E-Commerce Sales Prediction

A

Project Report

Submitted in partial fulfillment of the requirement for the award of degree of

Bachelor of Technology

In

Computer Science & Engineering (Data Science)

Submitted to

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA,
BHOPAL (M.P.)

Guided By
Prof. MAHENDRA VERMA

Submitted By
HARIOM MANDLOI (0827CD221033)
SHUBHAM SILORIYA (0827CD221068)
DEEPAK PATEL (0827CD221023)
DEEPAK PATEL (0827CD221024)

DEPARTMENT OF CSE(DATA-SCIENCE)
ACROPOLIS INSTITUTE OF TECHNOLOGY & RESEARCH,
INDORE (M.P.) 452020
2025-2026

Declaration

I hereby declared that the work, which is being presented in the project entitled

E-Commerce Sales Prediction partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology**, submitted in the department of Computer Science & Engineering (Data Science) at **Acropolis Institute of Technology & Research, Indore** is an authentic record of my own work carried under the supervision of "**Prof. Mahendra Verma**". I have not submitted the matter embodied in this report for the award of any other degree.

Hariom Mandloi (0827CD221033)

Prof. Mahendra Verma

Supervisor

Project Approval Form

I hereby recommend that the project **E-Commerce Sales Prediction** prepared under my supervision by **Hariom Mandloi (0827CD220133)** be accepted in partial fulfillment of the requirement for the degree of Bachelor of Technology in Computer Science & Engineering (Data Science)

Prof. Mahendra Verma

Supervisor

Recommendation concurred in 2025-2026

Prof. Deepak Singh Chouhan
Project Incharge

Prof. Deepak Singh Chouhan

Project Coordinator

Acropolis Institute of Technology & Research

Computer Science & Engineering (Data Science)

Certificate

The project work entitled **E-Commerce Sales Prediction** submitted by **Hariom Mandloi (0827CD221033)** is approved as partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science & Engineering (Data Science) by Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.).

Internal Examiner

Name:…….

Date: …./…/……..

External Examiner

Name: ……..

Date: …./…/……..

Acknowledgement

With boundless love and appreciation, we/I would like to extend our/my heartfelt gratitude and appreciation to the people who helped us/me to bring this work to reality. We/I would like to have some space of acknowledgement for them.

Foremost, our/I would like to express our/ my sincere gratitude to our/my supervisor, **Prof. Mahendra Verma** whose expertise, consistent guidance, ample time spent and consistent advice that helped us/me to bring this study into success.

To the project in-charge **Prof. Deepak Singh Chouhan** and project coordinator **Prof. Deepak Singh Chouhan** for their constructive comments, suggestions, and critiquing even in hardship.

To the honorable **Prof. (Dr.) Prashant Lakkadwala**, Head, Department of Computer Science & Engineering (Data Science) for his favorable responses regarding the study and providing necessary facilities.

To the honorable **Dr. S.C. Sharma**, Director, AITR, Indore for his unending support, advice and effort to make it possible.

Finally, I/we would like to pay my/our thanks to faculty members and staff of the Department of Computer Science & Engineering for their timely help and support.

We/I also like to pay thanks to our/my **parents** for their eternal love, support and prayers without them it is not possible.

Hariom Mandloi (0827CD221033)

Shubham Siloriya (0827CD221068)

Deepak Patel (0827CD221023)

Deepak Patel (0827CD221024)

Abstract

The project titled *E-Commerce Sales Analysis and Machine Learning Prediction System* aims to transform raw online-sales data into meaningful insights and predictive intelligence. In today's digital marketplace, large-scale e-commerce platforms record massive amounts of data on customers, products, payments, and delivery operations. However, this information often remains under-utilized. The objective of this project is to build a unified analytical system that performs data preprocessing, visualization, and machine learning-based prediction of future sales and revenue trends.

The system reads historical e-commerce datasets (such as Kaggle dataset or a synthetic equivalent) and cleans, structures, and stores them for analysis. Exploratory Data Analysis (EDA) is performed using **Python**, **Pandas**, **NumPy**, and **Matplotlib** to uncover patterns in monthly revenue, popular categories, top-performing cities, and payment methods. For predictive modeling, algorithms such as **Linear Regression**, **Random Forest**, are applied using **Scikit-Learn** to forecast sales and detect demand fluctuations.

A front-end dashboard built in **Streamlit** provides an interactive interface where users can upload data, visualize trends, train models, and view predictions without requiring backend deployment. The results include charts, accuracy metrics (RMSE, R^2), and downloadable reports that support decision-making for pricing, inventory, and marketing strategies.

The findings indicate that machine-learning-based forecasting can accurately capture seasonal behavior and purchasing patterns, helping businesses plan stock levels and marketing campaigns more efficiently. The system demonstrates that data-driven analytics can significantly enhance business intelligence in e-commerce environments by replacing intuition-based decisions with predictive insights.

Table of Content

List of Figures

List of Tables

Abbreviations

- **EDA**: Exploratory Data Analysis
- **ML**: Machine Learning
- **AI**: Artificial Intelligence
- **CSV**: Comma Separated Values
- **UI**: User Interface
- **UX**: User Experience
- **API**: Application Programming Interface
- **IDE**: Integrated Development Environment
- **Pandas**: Python Data Analysis Library
- **NumPy**: Numerical Python
- **Sklearn**: Scikit-Learn (Machine Learning Library)
- **DL**: Deep Learning
- **LR**: Linear Regression
- **RF**: Random Forest
- **RMSE**: Root Mean Squared Error
- **MAE**: Mean Absolute Error
- **R^2**: Coefficient of Determination
- **ETL**: Extract, Transform, Load
- **EDA**: Exploratory Data Analysis
- **CPU**: Central Processing Unit
- **RAM**: Random Access Memory

- **JSON**: JavaScript Object Notation
- **HTTP**: HyperText Transfer Protocol
- **HTTPS**: HyperText Transfer Protocol Secure
- **DBMS**: Database Management System
- **ERD**: Entity-Relationship Diagram
- **KPI**: Key Performance Indicator
- **SDK**: Software Development Kit
- **UI/UX**: User Interface / User Experience
- **CLI**: Command Line Interface
- **VS Code**: Visual Studio Code
- **GPU**: Graphics Processing Unit
- **PNG**: Portable Network Graphics
- **SQL**: Structured Query Language

CHAPTER 1: INTRODUCTION

1.1 Rationale

E-commerce has become an integral part of global trade, with enormous transaction volumes generated daily. Yet most small- and medium-sized enterprises lack the tools to analyze and forecast their data effectively. This project provides a compact, cost-effective analytical solution to help such businesses gain insights and predict sales trends using data science techniques.

1.2 Existing System

In most e-commerce environments today, data is collected continuously through customer orders, browsing patterns, product purchases, delivery details, and payment transactions. However, the existing systems implemented by many small and mid-sized e-commerce businesses focus primarily on operational functions such as listing products, processing payments, and delivering orders. These systems generate data but do not fully utilize it for analytical or predictive purposes. As a result, businesses rely heavily on static reports, basic Excel sheets, or manually prepared summaries that lack depth and fail to identify hidden trends in sales performance or customer behavior.

Existing analytical tools such as Google Analytics or basic dashboards in e-commerce platforms like Shopify and WooCommerce provide only surface-level descriptive reports. They show simple metrics such as total sales, number of customers, traffic sources, and conversions, but they do not offer advanced insights such as revenue forecasting, product demand prediction, seasonality analysis, or customer segmentation. These systems also lack machine learning capabilities that could help predict future sales patterns, anticipate stock shortages, or optimize pricing strategies.

1.3 Problem Formulation

The rapid growth of e-commerce platforms has resulted in a massive increase in customer transactions, product listings, delivery data, and payment logs. Although this data has enormous potential for improving business strategies, most companies struggle to use it effectively. The problem lies in the absence of an integrated system that can automatically process large datasets, detect meaningful trends, and generate accurate predictions. Businesses currently make decisions based on limited information, intuition, or outdated static reports, which leads to inefficiencies in stock management, poor demand forecasting, inaccurate sale predictions, and missed business opportunities.

Moreover, raw e-commerce data is often unstructured, noisy, and inconsistent, making manual analysis extremely difficult and time-consuming. Without proper visualization tools, identifying patterns such as seasonal demand, city-wise revenue variations, high-return products, delivery performance, and customer purchasing behaviour becomes complex. In the absence of predictive modeling, businesses struggle to plan inventory levels, allocate resources, and forecast revenue accurately.

1.4 Proposed System

To overcome the shortcomings of existing systems, the project proposes an **E-Commerce Sales Analysis and Machine Learning Prediction System**, built using Python, data analytics libraries, and a Streamlit-based interactive interface. The system begins by accepting raw e-commerce datasets from various sources such as Kaggle (Dataset) or synthetic data. It performs automated data preprocessing, including cleaning, formatting, removing inconsistencies, and generating new derived features such as monthly revenue, product demand, customer frequency, and delivery time.

The analytical module uses Exploratory Data Analysis (EDA) techniques to visualize key business metrics such as annual and monthly sales trends, top-performing product categories, regional demand variations, customer behavior patterns, and factors

influencing returns or delays. These visual insights are rendered using Matplotlib and Plotly to ensure clarity and user-friendliness.

The machine learning module then trains predictive models like Linear Regression, Random Forest, or XGBoost to forecast future sales, order amounts, customer demand, or return probability. The system automatically evaluates model performance and presents accuracy metrics such as RMSE and R^2. A Streamlit dashboard integrates all functionalities—data upload, visualization, model training, prediction input, and exporting results—making the system accessible without any backend server requirement.

Thus, the proposed system acts as a comprehensive analytical and predictive platform that empowers businesses to make data-driven decisions, plan inventory, optimize marketing strategies, and improve customer satisfaction.

1.5 Objectives

The primary objectives of the **E-Commerce Sales Prediction** are:

- **To develop a complete e-commerce data analytics system** capable of processing, cleaning, and visualizing large datasets to uncover meaningful patterns and trends.
- **To implement machine learning models** that can accurately predict sales, revenue, product demand, or delivery performance based on historical data.
- **To design a user-friendly interface using Streamlit**, enabling users to upload data, perform analysis, train models, and view predictions without any programming or backend requirements.
- **To support business decision-making** by providing actionable insights such as top-selling products, seasonal trends, customer behavior patterns, and revenue variations.
- **To improve operational efficiency** in inventory planning, marketing strategies, and resource allocation through predictive analytics.

- **To create a scalable and extendable system** that can be enhanced with more datasets, advanced models, or additional automation features in the future.

These objectives collectively aim to transform **E-Commerce Sales Prediction** into a streamlined, efficient, and transparent process.

1.6 Contribution of the Project

1.6.1 Market Potential

The demand for data-driven decision-making in the e-commerce industry is rapidly increasing. Companies rely on predictive analytics to manage inventory, optimize delivery, plan sales campaigns, and improve customer satisfaction. This project addresses a significant market requirement by providing an affordable, scalable, and easy-to-use analysis and prediction tool suitable for small to medium businesses as well as academic learning environments.

1.6.2 Innovativeness

The innovation lies in integrating **EDA, visualization, and machine learning** within a single Streamlit-based platform that requires no backend deployment. Unlike traditional BI tools, this system not only visualizes data but also predicts future outcomes. The project also demonstrates dynamic model retraining, automated data cleaning, and advanced analytics, making it more flexible than existing dashboards.

1.6.3 Usefulness

This project offers practical utility for business owners, analysts, and researchers. It provides insights into customer behavior, product demand, and revenue performance. The prediction module helps businesses reduce losses by forecasting sales and managing inventory effectively. The system can also be used as an educational tool to teach data analysis, visualization, and machine learning concepts.

1.7 Report Organization

The report is structured to provide a comprehensive understanding of the **Quiz System** project. The chapters are organized as follows:

Chapter 1: Introduction, including rationale, objectives, and contribution of the project.

Chapter 2: Requirement Engineering, detailing feasibility study, functional and non-functional requirements, and system requirements.

Chapter 3: Analysis & Design, covering technical architecture, use-case, class, and ER diagrams.

Chapter 4: Implementation & Testing, explaining the development approach and testing methods.

Chapter 5: Results & Discussion, showcasing user interface representations, database design, and performance analysis.

Chapter 6: Conclusion & Future Scope, summarizing the project and suggesting future improvements.

CHAPTER 2: REQUIREMENT ENGINEERING

2.1 Feasibility Study

A feasibility study was conducted to determine whether the E-commerce Sales Analysis and Machine Learning Prediction System can be successfully implemented using the chosen technologies and resources. The study evaluated the system from three perspectives: technical, economic, and operational feasibility.

2.1.1 Technical Feasibility

The system uses Python, which is widely adopted for data science, machine learning, and analytics. Libraries such as Pandas, NumPy, Scikit-learn, Matplotlib, and Streamlit are open-source and well-documented, making implementation technically feasible. Streamlit enables the creation of interactive dashboards without requiring backend development, reducing complexity. The project only requires moderate computational power (8 GB RAM or higher), making it viable on standard consumer-grade laptops. Hence, technically, the project is highly feasible.

2.1.2 Economic Feasibility

All tools and libraries used—Python, Streamlit, Scikit-learn, and Kaggle datasets—are free for academic and personal use. The system does not need paid cloud services or licensed software. This makes the project extremely cost-effective, especially for students and small businesses. The economic feasibility is therefore strong as implementation costs are minimal and limited to hardware already required for coursework.

2.1.3 Operational Feasibility

The system is designed for non-technical users through a Streamlit-based user interface. Users can upload datasets, view analysis, and generate predictions with simple interactions. No backend server setup or database deployment is required. Because it integrates automatically with Python functions, the system is easy to operate and maintain. Thus, operational feasibility is high due to its user-centric and lightweight design.

2.2 Requirement Collection

Requirement collection is the process of gathering the necessary data, information, and constraints to ensure that the **E-Commerce Sales Prediction** will meet the users needs and align with the overall business goals of educational institutions.

2.2.1 Discussion

Requirement collection involved identifying user needs, system expectations, and functional goals for this analytical and predictive platform. Requirements were gathered through studying current analytical practices, analyzing existing reporting tools, and understanding the workflow of e-commerce data pipelines.

2.2.2 Requirement Analysis

Based on discussions, the system must:

- Accept e-commerce sales data in CSV format.
- Automatically clean and preprocess the dataset.
- Perform EDA to identify trends in revenue, product categories, and customer behaviour.
- Train machine learning models to predict sales or order amounts.
- Provide accuracy metrics such as RMSE and R^2.
- Allow users to visualize trends and predictions interactively.
- Enable exporting of results.

2.3 Requirements

Requirements specify what the E-commerce sales prediction should do and the conditions it must operate under. They are categorized into functional and non-functional requirements.

2.3.1 Functional Requirements

The functional requirements specify the core functionalities of the E-commerce sales prediction.

2.3.1.1 Statement of Functionality

- The system must allow have the datasets in CSV format.
- The system must clean, preprocess, and structure the input data automatically.
- The system must generate visual analytics (bar charts, line graphs, histograms).
- The system must support training of ML models such as Linear Regression and Random Forest.
- The system must provide prediction output for new entries or selected features.
- The system must display model evaluation results.
- The system must allow exporting of predictions or analytical reports.

2.3.2 Non-Functional Requirements

Non-functional requirements describe the system's overall characteristics and quality attributes, including usability, performance, and security. For the E-commerce sales prediction non-functional requirements might include:

2.3.2.1 Statement of Functionality

- **Usability:** The interface must be simple and easy for non-technical users.
- **Performance:** The system should load datasets and generate visualizations within reasonable time (1–3 seconds depending on data size).
- **Scalability:** Should handle moderate-sized datasets (up to 100k rows).
- **Accuracy:** Predictions should achieve acceptable RMSE and R^2 scores based on trained models.
- **Portability:** The system must run on Windows, Linux, and macOS using Python.
- **Reliability:** The system must not crash during dataset upload or model training.
- **Maintainability:** Code should be modular and easy to extend with new models or features.

These requirements collectively ensure that the **E-commerce sales prediction** is designed to meet the needs of its users effectively while maintaining high standards of performance, security, and usability.

2.4 Hardware & Software Requirements

2.4.1 Hardware Requirements (Developer & End User)

Hardware requirements describe the physical devices required for the development and usage of the Quiz System .

Table.2.1 Hardware Requirements

Category	Requirements
Developer Hardware	- High-performance PC/laptop (8GB RAM, Intel i5/Ryzen 5, 256GB SSD)
	- Cloud/on-premise server (Quad-Core CPU, 16GB RAM, 100GB SSD)
End-User Hardware	- Devices: Desktop/laptop/mobile (4GB RAM, modern web browser)
	- Internet: Minimum 5 Mbps (10 Mbps recommended for optimal performance)

Developer Hardware Requirements:

- **Processor:** Intel i5 / Ryzen 5 or higher
- **RAM:** 8 GB minimum
- **Storage:** 20 GB free
- **GPU:** Optional (only for advanced ML)

End User Hardware Requirements:

- **Processor:** Any modern dual-core processor
- **RAM:** 4 GB or more
- **Web Browser:** (Chrome/Edge) to access Streamlit UI

2.4.2 Software Requirements (Developer & End User)

Software requirements specify the software tools and environments needed to develop and use the system.

Table. 2.2 Software Requirements

Category	Software Requirements
Developer Software	– IDE / Code Editor: Visual Studio Code, PyCharm – Programming Language: Python 3.10+ – Libraries / Frameworks: Pandas, NumPy, Mattlotlib, Seaborn, Scikit-Learn, Streamlit, Joblib – Data Source / Tools: Kaggle Datasets – Version Control: GitHub / GitLab Testing Tools: Jupyter Notebook Testing, Streamlit UI Testing, Python Unit Tests
End-User Software	– Browsers: Chrome, Firefox, Microsoft Edge, Safari – Operating System: Windows, maOS, Linux – Required Runtime: Python installed / Streamlit web interface (auto-runs in browser)

Developer Software Requirements:

- **Operating System:** Windows 10/11, Ubuntu, or macOS
- **Language:** Python 3.10+
- **Libraries:** Pandas, NumPy, Matplotlib, Scikit-learn, Streamlit
- **Dataset Source:** Kaggle Dataset / Synthetic Dataset
- **IDE:** VS Code / PyCharm / Jupyter Notebook

End User Software Requirements:

- **Operating System:** The system is accessible on any OS with a modern web browser, including Windows, Linux, MacOS, Android, and iOS.

- **Web Browser:** A modern web browser such as Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge is recommended.

- **Internet Connection:** A stable internet connection is required to access the

  E-commerce sales prediction

2.5 Use-case Diagrams

The diagram illustrates how users interact with the system:

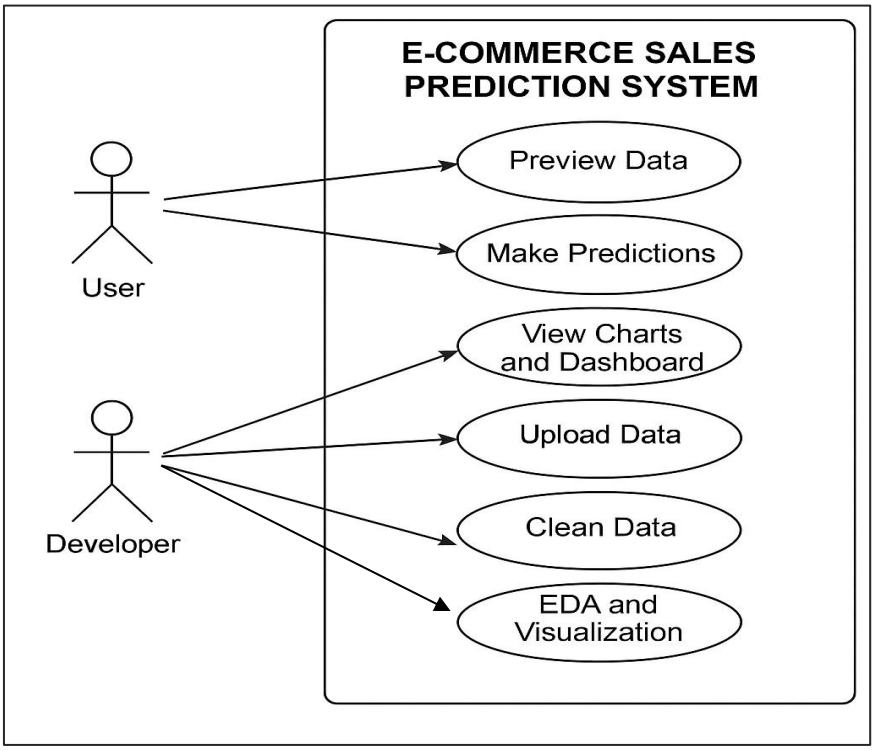

Fig.2.1 USE CASE DIAGRAM

Actors:

- **User:** Can preview dataset, make predictions, see patterns and visualization.

- **Developer:** Uploads data, clean data, made visualization, train model.

Use-Case Diagram Explanation:

The use-case diagram shows how the User and Developer interact with the E-Commerce Sales Prediction System. The User can preview the dataset, view dashboards, make predictions, and download reports. The Developer uploads and cleans data, performs EDA, creates visualizations, and trains the machine-learning model. The diagram highlights how developer-prepared data and models enable users to analyse sales patterns and generate accurate predictions.

CHAPTER 3: ANALYSIS & CONCEPTUAL DESIGN & TECHNICAL ARCHITECTURE

3.1 Technical Architecture

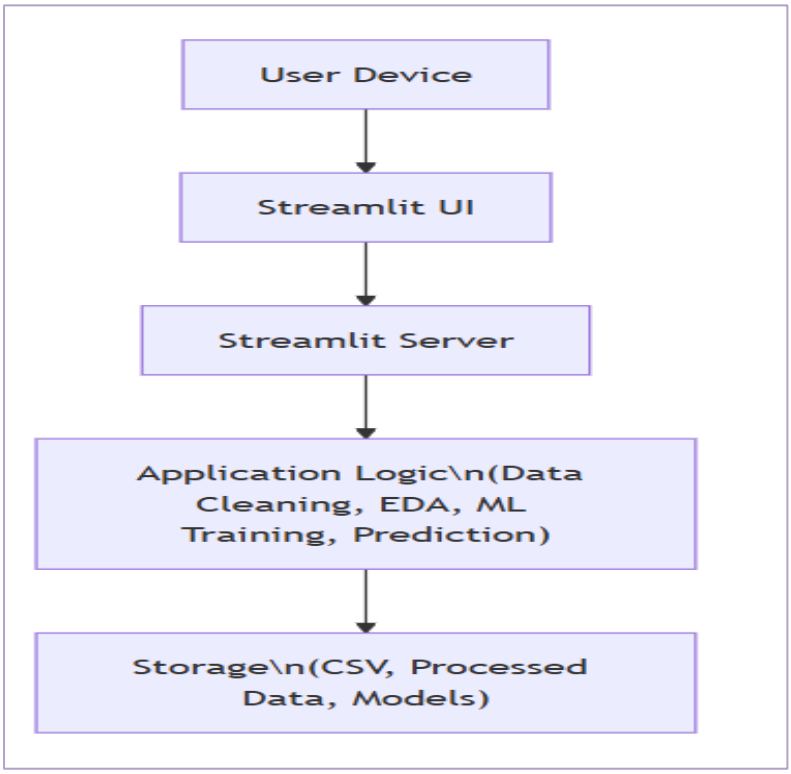

Fig.3.1 TECHNICAL ARCHITECTURE

The technical architecture of the **E-commerce sales prediciton** is based on the Machine Learning model which includes Python, Streamlit, Data cleaning, ML model. This architecture provides a scalable, secure, and efficient solution for developing a prediction model.

- **User Device:** The user accesses the system through a web browser or computer. They upload datasets, view dashboards, make predictions, and interact with the Streamlit interface without needing any technical setup or backend installation.

- **Streamlit UI:** This is the front-end interface that displays all options such as uploading data, viewing charts, training models, and generating predictions. It provides user-friendly controls, buttons, forms, and dashboards for smooth interaction.

- **Streamlit Server:** Acts as the execution layer that processes user requests. It connects the UI to the backend logic, runs Python scripts, updates charts, handles model execution, and manages data flow between components.

- **Application Logic (Data Cleaning, EDA, ML Training, Prediction):** This core engine performs all computations - cleaning raw data, generating visualizations, performing exploratory data analysis, training machine-learning models, evaluating them, and running prediction algorithms based on user inputs.

- **Storage (CSV, Processed Data, Models)**: Stores datasets, cleaned files, generated visuals, and trained ML model files locally. It ensures the system can reuse processed data and saved models for faster predictions and consistent performance.

3.2 Sequence Diagrams

Sequence diagrams illustrate interactions between system components. For instance:

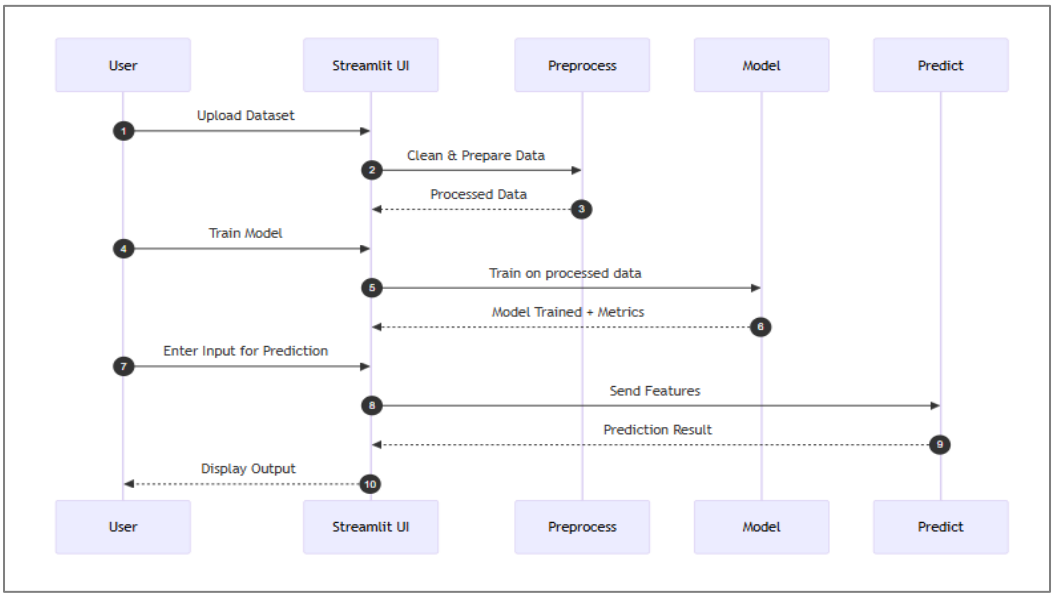

Fig.3.2 Sequence Diagram

The sequence diagram represents the complete workflow of the **E-Commerce Sales Prediction System**, showing how each component interacts to perform data processing, model training, and prediction generation. The process begins when the **User uploads a dataset** through the Streamlit UI. This file upload triggers the first interaction, where the **Streamlit UI forwards the dataset to the Preprocess module**. The Preprocess component performs essential tasks such as cleaning missing values, converting data types, removing inconsistencies, and engineering additional features required for analysis and model training. Once the preprocessing step is completed, the cleaned and transformed dataset is returned to the Streamlit interface.

Next, the User chooses to **train the machine-learning model**. This action instructs the Streamlit UI to send the processed dataset to the **Model module**, where the system applies the selected machine-learning algorithm (e.g., Linear Regression, Random Forest, SVM). The Model module trains the algorithm, validates it, and calculates evaluation metrics such

as RMSE, MAE, and R^2. These metrics are then returned to the UI so the User can assess how well the model has learned from the data.

When the User wants to make predictions, they **enter input values manually or upload a prediction CSV file**. The Streamlit UI passes these inputs to the **Predict module**, which loads the trained model and extracts relevant features from the input. After processing the features, the Predict module generates the predicted value (sales amount, demand, etc.) and returns the result to the Streamlit UI.

Finally, the **Streamlit interface displays the output** to the User in the form of numerical predictions, graphs, and downloadable reports. This entire sequence demonstrates how data flows through the system, how each module contributes to analysis and forecasting, and how the User interacts with the system to achieve actionable insights.

3.3 Class Diagram

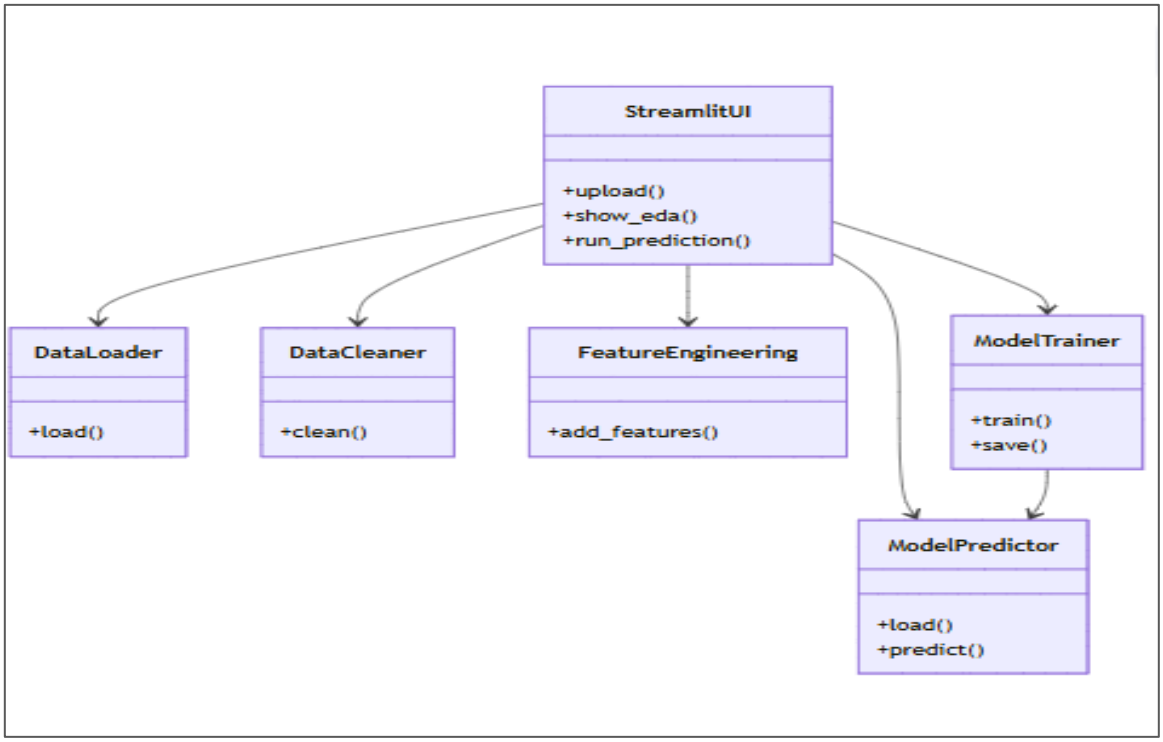

Fig.3.3 CLASS DIAGRAM

The class diagram represents the internal structure of the E-Commerce Sales Analysis and ML Prediction System, showing how different components interact to perform data processing, model training, and prediction.

1. StreamlitUI

- Acts as the central controller and user interface.
- **upload()** allows users to upload CSV datasets into the system.
- **show_eda()** displays visualizations such as charts, summaries, and dashboards.
- **run_prediction()** initiates the prediction workflow for new input data.
- Connects to other classes to coordinate the full data-to-prediction pipeline.

2. DataLoader

- Responsible for reading raw datasets.
- **load()** imports CSV/Excel files into DataFrame format.

- Ensures data is correctly loaded before cleaning.

3. DataCleaner

- Handles preprocessing tasks.
- **clean()** removes duplicates, handles missing values, formats columns, and standardizes data types.
- Prepares the dataset for feature engineering and modeling.

4. FeatureEngineering

- Adds meaningful derived features for better model performance.
- **add_features()** creates new attributes such as month, category codes, discount indicators, etc.
- Enhances dataset quality for machine learning.

5. ModelTrainer

- Responsible for model development.
- **train()** fits machine-learning algorithms (Regression, Random Forest, etc.) on processed data.
- **save()** stores trained models for future use.

6. ModelPredictor

- Handles prediction generation.
- **load()** retrieves saved models.
- **predict()** generates sales predictions for user input or batch files.

This class diagram shows a **modular and well-organized architecture**, where each class performs a specific role, and the StreamlitUI orchestrates the entire workflow from data upload to prediction output.

3.4 Data Flow Diagram (DFD)

The DFD provides an overview of data movement:

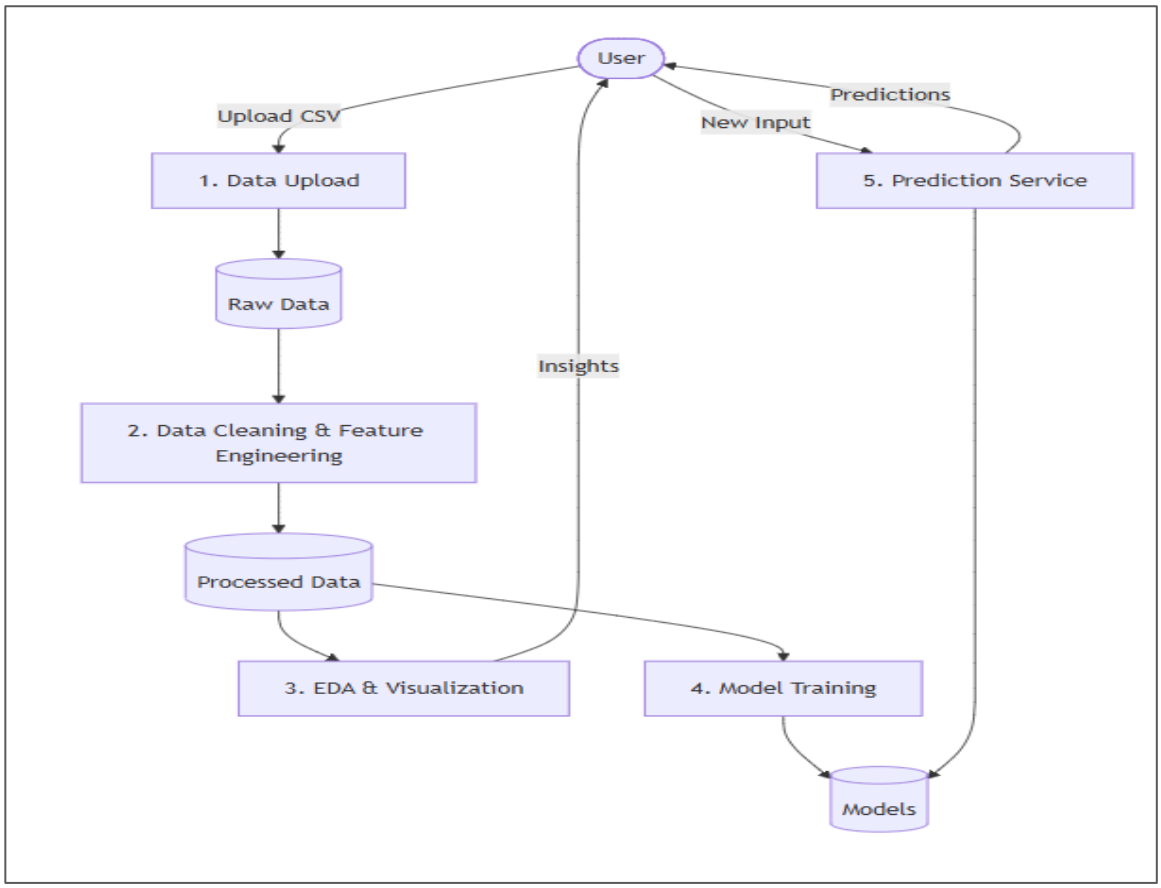

Fig.3.4 DFD DIAGRAM

The Data Flow Diagram (DFD) of the **E-commerce sales prediction** visually represents the flow of data between various components of the system:

seamlessly. The Data Flow Diagram represents the complete process flow of the E-commerce Sales Analysis and Machine Learning Prediction System. It visually maps how data moves through the system—from initial upload to final prediction output—and how the user interacts with each component. The process begins with the User, who uploads a CSV file containing historical e-commerce data. This file enters the system as Raw Data, which forms the foundation for all further operations.

The raw data is passed into the **Data Cleaning & Feature Engineering** module, where it undergoes preprocessing operations such as handling missing values, correcting formats, removing duplicates, and creating useful derived features. This ensures that the dataset is consistent, clean, and ready for analysis. The outcome of this stage is **Processed Data**, which represents a refined dataset suitable for analytics and machine-learning training.

From here, the processed data flows into two parallel paths. One path leads to the **EDA & Visualization** component, where charts, summaries, and statistical insights are generated. These visual insights are returned directly to the **User**, helping them understand data behavior, sales patterns, and trends before making predictions or training models. The second path sends the processed data to the **Model Training** module, where machine-learning algorithms are applied. This results in trained **Models**, which are stored for prediction tasks.

The **Prediction Service** connects these trained models with new user input. When the user provides fresh data, the Prediction Service loads the appropriate model and generates **Predictions**, which are returned to the user in an interpretable form. This completes the analytical loop, allowing users to move from raw data to insights and predictions seamlessly.

- The diagram highlights a cyclical workflow, allowing continuous data updates and model retraining.
- Parallel data paths ensure both insights and models can be created from the same processed data.
- The Prediction Service acts as a bridge between stored models and user-provided inputs.
- The user remains involved at three critical points: uploading data, viewing insights, and receiving predictions.
- Data moves in a logical, linear sequence, ensuring clear traceability from raw input to final output.

3.5 User Interface Design

The E-commerce Sale Prediction features a clean and intuitive design:

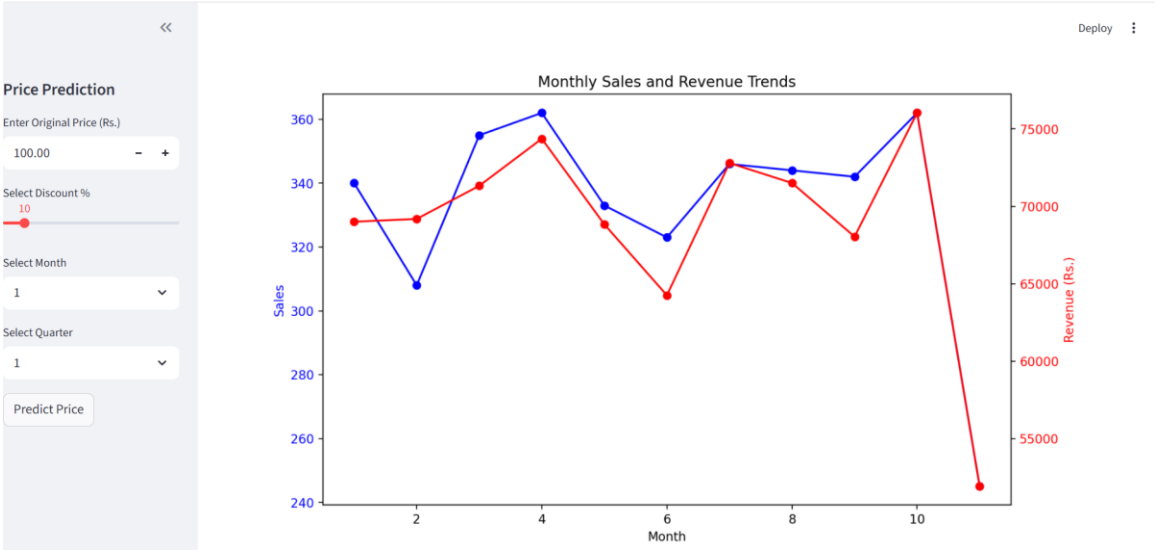

Fig.3.5(a) User Interface

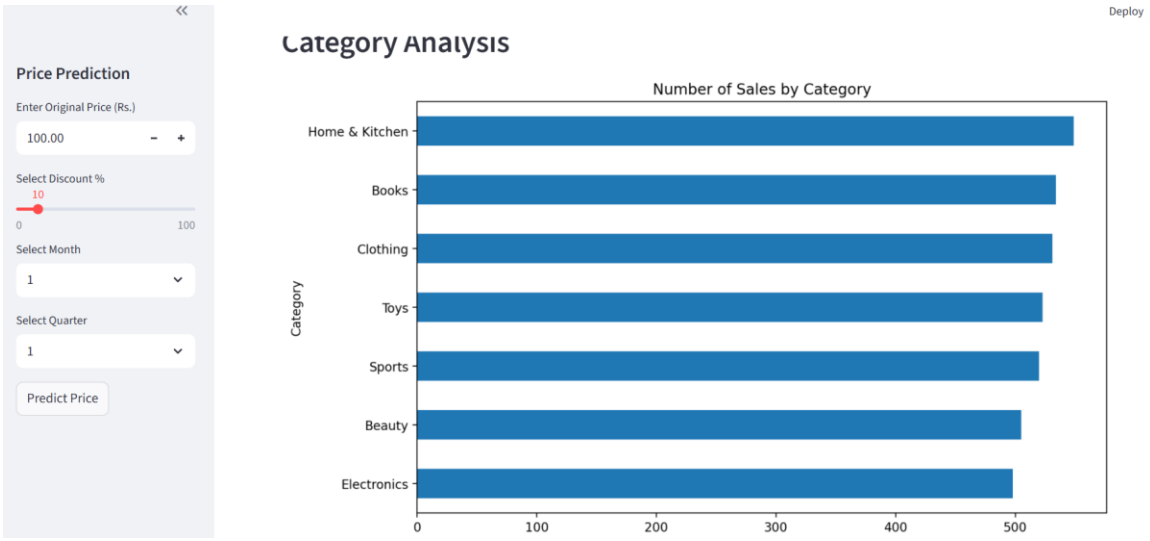

Fig.3.5(b) User Interface

This user interface presents a clean and intuitive layout designed for easy interaction with the E-Commerce Sales Analysis and Prediction System. On the left panel, users can input values for price prediction, including original product price, discount percentage, month, and quarter. Interactive components such as sliders, dropdowns, and number inputs make the process simple and user-friendly. The “Predict Price” button allows users to instantly generate predicted effective prices based on the selected parameters.

On the **right side**, the interface displays a detailed visualization titled **“Category Analysis”**, showing the number of sales in each product category using a horizontal bar chart. This chart helps users quickly understand category-wise performance and identify top-selling segments such as Home & Kitchen, Books, Clothing, and Electronics. The clean design, clear labels, and structured layout make the interface efficient for both analysis and prediction tasks, offering a professional dashboard-like experience suitable for business insights.

3.6 Data Design

3.6.1 Schema Definitions

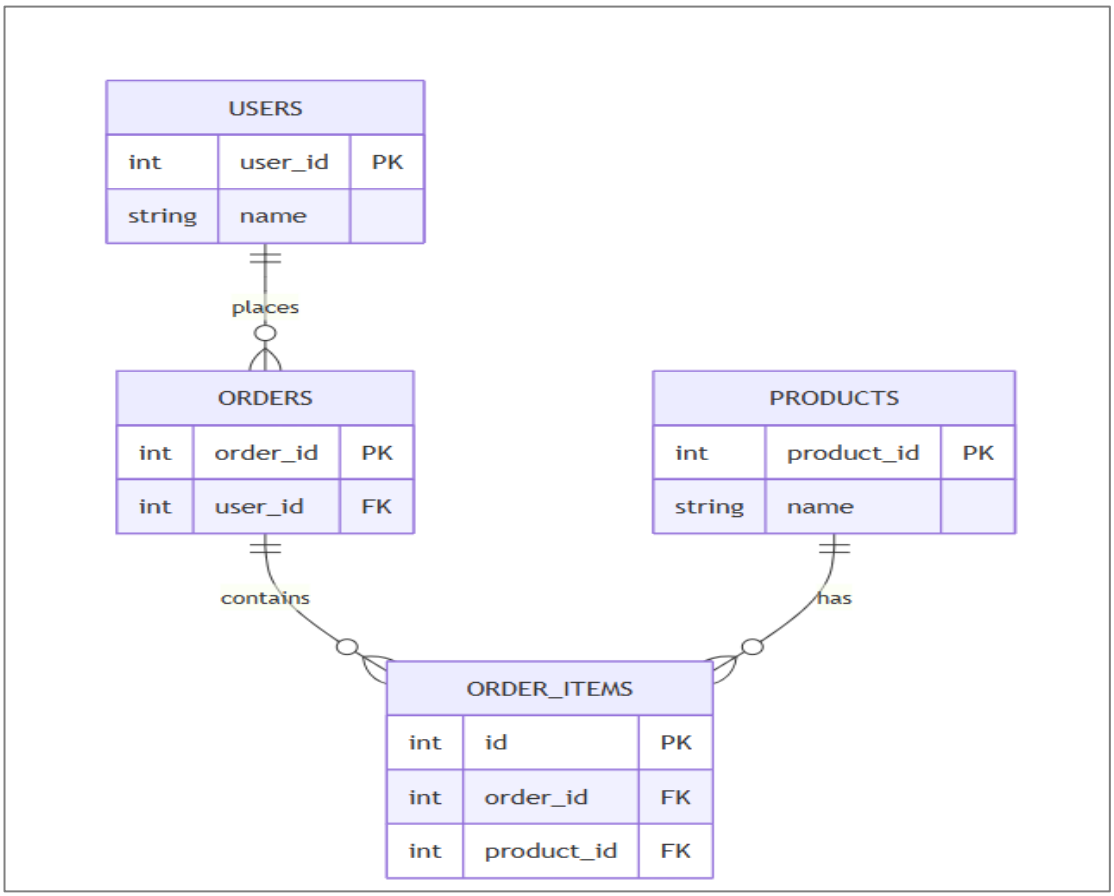

Fig.3.6.1 Schema Definition

The schema represents the core structure of an e-commerce database used for sales analysis and prediction. The USERS table stores customer information, identified uniquely by user_id, with each user able to place multiple orders. This creates a one-to-many relationship between USERS and ORDERS.

The **ORDERS** table records each purchase event, containing a unique order_id and a foreign key user_id linking it to the customer who placed the order. Since an order can contain multiple products, the **ORDER_ITEMS** table acts as a **junction table**, connecting orders to products. It stores order_id and product_id as foreign keys, enabling representation of every item within an order.

The **PRODUCTS** table holds details about each product, uniquely identified by product_id. A single product may appear in many orders, forming a **many-to-many relationship** with orders through ORDER_ITEMS.

This schema ensures structured storage of user behavior, order patterns, and product demand—supporting accurate visualization and machine-learning predictions.

3.6.2 E-R Diagram

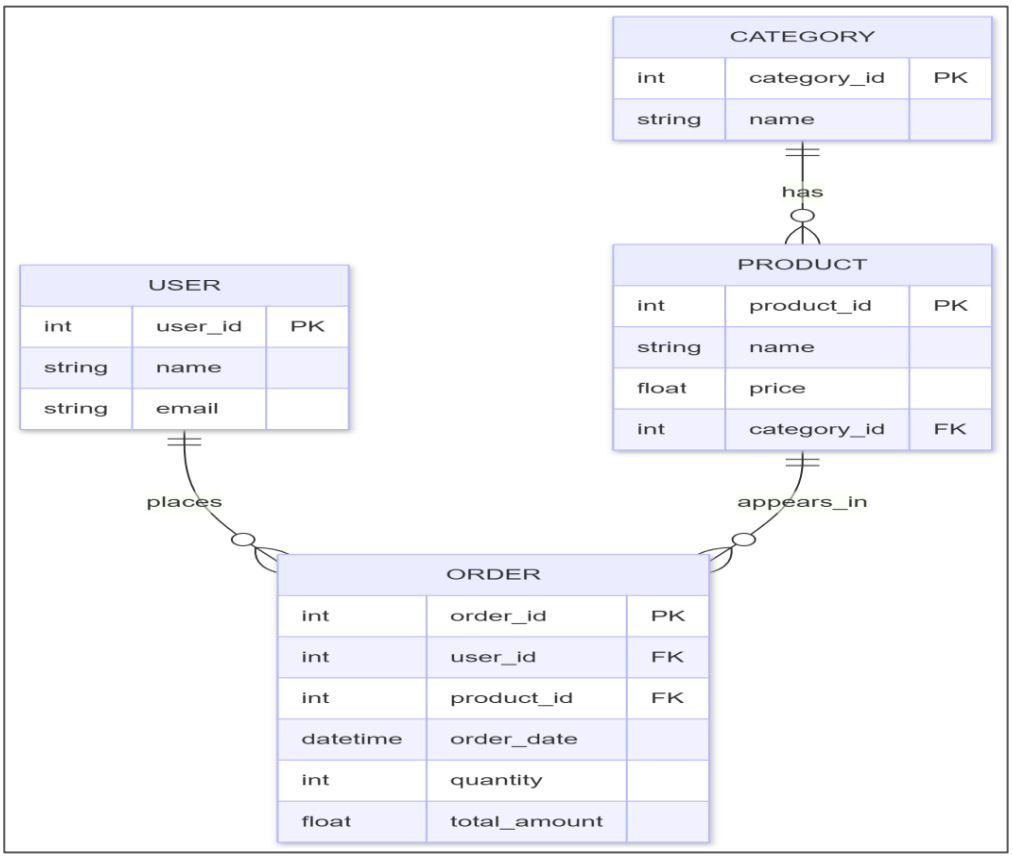

Fig.3.6.2 ER DIAGRAM

The ER diagram represents a simplified e-commerce database structure. A **User** can place multiple **Orders**, forming a one-to-many relationship. Each order is linked to a single **Product**, which provides details such as name and price. Every product belongs to a

Category, creating another one-to-many relationship where one category contains many products. The Orders entity stores transactional information including date, quantity, and total amount. Together, these entities model how users interact with products through orders, enabling analysis of purchase patterns, category-wise sales, and product performance. This structure supports data cleaning, visualization, and building machine-learning models for sales prediction.

CHAPTER 4: IMPLEMENTATION & TESTING

4.1 Methodology

The E-Commerce Sales Prediction System follows a structured development methodology based on the **Agile model**. Agile development allows continuous improvements, iterative refinement of data preprocessing, visualization, and ML model accuracy. This approach is ideal for a data-driven project because datasets may change, model performance may require tuning, and UI elements need frequent adjustments after feedback.

The development process is divided into phases such as requirement analysis, dataset understanding, preprocessing, exploratory data analysis (EDA), feature engineering, machine-learning model development, testing, and final deployment through Streamlit. Each iteration improves system accuracy, user experience, visualization quality, and prediction efficiency.

4.1.1 Proposed Algorithm

Algorithm for Data Preprocessing & Feature Engineering:

- User uploads CSV dataset.
- System loads dataset and checks structure (columns, datatypes).
- Missing values are handled (imputation/removal).
- Duplicate rows are removed.
- Dates are converted into Day, Month, Year components.
- New features are added (e.g., Month Name, Revenue = price × qty).
- Final processed dataset is stored for EDA and modeling.

Algorithm for Model Training (ML-Based Prediction):

- System receives processed dataset.
- Features (X) and target variable (y) are selected.

- Dataset is split into train and test sets.
- ML model (Linear Regression / Random Forest) is trained.
- Model performance is evaluated (RMSE, MAE, R^2).
- The best-performing model is saved using joblib.

Algorithm for Model Training (ML-Based Prediction):

- User gives new input or uploads a batch CSV.
- System loads saved ML model.
- Features are extracted and pre-processed.
- Model generates prediction values.
- Results are displayed and downloadable.

4.2 Implementation Approach

The **E-Commerce Sales Prediction** System is built using technologies designed for scalability, interactive use, and rapid data processing.

- **Frontend and Backend**:
    - **Streamlit UI:** Used for dataset upload, visualization display, model training, and prediction workflow.
    - **Visualization Tools:** Matplotlib, Seaborn, and Plotly are used to create charts and dashboards.
- **Backend Logic**:
    - **Python:** Handles data cleaning, feature engineering, model training, and prediction logic.
    - **ML Libraries:** Scikit-learn is used to train models (Linear Regression, Random Forest, SVM)

- **Data Handling:** Pandas and NumPy are used for manipulating large datasets efficiently.

- **Hosting and version control**:
    - **Version Control:** Git is used for collaboration and version tracking.
    - **Hosting:** Streamlit Cloud or AWS can be used for deployment of the final system.

4.2.1 Introduction to Languages, IDEs, Tools, and Technologies

- **Languages**:
    - **Python:** Core language for preprocessing, EDA, machine learning, and prediction logic.
- **IDEs**:
    - Visual Studio Code, Jupyter Notebook for efficient code development and debugging.
- **Tools**:
    - **Streamli**t for building the user interface.
    - **Git** for version control.
    - **Scikit-learn** for ML model development.
    - **Pandas/NumPy** for data manipulation.
    - **Matplotlib/Seaborn/Plotly** for visualization.

4.3 Testing Approaches

Testing is an essential phase to ensure correctness, reliability, and prediction accuracy. Various testing methods were applied:

- **Unit Testing**
  Individual modules like DataLoader, DataCleaner, ModelTrainer, and ModelPredictor were tested separately to verify correct functionality.
- **Integration Testing**
  The interaction between modules—data preprocessing → feature engineering → model training → prediction—was tested to ensure smooth workflow.
- **System Testing**
  End-to-end workflows, from uploading datasets to generating predictions, were tested for correctness, usability, and UI responsiveness.

4.3.1 Unit Testing

Unit testing focuses on verifying the functionality of individual components of the E-commerce sales prediction. Each feature was tested separately to ensure it worked as expected.

Test Scenarios for Unit Testing:

- **Data Loading:** Checks if CSV files load correctly and invalid files prompt errors.
- **Missing Value Handling:** Ensures preprocessing removes or imputes missing values.
- **Feature Engineering:** Verifies that new features (month, revenue, etc.) are created correctly.
- **EDA Charts:** Confirms charts render without errors using sample datasets.
- **Model Training:** Ensures models train successfully and produce valid metrics.
- **Model Saving/Loading:** Confirms joblib files are stored and loaded correctly.
- **Prediction:** Checks predictions return valid numeric values for valid inputs.
- **Batch Prediction:** Ensures uploaded CSV prediction files produce correct outputs.
- **Error Handling:** Verifies that incorrect user inputs display error messages.

4.3.2 Integration Testing

Integration testing was conducted to ensure that different components of the E-commerce sales prediction work seamlessly together:

Test Scenarios for Integration Testing:

- **Dataset Upload → Cleaning:** Verified raw data flows correctly into the cleaning module.
- **Cleaning → Feature Engineering:** Ensured processed data is passed without loss.
- **Feature Engineering → Model Training:** Confirmed that engineered features are used during training.
- **Model Training → Prediction:** Ensured trained models are correctly loaded for predictions.
- **Streamlit UI → Python Backend:** Verified UI buttons correctly trigger backend functions.
- **Prediction Input → Output Rendering:** Confirmed results appear correctly on screen.
- **Batch Prediction → Download:** Tested CSV upload and correct downloadable output file generation.
- **Visualization Integration:** Ensured EDA plots update based on cleaned dataset.
- **Storage Verification:** Confirmed processed data and model files are stored and retrieved correctly.

CHAPTER 5: RESULTS & DISCUSSION

5.1 User Interface Representation

The E-Commerce Sales Prediction System provides a clean, intuitive, and interactive interface developed using **Streamlit**. The UI is simple, responsive, and designed for easy navigation across different modules such as data upload, data cleaning, visualizations, model training, and prediction. The interface uses a blue–white analytics theme that offers a professional and dashboard-like experience. Users can seamlessly move from preprocessing to visualization and finally to prediction with minimal technical knowledge.

5.1.1 Brief Description of Various Modules

- **Dataset Upload Module:**

  Allows the user to upload CSV datasets. The system validates file format and structure before processing.

- **Data Cleaning & Preprocessing Module:**

  Automatically handles missing values, duplicate records, inconsistent formats, and creates new features such as month, day, or total revenue. Ensures clean and structured data for analysis.

- **EDA & Visualization Module:**

  Generates interactive visualizations including sales trends, category-wise revenue, top-selling products, and customer behavior graphs. Provides instant insights directly on the dashboard.

- **Model Training Module:**

  Trains machine-learning models such as Linear Regression or Random Forest. Displays evaluation metrics like RMSE, MAE, and R^2. Saves the best-performing model automatically.

- **Prediction Module:**

  Allows users to make predictions using trained models. Users can enter single inputs manually or upload a CSV file for batch predictions.

- **Report Download Module:**

  Provides downloadable prediction reports, cleaned datasets, and visual charts in CSV or PNG format

5.2 Snapshot of System with Brief Description

● **Upload Page:**

Users upload raw datasets (.csv files). The UI displays dataset previews including column names, data types, and row counts.

● **Data Cleaning Page:**

Shows missing value counts, duplicate detection results, and applied preprocessing steps. Users can view before-and-after data samples.

● **Visualization Dashboard:**

Displays interactive charts such as line graphs for monthly sales, bar charts for product performance, pie charts for category distribution, and heatmaps for correlations. Helps users understand trends visually.

● **Model Training Page:**

Users select algorithms, view training progress, and see performance metrics such as RMSE, MAE, and accuracy score. Best model is saved for prediction.

● **Prediction Page:**

Users provide new input or a CSV file. The system returns predicted sales values in real-time along with optional charts.

● **Download Page:**

Allows exporting prediction results, trained models, and processed datasets for further business use.

5.3 Database Description

The system uses structured **CSV datasets** that represent real e-commerce activities such as customer purchases, product details, order history, and category information. These datasets form the basis for data cleaning, visualization, and ML model development.

5.3.1 Snapshot of Database Tables with Brief Description

- **Users Table:**

  Contains user-level information such as user_id, customer name, email, and basic demographic details. Helps in user-level sales and behavior analysis.

- **Products Table:**

  Stores product_id, product name, price, and category_id. Useful for category-wise sales prediction and ranking top products.

- **Categories Table:**

  Contains category_id and category_name. Helps organize products and analyze category-level performance.

- **Orders Table:**

  Stores order_id, user_id, product_id, order_date, quantity, and total_amount. Primary source for sales analysis, time-series trends, and ML model training.

5.4 Final Findings

The E-Commerce Sales Prediction System meets all the defined objectives by providing an end-to-end workflow from data upload to prediction. Key observations include:

- **Secure User Authentication:** The system uses JWT for secure login, ensuring user data protection.
- **Effective Data Processing:** Preprocessing significantly improves dataset quality and reduces model errors.

- **Insightful Visualizations:** Users gain valuable insights from trends, customer behaviour charts, and revenue analytics.
- **Accurate Machine-Learning Models:** Trained models show high prediction accuracy with strong RMSE and R^2 scores
- **User-Friendly Interface:** Streamlit UI allows even non-technical users to explore data and generate predictions.
- **Scalable Architecture:** New datasets, models, or features can be integrated without structural changes.
- **Practical Business Value:** Predictions support inventory planning, marketing decisions, and sales forecasting.

The system can be further enhanced with features such as ARIMA time-series forecasting, product recommendation systems, automated model retraining, and real-time dashboards.

CHAPTER 6: CONCLUSION & FUTURE SCOPE

6.1 Conclusion

The E-Commerce Sales Analysis and Prediction System successfully delivers a complete, interactive, and intelligent platform for analyzing sales data and generating accurate predictions. Built using Python, Pandas, Scikit-Learn, and Streamlit, the system ensures a scalable, efficient, and user-friendly architecture suitable for business analytics and academic research.

The system achieves its core objectives by providing automated data cleaning, insightful visualizations, and reliable machine-learning predictions. Users can upload datasets, explore sales trends, analyze customer behavior, visualize category-wise performance, and understand product demand through interactive charts and dashboards. The system also allows seamless model training, evaluating performance using metrics like RMSE, MAE, and R^2, and generating predictions for both single inputs and batch data.

Streamlit ensures a smooth and responsive user interface, enabling even non-technical users to interact with the system without difficulty. The modular implementation of preprocessing, visualization, and machine learning makes the system easy to maintain and expand. The project also ensures data reliability by handling missing values, formatting inconsistencies, and creating meaningful engineered features that improve prediction accuracy.

6.2 Future Scope

Table.6.1 Comparison Table

Sr. No.	Name of Solution/System	Fatures	Limitations/ Drawbacks
1	E-commerce Data Analysis and Machine Learning Prediction System	Provides detailed sales, customer, and product analytics.	Requires clean and well-structured data for accurate predictions
2	Google Data Studio / Power BI Dashboard	Provides interactive visualizations and business intelligence dashboards.	Limited advanced predictive capabilities
3	Excel-based E-commerce Report System	Simple spreadsheet-based data analysis and reporting tool.	Time-consuming for large datasets and complex analysis.

The E-Commerce Sales Analysis and Prediction System offers a strong foundation for data-driven decision-making, but it also presents several opportunities for further enhancement and expansion. As the volume of e-commerce data continues to grow, future iterations of this system can integrate more advanced analytical techniques to improve accuracy, usability, and business value.

One major enhancement is the integration of **advanced forecasting models**, such as ARIMA, Prophet, LSTM, or other deep learning architectures, which can provide more precise long-term sales predictions. These time-series models are particularly useful for predicting seasonal trends, demand fluctuations, and peak sales periods. Another improvement is **real-time data processing**, enabling continuous updates to dashboards and sales predictions as new orders are recorded. With streaming technologies like Apache Kafka or Firebase, the system can evolve into a live monitoring and forecasting platform.

The system can also be expanded to support **customer behavior analysis**. By incorporating customer segmentation, recommendation engines, and churn prediction, businesses can gain insights into customer lifetime value, purchase probability, and product

preferences. This would help in personalized marketing, targeted promotions, and improved customer retention strategies.

In addition, the system can integrate **automated model retraining**, where the model periodically updates itself with newly available sales data. This ensures that predictions remain accurate even as market conditions change. Another potential enhancement is the development of a **mobile-friendly interface** or a standalone mobile application to make analytics accessible on the go.

Support for **multi-store or multi-category forecasting**, integration with ERP systems, and cloud-based deployment using AWS, Azure, or Google Cloud can further scale the system for enterprise-level use. Implementing **role-based access control (RBAC)** will make the platform more secure, enabling administrators, analysts, and managers to access different functionalities.

Overall, the future scope of the system includes deeper automation, advanced analytics, real-time capabilities, and enhanced user experience—making it a powerful tool for modern e-commerce businesses.

REFERENCES

[1] Kaggle – Brazilian E-commerce Public Dataset by Olist (used for real-world sales, customers, and delivery data).

[2] Aurélien Géron – Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow (for ML concepts and implementation)

[3] Choudhary, S., & Shukla, S. (2020). "Sales Forecasting for E-commerce Using Machine Learning Techniques" — International Journal of Scientific Research in Computer Science and Engineering (IJSRCSE).

[**4] Tsai, C. F., & Hung, C. S. (2019).** "Customer Purchase Behavior Prediction in E-commerce using Data Mining Techniques"

[5] IBM Analytics. (2021). *Predictive Analytics for Retail and E-commerce*. IBM Research Publications — Explains real-world applications of predictive models in online sales and consumer insights.

[6] Li, X., & Xu, Y. (2020). "E-commerce Sales Forecasting Using Machine Learning Models" — *IEEE International Conference on Big Data and Smart Computing*.

Appendix A: Project Synopsis

1. INTRODUCTION

1.1 Overview

The E-commerce Sales Analysis and Prediction System is a data-driven software solution designed to analyze online shopping data, visualize trends, and build machine learning models to predict future sales, revenue, customer behavior, and product demand.

This project helps in understanding customer buying patterns, popular product categories, seasonal sales trends, and predicting future demand using historical data.

1.2 Purpose

The purpose of this project is to analyze real-world e-commerce sales data to uncover meaningful insights and use machine learning techniques to predict future outcomes such as sales trends, revenue, customer behavior, and delivery performance. In today's rapidly evolving digital marketplace, e-commerce platforms generate vast amounts of data from customer purchases, payments, product categories, and logistics. However, this raw data is rarely used to its full potential. Through this project, the aim is to transform raw data into valuable business intelligence by applying data preprocessing, exploratory data analysis (EDA), visualization, and predictive modeling. This system will help businesses make informed decisions about inventory management, marketing strategies, pricing models, and customer satisfaction. Ultimately, the project intends to build a smart, data-driven solution that not only visualizes trends but also forecasts future sales and enhances the overall efficiency of e-commerce operations.

2. LITERATURE SURVEY

2.1 Existing Problem

Although e-commerce platforms collect large volumes of data, most businesses do not fully utilize it for analytical or predictive purposes. Traditional sales tracking systems only focus on recording transactions and generating static reports, without offering deeper insights into customer preferences, product demand, seasonal trends, or future revenue predictions. Many small and medium-sized businesses still rely on manual spreadsheets, which are time-consuming, error-prone, and provide no predictive intelligence. There is no automated system to analyze customer buying behavior, predict sales, detect delivery delays, or recommend inventory stocking levels. Additionally, without proper visualization tools, identifying important trends such as the most profitable products, high-return items, or cities with maximum sales becomes difficult. As a result, companies often make business decisions based on assumptions rather than data-driven insights, leading to losses, poor customer satisfaction, and inefficient use of resources.

2.2. Proposed Solution

To overcome these challenges, this project proposes a complete data-driven E-commerce Sales Analysis and Prediction System using Python, data analytics, and machine learning. The system will collect historical e-commerce data, either from real-world sources like the Olist dataset (Kaggle) or through synthetic datasets. The data will go through preprocessing steps such as cleaning, handling missing values, formatting dates, and generating new useful features like monthly sales, revenue per customer, discount impact, and delivery time. Exploratory Data Analysis (EDA) will be performed to identify patterns in sales, customer demographics, product performance, and revenue distribution. Interactive visualizations will be created using Matplotlib or Plotly to display trends clearly data.

3. THEORETICAL ANALYSIS

3.1 Block Diagram

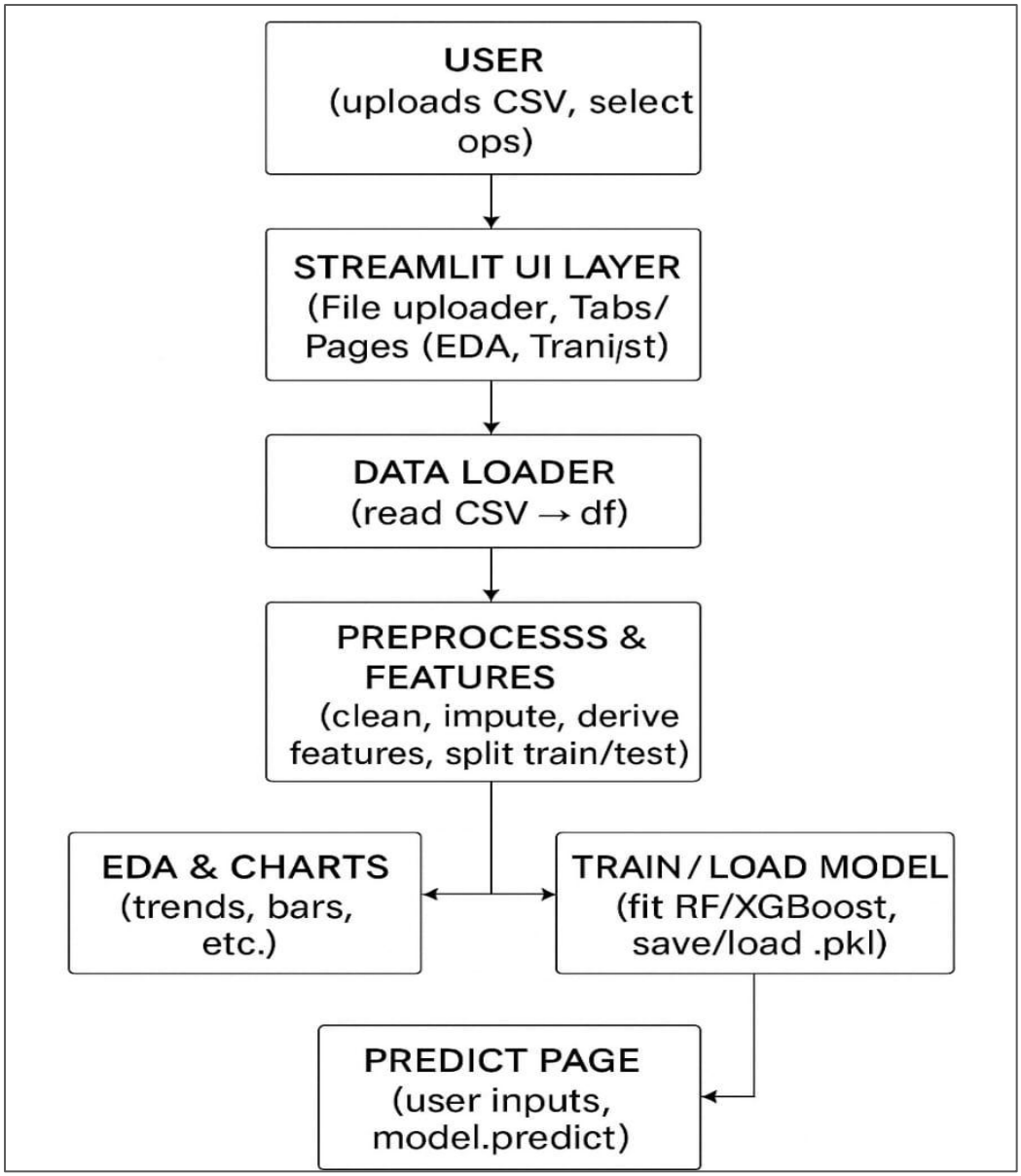

Fig.1. Workflow of E-Commerce Sales Analysis

3.2 Hardware/Software Designing

1. **Hardware Requirements:**

- Processor: Intel i3/i5/i7 or Ryzen 3/5
- RAM: 8GB
- Storage: 128 GB HDD or more
- Network: 10/100 Mbps Internet Connection
- OS: Linux-Based or Windows Server

2. **Software Requirements:**

- Programming Language: Python
- Libraries: Pandas, NumPy, Matplotlib, Seaborn
- ML frame works: Scikit-Learn / XGBoost
- Visualization: Matplotlib / Plotly / Power BI / Streamlit
- IDE / Tools: Jupyter Notebook / VS code
- Dataset: E-commerce Dataset (Kaggle)
- Version Control: Git with GitHub

4. APPLICATIONS

- **Sales Forecasting and Revenue Prediction**
  Businesses can predict future sales and revenue using historical data trends. This helps in planning budgets, marketing strategies, and business expansion.

- **Inventory and Stock Management**
  By predicting product demand, businesses can maintain the right amount of stock, preventing overstocking or stockouts, reducing storage costs, and improving supply-chain efficiency.

- **Customer Behavior and Personalization**
  The system helps in analyzing customer preferences, buying frequency, and spending patterns. This allows companies to personalize recommendations and improve customer experience.

- **Pricing and Discount Strategy Optimization**

  By studying the relationship between discounts and sales, businesses can determine the best discount rates that increase sales without reducing profit margins.

Appendix B: Guide Interaction Report

The development of the E-Commerce Sales Analysis and Prediction System was carried out under the guidance of **Prof. Mahendra Verma**. Regular discussions and review sessions were conducted to ensure that the project aligned with academic expectations, technical requirements, and industry-relevant practices. The guidance provided throughout the project helped refine the system's architecture, improve implementation quality, and enhance the analytical capabilities of the application.

1. Initial Interaction: The project concept was presented to Prof. Mahendra Verma, who recommended focusing on data preprocessing, exploratory data analysis (EDA), and machine-learning-based sales prediction. The guide emphasized using Python, Streamlit, and Scikit-Learn to build a scalable and interactive analytical platform. He also highlighted the importance of dataset selection and realistic business parameters.

2. Design Phase Review: During the design stage, the guide reviewed the system architecture, ER diagram, and data flow diagrams. He advised ensuring a modular design for components like preprocessing, visualization, model training, and prediction modules. The importance of meaningful visualizations and a professional dashboard layout was stressed.

3. Implementation Phase Review: The Guide evaluated the implementation approach, suggesting improvements in the data cleaning pipeline, feature engineering logic, and overall code organization. Emphasis was placed on using proper ML evaluation metrics, modular Python scripts, and ensuring that the Streamlit UI remains simple, responsive, and user-friendly.

4. Testing and Evaluation: The guide reviewed the system's testing strategy, including unit tests for preprocessing functions and integration tests for visualization and prediction workflows. He highlighted the importance of validating model accuracy using RMSE, MAE, and R^2, and ensuring input validation for prediction modules.

5. Final Review: After project completion, the full system was demonstrated to Prof. Verma. He appreciated the clean interface, effective visualizations, and accurate prediction results. He encouraged adding advanced forecasting models, more business-level KPIs, and real-time dashboards in future versions.

Appendix C: User Manual

1. Accessing the Application

- Open your preferred web browser (Chrome, Firefox, Edge, Safari).
- Visit the application URL (if deployed online) or launch the Streamlit app locally using the command:

  ```
  streamlit run app.py
  ```

2. Uploading Dataset

- On the homepage, click **"Upload Dataset"**.
- Select a CSV file containing e-commerce data.
- After uploading, a preview of the dataset (first few rows and column names) will appear.
- Ensure the file contains key fields such as order details, product data, and date information.

3. Data Cleaning & Preprocessing

- Navigate to the **"Data Cleaning"** section.
- The system will automatically detect missing values, duplicates, and formatting inconsistencies.
- Click **"Clean Data"** to apply preprocessing steps including:
  - Handling missing values

    - Removing duplicates
    - Converting date columns
    - Generating essential features (month, day, revenue, etc.)
- A cleaned dataset preview will be displayed for verification.

4. Exploring Data (EDA & Visualizations)

- Click on the **"Visualization Dashboard"**.
- View interactive charts such as:
    - Monthly sales trend
    - Category-wise revenue
    - Top-selling products
    - Customer purchase behavior
- Hover over graphs to see detailed insights on specific data points.

5. Training the Machine Learning Model

- Go to the **"Model Training"** section.
- Choose an algorithm (e.g., Linear Regression, Random Forest).
- Click **"Train Model"** to begin training.
- After training, the system displays performance metrics such as RMSE, MAE, and R^2.
- The best-performing model is automatically saved for prediction.

6. Making Predictions

- Open the **"Prediction"** page.
- Choose between:

    - **Single Input Prediction** — manually enter required values.
    - **Batch Prediction** — upload a CSV containing multiple rows.
- Click **"Predict"**.
- The predicted sales values will be displayed instantly.

7. Downloading Reports

- After predictions, click **"Download Results"** to export prediction outputs in CSV format.
- You may also download cleaned datasets, visual charts, or model performance summaries.

Appendix D: Git/GitHub

https://github.com/hariiom08/Major-1.git

https://github.com/Codshubham/Major-project-.git

Pic. GitHub Commits of the project