

Group Project - Milestone 2: Model Development & Product Integration

1. Model Development Documentation

This section documents the machine learning model development process, from data preprocessing to model evaluation.

Data Preprocessing

- **Data Loading and Cleaning:** The dataset was loaded using `pd.read_csv()` with a custom encoding (ISO-8859-1) to handle non-ASCII characters. The dataset included multiple columns, and the irrelevant ones (like 'Row ID', 'Order ID', etc.) were dropped to focus on the features relevant for prediction.
- **Feature Engineering:**
 - **Date Processing:** Columns like 'Order Date' and 'Ship Date' were converted to datetime objects.
 - **New Features:**
 - **Order_Month:** Extracted the month from 'Order Date'.
 - **Shipping_Duration:** Calculated the difference in days between 'Ship Date' and 'Order Date'.
- **One-Hot Encoding:** Categorical features (like 'Segment', 'Ship Mode', etc.) were one-hot encoded to convert them into numerical representations suitable for machine learning models.
- **Feature-Target Split:** The target variable was 'Profit', while the features consisted of the cleaned and encoded data (excluding 'Profit').

Model Selection

The following models were chosen for comparison:

1. **Linear Regression** (`LinearRegression()` from `sklearn`)
2. **Random Forest Regressor** (`RandomForestRegressor()` from `sklearn`)
3. **XGBoost Regressor** (`XGBRegressor()` from `XGBoost` library)

Model Training and Evaluation:

- **Train-Test Split:** The dataset was split into training and testing sets using `train_test_split()`, with 80% of the data for training and 20% for testing.
- **Evaluation Metrics:**
 - **R² Score:** Measures the proportion of variance explained by the model. A higher value indicates better fit.
 - **MAE (Mean Absolute Error):** Measures the average magnitude of the errors in the model's predictions.
 - **RMSE (Root Mean Squared Error):** Measures the square root of the average squared errors.

Results:

- **XGBoost** performed the best, with an R² score of 0.8143, an MAE of \$22.14, and an RMSE of \$94.88.
- **Random Forest** showed reasonable performance with an R² of -0.0643.
- **Linear Regression** performed poorly with a significantly negative R² and high MAE and RMSE values.

Feature Importance:

For the XGBoost model, feature importance analysis was conducted. This helps identify the key factors that drive the predictions. The model's most important features were identified using the `feature_importances_` attribute.

2. Product Integration Plan

This section describes how the model will be integrated into a product, focusing on user interface, backend operations, and deployment.

User Interface (UI):

The product includes an interactive interface that allows users to input transaction details and predict profit.

- **Widgets and Tabs:**
 - **Basic Info:** Sales, Quantity, Discount, Order Month, Shipping Duration.
 - **Product Info:** Segment, Ship Mode, Category, Sub-Category.
 - **Location:** Region, State, City.

The user inputs are collected through ipywidgets in Jupyter Notebook or through a Streamlit app. These inputs correspond to the features needed for prediction.

Backend Operations:

- **Model Loading:** The XGBoost model, trained using the dataset, is saved and loaded using `joblib.dump()` and `joblib.load()`. This ensures that the model is persistent and can be reused without retraining.
- **Prediction Workflow:**
 - The user provides input through the interface (e.g., Sales, Quantity).
 - These inputs are used to construct a feature vector that matches the format expected by the trained model.
 - The prediction is made using the XGBoost model.
 - Results are displayed, including predicted profit, profit margin, and insights (e.g., if the transaction will be profitable).

Deployment (Streamlit Integration):

- **Web Interface:** The prediction tool is deployed using Streamlit, a framework for creating interactive web applications.
 - The UI is responsive and intuitive, designed with tabs for organization and clarity.
 - The model is accessed through Streamlit's `st.cache_resource()` to ensure efficient resource management and caching of data and model operations.

Scalability and Maintenance:

- **Data and Model Updates:** The dataset and model can be periodically updated, ensuring that the system adapts to new trends or changes in the data.
- **Performance Monitoring:** After deployment, regular monitoring of prediction accuracy and user feedback is recommended to maintain the model's performance over time.

Integration with Business Operations:

- The profit prediction tool can be integrated with business processes, helping sales managers or product teams optimize their decision-making regarding pricing, promotions, and customer targeting.

3. Web App Results & Explanation:

The web app developed in this project predicts the profit for a given transaction based on sales data, product information, and shipping details. Users can input various parameters, such as sales, quantity, discount, and product category, and receive a predicted profit along with a profit margin and feedback on whether the transaction is likely to be profitable.

Web App Interface & User Input

The web app allows users to enter specific transaction details that influence the prediction, including:

- **Sales (\$):** The total sale amount of the transaction.
- **Quantity:** The quantity of items sold in the transaction.
- **Discount:** The discount applied to the transaction.
- **Order Month:** The month in which the order was placed.
- **Shipping Duration (days):** The number of days it took for the item to ship.

Additionally, the product and location information includes:

- **Segment:** The customer segment (e.g., Consumer, Corporate).
- **Ship Mode:** The mode of shipping (e.g., Second Class, First Class).
- **Category:** The product category (e.g., Furniture, Office Supplies).
- **Sub-Category:** The product's sub-category (e.g., Bookcases).
- **Region:** The geographical region of the customer.
- **State:** The state of the customer's location.

Outputs from the Web App

Once the user enters the data, the model generates the following outputs:

- **Predicted Profit:** The model estimates the profit for the transaction.
- **Profit Margin:** The percentage of profit relative to the sales amount.
- **Transaction Feedback:** A clear indication of whether the transaction is likely to be profitable or not.

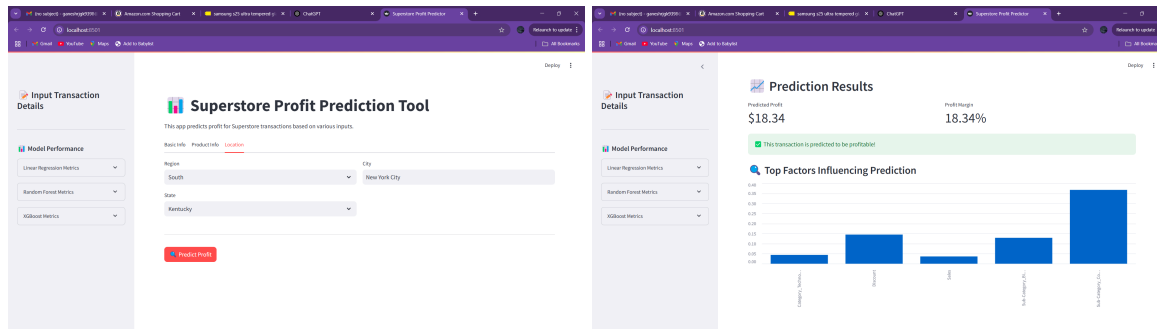
Example Output:

The image displays two side-by-side screenshots of the 'Superstore Profit Prediction Tool' web application interface. Both screenshots show the 'Input Transaction Details' section on the left and the 'Model Performance' section on the right.

Left Screenshot: The 'Input Transaction Details' section includes sliders for 'Sales (\$)' (set to 200.00), 'Discount' (set to 10.00), 'Quantity' (set to 2), 'Order Month' (set to 1), and 'Shipping Duration (Days)' (set to 10). The 'Model Performance' section shows a dropdown menu for 'Linear Regression Metrics'.

Right Screenshot: The 'Input Transaction Details' section includes dropdown menus for 'Segment' (set to Consumer), 'Ship Mode' (set to Second Class), 'Category' (set to Furniture), and 'Sub-Category' (set to Bookcases). The 'Model Performance' section shows a dropdown menu for 'Linear Regression Metrics'.


Both screenshots feature a red 'Predict Profit' button at the bottom of the 'Input Transaction Details' section.




Input:

- **Sales (\$):** 100.00
- **Quantity:** 2
- **Discount:** 0.10 (10%)
- **Order Month:** 6 (June)
- **Shipping Duration (days):** 3
- **Segment:** Consumer
- **Ship Mode:** Second Class
- **Category:** Furniture
- **Sub-Category:** Bookcases
- **Region:** South
- **State:** Kentucky

Output:

- **Predicted Profit:** \$18.34
- **Profit Margin:** 18.34%
- **Transaction Feedback:**  This transaction is predicted to be profitable!

Interpretation of the Results

- **Predicted Profit (\$18.34):** This indicates that the model predicts a profit of \$18.34 from this transaction based on the provided inputs.
- **Profit Margin (18.34%):** The model also calculates the profit margin, showing that the profit is 18.34% of the total sales.
- **Transaction Feedback:** The positive feedback () indicates that this transaction is likely to be profitable, providing reassurance to the user.

User Experience and Insights

The app offers an intuitive experience where users can quickly input their transaction details and receive actionable predictions. The feedback is designed to be user-friendly, providing both a numerical prediction and a clear indication of whether the transaction is likely to be profitable.