# Project Documentation: Sales Forecasting and Demand Prediction

## Table of Contents



**GitHub Repo**

## 1. Team Members

- Ahmed Ben Bella Ahmed Abd El-Nasser
- Ahmed Nassar El-Shabrawy Erfan
- Mahmoud Mohamed Saleh Mahmoud
- Mariam Hamdi El-Sayed
- Mohamed Naguib El-Sayed Naguib
- Omar Abd El-Salam Mohamed Hefny

## 2. Project Overview

This project aims to develop a sales forecasting and demand prediction system. The goal is to accurately predict unit sales per item type as well as total revenue and units sold for the whole data across all item types to assist in strategic business decisions such as inventory management, demand planning, and marketing efforts.

## 3. Team Background and Project Genesis

We are a team of six students. At the start of the project, we had minimal research and almost no concrete foundation. Initial brainstorming sessions led us to explore various dataset sources, including the idea of web scraping, before settling on using existing datasets.

Our first dataset seemed promising, offering decent features and structure. However, upon performing exploratory data analysis (EDA), we found the dataset small and unfit for building accurate predictive models. Attempts to proceed with modeling confirmed our concerns, as we couldn’t produce satisfactory results.

## 4. Dataset Selection and Preprocessing

After extensive searching, we discovered a clean and well-structured dataset containing around 1 million records and meaningful column:

- Order Date
- Ship Date
- Order ID
- Item Type
- Region
- Country
- Sales Channel (Online/Offline)
- Order Priority
- Unit Cost
- Unit Price
- Units Sold
- Total Cost
- Total Revenue
- Total Profit

Initial targets included predicting **Total Revenue** and **Units Sold** across the dataset. EDA, preprocessing, and feature engineering were performed.

We started modeling for sales forecasting (Total Revenue), and demand prediction (Units Sold. We got a decent $R^2$ score around 0.64, while the Units Sold was challenging and was of poor accuracy.

Our instructor suggested shifting focus to **item-wise** predictions. Specifically to predict a metric that includes both sales and demand which is Total Revenue/Units Sold. As for our dataset, that turned out to be the value of the **Unit Cost** column.

## 5. Target Variable Challenges

Item-wise predictions presented another challenge: the **Unit Cost** column (initially proposed as a target) was not meaningful to work on and predict. We instead chose to predict **Units Sold per item type**.

With proper feature engineering and the trial of multiple models we got accuracies ranging from 0.45 to 0.8 across different item types, which is good compared to their data sizes.

Also, we tried again with our main target variables (Total Revenue and Units Sold across the all items), and after implementing the same right steps of modeling we applied to the item-wise models, we got promising $R^2$ scores; **0.93** for **Total Revenue**, and **0.74** for **Units Sold**.

## 6. Model Development

Models tried:

- Linear Regression
- Random Forest
- XGBoost
- LightGBM (Light Gradient Boosting Machine)
- Gradient Boosting

XGBoost outperformed others consistently and was used for final modeling. Each item type had its own XGBoost model trained on 70K-80K records, resulting in $R^2$ scores ranging from 0.45 to 0.8. The full dataset models (for Total Revenue and Units Sold) had $R^2$ scores of 0.93 and 0.74 respectively.

## 7. Visualizations

We started with **Matplotlib** and **Seaborn**, then transitioned to **Plotly** to produce interactive, clean, and deployment-ready visuals. Visuals included insights about regional performance, sales channels, item types, and monthly trends.

## 8. Deployment

Initially, we attempted deploying via **Flask**, but persistent issues and setup complications led us to switch to **Streamlit**, which offered a faster and smoother experience.

The deployed app includes:

- Visualizations with user controls (show/hide, filter, etc.)
- Integration of XGBoost models via pickle files
- Interactive dashboard for business users

## 9. Business Impact

The model can assist businesses in:

- Anticipating demand per product to avoid overstock/understock
- Optimizing inventory and warehouse logistics
- Tailoring marketing campaigns per region/item type
- Planning staffing around seasonal demand trends

## 10. Challenges and Key Decisions

- Selecting a realistic and usable dataset
- Shifting from whole-dataset to item-wise modeling
- Choosing XGBoost for its balance between accuracy and interpretability
- Abandoning Flask for Streamlit to save time and reduce complexity
- Enhancing visuals with Plotly for better UX

## 11. Conclusion

This project was an insightful journey through real-world machine learning workflow: from initial data struggles to final deployment. It helped our team apply analytical thinking, deal with limitations, and build a practical solution with genuine business utility.

Despite being early in our ML journey, the process helped us solidify key concepts and sharpen our problem-solving skills while delivering a functional and impactful product.