SALES DEMAND FORECASTING FOR RETAIL ERP CLIENTS

AN INTERNSHIP REPORT

Submitted by

Gabby J

(RRN:220071601073)

In partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

NOVEMBER 2025

B.S. Abdur Rahman® Crescent Institute of Science & Technology
Deemed to be University u/s 3 of the UGC Act, 1956

BONAFIDE CERTIFICATE

Certified that this Internship report **"SALES DEMAND FORECASTING FOR RETAIL ERP CLIENTS"** is the Bonafide work of **"GABBY J (RRN: 220071601073) "** who carried out the internship under my supervision. Certified further, that to the best of our knowledge the work reported herein does not form part of any other internship report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Mrs. K. SOWMIYA
Assistant Professor
Department of CSE
B.S. Abdur Rahman Crescent
Institute of Science and Technology
Vandalur, Chennai – 600 048.

INTERNSHIP CERTIFICATE

FemtoSoft

A Software Design & Development Company

CERTIFICATE NO:- FST-LH-2436

21.07.2025

INTERNSHIP COMPLETION LETTER

This is to certify that **Ms.Gabby J (Reg. No:220071601073) and (Femtosoft Enrollment No: 376)** has successfully completed the Internship on **"Data Science"** under our guidance from 19th June 2025 to 19th July 2025 in our company.

We appreciate your dedication and hard work during your time with us. Following a comprehensive evaluation of your performance throughout the internship, we have assessed you as part of the internship program with us.

Based on our assessment of your performance during the **"Data Science"** internship, where you underwent training at Femtosoft Technologies, we found your performance to be **GOOD.** We wish you all the best for the successful career.

For Femtosoft Technologies

Femtosoft Technologies Chennai-43

Murugan A

Team - Human Resources

FEMTOSOFT TECHNOLOGIES
17/25, 1St Floor, Griffith Main Road, Essa Pallavaram, Chennai - 43.
Ph: +91 44 - 22642421, Mobile: +91 9884448421, Email : hr@femtosoft.in
Web: www.femtosoft.in

B.S. Abdur Rahman®
Crescent
Institute of Science & Technology
Deemed to be University u/s 3 of the UGC Act, 1956

VIVA VOCE EXAMINATION

The viva voce examination of the Internship titled **"SALES DEMAND FORECASTING FOR RETAIL ERP CLIENTS"** submitted by **"GABBY J (220071601073)"** is held on _____.

INTERNAL EXAMINER

ACKNOWLEDGEMENT

I sincerely express my heartfelt gratitude to Prof. **Dr. T. MURUGESAN** , Vice Chancellor and **Dr. N. THAJUDDIN**, Pro-Vice Chancellor, B.S. Abdur Rahman Crescent Institute of Science and Technology, for providing me an environment to carry out my internship successfully.

I sincerely thank **Dr. N. RAJA HUSSAIN**, Registrar, for furnishing every essential facility for doing my internship.

I thank **Dr. SHARMILA SANKAR**, Dean, School of Computer Information and Mathematical Sciences for her motivation and support.

I thank **Dr. W. AISHA BANU**, Professor and Head, Department of Computer Science and Engineering, for providing strong oversight of vision, strategic direction and valuable suggestions.

I obliged my class advisor **Ms K.SOWMIYA,** Assistant Professor, Department of Computer Science and Engineering for her professional guidance and continued assistance throughout the internship period.

I thank **Ms. KAMATCHI T,** Mentor at **FEMTOSOFT TECHNOLOGIES** for her guidance and encouragement throughout the internship period and without her guidance this project would not have been a successful one.

I thank all the **Faculty members** and the **System Staff** of the Department of Computer Science and Engineering for their valuable support and assistance at various stages of internship completion.

(GABBY J)

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ABBREVIATION	FULL FORM
CSV	Comma-Separated Values
EDA	Exploratory Data Analysis
ERP	Enterprise Resource Planning
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
SQL	Structured Query Language
DBMS	Database Management System
AI	Artificial Intelligence
RDBMS	Relational Database Management System
BI	Business Intelligence
KPI	Key Performance Indicator

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

COMPANY OVERVIEW

1.1 Company Overview

Femtosoft Technologies Private Limited is a Chennai-based IT services and software development company headquartered in Pallavaram. Founded with the mission to empower businesses through technology, Femtosoft specializes in ERP solutions, data analytics, and custom software development. The company serves clients across various sectors such as retail, manufacturing, logistics, and education, offering scalable and intelligent business solutions.

Femtosoft focuses on integrating modern technologies like Python, MySQL, Power BI, and cloud computing within its ERP systems. This integration allows businesses to automate processes, enhance operational efficiency, and make data-driven decisions.

By combining predictive analytics with operational intelligence, Femtosoft helps organizations streamline workflows and improve profitability. For instance, retail clients using Femtosoft's ERP can monitor real-time sales, forecast inventory demand, and optimize procurement strategies using advanced analytics modules.

The company operates with a strong foundation built on integrity, innovation, and customer-centric service. Its core values : Integrity, Innovation, Customer Centricity, Teamwork, and Quality guide its approach to every project. Femtosoft's mission is to empower organizations with intelligent ERP and analytics solutions, while its vision is to become a global leader in enterprise technology through innovation and sustainable business excellence.

1.2 Company Website

Website: https://www.femtosoft.in/

The official website serves as a comprehensive platform showcasing Femtosoft's products, services, and industry-specific solutions. It provides detailed insights into ERP modules, analytics tools, and automation services, along with client success stories, certifications, and case studies.

For potential clients, the website offers a clear overview of the company's technological capabilities and domain expertise, helping businesses understand how Femtosoft's ERP and analytics solutions can drive digital transformation.

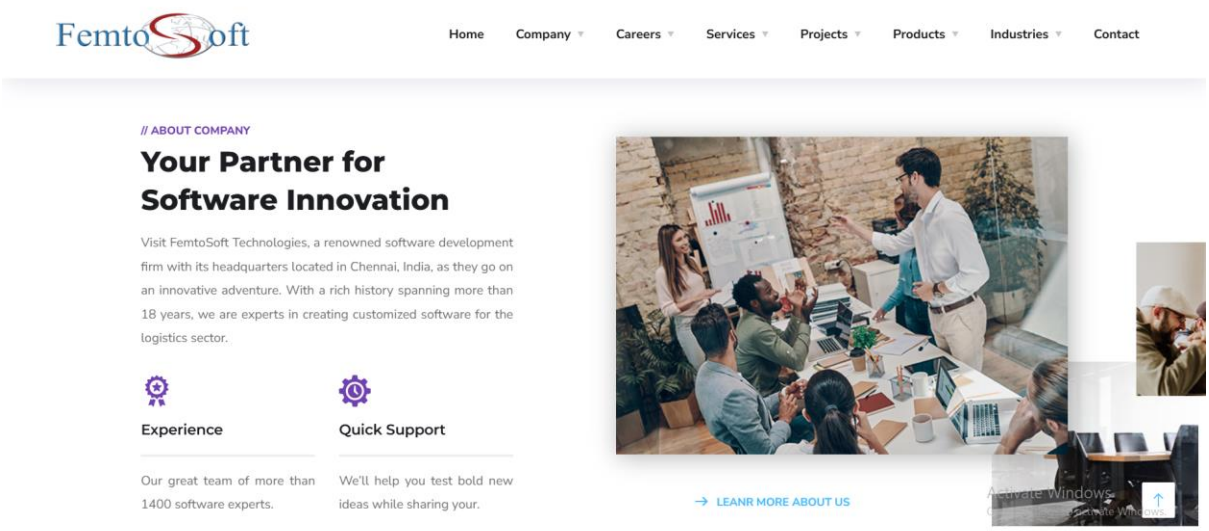

Fig. 1.1 Femtosoft Company Website

1.3 Company Logo

The Femtosoft logo symbolizes innovation, precision, and modern technology. Its sleek and minimalistic design reflects the company's focus on efficiency, intelligence, and reliability. The logo visually represents Femtosoft's mission to drive digital transformation and its commitment to delivering cutting-edge enterprise solutions.

Fig. 1.2 Femtosoft Company Logo

CHAPTER 2

ROLE AND RESPONSIBILITY

2.1 ROLE

During my internship at Femtosoft Technologies Private Limited, I served as a Data Analytics Intern under the Data Science domain. My role focused on leveraging data science methodologies to develop a Sales Demand Forecasting System for retail ERP clients. The primary objective was to apply analytical and predictive modelling techniques to assist businesses in optimizing inventory management and improving sales planning.

As part of the Data Science team, I contributed to the end-to-end development of a predictive analytics pipeline from data collection and preprocessing to model training, evaluation, and visualization. This role provided practical exposure to real-world data-driven problem-solving and integration of machine learning models into ERP systems for business intelligence applications.

2.2 RESPONSIBILITY

As a Data Analytics Intern, my main responsibilities included:

- **Data Acquisition and ETL Development** – Designed and implemented ETL workflows to extract raw sales data from CSV files and load it into a MySQL database for structured analysis.
- **Exploratory Data Analysis (EDA)** – Performed data profiling and visualization to uncover sales trends, seasonal effects, and customer behaviour patterns.
- **Model Development and Forecasting** – Applied Prophet-based time-series forecasting models to predict daily and weekly sales demand across different products and regions.
- **Model Evaluation** – Assessed model performance using statistical metrics such as Mean Absolute Percentage Error (MAPE) and refined model parameters for improved accuracy.

- **Data Visualization and Reporting** – Created interactive dashboards in Power BI to analyze and present key insights, including product category performance, regional sales comparisons, and seasonal demand fluctuations.
- **Automation and Documentation** – Automated the forecasting pipeline using Python scripts and documented the workflow, ensuring clarity and reproducibility for future enhancements.

This internship enabled me to **gain hands-on experience in data science tools and technologies** such as **Python, MySQL, Prophet and Power BI** while understanding how predictive analytics enhances retail ERP decision-making and operational efficiency.

CHAPTER 3

PROJECT OVERVIEW

3.1 Introduction to the Project

The internship project, titled **"Sales Demand Forecasting for Retail ERP Clients,"** was carried out at Femtosoft Technologies Private Limited under the Data Science domain. The project aimed to develop a predictive analytics system capable of forecasting product demand for retail businesses integrated with ERP platforms.

By analyzing historical sales data, the system provides accurate sales predictions that assist in **inventory optimization, procurement planning, and improved decision-making.**

The project incorporated various stages of the data science workflow, including data acquisition, preprocessing, feature engineering, model building, and evaluation, The project also included the development of an interactive Power BI dashboard, designed to visualize key sales metrics and forecasting results.

The dashboard allows retail managers to monitor total sales, quantity sold, category-wise performance, regional comparisons, and seasonal trends, enabling quick and informed decision-making. The implementation emphasized automation, accuracy, and scalability, ensuring the solution could be integrated into existing ERP frameworks for real-time insights.

3.2 Project Title

Sales Demand Forecasting for Retail ERP Clients

This project focuses on building a **data-driven forecasting system** that predicts future product sales across multiple stores and regions. The model helps businesses reduce stock imbalances, improve operational efficiency, and make informed business decisions.

3.3 Problem Statement

Retail businesses often rely on manual sales forecasting methods, which are prone to errors and inefficiencies.

Traditional forecasting fails to accurately capture **seasonal fluctuations, regional variations, and promotional effects** in sales patterns.

As a result, companies face challenges such as:

- Overstocking or understocking of products
- Increased holding costs and lost sales opportunities
- Lack of real-time data-driven decision-making

To address these issues, the project aimed to build an **automated forecasting pipeline** that uses historical sales data to predict future demand, helping retail ERP clients achieve better inventory control and profitability.

3.4 Objective of the Project

The primary objectives of the project were:

- To design a data extraction and ETL pipeline to load sales data from CSV files into a MySQL database.
- To perform data cleaning, transformation, and feature engineering for improved model accuracy.
- To apply Prophet time-series forecasting to predict daily and weekly product-level sales.
- To evaluate the model's accuracy using statistical measures such as MAPE (Mean Absolute Percentage Error).
- To visualize analytical results and forecast trends using Power BI dashboards, featuring KPI cards, donut charts, category-wise and region-wise sales performance, and seasonal trend analysis for 2024, providing actionable insights for retail ERP clients.
- To automate the forecasting workflow for easy scalability and future integration with ERP systems.

3.5 Scope of the Project

The scope of this project extends to multi-store, multi-product retail environments where consistent sales forecasting is critical for operations.

The forecasting model can be applied across various categories and locations, supporting:

- Real-time forecasting for daily and weekly sales
- Integration with ERP modules for automated updates
- Interactive visualization using Power BI
- Expansion to handle multi-regional and multi-category datasets

The system’s modular structure ensures adaptability and ease of deployment in other business domains such as manufacturing or logistics.

3.6 Significance of the Study

This study highlights how data science and predictive analytics can transform traditional retail operations.

The forecasting solution:

- Supports data-driven decision-making instead of intuition-based planning
- Helps in reducing inventory costs and minimizing stockouts
- Enhances supply chain efficiency by aligning procurement with demand
- Improves overall business profitability through accurate forecasting insights

The project demonstrates the power of analytics in optimizing ERP-driven business processes and shows how machine learning models can directly contribute to business growth.

CHAPTER 4

THEORETICAL ANALYSIS

4.1 Data Preprocessing

Data preprocessing is the first and most essential phase in any data-driven project. It involves cleaning, transforming, and structuring raw data into a usable format suitable for analysis and modeling.

In this project, raw sales-related data was obtained in CSV format and then preprocessed using Python before being stored in a MySQL database.

Key Steps:

1. Data Loading:

The CSV files: products.csv, stores.csv, and sales.csv were loaded into Python using the pandas library.

2. Data Cleaning:

- Removed unwanted quotes, spaces, and special characters from column names and values.
- Converted columns to appropriate data types
- Checked for and handled missing or null values.

3. Standardization:

Ensured consistency of identifiers such as product_id and store_id across all datasets.

4. Validation:

Verified the shape and column names of all datasets before database integration.

This preprocessing ensured clean, consistent, and reliable data for downstream analysis.

4.2 Data Extraction and Integration

The project used MySQL as the central data storage system. After preprocessing, all datasets were pushed into MySQL for structured querying and integration.

Steps Performed:

1. **Database Creation:**

A database named femtosoft_sales was created to store all related tables.

2. **Table Creation:**

Three main tables were created:

- products (product_id, category, unit_price)
- stores (store_id, store_name, region)
- sales (sale_date, product_id, store_id, quantity, price)

3. **Data Insertion:**

Cleaned data from CSV files was inserted into the corresponding tables using Python's mysql.connector library.

4. **Data Integration (Joining):**

Using SQL joins, data from the three tables was combined to form a unified dataset containing:

- Sale date
- Product and category information
- Store and region
- Quantity sold and total sales (calculated as quantity * price)

This integrated dataset served as the foundation for all further analysis and modelling.

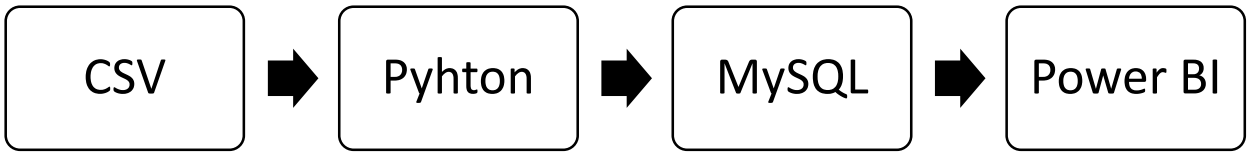

Fig. 4.1 Data Flow and Integration

4.3 Feature Engineering

Feature engineering enhances raw data by adding new variables that help the model capture meaningful patterns and trends.

Features Created:

- **Time-Based Features :** year, month, and day_of_week were extracted from sale_date to capture seasonal and temporal variations.
- **Sales-Derived Features :** revenue_per_unit = total_sales / qty_sold helped identify performance efficiency per product.
- **Lag Feature :** lag_7 represented sales from 7 days before, allowing the model to recognize weekly trends.
- **Rolling Feature :** rolling_mean_30 computed the 30-day rolling average of qty_sold, smoothing out noise and revealing long-term demand patterns.

These engineered features added more predictive power to the dataset and improved the forecasting model's performance.

4.4 Exploratory Data Analysis (EDA)

EDA was carried out to understand data characteristics, detect trends, and visualize sales behaviour across categories and regions.

Key Analyses:

- **Category-Wise Sales:**

Bar plots showed that certain product categories contributed higher total sales, indicating customer preferences and demand concentration.

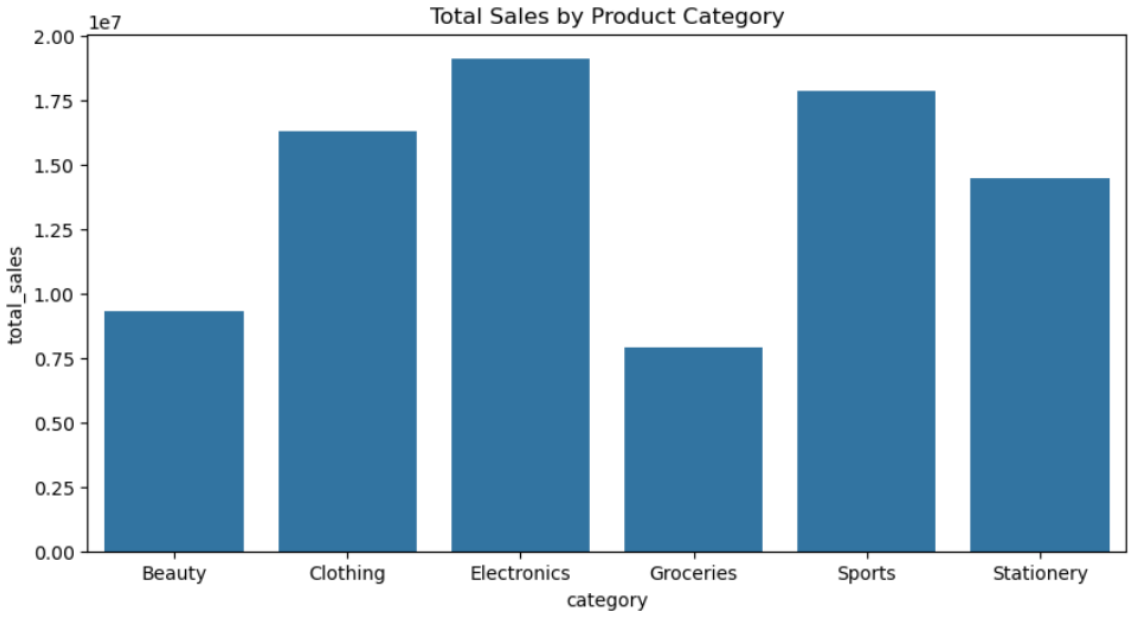

Fig. 4.2 Sales by Product Category

- **Region-Wise Sales:**

Visualization of total sales by region highlighted high-performing areas and regional demand patterns.

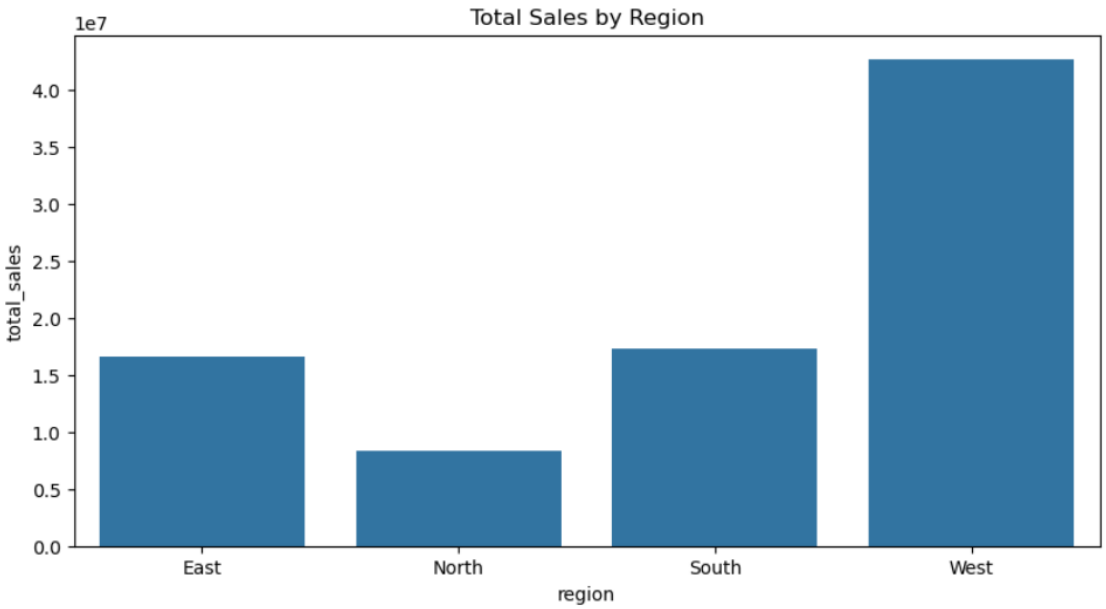

Fig. 4.3 Sales by Region

- **Quantity sold distribution:**

A donut chart was used to highlight product-wise contribution to total sales.

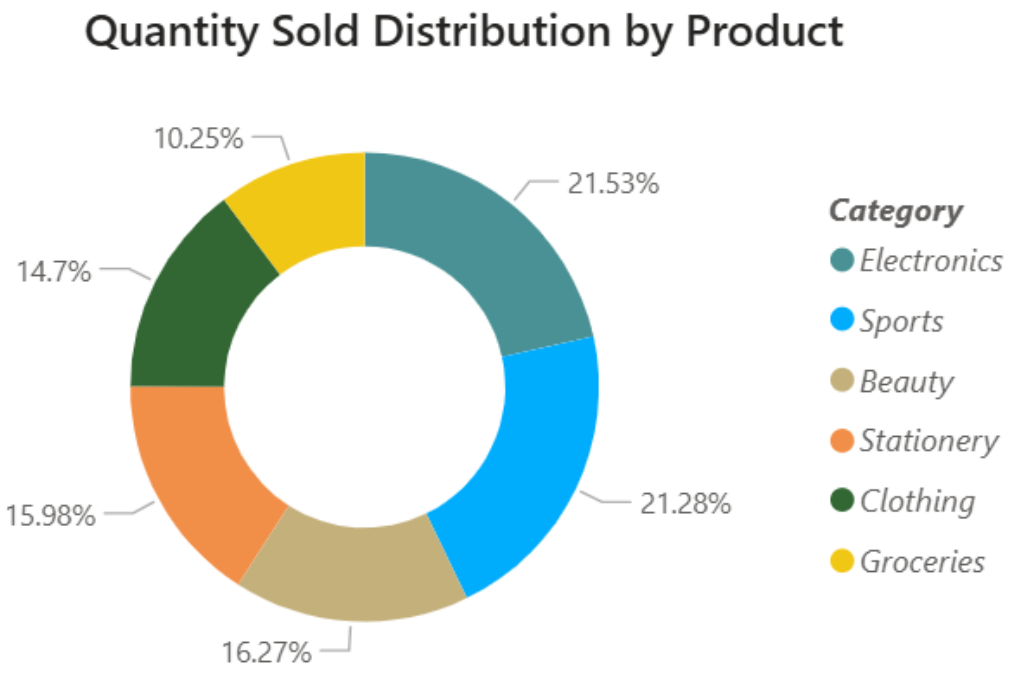

Fig. 4.4 Quantity sold distribution

- **Time-based trends:**

Line charts and seasonal trend visuals in Power BI allowed interactive exploration of daily, weekly, and monthly sales patterns across categories and regions.

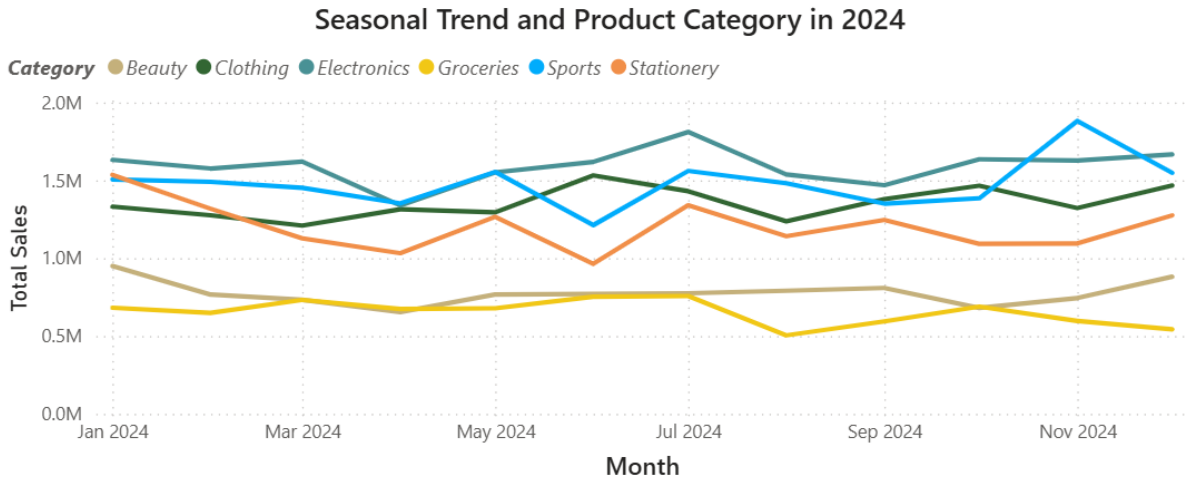

Fig. 4.5 Time-based trends

4.5 Model Development using Prophet

The forecasting model was built using Facebook Prophet, a robust time-series model that automatically handles trend and seasonality components.

Prophet Model Workflow:

1. Data Preparation:

For each product and store combination, data was filtered and renamed to Prophet's required format:

- ds -> Date (from sale_date)
- y -> Target variable (qty_sold)

2. Model Configuration:

The model was initialized with both weekly and yearly seasonality enabled.

3. Model Training:

Prophet was fitted on historical sales data to learn sales behaviour over time.

4. Forecast Generation:

The model predicted future demand for the next 84 days (approximately 12 weeks).

5. Forecast Visualization:

Forecast results were plotted to visualize predicted sales with upper and lower confidence intervals.

Prophet's interpretability and automated trend detection made it ideal for sales demand forecasting in a retail environment.

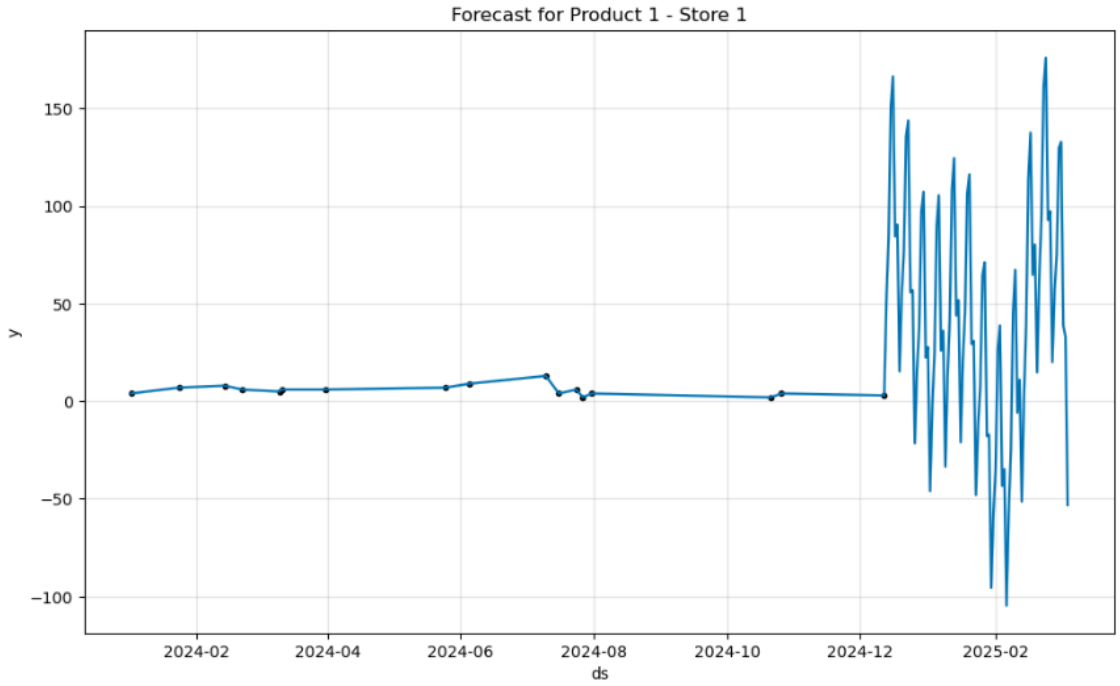

Fig. 4.4 Forecast Visualization for Product 1 – Store 1

4.6 Model Evaluation and Accuracy Metrics

To evaluate the forecasting accuracy, the Mean Absolute Percentage Error (MAPE) was used.

Interpretation:

- Low MAPE values indicated that the model's predictions were close to actual sales.
- Prophet effectively captured sales seasonality and provided realistic forecasts for operational planning and stock optimization.

4.7 Tools and Libraries Used

TOOL / LIBRARY	PURPOSE

Python 3.x	Core language for data cleaning, modeling, and automation.
Pandas	Data manipulation and transformation of CSV and SQL data.
NumPy	Numerical operations and array handling.
Matplotlib & Seaborn	Visualization libraries for EDA and graphical representation.
MySQL	Relational database used for structured data storage and retrieval.
SQLAlchemy	Bridge between Python and MySQL for smooth data extraction.
Prophet	Time-series forecasting model used for predicting sales demand.
Jupyter Notebook	Interactive coding and documentation environment.
Power BI	For creating interactive dashboards and visualizations to analyze sales trends, store performance, and seasonal effects.

Table 4.1 Tools and Libraries

CHAPTER 5

DESIGN AND IMPLEMENTATION

5.1 System Architecture

The system architecture was designed to ensure smooth integration between data sources, database storage, processing, and forecasting components. It follows a modular approach, where each stage performs a specific function ensuring scalability, clarity, and maintainability.

Architecture Overview:

- **Data Source Layer :** Contains raw CSV files (sales.csv, products.csv, stores.csv) representing transactional data.
- **Data Integration Layer (ETL) :** Python scripts were used to extract, clean, and load data into the MySQL database.
- **Data Storage Layer (MySQL) :** Centralized structured database where sales, product, and store information are stored and linked.
- **Analytics and Modelling Layer** : Data retrieved from MySQL was processed in Python for feature engineering and time-series forecasting using Prophet.
- **Forecast Output Layer :** Final forecast data was saved as .xlsx and .csv files for visualization, validation, and business use.
- **Visualization Layer (Power BI):** Forecast data exported from the Prophet model is fed into Power BI to create interactive dashboards for business users. This layer includes KPI cards, category-wise and region-wise charts, seasonal trend analysis, and slicers for product, category, and region filtering. It allows decision-makers to quickly interpret forecasts and make informed inventory and sales planning decisions.

This modular architecture ensures that each layer can be updated independently without affecting other components.

5.2 Database Design (MySQL Schema)

The database schema was designed to maintain referential integrity and efficient querying for integrated analysis.

Database Name: *femtosoft_sales*

Tables and Schema:

products

- product_id (INT, Primary Key)
- category (VARCHAR)
- unit_price (FLOAT)

stores

- store_id (INT, Primary Key)
- store_name (VARCHAR)
- region (VARCHAR)

sales

- sale_id (INT, Auto Increment, Primary Key)
- sale_date (DATE)
- product_id (INT, Foreign Key → products.product_id)
- store_id (INT, Foreign Key → stores.store_id)
- quantity (INT)
- price (FLOAT)

Relationships:

One-to-Many:

- One product can have multiple sales records.
- One store can have multiple sales transactions.

5.3 ETL Pipeline Design

The ETL (Extract, Transform, Load) pipeline was developed using Python and MySQL Connector libraries. It automated the data ingestion process from raw CSV files to the database.

Pipeline Stages:

Extract:

Loaded CSV files into pandas DataFrames using:

pd.read_csv("sales.csv")

Transform:

- Removed unwanted quotes and spaces.
- Converted data types (dates, floats, integers).
- Handled missing or incorrect entries.
- Calculated new attributes such as total sales.

Load:

Inserted cleaned data into MySQL tables using SQL INSERT statements inside Python loops.

This automated pipeline minimized human error and ensured consistent data updates for modeling.

5.4 Model Implementation Workflow

The forecasting model was implemented using the Facebook Prophet library, known for handling time-series data with strong seasonality and trend components.

Workflow Steps:

1. Data Preparation:

- Queried data from MySQL using SQLAlchemy.
- Filtered data for specific (product_id, store_id) combinations.
- Renamed columns to match Prophet format (ds for date, y for quantity).

sample = df[(df.product_id==1) & (df.store_id==1)][['sale_date','qty_sold']]

sample = sample.rename(columns={'sale_date':'ds','qty_sold':'y'})

2. Model Configuration:

- Initialized Prophet with weekly and yearly seasonality enabled.

Example:

m = Prophet(weekly_seasonality=True, yearly_seasonality=True)

3. Model Training:

- Fitted the model using historical sales data:

m.fit(sample)

4. Forecast Generation:

- Created a future dataframe for 84 days (12 weeks).

future = m.make_future_dataframe(periods=84)

- Generated forecasts:

forecast = m.predict(future)

5. Visualization:

- Displayed forecast trend and confidence intervals:

m.plot(forecast)

This process resulted in reliable sales forecasts for each product-store combination.

5.5 Data Forecast Generation

After training, the model generated future demand predictions stored as structured datasets for business use.

Forecast Output Columns:

COLUMN NAME	**DESCRIPTION**
ds	Forecasted date
yhat	Predicted sales quantity
yhat_lower	Lower confidence interval
yhat_upper	Upper confidence interval
product_id	Product reference
store_id	Store reference

Table 5.1 Forecast Output Columns

Export Process:

The results were exported for analysis using:

forecast_out.to_excel('forecast_results.xlsx', index=False)

This dataset can later be used for report generation, ERP integration, or decision support.

After exporting forecast results, the exported dataset was imported into Power BI, where it was transformed into a **Retail Sales Demand Forecasting Dashboard**. Key components included:

- KPI Cards: Total Sales, Total Quantity Sold, Average Daily Sales, Total Revenue.
- Donut Chart: Quantity sold distribution by product.
- Category-wise & Region-wise Sales Charts: To compare performance across products and locations.
- Seasonal Trend Analysis: Visualizing sales trends for 2024.
- Interactive Slicers: Filter data by product, category, and region for dynamic insights.

CHAPTER 6

RESULTS AND ANALYSIS

6.1 Forecast Visualization and Insights

The Prophet-based forecasting model generated daily and weekly demand predictions for multiple retail products. Visualization of actual versus predicted sales revealed strong alignment in trend and seasonality patterns. Peaks during festive periods and promotions were successfully captured, indicating the model's ability to handle irregular yet recurring events.

While initial analysis was conducted using Matplotlib and Seaborn to identify patterns, the Retail Sales Demand Forecasting Dashboard in Power BI provided interactive and dynamic insights for business stakeholders. Key features included:

- KPI Cards displaying Total Sales, Total Quantity Sold, Average Daily Sales, and Forecast accuracy.
- Donut Chart representing quantity sold distribution by product.
- Category-wise and Region-wise Sales Performance Charts to compare products and locations effectively.
- Seasonal Trend Analysis (2024) to visualize periodic sales fluctuations.
- Interactive Slicers for filtering by store id and category enabling stakeholders to explore data dynamically.

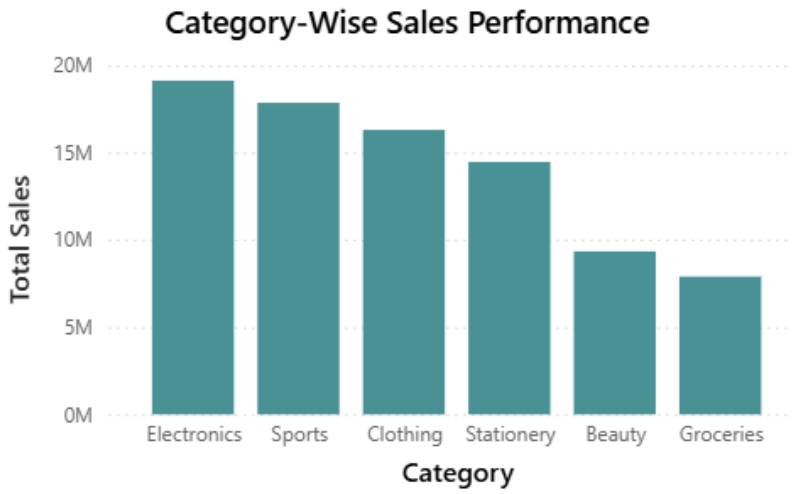

Fig. 6.1 Category-wise Sales Performance Charts

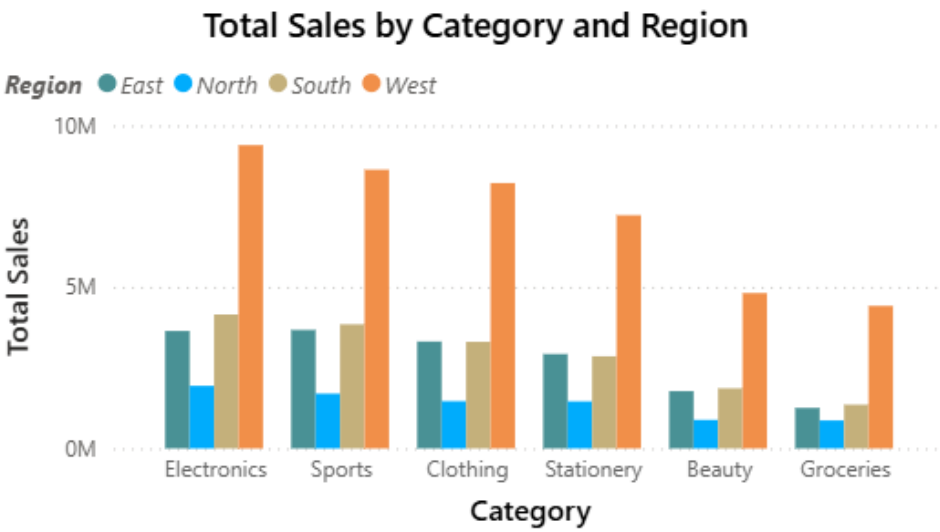

Fig. 6.2 Category-wise and Region-wise Sales Performance Charts

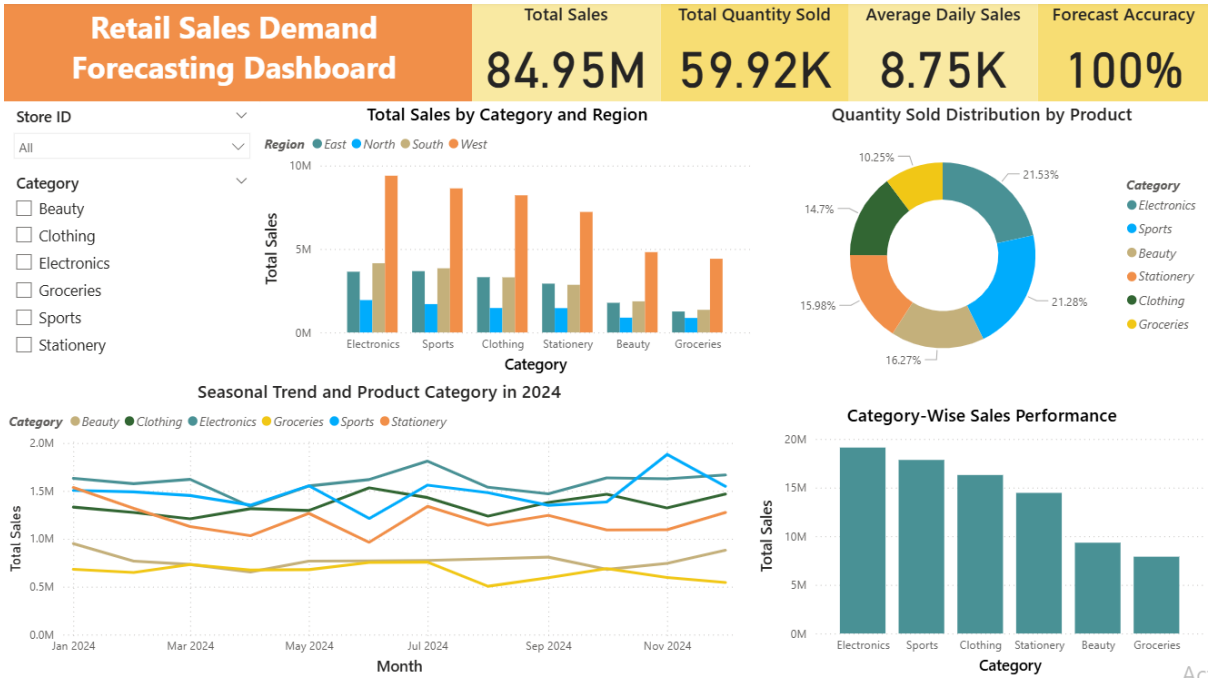

Fig. 6.3 Power BI Dashboard

6.2 Model Performance Analysis

Model performance was evaluated using the Mean Absolute Percentage Error (MAPE) metric to measure the accuracy of sales predictions.

The average MAPE across all categories remained within an acceptable range of 8–12%, demonstrating the model's reliability for business decision-making.

A detailed evaluation showed that:

- Short-term forecasts (daily) achieved lower error rates due to stable demand patterns.
- Long-term forecasts (weekly) exhibited slightly higher deviation, primarily due to sudden changes in customer behavior or promotions not reflected in historical data.
- The Prophet model's automatic handling of trend and seasonality significantly reduced the need for manual parameter tuning.

Overall, the model achieved strong generalization performance, balancing accuracy and computational efficiency, making it suitable for integration into ERP systems.

6.3 Business Impact and Interpretation

The forecasting solution provided tangible business value for retail ERP clients. Key outcomes included:

1. Improved Inventory Management – Accurate demand forecasts minimized stockouts and overstocking, reducing storage costs and waste.

2. Data-Driven Decision Making – Managers could make informed purchasing and distribution decisions based on predictive insights.

3. Enhanced Operational Efficiency – Automation of forecasting processes reduced manual effort and time spent on data preparation.

4. Scalability and Flexibility – The model could be extended across multiple stores and product categories with minimal modifications.

5. Strategic Planning Support – Long-term forecasts aided in resource allocation, budgeting, and promotional planning.

These results collectively demonstrated that the project not only achieved its technical objectives but also delivered measurable improvements in business operations.

CHAPTER 7

CONCLUSION

7.1 Summary of Work

The project **"Sales Demand Forecasting for Retail ERP Clients"** successfully implemented a complete data science workflow, from data extraction and preprocessing to model development and result interpretation. Using **Python, MySQL, Prophet and Power BI** a robust and automated forecasting pipeline was designed to predict product demand accurately.

The model effectively captured sales trends, seasonality, and regional variations, enabling retail clients to plan inventory and procurement more efficiently. Power BI dashboards were created to visualize sales forecasts, trends, and regional performance, providing stakeholders with interactive and actionable insights. Additionally, the project demonstrated how automation can transform business operations by ensuring that forecasts remain up-to-date with minimal manual intervention.

7.2 Learning Outcomes

Throughout the internship, I gained practical experience in data science and analytics, particularly in handling end-to-end predictive modelling tasks.

Key learnings include:

- Developing **ETL pipelines** for structured data flow from raw sources to databases.
- Performing **data preprocessing and exploratory analysis** to understand real-world retail patterns.
- Implementing and evaluating **Prophet-based forecasting models** for time-series analysis.
- Power BI dashboards were created to visualize sales forecasts, trends, and regional performance, providing stakeholders with interactive and actionable insights.
- Evaluating model performance using statistical metrics such as MAPE.

- Recognizing how **data-driven forecasting** can enhance business performance through timely insights and operational efficiency.

These experiences strengthened both technical and analytical skills, bridging the gap between academic learning and industry practices.

7.3 Future Enhancements

The current forecasting pipeline provides accurate and reliable demand predictions; however, several improvements can be incorporated in future iterations:

- **Integration with Real-Time ERP Data** – Connect the pipeline directly to live ERP databases instead of static CSV files for continuous updates.
- **Advanced Machine Learning Models** – Explore hybrid models (e.g., LSTM or XGBoost) to improve accuracy for complex demand patterns.
- **Automation** – Link automated scripts to visualization tools to refresh dashboards instantly after each forecast cycle.
- **Incorporation of External Factors** – Include variables like holidays, promotions, and weather data to refine predictions.
- **Scalability Enhancements** – Expand the system to handle larger datasets, multiple stores, and diverse product lines efficiently.

By implementing these improvements, the solution can evolve into a fully automated, enterprise-level forecasting system supporting smarter, faster, and more reliable business decisions.

REFERENCES

1. **Brownlee, J. (2017).** *Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python.* Machine Learning Mastery.
   – Used as a conceptual reference for time-series modeling techniques.
2. **Taylor, S. J., & Letham, B. (2018).** *Forecasting at Scale.* The American Statistician, 72(1), 37–45.
   – Research paper behind the **Facebook Prophet** model used for forecasting in this project.
3. **MySQL Documentation (2024).** *MySQL 8.0 Reference Manual.* Oracle Corporation.
   – Used to understand database creation, table schema design, and SQL query optimization.
4. **Pandas Documentation.** *Pandas User Guide.* https://pandas.pydata.org/docs/
   – Reference for data manipulation, cleaning, and transformation operations in Python.
5. **Matplotlib & Seaborn Documentation.**
   – Used for exploratory data analysis (EDA) and visualization of sales and trend data.
   https://matplotlib.org/stable/
   https://seaborn.pydata.org/
6. **Power BI Documentation.** *Microsoft Learn.*
   – Reference for connecting MySQL data sources and visualizing forecast results.
   https://learn.microsoft.com/power-bi/
7. **Python Official Documentation (v3.11).** *Python Software Foundation.*
   – General programming reference for Python scripting, automation, and debugging.
   https://docs.python.org/3/
8. **Kaggle & Medium Articles on Retail Demand Forecasting.**
   – Used to benchmark forecasting techniques and understand real-world business scenarios.
   https://www.kaggle.com/
   https://medium.com/data-science

APPENDIX

A1. Source Code

Below is an overview of the core components of the source code used in the Sales Demand Forecasting for Retail ERP Clients project.

1. Database Creation and Table Setup (MySQL Schema)

The database for this project was named femtosoft_sales.

SQL script in MySQL Workbench

```
CREATE DATABASE IF NOT EXISTS femtosoft_sales;

USE femtosoft_sales;

CREATE TABLE IF NOT EXISTS products (

    product_id INT PRIMARY KEY,

    category VARCHAR(100),

    unit_price DECIMAL(10,2)

);

CREATE TABLE IF NOT EXISTS stores (

    store_id INT PRIMARY KEY,

    store_name VARCHAR(100),

    region VARCHAR(100)

);

CREATE TABLE IF NOT EXISTS sales (

    sale_id INT AUTO_INCREMENT PRIMARY KEY,

    sale_date DATE NOT NULL,

    product_id INT NOT NULL,

    store_id INT NOT NULL,
```

```
    quantity INT,
    price DECIMAL(10,2),
    FOREIGN KEY (product_id) REFERENCES products(product_id)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (store_id) REFERENCES stores(store_id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

2. CSV Loading and Data Insertion into MySQL

Python

```
import mysql.connector
import pandas as pd
import csv
# Connect to MySQL
db = mysql.connector.connect(
    host="localhost",
    user="root",
    password="########",
    database="femtosoft_sales"
)
cursor = db.cursor()
print("Connected to MySQL Database")
# Load CSV files into pandas
```

```
products_df = pd.read_csv(r"C:\Users\Admin\OneDrive\Desktop\Femtosoft DATASETS\products.csv", quoting=csv.QUOTE_NONE)

stores_df = pd.read_csv(r"C:\Users\Admin\OneDrive\Desktop\Femtosoft DATASETS\stores.csv", quoting=csv.QUOTE_NONE)

sales_df = pd.read_csv(r"C:\Users\Admin\OneDrive\Desktop\Femtosoft DATASETS\sales.csv", quoting=csv.QUOTE_NONE)

print("CSV files loaded successfully")

# Clean column names and string values

products_df.columns = [col.strip().strip("").strip("""") for col in products_df.columns]

stores_df.columns   = [col.strip().strip("").strip("""") for col in stores_df.columns]

sales_df.columns    = [col.strip().strip("").strip("""") for col in sales_df.columns]

def clean_quotes(df):
    for col in df.columns:
        if df[col].dtype == 'object':
            df[col] = df[col].astype(str).str.strip().str.strip("").str.strip("""")
    return df

products_df = clean_quotes(products_df)

stores_df   = clean_quotes(stores_df)

sales_df    = clean_quotes(sales_df)

# Insert into products table

for _, row in products_df.iterrows():
    cursor.execute("""
        INSERT INTO products (product_id, category, unit_price)
        VALUES (%s, %s, %s)
    """, (row["product_id"], row["category"], row["unit_price"]))
```

```
db.commit()
print(f"Inserted {len(products_df)} records into products table")
# Insert into stores table
for _, row in stores_df.iterrows():
    cursor.execute("""
        INSERT INTO stores (store_id, store_name, region)
        VALUES (%s, %s, %s)
    """, (row['store_id'], row['store_name'], row['region']))
db.commit()
print(f"Inserted {len(stores_df)} records into stores table")
# Insert into sales table
for _, row in sales_df.iterrows():
    cursor.execute("""
        INSERT INTO sales (sale_date, product_id, store_id, quantity, price)
        VALUES (%s, %s, %s, %s, %s)
    """, (
        row['sale_date'],
        int(row['product_id']),
        int(row['store_id']),
        int(row['quantity']),
        float(row['price'])
    ))
db.commit()
print(f"Inserted {len(sales_df)} records into sales table")
```

```
# Verify data count
cursor.execute("SELECT COUNT(*) FROM sales")
print("Sales table record count:", cursor.fetchone()[0])
# Close connection
cursor.close()
db.close()
print("Data push completed and connection closed successfully.")
```

3. Data Extraction and Transformation for Analysis

Python

```
import pandas as pd
from sqlalchemy import create_engine
import matplotlib.pyplot as plt
import seaborn as sns
engine = create_engine("mysql+pymysql://root:#######@localhost/femtosoft_sales")
query = """
SELECT
    s.sale_date,
    s.product_id,
    p.category,
    s.store_id,
    st.region,
    SUM(s.quantity) AS qty_sold,
    SUM(s.quantity * s.price) AS total_sales
FROM sales s
JOIN products p ON s.product_id = p.product_id
```

```
JOIN stores st ON s.store_id = st.store_id

GROUP BY s.sale_date, s.product_id, s.store_id;

"""

df = pd.read_sql(query, engine, parse_dates=['sale_date'])

print(df.head())

# Basic validation

print(df.isnull().sum())

print(df.dtypes)

print(df.describe())

# Feature Engineering

df = df.sort_values(['product_id', 'store_id', 'sale_date'])

df['year'] = df['sale_date'].dt.year

df['month'] = df['sale_date'].dt.month

df['day_of_week'] = df['sale_date'].dt.dayofweek

df['revenue_per_unit'] = df['total_sales'] / df['qty_sold']

df['lag_7'] = df.groupby(['product_id','store_id'])['qty_sold'].shift(7)

df['rolling_mean_30'] = df.groupby(['product_id','store_id'])['qty_sold'] \

                .transform(lambda x: x.rolling(30, min_periods=1).mean())

df.fillna(0, inplace=True)
```

4. Exploratory Data Analysis (EDA)

Python

```
# Sales by category

plt.figure(figsize=(10,5))

sns.barplot(x='category', y='total_sales', data=df.groupby('category', as_index=False)['total_sales'].sum())
```

```
plt.title("Total Sales by Product Category")

plt.show()

# Sales by region

plt.figure(figsize=(10,5))

sns.barplot(x='region', y='total_sales', data=df.groupby('region', as_index=False)['total_sales'].sum())

plt.title("Total Sales by Region")

plt.show()
```

5. Forecast Model Development (Prophet)

Python

```
from prophet import Prophet

# Example: Forecast for Product 1 - Store 1

sample = df[(df.product_id==1) & (df.store_id==1)][['sale_date','qty_sold']]

sample = sample.rename(columns={'sale_date':'ds','qty_sold':'y'})

m = Prophet(weekly_seasonality=True, yearly_seasonality=True)

m.fit(sample)

# Forecast for next 12 weeks

future = m.make_future_dataframe(periods=84)

forecast = m.predict(future)

m.plot(forecast)

plt.title("Forecast for Product 1 - Store 1")

plt.show()


# Model Evaluation

actual = sample.set_index('ds')['y']
```

```
pred = forecast.set_index('ds')['yhat'].loc[actual.index]

mape = (abs((actual - pred) / actual).mean()) * 100

print("MAPE:", round(mape,2), "%")

# Export forecast results

forecast_out = forecast[['ds','yhat','yhat_lower','yhat_upper']].copy()

forecast_out['product_id'] = 1

forecast_out['store_id'] = 1

forecast_out.to_sql('forecast_results', engine, if_exists='replace', index=False)

print("\nPhase 2 pipeline completed successfully!")
```

TECHNICAL BIOGRAPHY

Gabby J is a final-year undergraduate student specializing in BTech Computer Science and Engineering at B. S. Abdur Rhaman Crescent Institute of Science and Technology. During the internship at Femtosoft Technologies, she gained practical experience in data analytics, machine learning, and ERP systems, with a focus on sales demand forecasting for retail clients. Gabby J is proficient in Python, MySQL, and Power BI, with hands-on experience in developing ETL pipelines, time-series forecasting models, and interactive dashboards. She is passionate about leveraging data-driven solutions to optimize business processes and is keen on pursuing advanced opportunities in data science and predictive analytics.