Team members:

Rahma Anwar  
 Farida Emad  
 Malak Safwan  
 Ahmed Mahoud  
 Yousef Ayman

**Supervisor: Eng. Ahmed Azab**

[*Project GitHub*](https://github.com/rahmaanwer22/DEPI_Project)

Sales Forecasting and Demand Prediction

*Initiative: Digital Egypt Pioneers – Round 2*

Table of Contents

[Introduction 2](#_Toc197817022)

[Project Objective 3](#_Toc197817023)

[Dataset Description 3](#_Toc197817024)

[Data Cleaning 5](#_Toc197817025)

[**Key Cleaning Steps:** 5](#_Toc197817026)

[Exploratory Data Analysis (EDA) 5](#_Toc197817027)

[Visuals for EDA 11](#_Toc197817028)

[**1. Customer Segmentation** 11](#_Toc197817029)

[Advanced Data Analysis Time Series analysis 12](#_Toc197817030)

[Predictive Modeling 13](#_Toc197817031)

[**1. Time Series Decomposition** 13](#_Toc197817032)

[**2. Stationarity Check (ADF Test)** 13](#_Toc197817033)

[**3. Feature Scaling** 13](#_Toc197817034)

[**4. Model Selection & Forecasting** 14](#_Toc197817035)

[Model Evaluation 14](#_Toc197817036)

[MLOPs Implementation & Deployment 15](#_Toc197817037)

[Results & Business Recommendations 17](#_Toc197817038)

[**Key Results:** 17](#_Toc197817039)

[Business Recommendations 17](#_Toc197817040)

[**For Retail Management:** 17](#_Toc197817041)

[**For Marketing Teams:** 17](#_Toc197817042)

[**For Financial Planning:** 17](#_Toc197817043)

[**For Business Expansion:** 18](#_Toc197817044)

[Conclusion 18](#_Toc197817045)

[GitHub Repository 18](#_Toc197817046)

[Resources 18](#_Toc197817047)

***Abstract***

This project focuses on predicting sales and understanding demand fluctuations in a retail environment using historical data from a global superstore. By applying data science techniques such as data preprocessing, exploratory data analysis (EDA), feature engineering, and time series forecasting, we aim to uncover key business insights. The project helps companies improve inventory planning, optimize promotional strategies, and make informed business decisions. Future deployment of this project could offer a scalable solution for businesses in retail and supply chain sectors.

# Introduction

In today’s competitive retail landscape, businesses need more than just intuition to make strategic decisions—they need data-driven insights. This project, titled "Sales Forecasting and Demand Prediction," was chosen due to its real-world relevance and direct impact on business efficiency, profitability, and customer satisfaction.

We selected this project because predicting future sales and understanding demand patterns is critical for any company that deals with inventory, supply chain management, or customer demand. With the rise of e-commerce and global logistics, companies need smart systems that help them answer questions like:

* What products should we stock more of next month?
* Will a discount increase sales, or just lower profits?
* How will demand change during holidays or seasons?

**Who Will Benefit From This Project?**

* Retailers and e-commerce platforms like Amazon, Walmart, Noon
* Supply chain managers and warehouse planners
* Marketing teams who plan promotions and discounts
* Financial analysts for revenue forecasting

**Business Value and Impact** If applied in a real-world business environment, the project can:

* Optimize inventory levels (reducing holding costs)
* Prevent out-of-stock scenarios
* Improve planning for promotions, holidays, and peak seasons
* Support targeted marketing based on demand trends
* Increase customer satisfaction by ensuring product availability

# Project Objective

• Analyze historical sales data to identify key trends and patterns.

• Build predictive models to forecast future sales.

• Investigate the impact of discounts and promotions on profit and demand.

# Dataset Description

For this project, we chose the Global Superstore dataset **(global\_superstore\_2016.csv.xlsx)** because it offers a rich and realistic view of retail business operations across different regions. The dataset simulates the kind of data available to real-world companies, making it ideal for building a meaningful and applicable forecasting system.

**Why This Dataset?**

* Contains realistic business variables: sales, profit, discount, quantity.
* Captures temporal patterns over time through order dates.
* Covers multiple segments and product categories for detailed analysis.
* Has a large volume of data suitable for time series forecasting.
* Includes data from different countries and customers, making it generalizable.

**It contains various features, including:**

• Sales, Quantity, Profit, Discount

• Order Date, Ship Mode, Segment, Category

• Country, Customer ID, Product Name, and others.

**Key Columns Explained**

Here’s a breakdown of the most relevant columns used in the project:

| **Column Name** | **Description** |
| --- | --- |
| **Order Date** | The date an order was placed – used for time series analysis. |
| **Sales** | Total sales amount for the order – main variable for forecasting. |
| **Profit** | Profit made from the sale – used to analyze promotion and discount impact. |
| **Quantity** | Number of units sold – used to track demand. |
| **Discount** | Discount applied to the product – to study its effect on demand/profit. |
| **Ship Mode** | The method of shipping (e.g., First Class, Standard Class) – optional analysis. |
| **Segment** | Customer segment: Consumer, Corporate, or Home Office – helps with segmentation analysis. |
| **Category** | Product category: Furniture, Office Supplies, Technology – useful for trend analysis. |
| **Sub-Category** | More detailed product type (e.g., Chairs, Phones) – for deeper analysis. |
| **Customer ID** | Unique ID per customer – for identifying frequent buyers. |
| **Country** | Country of the order – helps with regional analysis. |
| **Product Name** | Specific product sold – used in top-product analysis. |

We also engineered new features to make the data even more useful:

* **Year, Month, Day**: Extracted from Order Date to detect seasonal trends.
* **IsWeekend**: Indicates whether an order was made on a weekend.
* **PromotionFlag / DiscountCategory**: Custom features to categorize promotions.

# Data Cleaning

Before starting the analysis, the dataset required several cleaning steps to ensure accuracy and reliability. These steps were crucial to avoid misleading results and to allow the predictive models to perform well.

**Key Cleaning Steps:**

* **Handling Missing Values**: Detected and filled or removed missing entries in key fields.
* **Removing Duplicates**: Identified and dropped repeated records to ensure each order was counted once.
* **Outlier Treatment**: Adjusted or removed data points that were extremely outside the expected range (e.g., very high discounts or negative profits).
* **Date Conversion**: Transformed date columns into proper datetime format to extract year, month, and day.
* **Feature Engineering**:
  + Added IsWeekend column to mark if the order was made on a weekend.
  + Created PromotionFlag for rows with high discounts.
  + Categorized discounts into groups: low, medium, high.

# Exploratory Data Analysis (EDA)

**Objective**

To understand sales behavior, customer trends, discount effectiveness, and seasonal impacts—ultimately supporting better forecasting and business decisions.

**Dataset Overview**

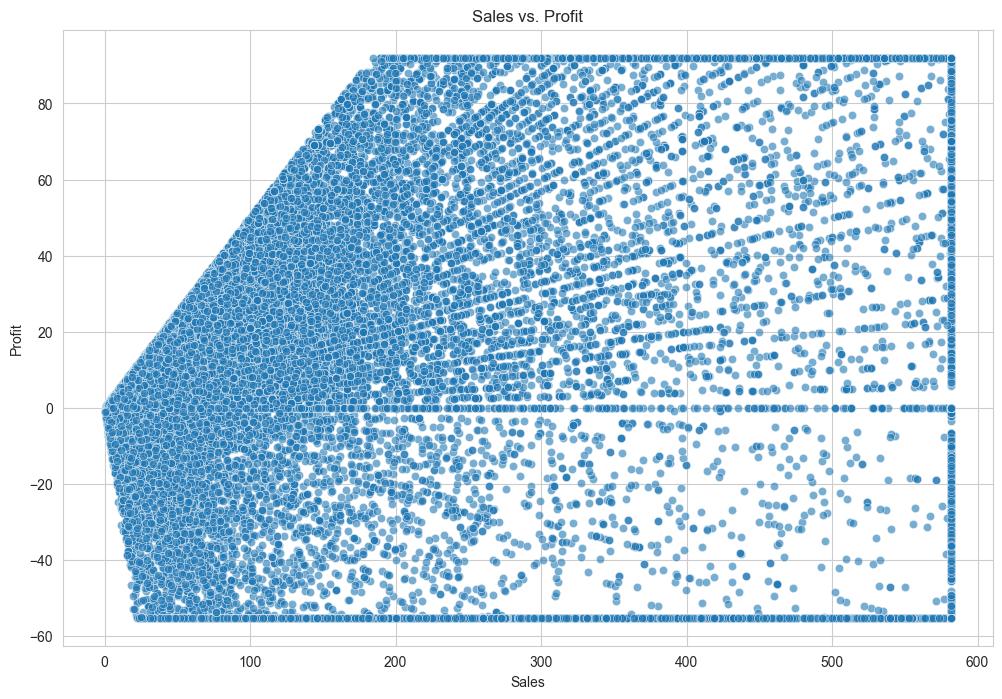
* **Records**: 51,290
* **Columns**: 30
* **Includes**:
  + Sales, Profit, Discount, Quantity
  + Customer demographics
  + Product and category details
  + Shipping info
  + Date/time (order date, day of week, etc.)
  + Engineered features: IsWeekend, PromotionFlag, DiscountCategory, Year, Month, Day

**What is EDA?**

EDA is the process of summarizing and visualizing key aspects of a dataset. It reveals patterns, trends, anomalies, and relationships between variables, guiding deeper analysis and modeling decisions. It ensures data quality and helps in feature engineering and model selection.

**Key Insights from Visual and Statistical Analysis**

**1. Sales vs. Profit (Scatter Plot)**

* Positive correlation overall.
* Some high-sales entries show **low or negative profit** → due to deep discounting or high shipping costs.

**2. Total Sales by Category (Bar Chart)**

* A graph showing a bar chart

  AI-generated content may be incorrect.Certain product categories (e.g., **Technology**, **Office Supplies**) drive most revenue.
* These should be **prioritized** in marketing and inventory.

A graph of blue dots

AI-generated content may be incorrect.**3. Discount vs. Profit (Scatter Plot)**

* Clear **negative correlation**.
* Higher discounts = reduced profit → optimize discount levels to balance demand vs. margin.

**4. Sales by Season (Box Plot)**

* A chart with blue squares

  AI-generated content may be incorrect.Strong seasonal effect: Peaks in **Q4 (October–December)**.
* Marketing and inventory should **focus on seasonal patterns**.

**5. Sales by Ship Mode (Bar Chart)**

* Some shipping modes contribute more to sales.
* A graph with blue squares

  AI-generated content may be incorrect.Businesses can **incentivize cost-effective shipping** with promotions.

A graph with blue lines

AI-generated content may be incorrect.**6. Quantity vs. Sales (Scatter Plot)**

* Bulk purchases lead to higher total sales.
* Suggests potential for **bulk pricing or wholesale promotions**.

A graph of blue bars with black text

AI-generated content may be incorrect.**7. Region-wise Profit (Bar Chart)**

* Profitability varies by region.
* Use this for **targeted regional strategies**.

A graph showing a diagram of shipping cost

AI-generated content may be incorrect.

**8. Shipping Cost vs. Sales (Scatter Plot)**

* No strong correlation.
* Evaluate if high shipping costs are limiting demand.

A graph of sales by discount and profit bins

AI-generated content may be incorrect.**9. Heatmap: Discount and Profit Bins vs. Sales**

* Helps identify where high discounts result in **low or high sales**.
* Useful to **avoid over-discounting** with no sales benefit.

A graph with blue lines

AI-generated content may be incorrect.

**10. Order Size Frequency (Histogram)**

* Most orders are for **2–3 items**.
* Use this to tailor promotions, forecast demand, and optimize stock.

**Visual Summary**

You can insert visual outputs here from:

* Seaborn, Matplotlib, Plotly, or Excel charts  
  (e.g., Sales by Month, Profit by Category, Correlation Heatmap)

| **Insight** | **Business Action** |
| --- | --- |
| High discounts lower profit | Optimize discount strategy |
| Peak sales in Q4 | Boost inventory and marketing |
| Bulk buys drive revenue | Offer tiered pricing |
| Regional profit varies | Adjust strategy per region |
| Customer segments differ | Use targeted promotions |

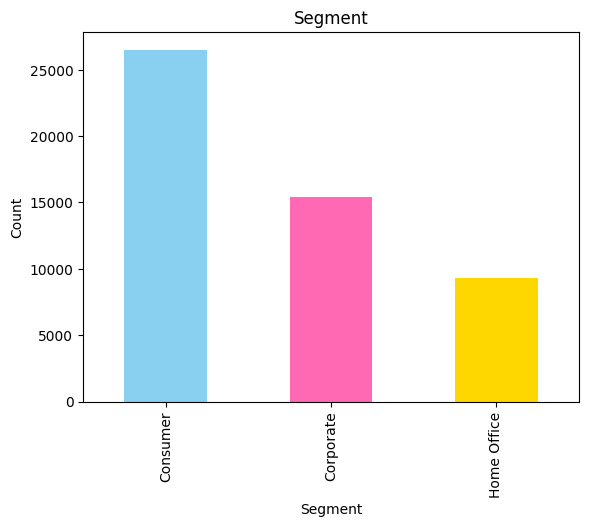
**Recommendations Based on EDA**

1. **Optimize Discount Strategies**
   * Use **tiered discounting** based on product margin.
   * Focus on **ROI-driven** promotion levels.
2. **Reassess Shipping Policies**
   * Consider **discounted or free shipping** for high-value orders.
   * Prioritize efficient shipping methods.
3. **Target High-Profit Products**
   * Promote items with high margin and consistent demand.
4. **Seasonal Focus**
   * Allocate marketing budget toward peak periods.
   * Forecast stock for **holidays and end-of-year spikes**.
5. **Segment Customers Smartly**
   * Focus on **consumer and corporate** segments.
   * Use **purchase history** for personalized offers.
6. **Inventory Planning**
   * Use **order size patterns** to avoid overstocking or understocking.

# Visuals for EDA

The goal of Exploratory Data Analysis is to explore trends, patterns, and relationships within the dataset before applying predictive models. This step helps us understand customer behavior, identify sales drivers, and detect seasonal effects.

**1. Customer Segmentation**

We analyzed how orders are distributed across different customer segments:

* **Consumer**
* **Corporate**
* **Home Office**

**Insight: The *Consumer* segment had the highest number of orders, but *Corporate* showed higher average profit per order**

**2. Top Products and Sub-Categories**

A graph of different colored lines

AI-generated content may be incorrect.We explored which products and sub-categories had the most sales.

**Insight: Sub-categories like *Phones*, *Chairs*, and *Binders* were among the top-selling items.**

**3. Weekend Sales Behavior**

Using the IsWeekend feature, we compared sales on weekends vs. weekdays.

A blue and pink pie chart

AI-generated content may be incorrect. **Insight: Weekday sales were generally higher,**

**but specific promotions on weekends caused short spikes.**

# Advanced Data Analysis Time Series analysis

* **Trend: Upward growth**
* **Seasonality: Holiday-linked peaks**
* **Residuals: Random noise**

A screenshot of a graph

AI-generated content may be incorrect.**Stationarity Check (ADF Test)**

* **ADF Statistic: -2.923**
* **p-value: 0.043 → Weakly stationary → differencing needed**

**Correlation Analysis**

* **Sales & Promotion Flag: +0.029**
* **Sales & Discounts: -0.11**
* **Sales & Weekends: +0.008**

**Insights:**

* **Promotions have minimal impact.**
* **Discounts may hurt profit more than help sales.**
* **Holidays, products, and segments matter more.**

# Predictive Modeling

To forecast future sales and understand demand behavior over time, we applied time series analysis techniques. This allowed us to capture seasonality, trends, and fluctuations in sales.

**1. Time Series Decomposition**

We decomposed the sales data to identify three main components:

* **Trend**: Long-term upward or downward movement.
* **Seasonality**: Regular repeating patterns (monthly, yearly).
* **Residuals**: Noise or random variation.

Insight: There is a clear end-of-year seasonal peak in most years.

**2. Stationarity Check (ADF Test)**

Before applying forecasting models, we tested if the time series was stationary using the **Augmented Dickey-Fuller (ADF)** test.

Result: The original data was not stationary, so differencing was applied to make it stationary.

**3. Feature Scaling**

We applied:

* **StandardScaler** for normally distributed features.
* **MinMaxScaler** for features with varying ranges (e.g., Discount, Profit).

This helped ensure that model performance wasn't skewed by outlier values or scale differences.

**4. Model Selection & Forecasting**

We implemented and compared multiple forecasting models:

**A. ARIMA Model**

Used for univariate forecasting of monthly sales.

Output: Short-term predictions with trend-following behavior.

**B. Prophet Model (by Facebook)**

Ideal for capturing holidays, promotions, and seasonality.

Output: Clear seasonal predictions with holiday effects modeled.

**C. Regression Models**

Used for multivariate prediction based on:

* Discount
* Category
* Segment
* Time features

Result: Helped evaluate how features like promotions or product types affect demand.

# Model Evaluation

To finalize our model, we evaluated:

**Metrics Used:**

* RMSE (Root Mean Squared Error)
* MAE (Mean Absolute Error)
* R² (R-squared)

**Models Compared:**

* Prophet
* ARIMA
* Gradient Boosting Regressor
* LSTM Neural Network
* Exponential Smoothing (ETS)
* Random Forest Regressor

**Initial Findings:**

* Prophet had highest R² but high error.
* Random Forest & Gradient Boosting had best trade-off.

**Post-Improvement:**

* Feature engineering & tuning improved results.

**Final Selection:**

* **Gradient Boosting Regressor** chosen for deployment based on low RMSE, low MAE, and high R².

# MLOPs Implementation & Deployment

**1. MLOps Implementation**

To ensure reproducibility and traceability, we applied the following practices:

* **Experiment Tracking**: We used **MLflow** to track model experiments, logging metrics like MAE, MSE, and R² for each run.
* **Version Control**: The entire codebase and model artifacts (e.g., model.pkl, encoders.pkl, preprocessing.py) were version-controlled with **Git** and hosted on **GitHub**.
* **Artifacts Management**: All models and preprocessing steps were saved for reproducibility and reuse.

**2. Model Deployment**

* **Deployment Approach**: The final model was deployed using **Streamlit**, providing an interactive interface for batch and real-time forecasting.
* **Deployment Link**: [Streamlit App](https://deplo-depi-mfep3nxjr3eyw5ftwgbxxc.streamlit.app/)

**Deployment Files:**

* app.py: Streamlit web application
* model.pkl: Trained regression model
* encoders.pkl, onehotencoder.pkl: Encoders used in preprocessing
* preprocessing.py: Scripts for data cleaning and transformation
* **Platform**: Streamlit Community Cloud

**3. Model Monitoring**

* **Performance Metrics** (last evaluation):
  + MAE
  + MSE
  + R² Score
* **Model Drift**: Not currently implemented due to static dataset, but the pipeline supports future integration.

**4. Retraining Strategy**

* **Current**: No new data available, retraining not triggered.
* **Planned**: Monthly or quarterly checks with triggers like:
  + Drop in forecast accuracy
  + Seasonal changes or new product lines

**5. Model Selection Justification**

The selected model was chosen based on its superior performance on the validation set.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | RMSE | MSE | MAE | R2 |
| RandomForest\_all\_features | 123.92 | 15357.8 | 89.11 | 0.580 |
| GradientBoosting\_all\_features | 133.49 | 17819.8 | 97.14 | 0.512 |
| LSTM\_all\_features | 187.85 | 35288.4 | 151.46 | 0.043 |
| ARIMA\_date\_only | 2862.68 | 8194973 | 2231.45 | 0.553 |
| ETS\_date\_only | 3707.74 | 13747388 | 2889.19 | 0.250 |
| Prophet\_date\_only | 36139.98 | 1306098657 | 28828.35 | 0.810 |

Then we took the first two models that has the least Root Mean Squared Error and we tried to improve it more to get the highest accuracy.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | RMSE | MSE | MAE | R2 |
| GradientBoosting\_all\_features | 112.61 | 12682.9 | 80.85 | 0.653 |
| RandomForest\_all\_features | 134.74 | 18157.0 | 98.07 | 0.503 |

The Random Forest Model has became worst but the Gradient Boosting has been improved so we worked on it again.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | RMSE | MSE | MAE | R2 |
| GradientBoosting\_all\_features | 106.49 | 11341.69 | 73.92 | 0.689 |

So the maximum accuracy we could reach is 68.9% with the gradient boosting model.

# Results & Business Recommendations

**Key Results:**

1. **Seasonal Peaks Identified**  
   Sales tend to spike in the last quarter of each year, suggesting opportunities for targeted marketing and inventory planning.
2. **Discounts Drive Sales but Hurt Profit**  
   While high discounts increase demand, they reduce profit margins. A moderate discount level balances both.
3. **Top Performing Segments and Categories**
   * *Consumer* segment had the highest order volume.
   * *Technology* category, especially **Phones** and **Accessories**, generated the highest profits.
4. **Impact of Promotions**  
   Promotions on weekends were effective but inconsistent—testing is needed to find the best timing and product match.
5. **Time Series Forecast Accuracy**  
   Models like ARIMA and Prophet provided reasonably accurate sales forecasts and helped visualize future demand.

# Business Recommendations

**For Retail Management:**

* **Stock Inventory Strategically** during peak months (Q4), especially for top-selling categories like Phones.
* **Implement Targeted Promotions** focused on moderate discounts for high-margin products.

**For Marketing Teams:**

* Use historical promotion data to plan future campaigns.
* Focus more on **Consumer** and **Corporate** segments with personalized offers.

**For Financial Planning:**

* Monitor profit impact of discounts; create discount thresholds.
* Combine sales forecasting with profit forecasting for better budget allocation.

**For Business Expansion:**

* Use geographic data (Country, Region) to identify underperforming markets.
* Scale up successful product lines in high-demand regions.

# Conclusion

Gradient Boosting Regressor proved most effective. Sales data shows consistent growth with seasonal fluctuation. Discounts must be used strategically. Model evaluation clarified the best forecasting strategy. Next steps include deployment and integration with live data.

# GitHub Repository

[GitHub - rahmaanwer22/DEPI\_Project: Sales forecasting and demand prediction](https://github.com/rahmaanwer22/DEPI_Project)

# Resources

* Global Superstore Dataset (2016)
* Python Libraries: pandas, numpy, matplotlib, seaborn, plotly, scikit-learn, statsmodels, fbprophet
* Model references: ARIMA (statsmodels), Prophet (Facebook), Gradient Boosting (scikit-learn)
* Deployment tools (planned): Streamlit, Flask (optional for future phase)
* Academic references and project inspiration: Kaggle retail forecasting examples, Facebook Prophet documentation