E-commerce Sales Forecasting Using Machine Learning Algorithm

Indrayani Desale

Student Id: 10637913

Applied Research Project submitted in partial fulfilment of the requirements for the degree of Masters of Science in Business Analytics at Dublin Business School

Supervisor: Charles Nwankire

January 2024

Declaration

‘I declare that this Applied Research Project that I have submitted to Dublin Business School for the award of Masters of Science in Business Analytics is the result of my own investigations, except where otherwise stated, where it is clearly acknowledged by references. Furthermore, this work has not been submitted for any other degree.’

Signed: Indrayani Desale

Student Number: 10637913

Date: 08/01/2024

Acknowledgment

I would want to express my sincere thanks to Charles Nwankire, my supervisor, for all of their help and support during this research. I would especially want to thank my family and friends for their constant support. I value the resources offered by Dublin Business School. I am grateful to everyone whose work has impacted our research and for their contributions. Without the combined assistance of these people, this expedition would not have been feasible.

Abstract

Businesses looking to maximise inventory, marketing tactics, and overall operational efficiency must consider e-commerce sales forecasts when making strategic decisions. The goal of this project is to determine the best method for predicting e-commerce sales by developing, assessing, and contrasting three time series machine learning models: ARIMA, Facebook (FB) Prophet and LSTM. The goals of the study are to prepare the dataset, create the model, tune the hyperparameters, and evaluate the performance. It explains how complicated e-commerce sales trends are and highlights the need for improved models or tactics to identify subtle patterns. Out of all the 3 models Facebook (FB) Prophet outperformed ARIMA and LSTM with decent evaluation matrix score like (RMSE): 273.79, (MSE): 74965.26, (MAE): 221.24. It also identifying intricate patterns and offering insightful analysis of e-commerce sales data. However the FB prophet also fails to give accurate e-commerce sales predictions like the ARIMA and LSTM models.

Keywords: e-commerce sales forecasting, ARIMA, FB prophet, LSTM, Time series analysis, machine learning, hyperparameter tuning

Table of Contents

Table of Figures 7

List of Tables 7

Chapter 1: Introduction 8

1.1. Background: 8

1.2. Research Aims: 10

1.3. Overview: 11

1.4. Rational: 13

1.5. Research Question: 13

Chapter 2 : Literature Review 15

2.1. Introduction: 15

2.2. E-Commerce Sales Forecasting Importance: 15

2.3. Time Series Models Sales Forecasting: 16

2.4. Why Auto Regressive Moving Average (ARIMA)? 16

2.5. Why FB Prophet? 17

2.6. Why Long Short Term Memory (LSTM)? 18

2.7. Conclusion: 19

Chapter 3 : Research Methodology 20

3.1. Research Planning: 20

3.2. Business Understanding: 20

3.3. Data Collection: 20

3.4. Data Description: 21

3.5. Data Pre-processing: 22

3.6. Data Analysis: 22

3.7. Missing Values: 23

3.8. Duplicate Values: 23

3.9. Feature Extraction: 24

3.10. Index Setting: 24

3.11. Data Normalization: 24

3.12. Data Splitting: 24

3.13. Machine Learning Models: 25

3.13.1. ARIMA (Auto Regressive Integrated Moving Average): 25

3.13. 2. FB (Facebook) Prophet: 25

3.13.3. LSTM (Long Short-Term Memory): 26

3.14. Model Training: 27

3.14.1. ARIMA (Auto Regressive Integrated Moving Average) Training: 27

3.14.2. FB Prophet Training: 29

3.14.3. LSTM (Long Term Short Memory) Training: 30

Chapter 4 : Apparatus 33

4.1. Google Colboratoy: 33

4.2. Python: 33

4.3. Scikit-learn: 33

4.4. Statsmodels: 34

4.5. Matplotlib: 34

4.6. MinMaxScaler: 34

4.7. Keras: 35

4.8. TimeseriesGenerator: 35

4.9. Pmdarima: 35

4.10. Prophet: 36

Chapter 5 : Results 37

5.1. ARIMA Results: 37

5.2. FB Prophet Results: 44

5.2.1. Basic FB prophet: 44

5.2.2. Tuned FB prophet: 48

5.2.3 Best FB prophet model: 50

5.3. LSTM Results: 51

5.4. Model Comparison: 54

Chapter 6 : Conclusion & Future Prospects 55

References: 57

Table of Figures

Figure 1: Time Series Plot of Amount 38
Figure 2: Augmented Dickey-Fuller Test for Stationarity 38
Figure 3: Autocorrelation 39
Figure 4: Partial Autocorrelation 40
Figure 5: Auto Arima Results 41
Figure 6: Arima Results 42
Figure 7: Arima Forecasting: Actual vs Predicted 43
Figure 8: Basic FB Prophet Model 44
Figure 9: Frecasted Dataframe Tail (Basic FB Prophet) 45
Figure 10: Forecasted Value Plot of Basic FB prophet 46
Figure 11: Forecast Component Plots (Basic FB Prophet) 47
Figure 12: Tuned FB Prophet Model 48
Figure 13: Frecasted Dataframe Tail (Tuned FB Prophet) 48
Figure 14: Forecasted Value Plot of Tuned FB prophet 49
Figure 15: Forecast Component Plots (Tuned FB Prophet) 50
Figure 16: E-commerce Time Series Seasonal Decomposition 52
Figure 17: LSTM Model Actual and Predicted Values 53
Figure 18: LSTM Forecasting Plot: Actual vs Predictions 53

List of Tables

Table 1: ARIMA Evaluation Matrix Results 43
Table 2: Basic FB Prophet Evaluation Matrix Results 47
Table 3: Tuned FB Prophet Evaluation Matrix Results 50
Table 4: FB Prophet Evaluation Matrix Results Comparison 51
Table 5: FB Prophet Evaluation Matrix Results Comparison 54
Table 6: Machine Learning Model Evaluation Matrix Results Comparison 54

Chapter 1: Introduction

Introduction is the chapter one in the proposed project where detailed background for the research study is provided with mentioning research aims, objective and outcomes for the same, also the rational is included. The second chapter is literature review, this gives information on the previous work done in context of the study topic and evaluation of the same to support the research question and form a base for main research objectives. The third chapter is research methodology including the detailed steps involved such as research planning, business understanding, data collection, data description, data pre-processing and model training. The fourth chapter involves details on technologies used thought out the research process development. The fifth chapter is based on the final research results, where detailed such statistic, visuals and interpretations are made to conclude the outcome of the study. The sixth chapter is the last one which discusses the relevant outcomes of the research study and suggest future prospects.

1.1. Background:

Majority of e-commerce industries and other companies revenue all over the globe depends upon the how much sales is being created. Rapid technology breakthroughs, shifting industry trends, and dynamic customer behaviour characterise the e-commerce sector. In this environment, firms trying to handle the intricacies of online shopping find that being able to predict sales properly becomes strategically critical. E-commerce sales forecasting is a powerful tool that uses data-driven insights to estimate client demand, improve inventory control, and create winning marketing campaigns. Furthermore it is beneficial for decision making and planning effective business strategies for any industry in a competitive market. In the past traditional sales forecasting methods such as statistical analysis and historical data has been used. However, in today’s modern world where the huge amount of data is being generated, use of technology such machine learning becomes an advantages to reach highest level of reliability and accuracy for forecasting sales in any industry. The development of machine learning has fundamentally changed how sales forecasting is done in the e-commerce industry. Businesses may use cutting-edge algorithms to mine past sales data for important patterns and trends that will help them remain ahead of the competition and make wise decisions.

This research aims to advance theoretical knowledge of e-commerce sales forecasting while also offering useful insights that companies can use to improve their operational effectiveness and strategic decision-making. This can be achieved by navigating the approaches, obstacles, and possible solutions. The research findings are intended to provide useful information to e-commerce sector stakeholders, promoting a data-driven approach to long-term expansion and flexibility in the online market. In order to better understand the complexities of e-commerce sales forecasting, this study will apply machine learning models and examine the complex nature of time series data. The research will be focused on building machine learning model using time series machine learning algorithms such as autoregressive moving integrated average model (ARIMA), FB-Prophet model and long short term memory (LSTM) model. The first constrain in the research was data pre-processing to build the desired machine learning model for forecasting the sales for an e-commerce store, as the data pre-processing steps for each algorithm used is different. The second constrain was to find the seasonality with the appropriate hyper parameter tuning to get the best predictive results because the e-commerce dataset is large having 129875 total observations and it is difficult to find trend with such a huge number of observation.

1.2. Research Aims:

This research aims to build a machine learning model to forecast the e-commerce sales, evaluate its performance and suggest the best predictive model for future sales forecasting to take the business decisions.

Following are the research outcomes and objectives to be studied during this research journey.

Anticipated outcomes from the research work are:

- To see which machine learning model is best suited for e-commerce sales forecasting by comparing evaluation matrix.
- Does this machine learning model helps to improve accuracy and enhance robustness for future business decisions.
- How the data seasonality can be used for the future sales decision making.

Research objectives are:

- To collect the diverse sales dataset enclosing various important characteristics required for sales prediction, analyse it and prepare the dataset to train machine learning algorithm.
- To develop a framework to build the 3 time series machine learning algorithm for e-commerce sales prediction.
- To tune the hyper parameters to get the best performing model.
- To evaluate the performance of the proposed machine learning models as well as focusing on the interpretability to choose the best model.

1.3. Overview:

Time series analysis, statistical modelling, and deep learning are some of the core ideas that this research endeavour combines in its quest to improve e-commerce sales forecasting via machine learning. The fundamental ideas of this investigation are as follows:

- **Time Series Data:**

  Through the successive recording of observations across time, time series data adds a temporal component to the study. Critical to comprehending the historical e-commerce sales statistics are its underlying patterns, trends, and seasonality. The temporal dynamics may be understood in large part by using techniques like autocorrelation analysis and decomposition.

- **Auto Regressive Integrated Moving Average (ARIMA):**

  A traditional statistical modelling method for time series forecasting is the Autoregressive Integrated Moving Average (ARIMA). In order to attain stationarity, the data must be transformed in order to capture the autoregressive and moving average components. This research's key finding is how well ARIMA captures seasonality and linear patterns.

- **Facebook Prophet (FB Prophet):**

  Facebook Prophet is a time series forecasting tool that takes a comprehensive approach, accounting for weekly seasonality and vacations among other temporal trends. Key ideas ingrained in the use of Prophet include the integration of domain expertise, managing incomplete data, and breaking down time series components.

- **Long Term Short Memory (LSTM):**

Deep learning is explored by Long Short-Term Memory (LSTM), a recurrent neural network (RNN) version. It is quite good at identifying both transient and permanent relationships in sequential data. The notion of memory cells and their capacity to hold onto data for long stretches of time makes the model useful for managing intricate patterns in e-commerce sales data.

- **Data Preparation:**

  In order to get the raw e-commerce sales data ready for analysis and model training, data preparation is essential. During this step, a number of crucial actions are involved in ensuring the data is acceptable for accurate forecasting by cleaning and converting it. A thorough approach to data pre-processing is required for the preparation of the e-commerce sales data. Crucial actions include cleaning, dealing with missing values, and handling outliers.

- **Model Evaluation:**

  Standard assessment measures, such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE), are used in the research study. These metrics are used as quantitative indicators to evaluate the forecasting models' performance and accuracy.

- **Interpretation of the Model and Its Business Outcomes:**

  Beyond the technical details, the research study investigates how interpretable the best model is and turns the results into useful business insights to take right decisions at right time. Knowing how the forecasting outcomes fit into actual situations and decision-making procedures is an essential part of any research work. As this enables to see if the proposed solution is feasible for the real world problem or it needs further improvements to perform better.

- **Progress in the Future and Ongoing Enhancement:**

  Recognising that e-commerce is dynamic, the study establishes the foundation for further advancements. Prospects for further study include the idea of continuous learning, possible improvements in feature engineering, and investigation of sophisticated forecasting models.

To put it simply, the research integrates machine learning algorithms, statistical techniques, and practical issues into a complex web of ideas. By combining these ideas, the intention is to both advance the area of e-commerce sales forecasting and provide a more comprehensive grasp of the dynamic interaction between big data and decision-making in the context of digital commerce.

1.4. Rational:

This research study is conducted to contribute to the previous work done in the field of e-commerce sales forecasting. The previous work for e-commerce sales forecasting is mostly done on the smaller datasets and basic machine learning models were used. Using a big dataset and hyperparameter tuning is a smart way to improve machine learning models' performance in the context of e-commerce sales forecasting. Diverse data and optimised hyperparameters work together to both speed up learning and guarantee that the models are tailored to the unique needs and features of the e-commerce industry.

1.5. Research Question:

Which machine learning model is best for e-commerce sales forecasting?

Sub Question:

How to pre-process the data for creating the machine learning model for sales forecasting?

How to analyse the seasonality from the e-commerce dataset?

How to interpret the model performance to choose the best machine learning algorithm for sales forecasting?

Chapter 2 : Literature Review

2.1. Introduction:

The e-commerce industry is growing at a fast pace and this has imposed a lot of challenges like big data generation, managing resources, and mainly forecasting the sales of different products. To solve the forecasting challenge in e-commerce industry the time series machine learning algorithms have been proven as the best tool with ease of implementation. This literature review is focused on the existing worked done in the field of sales forecasting using various time series machine learning algorithms. It also highlights the importance of time series analysis techniques in the field of forecasting the sales. Further are the sections involved in this literature are time series analysis for sales forecasting and conclusion.

2.2. E-Commerce Sales Forecasting Importance:

For any retail company which is selling products online platform like amazon it is crucial to pay attention to important factors like number of transactions, sales revenue with time and page visits to take the future decisions considering the mentioned values for effective planning. Additionally the correct sales prediction helps in cost reduction and enhances the customer satisfaction mainly in any e-commerce business. (Singh, B., Kumar, P., Sharma, N. and Sharma, K.P., 2020, January). As the technology is advancing the online selling platforms are also increasing rapidly with the increase in the e-commerce sales. So to make perfect predictions and sales improvement, the sales experts are in need of the technology that decreases the rate of human errors, helps in marketing, operational decision making, inventory management and improves the decision making for online platforms. The sales forecasting is crucial for any e-commerce platform because as the retail shops web presence is increasing so the digital data generated with time. This digitally generated data with time series is the primary base for sales forecasting and deep understanding of such data with the seasonality or holiday patterns is the main challenging factor for many online businesses. To understand such complex sales data implementing advanced technologies such as time series analysis is helpful to make write business decisions. (Bajoudah, M. Alsaidi and A. Alhindi, 2023). Big data is increasing with the advancement in the big data technology and its application in the e-commerce industry is also rapidly increasing. So creating a model using predictive analysis technology is beneficial for planning inventory strategy for a company, managing the supply chain and demand. (T. Tang, 2023).

2.3. Time Series Models Sales Forecasting:

In today’s world every business is going online and due to this amount of online generated sales data is skyrocketing. This online generated sales data is in time series format and hence using the time series based machine learning algorithm for forecasting the sales is most suited as they are widely used and efficient in identifying the seasonal patterns. (Ali, Y. and Nakti, S., 2023).

2.4. Why Auto Regressive Moving Average (ARIMA)?

Auto Regressive Integrated Moving Average is one of the popular and traditional machine learning algorithm used for forecasting the time series data. The ARIMA forecasting model is easy to implement and easy to understand. ARIMA analyses the historical time series data point and the model modifies the data by changing the values until it reaches to the optimal forecasting results. (Hlupić, T., Oreščanin, D. and Petric, A.M., 2020, September). The Hlupić, T., Oreščanin, D. and Petric, A.M. applied a novel approach for sales prediction for the wholesale industry. In this research a data mart was build that tracks the sales of the products sold and this servers as the basis for the sales prediction. The ARIMA (autoregressive integrated moving average) was chosen as the predictive model. Further they suggested to enhance the forecasting by introducing the neural network based forecast. (Hlupić, T., Oreščanin, D. and Petric, A.M., 2020, September). A comparative study between time series and regression algorithms was done for the Walmart sales dataset to predict seasonal sales, as it is very crucial to understand such trends for the multinational retails stores to bring the maximum possible profit by understanding the sales pattern. In this study the ARIMA model outperformed the other regression and time series models like Holt-Winters, linear regression, decision tree regression and random forest regression. (Vyas, R. and As, R., 2022, March). One more research paper shows implementation of 3 for forecasting methods namely holt-winters exponential smoothing, ARIMA and neural network auto regression model on the Amazon historical sales data. After analysis of the performance the researchers concluded that the seasonal ARIMA model best fits the sales dataset and gives better forecast as compared to the other machine learning approaches (Singh, B., Kumar, P., Sharma, N. and Sharma, K.P., 2020, January).

Considering all the past research work done for sales forecasting with the time series ARIMA model is most efficient in predicting the time series data, giving best predictions and it is also superior at handling the seasonality of the dataset.

2.5. Why FB Prophet?

FB Prophet has been introduced by the Facebook for as an open source forecasting tool. The main objective of the FB Prophet is to observe the time series data with various time intervals like hourly, daily and monthly. Additionally it also takes care of holiday seasonality, specific trends, missing values and outliers. The proposed research work has done time series forecasting for supermarket furniture sales data (9994 records) using FB-Prophet model. They compare 3 forecasting models namely the additive model, the autoregressive integrated moving average (ARIMA) model and FB Prophet model. After evaluating the performance for each of the algorithm they concluded FB Prophet as the best performing model with low error, better forecasting and better fitting. However they suggested to use fusion technique to improve the performance of FB prophet and also to work on the scalability as to make predictions on the large dataset can be challenging. (B. Kumar Jha and S. Pande, 2021)

The dataset used in this research work is from the e-commerce platform with large number of observations and according to the suggestion given in above research work the FB Prophet can be used to work on large dataset to capture the variations of the holiday seasonality from the dataset or other existing data trends.

2.6. Why Long Short Term Memory (LSTM)?

Long Short Term Memory is a class of RNN and can identify long term dependencies. They are developed in a way to avoid the long term reliance issues. When applied practically LSTM makes long term memory retention as default setting because of this it is best for time series data forecasting. (S. Prakash, A. S. Jalal and P. Pathak). Studies on the LSTM showed that it is best for the sales forecasting than the traditional machine learning algorithm. In this research paper LSTM technique is used for automatic sales forecasting for Walmart sales dataset containing 30491 goods sales record for 1913 days record. Researchers first worked on feature engineering to create features like last promotion sales or average sales for past month to give more contextual insights. Then build the LSTM model with 3 LSTM layers and one dense layer while the evaluation matrix chosen was RMSSE. They compared the performance of LSTM with traditional logistic regression and SVM. After evaluation the LSTM model outperformed both logistic regression and SVM. (X. Li, J. Du, Y. Wang and Y. Cao, 2020)

2.7. Conclusion:

The above review gives a broad overview of the importance of e-commerce sales forecasting and use of time series machine learning technique in the sales forecasting for different industries. It emphasises on the importance of the new emerging technology like FB Prophet and Long Short Term Memory model and how it can be beneficial to make accurate predictions about the future sales. Accurate sales predictions can be useful for any industry as it gives helps in resource allocation, business strategy planning, profit estimation, revenue prediction, etc. The above research work mainly used the simple time series model, some of they have combined time series model with other technologies and the dataset used for each research was comparatively smaller or medium size dataset. In this proposed research work for which the literature review has been done, the main focus will be the how effectively time series models can handle the large dataset and tuning of hyper parameters to get the best forecasting model. This review is important to understand the background behind the chosen topic and take appropriate steps.

Chapter 3 : Research Methodology

3.1. Research Planning:

The main objective of the research methodology is to create the 3 models Auto Regressive Moving Average (ARIMA), FB Prophet and Long Short Term Memory (LSTM) for forecasting the e-commerce sales data to give accurate future sales predictions and identify trends in the dataset. Choose the best predictive sales forecasting model in comparison to the various evaluation matrices used and tuning of the various hyper parameters. The in detailed steps involved in this research work are explained in further sections.

3.2. Business Understanding:

E-commerce business is widely developing across the various industries and forecasting the e-commerce sales is necessary to take appropriate business actions for marketing activities, operational activities and many more. Data generated through e-commerce platform is huge and compact with the complex patterns. The main challenge is to understand the e-commerce sales data and use it for predicting the future sales to benefit the business. So to overcome this challenge research work is focused on building a machine learning algorithm with fine tune hyper parameter to get the accurate sales forecasting.

3.3. Data Collection:

The data collection was done from the dataset repositories due to the data protection rules. Sales data for many industries, companies or e-commerce stores is private and sensitive information. So it is very difficult to get the sales data from any organization as there are many company privacy rules to use the given information. The data used in this research work was collected from the data.world, an open data community allows to work on any dataset. The chosen dataset has insights about the e-commerce sales on amazon platform with various attributes related to the sales. Each of the variable in dataset is explained below.

3.4. Data Description:

- Index (Integer): The column is normal index.
- Order ID (String): The column represents the unique identification number for each sale.
- Date (Date): The column represents the date on which sales is made.
- Status (String): The column represents the status of the sales.
- Fulfilment (String): The column represents method of order fulfilment. (Amazon, Merchant)
- Sales Channel (String): The column represents the channel for the order placed.
- Ship-service-level (String): The column represents the level of shipping service. (Standard or Expedited)
- Style (String): The column represents the style of products.
- SKU (String): The column represents the stock keeping unit.
- Category (String): The column represents type of product. (Blouse, Bottom, Dupatta, Ethnic Dress, Kurta, Saree, Set, Top, Western Dress)
- Size (String): The colun represents size for the each product category. (XS, S, M, L, XL, XXL, Free.)
- ASIN (String): The column represents the amazon standard identification number.
- Courier Status (String): The column represents status of the courier. (Cancel, Shipped, Unshipped)
- Qty (Interger): The column represents the quantity of the product sold.

- Currency (String): The column represents the currency used during the sales.
- Amount (Float): The column represents the amount of the sales. This column is also the target variable for forecasting the e-commerce sales.
- Ship-city (String): The column represents the shipment city.
- Ship-state (String): The column represents the shipment state.
- Ship-postal-code (Float): The column represents the shipment postal address.
- Ship-country (String): The column represents the shipment country.
- Promotion-ids (String): The column represents the promotional identification number.
- B2B (Boolean): The column represents business to business sale.
- Fulfilled-by (String): The column presents the shipment fulfilled by
- Unnamed: 22 (String): The column does not specify the information.

3.5. Data Pre-processing:

Data pre-processing is the most critical step involved in the machine learning technique. Data cleaning is important because the real world data has many gaps such as missing values, unsuitable data types, duplicate values, etc. and these gaps can impact the performance of machine learning algorithms. So one cannot overlook this step as it will compromise the quality of analysis and impact the performance of machine learning model. Below are the detailed steps involved in the data pre-processing.

3.6. Data Analysis:

First step in any machine learning model creation is analysing the given dataset to check its compatibility with the model. Compatibility involves various aspects like data types, null values, total number of observation and this information can be observe (.info), (.shape),

(.head) and (.describe). The (.info) variable shows column data types, total entries in the dataset and will also show the non-null count for each column. The (.shape) variable shows shape of the selected dataset i.e. total number of columns and total number of rows present. The (.head) shows the top five rows of the dataset. The (.describe) shows basic statistics for all the numeric columns in the dataset. Before analysing the data, it needs to be read using pandas.

3.7. Missing Values:

The next step in the data pre-processing is finding the missing values or the NaN observations in the dataset. It is crucial to handle the NaN values in the dataset, as these values can impact the predictions of the machine learning algorithm. To handle the NaN values in the dataset first find the count of the NaN values in each column and then delete the whole columns or delete rows containing NaN values or replace the NaN values with some other values (Statistical average). Deleting the columns or deleting only rows or replacing depends on the specific purpose of the research study. In this research the NaN values from the dataset are deleted using data.dropna() method.

3.8. Duplicate Values:

Finding duplicate values in the dataset is important because it can lead model to give biased prediction thus hampering the robustness of the machine learning model. Sometimes it may create problem for normalizing the dataset. This research work is based on the time series and handling duplicates is very crucial as the model should not do repeated observations. To handle duplicate values first find the dataset rows having duplicate values and then drop rows containing duplicate observations. The e-commerce dataset used in this research does not contain any duplicate values and this is verified by using data.duplicated().any() method.

3.9. Feature Extraction:

This research work is based on time series analysis and the target variable is ‘Amount’. The dataset used for the study has 23 variable. To build the time series model extracting the target variable and date column is necessary and create a new dataset to build the desired time series model. Both the ‘Date’ and ‘Amount’ column is extracted from the main data frame and created a new data frame called as ‘amount_series’.

3.10. Index Setting:

The next step is setting ‘Date’ variable from the dataset as index for the time series analysis. Indexing the ‘Date’ variable helps in sub-setting the time series data on specified time intervals and improves the data visualization to observe the specific data trends or data patterns or seasonality.

3.11. Data Normalization:

The proposed research work is conducted on the time series analysis and there are different reason why to normalize dataset while working on time series analysis. One of the reason is that machine learning algorithms can give biased prediction results with the unbalanced dataset because the dataset have some higher value and some lower values. The data compatibility with the machine learning model is another reason for data normalization, as in this research LSTM algorithm is used which is sensitive to the input variable scales. So it is important to bring the dataset to a common range for better results. To normalize the dataset feature means transforming them to have mean ‘0’ and variance ‘1’. Data normalization is done using minmax scaler method imported from the sklearn package.

3.12. Data Splitting:

After all the pre-processing of the data the final step is to split the dataset into training and testing sets. The training set will be used to train the machine learning model while the testing set will be used to make predictions with the trained model. There is set splitting ratio for the dataset that is 80% data should be training set and 20 % data should be testing set. The chosen dataset is a time series dataset so the splitting the data is based on the index rather than randomly shuffling the data, so that the machine learning model can concisely use past data to make the correct future predictions.

3.13. Machine Learning Models:

3.13.1. ARIMA (Auto Regressive Integrated Moving Average):

ARIMA stands for auto regressive integrated moving average also called as differential autoregressive moving average model. It is the combination of two Auto Regressive (AR) and Moving Average (MA) models. ARIMA is divided into 3 parameters such as p, d & q. The ‘p’ value represents the auto regressive component, ‘q’ value represents the moving average component and, ‘d’ value is number of differencing performed to make a time series stationary. (Shi, R. and Zhang, C., 2023). The primary objective of ARIMA is to find the values of p, d and q. To apply ARIMA model to the dataset it is necessary to make the time series data stationary as the series will have autocorrelation and harmonious mean over the time period.   (Ali, Y. and Nakti, S., 2023)

3.13. 2. FB (Facebook) Prophet:

The FB Prophet is an open source time series analysis and forecasting tool made available by Facebook. FB Prophet’s parameters are simple to comprehend, and its prediction capabilities don't require a large amount of time-series data. The method works best when there are significant seasonal characteristics in the time series data that function as influencing variables. It also handles scheduled holidays or pauses in the continuous data. When it comes to outlier identification, trend shifts, and missing information, Facebook Prophet performs better. (B. Kumar Jha and S. Pande, 2021). It is compatible with programming languages R and Python.

$$y(t) = g(t) + s(t) + h(t) + -(t)$$

Prophet operates on the 4 key components trends g(t), seasonality s(t), holidays h(t), -(t) represents the error. g(t) represents the non-periodic fluctuations in the time series data, s(t) represents periodic seasonality like daily, weekly, monthly, yearly and other seasonal changes. h(t) shows the impact of holidays on potentially inconsistent schedules spanning one or more days, and –(t) shows data not included in the model. (Bajoudah, M. Alsaidi and A. Alhindi, 2023).

3.13.3. LSTM (Long Short-Term Memory):

LSTM stands for Long Short-Term Memory. It is a better version of a recurrent neural network that can learn the inter-timestamp relationships in time series data and store knowledge over an extended length of time. It preserves the data information and keeps it from being lost over extended periods of time by using memory cells. The input gate, forget gate, and output gate are the three main parts of an LSTM cell. The forget gate modifies the data, the output gate sends the new data to the next timestamp, and the input gate decides whether the data from the previous timestamp should be kept. In an LSTM cell, these three elements are known as gates. When training LSTMs, a substantial amount of computer power is needed in comparison to statistical models. Pre-processing and LSTM parameter adjustments might be difficult as well. The effectiveness of LSTM can also be significantly impacted by the optimizer selection, the quantity of hidden layers, and the number of neurons in each hidden layer. While choosing the hyperparameters, great care must be given because even little adjustments can have a significant impact on the model's performance. (Ali, Y. and Nakti, S., 2023).

3.14. Model Training:

3.14.1. ARIMA (Auto Regressive Integrated Moving Average) Training:

- All the steps of data pre-processing should be completed before training the model with the dataset and the data is already pre-processed as mention in the data pre-processing section. After the pre-processed time series data is plotted to see if there are any patterns in the time series. The graph is discussed in results.
- Next step is to check whether the time series is stationary or non-stationary. To check if the data is stationary or not Augmented Dickey–Fuller test (ADF) is applied. This test checks the null hypothesis that states if the time series has unit root then it is said to be non-stationary and if the time series has no unit root then it rejects the null hypothesis declaring the series as stationary. This test gives values of ADF statistics, p-value, number of lags, number of observations used for ADF regression and critical values. In the test the null hypothesis is rejected based on p-value limit that the ADF statistic value should be less than the set p-value ($p <= 0.05$). So ADF test is run on the pre-processed e-commerce time series, to check if the e-commerce time series data is stationary or not.
- Next step is to plot the autocorrelation and partial autocorrelation graphs for the time series. This step is important in every time series analysis as it helps in identifying the data seasonality present and also identifies the values for parameters ‘p’ and ‘q’.

Generally an AR(p) process is suggested if the PACF has a fast cutoff after lag p and the ACF decays gradually. An MA(q) process is suggested if the PACF shows a progressive decrease after lag q whereas the ACF has a sudden cutoff. The e-commerce time series was plotted for ACF and PACF to find the AR (p) and MA(q) component values. The value ‘d’ is differencing order and can found after getting results for ADF test as differencing is done if the time series is not stationary.

- Next step did was finding best hyperparameters without identifying it from the ACF and PACF. To find the best hyper parameters (p, d, q) auto_arima function by ‘pmdarima’ library in python was used for e-comerce time series data. The auto_arima function looks for the optimal set of parameters using a step-by-step method based on information criteria such as the Bayesian Information Criterion (BIC) and the Akaike Information Criterion (AIC). After running auto_arima for the e-commerce time series data, it gave the best model hyperparameters (p,d,q) .
- The next e-commerce time series data was split into 80% training set, 20% testing set and the using index splitting because it helps maintaining the chronological sequence of observations through the division of time series data by the index enables more precise model validation, assessment, and training within the framework of time-dependent relationships.
- Then the ARIMA model was train and fitted by using best hyper parameters achieved through the auto_arima and training set. It also prints the result summary.
- After training the model predictions were made using testing set and were plotted as a graph for clear visualization.
- Final step was to calculate the evaluation matrices namely mean squared error (MSE), mean absolute error (MAE) and Root Mean Squared Error (RMSE).

3.14.2. FB Prophet Training:

- Dataframe with ‘ds’ and ‘y’ column: To build FB Prophet model it is necessary to convert the date column and the measurable target column into ‘ds’ and ‘y’ column format to create new dataframe, as it is the requirement for the FB prophet. The ‘ds’ represents datestamp column and the ‘y’ column represents the measurable target variable to be forecasted. The ‘ds’column is ‘Date’ with the format YYYY-MM-DD set by pandas and ‘y’ column is ‘Amount’ that needs to be forecasted in this research work.
- After creating new data frame with ‘ds’ and ‘y’ column, it was plotted as graph to observe the existing patterns in the dataset. As observed in ARIMA model e-commerce time series data does not have any specific pattern.
- The next step for the FB prophet was to split the dataset in training and testing set. The dataset was split in such a manner that the training set had all the observations except the last 100 data observations and the test had the last remaining 100 data observations only. The reason to split the data in such fashion is to make predictions for 30 days into the future.
- The next step after splitting the dataset is to train the model on the training set, create the future data frame for predictions and hyperparameter tuning to get the best fit model. At the first basic FB prophet for (period = 30) run was without setting any hyperparameters and evaluated the predictions. However for the first basic model the results were not satisfying so added more hyperparameters like frequency, holiday seasonality and number of change points. After trial and error best tuned hyperparammeters were found with best model predictions.
- After finalizing the hyperparameters, predictions for the future dates listed in the future dataframe are produced using the predict technique. The outcome is kept in the forecast

dataframe, which further includes sections for trends and seasonality. Other columns include 'ds' (dates), 'yhat' (predicted values), and 'yhat_lower' and 'yhat_upper' (uncertainty intervals).

- Then the forecasted values were plotted with ‘m.plot(forecast)’ function. It is a time series plot that shows the historical data, the model's fitted values, and the predicted values together with uncertainty intervals. This is a practical method for evaluating the FB prophet model's performance graphically and it is inbuilt in the feature of FB prophet.
- Additionally using the function ‘m.plot_components(forecast)’ individual component plots were generated for the e-commerce time series forecast created using the FB prophet model. The several components that go into the overall forecast, such as trend, seasonality, and holidays, are shown visually with the aid of these component charts. The component charts for e-commerce time series included the trend chart and weekly seasonality chart.
- The final step for the FB prophet model was to calculate the evaluation matrices. Similarly like the ARIMA, mean squared error (MSE), mean absolute error (MAE) and Root Mean Squared Error (RMSE) were calculated for prophet model.

3.14.3. LSTM (Long Term Short Memory) Training:

- After pre-processing the time series is broken down into its component parts (amount, trend, seasonal, and residual) using the seasonal_decompose function. With a period of 7, the decomposition is carried out on the assumption of daily seasonality. The disassembled components are visualised using the results.plot() line.

- The next is data set is divided into training (80%) and testing set (20%). The total observations in training first 96944 observations and testing set contains remaining 24236 observations.
- Most important step for the LSTM is to normalize the data to give accurate predictions. Train and test dataset were reshaped to make it compatible with the scaler before normalizing by using minmax scaler. Print the first 10 values of the normalized training set to validate the changes.
- Next step is to create sequences input data (X) and output data (Y) for the LSTM model. Here input sequence is 30 means, 30 observations will be used to give one output and this will be one batch for the LSTM. The output created will be used to create the next output and the process will be continued.
- Now the model is defined with the necessary hyperparameters, an LSTM layer, and a dense layer. The hyperbolic tangent activation function (tanh) is used by the 50-unit LSTM layer, which anticipates input sequences of the form (n_input, n_features).
- Then the model is fitted on the training data using 15 epochs and generated created in the previous step. Plotting training loss per epoch is important in LSTM to understand if the model is learning from the dataset or not. This can be ensure if the training loss in the graph is continuously decreasing then it can be said that the model is learning and better fitting its parameter to fit the training sets. It is also important to decide the number of epochs to select, if the loss is still decreasing after certain number of epoch then epoch number can be increased and if the loss is plateaued then no need to add more number of epoch. In regard of the research dataset for first trial 10 epochs were used, however after observing the loss per epoch was still decreasing then epoch number increased to 15 and then the loss plateaued.

- In order to make predictions, a batch of the latest 30 values from the training set is created and the trained model is used to make next value prediction in the sequence. Make the prediction using the test set and create a new dataframe containing original scale prediction created by inversely transformed scaled prediction. Plot the predictions to understand visually.
- Last step is the same like ARIMA and FB prophet to calculate the evaluation matrices mean squared error (MSE), mean absolute error (MAE) and Root Mean Squared Error (RMSE) were calculated for prophet model for comparing the models.

Chapter 4 : Apparatus

4.1. Google Colboratoy:

“Google Colaboratory” or simply “GoogleColab” is a product of Google Research. Colab is particularly useful for machine learning, data analysis, and data cleaning since it enables anybody to create and run arbitrary python code through a browser. Google colab does not need any special installation set ups to use it and it is free to access. This is the primary source used for this research to create machine earning algorithms.

4.2. Python:

Python is created under an open source licence that has been accepted by OSI, allowing for unrestricted use and distribution. Python is the ideal option because of its many features, including its huge framework library, versatility, and convenience of use for creating machine learning models. The python language is straightforward syntax, it accelerates the scraping, processing, refining, cleaning, organising, and analysis procedures and makes data validation easier. Python is a language for beginners and very beneficial as it is easy to read, will enhance the construction of many applications, ranging from basic text processing to sophisticated models. It can be executed in any operating system, including Windows, macOS, Linux, Unix, and others. The Python online community facilitates the process of constructing or debugging machine learning models by providing information and responses.

4.3. Scikit-learn:

Scikit-learn is an open-source machine learning framework for Python that offers effective and user-friendly tools for modelling and data analysis. It is constructed on top of existing libraries for scientific computing, including Matplotlib, SciPy, and NumPy. With a standardised interface for a range of machine learning applications, scikit-learn is made with ease of use in mind. It offeres a range of various machine learning algorithm like regression, classification and clustering. In conclusion, scikit-learn is a versatile, user-friendly library that is indispensable for practitioners, scholars, and machine learning enthusiasts. It has consistent and comprehensive documentation that can help to find solutions for each problem regarding machine learning.

4.4. Statsmodels:

A Python module called statsmodels offers statistical calculations as an addition to scipy, encompassing estimation and inference for statistical models as well as descriptive statistics. For statistical modelling applications such as linear regression and time series analysis, Statsmodels is frequently utilised. Numerous statistical models are available for application, such as ARIMA (Auto Regressive Integrated Moving Average), which is often employed in time series forecasting.

4.5. Matplotlib:

A Python charting package called Matplotlib creates static, animated, and interactive visualisations. It is extensively utilised for producing different kinds of plots, graphs, and charts. Matplotlib is a valuable tool for data visualisation in exploratory data analysis, model performance evaluation, and result presenting. It may be used to make histograms, line graphs, scatter plots, and more.

4.6. MinMaxScaler:

Numerical features can be scaled to a specified range, usually between 0 and 1, using the data preparation technique MinMaxScaler. To ensure that features are on the same scale, MinMaxScaler is frequently used in machine learning. This helps keep some characteristics from predominating over others, which is especially helpful for algorithms that rely on gradient-based optimisation or distance measurements.

4.7. Keras:

A Python deep learning package available for free is called Keras. It offers a sophisticated interface for configuring and learning neural networks. It is a popular tool for creating and experimenting with deep learning models. It supports several backends, such as TensorFlow and Theano, and enables users to easily design neural networks. Building intricate neural network topologies is made easier using Keras.

4.8. TimeseriesGenerator:

Created especially for time series forecasting, the TimeseriesGenerator utility class in Keras creates batches of temporal data sequences. When producing time series data for deep learning model training, TimeseriesGenerator comes in handy. Recurrent neural networks (RNNs) and long short-term memory networks (LSTMs) may both use it since it generates input-output pairs from time series data.

4.9. Pmdarima:

A Python module called Pmdarima expands Statsmodels' capacity for choosing ARIMA models automatically. It seeks to make choosing the best ARIMA model parameters easier. More accessible to users without in-depth knowledge of time series modelling, Pmdarima automates the process of determining which ARIMA model is optimal for a particular time series dataset.

4.10. Prophet:

Facebook created Prophet, an open-source forecasting tool for time series data. Holidays, seasonality, and other trends in time series data are all handled by design. Forecasting jobs involving daily data that exhibit patterns like weekly and annual seasonality are especially well-suited for Prophet. Utilising it is simple, and it handles outliers and missing data well.

Chapter 5 : Results

This chapter is to discuss the results got from all the 3 ARIMA, FB Prophet, LSTM machine learning algorithms and to get the best fitting model for e-commerce sales forecasting by comparing the evaluation matrices of each model built. Also this section will discuss about the seasonality trend analysed for the e-commerce data, the best tuned hyperparameters for each algorithm and visualization prediction results of the models. All the result section is divided into ARIMA results, FB prophet results, LSTM results and model comparison.

5.1. ARIMA Results:

- **E-commerce time series plot:**

  The pre-processed time series data was plotted to observe the trends or pattern or seasonality. Below is the graph plotted ‘Date’ column v/s ‘Amount’ column. From the plotted graph it can be seen that e-commerce time series does not have trend or pattern that can be identified.

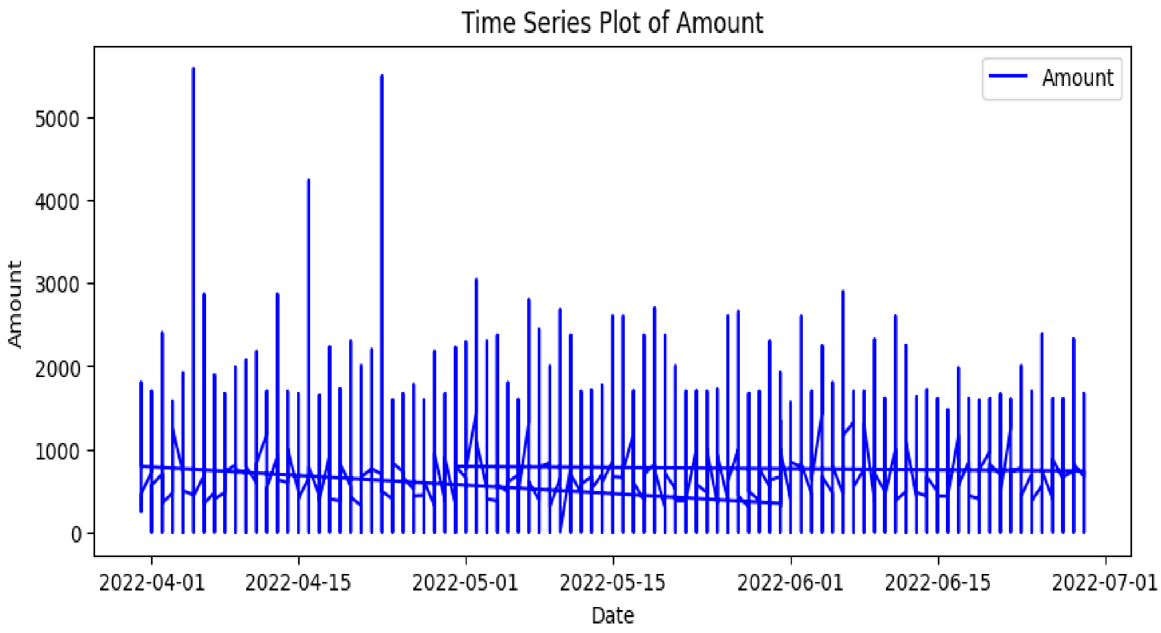
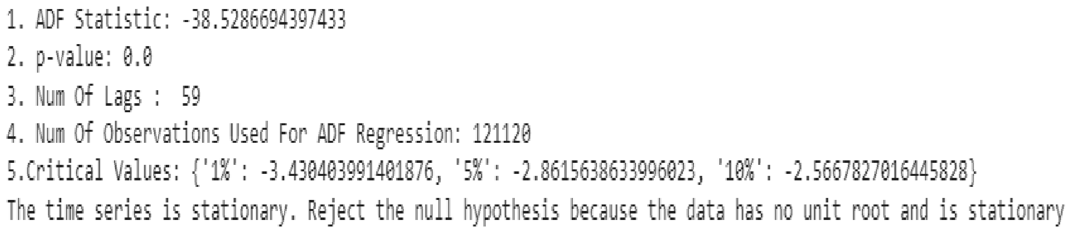


Figure 1: Time Series Plot of Amount

- **ADF Test:**

ADF test is run on the pre-processed e-commerce time series, it reject the null hypothesis as the ADF statistic value for e-commerce time series data is less than the p value limit (<= 0.05) hence the e-commerce time series is stationary. The ADF test results are as below:



```
1. ADF Statistic: -38.5286694397433
2. p-value: 0.0
3. Num Of Lags :  59
4. Num Of Observations Used For ADF Regression: 121120
5.Critical Values: {'1%': -3.430403991401876, '5%': -2.8615638633996023, '10%': -2.566782701644582 8}
The time series is stationary. Reject the null hypothesis because the data has no unit root and is stationary
```

Figure 2: Augmented Dickey-Fuller Test for Stationarity

- **ACF and PACF plots:**

The e-commerce time series was plotted for ACF (autocorrelation) and PACF (partial autocorrelation) however it did not show any significant seasonality. Hence it was hard to find the AR (p) and MA(q) component values. Both the plots are as below:

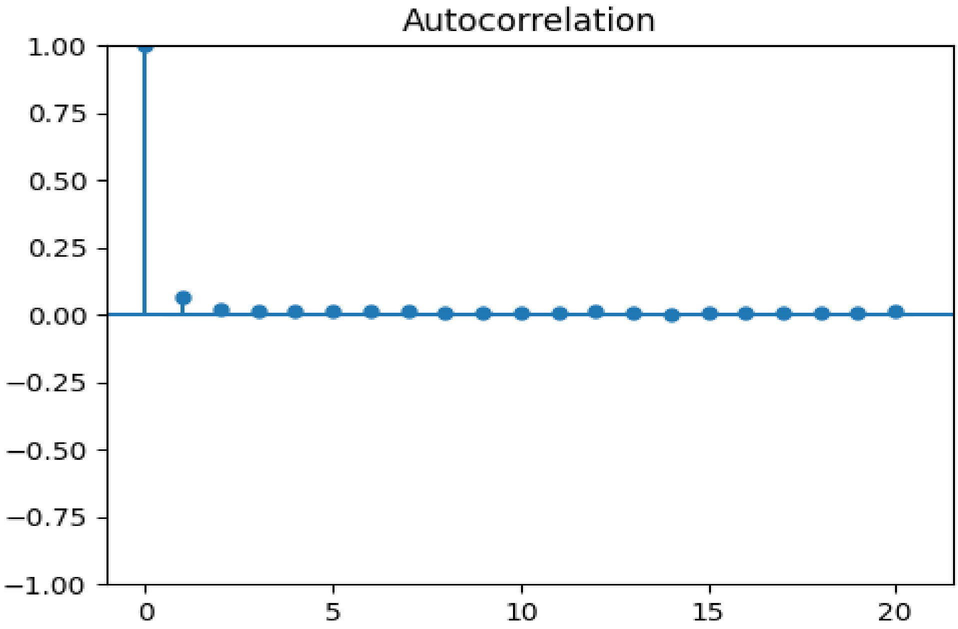


Figure 3: Autocorrelation

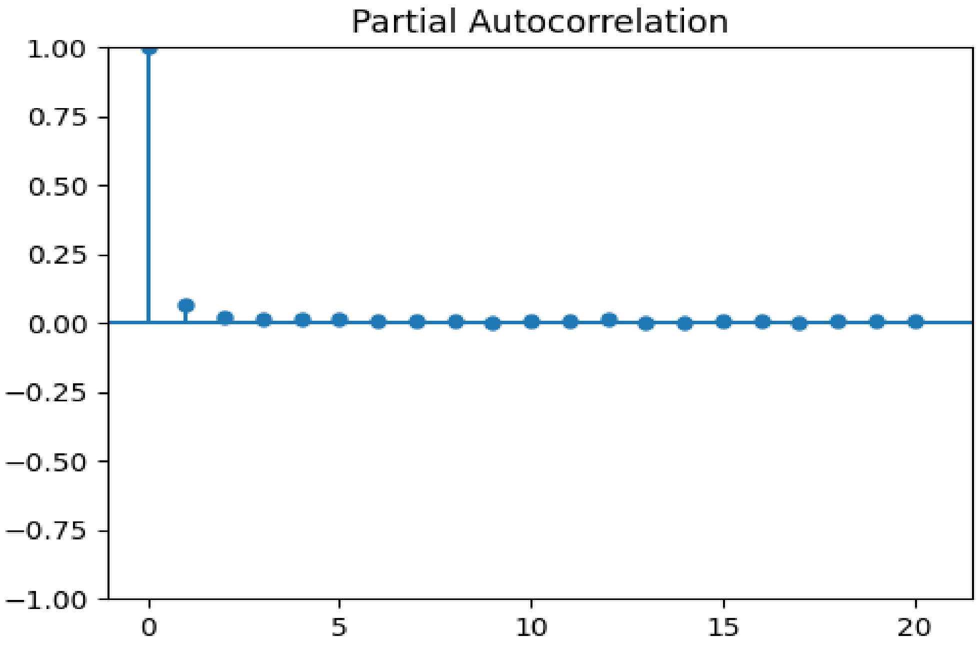
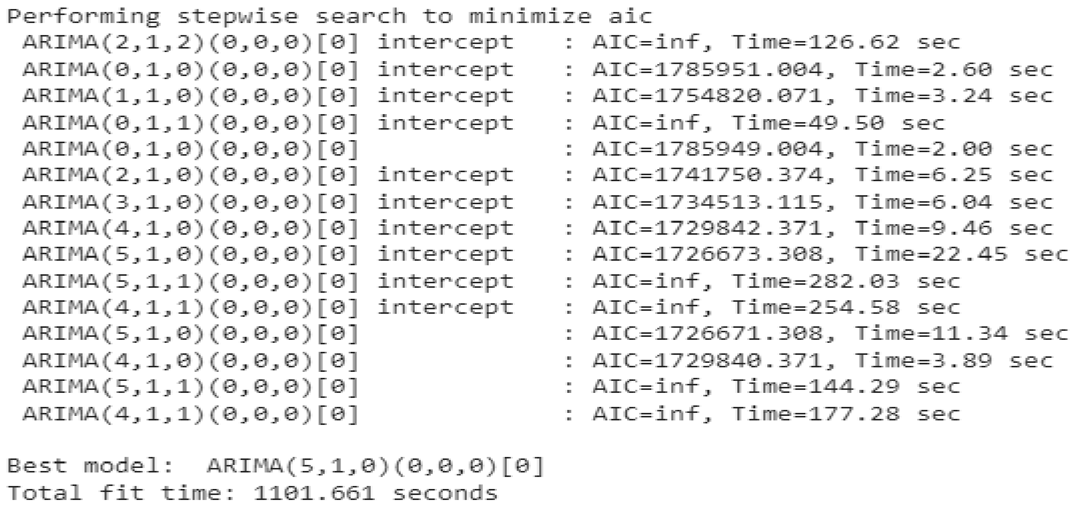


Figure 4: Partial Autocorrelation

- **Auto_arima:**

  As it was difficult to get optimum hyperparameters from ACF and PACF plots, auto_aria function was used to find the best hyperparaeters (p,d,q) to build final ARIMA forecasting model. Based on information criteria like the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC), the auto_arima function iteratively searches for the ideal grouping of parameters. The results shows best model ARIMA (5,1,0)(0,0,0)[0] and fit time Total fit time: 1101.661 seconds for e-commerce time series.



```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=inf, Time=126.62 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=1785951.004, Time=2.60 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=1754820.071, Time=3.24 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=inf, Time=49.50 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=1785949.004, Time=2.00 sec
 ARIMA(2,1,0)(0,0,0)[0] intercept   : AIC=1741750.374, Time=6.25 sec
 ARIMA(3,1,0)(0,0,0)[0] intercept   : AIC=1734513.115, Time=6.04 sec
 ARIMA(4,1,0)(0,0,0)[0] intercept   : AIC=1729842.371, Time=9.46 sec
 ARIMA(5,1,0)(0,0,0)[0] intercept   : AIC=1726673.308, Time=22.45 sec
 ARIMA(5,1,1)(0,0,0)[0] intercept   : AIC=inf, Time=282.03 sec
 ARIMA(4,1,1)(0,0,0)[0] intercept   : AIC=inf, Time=254.58 sec
 ARIMA(5,1,0)(0,0,0)[0]             : AIC=1726671.308, Time=11.34 sec
 ARIMA(4,1,0)(0,0,0)[0]             : AIC=1729840.371, Time=3.89 sec
 ARIMA(5,1,1)(0,0,0)[0]             : AIC=inf, Time=144.29 sec
 ARIMA(4,1,1)(0,0,0)[0]             : AIC=inf, Time=177.28 sec

Best model:  ARIMA(5,1,0)(0,0,0)[0]
Total fit time: 1101.661 seconds
```

Figure 5: Auto Arima Results

- **ARIMA model with best hyperparameters:**

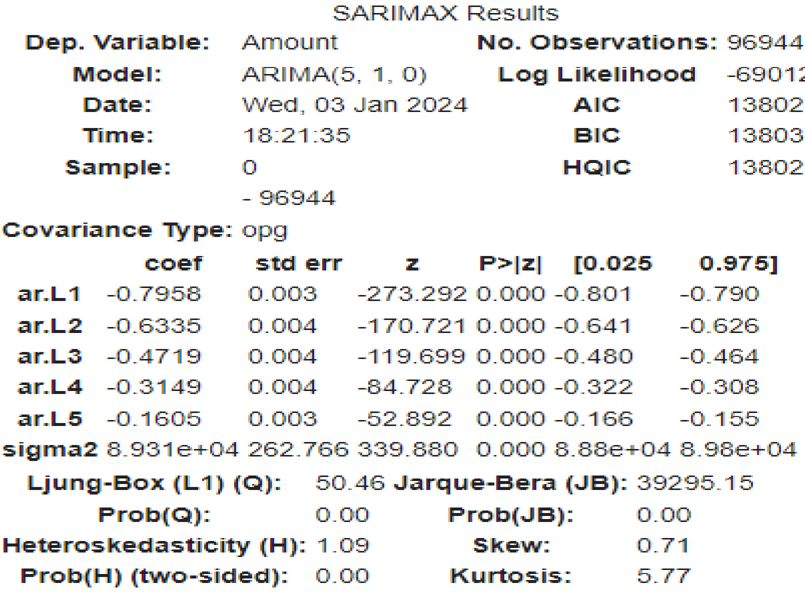
The best model hyperparameters (p=5, d=1, q=0) from the auto_arima was used to train the training set and fit the final ARIMA model. The ARIMA result summary and key observations is as below:

Log Likelihood: A measure of how well the model describes the observed data is the log-likelihood, which is -690123.474. A greater log-likelihood signifies a more favourable match and based on log likelihood value the ARIMA model seems to fit well on the e-commerce time series data.

AIC (Akaike Information Criterion): 1380258.948 - The AIC penalises complexity and measures how well the model fits the data. Reduced AIC values indicate an improved model.

BIC (Bayesian Information Criterion): 1380315.839 - Model complexity is more severely penalised than AIC.

AR Coefficients: ar.L1 to ar.L5 values are autoregressive coefficients, representing impact of past value on the current value.



```
                              SARIMAX Results
    Dep. Variable:      Amount               No. Observations: 96944
        Model:          ARIMA(5, 1, 0)        Log Likelihood   -690123.474
         Date:          Wed, 03 Jan 2024           AIC         1380258.948
         Time:          18:21:35                   BIC         1380315.839
       Sample:          0                         HQIC         1380276.237
                        - 96944
  Covariance Type: opg
                 coef      std err      z       P>|z|   [0.025     0.975]
  ar.L1   -0.7958    0.003    -273.292 0.000 -0.801     -0.790
  ar.L2   -0.6335    0.004    -170.721 0.000 -0.641     -0.626
  ar.L3   -0.4719    0.004    -119.699 0.000 -0.480     -0.464
  ar.L4   -0.3149    0.004    -84.728  0.000 -0.322     -0.308
  ar.L5   -0.1605    0.003    -52.892  0.000 -0.166     -0.155
  sigma2 8.931e+04 262.766 339.880  0.000 8.88e+04 8.98e+04
    Ljung-Box (L1) (Q):      50.46 Jarque-Bera (JB): 39295.15
           Prob(Q):          0.00        Prob(JB):       0.00
  Heteroskedasticity (H): 1.09           Skew:           0.71
   Prob(H) (two-sided):    0.00        Kurtosis:        5.77
```

Figure 6: Arima Results

- **ARIMA forecasting results:**

The final ARIMA model was used to forecast the unseen testing and results of actual v/s predicted values were plotted to have clarity how well predictions have been made. By looking at the prediction graph it can be concluded that the ARIMA model could not predict the future e-commerce sales accurately. Possible reason can be that the model failed to capture e-commerce time series data dependencies. The ARIMA e-commerce sales forecasting results are as below:

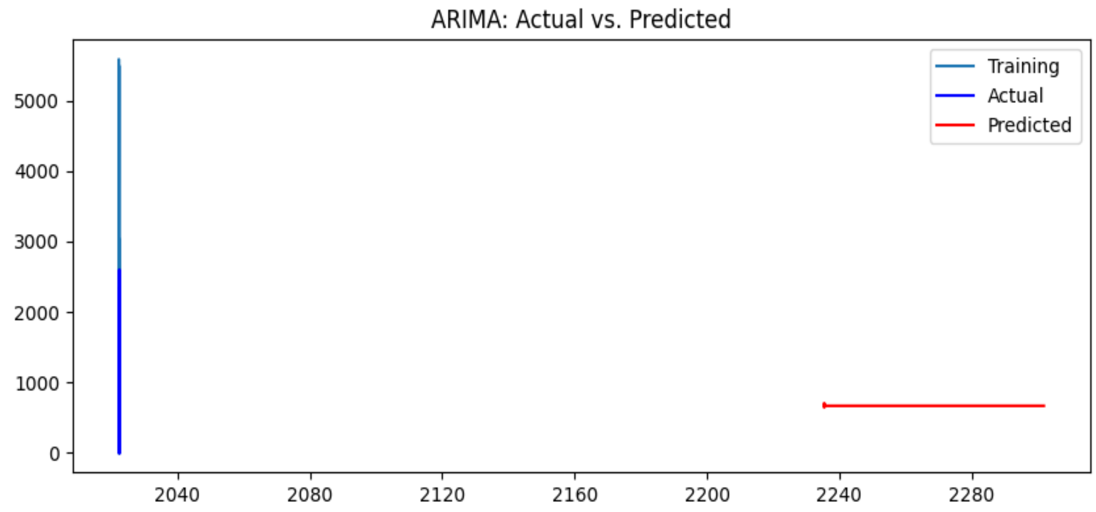


Figure 7: Arima Forecasting: Actual vs Predicted

- **ARIMA performance evaluation:**

There are 3 evaluation matrices used Mean Squared Error (MSE), Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). Evaluation matrix results are as below:

Table 1: ARIMA Evaluation Matrix Results

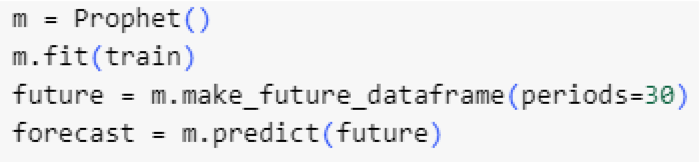
Evaluation Matrix Name	**Results**
Mean Squared Error (MSE)	83930.8490217562
Mean Absolute Error (MAE)	226.16292691242236
Root Mean Squared Error (RMSE).	289.70821359042657

5.2. FB Prophet Results:

Two models were built a basic FB prophet model and another tuned FB prophet model.

5.2.1. Basic FB prophet:

- This is basic prophet model with no specific hyperparaeters mentioned, it is trained on the e-commerce training data set to make predictions for 30 days into the future. Below are the parameters used for this model.



```
m = Prophet()
m.fit(train)
future = m.make_future_dataframe(periods=30)
forecast = m.predict(future)
```

Figure 8: Basic FB Prophet Model

- **Forecasted Values of Basic FB prophet:**

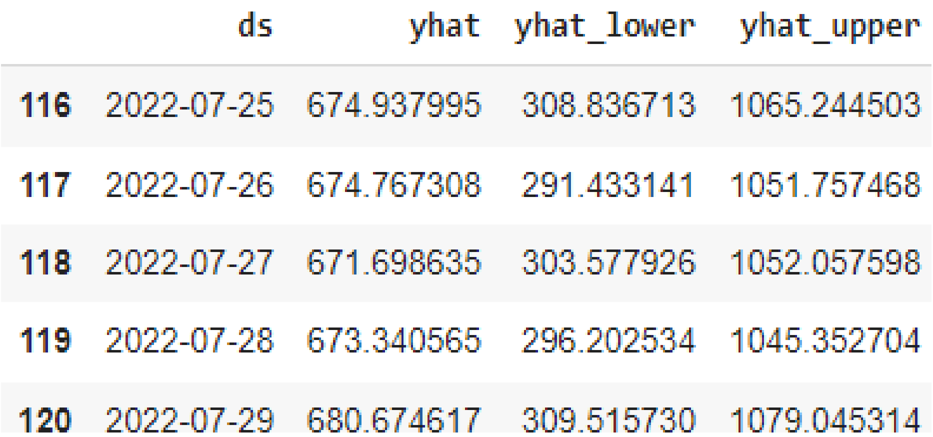
  The below table represents forecasted values for the e-commerce sales data. Predictions and evaluation matrices score for this model is as below:

  ds: This column shows the dates for which future predictions are made.

  yhat: This column shows the forecasted e-commerce sales values for dates in ds column.

  yhat_lower: The lower bound of the prediction interval is shown in this column, showing the lowest range that the actual values are most likely to fall inside.

yhat_upper: This column shows, the top bound of the prediction interval, or the upper range within which the actual values are most likely to fall.



	ds	yhat	yhat_lower	yhat_upper
116	2022-07-25	674.937995	308.836713	1065.244503
117	2022-07-26	674.767308	291.433141	1051.757468
118	2022-07-27	671.698635	303.577926	1052.057598
119	2022-07-28	673.340565	296.202534	1045.352704
120	2022-07-29	680.674617	309.515730	1079.045314

Figure 9: Frecasted Dataframe Tail (Basic FB Prophet)

- The below plot is visual representation of the forecasted values for the e-commerce sales data by basic FB prophet model. From the above plots it can be seen that the model could not catch the dataset trend hence making inaccurate predictions.

  Black dots: Represents the actual observed data points from the e-commerce time series.

  Blue Line: Represents the predicted e-commerce sale values.

  Light Blue Area: Represents the prediction intervals.

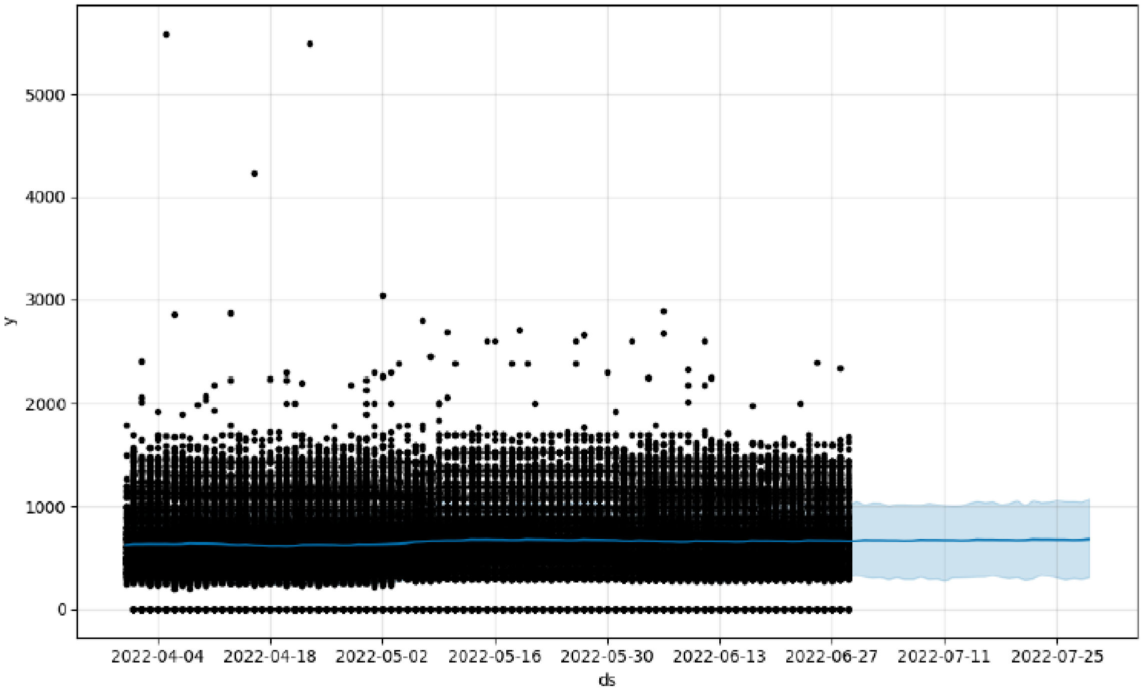


Figure 10: Forecasted Value Plot of Basic FB prophet

- The below image represents individual component plots generated for the e-commerce sales forecast by basic FB prophet model. The first plot represents the e-commerce sales dataset trend and it can be observe that the e-commerce time series dataset had initial trend however after certain time interval the trend got stationary over the time period. The second plot represents the e-commerce sales dataset weekly seasonality and it can be observed that there is daily seasonality in e-commerce dataset. The trend starts to gradually decrease from Friday till Wednesday and again gradually increases till Friday. Concluding the maximum e-commerce sales is happing on the Friday and the minimum sales is happing on Wednesday.

  Blue Line: Represents the predicted trend for e-commerce sale values and show the underlying dataset pattern.

Light Blue Area: Represents the prediction intervals for the identified seasonal component trend.

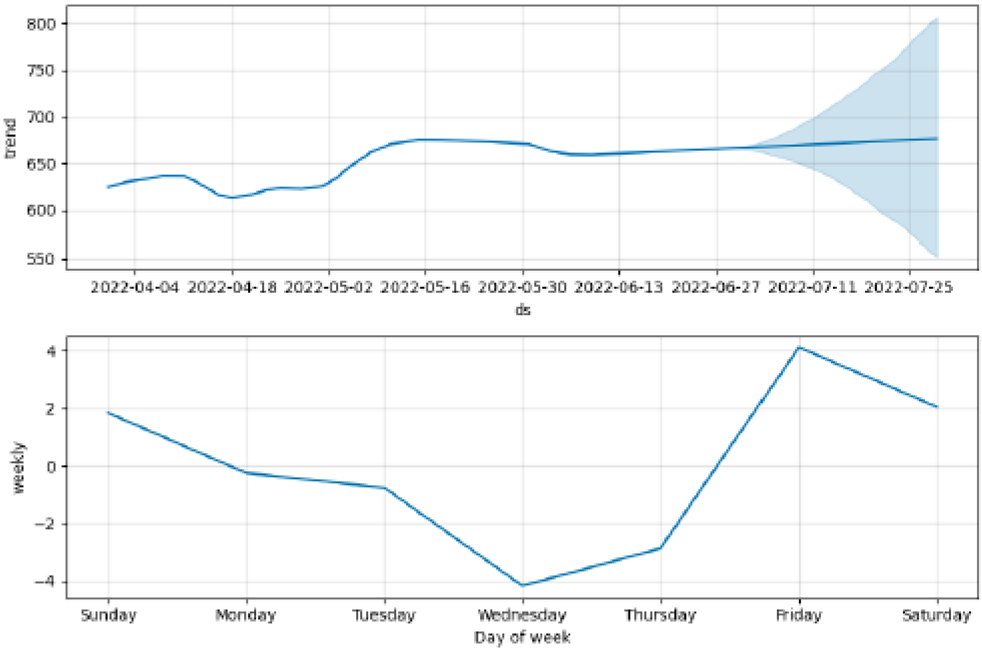


Figure 11: Forecast Component Plots (Basic FB Prophet)

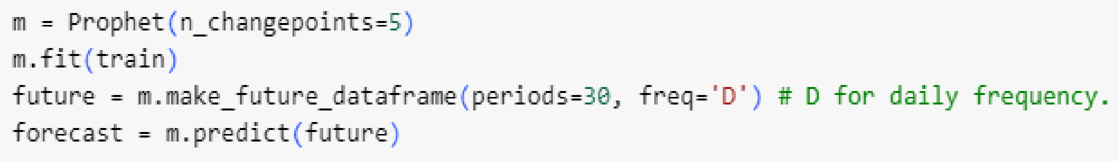
- **Basic FB prophet performance evaluation:**

Table 2: Basic FB Prophet Evaluation Matrix Results

Evaluation Matrix Name	**Results**
Mean Squared Error (MSE)	75379.90
Mean Absolute Error (MAE)	222.04
Root Mean Squared Error (RMSE).	274.55

5.2.2. Tuned FB prophet:

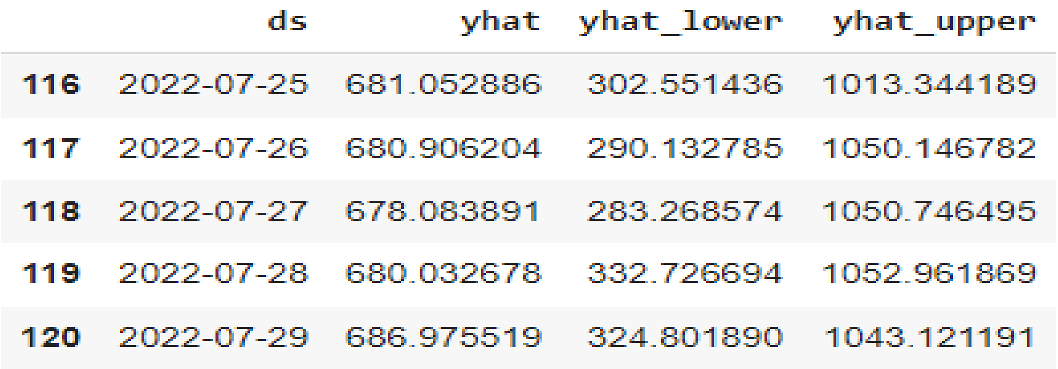
- This is tuned prophet model with hyperparameters like n_changepoints and frequency set as daily. It is trained on the e-commerce training data set to make predictions for 30 days into the future. Hyperparameters for the model are as below:



```
m = Prophet(n_changepoints=5)
m.fit(train)
future = m.make_future_dataframe(periods=30, freq='D') # D for daily frequency.
forecast = m.predict(future)
```

Figure 12: Tuned FB Prophet Model

- The below table represents forecasted values for the e-commerce sales data. It shows the values ds, yhat, yhat_lower and yhat_upper and the explanation for all the columns is done in the basic FB prophet model result. Also it can be seen that the predicted values are slightly different than those predicted in the basic FB prophet model. Predictions and evaluation matrices score for this model is as below:



	ds	yhat	yhat_lower	yhat_upper
116	2022-07-25	681.052886	302.551436	1013.344189
117	2022-07-26	680.906204	290.132785	1050.146782
118	2022-07-27	678.083891	283.268574	1050.746495
119	2022-07-28	680.032678	332.726694	1052.961869
120	2022-07-29	686.975519	324.801890	1043.121191

Figure 13: Frecasted Dataframe Tail (Tuned FB Prophet)

- The below plot is visual representation of the forecasted values for the e-commerce sales data by tuned FB prophet model. The above tuned FB prophet forecasting plot is slightly better than basic FB prophet model. However it also not able to predict the accurate e-commerce sales forecasting.

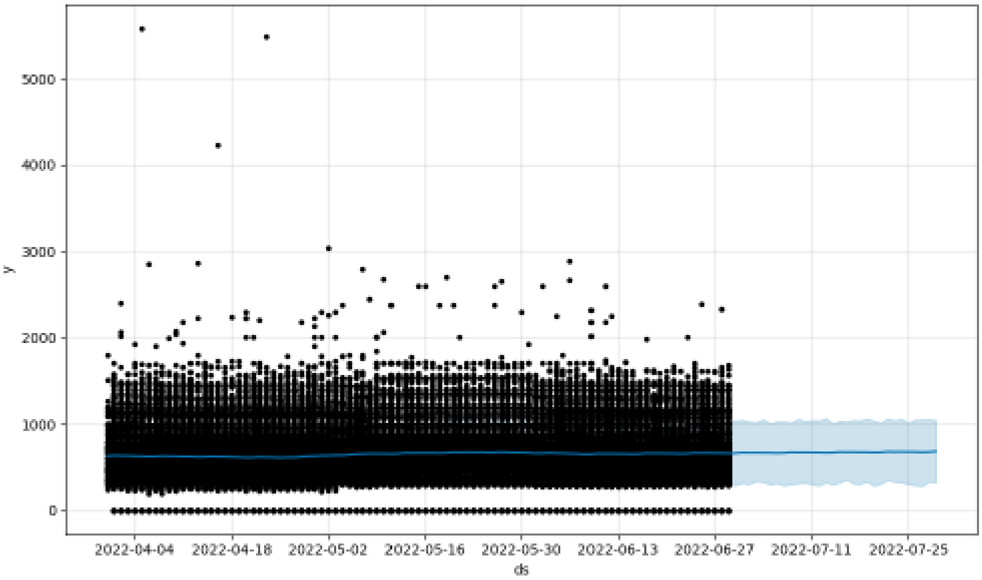


Figure 14: Forecasted Value Plot of Tuned FB prophet

- The below image represents individual component plots generated for the e-commerce sales forecast by tuned FB prophet mode. The first plot represents the e-commerce sales dataset trend. It is clearly visible from the graph how the amount is changing drastically for 5 data points over the period of 15 days and after that increasing steadily. The second plot represents the e-commerce sales dataset weekly seasonality and it gives the similar insights as observed in the basic FB prophet model such as maximum sale on Friday and minimum on Wednesday.

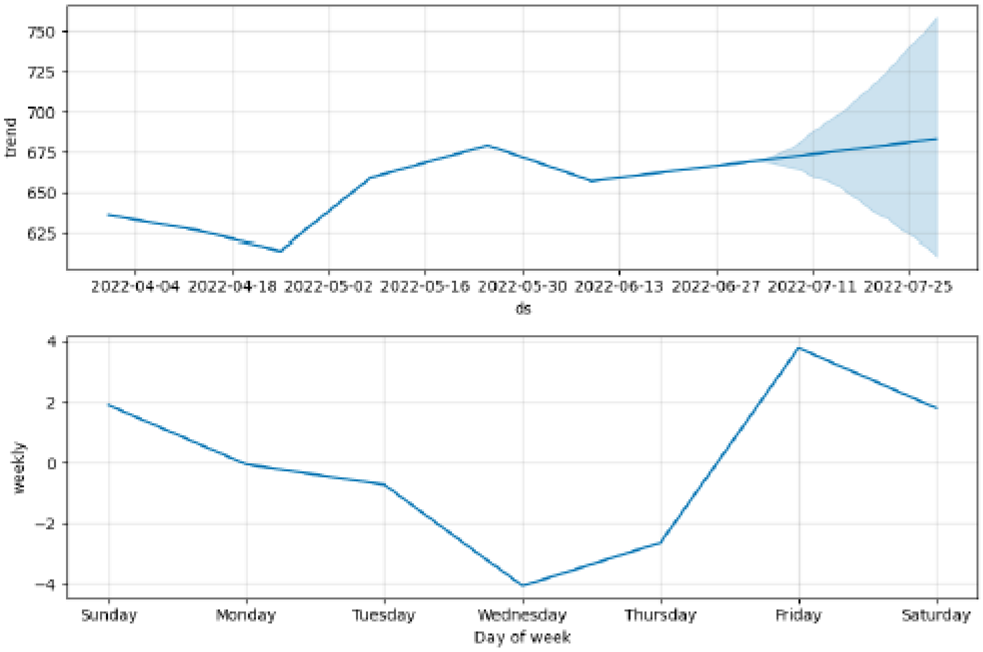


Figure 15: Forecast Component Plots (Tuned FB Prophet)

- **Tuned FB prophet performance evaluation:**

Table 3: Tuned FB Prophet Evaluation Matrix Results

Evaluation Matrix Name	**Results**
Mean Squared Error (MSE)	74965.26
Mean Absolute Error (MAE)	221.24
Root Mean Squared Error (RMSE).	273.79

5.2.3 Best FB prophet model:

- Both the basic FB prophet and tuned FB prophet model could not predict the e-commerce sales effectively. However based the tuned FB prophet could give predictable time series trend and the evaluation matrices result of tuned FB prophet model were better than the basic prophet. Hence it can be concluded that the tuned FB prophet is better to forecast e-commerce sales.

Table 4: FB Prophet Evaluation Matrix Results Comparison

Evaluation Matrix Name	**Basic FB Prophet**	**Tuned FB Prophet**
Mean Squared Error (MSE)	75379.90	74965.26
Mean Absolute Error (MAE)	222.04	221.24
Root Mean Squared Error (RMSE)	274.55	273.79

5.3. LSTM Results:

- **Seasonal Decomposition:**

  The seasonal decomposition here is used to break down the e-commerce time series sales data into seasonality, trend and residual. Considering below visualization the data shows trend but it is not consistent, the seasonality is also unpredictable and residual represent the error in the data. The seasonal decomposition plot does not contribute much in identifying the e-commerce time series trend or seasonality.

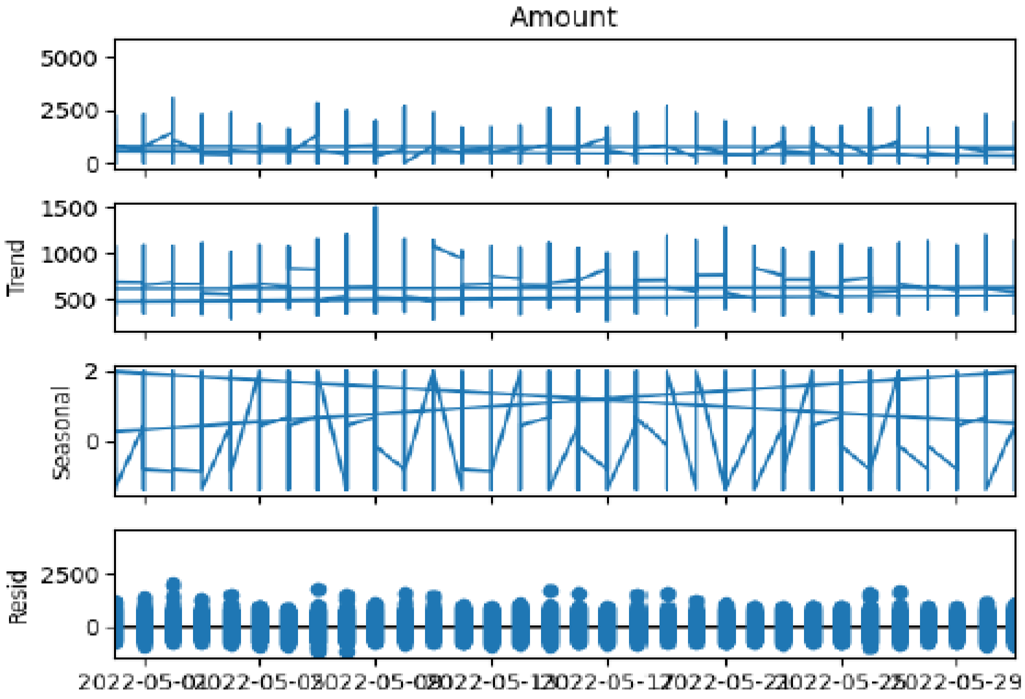


Figure 16: E-commerce Time Series Seasonal Decomposition

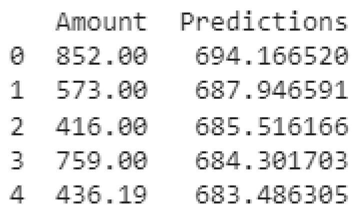
- **LSTM model best hyperparameters:**

  Below are the best hyperparameters for LSTM model. The model is trained using these hyperparameters and used to make predictions on the training set.

  LSTM layer: 50, Activation: 'tanh', Dense layer: 1, Epochs: 15.

- **LSTM forecasting results:**

  The below table displays the actual value of the test data and the predicted values after training the LSTM model. It is clearly visible that there is quite big difference in the actual and predicted values.



```
   Amount  Predictions
0  852.00   694.166520
1  573.00   687.946591
2  416.00   685.516166
3  759.00   684.301703
4  436.19   683.486305
```

Figure 17: LSTM Model Actual and Predicted Values

- Below plot is the visual presentation of actual and predicted values by the LSTM model. The blue part represents the actual amount values from the dataset while the orange line represents predicted values. It is clearly visible that the prediction are not following the actual values from the dataset. Hence the LSTM model could not accurately forecasted the e-commerce sales.

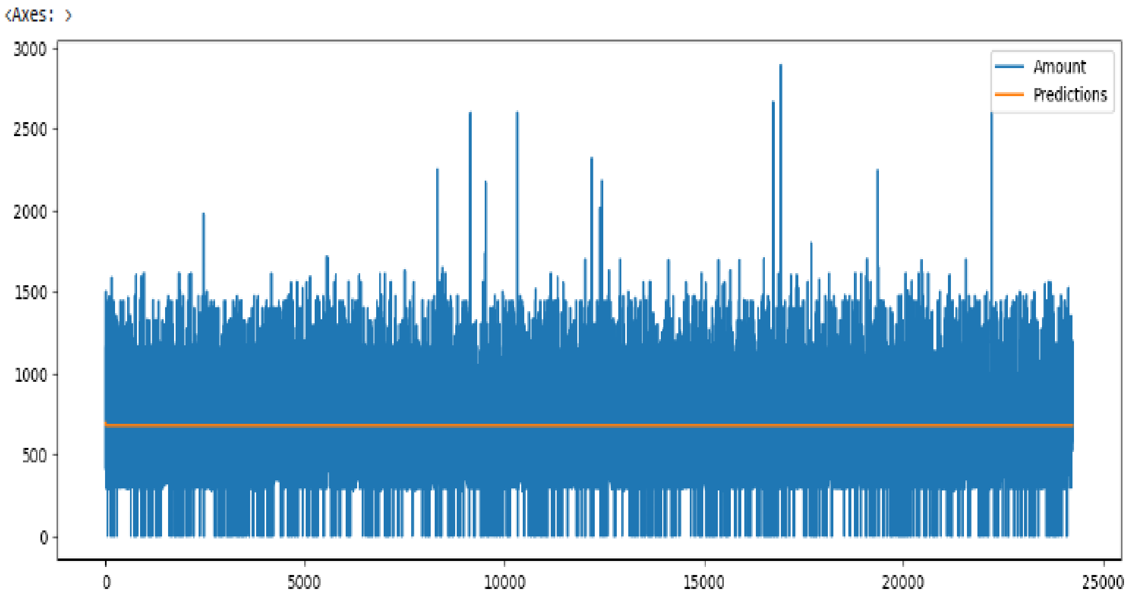


Figure 18: LSTM Forecasting Plot: Actual vs Predictions

LSTM performance evaluation:

Table 5: FB Prophet Evaluation Matrix Results Comparison

Evaluation Matrix Name	**Results**
Mean Squared Error (MSE)	83274.65
Mean Absolute Error (MAE)	225.00
Root Mean Squared Error (RMSE)	288.57

5.4. Model Comparison:

Below table gives comparative representation of evaluations matrices used to predict the performance of all the three machine learning algorithms selected for e-commerce sales forecasting. Considering comparative results of ARIMA, FB Prophet and LSTM, the FB Prophet model gives best result than the other 2 machine learning algorithms. Hence the FB Prophet is best machine learning algorithm for e-commerce sales forecasting considering evaluation matrices.

Table 6: Machine Learning Model Evaluation Matrix Results Comparison

Evaluation Matrix	**ARIMA**	**FB prophet**	**LSTM**
Mean Squared Error (MSE)	83930.84	74965.26	83274.65
Mean Absolute Error (MAE)	226.16	221.24	225.00
Root Mean Squared Error (RMSE)	289.70	273.79	288.57

Chapter 6 : Conclusion & Future Prospects

Finding and assessing the best forecasting machine learning models for e-commerce sales was the aim of this research work. Three models were taken into consideration: LSTM, Facebook Prophet, and ARIMA. Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) were the metrics used for the evaluation in addition to information obtained from the e-commerce time series sales trends.

ARIMA Model: Root Mean Squared Error (RMSE): 289.70, Mean Squared Error (MSE): 83930.84, Mean Absolute Error (MAE): 226.16. Though ARIMA was unable to fully capture the abrupt shifts that were seen in the first 15 days, ARIMA showed a respectable level of prediction ability. Out of the three models, it had the greatest RMSE, MSE and MAE.

FB Prophet Model: Root Mean Squared Error (RMSE): 273.79, Mean Squared Error (MSE): 74965.26, Mean Absolute Error (MAE): 221.24. Prophet showed stronger metrics and outperformed both ARIMA and LSTM. It gave significant insights for e-commerce sales data such as change in the trend over 15 days period, weekly seasonality was effectively caught, matching the largest sales on Fridays and the lowest sales on Wednesdays.

LSTM Model: Root Mean Squared Error (RMSE): 288.57, Mean Squared Error (MSE): 83274.65, Mean Absolute Error (MAE): 225.00. LSTM performed closely like ARIMA model and less good than the FB prophet. Though LSTM neural network based models can capture intricate temporal correlations of the complex data, it failed to capture the complex trend of the e-commerce time series sales data.

Concluding, the performance metrics of the three models (LSTM, Facebook Prophet, and ARIMA) are comparable, and none of them appear to be able to predict the e-commerce sales forecasting with a high degree of accuracy. However the FB prophet could give valuable insights those can be used by the sales experts to take informed business decisions and improve the e-commerce sales. A more complex strategy or feature engineering may be needed to capture the reported sharp shifts in the first fifteen days, which might be a difficult pattern for current models to represent. The precise reason for the models' limitations cannot be determined therefore, more research, adding more variables to the dataset, feature engineering, or testing more sophisticated models may be required.

Considering the shortcomings of the existing models, investigating techniques for detecting outliers or creating hybrid models that blend several forecasting philosophies might improve performance. A deeper comprehension of the unique features of the e-commerce sales dataset combined with iterative model building and improvement should result in increased forecasting accuracy. The research highlights the necessity of constantly improving and investigating different modelling techniques in order to provide more precise forecasts for the e-commerce sales dataset.

References:

1. Ali, Y. and Nakti, S., 2023, March. Sales Forecasting: A Comparison of Traditional and Modern Times-Series Forecasting Models on Sales Data with Seasonality. In 2023 10th International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 159-163). IEEE.
2. Hlupić, T., Oreščanin, D. and Petric, A.M., 2020, September. Time series model for sales predictions in the wholesale industry. In 2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO) (pp. 1263-1267). IEEE.
3. Vyas, R. and As, R., 2022, March. Seasonal Sales Prediction and Visualization for Walmart Retail Chain Using Time Series and Regression Analysis: A Comparative Study. In 2022 International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN) (pp. 1-6). IEEE.
4. Singh, B., Kumar, P., Sharma, N. and Sharma, K.P., 2020, January. Sales forecast for amazon sales with time series modeling. In 2020 first international conference on power, control and computing technologies (ICPC2T) (pp. 38-43). IEEE.
5. B. Kumar Jha and S. Pande, "Time Series Forecasting Model for Supermarket Sales using FB-Prophet," *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, 2021, pp. 547-554, doi: 10.1109/ICCMC51019.2021.9418033.
6. Bajoudah, M. Alsaidi and A. Alhindi, "Time Series Forecasting Model for E-commerce Store Sales Using FB-Prophet," *2023 14th International Conference on Information and Communication Systems (ICICS)*, Irbid, Jordan, 2023, pp. 1-6, doi: 10.1109/ICICS60529.2023.10330530.

7. L. Yan, "Smartwatch Sales Forecast Based on CNN-LSTM," *2023 IEEE 6th Information Technology,Networking,Electronic and Automation Control Conference (ITNEC),* Chongqing, China, 2023, pp. 1034-1038, doi: 10.1109/ITNEC56291.2023.10082702.
8. X. Li, J. Du, Y. Wang and Y. Cao, "Automatic Sales Forecasting System Based On LSTM Network," *2020 International Conference on Computer Science and Management Technology (ICCSMT),* Shanghai, China, 2020, pp. 393-396, doi: 10.1109/ICCSMT51754.2020.00088.
9. T. Tang, "Analysis and Demand Forecasting Based On e-Commerce Data," *2023 6th International Conference on Artificial Intelligence and Big Data (ICAIBD)*, Chengdu, China, 2023, pp. 64-68, doi: 10.1109/ICAIBD57115.2023.10206072.
10. S. Prakash, A. S. Jalal and P. Pathak, "Forecasting COVID-19 Pandemic using Prophet, LSTM, hybrid GRU-LSTM, CNN-LSTM, Bi-LSTM and Stacked-LSTM for India," *2023 6th International Conference on Information Systems and Computer Networks (ISCON),* Mathura, India, 2023, pp. 1-6, doi: 10.1109/ISCON57294.2023.10112065.
11. Shi, R. and Zhang, C., 2023, August. A study of sales forecasting in multinational retail companies: a feature extraction-machine learning-classification based forecasting framework. In *2023 IEEE International Conference on Sensors, Electronics and Computer Engineering (ICSECE)* (pp. 401-405). IEEE.)