Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester:(Summer, Year: 2025), B.Sc. in CSE (Day)

Super Shop Sales Prediction ML-Project

Course Title: Machine Learning Lab
CSE 310 *Course Code: CSE 412*
Section: 221-D2

Students Details

Name	ID
Sabbir Ahmed	221002398
Monowar Hossen Nirob	221002578

Submission Date: 21-08-25
Course Teacher's Name: Md. Riad Hassan

[For teachers use only: Don't write anything inside this box]

Lab Project Status	
Marks:	**Signature:**
Comments:	**Date:**

Contents

Chapter 1

Introduction

1.1 Overview

This project predicts supermarket item sales using machine learning. The dataset includes product details (like weight, price, type, fat content) and outlet details (like size, location, and type). We cleaned the data by filling missing values, analyzed it to see patterns, and converted text into numbers for model training. An XGBoost Regressor [1] was used to train on the data and then tested for accuracy using the R^2 score. The model showed good performance and can help predict future sales, which is useful for better inventory and business decisions

1.2 Motivation

This project aims to make supermarket sales prediction [2] easier and more accurate. Supermarkets often face challenges such as overstocking, understocking, and uncertain customer demand. By applying machine learning, past sales data and product details can be analyzed to build a system that predicts future sales. This will support better stock management, smarter pricing strategies, reduced waste, and improved overall business performance. [3]

1.3 Problem Definition

1.3.1 Problem Statement

Supermarkets often face issues like overstocking, understocking, and lost sales due to inaccurate demand predictions. This project aims to use machine learning to analyze past sales and product data to predict future sales, helping in better inventory management and business decisions

1.3.2 Complex Engineering Problem

- Predicting supermarket sales involves multiple factors like product type, price, visibility, store size, and location.
- Customer demand is unpredictable and varies over time.
- Needs encoding of categorical features for machine learning models
- Requires data cleaning and handling of missing values.
- Model training and tuning XGBoost is challenging to achieve accurate predictions.

1.4 Design Goals/Objectives

- To predict supermarket item sales using machine learning.
- To clean data and handle missing values.
- To analyze product and outlet patterns
- To convert categorical data into numbers
- To build and evaluate an XGBoost model for accurate sales forecasting.

1.5 Application

This project can be applied in supermarkets to improve overall business operations. It helps in managing inventory efficiently by predicting which products will be in demand, reducing issues like overstocking or understocking. It can forecast future sales for different products and outlets, allowing better demand planning and timely replenishment. Additionally, the predictions support smarter pricing strategies and informed business decisions, ultimately reducing losses, improving customer satisfaction, and increasing overall profitability.

Chapter 2

Design/Development/Implementation of the Project

2.1 Introduction

The design and implementation of the supermarket sales prediction project are explained, covering data preparation, cleaning, feature encoding, and exploratory analysis. It also describes how the XGBoost model is built, trained, and evaluated to predict sales for better inventory management and business decision-making.

2.2 Project Details

2.2.1 Architecture Overview

- **Data Collection:** Gather historical sales and product/outlet information from the dataset.
- **Data Preprocessing:** Handle missing values, clean data, and perform feature encoding for categorical variables.
- **Exploratory Data Analysis (EDA):** Analyze patterns, distributions, and relationships between features and sales.
- **Model Training:** Split data into training and testing sets, and train an XGBoost regression model.
- **Model Evaluation:** Test the model on unseen data and measure performance using metrics like R^2 score.
- **Prediction & Application:** Use the trained model to predict future sales, aiding inventory management and business decisions.

2.3 Implementation

2.3.1 Environment and Libraries

The project is implemented Python 3.10 using Google Colab. Key libraries used:

- **pandas** for data manipulation
- **numpy** for numerical operations
- **matplotlib** and **seaborn** for data visualization
- **sklearn** for preprocessing, train-test splitting, and evaluation metrics
- **xgboost** for building the regression model

2.3.2 Dataset

The dataset used for this project is obtained from Kaggle and contains 8,523 rows and 12 columns. It includes details about products, outlets, and sales, which are used to predict future supermarket sales. The dataset consists of both numerical and categorical features, such as `Item_Identifier`, `Item_Weight`, `Item_Visibility`, `Item_MRP`, `Outlet_Identifier`, `Outlet_Size`, and `Item_Outlet_Sales` (target variable).

The dataset can be accessed from the following link: Dataset.

2.3.3 Implementation Steps

- **1. Data Collection:** Load the dataset containing product and outlet details.
- **2. Data Exploration:** View the first few rows, check the shape, and understand feature types.
- **3. Handling Missing Values:**
  - Fill missing `Item_Weight` with the mean.
  - Fill missing `Outlet_Size` using the most frequent value per outlet type.
- **4. Exploratory Data Analysis (EDA):** Visualize distributions of features like `Item_Weight`, `Item_Visibility`, `Item_MRP`, and `Item_Outlet_Sales`.
- **5. Data Preprocessing:** Encode categorical variables (`Item_Type`, `Item_Fat_Content`, `Outlet_Type`, etc.) using label encoding.
- **6. Feature and Target Split:** Separate input features (**X**) and target variable (**Y = Item_Outlet_Sales**).
- **7. Train-Test Split:** Split data into training (80%) and testing (20%) sets.
- **8. Model Training:** Train an XGBoost Regressor on the training set.

- **9. Model Evaluation:** Predict on training and test sets and calculate R^2 score to measure performance.
- **10. Prediction:** Use the trained model to forecast sales for new or unseen data.

2.3.4 Major Code Part

```
super_shop_data['Item_Weight'] = super_shop_data['Item_Weight']
.fillna(super_shop_data['Item_Weight'].mean())
```

Figure 2.1: Mean Imputation for Item Weight

```
super_shop_data.loc[miss_values, 'Outlet_Size'] = super_shop_data
.loc[miss_values,'Outlet_Type'].apply(lambda x: mode_of_outlet_size[x])
```

Figure 2.2: Mode Imputation for Outlet Size

```
# Item Item_Weight distribution
plt.figure(figsize=(6,6))
sns.histplot(data= super_shop_data, x='Item_Weight', kde=True, color='blue')
plt.show()
```

Figure 2.3: Distribution of Item Features

Label Encoding

```
encoder = LabelEncoder()
```

```
super_shop_data['Item_Identifier'] = encoder.fit_transform(super_shop_data['Item_Identifier'])

super_shop_data['Item_Fat_Content'] = encoder.fit_transform(super_shop_data['Item_Fat_Content'])

super_shop_data['Item_Type'] = encoder.fit_transform(super_shop_data['Item_Type'])

super_shop_data['Outlet_Identifier'] = encoder.fit_transform(super_shop_data['Outlet_Identifier'])

super_shop_data['Outlet_Size'] = encoder.fit_transform(super_shop_data['Outlet_Size'])

super_shop_data['Outlet_Location_Type'] = encoder.fit_transform(super_shop_data['Outlet_Location_Type'])

super_shop_data['Outlet_Type'] = encoder.fit_transform(super_shop_data['Outlet_Type'])

```

Figure 2.4: Encoding of Categorical Variables using Label Encoding

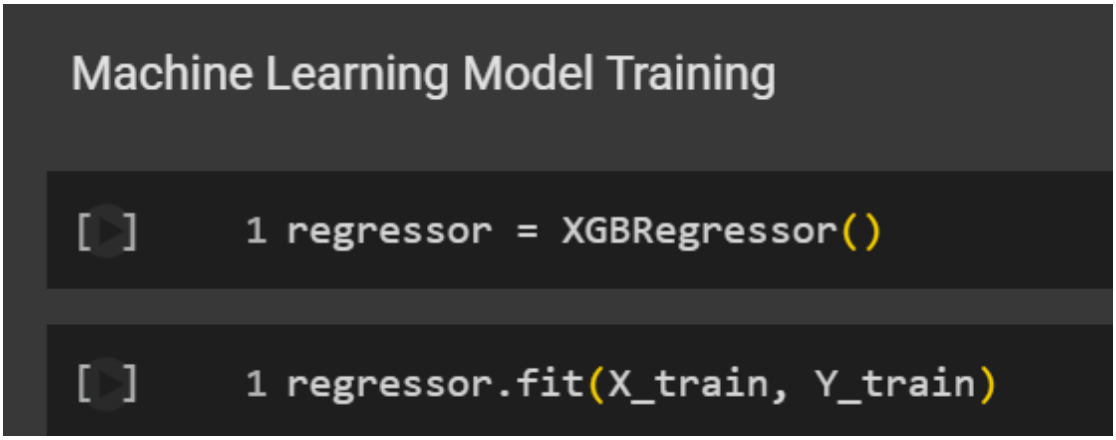

Figure 2.5: XGBoost Regression Model Architecture

Chapter 3

Performance Evaluation

3.1 Simulation Environment/ Simulation Procedure

The project was done in Python 3.13.5 using Google Colab with libraries like pandas, numpy, matplotlib, seaborn, scikit-learn, and xgboost. The dataset was cleaned, features encoded, and split into 80% training and 20% testing. The XGBoost model was trained and evaluated using the R^2 score.

3.2 Result and Evaluation

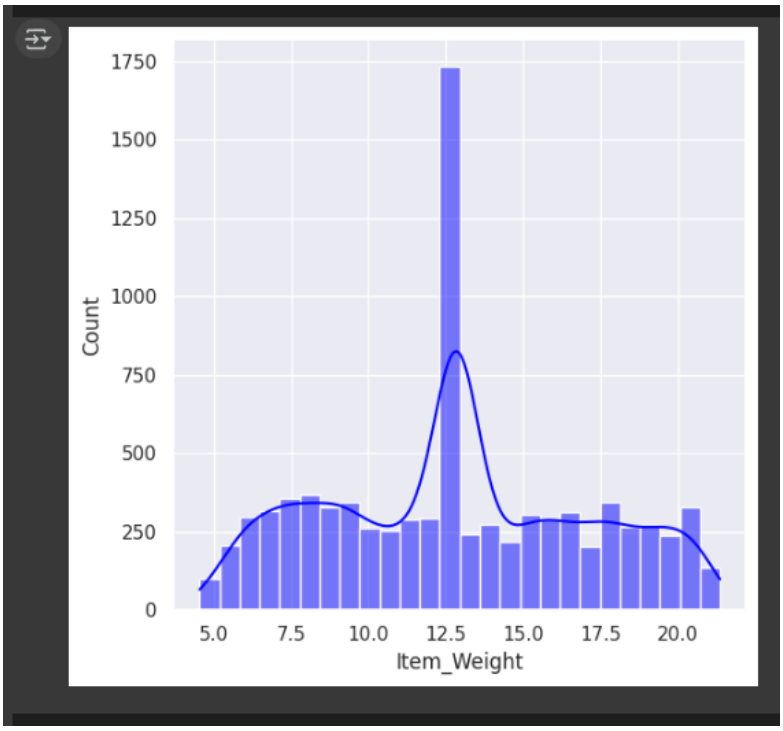

Figure 3.1: Distribution of Item Weight

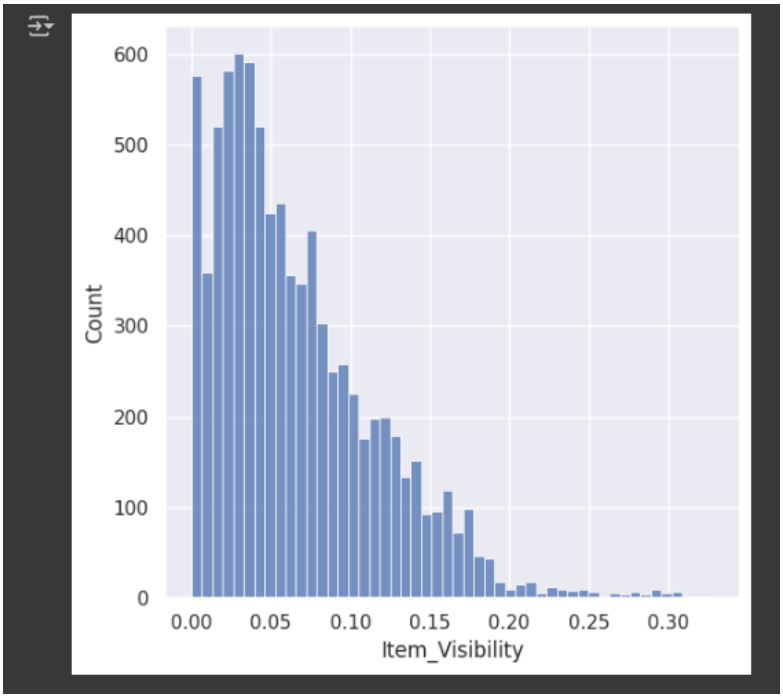

Figure 3.2: Distribution of Item Visibility

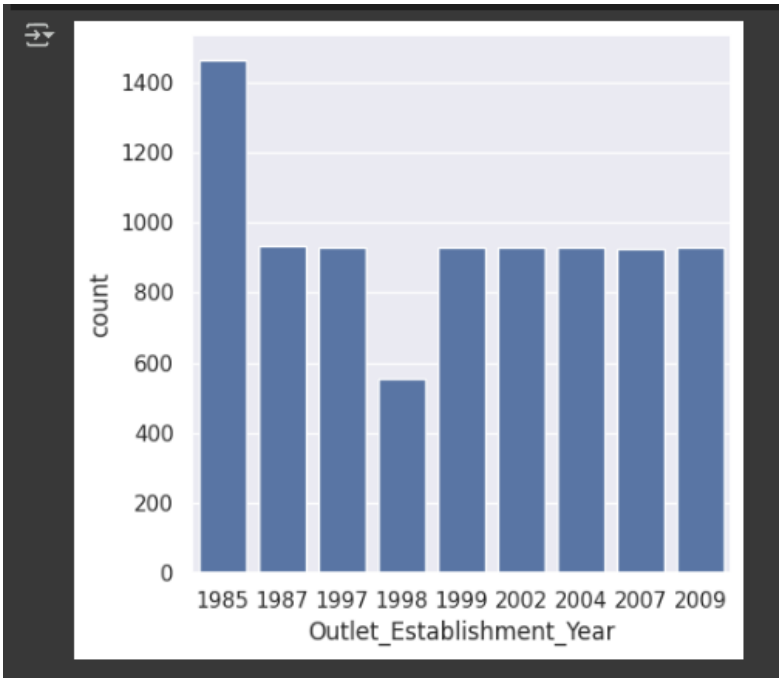

Figure 3.3: Distribution of Outlet Establishment Year

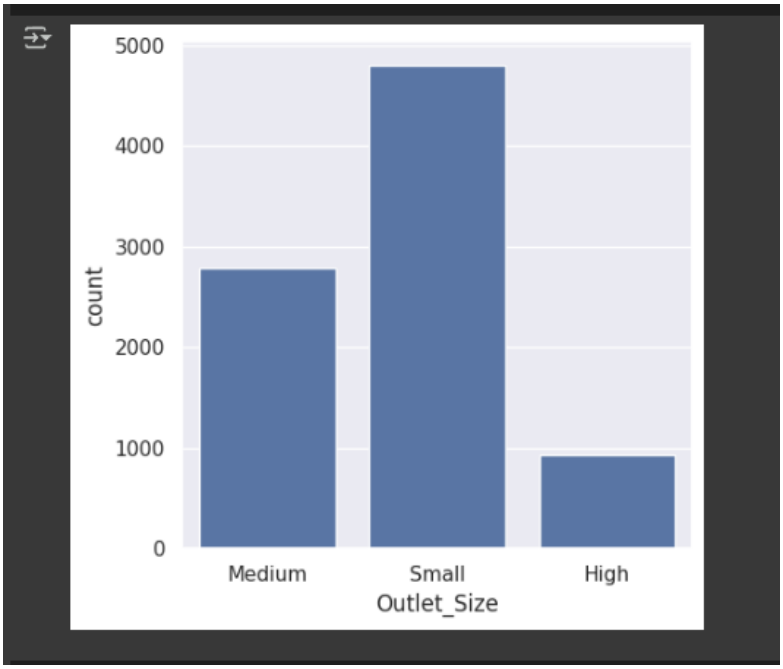

Figure 3.4: Distribution of Outlet Size

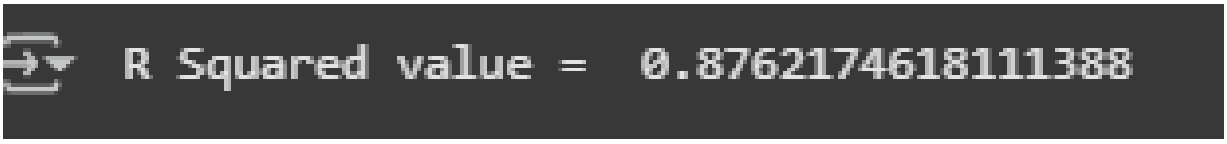

Figure 3.5: Training Data Accuracy of the Model

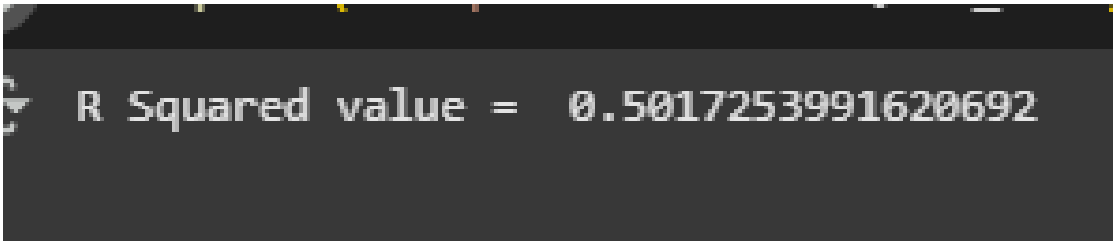

Figure 3.6: Test Data Accuracy of the Model

3.3 Results Overall Discussion

The Supermarket Sales Prediction project uses historical product and outlet data to forecast item sales. The dataset was cleaned, and categorical features were encoded for machine learning. An XGBoost model was trained, achieving good accuracy on training data and moderate accuracy on test data. The model helps supermarkets manage inventory, plan pricing, meet customer demand, and make better business decisions.

Chapter 4

Conclusion

4.1 Discussion

We talked about our To-Do List Shell Scripting Project. Our goal is to make a simple tool for managing tasks easily from the command line. We've included features like adding, removing, editing, listing, and marking tasks as done, as well as the option to set reminders for tasks with deadlines. We want to make the tool easy for everyone to use, so it can help people stay organized and get things done. We'll keep testing and improving it based on feedback to make sure it meets users' needs.

4.2 Limitations

- The model has moderate accuracy on test data and may be overfitting.
- It only uses the available features and ignores trends, promotions, or customer behavior.
- The dataset is small and may not cover all situations, affecting predictions.
- More features or extra data are needed to make the model better.
- The model cannot handle sudden market changes or unexpected events.

4.3 Scope of Future Work

- Use more data and advanced feature engineering to improve accuracy.
- Include seasonal trends, promotions, and customer behavior.
- Try other ML or deep learning models for better predictions.
- Build a real-time sales prediction system.
- Integrate with inventory and pricing tools for automated decisions.

References

[1] S. Aishwarya. Prediction of sales using xgboost. https://www.kaggle.com/code/aishwarya2210/prediction-of-sales-using-xgboost, 2022. Accessed: 2025-08-20.

[2] D. Swami, A. D. Shah, and S. K. B. Ray. Predicting future sales of retail products using machine learning. *arXiv preprint arXiv:2008.07779*, 2020. Accessed: 2025-08-20.

[3] P. Ganguly and I. Mukherjee. Enhancing retail sales forecasting with optimized machine learning models. *arXiv preprint arXiv:2410.13773*, 2024. Accessed: 2025-08-20.