DEMAND FORECASTING IN E-COMMERCE

A PROJECT REPORT

Submitted by

YUVASHREE.R [REGISTER NO:211422104565]

THIRUMALAVIKA.S [REGISTER NO:211422104520]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE,

CHENNAI- 600123.

(An Autonomous Institution Affiliated to Anna University, Chennai)

OCTOBER 2024

BONAFIDE CERTIFICATE

Certified that this project report **“DEMAND FORECASTING IN E-COMMERCE”** is the bonafide work of “**YUVASHREE.R (211422104565), THIRUMALAVIKA.S (211422104520)”** who carried out the project work under my supervision.

SIGNATURE

Dr.L.JABASHEELA,M.E.,Ph.D.,
PROFESSOR
HEAD OF THE DEPARTMENT

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING
COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

SIGNATURE

DR.T.TAMILVIZHI, M.E.,Ph.D.,
PROFESSOR
SUPERVISOR

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING
COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

Certified that the above candidates were examined in the End Semester Project

Viva-Voce Examination held on..........................

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We YUVASHREE.R (211422104565), THIRUMALAVIKA.S (211419104220) hereby declare that this project report titled "**DEMAND FORECASTING IN E-COMMERCE**", under the guidance of DR.T.TAMILVIZHI, M.E.,Ph.D., is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

1. **YUVASHREE.R**

2. **THIRUMALAVIKA.S**

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere and hearty thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHI KUMAR,M.E.,Ph.D** and **Dr.SARANYASREE SAKTHI KUMAR B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. L.JABASHEELA , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my Project Guide **DR.T.TAMILVIZHI, M.E.,Ph.D.,** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

YUVASHREE R

THIRUMALAVIKA S

ABSTRACT

The report presents the methodologies employed in developing the demand forecasting system. The primary goal of the undertaking is to improve the aspects of inventory control and production scheduling through means of sales history and other factors that affect the sales of the product. Three models namely Dynamic Regression, Multivariate Regression and Time Series Analysis were adopted to determine the best model for application in this case. The Dynamic Regression model provides for the use of outside variables and can adjust the forecast according to seasonality factors and market conditions as well. Unlike earlier models that predicted a better future performance, this model showed the flexibility to adapt to changes in demand drivers. The Multivariate Regression approach uses more than one predictor variable, which is economic measurements and prior sales data works to account for demand forces and make a better decision. They also used Time Series Analysis to record cycle and seasonality within sales data of the products in the past. This model incorporated the use of tools like the Autoregressive Integrated Moving Average (ARIMA) in order to create accurate forecast requirements hence making certain that a proper understanding of the demand variation over time was made. During the project implementation proper data cleaning and preparation was completed to maintain the forecasting models' accuracy. Other criterion include Mean Absolute Percentage Error (MAPE) was used to determine the level of accuracy of forecast in a bid to determine the most suitable forecasting techniques. Therefore, this paper argues that the implemented demand forecasting system is a crucial tool for the cardboard manufacturing company, as it optimizes supply chain execution and supports tactical decision making based on accurate demand estimates. The combined forecast approach of the system ensures a broad solution that can suitably capture product demand patterns hence making the general vision of the company, to minimize wastage, improve inventory and satisfy the customers.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The paper describes the process of implementing demand forecasting at the cardboard manufacturing company using the following methods. The major goal of this project is to improve inventory control and manufacturing forecasting with the help of historical sales data and other factors affecting sales in the future. To achieve this, three different approaches were explored: Measurements included Dynamic Regression, Multivariate Regression, and Time series analysis. All of them focus on finding the most suitable approach to manage demand forecasting in manufacturing environment, yet including seasonal characteristics and other factors.

This makes demand forecasting an invaluable component of contemporary enterprise management since it can make a crucial contribution to correct evaluations of required stocks, thus reducing potential costs associated with direct material management. This project deals with enhancing the impact of demand forecasting by using aspects such as sales history, economic factors, and other factors. The Dynamic Regression method involves external factors and rebases forecasted values in relative to the trends that include factors such as seasonality and the market trends hence appropriate in situations that show that star effects from the outside world have a very notable effect on the sales. The Multivariate Regression technique employs several predictor variables like economic measures and prior sale’s record that helps in forecasting intricate relations between factors and demand and so is beneficial in decision-making. Finally, Time Series Analysis focuses on past data trends of sales volume in an attempt to predict future sales volume This method involves use of

ARIMA in modeling trends and seasonality.

These forecasting techniques have real-world applications, such as:

- Improving the scheduling of production to effectively meet most of the customer demands.
- Absorbed; that is, it effectively reduces the costs relating to too much inventory or immediate restocking.
- Helping to support the ability to make strategic decisions in the supply chain.

This paper demonstrates the steps taken, the findings that were made when using these models to analyze the cardboard manufacturing company and how to offer a systematic method for demand forecasting.

1.2 PROBLEM DEFINITION

The forecasting of future demand is a major problem in the cardboard manufacturing industry because of the instability of the market situations, climatic changes, and other circumstances. Some of the mismanagement risks caused by poor demand forecasting are; overstocking, outright stock out, high stocking expense, and missed sales. All these issues affect not only the logistics or performance but also the economic profitability of the firm. At the moment, the cardboard manufacturing company still has no adequate system in place as a way of predicting demand in the production and marketing of the manufactured products. For this reason, the company experiences a lot of difficulties in achieving the optimal volume of product stock, satisfying customers' needs when they need it, and excluding extra expenses on storage and immediate replenishment. In order to overcome these challenges, an integrated demand forecasting system must be established that need to involve the use of techniques such as analysis of big data. Combining historical sales data, economic variables, and other parameters the goal of creating a prognosis model for future demand to increase the effectiveness of inventories control and production .

CHAPTER 2

LITERATURE SURVEY

The coupling of incentive design and differential privacy is suggested to have a great potential in boosting the capability of demand forecasting in e-commerce. Given in the model incentives aspects can help to force data owners and interested parties to contribute key data that enhance sales forecast precision. In parallel, asserting differential privacy could protect the contribution of individual data and enhance the security of the model, therefore suitable for sensitive data settings. This approach suits the user requirements of e-commerce turnover applications since reliable and ethical sales forecasting necessitates high-quality and privacy preserving data[7].

An approach known as H-FL together with ConvLSTM to forecast the demand of the e-commerce products since such systems are vulnerable to the bullwhip effect and privacy issues. In turn, it promotes the sharing of data, which usually helps improve the accuracy of a forecast without compromising privacy. Compared to the basic models such as LSTM and BiLSTM, the proposed model minimizes the loss of efficiency in the supply chain while considering the development of sustainable e-commerce. This work presents a practical solution for improving demand forecasting based on federated learning in large scale and insecure environments, consequently establishing the theme for future studies of federated learning.[15]

The paper discusses the use of Federated Learning (FL) as a promising solution to address privacy and scalability issues associated with traditional centralized machine learning models. FL allows IoT devices to collaboratively train a global model without sharing raw data, thus enhancing data privacy and reducing computation costs. The authors propose a novel incentive mechanism that integrates Mechanism Design (MD) and Differential Privacy (DP) to encourage active participation from data owners while ensuring privacy. By combining these techniques, the paper aims to optimize the performance of FL systems in dynamic environments, striking a balance between privacy preservation and system efficiency. Experimental results indicate that the proposed method achieves superior device participation, system throughput, and learning performance compared to existing FL protocols[9].

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Presently, e-commerce firms usually carry out their demand forecasting in a basic way through the use of he sales history and simple mathematical formulae. Regular techniques like moving averages or a seasonal pattern are often employed in sales projection. These methods often do not consider things like advertisements, changes in the occasion, or competitor’s action thereby making predictions less accurate. Further, most of the current systems do not support real time forecasting of market conditions and hence the business cannot easily change its strategies to meet changing market conditions. As useful as such conventional tools may be in establishing some of the simplest trends, they are inadequate in capturing more intricate trends in sales data as observed in e-commerce applications.

3.2 PROPOSED SYSTEM

In this approach, a model is created to predict high accurate demand for a cardboard manufacturing company. The model integrates three advanced statistical methods: Dynamic Regression used for analyzing high volume of previous sales data, Multivariate Regression is also used for analyzing the historical sales data and the Time Series Analysis will be helpful in identifying the external influencing factors. The model also takes external information that includes the economic factors and the market environment so adjustments for the model’s forecast can be made. This model therefore plays particular importance in establishing the possible changes in demand as a result of shift in market forces. On the other hand, in the Multivariate Regression model multiple independent variable values are employed to draw inference from economic figures as well as from previous sales. To this end, they proposed a model for demand analysis that discusses the interrelation of the aforementioned variables in an attempt to better understand demand drivers.

While the Time Series Analysis centers on the historical sales information and trends and seasonality are used to coming up with short term forecast. The chief advantage of this method is that is serves well to capture the fluctuations characteristic of product demand. The suggested system has a theoretical background based on statistical hypothesis with added mathematical models for improved certainty. To ensure highly effective and reliable demand forecasts are incorporated into the framework, the greatest probability of the actual sales figures is sought after. The system also provides for model evaluation by performance measures, as well as its flexibility and ability to change in response to fluctuations in market conditions. In conclusion, this proposed system aims at increasing the cardboard manufacturing company's decision-making capability regarding inventory and production schedules to achieve improved company efficiency.

3.3 FEASIBILITY STUDY

As with most feasibility studies, the goal is to address the problem of demand forecasting not only to meet the need but to understand its magnitude as well. In this research, the problem statement was refined and the main characteristics of the forecasting system to be incorporated have been established. Therefore, at this stage, benefits are expected and estimated quantitatively with better precision. The key Considerations are

Economic Feasibility:

Based on the findings obtained from the economic assessments not only the costs related to the acquisition of the hardware and software are included but also the expected improvements regarding the management of inventories and reduction of stockouts. The put in place of this demand forecasting system will result into reduction in the amount of time and rigidity used in manual overall forecasting process thus enhancing the operating cycle and accuracy in decision making.

Total= 450

KLOC = 450/1000 = 0.450

Effort = 2.4 * (0.450)^1.05 = 1.017 person-month

Development time = 155.5 = 2.5 * 1.017^0.38 = 2.527 months

Average staff size= 1.017/ 2.527= 0.402 staff

Their productivity was 0.450 KLOC/person-months divided by 1.017

KLOC/person-months = 0.442 KLOC/person-months.

P = 442 LOC/person-months

Technical Feasibility:

Technical feasibility looks at the hardware endowment, software technology, and human resource in as far as they can support the abilities needed to contain the forecasting models.

Dynamic Regression Algorithm, Multi-Regression Algorithm, Time Series Analysis

Software Tools - R, Python

IDE - RStudio, Google Colab

Data Storage : Cloud Storage Solutions (Examples Google Drive)

Social Feasibility:

Social feasibility can be described as an analysis of the relationship that is created between the new forecasting system being introduced and the work environments already in practice in the company. This is what a social impact analysis is meant to do; it focuses on such an impact with a view of dissecting its extent and ambit in the Furtherance of the project's objectives. This analysis really minimizes the total risks of the project as it also limits and enhances the resistance to the system.

Better Decision-Making – The provision of information to managers as a tool in their decision making process.

Training as Implication of System Implementation – The staff is going to need training when it comes to using the forecasting system.

Better Customer Satisfaction – This should help customers since deliveries will now be made in the shortest time possible since the inventory is well managed.

Increased Sales Forecast Accuracy – Increased employee participation from sales,.

3.4 DEVELOPMENT ENVIRONMENT

Hardware Requirements

Processor :Intel Core i5

RAM : 512 MB and above

Hard Disk : 40 GB and above

Software Requirements

Programming language: PYTHON Technology: Deep Learning

Operating System : Windows 7

Tools: Anaconda Navigator /TensorFlow/Jupyter/Google colab

CHAPTER 4

SYSTEM DESIGN

4.1 FLOW DIAGRAM

This project requires a dataset which have Quantity of sales, External predictors such as google clicks and Facebook Impressions. The dataset should be able to train the forecasting models.

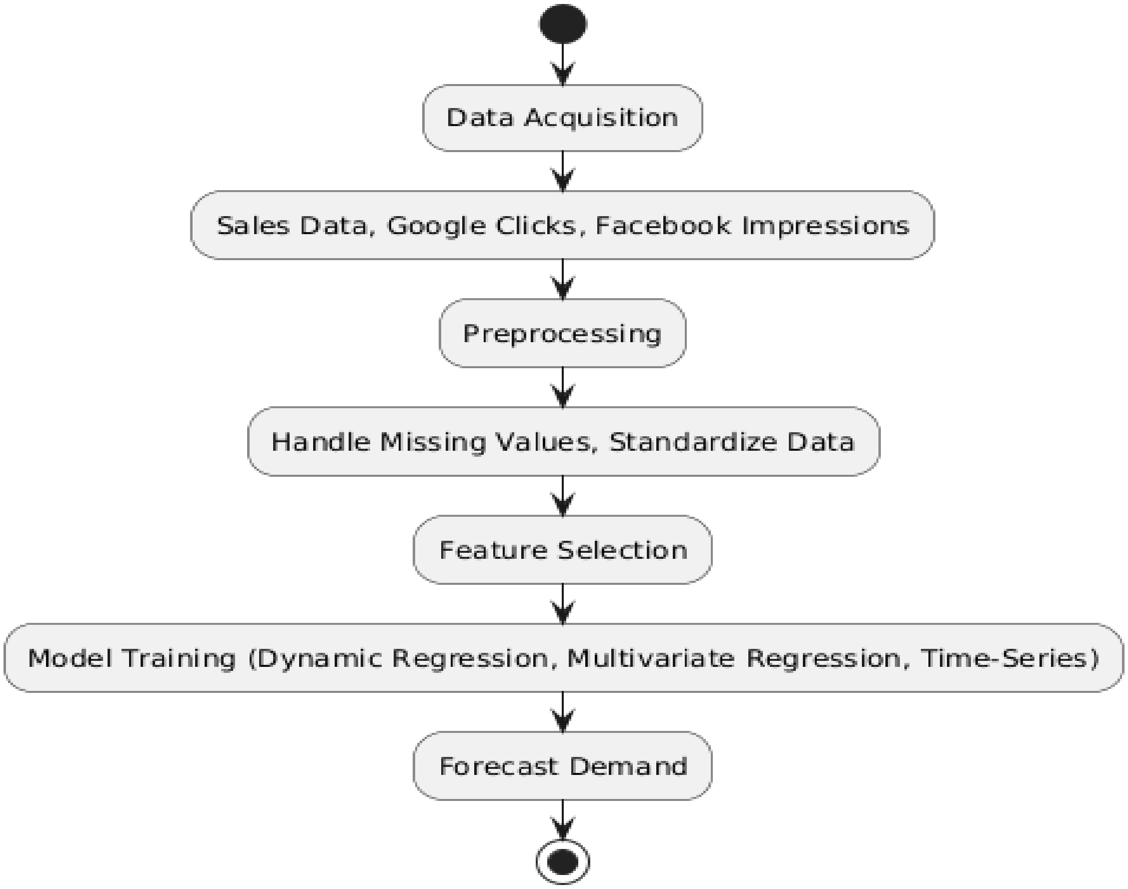

Data Acquisition
Sales Data, Google Clicks, Facebook Impressions
Preprocessing
Handle Missing Values, Standardize Data
Feature Selection
Model Training (Dynamic Regression, Multivariate Regression, Time-Series)
Forecast Demand

Fig. 4.1 Working flow of the model

4.2 DATASET

ProductA.csv

Time Frame: Controlling from December 1, 2021 to June 30, 2022 which is 212 days.

Key Stats:

Average Daily Quantity: ~17 units

Max Quantity: 38 units (Feb 17, 2022)

Min Quantity: 5 units (Feb 24, 2022)

Trends:

Peak Periods:

Dec 2021 to Jan 2022: A relatively stable sales level with occasional and not very high spikes (22–33 in late December).

Feb 2022: Higher volatility is observed with small fluctuations up to February 17th when 38 units of this type of bicycles were sold.

April 2022: Increased sales at month end to a maximum of 36 units.

Low Periods:

2/24/2022 and 5 units sold

Low Sales do any three slow selling days; May and June.

Insights:

Seasonality: The numbers drawn from the dataset expose some moments when sales experimented a slight increase, which could be associated with the influence of factors from the external environment for example, marketing and promotions.

ProductA_google_clicks.csv

Time Frame: From December 1, 2021, to June 30, 2022, inclusive (212 days).

Key Stats:

Average Clicks: ~135/day

Max Clicks: 726 (Feb 21, 2022)

Min Clicks: 60 (June 29, 2022)

Trends:

Peak Period: During February – April 2022, with the number of clicks recorded sometimes surpassing 500.

Low Period: Average number of clicks registering decreased from late may to June 2022.

Anomalies: A significant increase on February 21st, 2022 with 726 clicks.

Insights:

Seasonal Patterns: Relatively apparent seasonality, where possibly their traffic spikes can be attributed to external circumstances such as campaigns.

ProductA_fb_impressions.csv

Time Frame: Since December 1, 2021, to June 30, 2022, (212 days).

Key Stats:

Average Daily Impressions: ~1485 impressions

Max Impressions: 2707 distinct impact (February 14, 2022).

Min Impressions: 620 impressions (Dec 1, 2021)

Trends:

Peak Periods:

Dec 2021 to Jan 2022: Ascending tendency of the impressions with multiple spikes in the range of 1814-2549 impressions in the end of December beginning of January.

Feb 2022: Large bar rises, especially on average in the middle of February – February 14, with 2,707 impressions.

March 2022: Daily impression is fairly steady, and reaches its maximum number at

2572 on the March 20.

Low Periods:

June 2022: Decreasing in the middle and towards the end of the month, getting down to 865 impressions in the middle of the last week.

April and May 2022: Certain unpredictable lows mainly towards the end of these months.

Insights:

Seasonality: Increase in number of Facebook impression may indicate that there where some special campaigns or events such as February and so on.

4.3 DATA PREPROCESSING:

The **Dynamic Regression** model, ARIMA, comes next on the cleaned time series data by using stationarity tests such as the ADF, PP and KPSS to tell whether differencing should be done. An appropriate ARIMA model is obtained using the auto.arima() function which selects appropriate values of p, d and q according to AIC / BIC criteria. Forecasting is then performed and provide the predicted sales for the duration of the test set, and model performance is evaluated using Mean Absolute Percentage Error (MAPE). Last but not the least, visualizations are created to compare actual and forecasted sales in order to assess how accurately the model can make the forecast in reality.

The earlier steps necessary while working with **Multivariate Regression** structure involves several tasks of data preparation triggering pre-processing. First, four sets of data: sales quantity data, Google click data, and Facebook impression data are read from CSV files. The data is next cleaned where volatility is dealt with using moving average that is applied in a window form. Data normalization is applied in order to bring Google clicks and Facebook impressions data into a [0, 1] range, thereby improving comparison. Next, at the end of cleaning step, new data frame is established and data is molded in a structured format for sales, clicks and impressions then the data is divided in to train set(90%) and test set(10%).

In the **Time-Series Analysis**, exactly the same way as in the Linear Regression implementation, the sales quantity data is loaded from the CSV file and partitioned into training (80%) and test (20%) data. Time series characteristics of the training data are used to create a time series object for time series modeling with the start of time and the frequency. In order to improve the data quality similar to above analysis, sliding window technique is applied again to clean up the sales data and remove noise to find trends. Initial data exploration is done with regard to seasonality and trends via added visualization components such as an ACF and PACF diagrams.

CHAPTER 5

SYSTEM ARCHITECTURE

5.1 ARCHITECTURE OVERVIEW

There are several components that are in the architectural design of the demand forecasting project about e-commerce. Raw data captured by such systems include sales records, number of Google clicks, and impressions in Facebook, which the Data Ingestion Layer acquires and stores. After this, the Data Preprocessing Layer prepares the above data by means of smoothing transformation, normalization, and conversion of data into training and testing data. The Modeling Layer uses multivariate regression and time series analysis and the ARIMA model to predict sales. The Evaluation Layer measures accuracy through EMAPE and ERMSE checking and graphs actual as well as predicted sales.

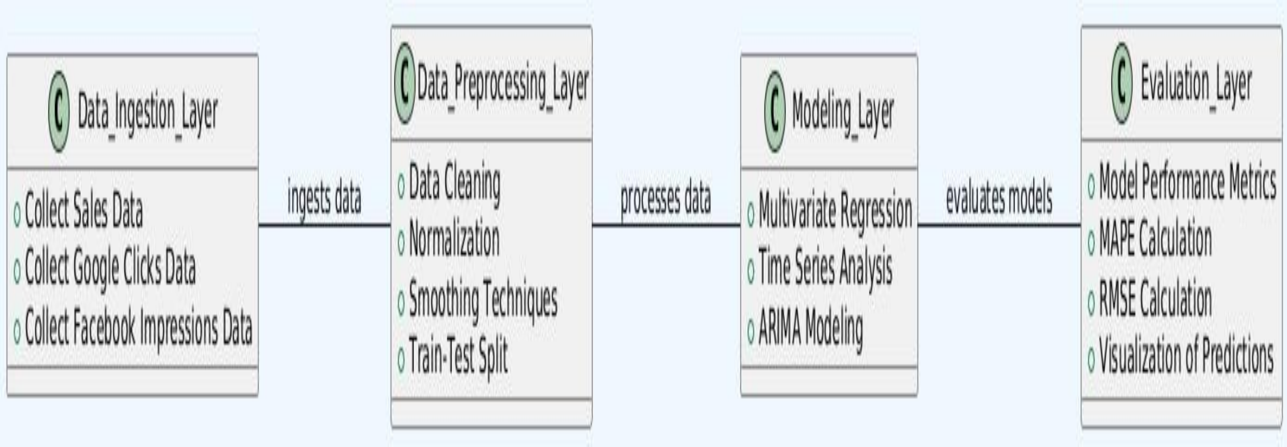


Fig.5.1 Architecture overview

5.2 MODULES

Dataset overview

Measuring the Product A sales data from December 1, 2021, to June 30, 2022, gives an approximate average daily stock out of 17 units with a high of 38 units on February 17, 2022. A relatively constant level of sales was observed between the periods December and January with the fluctuating pattern in February and at the end of April. They were low on some days in February and the whole of May and June.

The datasets used for the analysis include Product A Sales Data in the format **ProductA.csv**

DATE	QUANTITY
2021-12-01	15
2021-12-02	17
2021-12-03	16
2021-12-04	20
…	…
2022-06-29	10
2022-06-30	12

Table 5.1: ProductA.csv

Regarding the Product A Google clicks, there was an average of 135 clicks per day; the highest number of clicks detected was 726 on February 21, 2022 and the main increase in the number of clicks was observed between February to April. Nevertheless, they have decreased from end of May to early June meaning that people have reduced their interaction with the sites.

Product A Google Clicks Data file is named as **ProductA_google_clicks.csv**.

DATE	**CLICKS**
2021-12-01	130
2021-12-02	145
2021-12-03	120
2021-12-04	155
…	…
2022-06-29	110
2022-06-30	115

Table 5.2: ProductA_google_clicks.csv

Similar to those of Product A's Facebook impressions; with approximately 1,485 daily impressions and the daily impression reaching the highest value of 2,707 on 14/02/2022. Total impressions have gradually grown from December- January, slight rise was seen in February and there was a slight drop in June. Such trends may correlate with seasonal promotion or participation activities and/firm events.

For Product A there is the data set entitled, '**ProductA_fb_impressions.csv**,' and contains the number of impressions on Facebook.

DATE	QUANTITY
2021-12-01	620
2021-12-02	700
2021-12-03	740
2021-12-04	780
…	…
2022-06-29	860
2022-06-30	880

Table 5.3:ProductA fb impressions.csv

Data Preprocessing

The project involves three main modeling approaches: Dynamic regression incorporating Auto Regression Integrated Moving Average, regression and Time series Analysis. In the case of the ARIMA model, the stationarity tests (ADF, PP and KPSS tests) are carried out for purpose of determining whether differencing is necessary. The auto.arima() function selects appropriate values of (p, d, q) with the help of AIC/BIC standards and further conducts a forecast for estimating the magnitude of the sales during the test period adopted from Cozer and Beckman (2006) using Mean Absolute Percentage Error (MAPE). The forecasts of this indicator are then graphed and compared with the original values.

Four data sets (sales quantity, number of Google clicks, number of Facebook impressions) are read from the CSV files in the Multivariate Regression structure. Some data cleaning work includes handling volatility through moving average and normalizing the Google and Facebook data to [0, 1] range. After cleaning the new structured data frame is build and the dataset is divided into training set 90% and test set 10%.Like for Time-Series Analysis, the sales quantity data is pre-loaded and then split into training set (80%) and test set (20%). As for the training data the time series object is generated with noise reduction and trend identification made with the help of the sliding window technique. A first step …[involves]…residual plots, preliminary data visualization of shapes, ACF and PACF diagrams for evaluating seasonality and trends.

Project File Structure

.RData: Includes the results of your R commands, including, loaded variables that are in the R environment, data frames etc.

.Rhistory: Remembers the sequence of commands you have run in the R environment.

.Rprofile: An R startup file is one from which you want custom configurations or settings to run each time you launch R.

Demand_Forecasting.Rproj: the master file of the workspace that uses and RStudio for handling the paths and setting will be created in this project.

ProductA_Dynamic_Regression.R: A document specifically on analysis using R script specifically the dynamic regression analysis in Product A.

ProductA_fb_impressions.csv: A CSV file which has the banner impressions data of Facebook for Product A.

ProductA_google_clicks.csv: Data in the format of a CSV file containing the click through rates from the Google Ads for Product A.

ProductA_Multivariate_Regression.R: Multivariate regression analysis R script of Product A.

ProductA_timeseries.R: An R script kind of time series analysis for Product A.

ProductA.csv: A CSV file contains the main product demand or sales data for a part.

Defining the model

Dynamic Regression:

This model is a blend of the standard regression analysis model and the time series model. It uses past demand, in conjunction with other predictors such as google clicks and face book impressions to predict future demand. The dynamic aspect is due to the current values for lagged measures or first differences of predictors.

Multivariate Regression:

Linear regression model is one type of regression analysis which utilizes more than one independent variable (or predictor) to estimate the dependent variable (or sales). In your project, such measures like Google clicks and Facebook impressions are employed to predict the number of sales. It quantifies the nature of the relationship between these variables and the demand which gives a platform for forecasting following changes in conditions.

ARIMA (AutoRegressive Integrated Moving Average):

ARIMA stands for Auto regressive integrated moving average and it is a method of forecasting data which is in form of time series. It captures the patterns in the data by combining three components:

AutoRegression (AR): A process of making forecasts, the use of prior values to pertain current and future values.

Integrated (I): Using differencing to one step in order to obtain stationary data where the mean and variance is constant.

Moving Average (MA): Corrects forecast errors for future use of the model.

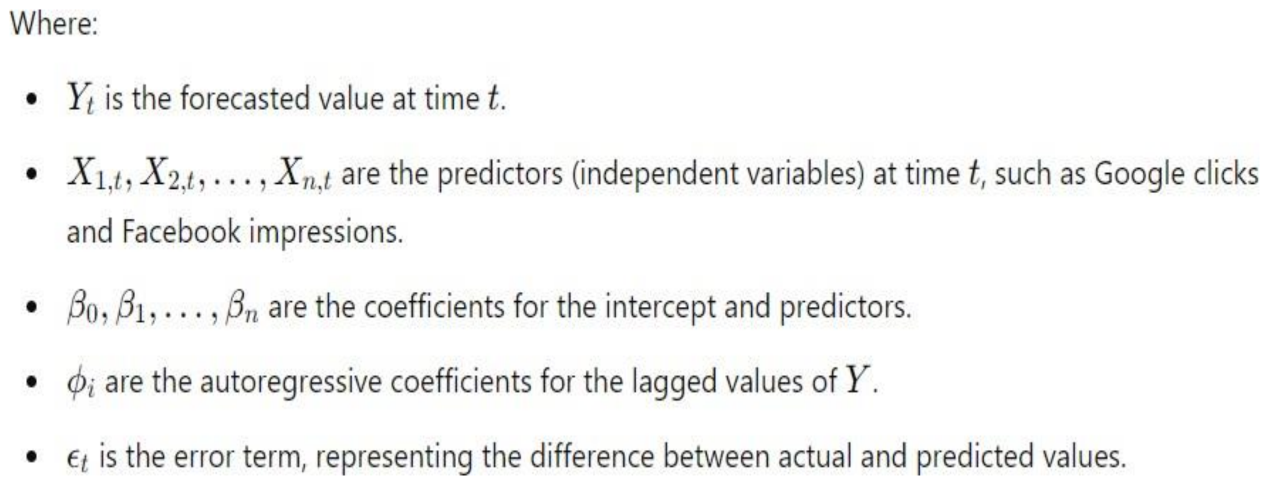
5.3 ALGORITHMS

Dynamic Regression

Dynamic Regression is an amended version of massive regression that involves time varying parameters and specified delay variables. It is used in cases where the aim is to analyze a relationship between a dependent variable an a set of predictors, where there is also the aspect of time.

Formula:

$$Y_t = \beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t} + \ldots + \beta_n X_{n,t} + \sum_{i=1}^{p} \phi_i Y_{t-i} + \epsilon_t$$



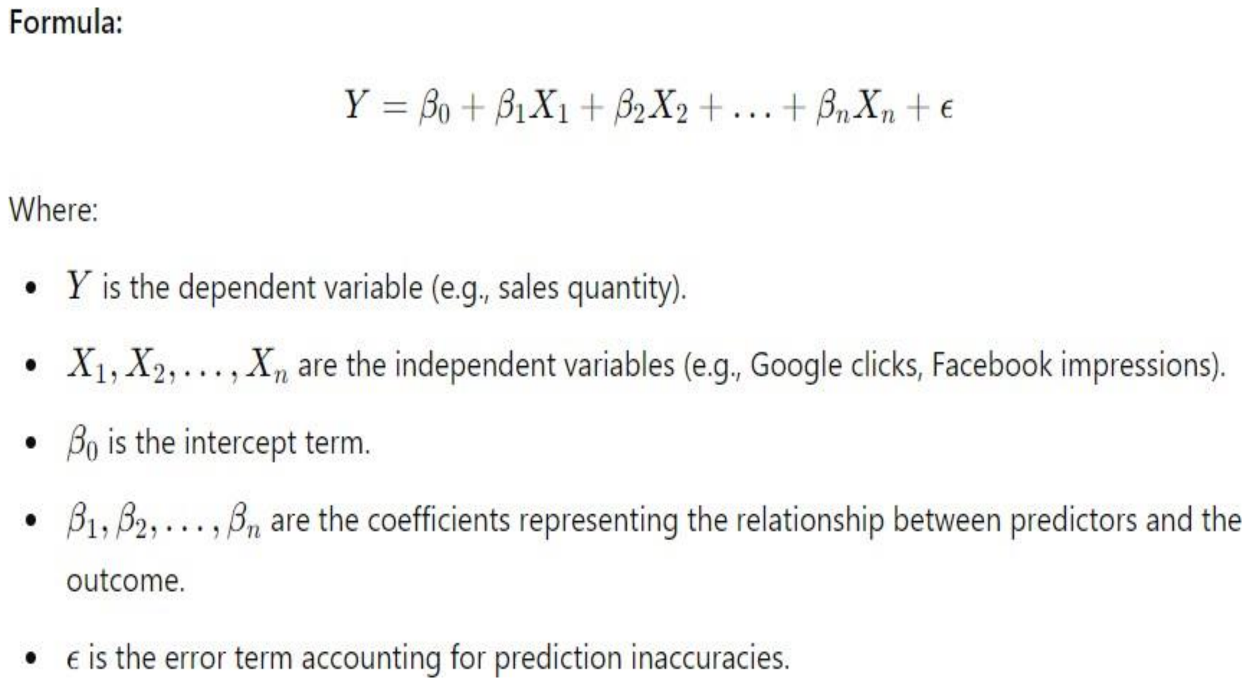
Where:

- Y_t is the forecasted value at time t.
- $X_{1,t}, X_{2,t}, \ldots, X_{n,t}$ are the predictors (independent variables) at time t, such as Google clicks and Facebook impressions.
- $\beta_0, \beta_1, \ldots, \beta_n$ are the coefficients for the intercept and predictors.
- ϕ_i are the autoregressive coefficients for the lagged values of Y.
- ϵ_t is the error term, representing the difference between actual and predicted values.

Another added advantage of Dynamic Regression is that lagged predictors can be incorporated into a model in order to model the impacts of time-lagged variables on dependent variable.

2. Multivariate Regression

The outcome variable is then predicted in the Multivariate Regression where more than one predictors or independent variable are used. This model identifies associations between dependent variable and independent variables (predictors); thereby, can be used to predict market demand with extraneous variables.



Formula:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \epsilon$$

Where:

- Y is the dependent variable (e.g., sales quantity).
- $X_1, X_2, \ldots, X_n$ are the independent variables (e.g., Google clicks, Facebook impressions).
- β_0 is the intercept term.
- $\beta_1, \beta_2, \ldots, \beta_n$ are the coefficients representing the relationship between predictors and the outcome.
- ϵ is the error term accounting for prediction inaccuracies.

The coefficients are estimate using method such as Ordinary Least Squares (OLS) whose aim is to find the set of coefficients that will minimize the total of squared residuals.

3. ARIMA

Stands for AutoRegressive Integrated Moving average

ARIMA is an approach towards time series forecasting. ARIMA is a forecast model that integrates attributes of AutoRegression (AR), Integration (I), and Moving Average (MA) models to define trends of the data.

3.1. AutoRegression (AR) Component:

The AR part requires the variable to be regressed on past values of itself. The AR model of order p represented as:

$$Y_t = c + \sum_{i=1}^{p} \phi_i Y_{t-i} + \epsilon_t$$

Where:

- Y_t is the value at time t.
- c is a constant.
- ϕ_i are the autoregressive coefficients.
- ϵ_t is the error term.

3.2. Integrated (I) Component:

This particular part of an integration transforms the time series to be stationary by differencing it. First-order differencing can be expressed as:

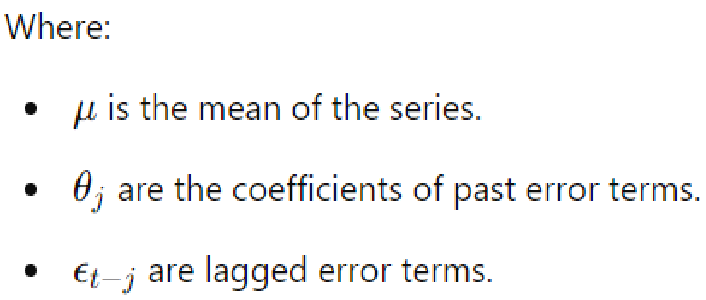
$$Y_t' = Y_t - Y_{t-1}$$

Where Y_t' is the differenced series.

3.3. Moving Average (MA) Component:

Through the use of past forecast errors, the MA part of the algorithm can make better predictions. Let ma model of order be represent q as:

$$Y_t = \mu + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} + \epsilon_t$$



Where:

- μ is the mean of the series.
- θ_j are the coefficients of past error terms.
- ϵ_{t-j} are lagged error terms.

3.4. Complete ARIMA Model:

Combining AR, I, and MA components gives the ARIMA model:

$$Y_t = c + \sum_{i=1}^{p} \phi_i Y_{t-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} + \epsilon_t$$

Differencing d-times is applied to achieve stationarity, where d indicates the order of differencing.

5.4 UML DIAGRAMS:

CLASS DIAGRAM

This diagram reflects the main components and relationships for a demand forecasting project, capturing data management, model training, prediction, evaluation, and visualization**.**

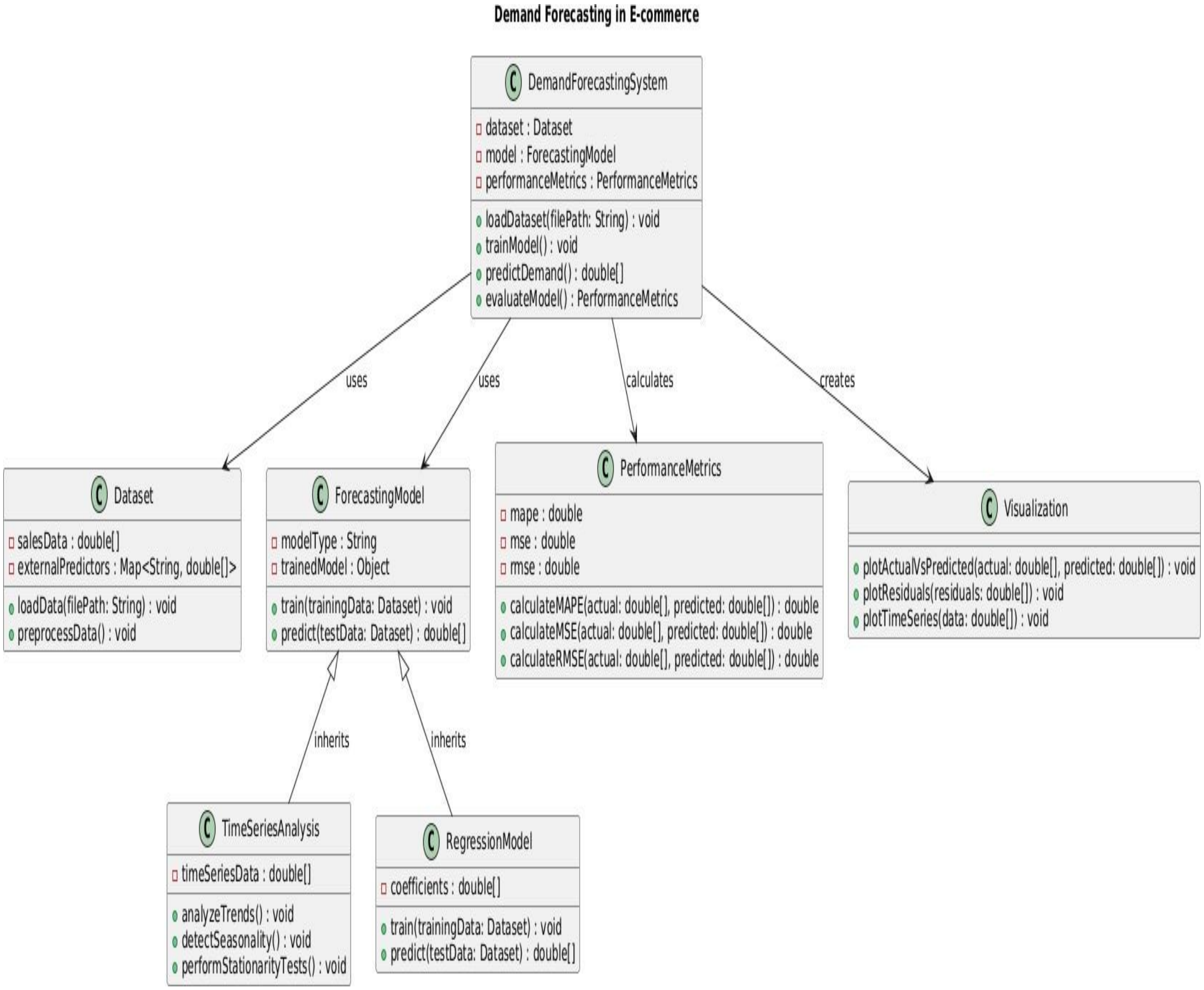

Demand Forecasting in E-commerce
DemandForecastingSystem
dataset : Dataset
model : ForecastingModel
performanceMetrics : PerformanceMetrics
loadDataset(filePath: String) : void
trainModel() : void
predictDemand() : double[]
evaluateModel() : PerformanceMetrics
uses
uses
calculates
creates
Dataset
salesData : double[]
externalPredictors : Map<String, double[]>
loadData(filePath: String) : void
preprocessData() : void
ForecastingModel
modelType : String
trainedModel : Object
train(trainingData: Dataset) : void
predict(testData: Dataset) : double[]
PerformanceMetrics
mape : double
mse : double
rmse : double
calculateMAPE(actual: double[], predicted: double[]) : double
calculateMSE(actual: double[], predicted: double[]) : double
calculateRMSE(actual: double[], predicted: double[]) : double
Visualization
plotActualVsPredicted(actual: double[], predicted: double[]) : void
plotResiduals(residuals: double[]) : void
plotTimeSeries(data: double[]) : void
inherits
inherits
TimeSeriesAnalysis
timeSeriesData : double[]
analyzeTrends() : void
detectSeasonality() : void
performStationarityTests() : void
RegressionModel
coefficients : double[]
train(trainingData: Dataset) : void
predict(testData: Dataset) : double[]

Fig 5.2 Class Diagram for Demand Forecasting

ACTIVITY DIAGRAM:

This diagram captures the key activities in the demand forecasting process, providing a structured overview of the workflow.

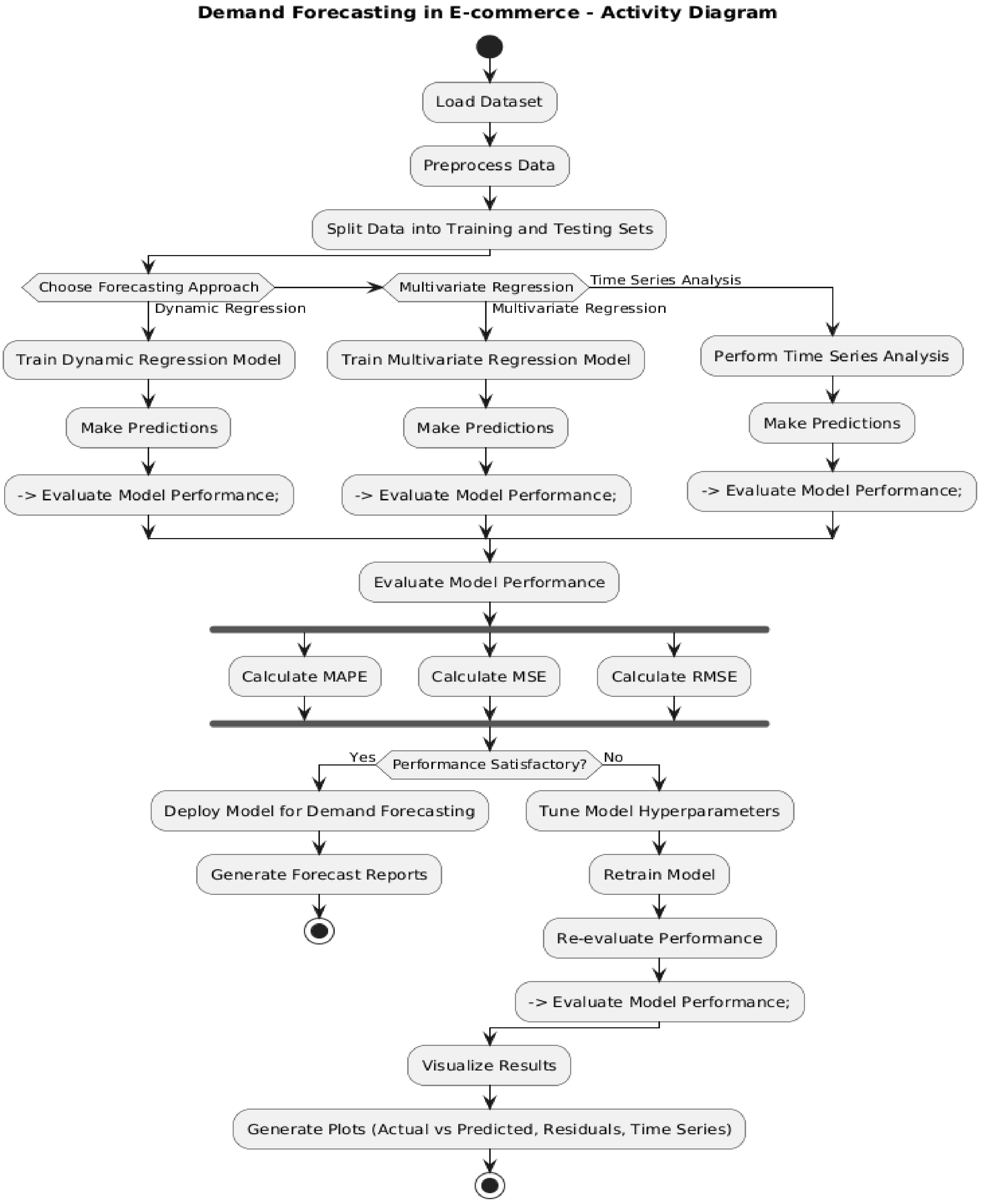
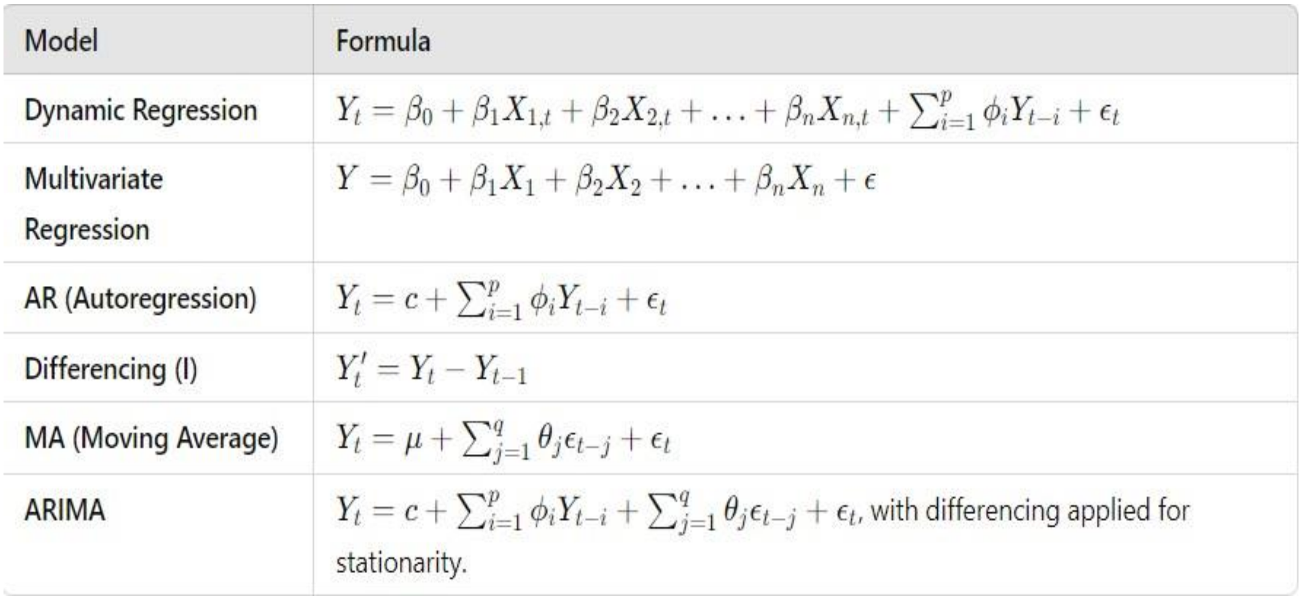

Demand Forecasting in E-commerce - Activity Diagram
Load Dataset
Preprocess Data
Split Data into Training and Testing Sets
Choose Forecasting Approach
Multivariate Regression
Time Series Analysis
Dynamic Regression
Multivariate Regression
Train Dynamic Regression Model
Train Multivariate Regression Model
Perform Time Series Analysis
Make Predictions
Make Predictions
Make Predictions
-> Evaluate Model Performance;
-> Evaluate Model Performance;
-> Evaluate Model Performance;
Evaluate Model Performance
Calculate MAPE
Calculate MSE
Calculate RMSE
Yes
Performance Satisfactory?
No
Deploy Model for Demand Forecasting
Tune Model Hyperparameters
Generate Forecast Reports
Retrain Model
Re-evaluate Performance
-> Evaluate Model Performance;
Visualize Results
Generate Plots (Actual vs Predicted, Residuals, Time Series)

Fig 5.3 : Activity Diagram For Demand Forecasting

SUMMARY OF FORMULAS USED:



Model	Formula
Dynamic Regression	$Y_t = \beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t} + \ldots + \beta_n X_{n,t} + \sum_{i=1}^{p} \phi_i Y_{t-i} + \epsilon_t$
Multivariate Regression	$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \epsilon$
AR (Autoregression)	$Y_t = c + \sum_{i=1}^{p} \phi_i Y_{t-i} + \epsilon_t$
Differencing (I)	$Y_t' = Y_t - Y_{t-1}$
MA (Moving Average)	$Y_t = \mu + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} + \epsilon_t$
ARIMA	$Y_t = c + \sum_{i=1}^{p} \phi_i Y_{t-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} + \epsilon_t$, with differencing applied for stationarity.

Table 5.4: Summary of Formulas used

These models and formulas facilitate demand forecasting to happen by capturing the associations between variables and time series. Every model has its functionality; Dynamic Regression uses predictors that change over time, Multivariate Regression analyzes the interdependence of the variables, and ARIMA controls for time series.

CHAPTER 6

SYSTEM IMPLEMENTATION

DYNAMIC REGRESSION:

```
# load the necessary libraries

library(tidyverse)

library(forecast)

library(evobiR)

library(tseries)

library(urca)

library(TSstudio)

productA_sales <- read.csv("ProductA.csv")       # Read sales quantity from csv file

productA_google_clicks <- read.csv("ProductA_google_clicks.csv") # Read clicks data from csv file

productA_fb_impressions <- read.csv("ProductA_fb_impressions.csv") # Read impressions data from csv file

split_point <- floor(0.8 * nrow(productA_sales))

# Define time series object for sales

productA_sales <- SlidingWindow("mean",productA_sales$Quantity,3,1)

productA_sales <- productA_sales[1:split_point]

productA_sales_ts <- ts(productA_sales, start = 1, frequency = 7)

# Cleaning variables using moving average and normalizing predictor variables

productA_google_clicks <- SlidingWindow("mean",productA_google_clicks$Clicks,3,1)
productA_google_clicks <- productA_google_clicks[1:split_point]

productA_google_clicks <- productA_google_clicks/(max(productA_google_clicks)-min(productA_google_clicks))

productA_fb_impressions <- SlidingWindow("mean",productA_fb_impressions$Impressions,3,1)

productA_fb_impressions <- productA_fb_impressions[1:split_point]

productA_fb_impressions <- productA_fb_impressions/(max(productA_fb_impressions)-min(productA_fb_impressions))

# Test for stationarity

kpss.test(productA_sales_ts)

# Dynamic Regression Modelling

fit_productA_sales <- auto.arima(productA_sales_ts, xreg = cbind(productA_google_clicks,productA_fb_impressions), seasonal = TRUE)

summary(fit_productA_sales)

checkresiduals(fit_productA_sales)
```

MULTIVARIATE REGRESSION:

```
# Load necessary libraries

library(tidyverse)

library(evobiR)

library(MLmetrics)

library(Metrics)

# Read sales and predictor data from CSV files
```

```
productA_sales <- read.csv("ProductA.csv")  # Sales quantity

productA_google_clicks <- read.csv("ProductA_google_clicks.csv")  # Clicks data

productA_fb_impressions <- read.csv("ProductA_fb_impressions.csv")
#Impressions data

# Clean and normalize predictor variables using moving average

productA_google_clicks <- SlidingWindow("mean",
productA_google_clicks$Clicks, 3, 1)

productA_google_clicks <- productA_google_clicks /
(max(productA_google_clicks) - min(productA_google_clicks))

productA_fb_impressions <- SlidingWindow("mean",
productA_fb_impressions$Impressions, 3, 1)

productA_fb_impressions <- productA_fb_impressions /
(max(productA_fb_impressions) - min(productA_fb_impressions))

productA_sales <- SlidingWindow("mean", productA_sales$Quantity, 3, 1)

# Combine the cleaned variables into a data frame

regression_productA_sales <- data.frame(productA_sales,
productA_google_clicks, productA_fb_impressions)

colnames(regression_productA_sales) <- c('sales', 'google_clicks', 'fb_impressions')

# Split data into training and testing sets (90% training, 10% testing)

split_point <- floor(0.9 * nrow(regression_productA_sales))

productA_training <- regression_productA_sales[1:split_point, ]

productA_testing <- regression_productA_sales[(split_point +
1):nrow(regression_productA_sales), ]



# Build the Multivariate Regression model

fit_productA_sales <- lm(sales ~ google_clicks + fb_impressions, data =
productA_training)
```

```
# Display the model summary
print(summary(fit_productA_sales))
# Generate predictions on the testing set
forecast <- predict(fit_productA_sales, productA_testing)

# Calculate performance metrics
mape_result <- Metrics::mape(productA_testing$sales, forecast)  # Mean Absolute Percentage Error
mse_result <- Metrics::mse(productA_testing$sales, forecast)       # Mean Squared Error
rmse_result <- Metrics::rmse(productA_testing$sales, forecast)     # Root Mean Squared Error
# Print performance metrics
cat("MAPE: ", mape_result, "\n")
cat("MSE: ", mse_result, "\n")
cat("RMSE: ", rmse_result, "\n")

# Plot Actual vs Predicted Values
actual_vs_predicted_plot <-ggplot(productA_testing, aes(x = sales, y = forecast)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, color = 'red') +  # Diagonal line for perfect predictions
  labs(title = "Actual vs Predicted Sales", x = "Actual Sales", y = "Predicted Sales") +
  theme_minimal()
print(actual_vs_predicted_plot)  # Explicitly print the plot
```

```
# Residuals Plot
residuals <- productA_testing$sales - forecast
residuals_plot <-ggplot(data.frame(residuals), aes(x = residuals)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Residuals Histogram", x = "Residuals", y = "Frequency") +
  theme_minimal()
print(residuals_plot)  # Explicitly print the plot
```

TIME-SERIES ANALYSIS:

```
#load the necessary libraries
library(tidyverse)
library(forecast)
library(evobiR)
library(tseries)
library(urca)
library(TSstudio)
library(Metrics)
productA <- read.csv("ProductA.csv") # Read sales quantity from csv file
# Splitting of time-series data into training and testing set
split_point <- floor(0.8 * nrow(productA))
productA_training <- productA[1:split_point,]
productA_testing <- productA[(split_point + 1):nrow(productA), ]
```

```
# Defining a time series object using the tarining data set

productA_ts <- ts(productA_training$Quantity, start = 1, frequency = 7)

autoplot(productA_ts) + ggtitle("Product A sales") + labs(x = 'time', y = 'Sales')

productA_clean <- SlidingWindow("mean",productA_training$Quantity,3,1) # Cleaning of data set using sliding window

productA_ts <- ts(productA_clean, start = 1, frequency = 7)

autoplot(productA_ts) + ggtitle("Product A sales") + labs(x = 'time', y = 'Sales')

ggAcf(productA_ts) + ggtitle("ACF of sales")

ggPacf(productA_ts) + ggtitle("PACF of sales")

d_productA_ts <- diff(productA_ts)

ggAcf(d_productA_ts) + ggtitle("ACF")

ggPacf(d_productA_ts) + ggtitle("PACF")

# Decomposition of time series object

ts_decompose(productA_ts, type = "additive", showline = TRUE)

# Tests for stationarity

adf.test(productA_ts)

adf.test(productA_ts, k=1)

adf.test(productA_ts, k=2)

pp.test(productA_ts)

pp.test(d_productA_ts)

kpss.test(productA_ts)

kpss.test(d_productA_ts)


# ARIMA modelling
```

```
fit_productA_ts <- auto.arima(productA_ts)
summary(fit_productA_ts)
forecast <- data.frame(forecast(fit_productA_ts, h = nrow(productA)-split_point))
forecast <- forecast$Point.Forecast
productA_testing <- productA_testing$Quantity
mape(productA_testing,forecast)
view(forecast)
time_index <- 1:(nrow(productA)-split_point)
df <- data.frame(time_index, productA_testing, forecast)
plot(df$time_index, df$productA_testing, type = "l", col = "blue",
     xlab = "Time Index",
     ylab = "Values")
lines(df$time_index, df$forecast, col = "red")
```

CHAPTER 7

PERFORMANCE ANALYSIS

7.1 INTRODUCTION

This section analyzes the performance of three models applied to forecast sales of Product A:Dynamic Regression – Autoregressive Integrated Moving Average with External Variables

Multiple Linear Regression

Time Series Analysis (ARIMA).

These models were assessed using several performance and diagnostic measures including above said criteria. For each model, a training set and a testing set were used for validating its accuracy and reliability of the model. The results are given in the form of accuracy measures including Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE), and diagnostic plots.

7.2. DYNAMIC REGRESSION WITH ARIMA

Model Overview:

In this study, the dynamic regression model utilized ARIMA(2,0,1)(1,0,1)[7] with exogenous predictors for considering auto-regression and moving-average ingredients along with the seasonality factor of 7 (weekly). The model was intended to include dynamic features of the sales data by seasonality and to include the exogenous variables such as marketing or advertising measures.

Residual Diagnostics:

The residual diagnostics plots indicate that the model captures most of the underlying sales patterns:

Residual Plot: This paper shows that the residuals are normally and equally distributed with mean zero, thus pointing to the fact that the model used has appropriately captured the whole picture.

ACF of Residuals: The residual autocorrelation function (acf of residuals) reveals that there is no serious redundancy present in the model by responding to the serial correlation issue within the data set, however, there is some evidence of correlation.

Histogram of Residuals: The residual distribution is approximately normal subject to some distortion suggesting that the residuals of the resulting model are near optimal, though there are larger prediction errors than the rest.

Ljung-Box Test:

So Ljung-Box test, we get (p-value = 0.00178), so there may be still some autocorrelation which is not being captured by ARIMA model. This could imply that the model may require other enhancements or that one or many other predictors could be added to the model.

Conclusion:

All in all, it can be said that the dynamic regression model rather well fits the sales data. However, there is evidence of remaining autocorrelation in the residuals, suggesting a possibility for further efficiency gain. This could be resolved by trying out other specifications of the ARIMA model or including more external variables to the model.

7.3 MULTIVARIATE REGRESSION

Model Overview:

A multivariate linear regression model was constructed to predict sales using two external predictors: Click through rate on Google and number of impressions on face book. These two predictors were standardized close to zero mean with a standard deviation of one and used in the regression model.

Performance Metrics:

The following performance metrics were calculated based on the model's predictions:

MAPE (Mean Absolute Percentage Error): 11.37%

MSE (Mean Squared Error): 3.61

RMSE (Root Mean Squared Error): 1.90

These findings suggest that on balance, the model yields prediction errors of close to 11.37% of the true sales magnitudes, which is reasonable for forecasting purposes.

Model Fit:

R-squared: 26.41% (Mutated R-squared:.2562)

Interpretation: The value of R-squared shows that 26.41 % or rather.2641 of the total variance in the sales is traced to the two predictors which are Google clicks and Facebook impressions. This means that the model is able to explain part of the coherent pattern between the predictors and sales while reducing the extent of the remaining pattern.

Residual Analysis:

Residual Plot: The residuals look somewhat normally distributed but with large outliers which suggest that the model does not do a good job working with high extreme values on the data.

Actual vs Predicted Sales Plot: From the figure showing the scatter plot it can be seen that predictions generally move in the same way as the actual sales but as values increase more so for larger ones, the prediction is off the mark.

Conclusion:

This is a simplistic and easily understandable model which gives a connection between the sales and external factors. Nevertheless, low coefficient of determination (R-squared), and residual analysis indicates that there may be other important independent variables, which the current model of linear regression does not capture to analyze the sale’s behavior adequately.

7.4 TIME SERIES ANALYSIS (ARIMA)

Model Overview:

Specifically, the ARIMA model was used for sales forecasting while only using sales data as the predictor. The above model was used to train the system using 80% data from the database while the remaining 20% was used to test the model that was trained. For the training data, trend and seasonality were obtained by decomposing the training data set and the model was refitted using automatic selection from the ARIMA model class.

Stationarity Tests:

To validate the assumptions of ARIMA, several tests were performed:

ADF Test (Augmented Dickey-Fuller): The result from both analysis showed that both the series were stationary.

KPSS Test (Kwiatkowski-Phillips-Schmidt-Shin): This test gave mixed signals, as it pointed to non-stationarity, and therefore the data must be preprocessed even further (for example by applying more aggressive levels of differencing).

Model Performance:

MAPE: 14.65%—The distribution of the numbers from the forecasts from actual sales is close to 14.65% which is acceptable for time series forecasting.

Visual Analysis:

ACF/PACF Plots: The ACF and PACF of the differenced data highlighted high autocorrelation, thus explanatory for applying ARIMA models to model such structure.

Decomposition Plot: The decomposition process demonstrated proper disaggregation of the series into the seasonal and trend factors with which the model appropriately identifies the sales pattern.

Forecast Plot:

A comparison of the forecast plot to the actual sales data reveals that while the ARIMA model mimics the large overall trend of the general sales, it fails to capture most of the other changes in sales revenue. Even the variations between the actual and forecasted values though small call for better management of such deviations.

Conclusion:

Indeed, the time series ARIMA model held fair accuracy for recognizing the trend and seasonality of the sales data. MAPE 14.65% meaning the model has a good prediction accuracy as it depictions. Nevertheless, non-stationary problems suggest that the model will require further pre-processing and improve parameter setting of the ARIMA model for better performance.

7.5COMPARATIVE EVALUATION OF MODELS:

Dynamic Regression (ARIMA with External Regressors): Records two kinds of experiences: sales experience and outside experience (including marketing experience), fast and flexible. However it detected slightly skewed residuals distribution, residual autocorrelation and minor model misspecification which require some improvement.

Multivariate Linear Regression: Gives a rough and easily interpreted model that accounts for a part of the variance but in so doing leaves out major aspects of sales behavior because it may lack some variables or it may not capture non-linear relations correctly.

Time Series ARIMA: This model does incorporate historic sales and seasonal variations satisfactorily. All in all the model can be considered as quite stable, however the points connected to stationarity and occasional discrepancy in forecasts do require some fine-tuning.

Final Thoughts:

Each model brings unique strengths and weaknesses to the task of forecasting Product A sales:

Exogenous variable in ARIMA model has an opportunity to identify extra-stochastic variables; however, it contains some residual auto-correlation.

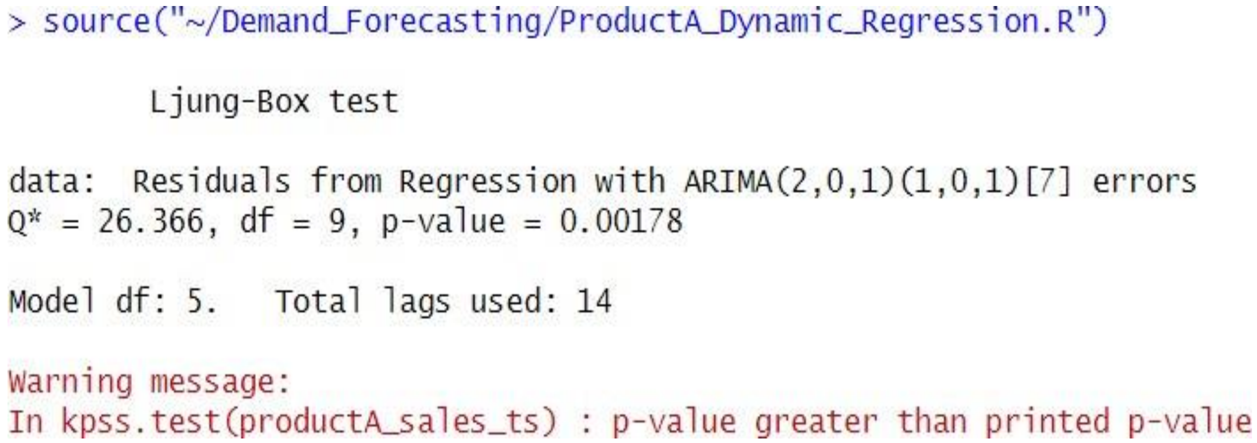
Multivariate Linear Regression is pretty straightforward but can be quite basic at times.

Time-Series ARIMA models are quite capable of capturing temporal patterns good enough but it lacks refined pre-processing of data and better feature extraction techniques.

Altogether, such approaches in combination with the further model’s improvement or with a combination of the improved ARIMA model and added techniques may offer highly accurate and definite forecasts for further sales.

7.6 OUTPUT

ProductA_Dynamic_Regression.R :



```
> source("~/Demand_Forecasting/ProductA_Dynamic_Regression.R")

        Ljung-Box test

data:  Residuals from Regression with ARIMA(2,0,1)(1,0,1)[7] errors
Q* = 26.366, df = 9, p-value = 0.00178

Model df: 5.   Total lags used: 14

Warning message:
In kpss.test(productA_sales_ts) : p-value greater than printed p-value
```

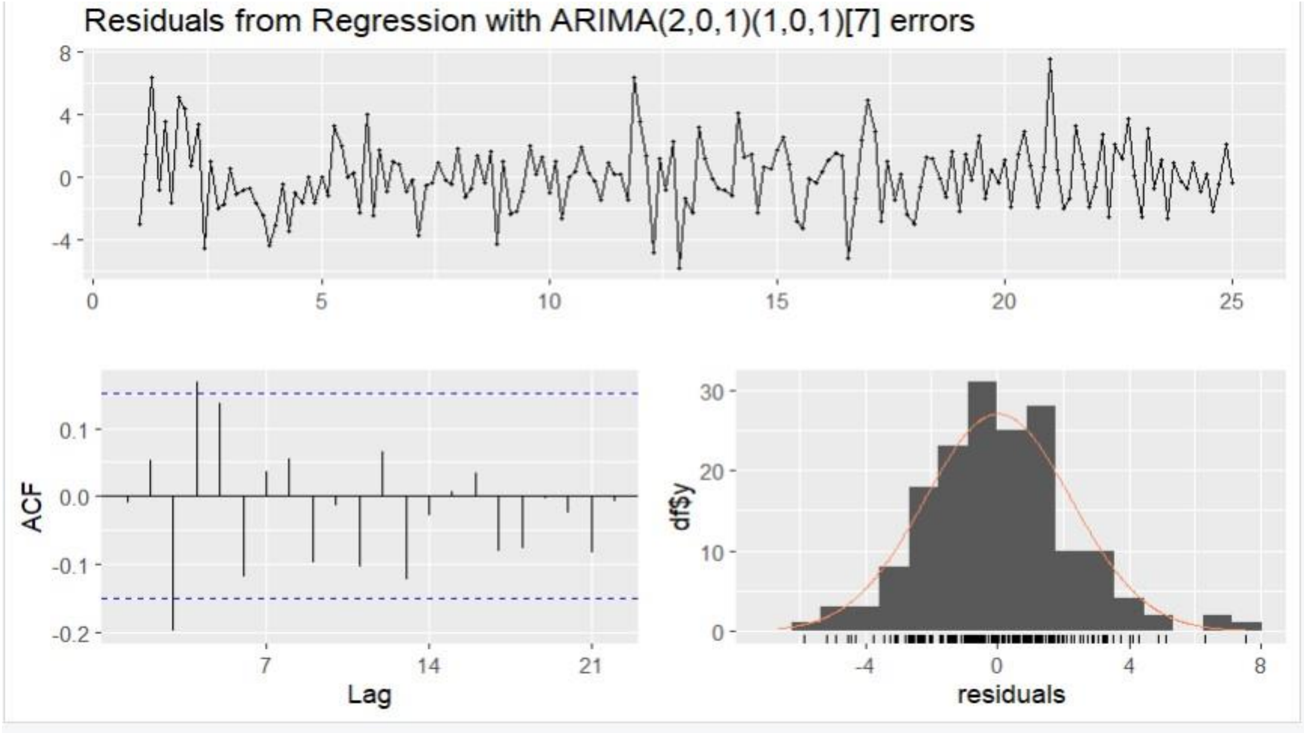
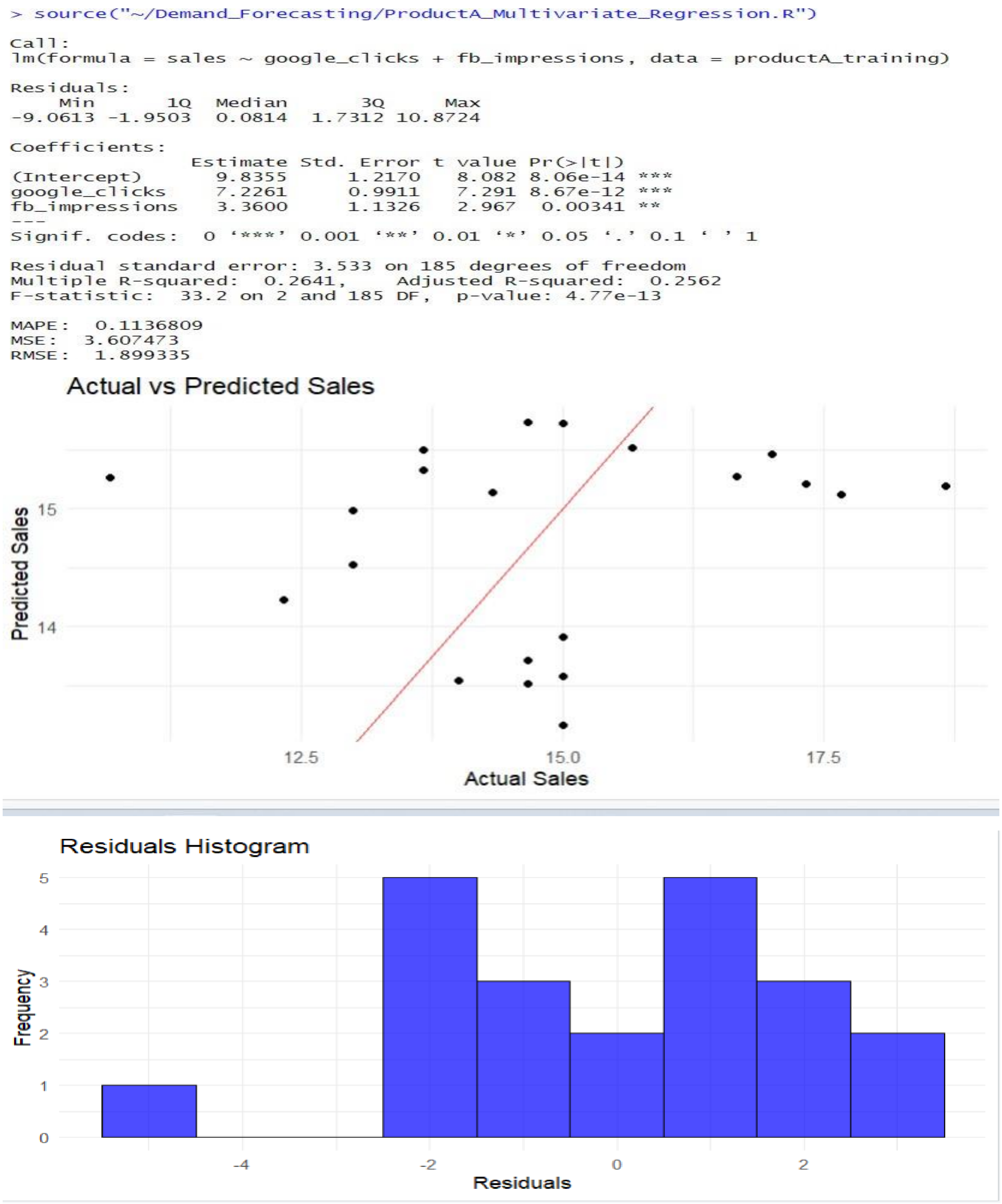


Fig 7.1 : Output of Dynamic_Regression model

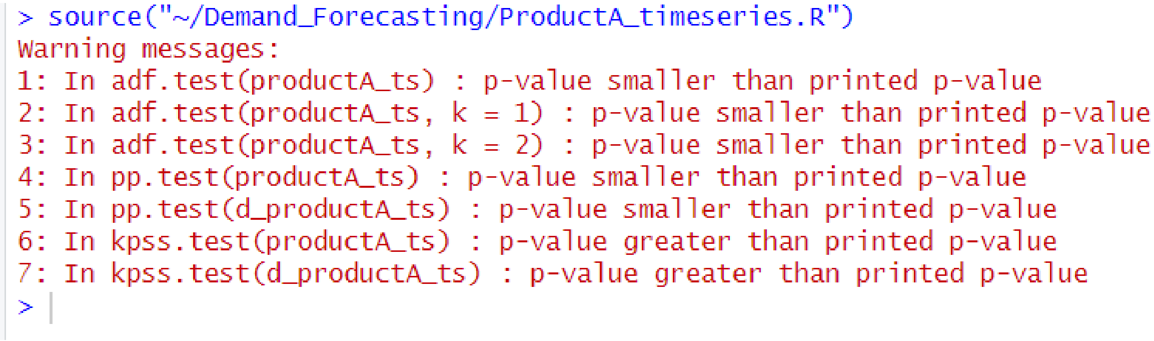
ProductA_Multivariate_Regression.R



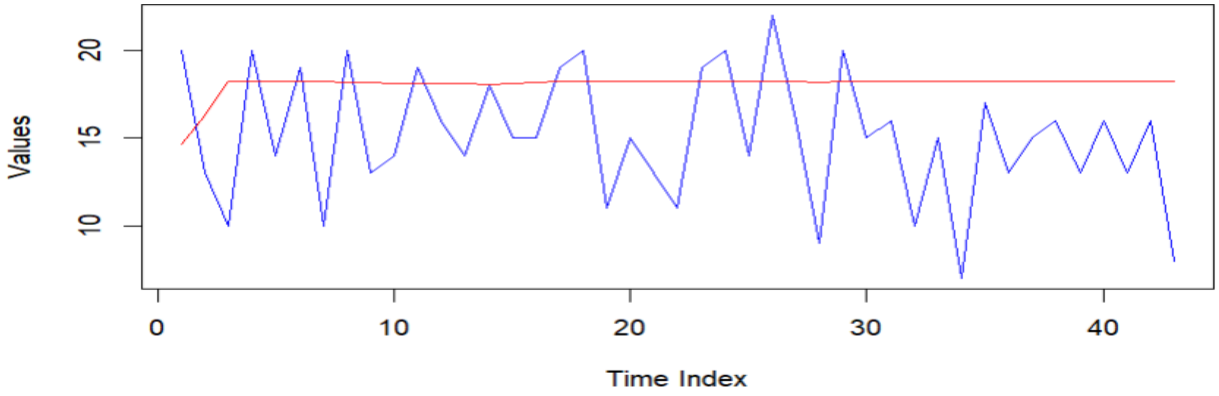
```
> source("~/Demand_Forecasting/ProductA_Multivariate_Regression.R")

Call:
lm(formula = sales ~ google_clicks + fb_impressions, data = productA_training)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.0613 -1.9503  0.0814  1.7312 10.8724 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)      9.8355     1.2170   8.082 8.06e-14 ***
google_clicks    7.2261     0.9911   7.291 8.67e-12 ***
fb_impressions   3.3600     1.1326   2.967  0.00341 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.533 on 185 degrees of freedom
Multiple R-squared:  0.2641,	Adjusted R-squared:  0.2562 
F-statistic:  33.2 on 2 and 185 DF,  p-value: 4.77e-13

MAPE:  0.1136809 
MSE:  3.607473 
RMSE:  1.899335 
```

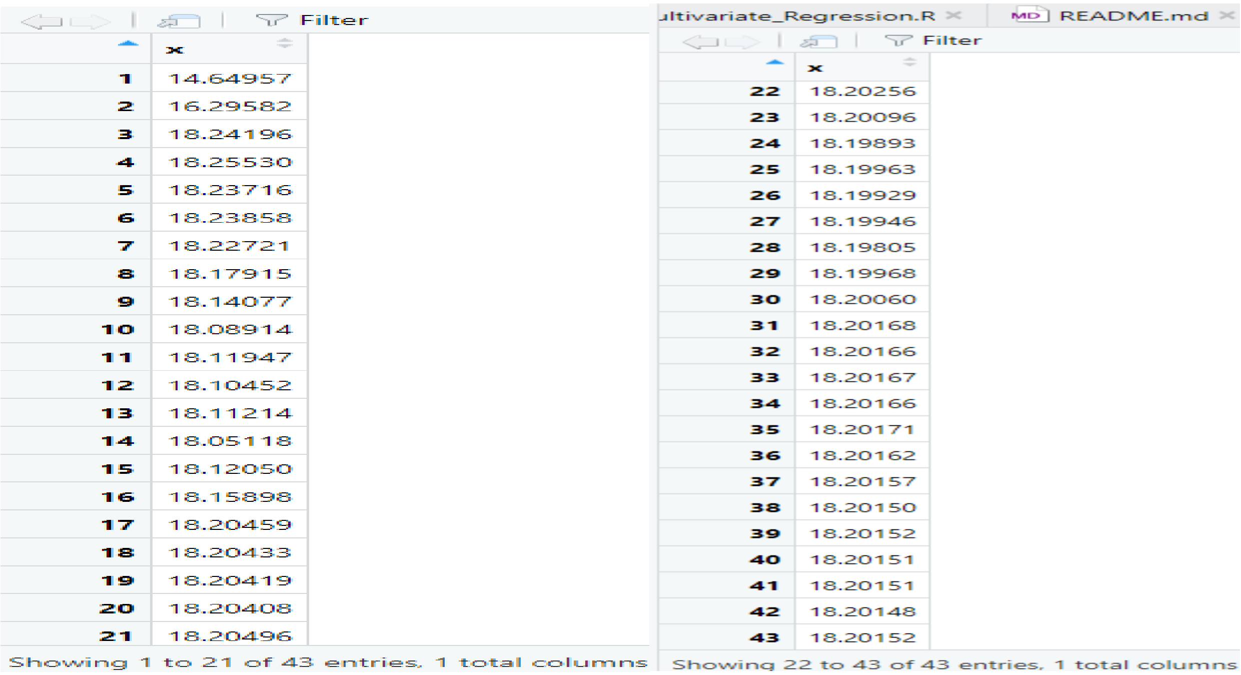
Fig 7.2 : Output for MultiVariate_Regression model

ProductA_timeseries.R :



```
> source("~/Demand_Forecasting/ProductA_timeseries.R")
Warning messages:
1: In adf.test(productA_ts) : p-value smaller than printed p-value
2: In adf.test(productA_ts, k = 1) : p-value smaller than printed p-value
3: In adf.test(productA_ts, k = 2) : p-value smaller than printed p-value
4: In pp.test(productA_ts) : p-value smaller than printed p-value
5: In pp.test(d_productA_ts) : p-value smaller than printed p-value
6: In kpss.test(productA_ts) : p-value greater than printed p-value
7: In kpss.test(d_productA_ts) : p-value greater than printed p-value
> 
```





	x
1	14.64957
2	16.29582
3	18.24196
4	18.25530
5	18.23716
6	18.23858
7	18.22721
8	18.17915
9	18.14077
10	18.08914
11	18.11947
12	18.10452
13	18.11214
14	18.05118
15	18.12050
16	18.15898
17	18.20459
18	18.20433
19	18.20419
20	18.20408
21	18.20496
22	18.20256
23	18.20096
24	18.19893
25	18.19963
26	18.19929
27	18.19946
28	18.19805
29	18.19968
30	18.20060
31	18.20168
32	18.20166
33	18.20167
34	18.20166
35	18.20171
36	18.20162
37	18.20157
38	18.20150
39	18.20152
40	18.20151
41	18.20151
42	18.20148
43	18.20152

Showing 1 to 21 of 43 entries, 1 total columns

Showing 22 to 43 of 43 entries, 1 total columns

Fig 7.3 : Output for TimeSeries_Analysis model

7.7 OBSERVATION OF RESULTS

Dynamic Regression Model: In combination with external variables, the autoregressive integrated moving-average of 2, 0, 1, with seasonality of 1, 0, 1 at the seventh observation level, known as Dynamic Regression model, provides the best estimate of the sales trends fitted by the model. But the result of Ljung-Box test is a little bit higher than the critical value and the p-value = 0.00178 which indicates there is a little bit residual autocorrelation and it means that the model is not perfect and cannot explain all the experience in the data series. As indicated by the residual histogram, the residuals are seemingly alike mean normal distribution but differ in that they feature heavier tails. The model in its current form can provide general brushes, although a refinement of the model might increase the total effectiveness.

Multivariate Regression Model: The analysis with help of the Multivariate Regression shows that Google clicks and Facebook impressions have the greatest impact on the sales, as indicated in the table by rather low p-values. Still, the Adjusted R-squared of 0.2562 means that the model only accounts for nearly 26% of what happened on the sales front, meaning that there remains a lot of ‘unexplained’ value out there. Even though the residuals appear to be randomly scattered and there are very few outliers one can conclude that the introduction of more variables or the using of more complex equation can improve the model.

Time Series Analysis:The Time Series Analysis avails that sales data is stationary grounded on ADF, PP and KPSS tests as corroborated by the results here above. This stationarity of the data makes it fit for time series forecasting models such as the ARIMA models. From the augmented time series plot, we realize that sale varies over time, with no general upward or downward trends or annual pattern, hence, it shows that while the model identifies short-term changes it fails to identify long term.

CHAPTER 8

CONCLUSION

This report compared three models of product sale prediction; Dynamic Regression, Multivariate Regression and Time Series Analysis. Dynamic regression model was able to capture the sales for external variables with residual autocorrelation which called for further modification. This study can understand Google clicks' and Facebook impression's impact on sales using the Multivariate Regression Model, although adjusted R^2 is only moderately high, so more variables or a more sophisticated approach may enhance prognoses. The Time Series Analysis affirmed data stationarity, which is appropriate for short term prediction but gave no clear trends for the long term. All in all, for each of the proposed models, the behaviour of sales can be described fairly well, however, several additional improvements can be made, for example, tuning of parameters, adding more variables or studying residual patterns.

Future Scope

Additional Predictors: Add seasonality, promotional activities, competitors' data, or macroeconomic trends if the company's sales data is not captured accurately.

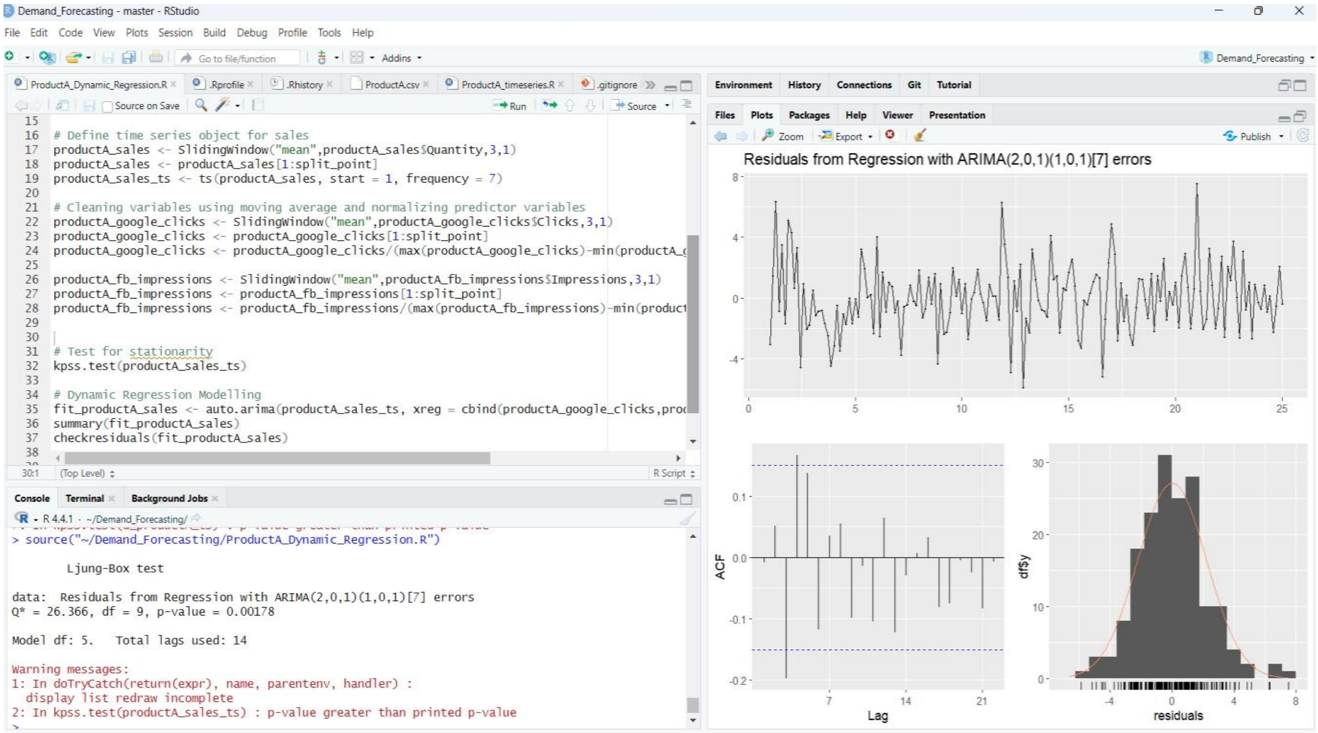
Advanced Machine Learning: In cases where there is a more complex relationship between features Random Forest or XGBoost and neural networks may be used, or high-quality mixed models.

Automated Hyperparameter Tuning: Use Grid Search or use some Bayesian Allocation for best parameters selection.

Real-time Forecasting: Include the streaming data about e-commerce platforms to make real-time predictions and to respond to the market instantly.

APPENDICES

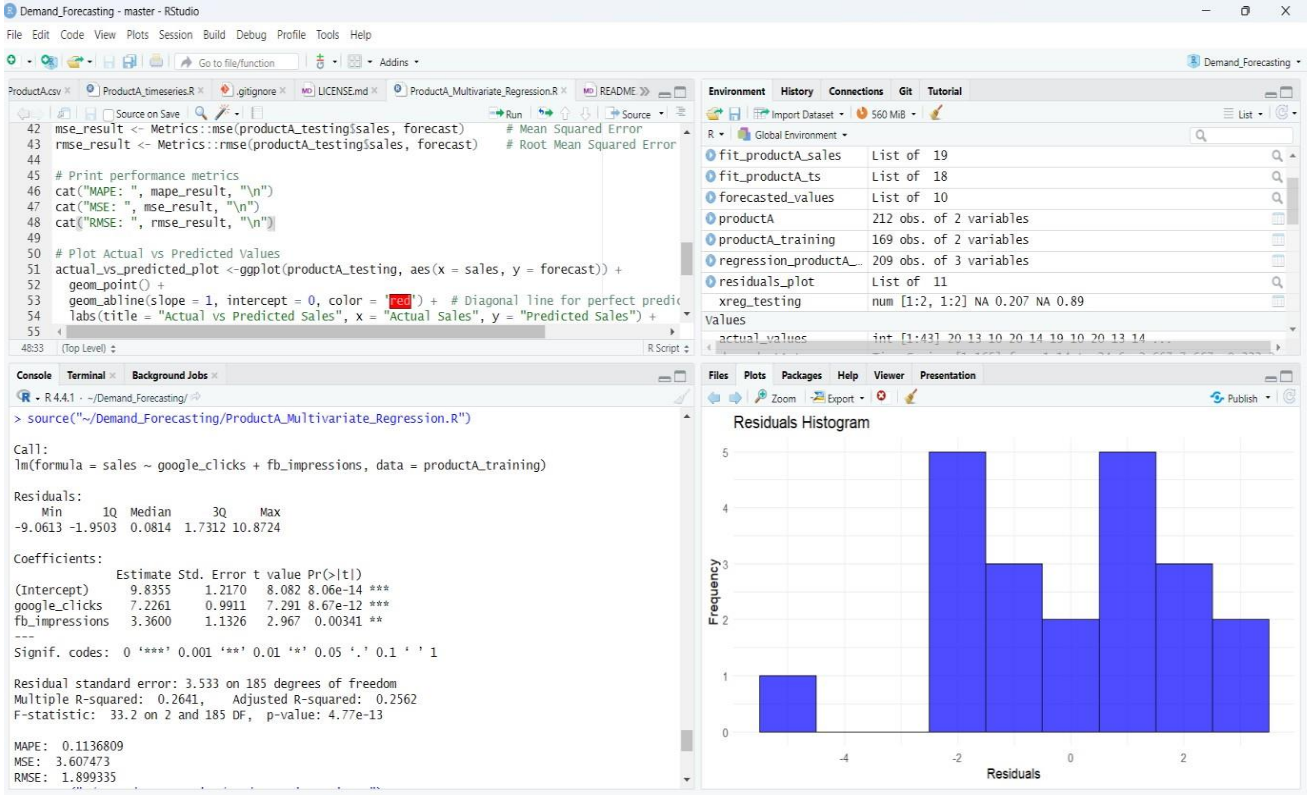
A.1 SAMPLE SCREENSHOTS



```
# Define time series object for sales
productA_sales <- SlidingWindow("mean",productA_sales$Quantity,3,1)
productA_sales <- productA_sales[1:split_point]
productA_sales_ts <- ts(productA_sales, start = 1, frequency = 7)

# Cleaning variables using moving average and normalizing predictor variables
productA_google_clicks <- SlidingWindow("mean",productA_google_clicks$Clicks,3,1)
productA_google_clicks <- productA_google_clicks[1:split_point]
productA_google_clicks <- productA_google_clicks/(max(productA_google_clicks)-min(productA_g

productA_fb_impressions <- SlidingWindow("mean",productA_fb_impressions$Impressions,3,1)
productA_fb_impressions <- productA_fb_impressions[1:split_point]
productA_fb_impressions <- productA_fb_impressions/(max(productA_fb_impressions)-min(product


# Test for stationarity
kpss.test(productA_sales_ts)

# Dynamic Regression Modelling
fit_productA_sales <- auto.arima(productA_sales_ts, xreg = cbind(productA_google_clicks,prod
summary(fit_productA_sales)
checkresiduals(fit_productA_sales)
```

```
> source("~/Demand_Forecasting/ProductA_Dynamic_Regression.R")

        Ljung-Box test

data:  Residuals from Regression with ARIMA(2,0,1)(1,0,1)[7] errors
Q* = 26.366, df = 9, p-value = 0.00178

Model df: 5.   Total lags used: 14

Warning messages:
1: In doTryCatch(return(expr), name, parentenv, handler) :
  display list redraw incomplete
2: In kpss.test(productA_sales_ts) : p-value greater than printed p-value
```

Fig 8.1 : Sample Screenshot 1

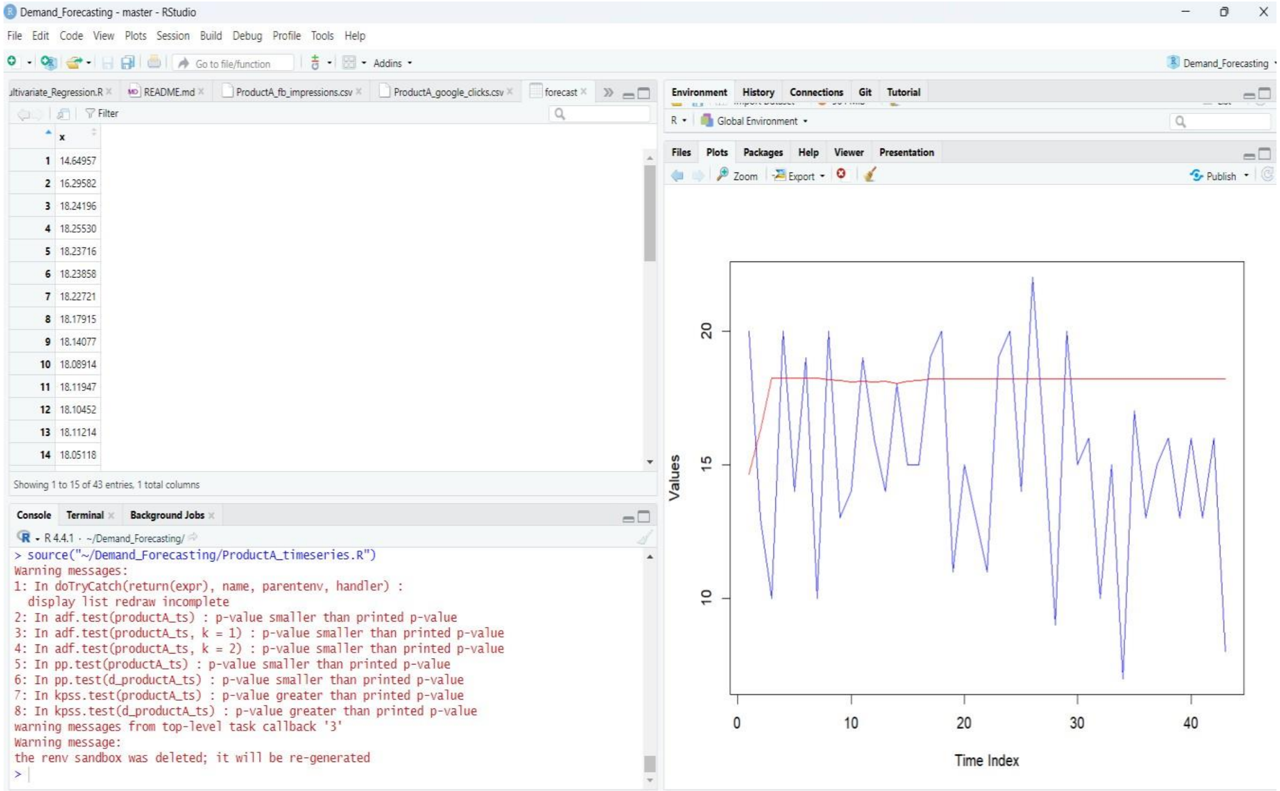


```
mse_result <- Metrics::mse(productA_testing$sales, forecast)      # Mean Squared Error
rmse_result <- Metrics::rmse(productA_testing$sales, forecast)    # Root Mean Squared Error

# Print performance metrics
cat("MAPE: ", mape_result, "\n")
cat("MSE: ", mse_result, "\n")
cat("RMSE: ", rmse_result, "\n")

# Plot Actual vs Predicted Values
actual_vs_predicted_plot <-ggplot(productA_testing, aes(x = sales, y = forecast)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, color = 'red') +  # Diagonal line for perfect predic
  labs(title = "Actual vs Predicted Sales", x = "Actual Sales", y = "Predicted Sales") +
```

```
fit_productA_sales      List of  19
fit_productA_ts         List of  18
forecasted_values       List of  10
productA                212 obs. of 2 variables
productA_training       169 obs. of 2 variables
regression_productA_…   209 obs. of 3 variables
residuals_plot          List of  11
xreg_testing            num [1:2, 1:2] NA 0.207 NA 0.89
Values
actual_values           int [1:43] 20 13 10 20 14 19 10 20 13 14 ...
```

```
> source("~/Demand_Forecasting/ProductA_Multivariate_Regression.R")

Call:
lm(formula = sales ~ google_clicks + fb_impressions, data = productA_training)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.0613 -1.9503  0.0814  1.7312 10.8724 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)      9.8355     1.2170   8.082 8.06e-14 ***
google_clicks    7.2261     0.9911   7.291 8.67e-12 ***
fb_impressions   3.3600     1.1326   2.967  0.00341 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.533 on 185 degrees of freedom
Multiple R-squared:  0.2641,	Adjusted R-squared:  0.2562 
F-statistic:  33.2 on 2 and 185 DF,  p-value: 4.77e-13

MAPE:  0.1136809 
MSE:  3.607473 
RMSE:  1.899335 
```

Fig 8.2 : Sample Screenshot 2



Demand_Forecasting - master - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Demand_Forecasting
ultivariate_Regression.R
README.md
ProductA_fb_impressions.csv
ProductA_google_clicks.csv
forecast
Filter
Environment History Connections Git Tutorial
R
Global Environment
Files Plots Packages Help Viewer Presentation
Zoom
Export
Publish
Console Terminal Background Jobs
R 4.4.1 · ~/Demand_Forecasting/
Values
20
15
10
0
10
20
30
40
Time Index

	x
1	14.64957
2	16.29582
3	18.24196
4	18.25530
5	18.23716
6	18.23858
7	18.22721
8	18.17915
9	18.14077
10	18.08914
11	18.11947
12	18.10452
13	18.11214
14	18.05118

Showing 1 to 15 of 43 entries, 1 total columns

```
> source("~/Demand_Forecasting/ProductA_timeseries.R")
Warning messages:
1: In doTryCatch(return(expr), name, parentenv, handler) :
  display list redraw incomplete
2: In adf.test(productA_ts) : p-value smaller than printed p-value
3: In adf.test(productA_ts, k = 1) : p-value smaller than printed p-value
4: In adf.test(productA_ts, k = 2) : p-value smaller than printed p-value
5: In pp.test(productA_ts) : p-value smaller than printed p-value
6: In pp.test(d_productA_ts) : p-value smaller than printed p-value
7: In kpss.test(productA_ts) : p-value greater than printed p-value
8: In kpss.test(d_productA_ts) : p-value greater than printed p-value
warning messages from top-level task callback '3'
Warning message:
the renv sandbox was deleted; it will be re-generated
>
```

Fig 8.3: Sample Screenshot 3

REFERENCES

[1] Zhao, B.; Liu, X.; Chen, W. When Crowdsensing Meets Federated Learning:Privacy-Preserving Mobile Crowdsensing System. arXiv 2021, arXiv:2102.10109.

[2] Mohassel, P.; Zhang, Y. SecureML: A System for Scalable Privacy-Preserving Machine Learning. IEEE Symp. Secur. Privacy 2017, 19–38.

[3] Gao, D.; Liu, Y.; Huang, A.; Ju, C.; Yu, H.; Yang, Q. Privacy-preserving Heterogeneous Federated Transfer Learning. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 2552–2559.

[4] Zhao, Y.; Zhao, J.; Jiang, L.; Tan, R.; Niyato, D.; Li, Z.; Liu, Y. Privacy-Preserving Blockchain-Based Federated Learning for IoT Devices. IEEE Internet Things J. 2021, 8, 1817–1829.

[5] Cong, M.; Yu, H.; Weng, X.; Yiu, S.M. A Game-Theoretic Framework for Incentive Mechanism Design in Federated Learning. In Federated Learning; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12500, pp. 205–222.

[6] Wei, S.; Tong, Y.; Zhou, Z.; Song, T. Efficient and Fair Data Valuation for Horizontal Federated Learning. In Federated Learning; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12500, pp. 139–152.

[7] Kim, S. Incentive Design and Differential Privacy Based Federated Learning: A Mechanism Design Perspective. IEEE Access 2020, 8, 187317–187325.

[8] Zhan, Y.; Zhang, J.; Hong, Z.; Wu, L.; Guo, S. A Survey of Incentive Mechanism Design for Federated Learning. IEEE Trans. Emerg. Top. Comput. 2021, 99, 1.

[9] Alberternst, S.; Anisimov, A.; Andre, A.; Benjamin, D.; Hilko, H.; Michael, M.; Muhammad, M.; Daniel, S.; Ingo, Z. Orchestrating Heterogeneous Devices and AI Services as Virtual Sensors for Secure Cloud-Based IoT Applications. Sensors 2021, 21, 7509

[10] Huang, W.; Yang, Y.; Chen, M.; Liu, C.; Feng, C.; Vincent, H.P. Wireless Network Optimization for Federated Learning with Model Compression in Hybrid VLC/RF Systems. Entropy 2021, 23, 1413.

[11] Vasiliki, K.; Vasileios, A.; Thomas, L.; George, F.; Elisavet, G.; Panagiotis, S. IDS for Industrial Applications: A Federated Learning Approach with Active Personalization. Sensors 2021, 21, 6743.

[12] Venkataramanan, K.; Kaza, S.; Annaswamy, A.M. DER Forecast using Privacy Preserving Federated Learning. arXiv 2021, arXiv:2107.03248.

[13] Susan Li (Jul 9, 2018) An End-to-End Project on Time Series Analysis and Forecasting with Python. https://towardsdatascience.com/an-end-to-end-project-on-time-seriesanalysis-and-forecasting-with-python-4835e6bf050b

[14] Mupparaju, Kalyan, Anurag Soni, Prasad Gujela, and Matthew A. Lanham. "A Comparative Study of Machine Learning Frameworks for Demand Forecasting." (2018).

[15]Demand Forecasting of E-Commerce Enterprises Based on Horizontal Federated Learning from the Perspective of Sustainable Development Juntao Li 1 , Tianxu Cui 1,* , Kaiwen Yang 1 , Ruiping Yuan 1 , Liyan He 1 and Mengtao Li 2