighting the effectiveness of both models in capturing complex sales patterns.

Acknowledgments

I would like to express my sincere gratitude to everyone who supported me throughout this research project. First, I would like to thank my supervisor, Tao Gao, for his valuable guidance and insightful feedback which were crucial in shaping this dissertation. I would also like to extend my thanks to the University of Essex, whose resources contributed to my work. Special thanks to my friends and family members for their continued support throughout this master's journey. Finally, I would like to acknowledge the Kaggle platform for providing access to the dataset used in this research study and all the authors whose work helped me in understanding the various concepts related to this dissertation.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

The retail industry is one of the fastest-growing industries in the world and its contribution to the global economy is huge [1]. It includes small-scale independent stores to large multinational superstore chains operating in different formats such as physical and online stores. The retail industry also serves as a connection between manufacturers and wholesalers that helps the products reach end customers. Ongoing unstable economic conditions and changes in customer expectations shape the retail sector's future. Companies that can adapt to new technologies and customer needs are likely to sustain in this industry.

Retail superstores are large-scale stores that offer a wide variety of products all in one place which includes groceries, household items, electronics, and clothing. These stores are designed in such a way that they will provide a one-stop experience to customers. They attract a wide range of customers with a vast variety of products available at competitive and reasonable prices. However, the retail industry is highly competitive, and superstores are facing high competition [2] from fellow superstore chains, online retailers, and specialty stores. It has significantly affected the customer demands and their shopping behavior as they are looking for convenience and continued availability of products. To reach customers' demands and survive superstores must adapt to the demands to sustain and earn profit. Retail chains need to make data-driven decisions in maintaining stock, inventory management, pricing, and promotional activities.

Accurate sales forecasting plays a key role in implementing these strategies and meeting customer requirements by enabling superstores to meet the product demand by minimizing stockouts situations, and overstock situations and allocating resources as per the requirements. By predicting future sales and trends, stores can manage their supply chain and optimize inventory levels [3] that drive the sales. As the intense competition in the retail sector increases day by day, accurate forecasting helps retail chains to make data-driven decisions and sustain in this challenging market environment.

Data science has emerged as a crucial asset in the retail sector, enabling companies to better understand customer needs and act accordingly to gain a competitive edge. It allows the retailers to analyze the vast amount of sales data including customers' purchase history. Using advanced data analytics and machine learning algorithms [4], retailers can group the products based on demand and trend and manage the inventory appropriately without overstocking or understocking the products. The retail industry is subjected to rapid changes, and data science and analytics help to adjust to the changes. Retailers can monitor real-time data to identify and quickly react to the changes.

Time series forecasting [5] is a technique used in data science to predict future values based on previous values observed over time. This method is particularly helpful when historical data is available. Sales data recorded over time has patterns such as seasonality, cycles, and trends. Understanding these patterns is crucial to forest sales using time series data. In time series data, values are sequential, which means that each data point is affected by its previous values making this a difficult technique compared to traditional prediction techniques. Some of the key advantages of time series forecasting are improved decision-making, adaptability to changes, and understanding of patterns. Despite having significant advantages, time series forecasting is challenging [6] due to the nature of the data. The frequency of data collection can impact the forecasts as high-frequency data might contain more noise and low-frequency data might miss short-term patterns. Data contains noise and outliers which leads to unreliable forecasting. The quality of data available has been improving over time which helps time series models to be applicable in various fields.

A wide range of techniques and methods are available in time series forecasting, each

suits different types of data and different forecasting periods. Researchers have been using these methods to capture trends, seasonal effects, cyclic effects, and relationships within data to forecast future sales accurately. The revolution of time series forecasting started with the introduction of the traditional time series model ARIMA. It has been used widely to forecast future sales in the past. Over time, other statistical models [7] such as SARIMA and Exponential Smoothing (ETS) evolved. Later, machine learning [8] and deep learning [9] based models like Decision Trees, Random Forests, Support Vector Machines (SVM), Gradient Boosting Machines (GBM), Facebook Prophet, Artificial Neural Networks (ANN), and Long Short-Term Memory (LSTM) Networks were used by researchers to deal with time series data. In recent times, hybrid models have gained popularity because of their ability to deal with both linear and non-linear data. The choice of the model depends on the nature of the time series data and forecasting horizon. Researchers often study multiple methods to achieve the best and most reliable forecasting results. Time series forecasting is applied in a wide range of industries such as finance, banking, retail, e-commerce, logistics, inventory management, demand forecasting, healthcare, agriculture, and weather forecasting.

Optimal forecasting is essential for maximizing the benefits of the predictions. Suboptimal forecasting can lead to significant inefficiencies [10], increased costs, and decreased revenue. Considering the importance of optimal forecasting, it is very important to investigate various models used by researchers in previous studies to compare the model performance and find out the optimal model for this specific time series data. Therefore, this dissertation intends to explore models like SARIMA, Random Forests, Gradient Boosting Machines (GBM), SVM, Facebook Prophet, and LSTM to get reliable forecasted sales of a superstore.

There are four primary aims of this research project -

1. Perform EDA and feature engineering to extract relevant features that help to enhance model performance.
2. Develop an effective time series model using recursive time series forecasting technique to predict sales accurately.
3. Explore various time series forecasting algorithms and compare the results to identify the optimal forecasting model.

4. To explore the importance of external features in improving model performance.

The subsequent chapters of this report are organized to provide a detailed overview of the research and findings. The second chapter details the history, existing methodologies, and advancements in time series forecasting with a focus on the retail industry that contributes to the research. The third chapter describes the datasets used. Following this, the fourth chapter describes the various models and techniques used in this study. The fifth chapter provides the results of each model, and the sixth chapter discusses and compares the forecasting accuracy of one model with another model. Finally, the seventh chapter offers the conclusion of this research and the future work that could help improve upon this project's findings.

CHAPTER 2

Literature Review

The literature review provides detailed information about existing research and methodologies relevant to time series forecasting. It explores the concept of time series forecasting and its applications in various industries. Then, it briefs about the historical development of time series forecasting including the evolution of traditional statistical methods to advanced machine learning and deep learning approaches. This provides an understanding of the development and adaption of various forecasting approaches over time. The specific research related to the retail industry was examined. This section highlights the methods used in prior work by researchers in the retail context and sets the foundation for methodology and analysis.

2.1 Time Series Forecasting and Applications

Time Series Forecasting is a statistical method used to forecast future data points based on previous data points recorded in sequence over specific periods in successive time intervals [11]. This approach is particularly useful when future values are driven by historical data points. It involves using a wide range of methods and algorithms to understand the relationships within the data such as seasonality, patterns, cycles, and trends, which play a key role in making future predictions. This is one of the most popular and reliable methods to forecast future values. Time series analysis is a primary step of forecasting that involves analyzing the characteristics of response variables

(e.g., sales, price, and demand), considering time as an independent variable. Unlike traditional regression algorithms, the order and timing of the data points are very crucial for Time Series Analysis (TSA). The applications of Time Series Forecasting are in a wide range of industries such as retail [12], finance [13], climate [14], supply chain management [15], health care [16], transportation and logistics [17] industries where accurate future forecasting is essential for planning and taking data-driven decisions.

2.2 History of Time Series Forecasting and Approaches

Time Series Forecasting has an exceptionally long history, and the concept first emerged in the early 20th century. This approach was first proposed by Yule in 1927 by studying the concept of Auto-Regressive (AR) models [18]. This was the first instance where models were considered time-dependent, and the future values of a time series are modeled as a linear combination of past values. The concept of Moving Averages (MA) in forecasting evolved with its first reference in the early 1900s.

The concepts of AR and MA models led to the introduction of the Autoregressive Integrated Moving Average (ARIMA) model in the early 1970s by George Box and Gwilym Jenkins [19]. It is one of the most widely used methods in time series forecasting to predict future points in time series. It combines three components namely the autoregressive (AR) component, the differencing (I for Integrated) component, and the moving average (MA) component. The development of the ARIMA statistical model is a significant milestone in the history of time-dependent forecasting as it offers robust methods to deal with non-stationary data by making use of differencing components. The work of Box and Jenkins was published as a book "Time Series Analysis: Forecasting and Control" [20] in 1970. This publication provided a systematic approach to applying time series methods in forecasting.

In the late 20th century the advancement in time series forecasting was achieved with the increase in computing power. The complex models were developed to deal with non-linear and high-frequency data. These models were capable of capturing nonlinearity in the data which is often present in real-world data recorded over time. In the 1980s two models namely the autoregressive conditional heteroskedasticity (ARCH) model and

the generalized autoregressive conditional heteroskedasticity (GARCH) model were introduced to deal with change in variance over time [21] which is usual in time series data.

The introduction of machine learning approaches to deal with time series data was first proposed in the early 1980's. Statistical models are not capable of capturing nonlinear and complex relationships in data. Unlike traditional methods, machine learning methods can capture these complex relationships in data without requiring strong assumptions about the underlying data distribution. These models can handle the data with patterns such as seasonality, trends, and cycles. Artificial Neural Networks (ANN) [22] were the first ML models applied to time series data to forecast. These models can model the non-linear relationship between input and output variables that helps to capture complex patterns. Later advanced ML algorithms like SVM [23] and ensemble methods like Random Forests [24], Gradient Boosting Machines [25] were introduced to deal with high dimensional and non-linear data.

The major turning point of time series forecasting occurred in the early 1980s [26] with the evolution of deep learning algorithms. Deep learning brought significant advancements in forecasting. The most remarkable of the advancements was the development of Recurrent Neural Networks (RNN). They were first brought up by two significant works by Jordon [27] in 1986 and Medsker, L [28] in 1999. They were designed to process the sequential data and to be compatible with time series data. The standard RNN suffered from a vanishing gradient problem which was then overcome with the evolution of the Long Short-Term Memory (LSTM) [29] algorithm. This model is capable of modeling long-term dependencies and handling them in sequential data. These models have been widely used in various fields such as retail, finance, and climate forecasting. The breakthrough of sequential models occurred in the 2010s with the evolution of deep learning frameworks like TensorFlow and PyTorch.

Recent advancement in time series forecasting is achieved with the adoption of transformers for time series forecasting tasks. Transformers were ideally developed to handle the natural language processing tasks. In early 2010 they were adopted for time series forecasting [30] because of their ability to handle parallel computations and capture global dependencies in complex and high dimensional time series data. Vaswani [31]

introduced a transformer model for time series tasks in 2017 and it was explored in subsequent years to handle multivariate time series and complex sequential data.

The 2020s have seen the increased use of hybrid models to deal with time series data and it has gained popularity in recent studies. In a time series if both linear and non-linear relationships are present during the same interval, it is not easy to model both using a standalone model [32]. This limitation can be overcome by using a hybrid model. These models are an integration of traditional statistical models with machine learning or deep learning algorithms such as ARIMA-SVM [33], ARIMA-LSTM, and CNN-RNN for multidimensional time series forecasting. These hybrid models combine the strength of both the models, allowing the modeling of both linear and non-linear patterns [34] in time series data. These models are used widely as they address the limitations of using statistical models or machine learning models alone. For instance, the ARIMA-LSTM hybrid model combines the strengths of both models, where the ARIMA model captures the linear relationships such as trends and seasonality and the LSTM model captures the non-linear relationships.

In 2017, Facebook developed and introduced an open-source forecasting tool specially designed to deal with time series data. It was developed internally at Facebook by Sean J. Taylor and Ben Letham [35]. The reason behind the development of the prophet model is the Facebook data science team faced challenges in forecasting various business metrics. Traditional time series models are available, but they often need domain knowledge to tune the models, and they have limitations in dealing with irregularities in the time series data. Prophet is designed to be user-friendly and accessible with minimal statistical expertise. It includes automatic hyperparameter tuning, allowing users to forecast with few lines of code. This model has been widely used by data scientists and business analysts in various business fields like sales forecasting [36], demand forecasting, and inventory management.

2.3 Related Works

There are numerous methods available in statistics, machine learning, and deep learning to forecast future sales considering date as an independent variable. In the past, researchers have made use of different methods available to forecast sales and compare

the results between the models. This related works section of the literature review explores past research, methods used by researchers to forecast sales, and the performance of different algorithms used in the context of superstore sales forecasting, highlighting their strengths and limitations. In the history of time series forecasting, ARIMA and SARIMA are the two most widely used traditional time series models to forecast future sales. Among the various time series models available, these two models are extensively used to forecast retail store sales. In [37] Singh *et al.* explored time series modeling to forecast sales. Accurate sales prediction helps stores in reducing losses and enhance the customer experience. This paper explores sales data using three different approaches such as Holt-Winters exponential smoothing, neural network autoregression, and ARIMA models. This study aimed to understand the best-performing model on historical sales data with patterns. The model performance was evaluated using mean absolute percentage error (MAPE). The results indicated that seasonal ARIMA performed much better than the neural network autoregression model, with a MAPE of 2.88 compared to 4.66. Despite the low error, research has a scope to forecast sales precisely by making use of comprehensive data, which makes forecasting reliable.

Jiang *et al.* [38] made use of statistical and machine learning models and compared the performance of the models using weighted root mean square error (WRMSE). In this paper, the superstore sales data has a seasonal effect, and to deal with seasonality and trends, seasonal ARIMA (SARIMA) was used as a traditional time series model. Machine learning approaches such as LightGBM, and Prophet models were examined to improve sales forecasting accuracy. This study showed that machine learning models outperformed the traditional model SARIMA because of their ability to capture the non-linear relationship in the data. The performance of LightGBM surpassed the Prophet model with a lower WRMSE of 0.617 compared to 0.694. The results suggest that the machine learning models are effective for forecasting sales when the data has seasonal effects by offering valuable insights to the retailers to forecast sales by category and region.

Exploratory data analysis and feature engineering play a key role in data preparation, which has a direct impact on model performance. [39] The research in this paper focuses on forecasting weekly sales using various machine learning algorithms and comparing the performance differences between them. In this research, S. Pang stated

the importance of EDA in finding out key factors influencing sales. The analysis showed that key factors impacting sales include store size and departments like grocery, electronics, pharmacies, and personal care. It also revealed that larger stores are likely to get higher weekend sales. Three machine learning models, Linear Regression, Random Forests (RF), and Decision Tree (DT), were used to forecast the sales. Results show that Linear Regression was the least-performing model with a high error rate and low r2 score. This is because of the non-linear patterns that exist in sales data recorded over time, which Linear Regression cannot capture effectively. Random Forest outperformed Decision Tree with a low RMSE error of 3804.7 compared to 5248.3 and a higher r2 score.

In this paper [40], Deng *et al.* stated that the combination of machine learning and deep learning helps time series forecasting to predict sales precisely. This paper explores the use of the GBDT model based on the LightGBM framework, an advanced gradient boosting model, logistic regression, and support vector machines (SVM) for forecasting Walmart superstore sales. This study highlights the importance of data science and machine learning in enhancing retail competitiveness. It also involves the preprocessing of large datasets and feature engineering by removing irrelevant features. The performance of SVM and LightGBM-based GBDT models was far better compared to linear regression. Although SVM performed better, LightGBM-based GBDT outperformed SVM with a low root mean square scaled error (RMSSE) of 0.641 compared to 0.732. Additionally, this paper identified the top 20 most important features, providing valuable insights for optimizing sales strategies.

The introduction of deep learning made a huge difference in time series forecasting as it offers several advantages such as automatic feature extraction, handling non-linearity by modeling complex patterns, handling high dimensional and multivariate data, capturing long-term dependencies, and parallel processing capabilities. In recent times, there has been increased usage of deep learning in time series forecasting. Li *et al.* [41] presented a deep learning approach using Long Short-Term Memory (LSTM) networks for predicting future sales based on historical superstore sales data. This study also includes advanced feature engineering techniques used to improve the model performance. This paper analyzed two machine learning algorithms, Linear Regression and Support Vector Machines (SVM), along with LSTM to compare the

model's performance. Experimental results showed that the LSTM model achieved a root mean squared scaled error (RMSSE) of 0.834 which is lower than the SVM and Linear Regression model's error. This study details the effective use of hyperparameter tuning and feature extraction processes which led to LSTM's better performance and showed LSTM's recurrent structure ability to capture the patterns in time series data. The significant limitations of this study are computational resources and time required for hyperparameter tuning.

A lot of research has been done in time series forecasting and different challenges were faced while using different methods. Facebook data science team faced difficulties in handling time series data and to overcome the limitations of existing methods, they developed a forecasting tool named Facebook Prophet. In [42] Jha and Pande explored various forecasting models for superstore sales prediction with a focus on the Facebook Prophet model. This study compared the performance of the prophet model with the additive model and traditional time series model ARIMA. The research results indicate that the performance of Facebook Prophet is superior to other models with low error and better model fitting. Besides highlighting the effectiveness of the proposed model, the study also stated that performance could be enhanced further by using fusion techniques and addressing scalability challenges for handling larger datasets. Unlike other machine learning algorithms, there is no control over hyperparameters in the Prophet model which is considered a limitation of this model.

2.4 Summary

The literature review highlighted the wide range of methodologies and approaches used in the field of time series forecasting and provided a base for the current research. Different methods have performed better on different data, and it is unclear which models yield the best forecasting results. Linear models like ARIMA, and SARIMA perform better when the relationships in data are not complex. When non-linear relationships exist in data, machine learning, and deep learning models are necessary to capture these complex relationships. Hence, this research aims to investigate a variety of algorithms that performed better in the past to forecast superstore sales and explore the effective approaches for forecasting using time series data. In this study, the recursive

time series forecasting technique [43] has been employed to make future forecasting. It is a method used to predict future values using one step ahead prediction process. It repeatedly uses the trained model to predict the next value in a series and uses the predicted value as input to make the next prediction. This method is called "recursive" because the model's predictions are recursively used as inputs for further predictions. The best-performing models from previous studies were employed using recursive time series forecasting techniques to find effective forecasting models.

CHAPTER 3

Data Description

The datasets used in this research were taken from the Kaggle platform. The data is taken from a competition titled "Store Sales - Time Series Forecasting" which was hosted on the Kaggle platform. The sales data is related to Corporacion Favorita, a large Ecuadorian-based grocery retailer. It contains daily historical sales transactions of 54 stores located across Ecuador and each of the 54 stores contains sales data of 33 categories and the data was collected from January 2013 to August 2017. Furthermore, these stores are categorized into 5 types and 17 clusters. There are 6 datasets such as train, holidays and events, oil, stores, and transactions each having specific information regarding sales. Among these datasets, the useful ones for this research were used. The datasets used in this research project can be accessed at the following link: Store Sales - Time Series Forecasting.

3.1 Primary Dataset

The primary dataset used in this research was a train dataset taken from the Kaggle competition. It consists of daily information related to sales such as date, store number, category of the product, sales, and number of products on promotion.

3.2 Supporting Datasets

In this research 3 supporting datasets were used such as stores dataset, holidays events dataset, and oil dataset. The stores' dataset contains useful information including city, state, store type, and store cluster. The holidays and events dataset has data like holiday type, locale, locale name, description, and a binary feature transferred which details if the holiday is transferred or not. The oil prices dataset comprises daily oil prices. Ecuador is an oil-dependent country, and its economic health is liable to changes in oil prices.

Before preparing the final dataset, data exploration was carried out on each dataset to thoroughly understand the underlying patterns, trends, and relationships within the data and identify useful features for modeling. Feature engineering was performed to create important features based on the existing features in supporting datasets. The final dataset includes the external features from supporting datasets. **Table 3.1** shows the description of each feature in the datasets.

Feature	Description
store_ nbr	Store number
family	Product category
sales	Number of sales
onpromotion	Number of promotions
city	City in which the store is located
state	State in which the store is located
type	Type of the store
cluster	Cluster of the store
dcoilwtico	Daily oil price

Table 3.1: Data Description

CHAPTER 4

Methodology

This section outlines the systematic approach followed in this study including models, techniques, and methods that were used to achieve the objectives of this research project. Data preprocessing was performed to process the inconsistencies in data. Then the datasets were merged based on the common features such as date and store number. New features were created by transforming existing features that help enhance model performance and it was achieved by using feature engineering technique. The patterns in data such as cycles, seasonality, trends, and relationships within data were investigated using exploratory data analysis (EDA). The final dataset was created in such a way that the data suits the problem statement of the project. The methodology mainly focuses on building time series forecasting models using recursive forecasting techniques. Various statistical, machine learning, and deep learning models were extensively trained, tuned, and evaluated to find the optimal and reliable model for forecasting accurate sales.

4.1 Data preprocessing

Data preprocessing is a crucial step in any data science research project as it transforms raw data into machine-compatible data for modeling and analysis. This process involves data cleaning, data transformation, dealing with anomalies like missing values, and data encoding. In this project, data preprocessing includes filling in missing values,

merging datasets, and encoding categorical variables. As multiple datasets have been used in this project, the first step involved checking each dataset thoroughly. This process included checking the data information and null values using functions in Python.

4.1.1 Imputation of Missing Values

After examining each dataset, it was found that only the oil prices dataset has missing values. Time series data often contain temporal patterns such as trends and seasonality. Imputing missing values using basic traditional methods such as mean, and median will miss the natural flow of the series and lead to information loss as these methods don't account for patterns. Therefore, missing values in the oil prices dataset were imputed using the linear interpolation method. It is one of the most widely used methods to impute missing values in time series data because of its effectiveness in estimating unknown values. This method makes sure that the trend is maintained while filling in missing values and helps maintain the consistency of the time series.

4.1.2 Merging Datasets

Following the imputation of missing values, the datasets were merged into one single dataset to create a detailed dataset for analysis. Datasets were merged using common features between them such as date and store number. The train dataset which is a primary dataset was merged with the stores dataset based on a common key store number. Then the oil dataset was merged with the combined dataset based on the date feature. This integration of datasets ensured that relevant external features were included to enhance the dataset for predicting sales using time series forecasting techniques.

4.2 Feature Engineering

Feature Engineering is the process of creating, modifying, and transforming the raw features in the data into useful features. It is a crucial step in data science projects as these features help in capturing patterns and underlying relationships in data which in turn improves the accuracy of model predictions. This process includes creating new features from existing features, adding external features, and identifying the influencing

features in the data. Initially from date, feature time-based features such as day, month, year, and day of the week were extracted and appended to the dataset. Secondly, the holidays events dataset was used to create two features namely holidays and national holidays. This dataset contains information about regional and national holidays, events, bridge holidays, and additional holidays. The holidays' feature is a binary data feature where 1 represents a holiday and 0 represents not a holiday. This feature considers local holidays. The national holiday feature is also a binary data feature, and it considers only nationwide holidays. These engineered features played an important role in boosting the overall performance of the models.

4.3 Exploratory Data Analysis

Exploratory data analysis (EDA) is an essential step in data analysis and its primary goal is to understand the data distribution, relationships within data, and underlying patterns in data such as trends, cycles, and seasonality. EDA involves various methods that include statistical techniques along with data visualization techniques that help in visualizing the distribution of sales across different stores and product categories and understanding data characteristics. EDA helps in making informed decisions about final dataset preparation and model selection which ultimately leads to optimal forecasting.

4.3.1 Sales per Store

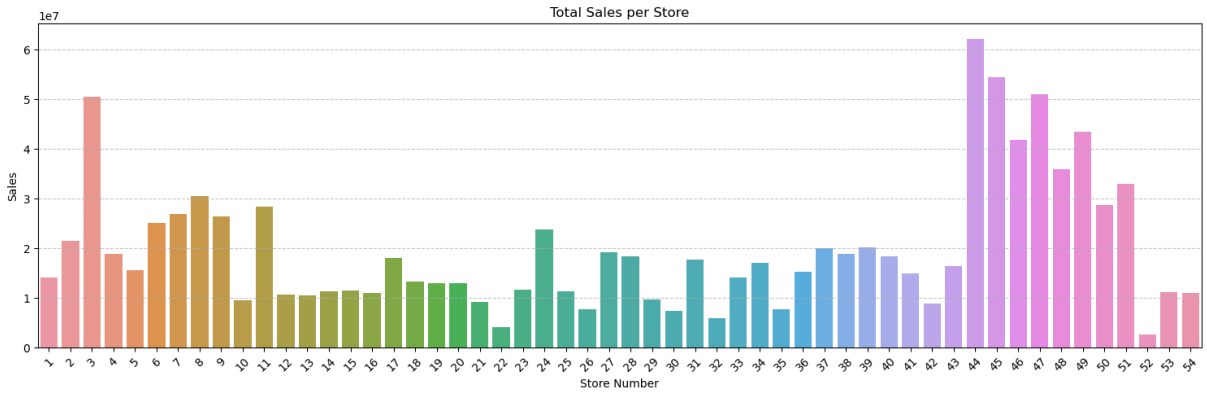

Figure 4.1: Total Sales per Store.

This analysis focused on exploring the total sales of each store to understand the sales distribution of the superstore chain across different stores. It was observed that the sales varied extremely across stores. Few stores recorded higher sales compared to the majority of the stores. Store number 44 saw the highest number of sales and store number 52 recorded the least number of sales.

4.3.2 Promotions per Store

The analysis of promotions per store was conducted to understand the impact of promotions on sales in various stores and visualize the total number of promotions in each store. By examining the frequency of promotions, it was observed that sales are related to the number of promotions. The stores with a higher number of promotions recorded higher sales compared to stores with the lowest number of promotions. Store number 52 recorded the least number of promotions which has a direct impact on total sales in that specific store.

Figure 4.2: Total Promotions per Store.

4.3.3 Total Sales per Family

The exploration of total sales per family helps to understand the sales distribution and performance of various product categories. Grouping the sales according to categories across stores helps in identifying patterns that include finding top-selling product categories along with the categories with the least number of sales. It was observed that groceries, beverages, produce, cleaning, and dairy were among the top-selling

categories. The sales of groceries and beverages alone accounted for around half of the total sales.

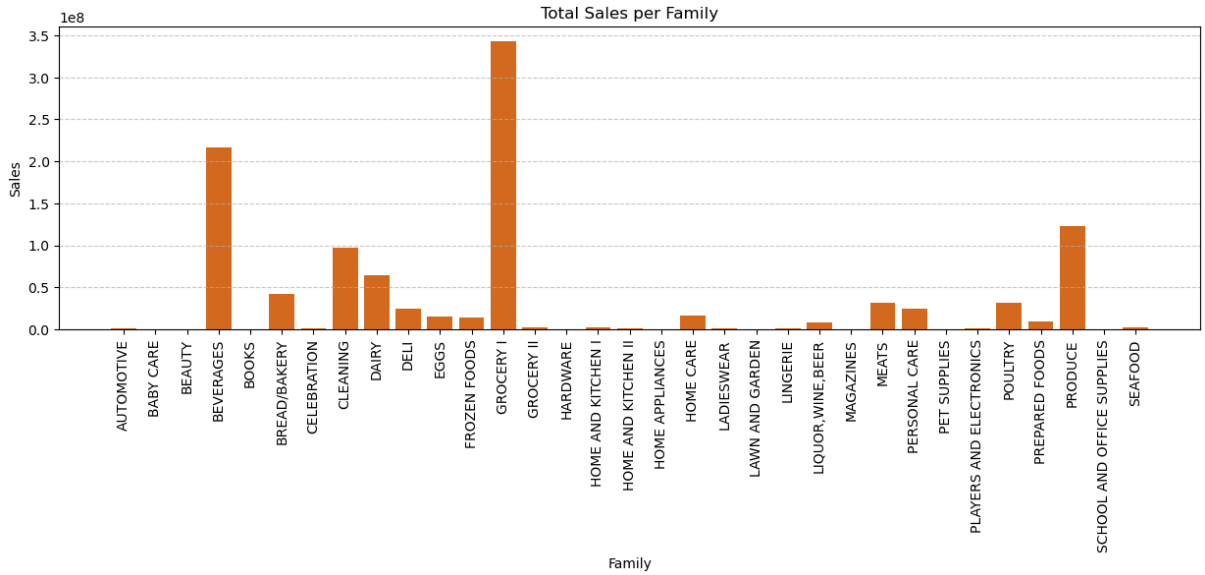

Figure 4.3: Total Sales per Family.

4.3.4 Total Promotions per Family

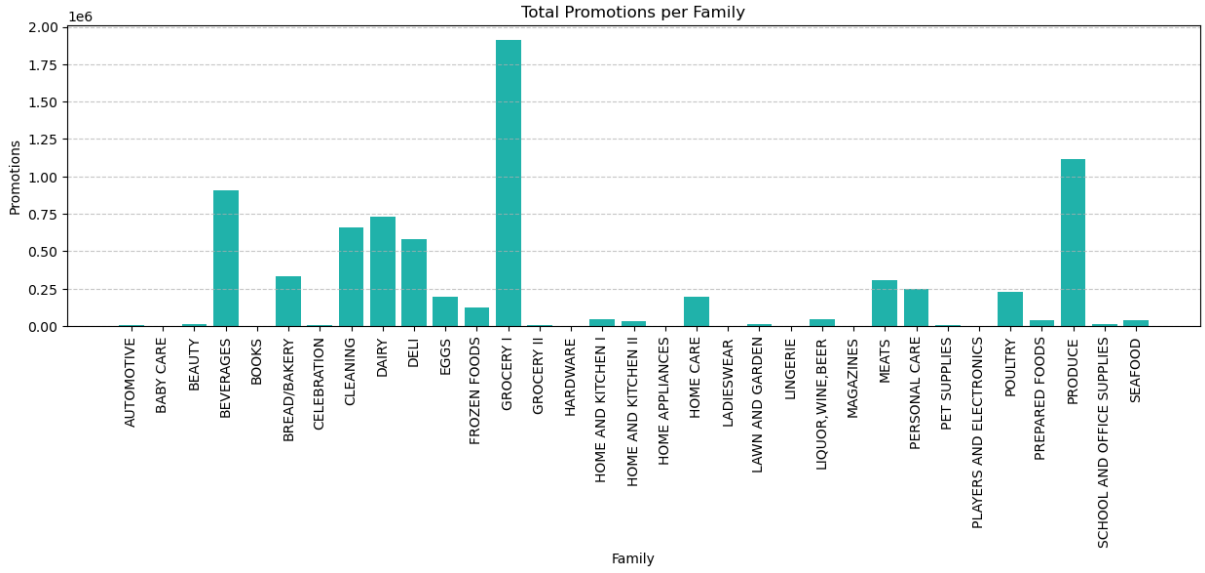

Figure 4.4: Total Promotions per Family.

This analysis was carried out to examine the impact of promotions on sales of each product family. The number of promotions by product family was aggregated to identify

the categories with the highest promotions and to explore how these promotions drive sales of each family. It was observed that the sales are directly related to promotions. The product categories with the highest total sales have the highest number of promotions as well.

4.3.5 Total Sales and Promotions per Year

In this analysis, annual trends in total sales and frequency of promotions were explored to understand the relationships and patterns in the data. The sales and promotions were grouped by year to see the trend in sales over the years and investigate how the frequency of promotions changed over the years. It also includes understanding the impact of changes in promotion frequency on yearly sales. The analysis revealed that there is a clear yearly trend, and the percentage of total sales increased year by year. Results show that 16 percent of total sales were recorded in 2013 and gradually increased every year. It is the same with frequency of promotions as well where in 2013 the percentage of promotions was 0 and in 2017 it was around 69 percent. This indicates that there is a positive correlation between total sales and the frequency of promotions.

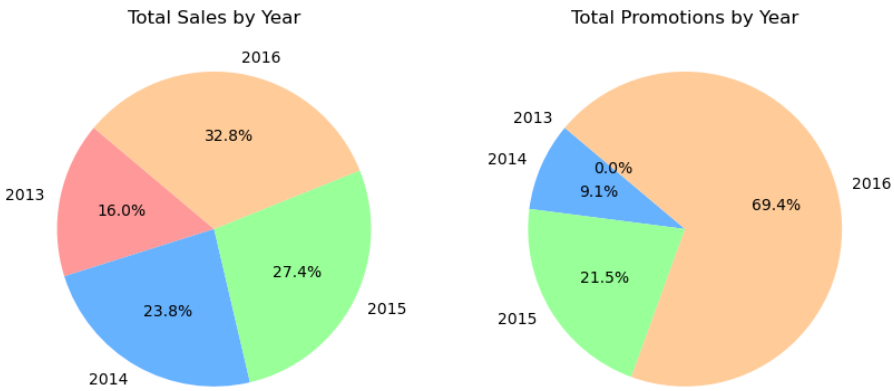

Figure 4.5: Total Sales and Promotions per Year.

4.3.6 Oil Prices over the Years

Ecuador is an oil-dependent country and oil accounts for nearly one-third of the country's GDP. The economy of the country is highly sensitive to changes in oil prices. This analysis focused on exploring the oil prices over the years. The data analysis showed

that there are patterns in oil prices over the years. The oil prices were at peak around mid-2013 and reached the lowest mark at the start of 2016. It was observed that the oil prices were around the same range from 2013 until the middle of 2014 and then there was a drastic fall in oil prices until the start of 2016.

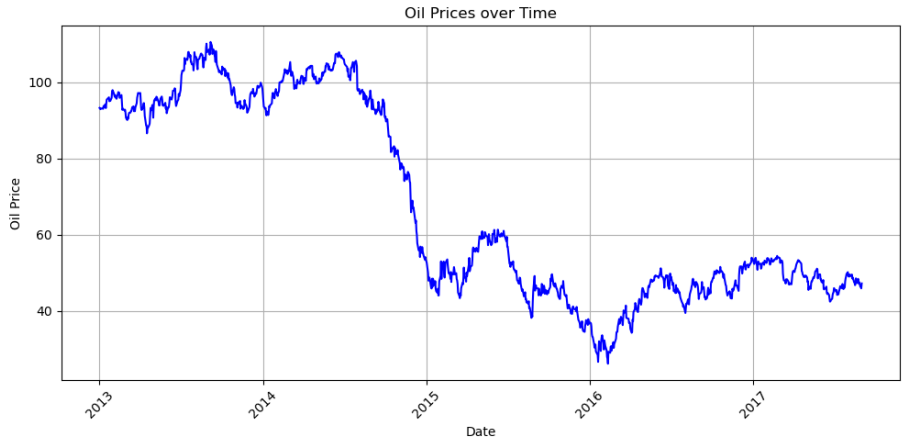

Figure 4.6: Oil Prices over the Years.

4.3.7 Daily Sales vs Oil Prices

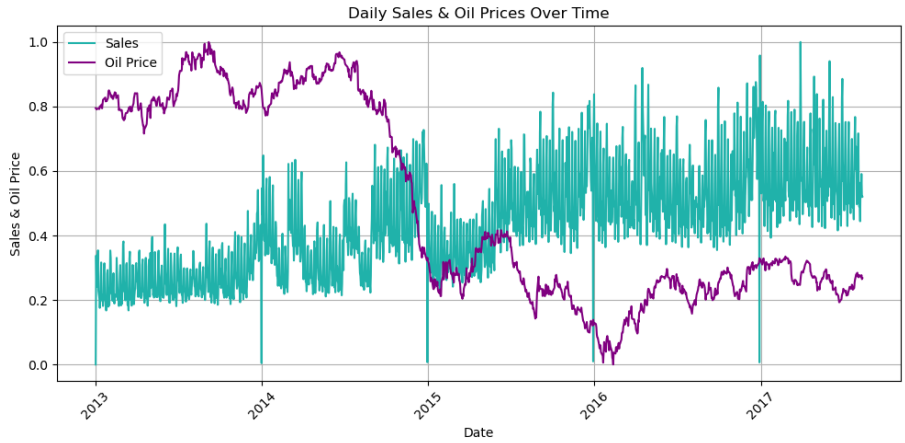

Figure 4.7: Daily Sales vs Oil Price.

This analysis was carried out to explore the relationship between daily sales and oil prices and to understand how changes in oil prices affected sales performance. Sales and oil prices were plotted using a line plot and it was observed that there are clear patterns in the data. The analysis showed that there are some periods where the change in oil prices has affected sales, and there seems to be a negative correlation between oil prices and sales. This analysis helped to understand to what extent the oil prices

impacted the overall sales, and it provided insights for incorporating the oil prices feature into forecasting models to improve the performance.

4.3.8 Monthly Sales Analysis by Year

In general sales data exhibits patterns like weekly or monthly trends. This analysis focused on examining the monthly sales trend in each year to identify the seasonal patterns in the data. The sales data was grouped by month and year and plotted using a line plot. The analysis showed that there are some peak months and the sales follow a similar pattern each year. It was observed that the sales have gradually grown throughout the year and reached their peak at the end of each year. The data is available for five years from 2013 to 2017 and every year recorded the highest sales at the end of the year. Each year showed a gradual increase in sales except 2014 as it has fluctuations in sales throughout the year.

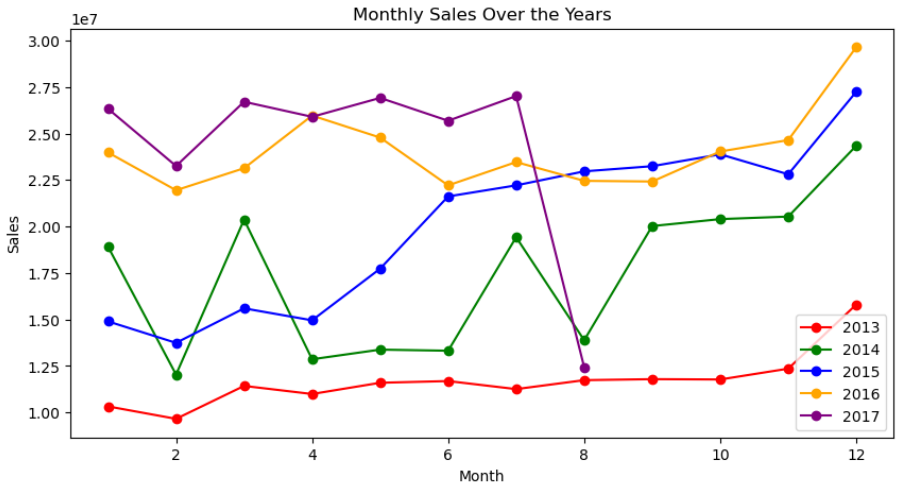

Figure 4.8: Monthly Sales over the Years.

4.4 Data Preparation

Data preparation is the process of preparing the final data including filtering data, removing unwanted columns, and encoding the categorical features according to the problem statement. It is one of the crucial steps in data science projects that directly impact the performance of the model. Proper data preparation ensures that data is clean and consistent which is essential for making accurate predictions. The dataset consists

of daily sales data of 54 stores each with 33 categories. In preparing the final dataset, initially, the dataset was filtered to include the sales data from only one randomly selected store and 3 categories for detailed forecasting. This approach allows for a deeper exploration of sales patterns and allows accurate forecasting for these specific categories. Then features such as state, city, cluster, and type were removed as data from a single store was considered for analysis. Finally, categorical features in the data were encoded using ordinal encoding as machine learning models can only work with numeric data.

In this research project, the data of the selected three categories was analyzed separately for forecasting. Therefore, three data frames were created by filtering the dataset based on a product category for ease of analysis. The historical data is available for 1684 days for each category and the models were trained using data from 1654 days to forecast the sales for the next 30 days. The data was separated based on the category and each category's data was divided into train and test sets with the train set containing 1654 rows and the test set containing 30 rows. Machine learning and deep learning models perform better with data where all features are on the same scale. Therefore, train and test data was scaled using a standard scaler. In general time series data has outliers and to avoid the influence of outliers standard scaler was used.

date	family	sales	onpromotion	dcoilwtico	year	month	day	dayname	holiday	nationalholiday
2013-01-01	BEVERAGES	0.0	0	93.31	2013	1	1	Tuesday	1	1
2013-01-01	DAIRY	0.0	0	93.31	2013	1	1	Tuesday	1	1
2013-01-01	GROCERY I	0.0	0	93.31	2013	1	1	Tuesday	1	1
2013-01-02	BEVERAGES	1091.1	0	93.14	2013	1	2	Wednesday	0	0
2013-01-01	DAIRY	579.0	0	93.14	2013	1	2	Wednesday	0	0

Table 4.1: Final Dataset.

4.5 Time Series Analysis

Time series analysis involves exploring data points collected in regular intervals over time to find patterns in data such as seasonality, trends, and cycles and to understand the relationships within data which helps to forecast future data points accurately. Time series forecasting is the process of predicting future data points based on historical data points. It uses the insights gained from time series analysis to forecast reliable future data points. This approach is widely used in various industries such as finance, banking,

retail, logistics, and many other industries. In this research time series forecasting approach was used to forecast the sales of a retail superstore chain. Various models have been used in this study to find the optimal performing model for this specific data. The models studied in this research include traditional time series models such as ARIMA, and SARIMA and advanced machine learning models like Random Forest, Extreme Gradient Boosting Machine (XGB), Support Vector Machine (SVM), and a deep learning algorithm LSTM.

4.5.1 Traditional Time Series Models

Traditional time series models are foundational models in time series analysis. These models were used to forecast future data points making use of past data points recorded over time in successive time intervals like hourly, daily, weekly, monthly, quarterly, or yearly. The predictions were made based on the patterns and structures within the data such as trends, seasonality, and cyclic effects. In time series forecasting, traditional models serve as baseline models for more advanced forecasting models. They are used in a wide range of applications because of their simplicity in implementation and interpretation. Traditional models excel in capturing linear relationships and simple patterns in data. However, they are limited in handling data with complex patterns and nonlinear relationships and more advanced techniques and models were required to handle complex data.

4.5.1.1 Data Stationarity

Data stationary is an important concept in time series analysis. It refers to the statistical features of the data such as mean, variance, and covariance remaining constant over time. A stationary series is one with properties of mean, variance, and covariance that do not vary over time. Stationary data helps traditional time series models in simplifying the analysis and making relationships in data consistent and predictable. Time series models like ARIMA and SARIMA expect stationarity in data to make reliable predictions. There are techniques available to convert nonstationary data to stationary such as differencing, transformation, seasonal differencing, and detrending. These techniques are often used to achieve data stationary and make it suitable for analysis.

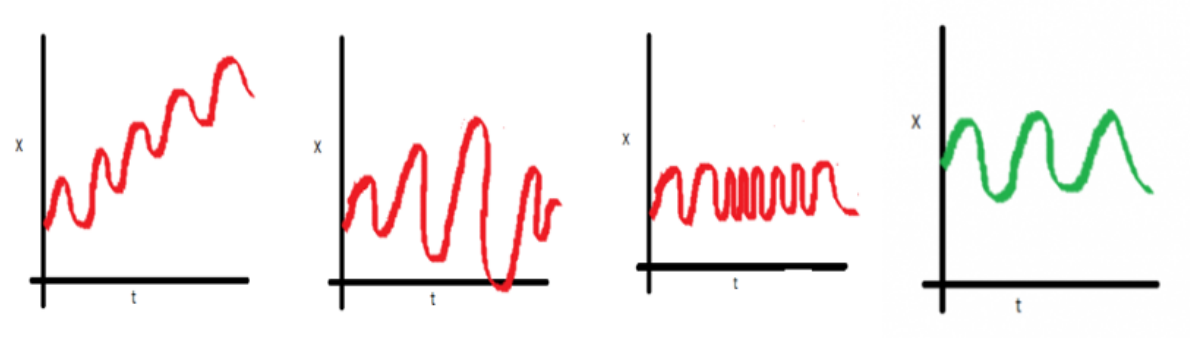

Figure 4.9: Non-stationary (red curves) and Stationary Data (green curve) [44].

The first plot in figure **Figure 4.9** shows that the mean increases over time indicating a non-stationary time series with the mean varying over time and there is an upward trend in the time series data. In general, stationary time series data doesn't exhibit a trend. The second plot shows a time series with variance changing over time indicating the variance is a function of time. For a series to be stationary, it must have a constant variance. The third plot represents a non-stationary time series with varying covariance and the spread of data becomes close with the increase in time. It represents that covariance is a function of time. However stationary data must have constant covariance. The fourth plot represents a stationary time series with constant mean, variance, and covariance over time. Data that inherit these characteristics will make forecasting easier and increase the model's prediction accuracy. There are many techniques available to make non-stationary data stationary and make it suitable for analysis.

4.5.1.2 Seasonal Decomposition

Seasonal Decomposition is a statistical technique used in time series analysis to visualize and understand the core components of time series data. This technique helps to separate time series into components such as trend, seasonality, and residuals also known as noise. Separating the series into these components uncovers the underlying patterns in the data and gains valuable insights into long-term trends, repeated seasonal effects, and random fluctuations that help in making data-driven decisions and improving forecasting accuracy. This method is particularly useful in identifying the structure of the data to build reliable models for forecasting future values. There are two models available in this technique such as additive and multiplicative models. Based on the

nature of the time series either one of these models can be used to perform seasonal decomposition. The two types of models used in seasonal decomposition are-

Additive Model- The additive model assumes that the seasonal components were added together to form a series. Therefore, the time series is represented as the sum of trend, seasonality, and residual components. This model is suitable when variations in seasonality remain consistent over time. The equation of the additive model is given by **Equation 4.1**.

$$Y(t) = T(t) + S(t) + R(t) \tag{4.1}$$

Where:

- $Y(t)$ = Series data at time period t.
- $T(t)$ = Trend component at time period t.
- $S(t)$ = Seasonal component at time period t.
- $R(t)$ = Residual component at time period t.

Multiplicative Model- The multiplicative model assumes that the seasonal components multiply together to form a series. Therefore, the time series is represented as the product of trend, seasonal, and residual components. This model is suitable when the variations in seasonality change proportionally with the trend. The equation of the additive model is given by **Equation 4.2**.

$$Y(t) = T(t) \times S(t) \times R(t) \tag{4.2}$$

Where:

- $Y(t)$ = Series data at time period t.
- $T(t)$ = Trend component at time period t.
- $S(t)$ = Seasonal component at time period t.
- $R(t)$ = Residual component at time period t.

Components of Time Series Data

Seasonal Component- The seasonal component of the time series data is defined as the repeating patterns in data at regular time intervals. It is one of the basic components of the time series data. These seasonal patterns repeat in specific periods such as weekly, monthly, yearly, or fixed periods. This component captures the fluctuations that occurred due to seasonal effects like holidays, weather, and economic changes. Identifying and interpreting the seasonal component helps in preparing the model for seasonal fluctuations.

Trend Component- Trend is one of the core components of the time series data and refers to the long-term changes or patterns in the time series. It shows whether the time series is moving upwards, downwards, or remaining constant over time. It captures the general direction in which the data is moving ignoring the short-term fluctuations. Separating the trend component helps in making data-driven decisions based on the direction of the data.

Residual Component- The residual component of the time series data represents the random fluctuations in the data, and it is referred to as the remaining component after removing trend and seasonal components. This component captures the noise in the data such as random and irregular fluctuations that cannot be explained by the other two core components of the time series. By removing the trend and seasonal components, it is possible to analyze the random patterns in the data that help to improve the forecasting accuracy.

In this research project additive model was used in the seasonal decomposition method to check for seasonality in the sales data. It divides the time series into core components such as the seasonal, trend, and residual components. The seasonal decomposition technique was used to check seasonality in sales data of all three categories and the results indicated there are strong seasonal effects in sales data. **Figure 4.10** shows the seasonal decomposition plot of sales data related to the beverages category. The seasonal component in the decomposition plot shows patterns repeated in regular intervals indicating the data has seasonal effects. The trend component in the plot shows the line is moving upward suggesting the sales data has an upward trend.

4.5.1.3 Statistical Methods to Investigate Stationarity

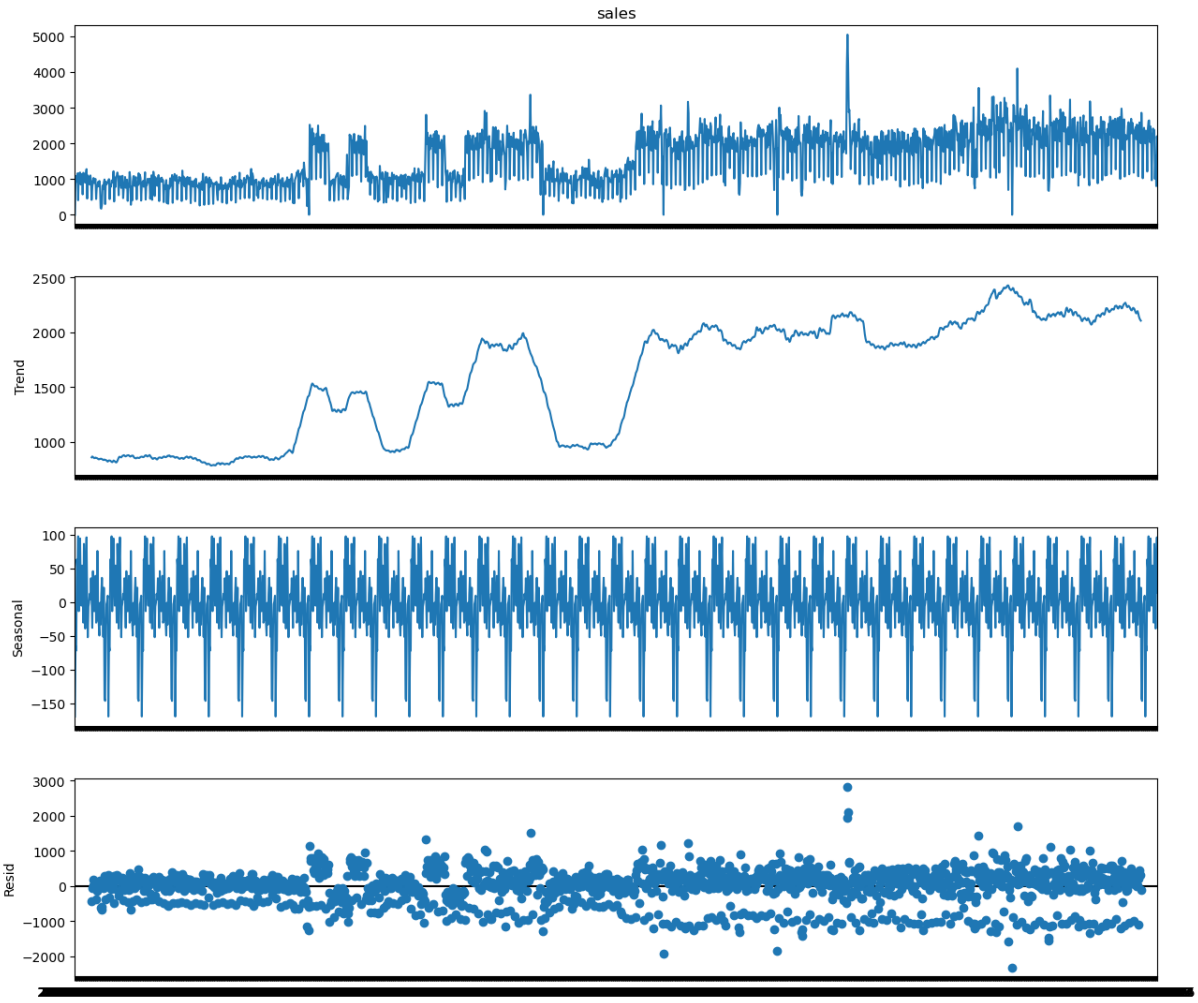

Figure 4.10: Seasonal Decomposition Plots of Beverages Sales Data.

Investigating data stationarity is a crucial step in time series analysis. Many traditional forecasting models work on the assumption that data is stationary. Stationary data have statistical properties such as mean, variance, and covariance that remain constant over time. By checking data stationarity, the decision can be made whether data is suitable for modeling or data needs to be stabilized before using it for modeling. There are many statistical and visualization methods available to investigate data stationarity. Among them, the three widely used methods are the Augmented Dickey-Fuller (ADF) test, the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test, and the Rolling Statistics test.

Augmented Dickey-Fuller (ADF) test- The ADF test is the most widely used statistical hypothesis test in time series analysis to investigate data stationarity. It is an extension of the basic Dickey-Fuller test by adding lagged terms to account for autocorrelation. It is used to determine whether a time series is a stationary series or a non-stationary

series by testing for the presence of a unit root. The presence of unit roots indicates that statistical properties of data vary with time. The null hypothesis (H0) of this test is that the data is non-stationary, and the alternate hypothesis (H1) is that the data is stationary. The result of this statistical test includes the test statistic, p-value, and critical values. The test statistic usually returns a negative value, and the more negative value provides strong evidence to reject the null hypothesis. If the p-value is less than the significance level (0.05) and the test statistic is less than the critical values at different significance levels (1% , 5% , 10%), the null hypothesis is rejected and indicates the time series is likely stationary.

Results of Adfuller (ADF) Test :	
Test Statistic	-3.113822
p-value	0.025547
Critical Value (1%)	-3.434295
Critical Value (5%)	-2.863283
Critical Value (10%)	-2.577698
The series is likely non-stationary	

Table 4.2: ADF Test Results of Beverages Sales Before Differencing.

Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test- The Kwiatkowski-Phillips-Schmidt-Shin test is used to investigate the stationarity of time series data. It is also one of the widely used statistical hypotheses tests in time series analysis and is specifically used to check whether a time series is trend stationary. The KPSS test also includes lag terms to deal with autocorrelation. Unlike the ADF test which checks for non-stationarity, this test considers the null hypothesis (H0) as data is stationary and the alternate hypothesis (H1) as data is non-stationary. It tests considering that the unit root is not present which indicates the statistical properties of the data are constant over time. This method calculates the test statistic, and it is compared against the critical values at different significance levels. The higher test statistic provides stronger evidence against stationarity and indicates the time series is likely non-stationary.

Results of KPSS Test :	
Test Statistic	4.809765
p-value	0.010000
Critical Value (10%)	0.347000
Critical Value (5%)	0.463000
Critical Value (2.5%)	0.574000
Critical Value (1%)	0.739000
The series is likely non-stationary	

Table 4.3: KPSS Test Results of Beverages Sales Before Differencing.

Rolling Statistics test- The rolling statistics test is a statistical and visualization method to test data stationarity. It is a very simple yet very effective technique to investigate the time series data. This approach involves calculating the rolling mean and rolling standard deviation (SD) over a fixed window of time and plotting them to check for stationarity. The rolling mean is calculated to capture the trend in the data and the standard deviation is calculated to check the spread of the data around the mean. In stationary time series data, the moving mean and standard deviation remain relatively constant over time. If the plot shows significant fluctuations, it indicates the time series is not stationary. The data transformation technique helps in converting non-stationary time series data to stationary time series data and makes it suitable for analysis. **Figure 4.11** shows the rolling statistics of beverages category sales.

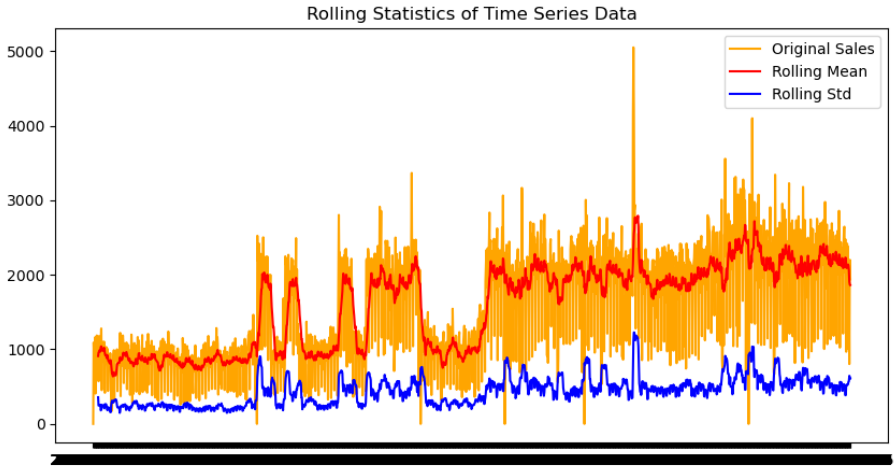

Figure 4.11: Rolling Statistics of Beverages Sales Before Differencing.

The ADF test and KPSS tests are statistical hypotheses tests and the rolling statistics test is a visualization technique to investigate data stationarity. It is always recommended to use more than one method to check data stationarity as each method has unique strengths and weaknesses. Investigating with multiple methods helps to perform an accurate analysis of time series data and achieve reliable results. Cross-validating the results of multiple methods reduces the risk of misleading results. This approach helps to make well-informed decisions when preparing data for modeling.

Differencing- Differencing is one of the many available techniques to transform non-stationary data into stationary data. It is one of the most widely used methods in time series analysis. There are many other methods available such as log transformation, square root transformation, box-cox transformation, and power transformation. It is preferred over other methods because of its ability to directly address the non-stationarity in data by removing trends and stabilizing the mean and standard deviation. It is flexible in handling non-stationarity as it can be applied at multiple levels such as first order, second order, etc., based on the structure of the data. It is effective in removing trends and maintains the structure of the data by focusing only on removing non-stationarity. The equation of the differencing is given by **Equation 4.3**.

$$Y'(t) = Y(t) - Y(t-1) \tag{4.3}$$

Where:

- $Y(t)$ = Value in the series at time t.
- $Y(t-1)$ = Value in the series at time $t-1$.
- Y' = Differenced Value in the series at time t.

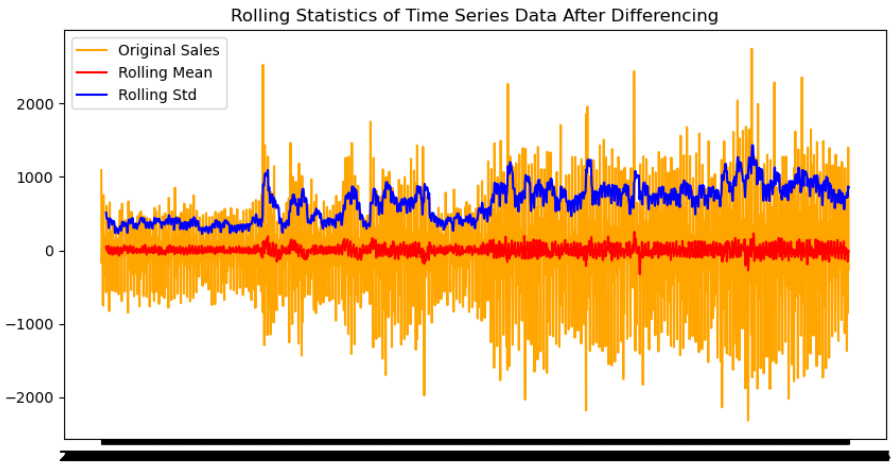

Figure 4.12: Rolling Statistics of Beverages Sales After Differencing.

Results of Adfuller (ADF) Test :	
Test Statistic	-1.021868e+01
p-value	5.396279e-18
Critical Value (1%)	-3.434293e+00
Critical Value (5%)	-2.863282e+00
Critical Value (10%)	-2.567697e+00
The series is likely stationary	

Table 4.4: ADF Test Results of Beverages Sales After Differencing.

Results of KPSS Test :	
Test Statistic	0.031509
p-value	0.100000
Critical Value (10%)	0.347000
Critical Value (5%)	0.463000
Critical Value (2.5%)	0.574000
Critical Value (1%)	0.739000
The series is likely stationary	

Table 4.5: KPSS Test Results of Beverages Sales After Differencing.

4.5.1.4 Forecasting with ARIMA

The Auto-Regressive Integrated Moving Average (ARIMA) model is a widely used statistical model in time series analysis and it is the combination of two simple models such as autoregressive and moving average models. It consists of three components autoregressive (AR) component, integration (I) component, and moving average (MA) component. This model works by modeling the relationship between past values and future values using its three components. The prominent feature of the ARIMA model is it can handle both stationary and non-stationary data with the help of an integration component. It is particularly effective with univariate time series data and any non-stationarity in the data can be removed by differencing. ARIMA is a very effective forecasting model, yet the data preparation and tuning parameters are time-consuming. There are methods available for selecting the parameter values of auto-regressive and moving average components. By carefully selecting the appropriate parameters, the ARIMA model can be used for accurate forecasting, and it serves as a base model for time series analysis.

Components of ARIMA

Autoregressive (AR) component- The autoregressive component of the ARIMA model captures the relationship between a specific value and a certain number of past values in a time series. The past values are also referred to as lagged values in time series analysis. The parameter of the autoregressive component is denoted as p, and it represents the number of lag terms included in the model. The mathematical equation of the AR model of order p is given by **Equation 4.4** [45].

$$Y(t) = c + \phi 1 Y(t-1) + \phi 2 Y(t-2) + ... + \phi p Y(t-p) + \epsilon(t) \tag{4.4}$$

Where:

- $Y(t)$ = Value in the series at time t.
- ϕ = Coefficients of auto-regressive model.
- c = Constant or intercept term.
- ϵ = Error term at time t.

- p = Order of auto-regressive model.

Integrated (I) component- The integrated component of the ARIMA model refers to differencing the time series data to remove the non-stationarity. It removes patterns in data such as trend and seasonality to achieve stationarity and make it suitable for modeling. The integrated component is denoted as d, and it defines the number of times differencing needs to be applied to make data stationary. The mathematical equation of the integrated component is given by **Equation 4.3**.

Moving Average (MA) component- The moving average component of the ARIMA model models the relationship between a value and residual errors of lagged values. It captures the influence of past forecast errors on future values. This component adjusts the prediction values based on previous errors and by considering the noise and random fluctuations in the data. The moving average component is represented as q, and it indicates the number of lagged forecast errors in the model. The mathematical equation of the MA model of order q is given by **Equation 4.5** [45].

$$Y(t) = c + \epsilon(t) + \theta 1\epsilon(t-1) + \theta 2\epsilon(t-2) + ... + \theta q\epsilon(t-q) \tag{4.5}$$

Where:

- $Y(t)$ = Value in the series at time t.
- θ = Coefficients of moving average model.
- c = Constant or intercept term.
- ϵ = Error term at time t.
- q = Order of moving average model.

4.5.1.5 Forecasting with SARIMA

The Seasonal Auto-Regressive Integrated Moving Average (SARIMA) model is one of the basic statistical models used widely in time series analysis to deal with data that contains strong seasonal effects. It is an extension of the autoregressive integrated moving average (ARIMA) model that incorporates seasonality making it an ideal model for forecasting time series data with repeating seasonal patterns at regular intervals.

Unlike the ARIMA model, which is specifically designed to handle non-seasonal data, the SARIMA model is designed to handle seasonal data. The model architecture is the combination of the ARIMA model and additional seasonal components which makes it an effective method to handle complex time series data with seasonality. The SARIMA model is defined as:

$$ARIMA(p, d, q) \times (P, D, Q, s) \tag{4.6}$$

Where:

- p = Order of the non-seasonal auto-regressive component.
- d = Number of times differencing is required to achieve stationarity.
- q = Order of the non-seasonal moving average component.
- P = Order of the seasonal auto-regressive component.
- D = Number of times seasonal differencing is required to achieve stationarity.
- Q = Order of the seasonal moving average component.
- s = Length of the seasonal cycle.

4.5.1.6 Determining Optimal Values of Non-seasonal and Seasonal Components

There are many methods and tools available to find the optimal values of non-seasonal component parameters p, q, and seasonal component parameters P, Q, and s. The most basic and widely used tools are the auto-correlation function (ACF) plot and the partial auto-correlation function (PACF) plot to find the optimal values of non-seasonal parameters q and p. To find the optimal values of seasonal parameters, seasonal auto-correlation function (Seasonal ACF) and seasonal partial auto-correlation function (Seasonal PACF) plots were used.

Although ACF and PACF are effective in finding the optimal values of the AR and MA components, sometimes identifying the values from these plots is complex and the plots are inconclusive due to seasonality, overlapping patterns, data complexity, and noise present in the data. The effective alternative to these plots is finding optimal values by performing a grid search. There are several advantages of using this method like

the possibility to explore a range of parameters and testing multiple combinations of parameters to capture the complex relationships in data.

In this research project, initially, ACF and PACF plots were used to find the optimal value of non-seasonal components of the ARIMA model. But the plots were inconclusive and confusing possibly due to the noise and complex patterns present in the data. Later, an alternative approach manual grid search was implemented to find the suitable values of the parameters that help to forecast future sales accurately.

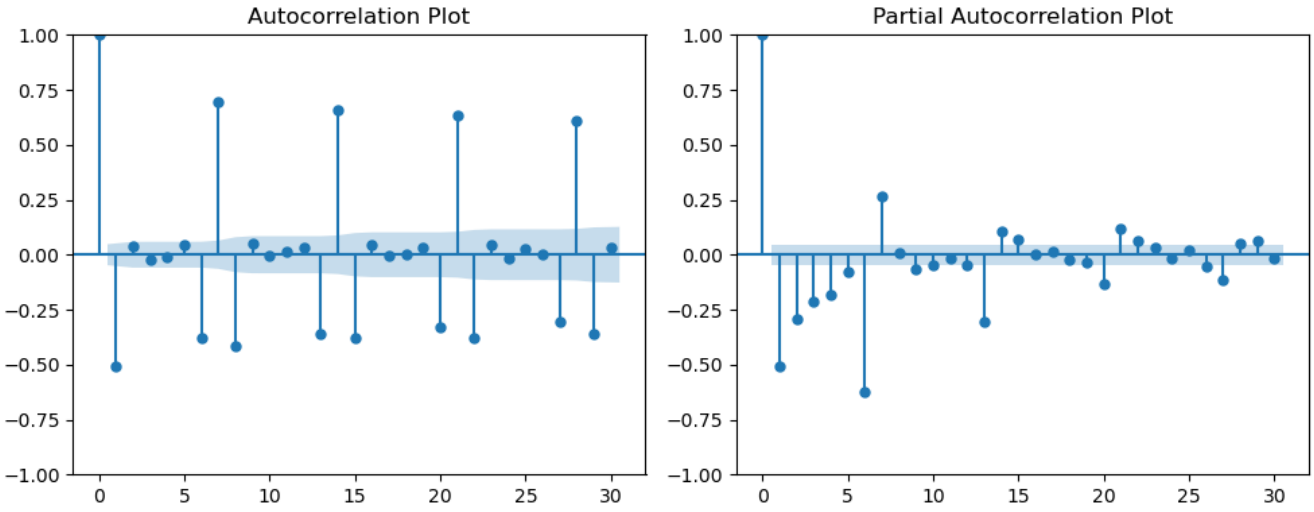

Figure 4.13: ACF and PACF Plots of Beverages Sales Data.

4.5.2 Machine Learning Models

Time series analysis using machine learning is an advanced approach to forecast future data points. It involves providing historical data in sequential order to advanced algorithms and predicting the next values in the sequence. Traditional time series models are capable of capturing only linear and simple patterns in the data and they depend on the assumptions of linearity and stationarity. Unlike these models, Machine learning models are capable of capturing complex and non-linear patterns in data and offer more robust approach for forecasting future values. Machine learning models can handle real-world time series data that often contains patterns like trend and seasonality. They can easily incorporate external features that help in improving forecasting accuracy. These models can be used recursively to perform multi-step forecasting using a recursive time series forecasting technique where the output of the model is given as

input to predict the next value in the sequence. Overall machine learning models can offer more flexibility and tuning hyperparameters helps in improving the forecasting accuracy.

4.5.2.1 Recursive Time Series Forecasting

Recursive time series forecasting is one of the standard methods used in machine learning models to predict future values. It is a technique where predictions are made one step at a time and the predicted value is then used as an input to forecast the subsequent value. Initially, the model is trained based on the available historical data and predicts the next value in the time series data. The forecasted value is then given as an input to the model along with past values to forecast the subsequent value in the series and this process is repeated iteratively. This technique is particularly helpful when forecasting for multiple time steps and it is effective with data where future values depend on the past values. As this approach allows the model to predict the values in series, it is necessary to consider the possibility of increasing error since the accuracy of each prediction depends on the accuracy of previous predictions. The major advantages of this technique are capturing long-term dependencies and flexible for varying forecast horizons.

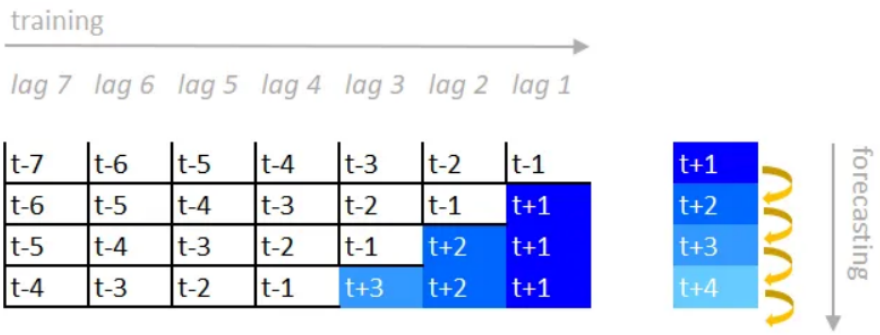

Figure 4.14: Recursive Time Series Forecasting [46].

In recursive forecasting, including external features to the model input can significantly enhance the model performance. External features also known as exogenous variables in data science are additional inputs that provide related information to the models other than past values in time series data. Including these features helps the model to understand the impact of external factors as well instead of depending only on past

values and they assist the model to adapt to varying conditions and trends. These features help in improving forecasting accuracy and making reliable forecasts.

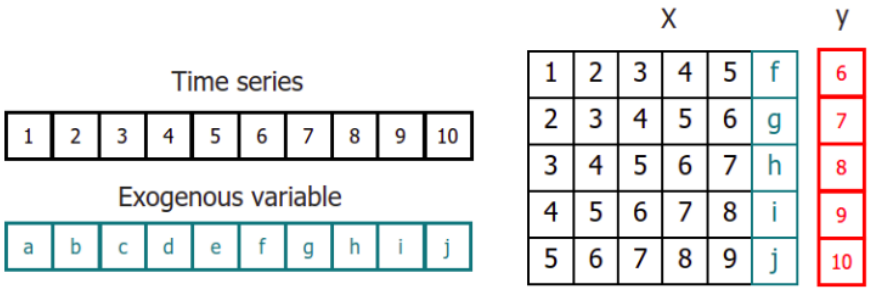

Figure 4.15: Recursive Time Series Forecasting with External Features [47].

4.5.2.2 Hyperparameters

Hyperparameters are the parameters of a machine learning model that control the learning process. In general, the machine learning model has many parameters. They are learned from the training data during the training process. Unlike them, hyperparameters are chosen before the model's training process starts. They are high-level parameters that control the model learning process, and the other parameters learn through this process. Choosing the optimal hyperparameter values can significantly impact the model learning process and make it a reliable model to make accurate predictions on unseen data. Hyperparameter tuning helps in finding optimal values of these crucial parameters. In this research project, fine-tuning of key hyperparameters was performed using a manual grid search method.

4.5.2.3 Forecasting with Random Forest Algorithm

Random Forest is one of the most widely used algorithms in machine learning. It is suitable for classification and regression tasks along with time series analysis tasks. It is a type of ensemble learning and a tree-based algorithm. This algorithm is designed using a group of decision trees and operates based on the outcome of these decision trees. The output of random forest is the aggregate of decision tree results. This approach helps the model to improve accuracy and reduce overfitting.

The important feature of the random forest algorithm is bootstrapping. It is the process of sampling data with replacement. The data is divided into multiple bootstrap samples. Each tree in the model trains on different subsets created using bootstrap sampling

and then predictions are made on unseen data and the output of these models are aggregated as the final result. The Random Forest algorithm is effective in capturing complex and non-linear patterns in data, making it an ideal model for time series forecasting. There are many advantages of using a random forest algorithm including the ability to handle large and complex datasets, robustness to noise, reduced overfitting and it is not sensitive to outliers.

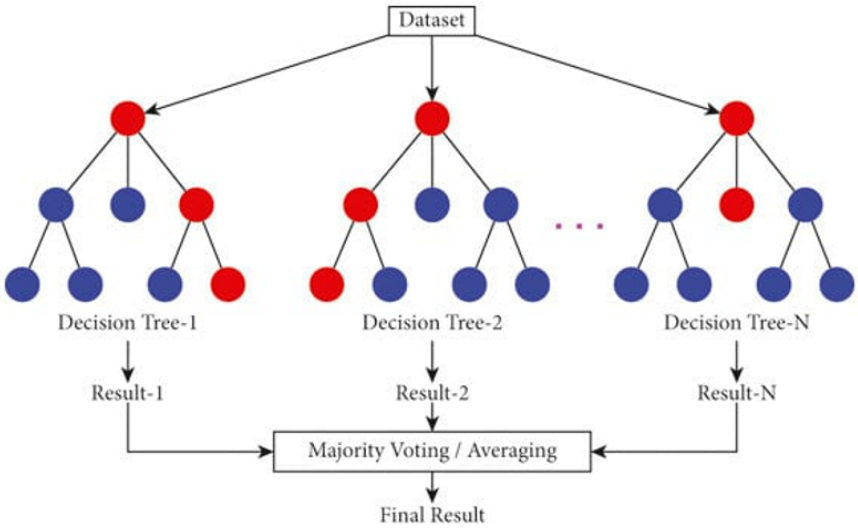

Figure 4.16: Random Forest Algorithm [48].

Hyperparameter tuning plays an important role in improving the model performance. It is the process of selecting an optimal value of a hyperparameter that can significantly improve the training process and generalization of unseen data. The Random Forest algorithm has many hyperparameters that can control the training process. Among them 'n estimators' and 'max depth' plays a crucial role in deciding model complexity. The 'n estimators' parameter indicates the number of decision trees that will be used in the model. In general, more number of trees helps to reduce the variance in the model and leads to improved model performance. The 'max depth' parameter specifies the depth of each decision tree present in the random forest model. The depth of a tree defines the number of splits made from root node to leaf node and it helps in capturing the underlying patterns in the data. A decision tree with more depth can capture the patterns better but it can also lead to overfitting. By selecting optimal depth values, it is possible to reduce the complexity of the trees and speed up the training process.

4.5.2.4 Forecasting with Extreme Gradient Boosting (XGB) Algorithm

Extreme Gradient Boosting is a widely used machine learning algorithm, and it is a type of Gradient Boosting Machine algorithm designed specifically for supervised learning. It is used for classification and regression tasks including time series forecasting. This algorithm belongs to the ensemble learning family and it is a tree-based algorithm. The base estimator in XGB is a decision tree and multiple decision trees are sequentially arranged to improve the prediction accuracy. The basic concept of XGB is to combine multiple weak learners to design a strong model capable of making accurate predictions.

The key concept of the XGB algorithm is boosting and it is an ensemble learning technique. It is the process of combining weak models sequentially to create a strong model. In bagging, models are trained independently and the average of all the models is considered as final output. But, in boosting models are trained in sequence to correct the error made by the previous model. XGB follows a gradient descent optimization process to minimize the loss function. It is used to find the optimal model parameters by reducing the error.

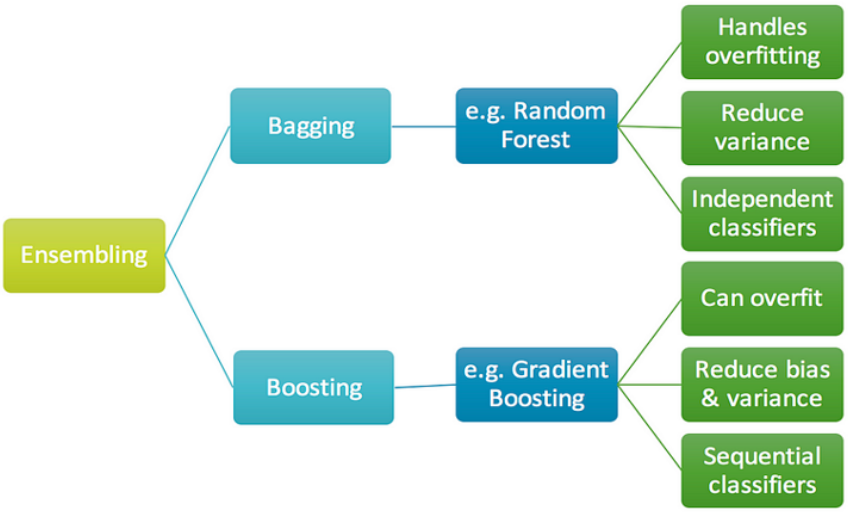

Figure 4.17: Ensemble Learning Techniques [49].

Initially, XGB assigns the same weights to all data points and is given to the first decision tree in sequence. Later, the weights are adjusted based on the misclassifications by increasing the weights of misclassified points and reducing the weights of correctly

classified data points. This approach increases the chances of misclassified points to correctly classified by the subsequent models. The XGB model aims to reduce the classification error of each decision tree by adjusting the weights accordingly and gradually increasing the overall performance of the model. There are many advantages of using XGB including prediction speed, accuracy, feature importance, and handling missing values. However, it has disadvantages such as higher computation time and chances of overfitting.

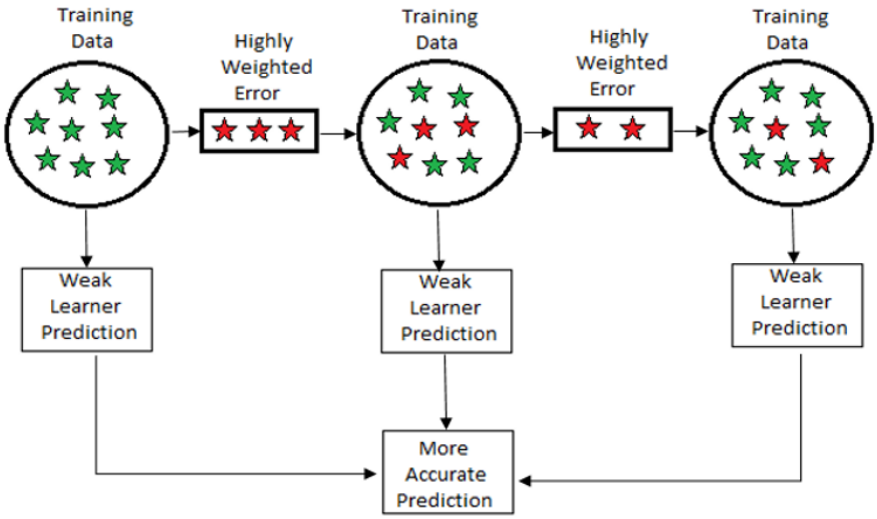

Figure 4.18: XGBoost Algorithm [50].

Machine learning models' performance can be maximized by fine-tuning key hyperparameters. XGB has many hyperparameters that can control the model performance and its complexity. Among them, some of the parameters have a greater influence on the model such as 'n estimators', 'max depth', and 'learning rate'. In this research project, the above-mentioned parameters were fine-tuned to improve model performance. The 'n estimators' parameter specifies the number of decision trees that will be used in the model and the 'max depth' parameter indicates the depth of each decision tree in the XGB model. The 'learning rate' parameter controls the contribution of each tree to the final model. By fine-tuning these parameters, it is possible to maximize the model accuracy and control the complexity of the model.

4.5.2.5 Forecasting with Support Vector Machine (SVM) Algorithm

Support Vector Machine is a supervised machine learning algorithm. It is one of the

most widely used algorithms and it is used for both classification (SVC) and regression (SVR) tasks that include time series forecasting. SVM works based on hyperplane, and it is a fundamental concept of this algorithm. It works differently for classification tasks and regression tasks. In classification tasks, a hyperplane is used to separate different classes. In regression tasks, a hyperplane is used as a prediction curve to predict future values by approximating the relationship between input features and target variables and minimizing the prediction error.

The hyperplane in SVM contains a margin around it and the margin is defined by using the epsilon parameter and it is designed to minimize the error. The data points that fall within the range of this margin are not penalized and the points outside the margin are penalized. The purpose of the margin is to find the hyperplane within range. SVM consists of crucial components called support vectors. The data points that fall on the line of the epsilon margin are support vectors. These points play a key role in positioning the hyperplane.

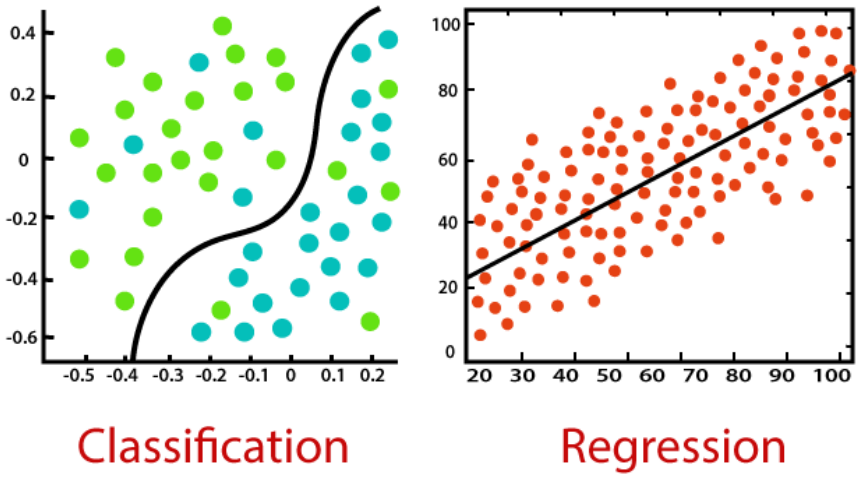

Figure 4.19: Hyperplane in SVC and SVR [51].

The support vector regressor of SVM has the flexibility to work with non-linear kernels and it can deal with noise in the data helping the model to capture the complex and non-linear patterns in the data. SVR can handle both linear and non-linear data and based on the data linear form or non-linear form can be applied. Linear SVR can be used when the relationship in the data is simple and linear. When data has complex patterns, non-linear SVR is used, and it uses the kernel function to transform the data that helps to fit the hyperplane.

The use of kernel functions, hyperplane, and support vectors makes SVM a robust model for time series analysis with an ability to handle non-linear and complex data. It has many advantages including flexibility in dealing with complex data and robustness to noise and outliers. Like any other model, it has some disadvantages as well such as being computationally expensive, difficult to interpret, and complexity in parameter tuning.

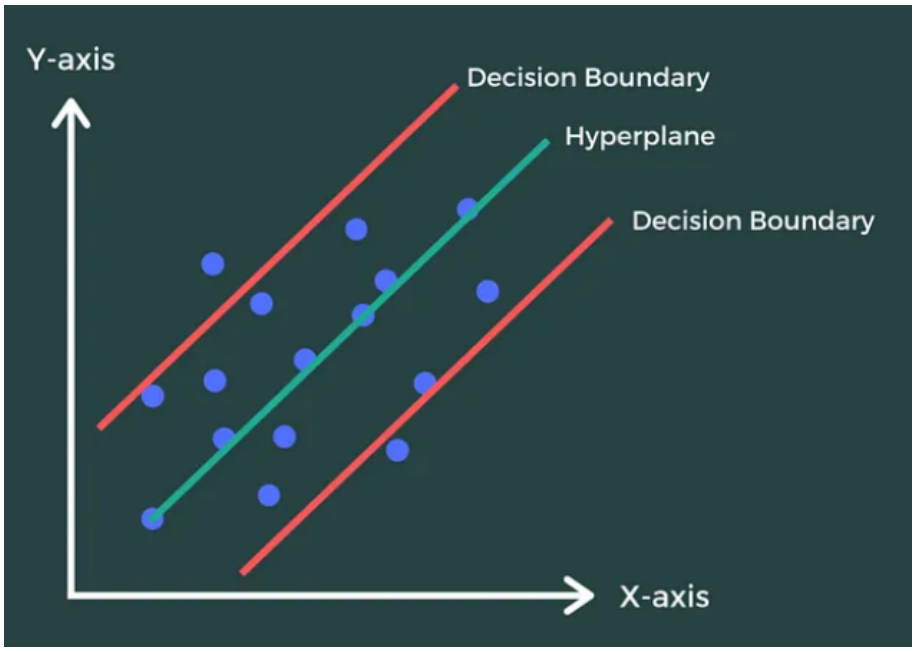

Figure 4.20: Support Vector Regressor (SVR) Algorithm [52].

The performance of any machine learning model can be optimized by tuning key hyperparameters carefully. SVM has some key hyperparameters such as 'C' and 'epsilon'. In this research project, these hyperparameters were tuned properly to maximize the model performance. The parameter 'C' is a regularization parameter, and it balances the model in achieving low training error by fitting training data well and by controlling the complexity of the model. It is necessary to fine-tune the value of C as a higher value results in model overfitting and a lower value leads to model underfitting. The epsilon parameter is used to define the margin around the hyperplane. The errors in predictions like small deviations that lie in this margin are not penalized. The larger epsilon makes the model less sensitive to small deviations in the data and the smaller value of epsilon makes the model to be more accurate with predictions, but the model results might be affected due to the noise present in data. Fine-tuning these parameters helps in balancing the model complexity with the accuracy of the model predictions.

4.5.2.6 Forecasting with FB Prophet Algorithm

FB Prophet also known as Facebook Prophet is one of the most widely used algorithms in time series analysis. It is an open-source tool for forecasting developed by Facebook to deal with time series data. It is specifically designed to learn from data that contains strong seasonality, trends, and non-linear patterns and make future predictions on unseen data. The key features of the prophet model include the ability to handle data with missing values and outliers, and flexibility in modeling non-linear trends. Unlike ARIMA, where stationary data is required to perform analysis FB Prophet can handle the stationarity itself.

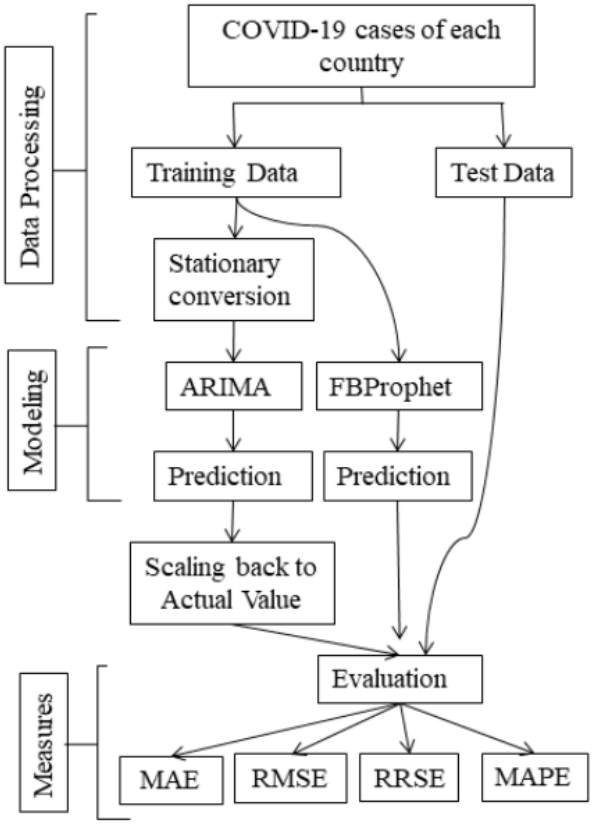

Figure 4.21: Comparison of FB Prophet with ARIMA model [53].

The prophet model is designed based on the additive model and it decomposes the time series data into three key components such as trend, seasonality, and holiday effect components. The long-term change in the data is known as the trend and it is captured by the trend component. The growth in the trend component can be linear or logistic and it can be automatically captured by the prophet model. The seasonal component captures the repeating patterns in the data at regular intervals. The seasonality in the

data can be weekly, monthly, quarterly, or yearly. The holiday effect component of the prophet model allows us to include the holidays or events that might have an impact on the target variable. The decomposable components of the time series data in the FB Prophet model can be mathematically represented as shown in **Equation 4.7** [54].

$$y(t) = g(t) + s(t) + h(t) + e(t) \tag{4.7}$$

Where:

- $g(t)$ = Function that represents non-periodic changes in time series data.
- $s(t)$ = Function that represents seasonal periodic changes in time series data.
- $h(t)$ = Function that represents the effect of events that occur in irregular intervals.
- $e(t)$ = Function that represents the error changes not accommodated by the model.

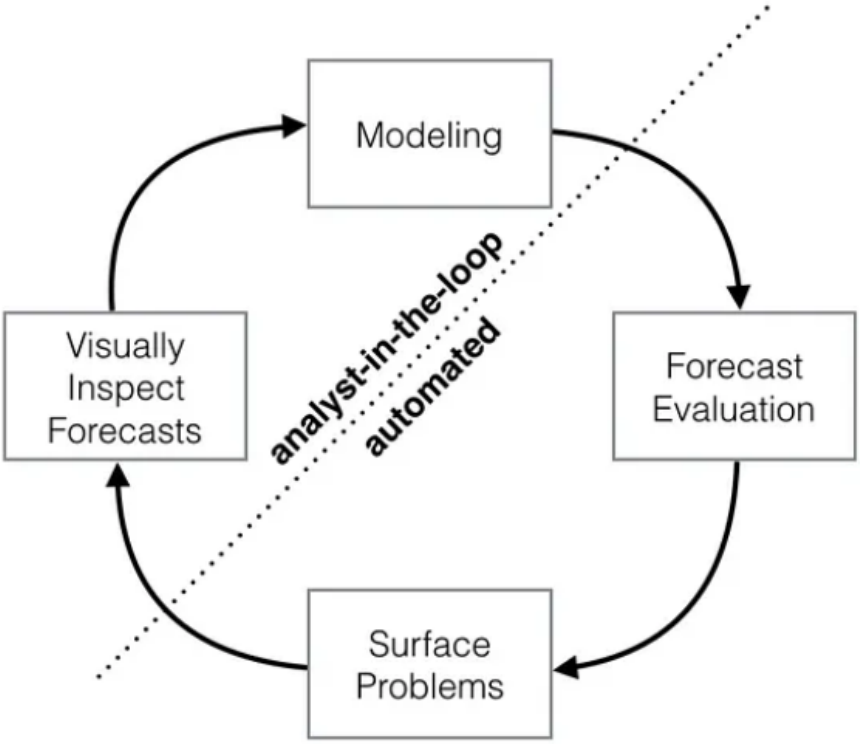

Figure 4.22: Facebook Prophet Model [55].

In general, real-world time series data consists of missing values and outliers. The prophet model is designed to handle these anomalies in data by itself without requiring imputing missing values or filtering outliers explicitly. This model is designed in a user-friendly manner with a simple structure and minimal tuning is enough to achieve

greater forecasting accuracy. The advantage of the FB Prophet model includes easy interpretation of model components, handling non-stationary data, modeling complex seasonal patterns, and the ability to handle small and large datasets. It also has disadvantages such as limited control on hyperparameters and is suitable for only data with strong seasonal effects. Overall, FB Prophet is a powerful tool for time series forecasting provided the data has strong seasonal effects.

4.5.3 Deep Learning Models

Deep learning is a branch of machine learning, and it involves neural networks. Time series forecasting using deep learning is an advanced approach to forecasting that involves forecasting future values based on the past values recorded in sequence over time. Traditional time series models assume data stationarity and linearity and they can capture only linear relationships and simple patterns in the data. Unlike the traditional time series models, the deep learning models can handle non-stationary, complex, and non-linear data. They can capture the complex patterns, non-linear relationships, and dependencies in the data making them valuable tools for time series forecasting. Manual feature engineering is not necessary for these models due to their ability to learn features from raw data. Deep learning models have many hyperparameters and they drive the overall performance of the model. Hyperparameter tuning plays a key role in optimizing the deep learning models and with careful hyperparameter tuning, it has the potential to forecast future values accurately.

4.5.3.1 Forecasting with Long Short-Term Memory (LSTM) Algorithm

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) and it is developed specifically to address the challenges faced when using sequential time series data. RNN networks usually struggle with vanishing gradient problems when training the model which leads to slow learning. LSTM model is particularly designed to address the problem of vanishing gradient by using the memory cells and gating mechanisms. It consists of three gates an input gate, forget fate, and an output gate. These gates and memory cells help the model retain information over long periods and control the flow of information. These features of LSTM make it a crucial model in various applications where the order of the data matters.

The gating mechanism in LSTM networks helps to overcome the challenges faced by RNNs and capture both long-term and short-term dependencies in the sequential data. The input gate of the LSTM model controls the information that should be added to the cell state. It controls the filtering of inputs and storing relevant information. The forget gate of the LSTM model controls which part of the information from the previous step should be removed. It helps in discarding irrelevant information from the prior cell state. The output gate of the LSTM model decides which part of the information should be given as output in the current time step. The output at the current time step has an impact on the next time step in the sequence. These gates of the LSTM model allow the network to maintain long-term dependencies and capture complex patterns.

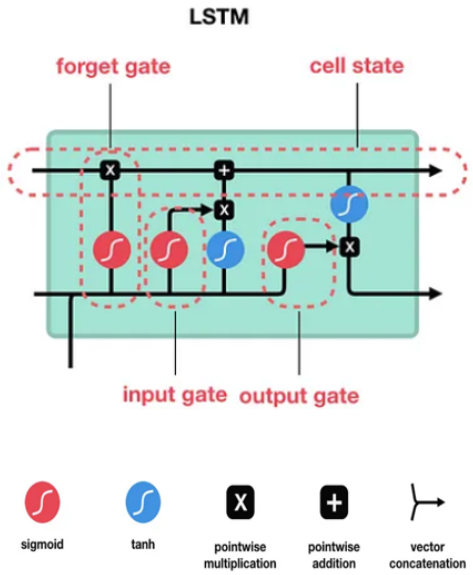

Figure 4.23: Gating Mechanism and Memory Cells in LSTM [56].

The LSTM model is flexible in handling the raw data. Unlike traditional time series models, where the models assume linearity and stationarity, LSTM can handle non-stationary and complex data. It can automatically extract the important features and learn from them without needing to perform feature engineering externally. LSTM can capture the non-linear relationships and underlying patterns in the data by storing information for longer durations. Many advantages of LSTM include capturing long-term dependencies, handling sequential data, automatic feature extraction, and many more. It has disadvantages such as difficulty in fine-tuning hyperparameters, high memory consumption, computationally expensive and it is prone to overfitting. The main drawback of LSTM is that it requires huge amounts of data to effectively uncover

the underlying patterns and temporal dependencies in data during the training process. With access to large data and careful hyperparameter tuning, it is possible to maximize the model performance with time series data.

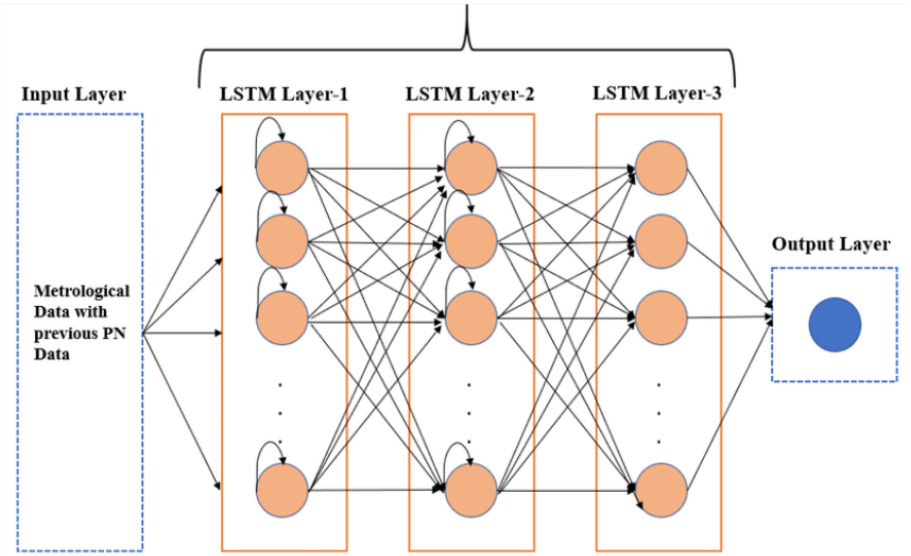

Figure 4.24: Long Short-Term Memory Network Architecture [57].

LSTM Architecture- LSTM architecture is similar to RNNs architecture with a more complex structure. It consists of many layers with each layer having different functionality and all of these layers are connected to handle sequential time series data. Each layer plays a crucial role in building an efficient LSTM model that can learn from sequential data and make reliable predictions on unseen data. The different layers in the LSTM model include-

Input Layer- Input layer is the first layer in LSTM architecture, and it receives the time series data as sequential input. It is usually provided with 3-dimensional data with dimensions related to the number of samples, number of time steps, and number of features. The input layers accept both univariate and multivariate time series data and it prepares and organizes the data into suitable format before providing it to the LSTM layer.

LSTM Layer- The LSTM layer is followed by the input layer, and it is considered the core layer of the LSTM network and sequential processing happens in this layer. The key component of the LSTM layer includes memory cells, and three gates namely

the input gate, forget gate, and the output gate. These components are responsible for the information flow in LSTM networks. Memory cells store important information to capture the long-term dependencies and patterns in the data. The gating mechanism decides which information to store and give as an output to the next time step. The LSTM layer expects to provide a neuron count, and each neuron represents a memory cell.

Dropout Layer- The dropout layer in LSTM is an optional layer but it is a crucial layer in the model's architecture that helps in optimizing the model performance. The LSTM model is prone to overfitting, and it can be controlled by using regularization techniques. Using dropout layers is considered a regularization technique and it helps the model to prevent overfitting during the training process. This layer randomly drops the provided percentage of units during training. This process helps the model's ability to generalize new data and improves the overall performance of the model.

Dense Layer- The dense layer of LSTM is also known as the fully connected layer. This layer transforms the output processed by the LSTM layer into the required output format. The shape of the output depends on the number of neurons provided to the dense layer. It can be either a single value or a sequence of values depending on the data. In this layer linear transformation or SoftMax transformation is applied to the output based on the problem statement and then followed by an activation function to produce the final output.

Output Layer- The output layer in LSTM is the final layer and it provides the final predictions of the model. The shape of the output layer depends on the problem statement. In time series forecasting the output can be either a single value or a sequence of values.

In the research project, a simple LSTM model with an input layer, LSTM layer, dropout layer, dense layer, and output layer is used for forecasting future sales of a retail store based on historical sales data. The dropout layer was used to prevent the model from overfitting and learn the patterns in the data during the training process. Hyperparameter tuning plays a key role in optimizing the LSTM model's performance. Many key hyperparameters of the LSTM model include neuron count in the LSTM layer, training

batch size, learning rate, and activation function. Along with these parameters, the number of lag features was also tuned to find the optimal values for these hyperparameters. Fine-tuning LSTM parameters is computationally expensive and consumes huge memory. With careful hyperparameter tuning, it is possible to achieve greater accuracy and control the complexity of the model.

4.5.4 Evaluation Metrics

In data science research projects, the performance of machine learning, deep learning, and traditional time series models is evaluated using evaluation metrics. These metrics are essential in measuring how well the model predictions match with actual data. The evaluation metrics differ from problem to problem. Classification problem performance is evaluated using different metrics compared to regression problems. There are many metrics available to measure the regression task performance including mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE). These methods are used to measure the accuracy and error magnitude of the model predictions. The R-squared also known as the coefficient of determination is also one of the key evaluation metrics used to measure the model performance in terms of variability. It is always a good practice to use multiple evaluation metrics to compare the model's performance.

4.5.4.1 Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) is one of the most widely used performance evaluation metrics in regression analysis including time series analysis. It is measured by taking the square root of the mean of the squared differences between actual values and predicted values. RMSE is easy to interpret as it provides the direct measure of the error, and it is measured in the same units as the target variable. RMSE is scale-dependent, which means it is affected by the scale of the data. The smaller RMSE error indicates the model has performed better on unseen data.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.8}$$

Where:

- y_i = Actual value.
- $\hat{y}_i$ = Predicted value.
- n = Number of observations in the data.

4.5.4.1 Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error (MAPE) is a widely used performance measurement metric in time series forecasting tasks. It is measured by calculating the percentage of average absolute error between the actual and forecasted values. MAPE is scale-independent which means it can be compared with data with varying scales and it is not influenced by the scales of the data. A lower MAPE suggests the model has generalized better on new data and higher values indicate the huge difference between the actual and forecasted values.

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{4.9}$$

Where:

- y_i = Actual value.
- $\hat{y}_i$ = Predicted value.
- n = Number of observations in the data.

4.5.4.1 R-Squared (R2)

The R-squared is a key evaluation metric to measure the performance of the regression model including time series forecasting tasks. It is also known as the coefficient of determination and measures the model's performance in terms of variance. It indicates the proportion of variance in dependent variables that is explained by independent variables. In general, R2 score ranges between 0 and 1. A value close to 1 indicates better model performance. There is a possibility of getting a negative R2 score and it implies the model's inability to generalize on new data.

CHAPTER 5

Results

This section presents the results of time series forecasting models used in this research project. This study aimed to forecast future sales of selected categories using various models that include traditional time series models such as ARIMA and SARIMA, and machine learning like Random Forest, XGB, SVM, and LSTM a type of recurrent neural network. The main focus is on the performance of various models developed using a specialized time series forecasting technique known as recursive time series forecasting. This method is effective in making multi-step forecasting where the model uses the predictions as input lag features for future forecasting.

In this research project, traditional time series models such as ARIMA and SARIMA were considered as base models and compared the performance of these models with advanced machine learning models. Initially, the stationarity of the data was investigated using multiple methods such as the Augmented Dickey-Fuller (ADF) test, the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test, and the Rolling Statistics test. Then non-stationary data was converted into stationary data to make it suitable for modelling using differencing a data transformation technique. After preparing the data, various models were trained by fine-tuning the hyperparameters.

The models were trained and evaluated separately for each of the three product categories such as beverages, dairy, and grocery. Initially, the models were trained on lag features alone to forecast future sales. Later external features were combined with lag

features and trained the models to predict future values in the sequence. The sales forecasted by the models trained with and without external features were compared to explore the impact of external features on model performance. The results section outlines the performance of each model, evaluated using metrics such as Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and R-squared (R2).

5.1 Traditional Time Series Models

5.1.1 ARIMA

ARIMA model is considered the basic time series model yet effective in forecasting future values in time series analysis tasks. It works on the assumptions of data linearity and data stationarity, and it can forecast future values accurately when the data meets these requirements. Initially, the data stationarity was investigated using the ADF test, KPSS test, and Rolling Statistics test. The results showed that the data of all three categories is non-stationary. Differencing was applied to the data to make it stationary. The stationarity checks were performed again after applying differencing to confirm data stationarity and the tests indicated the data was stationary. One of the research objectives was to investigate the influence of external features on models in forecasting future sales. To carry out the analysis, two ARIMA models were designed where one model works with historical sales data alone to provide lag features and the other model can accommodate the external features along with sales data. The model hyperparameters were fine-tuned using a manual grid search method. The ARIMA model's evaluation metrics are shown below in a table that includes the metrics of the model trained and evaluated with lag features alone and the model with external features and lag features.

Category	**Lag Features**			**Lags + External Features**		
	R2 Score	RMSE	MAPE	R2 Score	RMSE	MAPE
Beverages	0.619	329.62	13.51%	0.623	328.05	14.91%
Dairy	0.562	110.16	16.83%	0.613	103.46	16.10%
Grocery	0.656	383.11	14.74%	0.814	281.83	10.25%

Table 5.1: ARIMA Model Evaluation Metrics.

Firstly, the ARIMA model was trained with historical sales data alone, provided as lag features without incorporating any external features. The future sales were forecasted using the trained model and the performance was measured using evaluation metrics, and the results were shown in **Table** 5.1. Secondly, the ARIMA model was trained and evaluated by incorporating external features such as oil prices, holidays, and promotions along with lag features. The model with external features showed a significant improvement in forecasting future sales and the metrics are shown in **Table** 5.1. ARIMA model's ability to understand the patterns is limited to linear and simple patterns. The model was able to capture the patterns to some extent and it faced difficulty in understanding the underlying complex and non-linear patterns in the data. Among the three categories, the model performed better on grocery data compared to the other two categories with a R2 score of 0.81. The predicted sales of beverages and dairy categories were inaccurate with R2 scores of 0.62 and 0.61 respectively and the forecasted sales of these two categories were not aligned with actual sales. The line plot of actual sales and forecasted sales of the beverages category can be seen in **Figure** 5.1.

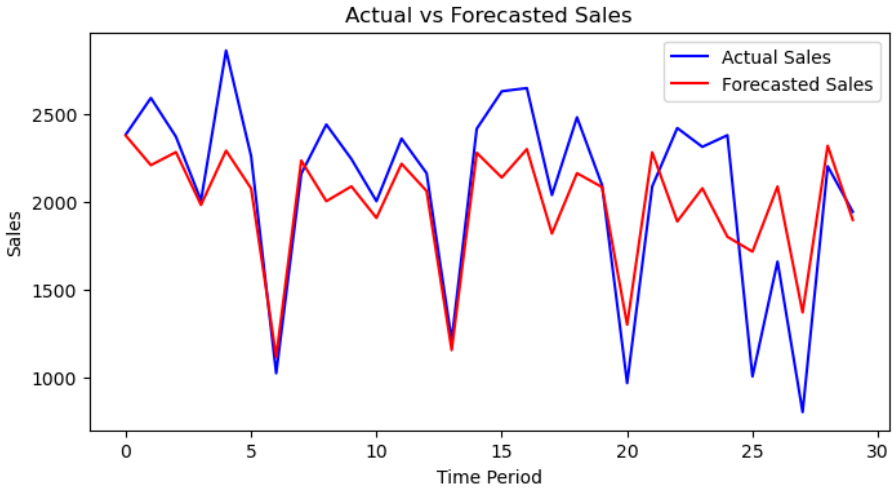

Figure 5.1: ARIMA - Actual vs Forecasted Sales of Beverages.

5.1.2 SARIMA

The SARIMA model is the extension of the ARIMA model, and a seasonal component is included in its architecture to handle the data with strong seasonal effects. At first, the historical sales data of three categories was examined using the seasonal

decomposition technique. This method separates the data into trend, seasonality, and residual components. It was observed from the seasonal decomposition plots that the sales data has strong seasonality and trends. Seasonal differencing of order one was performed to deal with seasonality in the data. Two SARIMA models were designed to investigate the impact of external features, one model to train with lag features alone and another model to accommodate external features as well. The hyperparameters of the models were finetuned to achieve the optimal performance. The evaluation metrics of the models trained and evaluated with the data of all three categories are shown below.

Category	**Lag Features**			**Lags + External Features**		
	R2 Score	RMSE	MAPE	R2 Score	RMSE	MAPE
Beverages	0.647	317.39	11.41%	0.784	248.0	10.40%
Dairy	0.721	87.82	10.17%	0.745	84.03	11.39%
Grocery	0.70	353.19	12.78%	0.832	267.47	10.82%

Table 5.2: SARIMA Model Evaluation Metrics.

Initially, the SARIMA model was trained and evaluated with sales data of each of the three categories, provided as lag features without including external features. The model performance was measured using evaluation metrics and the results were displayed in **Table 5.2**. To examine the impact of external features on model performance, the SARIMA model designed to include external features along with lag features was trained and the forecasted sales were compared to actual sales to measure model performance. The results in **Table 5.2** clearly show that the external features have an impact on forecasted sales. They also indicate that the sales data consists of key patterns and the model was able to recognize those patterns with the help of external features. Therefore, the analysis showed that external features played a crucial role in improving forecasting accuracy by learning the underlying patterns in the data.

Unlike the ARIMA model, which performed better with sales data related to the grocery category alone, the SARIMA model has performed better with data related to all three categories. It effectively captured the seasonal patterns, trends, and underlying relationships in the data. The performance of SARIMA was consistent across all three categories with higher R2 scores and lower errors. The R2 scores achieved by the

SARIMA model with sales data of beverages, dairy, and grocery categories are 0.78, 0.74, and 0.83 respectively. The results show the model's strength in handling seasonality and variations within each category. The graph of actual sales against the forecasted sales of the beverages category can be seen in **Figure 5.2** and it shows the model's ability to capture the patterns to some extent. However, there is a scope for improving forecasting accuracy using advanced machine learning algorithms due to their ability to find complex and non-linear patterns in the data.

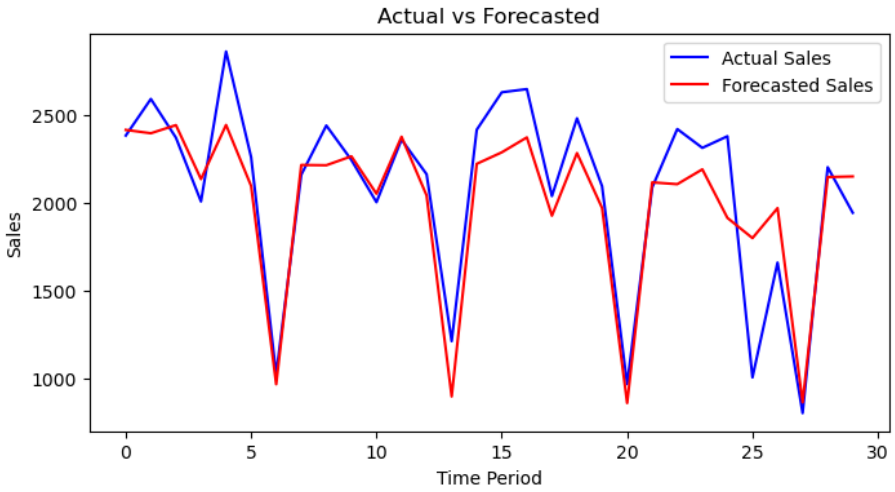

Figure 5.2: SARIMA - Actual vs Forecasted Sales of Beverages.

5.2 Impact of External Features

The time series analysis performed using ARIMA and SARIMA models indicated that incorporating external features along with lag features played a crucial role in enhancing the model performance. This behavior completely depends on the nature of the data. The intricate patterns present in the historical sales data used in this research project can be learned by the models with the help of external features. Therefore, it was decided to carry out further analysis that includes forecasting using advanced machine learning and deep learning algorithms by including the external features along with lag features instead of using lag features alone.

5.3 Machine Learning and Deep Learning Models

In this research project, machine learning algorithms such as Random Forest, Extreme Gradient Boosting (XGB), Support Vector Machine (SVM), a specialized time series model Facebook Prophet model, and deep learning model Long Short-Term Memory (LSTM) were employed using recursive time series forecasting an advanced time series forecasting technique to forecast the sales of a superstore for three selected categories such as beverages, dairy and grocery. The previous analysis performed using ARIMA and SARIMA models showed that the external features have a greater impact on the model's forecasting accuracy. Therefore, it was decided to include external features along with lag features while training and evaluating machine learning and deep learning models instead of using lag features alone.

5.3.1 Random Forest Algorithm

Random Forest Algorithm is an ensemble learning and a tree-based algorithm. It is a group of decision trees connected to form a random forecast algorithm and the result of this model is the aggregate of decision trees. In this research project, the model was designed using an advanced forecasting technique known as recursive time series forecasting. This technique helps the model to do multi-step forecasting. The model was trained and evaluated using the data of all three categories by providing historical sales data and external features such as oil prices, promotions, holidays, and events to the model. The sales data was provided to the model in the form of lag features and the number of lag features was decided by performing a grid search. The model hyperparameters were fine-tuned using the manual hyperparameter tuning method. The performance of the model was measured using evaluation metrics and the values are provided in the **Table 5.3**.

Category/Family	RF Evaluation Metrics		
	R2 Score	RMSE	MAPE
Beverages	0.829	220.83	10.13%
Dairy	0.747	83.68	11.23%
Grocery	0.841	260.55	8.34%

Table 5.3: Random Forest Model Evaluation Metrics.

5.3.1.1 Beverages

The random forest model trained and evaluated using sales data of beverages, performed steadily in forecasting future sales with a R2 score of 0.82 indicating the model was able to explain 82 percent of variance in the target variable. Additionally, the model's RMSE was 220.83 and MAPE was 10.13. It indicates that there was around a 10 percent error between the actual sales and the predicted sales. The model's high R2 score and high RMSE error imply that the model was able to capture patterns in the data effectively but the errors in the model are significant. The plot of actual sales against the predicted sales can be seen **Figure** 5.3. The graph indicates that the predicted sales of the beverages category closely followed the trend of actual sales.

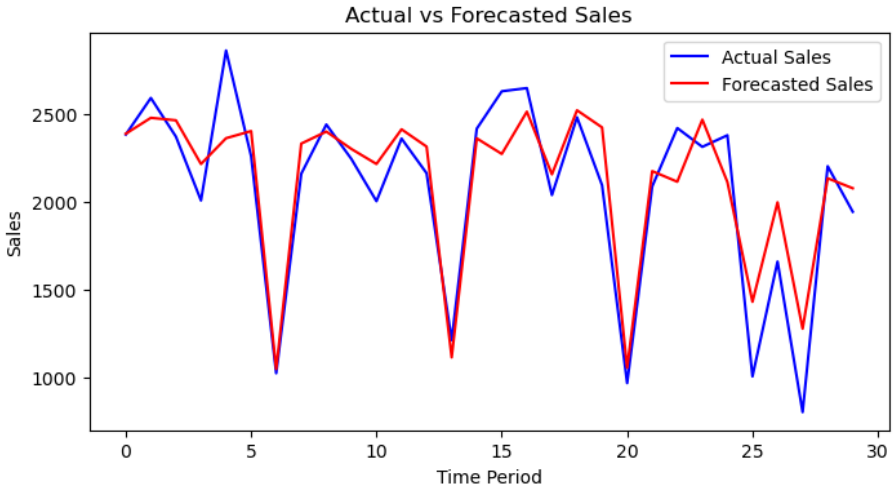

Figure 5.3: Random Forest - Actual vs Forecasted Sales of Beverages.

5.3.1.2 Dairy

The analysis carried out with the Random Forest algorithm using the sales data related to the dairy category showed moderate performance in forecasting future sales. The

R2 score of the model was around 0.74 implying that 74 percent of the variance in the target variable was explained by the model with the help of provided input features. The average R2 score represents that the model struggled to accurately identify the patterns in data. The error between the actual sales and predicted sales was measured using RMSE and MAPE and the values of these metrics are 83.68 and 11.23 respectively. Even though the R2 score of the model is less, the small RMSE value indicates there isn't much difference between the predicted values and actual values. **Figure 5.4** shows the comparison between actual sales and predicted sales of the diary category, and it demonstrates the extent to which the model was able to capture the patterns in the sales data.

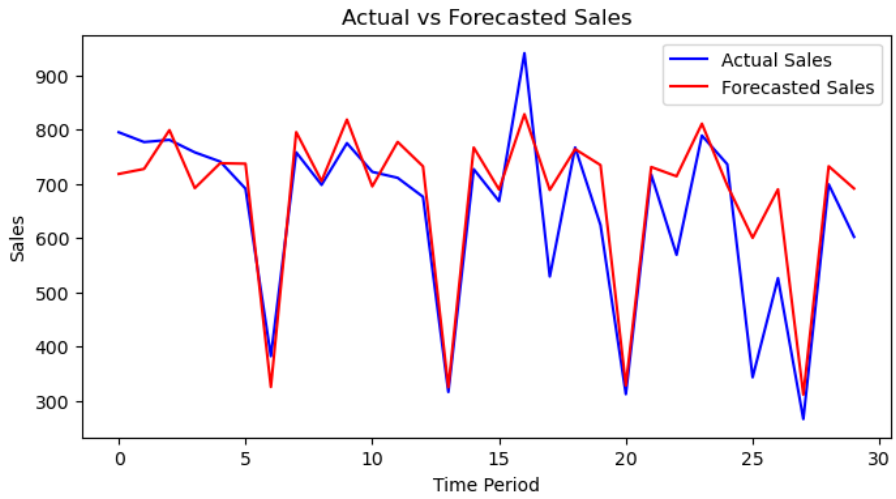

Figure 5.4: Random Forest - Actual vs Forecasted Sales of Dairy.

5.3.1.3 Grocery

The Random Forest model trained and evaluated using the time series sales data related to the grocery category performed better compared to the models that used sales data of the other two categories. The model achieved an R2 score of 0.84 suggesting that the model was able to explain 84 percent of variance in the target variable. Furthermore, the errors in model predictions were measured in terms of RMSE and MAPE and the values of these metrics are 260.55 and 8.34 respectively. The high R2 score of the models indicates the model's ability to find the patterns in the data and the high RMSE error suggests that the errors in the predicted sales are significant. Overall, the results suggest that the model was able to capture the patterns and might struggle sometimes due to extreme values in the sales data. The plot of actual sales against the predicted sales

of the grocery category can be seen in the **Figure** 5.5 and the plot shows how well the predicted sales are aligned with actual sales.

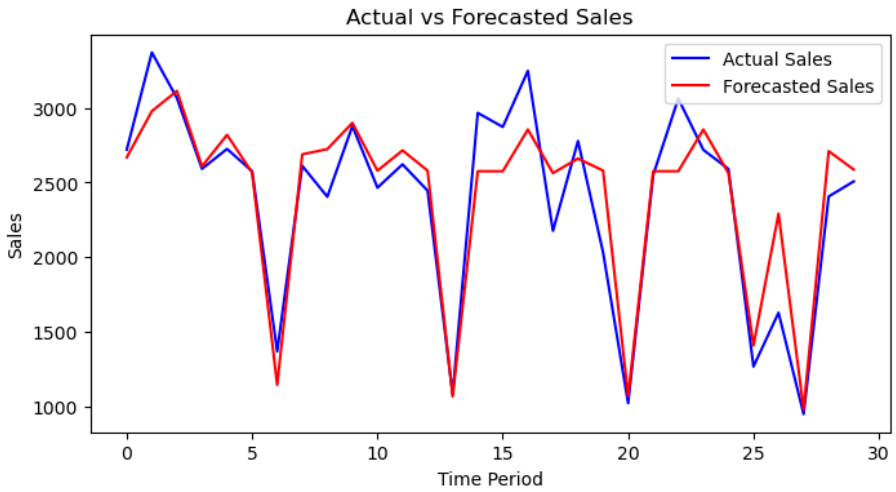

Figure 5.5: Random Forest - Actual vs Forecasted Sales of Grocery.

5.3.2 XGBoost (XGB) Algorithm

XGBoost algorithm also known as Extreme Gradient Boosting Machines belongs to the ensemble learning family and it is a tree-based algorithm. It is an enhanced implementation of Gradient Boosting Machines. XGB combines the results of weak learning to make strong and accurate predictions. The model consists of a group of decision trees connected sequentially and each tree tries to correct the error of the previous model. In this research project, the model architecture is designed using recursive time series forecasting to perform multi-step forecasting of future sales. The time series analysis was carried out to forecast the future sales of all three categories such as beverages, dairy, and grocery. The historical sales data was provided as lag features along with the external features and the optimal value of lag features was found out using grid search. The model was trained and forecasted using sales data of each category separately. The hyperparameters of the XGB model were fine-tuned using a manual grid search method. The performance of the model was measured using evaluation metrics and provided in the **Table** 5.4.

Category/Family	XGB Evaluation Metrics		
	R2 Score	RMSE	MAPE
Beverages	0.860	199.36	8.39%
Dairy	0.819	70.65	9.89%
Grocery	0.825	272.89	9.28%

Table 5.4: XGBoost Model Evaluation Metrics.

5.3.2.1 Beverages

The XGBoost model trained and evaluated using sales data related to the beverages category has performed better in forecasting future sales by understanding the patterns and relationships in the data. The model's forecasted sales were compared with the actual sales and achieved an R2 score of 0.86 indicating the model was able to explain 86 percent of the variance in the target variable with the help of lag features and external features. The difference between the forecasted sales and actual sales measured in terms of RMSE and MAPE and the values of these metrics are 199.36 and 8.39 respectively. The MAPE error of 8.39 indicates that the model predictions are away from actual sales by around 8 percent. This is usually considered a good result in time series forecasting. The small error also indicates the model was able to capture the patterns in the data precisely. **Figure 5.6** shows the grocery category actual sales plotted against the predicted sales and the plot shows how closely the model predictions aligned with actual sales.

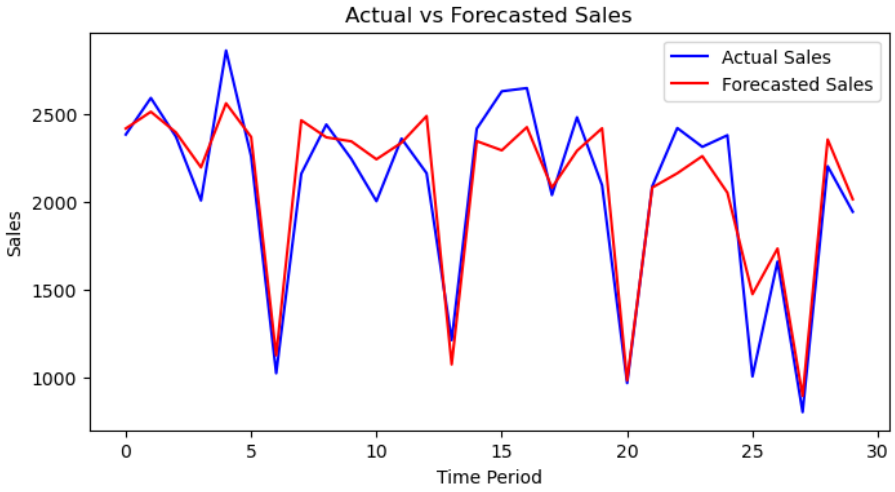

Figure 5.6: XGBoost - Actual vs Forecasted Sales of Beverages.

5.3.2.2 Dairy

The time series analysis carried out with the XGBoost algorithm to forecast future sales using sales data related to the dairy category has forecasted future sales precisely with a decent R2 score of 0.81 and it implies that 81 percent of the variance in sales data was explained by the model. The high R2 score indicates that the model was able to capture the complex patterns and relationships within the data. Furthermore, the difference between the actual sales and forecasted sales was measured and the RMSE of the model was 70.65 and MAPE was 9.89. The low RMSE error of the model indicates that there isn't much difference between the actual and forecasted sales. It also implies that the model was able to capture the extreme values in the data efficiently and the MAPE error is also considerable. Overall, the model has performed with the sales data related to the dairy category. The line plot of predicted sales plotted against actual sales can be seen in **Figure** 5.7.

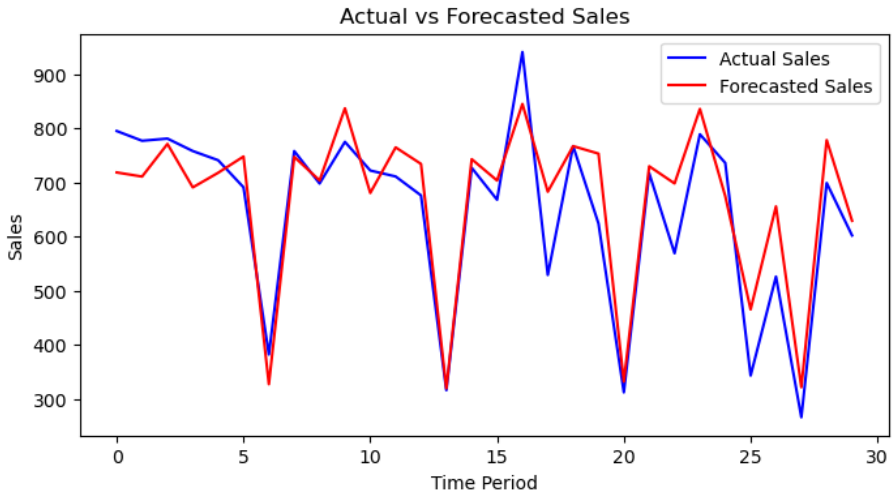

Figure 5.7: XGBoost - Actual vs Forecasted Sales of Dairy.

5.3.2.3 Grocery

The XGBoost model was trained using sales data related to the grocery category to forecast future sales. The model showed solid performance with an R2 score of 0.82 indicating that 82 percent of the variance in the target variable was explained by the model. The error in the predicted sales was measured using metrics such as RMSE and MAPE. The RMSE error is 272.89 and it indicates the average deviation of predictions from the actual sales. The MAPE of the model is 9.28 percent and it gives the average

percentage of error in predicted sales. The model with a high R2 score and high RMSE error implies that the data has extreme values, and the model was not able to capture them efficiently. This behavior is expected when dealing with time series data. **Figure** 5.8 shows the grocery category predicted sales plotted against the actual sales.

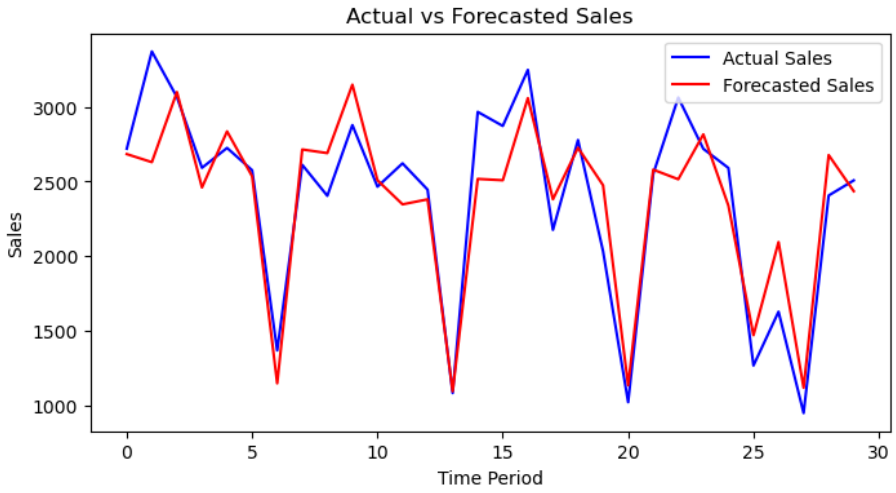

Figure 5.8: XGBoost - Actual vs Forecasted Sales of Grocery.

5.3.3 Support Vector Machine (SVM) Algorithm

Support Vector Machine is a power and one of the most widely used supervised machine learning algorithms. It works based on a hyperplane and in regression analysis that includes time series forecasting the hyperplane works as a prediction curve to predict the future values in the time series. It models the relationship between input features and the target variable to forecast the values. In this research project, the SVM model was designed using an advanced forecasting technique known as recursive time series forecasting. It helps in performing multi-step forecasting of future sales. The time series analysis was carried out using the sales data related to three categories namely beverages, daily, and grocery. The designed model was trained with historical sales data provided as lag features along with external features. The optimal number of features was decided by performing a manual grid search. The analysis was carried out separately for each of the three categories and the hyperparameters of the model were fine-tuned using the manual hyperparameter tuning method. The model performance was measured using selected evaluation metrics such as R2 score, RMSE, and MAPE, and the values of these metrics are provided in **Table** 5.5.

Category/Family	SVM Evaluation Metrics		
	R2 Score	RMSE	MAPE
Beverages	0.853	204.45	9.10%
Dairy	0.821	70.44	9.12%
Grocery	0.848	254.59	8.95%

Table 5.5: SVM Model Evaluation Metrics.

5.3.3.1 Beverages

The SVM algorithm trained and evaluated using sales data related to the beverages category has performed better by accurately forecasting future sales. The evaluation metrics of the model such as R2 score, RMSE, and MAPE are 0.85, 204.45, and 9.10 respectively. The R2 score of 0.85 indicates that 85 percent of the variance in the target variable was explained by the model. The RMSE error of the model is comparatively less and MAPE implies that the average error in predicted sales was around 9 percent. The high R2 score indicates the model's ability to capture the patterns in the sales data. **Figure** 5.9 shows the actual sales plotted against the forecasted sales and the plot shows how accurately the predicted sales aligned with the actual sales.

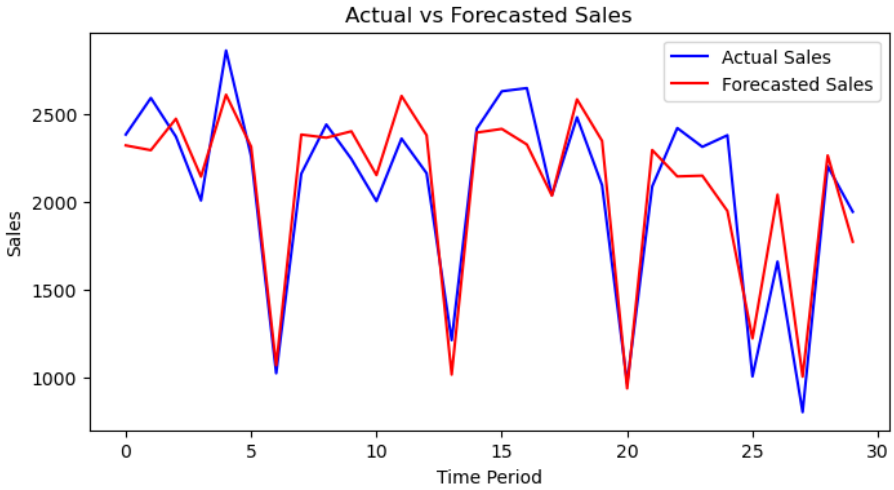

Figure 5.9: SVM - Actual vs Forecasted Sales of Beverages.

5.3.3.2 Dairy

The time series analysis was carried out using the SVM algorithm to forecast future sales using sales data related to the dairy category. The model has shown great performance

in predicting future values by understanding the complex relationships within the sales data. The model predictions were compared to actual sales and measured the model performance. It has achieved a R2 score of 0.82 suggesting that the model was able to explain 82 percent of the variance in the target variable. The difference between the predicted sales and actual sales was measured and the values of the metrics RMSE and MAPE are 70.44 and 9.12 respectively. The small RMSE values suggest that the model's predictions were closely aligned with the actual sales. The line plot of actual sales plotted against the predicted sales can be seen in the **Figure** 5.10.

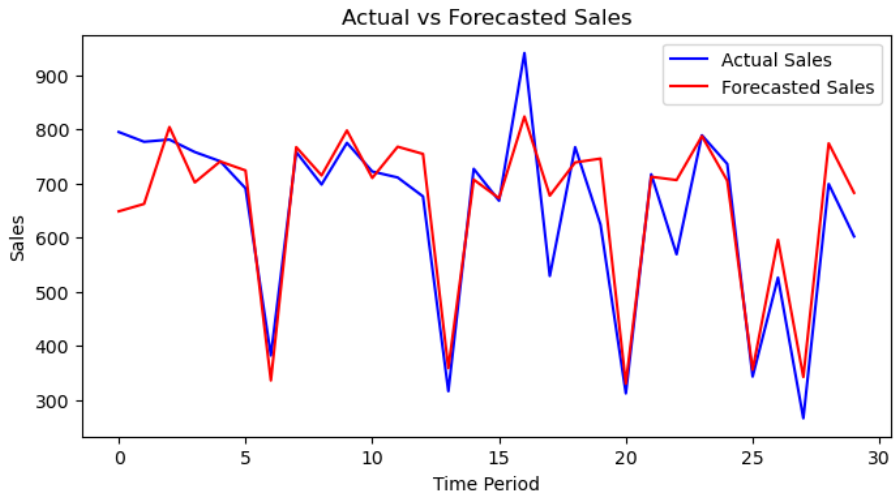

Figure 5.10: SVM - Actual vs Forecasted Sales of Dairy.

5.3.3.3 Grocery

The future sales of the grocery category were forecasted using the SVM algorithm by training the model with provided historical sales data and external features. The model performance was measured, and the evaluation metrics indicated that the model was effective in forecasting sales. The R2 score is 0.84 and it suggests that the model explained 84 percent of the variance in the sales data. The errors in predictions were measured using RMSE and MAPE and the values of these metrics are 254.59 and 8.95 respectively. The RMSE value is comparatively higher, and it indicates that the model was able to capture the patterns in the data, but it struggled with extreme values in the time series data. The small MAPE error suggests that the model has forecasted sales accurately with an average error percentage of around 9 percent. **Figure** 5.11 shows the actual sales plotted against the forecasted sales.

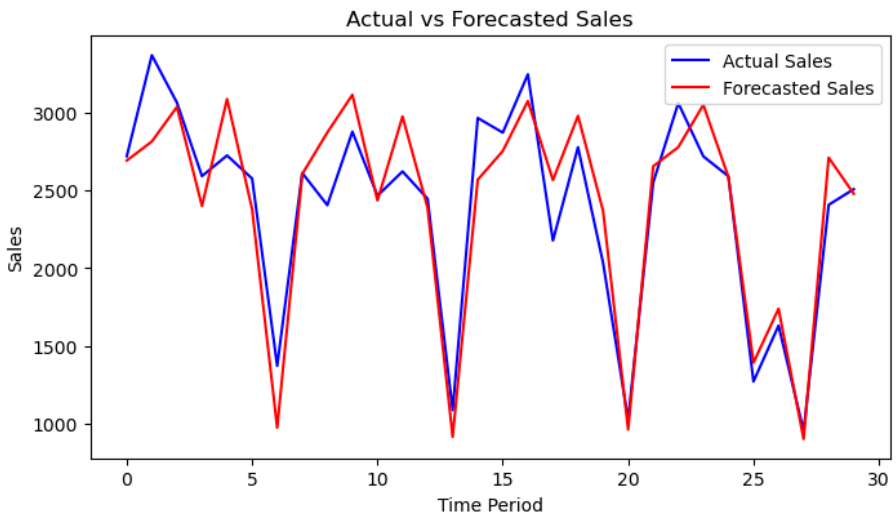

Figure 5.11: SVM - Actual vs Forecasted Sales of Grocery.

5.3.4 Facebook Prophet Model

FB Prophet also known as the Facebook Prophet model, is an open-source tool developed by Facebook to handle time series analysis tasks. It is particularly designed to deal with data that has strong seasonal effects. The model architecture is simple and there isn't much scope for tuning parameters. It can handle missing values, outliers, and non-linear patterns in the data. In this research project, the FB Prophet model was designed with simple architecture to deal with time series data containing seasonal effects. The model was trained with data that includes external features and model performance was measured using evaluation metrics such as R2 score, RMSE, and MAPE. The sales data related to three categories namely beverages, dairy, and grocery was used separately to carry out the analysis. The evaluation metrics of the model are provided in **Table 5.6**.

Category/Family	**FB Prophet Evaluation Metrics**		
	R2 Score	**RMSE**	**MAPE**
Beverages	0.694	295.62	14.09%
Dairy	0.746	83.80	11.07%
Grocery	0.734	336.81	12.28%

Table 5.6: FB Prophet Model Evaluation Metrics.

5.3.4.1 Beverages

The FB Prophet model trained and evaluated using sales data related beverages category has shown average performance with an R2 score of 0.69 indicating the model was able to explain only 69 percent of the variance in the target variable. The low R2 score indicates the model was not able to capture the patterns in the data effectively. To further investigate the results the difference between the predicted sales and actual sales was measured in terms of RMSE and MAPE and the values of these metrics are 295.62 and 14.09 respectively. The errors in the predicted sales are quite high which implies the model was unable to capture extreme points and outliers in the data. **Figure 5.12** shows the actual sales plotted against the predicted sales and it can be seen from the plot that predicted sales are deviated from the actual sales.

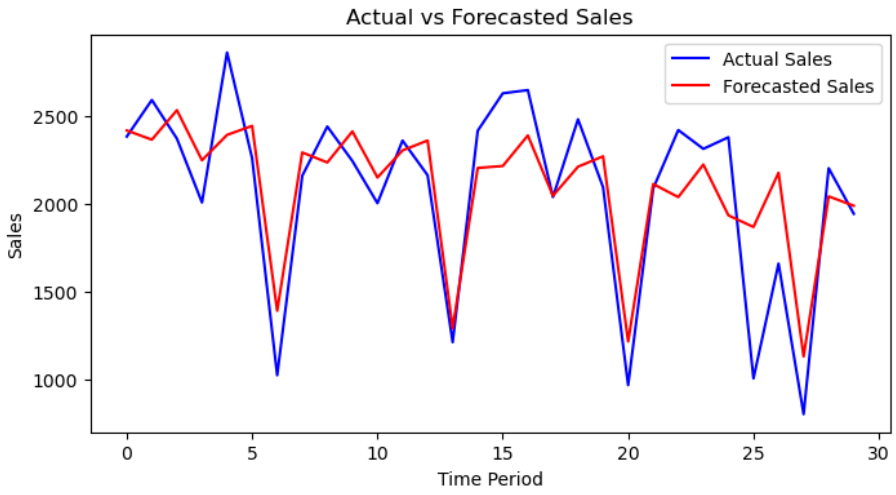

Figure 5.12: FB Prophet - Actual vs Forecasted Sales of Beverages.

5.3.4.2 Dairy

The time series forecasting carried out with the FB Prophet model using sales data related to the dairy category has performed moderately in forecasting future sales. The predictions were evaluated against the actual sales and the model achieved an R2 score of 0.74, RMSE of 83.8, and MAPE of 11.07. The R2 score implies that 74 percent of the total variance in the sales data was explained by the model. The low R2 score and low RMSE indicate that the model was not able to capture the patterns in the data effectively but there isn't much difference between predicted and actual sales. The line plot of actual sales plotted against predicted sales can be seen in **Figure 5.13**.

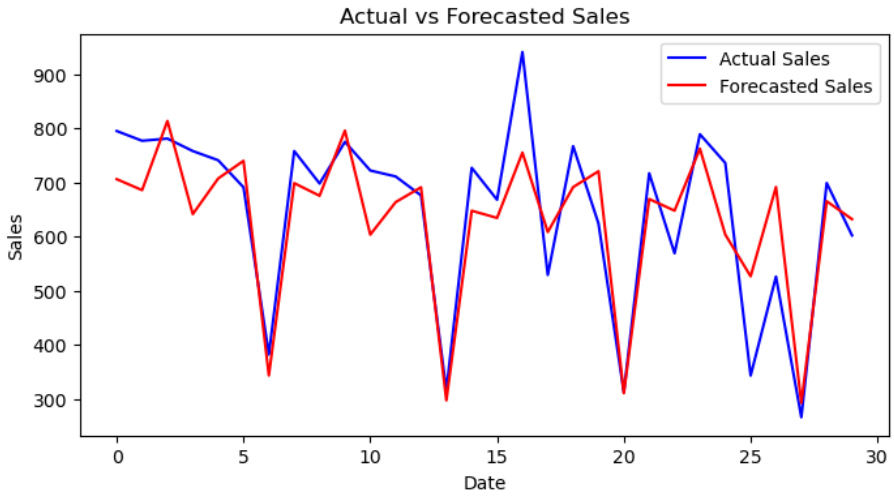

Figure 5.13: FB Prophet - Actual vs Forecasted Sales of Dairy.

5.3.4.3 Grocery

The FB Prophet model was trained using the time series sales data related to the grocery category to forecast future sales. The forecasted sales were evaluated by comparing them with actual sales and the model achieved a R2 score of 0.73. It indicates that 73 percent of the variance in the target variable was explained by the model with the help of provided input features. The RMSE error of the model is 336.81 and MAPE is 12.28. The high RMSE error indicates the difference between actual sales and predicted sales is large and the model was not able to capture the extreme points in the sales data. **Figure** 5.14 shows the actual sales plotted against the forecasted sales.

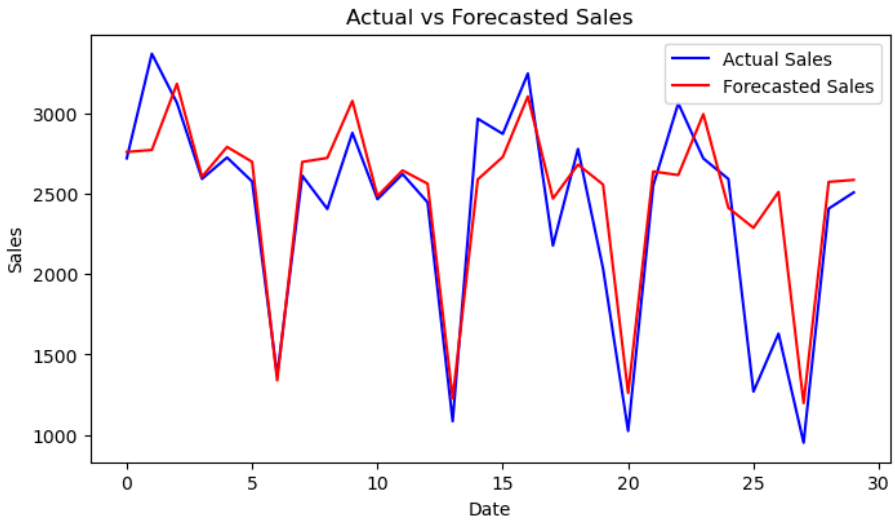

Figure 5.14: FB Prophet - Actual vs Forecasted Sales of Grocery.

5.3.5 Long Short-Term Memory (LSTM) Algorithm

Long Short-Term Memory is a type of Recurrent Neural Network, and it is widely used in time series analysis tasks. The gating mechanism and memory cells present in the model help to avoid the vanishing gradient problem and increase the learning speed during the training process. LSTM model can handle complex and non-linear data by modeling the relationships with the help of input features. In this research project, a simple LSTM model was designed using an advanced forecasting technique known as recursive time series forecasting to perform multi-step forecasting. The model architecture consists of an LSTM layer, a dropout layer, and a dense layer. The analysis was carried out using sales data of three selected categories from a superstore. The model was trained using each of the three categories separately and the performance of the models was measured using evaluation metrics such as R2 score, RMSE, and MAPE. The values of these evaluation metrics can be seen in **Table 5.7**.

Category/Family	**LSTM Evaluation Metrics**		
	R2 Score	**RMSE**	**MAPE**
Beverages	0.660	311.45	12.69%
Dairy	0.699	91.33	10.76%
Grocery	0.700	357.94	12.50%

Table 5.7: LSTM Model Evaluation Metrics.

5.3.5.1 Beverages

The LSTM model was trained and evaluated using sales data related to the beverages category and the model performance was average in forecasting future sales. The model predictions were evaluated and the R2 score of the model is 0.66 indicating that 66 percent of the variance in the target variable was explained by the model. The error in forecasted sales was measured using RMSE and MAPE; these metrics' values are 311.45 and 12.28 respectively. The RMSE error of the model is quite high, and it implies a huge difference between actual and forecasted sales. **Figure 5.15** shows the line plot of actual sales and forecasted sales of the beverages category.

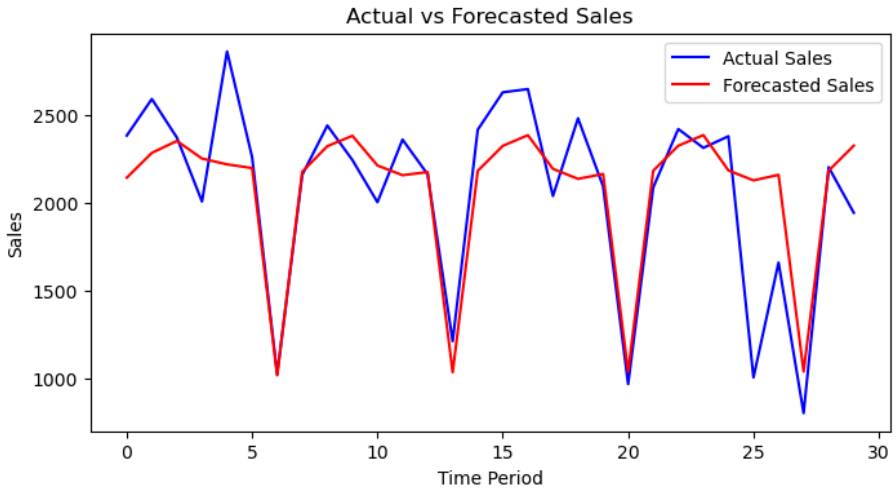

Figure 5.15: LSTM - Actual vs Forecasted Sales of Beverages.

5.3.5.2 Dairy

The time series analysis carried out with the LSTM model using sales data related to the dairy category has shown moderate performance with an R2 score of 0.69. The R2 score is considerably low indicating the model was unable to capture the underlying patterns effectively. The difference between the actual sales and forecasted sales was measured in terms of RMSE and MAPE and the value of these metrics are 91.33 and 10.76. The RMSE error of the model is small suggesting that there isn't much difference between the predicted sales and actual sales. The line plot in **Figure 5.16** shows how the forecasted sales are aligned with the actual sales.

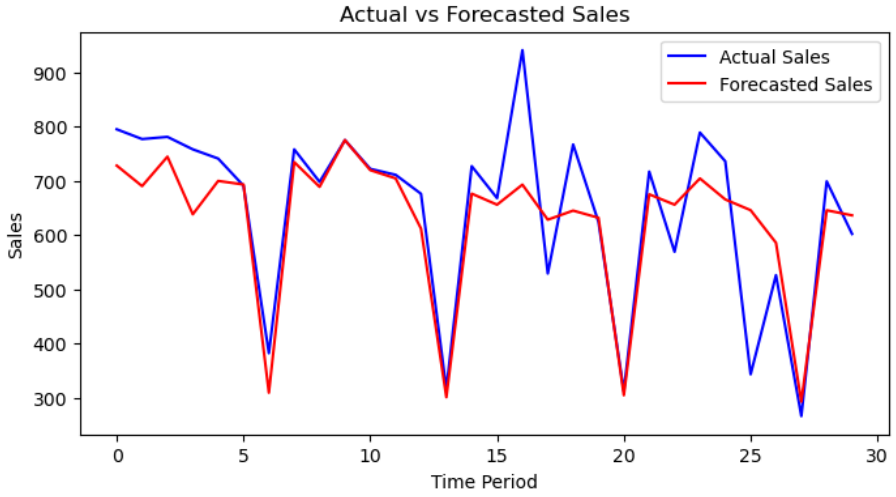

Figure 5.16: LSTM - Actual vs Forecasted Sales of Dairy.

5.3.5.3 Grocery

The LSTM model was trained using sales data related to the grocery category and forecasted future sales in the time series. The model performance was measured using evaluation metrics by comparing the predicted sales with actual sales. The R2 score of the model is 0.70 indicating 70 percent of variance in the sales data was explained by the model. The error in predicted sales was measured, and the RMSE error is 357.94 and the MAPE error is 12.5 percent. The high RMSE error implies that sales data has some extreme values, and the model was unable to capture those points effectively. **Figure** 5.16 shows the line plot of actual sales plotted against the forecasted sales of the LSTM model.

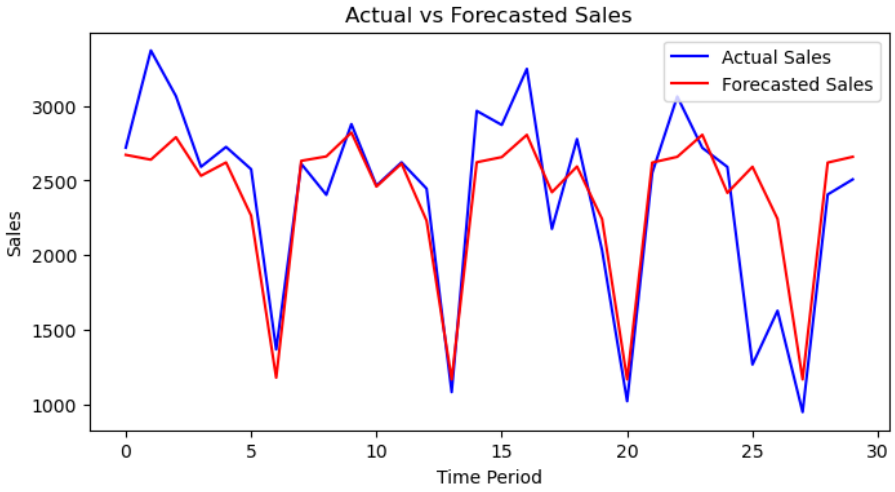

Figure 5.17: LSTM - Actual vs Forecasted Sales of Grocery.

CHAPTER 6

Discussion

The primary objective of this research project is to explore various time series models to forecast the future sales of a superstore and find an effective time series forecasting model. To perform a deeper exploration of patterns in sales data that allow accurate forecasting, three categories were selected from a superstore to carry out the analysis. Various statistical, machine learning and deep learning algorithms were studied to find an effective model for forecasting the future sales of selected categories accurately. The machine learning and deep learning models were employed using an advanced time series forecasting technique known as recursive time series forecasting. It allows the model to perform multi-step forecasting, and the result of each time step is provided as input lag to the next time step. The forecasted sales of each model were compared with actual sales to measure the performance of the model using evaluation metrics such as R2 score, RMSE, and MAPE.

6.1 Traditional Time Series Models

Traditional time series models such as ARIMA and SARIMA were trained to forecast the future sales of selected three categories such as beverages, dairy, and grocery. Initially, the stationarity of the sales data was investigated using statistical tests such as the Augmented Dickey-Fuller (ADF) test, Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test, and Rolling Statistics test and the results indicated that data is non-stationary.

Differencing a data transformation technique was used to make data stationary. ARIMA model was employed by finding the optimal values of non-seasonal parameters p and q using the grid search method. Later, the seasonality in the data was examined using the seasonal decomposition method, and the seasonal component plot regular repeating patterns over time suggesting the data has strong seasonal effects. The SARIMA model is an extension of ARIMA designed to handle the seasonality in data by adding seasonal parameters such as P, Q, and m to the model. The SARIMA model was trained on sales data by finding the optimal values of non-seasonal components and seasonal components using the grid search method.

The impact of external features in forecasting future sales accurately was explored in this research project using ARIMA and SARIMA models. Firstly, the models were evaluated by providing lag features alone, created using historical sales data, and measured the performance of the models. Then the external features were provided along with lag features. The models trained including external features outperformed the models trained with lag features alone. This indicates the models were able to understand the underlying patterns and relationships in the data with the help of external features. Understanding the importance of external features, further analysis that involves using advanced machine learning algorithms was carried out including external features along with lag features instead of using lag features alone.

6.2 Machine Learning and Deep Learning Models

The machine learning models such as Random Forest, XGBoost (XGB), Support Vector Machine (SVM) and Facebook Prophet (FB Prophet), and a deep learning model Long Short-Term Memory (LSTM) were used to forecast the future sales of selected three categories from a superstore. An advanced forecasting technique known as recursive time series forecasting was used to design the machine learning and deep learning algorithms making them suitable to perform multi-step forecasting. Unlike traditional time series models, which assume data stationarity and linearity, machine learning and deep learning models can handle non-stationary data by capturing the complex and non-linear patterns in the data. The models were provided with lag features along with external features such as oil prices, promotions, holidays, and events to forecast future

sales accurately.

Machine learning and deep learning models were trained and evaluated using sales data related to all three categories and the results indicate the ability of machine learning and deep learning models to handle complex and non-linear patterns in the sales data. The hyperparameters of each model were fine-tuned using a manual grid search method to find the optimal values. Firstly, the analysis was carried out using the Random Forest algorithm and the model has performed decently in forecasting future sales, but the performance of the model is largely varied across the categories. Secondly, more sophisticated algorithms such as XGB and SVM were used for future sales. Predictions of both the models were accurate compared to previous models and have shown consistent performance using sales data of various categories.

Later, the FB Prophet model was employed to predict future sales and the model performed moderately in forecasting future sales. Lastly, the LSTM model a type of recurrent neural network was used to forecast future sales. LSTM is specialized in handling time series data by creating non-linear and complex patterns in the data. The model performance in forecasting sales was average compared to machine learning models. Ideally, the LSTM model requires huge data to effectively capture the underlying patterns in the data. The sales data available in this research project was limited and it is the reason behind the average performance of the LSTM model. The evaluation metrics of all the models used in this research project to forecast the sales of beverages, dairy, and grocery categories can be seen in **Table 6.1**.

6.3 Comparative Analysis

The ARIMA model was effective in capturing linear patterns and basic trends in sales data but struggled to capture seasonal and complex patterns in the sales data. The model ARIMA model performed better with sales data related to the grocery category with the help of lag features and external features. The R2 score of the model is 0.81 indicating the model was able to capture the patterns in the data effectively. However, the model struggled to forecast the future sales of beverages and dairy categories with relatively higher MAPE scores of 14.91 and 16.10 respectively. On the other hand, SARIMA has performed consistently across all three categories suggesting the model

was able to find the seasonal patterns in the sales data. The SARIMA model showed significant improvement in performance over ARIMA making it a more reliable model to forecast sales with seasonal patterns. The R2 scores of the model across all three categories are 0.78, 0.74, and 0.83 respectively.

Among three categories, the Random Forest algorithm has performed solidly using sales data related to beverages and grocery categories with MAPE errors of 10.13 % and 8.34 % respectively. The performance of Random Forecast has varied across the categories indicating the model has limitations in handling complex patterns and temporal dependencies within the data. However, the model has performed better than ARIMA and SARMIMA models across all three categories with R2 scores of 0.82, 0.74, and 0.84 respectively. The XGB model showed superior performance among all categories compared to other algorithms indicating its ability to model non-linear relationships in the data. The forecasted sales of the model were accurate and consistent across all three categories with small MAPE errors of 8.39 % 9.89 % and 9.28 %Ṫhe model is particularly effective in forecasting future sales related to the beverages category with a high R2 score of 0.86.

The SVM algorithm has shown great performance in forecasting future sales across all three categories. The results indicate the model was effective in finding the regression hyperplane that helps in minimizing the error. The model shows consistent performance in predicting future sales among all three categories. Among three categories, SVM has outperformed other models in forecasting sales of dairy and grocery categories with high R2 scores of 0.82 and 0.84 respectively. FB Prophet a specialized time series model has performed moderately across all three categories compared to other machine learning models. The forecasting error was comparatively high and the MAPE error of the three categories are 14.09 % 11.07 % and 12.28 % respectively. Lastly, LSTM an effective time series forecasting model was used to carry out the analysis. The model performance was average in forecasting sales with low R2 scores of 0.66, 0.69, and 0.70 respectively across the three categories. Overall, **Table 6.1** shows that machine learning models such as SVM and XGB have performed better in forecasting the future sales of three categories accurately compared to other machine learning, deep learning, and traditional time series models.

Model	Beverages			Dairy			Grocery		
	R2	**RMSE**	**MAPE**	**R2**	**RMSE**	**MAPE**	**R2**	**RMSE**	**MAPE**
ARIMA	0.623	328.05	14.91%	0.613	103.46	16.10%	0.814	281.83	10.25%
SARIMA	0.784	248.0	10.40%	0.745	84.03	11.39%	0.832	267.47	10.82%
RF	0.829	220.83	10.13%	0.747	83.68	11.23%	0.841	260.55	8.34%
XGB	0.860	199.36	8.39%	0.819	70.65	9.89%	0.825	272.89	9.28%
SVM	0.853	204.45	9.10%	0.821	70.44	9.12%	0.848	254.59	8.95%
FB Prophet	0.694	295.62	14.09%	0.746	83.80	11.07%	0.734	336.81	12.28%
LSTM	0.660	311.45	12.69%	0.699	91.33	10.76%	0.700	357.94	12.50%

Table 6.1: Evaluation Metrics across 3 Categories.

6.4 Limitations and Challenges Faced

In this research project, several changes and limitations were faced during training and evaluating various statistical, machine learning, and deep learning models. Among them, the most challenging one was tuning the hyperparameters of each model three times using sales data related to three different categories. Identifying optimal values for parameters of seasonal and non-seasonal components in ARIMA and SARIMA models using ACF and PACF plots was inconclusive. Therefore, a manual grid search method was used to find the optimal values for these parameters. LSTM model generally performs better with large datasets by capturing long-term dependencies. The data available in this research project is small and limited. It affected the LSTM model's ability to model the relationships and long-term dependencies in the sales data and because of the limitations in this particular sales data, the LSTM model was not able to forecast future sales accurately. These are the challenges and limitations faced in this research project.

CHAPTER 7

Conclusions

This research project aims to identify the most effective and optimal time series models to forecast the future sales of three selected categories from a single retail superstore. Various models explored in this study include traditional time series models such as Auto Regressive Integrated Moving Average (ARIMA) and seasonal Auto-Regressive Integrated Moving Average (SARIMA) along with advanced machine learning models like Random Forest, XGBoost (XGB), Support Vector Machine (SVM), a specialized time series model Facebook Prophet (FB Prophet) and a deep learning model Long Short-Term Memory (LSTM). Objectives of this study also include designing machine learning and deep learning models using recursive time series forecasting techniques, exploring the patterns in the data to find key insights in the sales data, creating new features using findings from data exploration, and investigating the impact of external features on model performance.

ARIMA and SARIMA models were implemented with and without external features and the results indicated the importance of external features in enhancing model performance. Both the models have performed better when trained with sales data that includes external features. Understanding the significance of external features, machine learning, and deep learning models were employed including external features instead of using historical sales data alone. Among the machine learning models, XGB and SVM performed better compared to other models across all three categories. Generally, the

LSTM model performs better with time series data by creating non-linear relationships and understanding long-term dependencies in data. On the contrary, the LSTM model performed the least compared to traditional time series and machine learning models. One potential reason behind the poor performance of the LSTM model is the size of the data available. LSTM ideally requires large data to understand the complex patterns and non-linear relationships that exist in time series data.

Evaluation metrics such as R2 score, RMSE, and MAPE were used to measure the performance of the models. Among all the models used in this research project, SVM and XGB have performed consistently across various categories with low forecasting errors. SVM is the best-performing model for dairy and grocery categories while XGB yielded the best results using beverage sales data. The key findings of this study include that the performance of the model depends on the nature of the data and one model outperforms another model based on the data size and complexity. In forecasting future sales of the grocery category, the SARIMA model has performed relatively similarly to the best-performing model SVM. This implies sometimes a simple model is enough to produce the results that can be achieved using more complex models. Overall, SVM and XGB models outperformed other models in forecasting the future sales across all three categories of a superstore with high R2 scores and low MAPE values highlighting the effectiveness of SVM and XGB models in capturing complex sales patterns.

7.1 Future Work

This study has provided valuable insights into time series analysis performed using recursive forecasting techniques. While this study has several findings, there is scope for improvement and future work. The availability of relatively small data is one of the limitations of this project. The limited number of observations were used to train the model, and the forecasting horizon is small due to the size of the data available. In particular, deep learning models like LSTM work best with large data. Expanding the data and increasing the forecasting horizon could help to improve the model's forecasting accuracy. In recent times hybrid models designed by combining traditional time series models with machine learning or deep learning models have gained popularity. Exploring hybrid models could help to improve upon the findings of this project and enhance

the prediction accuracy. In this research project, external features helped to improve model performance. In future work, incorporating additional external features could assist the model to better capture the patterns and trends in sales data. In the future addressing these areas could enhance the forecasting accuracy of the models.

Bibliography

[1] L. L. Har, U. K. Rashid, L. T. Chuan, S. C. Sen, and L. Y. Xia. Revolution of retail industry: from perspective of retail 1.0 to 4.0. *Procedia Computer Science*, 200:1615–1625, 2022. Elsevier.

[2] B. Berman. Flatlined: Combatting the death of retail stores. *Business Horizons*, 62(1):75–82, 2019. Elsevier.

[3] J. G. Wacker and R. R. Lummus. Sales forecasting for strategic resource planning. *International Journal of Operations & Production Management*, 22(9):1014–1031, 2002. MCB UP Ltd.

[4] A. Khakpour. Data science for decision support: Using machine learning and big data in sales forecasting for production and retail. Master's thesis, 2020.

[5] P. J. Brockwell and R. A. Davis, editors. State-space models. In *Introduction to Time Series and Forecasting*, pages 259–316. Springer New York, New York, NY, 2002. doi:10.1007/0-387-21657-X_8.

[6] R. Fildes, S. Ma, and S. Kolassa. Retail forecasting: Research and practice. *International Journal of Forecasting*, 38(4):1283–1318, 2022. Elsevier.

[7] M. M. A. Alfaki and S. B. Masih. Modeling and forecasting by using time series ARIMA models. *International Journal of Engineering Research & Technology (IJERT)*, 4(3):2278–0181, 2015.

[8] H. Allende and C. Valle. Ensemble methods for time series forecasting. *Claudio Moraga: A Passion for Multi-Valued Logic and Soft Computing*, pages 217–232, 2017. Springer.

[9] B. Lim and S. Zohren. Time-series forecasting with deep learning: A survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021. The Royal Society Publishing.

[10] C. Cheng, A. Sa-Ngasoongsong, O. Beyca, T. Le, H. Yang, and Z. Kong. Time series forecasting for nonlinear and non-stationary processes: A review and comparative study. *DOI: https://doi.org/10.1080 X*, 740817:1053–1071, 2015.

[11] D. C. Montgomery, C. L. Jennings, and M. Kulahci. *Introduction to Time Series Analysis and Forecasting*. John Wiley & Sons, 2015.

[12] G. Nunnari and V. Nunnari. Forecasting monthly sales retail time series: A case study. In *2017 IEEE 19th Conference on Business Informatics (CBI)*, volume 01, pages 1–6, 2017. doi:10.1109/CBI.2017.57.

[13] A. Dingli and K. S. Fournier. Financial time series forecasting—a deep learning approach. *International Journal of Machine Learning and Computing*, 7(5):118–122, 2017.

[14] P. Chen, A. Niu, D. Liu, W. Jiang, and B. Ma. Time series forecasting of temperatures using SARIMA: An example from Nanjing. *IOP Conference Series: Materials Science and Engineering*, 394(5):052024, July 2018. IOP Publishing. doi:10.1088/1757-899X/394/5/052024.

[15] Y. Aviv. A time-series framework for supply-chain inventory management. *Operations Research*, 51(2):210–227, 2003. INFORMS.

[16] J. Diaz-Hierro, J. J. Martín, A. Vilches Arenas, M. P. Gonzalez, J. M. Arevalo, and C. Varo González. Evaluation of time-series models for forecasting demand for emergency health care services. *Emergencias*, 24(3):181–188, 2012.

[17] A. Alqatawna, B. Abu-Salih, N. Obeid, and M. Almiani. Incorporating time-series forecasting techniques to predict logistics companies' staffing needs and order volume. *Computation*, 11(7):141, 2023. MDPI.

[18] J. G. De Gooijer and R. J. Hyndman. 25 years of time series forecasting. *International Journal of Forecasting*, 22(3):443–473, 2006. Elsevier.

[19] G. E. P. Box, G. M. Jenkins, and J. F. MacGregor. Some recent advances in forecasting and control. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 23(2):158–179, 1974. Wiley Online Library.

[20] R. S. Tsay. Time series and forecasting: Brief history and future research. *Journal of the American Statistical Association*, 95(450):638–643, 2000. Taylor & Francis.

[21] A. K. Bera and M. L. Higgins. A test for conditional heteroskedasticity in time series models. *Journal of Time Series Analysis*, 13(6):501–519, 1992. Wiley Online Library.

[22] M. Buscema. Back propagation neural networks. *Substance Use & Misuse*, 33(2):233–270, 1998. Taylor & Francis.

[23] D. Boswell. Introduction to support vector machines. *Department of Computer Science and Engineering, University of California San Diego*, 11:16–17, 2002.

[24] K. Fawagreh, M. M. Gaber, and E. Elyan. Random forests: From early developments to recent advancements. *Systems Science & Control Engineering: An Open Access Journal*, 2(1):602–609, 2014. Taylor & Francis.

[25] J. H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002. Elsevier.

[26] S. Zargar. Introduction to sequence learning models: RNN, LSTM, GRU. *Department of Mechanical and Aerospace Engineering, North Carolina State University*, 2021.

[27] M. I. Jordan. Serial order: A parallel distributed processing approach. Technical report, June 1985-March 1986. *California Univ., San Diego, La Jolla (USA). Inst. for Cognitive Science*, 1986.

[28] L. Medsker and L. C. Jain. *Recurrent Neural Networks: Design and Applications*. CRC Press, 1999.

[29] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. MIT Press.

[30] S. Ahmed, I. E. Nielsen, A. Tripathi, S. Siddiqui, R. P. Ramachandran, and G. Rasool. Transformers in time-series analysis: A tutorial. *Circuits, Systems, and Signal Processing*, 42(12):7433–7466, 2023. Springer.

[31] A. Vaswani. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[32] M. Khashei and M. Bijari. A new class of hybrid models for time series forecasting. *Expert Systems with Applications*, 39(4):4344–4357, 2012. Elsevier.

[33] P.-F. Pai and C.-S. Lin. A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega*, 33(6):497–505, 2005. Elsevier.

[34] M. Hibon and T. Evgeniou. To combine or not to combine: Selecting among forecasts and their combinations. *International Journal of Forecasting*, 21(1):15–24, 2005. Elsevier.

[35] G. Rafferty. *Forecasting Time Series Data with Facebook Prophet: Build, improve, and optimize time series forecasting models using the advanced forecasting tool*. Packt Publishing Ltd, 2021.

[36] S. Sivaramakrishnan, T. F. Fernandez, R. G. Babukarthik, and S. Premalatha. Forecasting time series data using ARIMA and Facebook Prophet models. In *Big Data Management in Sensing*, pages 47–59, 2022. River Publishers.

[37] B. Singh, P. Kumar, N. Sharma, and K. P. Sharma. Sales forecast for Amazon sales with time series modeling. In *2020 First International Conference on Power, Control and Computing Technologies (ICPC2T)*, pages 38–43, 2020. IEEE.

[38] H. Jiang, J. Ruan, and J. Sun. Application of machine learning model and hybrid model in retail sales forecast. In *2021 IEEE 6th International Conference on Big Data Analytics (ICBDA)*, pages 69–75, 2021. IEEE.

[39] S. Pang. Retail sales forecast based on machine learning methods. In *2022 6th Annual International Conference on Data Science and Business Analytics (ICDSBA)*, pages 357–361, 2022. IEEE.

[40] T. Deng, Y. Zhao, S. Wang, and H. Yu. Sales forecasting based on LightGBM. In *2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pages 383–386, 2021. IEEE.

[41] X. Li, J. Du, Y. Wang, and Y. Cao. Automatic sales forecasting system based on LSTM network. In *2020 International Conference on Computer Science and Management Technology (ICCSMT)*, pages 393–396, 2020. IEEE.

[42] B. K. Jha and S. Pande. Time series forecasting model for supermarket sales using FB-prophet. In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pages 547–554, 2021. IEEE.

[43] Y. Ji, J. Hao, N. Reyhani, and A. Lendasse. Direct and recursive prediction of time series using mutual information selection. In *Computational Intelligence and Bioinspired Systems: 8th International Work-Conference on Artificial Neural Networks, IWANN 2005, Vilanova i la Geltrú, Barcelona, Spain, June 8-10, 2005. Proceedings 8*, pages 1010–1017, 2005. Springer.

[44] A. Singh. An Introduction to Non-Stationary Time Series in Python. Available at: https://www.analyticsvidhya.com/blog/2018/09/an-introduction-to-non-stationary-time-series-in-python/, 2018. Accessed: 2024-08-23.

[45] V. Sharma. The ARMA (Autoregressive Moving Average) model: A popular statistical model used in time series forecasting. Available at: https://medium.com/@vaibhav1403/the-arma-autoregressive-moving-average-model-is-a-popular-statistical-model-used-in-time-series-24ce43049366, 2023. Accessed: 2024-08-23.

[46] M. Cerliani. How to improve recursive time series forecasting. *Towards Data Science*, 2024. Available at: https://towardsdatascience.com/how-to-improve-recursive-time-series-forecasting-ff5b90a98eeb. Accessed: 2024-08-22.

[47] J. Amat Rodrigo and J. Escobar Ortiz. Recursive multi-step forecasting. Available at: https://skforecast.org/0.9.1/user_guides/autoregresive-forecaster, 2023. Accessed: 2024-08-22.

[48] A. Raafat. Decision Trees and Random Forest: All You Need to Know. Available at: https://mlarchive.com/machine-learning/decision-

trees-and-random-forest-all-you-need-to-know/, 2022. Accessed: 2024-08-27.

[49] P. Grover. Gradient Boosting From Scratch. Available at: https://blog.mlreview.com/gradient-boosting-from-scratch-1e317ae4587d, 2017. Accessed: 2024-08-28.

[50] A. Chauhan. Fully Explained Gradient Boosting Technique in Supervised Learning. Available at: https://pub.towardsai.net/fully-explained-gradient-boosting-technique-in-supervised-learning-d3e293ca70e1, 2021. Accessed: 2024-08-28.

[51] P. Premanand. Support vector machine: Better understanding. Available at: https://www.analyticsvidhya.com/blog/2021/06/support-vector-machine-better-understanding/, 2023. Accessed: 2024-08-30.

[52] D. Choudhary. Why Support Vector Regression? Difference Between SVR and a Simple Regression Model. Available at: https://medium.com/@dhirendrachoudhary_96193/why-support-vector-regression-difference-between-svr-and-a-simple-regression-model-8cd752a77bbc, 2023. Accessed: 2024-08-30.

[53] N. Kumar and S. Susan. COVID-19 Pandemic Prediction using Time Series Forecasting Models. In *Proceedings of the International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2020.

[54] R. Dhupkar. Demand forecasting using FB Prophet. Available at: https://towardsdatascience.com/demand-forecasting-using-fb-prophet-e3d1444b9dd8, 2020. Accessed: 2024-08-31.

[55] S. Timalsena. Time Series Analysis: A Quick Tour of FBProphet. Available at: https://medium.com/analytics-vidhya/time-series-analysis-a-quick-tour-of-fbprophet-cbbfbffdf9d8, 2020. Accessed: 2024-08-31.

[56] C. Olah. Illustrated Guide to LSTMs and GRUs: A Step by Step Explanation. Available at: https://towardsdatascience.com/illustrated-guide-to-

lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21, 2018. Accessed: 2024-09-01.

[57] O. Surakhi, M. A. Zaidan, P. L. Fung, N. H. Motlagh, S. Serhan, M. AlKhanafseh, R. M. Ghoniem, and T. Hussein. Time-lag selection for time-series forecasting using neural network and heuristic algorithm. *Electronics*, 10(20):2518, 2021.