AMRITA SCHOOL OF ARTIFICIAL INTELLIGENCE

23AIE232M PYTHON FOR AI
END SEMESTER REVIEW REPORT

______________________ ______________________

Marks **Faculty Incharge**

AMRITA SCHOOL OF ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641112

BONAFIDE CERTIFICATE

This is to certify that the project entitled “Big Mart Sales Prediction” submitted by

CB.EN.U4CCE23006	Bavya S
CB.EN.U4EEE23109	Deepana S
CB.EN.U4ECE23224	Kadiri Sree Charitha
CB.EN.U4ECE23040	Sai Dharsana I

for the award of the **Degree of Minor** in “**ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING(23AIE232M)** ” is a bonafide record of the work carried out by them under my guidance and supervision at Amrita School of Artificial Intelligence, Coimbatore.

Chandni M

Project guide

Assistant Professor

Submitted for the university examination held on 24-04-2025

INTERNAL EXAM **EXTERNAL EXAMINER**

AMRITA SCHOOL OF ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112

DECLARATION

We, Bavya S (CB.EN.U4CCE23006), Deepana S (CB.EN.U4EEE23109), Kadiri Sree Charitha (CB.EN.U4ECE23224) , Sai Dharsana I (CB.EN.U4ECE23040) hereby declare that this project entitled **"Bigmart Sales Prediction"**, is the record of the original work done under the guidance of Chandni M, Assistant professor, Amrita School of AI, Coimbatore. To the best of my knowledge this work has not formed the basis for the award of any degree/diploma/ associateship/fellowship/or a similar award to any candidate in any University.

Place: Amrita School of AI, Coimbatore **Signature of the Students**

Date: 24-04-2025

COUNTERSIGNED

Dr. K.P.Soman
Professor and Head
Center for Computational Engineering and Networking

Contents

Acknowledgement

Please accept my sincere gratitude for all those who assisted and in some way supported us throughout this project. We especially appreciate our faculty mentor Ms. Chandini M for her enduring support, wise suggestions, and professional critique which shaped the outcome of our project in so many ways. Our thanks also goes to our institution, Amrita Vishwa Vidyapeetham, for the infrastructure and education ecosystem that enabled us to undertake and complete this project successfully. My heartfelt appreciation goes to our wonderful teammates – Bavya S, Deepana S, K Sree Charitha, Sai Dharsana I – for their committed effort, collaboration, and team spirit which turned this project into an enjoyable experience. Last but not least, we are grateful to every member of the faculty and classmates who at some point helped us directly or indirectly. This has been an enriching learning experience and we are grateful for all the support that we got throughout.

List of Abbreviations

EDA – Exploratory Data Analysis
SQL – Structured Query language
DB - Data Base
CSV – Comma Seperated Values
RMSE – Root Mean Squared Error
MAE - Mean Absolute Error
MSE - Mean Squared Error

Abstract:

This project focuses on predicting product sales for Big Mart, a major retail chain, using machine learning techniques. Leveraging a dataset containing historical sales data and product-outlet features, the team explored various regression models to identify the most accurate predictor. Extensive data preprocessing was conducted, including handling missing values, encoding categorical features, and normalization. A comparative analysis of models such as Linear Regression, Decision Trees, Random Forest, Gradient Boosting, SVR, Lasso, and MLP Regressor revealed that XGBoost delivered superior performance with the highest R^2 score and the lowest error metrics. The final model was deployed using the Django web framework, integrating real-time sales prediction through a user-friendly interface and backed by an SQLite database. Web scraping functionality was also introduced to enable future expansion into real time data acquisition. This end-to-end solution exemplifies how machine learning and full-stack development can be synergized to address practical business problems in retail analytics.

1.Introduction

1.1.Problem Statement

Big Mart is a retail chain operating multiple stores across different cities. The company has gathered historical data on product sales along with information related to the products and their respective outlets.
The task is to develop a **machine learning model** that can **predict the sales of a particular product** based on available features such as item type, item price, store size, store location, and other product/outlet characteristics.

1.2. Objectives

- Understand the factors influencing product sales
- Clean and preprocess real-world retail data
- Train and evaluate multiple regression models (Linear Regression, Random Forest, XGBOOST, Lasso, Decision Tree etc.)
- Select the best-performing model to make final prediction
- Integrated the model into a Django web app with SQL DB along with web scrapping

1.3. Literature Review

PAPER1 [1]

Title: Big Mart Sales Prediction Using Machine Learning

Methodology

- Data cleaning, handling missing values, and feature encoding
- Tried multiple models: Linear, Polynomial, Ridge, XGBoost
- XGBoost performed best (based on RMSE, MAE, MSE)
- Integrated the model into a Django web app with SQL DB for real-time predictions

Highlights

- XGBoost achieved the **best performance** among all models
- Feature analysis provided insights into customer preferences and outlet impact
- Supports enhanced **inventory management** and **business decision-making**
- Suggests potential for integration with advanced techniques like ARIMA
- Real-time system design enables quick adaptation to changing sales trends

Limitations

- Limited focus on real-time deployment or application
- No web interface or database integration mentioned

PAPER2 [2]

Title: Machine Learning Insights into Retail Sales Prediction: A Comparative Analysis of Algorithms

Methodology

- Applied Z-score for outlier removal and imputation for missing values
- Trained models: Linear Regression, Random Forest, XGBoost
- Visualized correlations and feature impacts
- Evaluated using R^2, MAE, and RMSE

Highlights

- **Random Forest Regression** showed the best performance:
  - R^2: 0.545
    MAE: 781.64
    RMSE: 1106.81
- **XGBoost** performed second-best:
  - R^2: 0.508
    MAE: 821.80
    RMSE: 1157.61
- **Linear Regression** had the weakest performance:
  - R^2: 0.504
    MAE: 880.99
    RMSE: 1162.44
- Emphasis on visual storytelling using heatmaps and bar plots to identify key features
- XGBoost and Random Forest considered superior for non-linear relationship

Limitations

- Limits generalization
- No use of cross validation
- No time series modelling

PAPER3 [3]

Title: Grid Search Optimization (GSO) Based Future Sales Prediction for Big Mart

Methodology:

- Enhanced XGBoost using Grid Search for hyperparameter tuning
- Created derived features like Item_Type_New, Outlet Age
- Focused on reducing log loss, RMSE, MAE
- Used 10-fold cross-validation for validation

Highlights

- **Ensemble XGBoost + Grid Search** achieved better results than default XGBoost
- Feature importance:
  - Item MRP is the most informative feature

- Performance Comparison:

Metric	**XGBoost (Before Tuning)**	**XGBoost (After Tuning)**
RMSE (Train)	1066	1052
MAE (Train)	749.78	739.03
RMSE (Test)	180.2	178.7
MAE (Test)	134.08	129.90

- Best estimator performance at 300 estimators and learning rate 0.1

Limitations

- Limits generalization
- No use of cross validation
- No time series modelling

2.Background

2.1 Introduction to BigMart Sales Prediction

- Predicting future sales helps optimize inventory and marketing strategies.
- Focuses on analyzing historical sales data from multiple stores and products.
- Enables data-driven decisions for better business outcomes.
- Uses machine learning models to forecast sales accurately.

2.2 Sales Trends and Seasonal Impact

- Sales show clear patterns across months, festivals, and holidays.
- Certain product categories perform better during specific seasons (e.g., summer drinks in hot months).
- Store location and type also affect sales patterns.
- Promotions and discounts lead to temporary spikes in sales.

2.3 Challenges in Predicting Sales Across Stores

- Large variability in product availability and visibility across stores.
- Inconsistent promotional strategies across regions.
- Missing or inconsistent data for some items or outlets.
- External factors like weather or regional events are hard to quantify.
- Model needs to generalize across different store formats and customer behaviour.

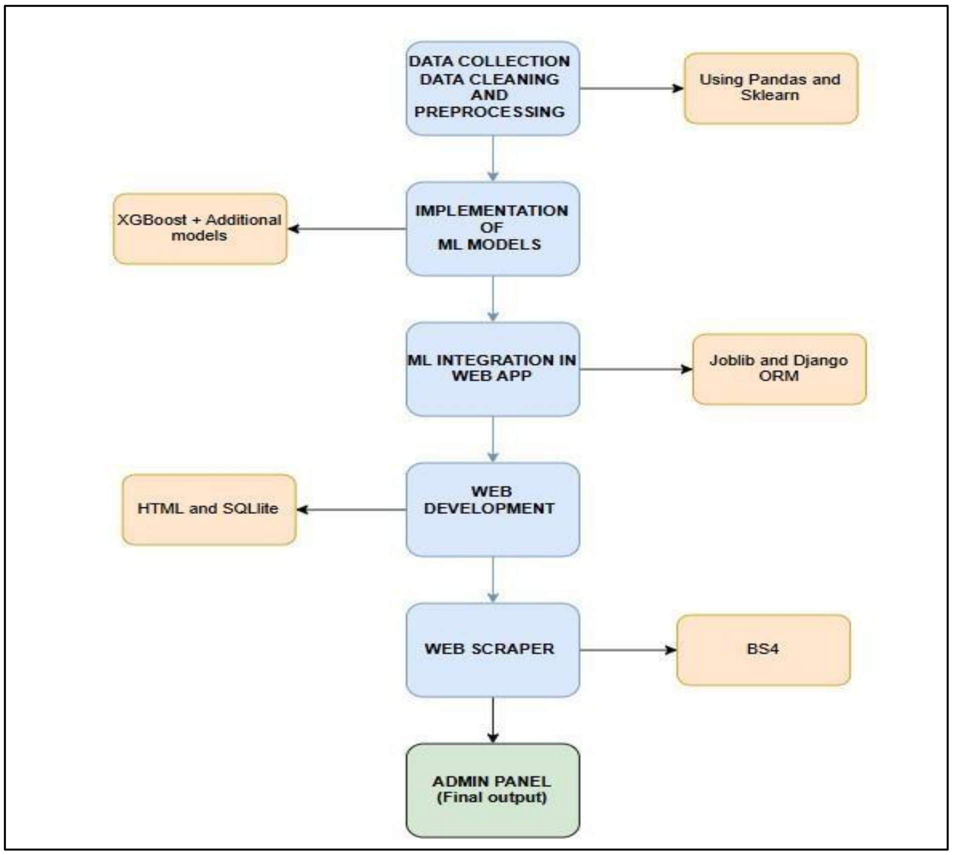
2.4 Dataset Description

- Item_Identifier: Unique ID for each product.
- Item_Weight: Weight of the product.
- Item_Fat_Content: Type of fat content (Low Fat, Regular, etc.).
- Item_Visibility: Percentage of shelf space allocated to the product.

- Item_Type: Category of the product (Dairy, Snacks, etc.).
- Outlet_Identifier: Unique ID for each store.
- Outlet_Establishment_Year: Year the store was established.
- Outlet_Size: Size of the store (Small, Medium, High).
- Outlet_Location_Type: Tier of the city (Tier 1, Tier 2, etc.).
- Outlet_Type: Type of store (Grocery, Supermarket Type1, etc.).
- Item_Outlet_Sales: Target variable representing sales.

3.Proposed Work

3.1 : WORKFLOW :


DATA COLLECTION DATA CLEANING AND PREPROCESSING
Using Pandas and Sklearn
IMPLEMENTATION OF ML MODELS
XGBoost + Additional models
ML INTEGRATION IN WEB APP
Joblib and Django ORM
WEB DEVELOPMENT
HTML and SQLlite
WEB SCRAPER
BS4
ADMIN PANEL (Final output)

3.2 : Technologies & Libraries Used

NumPy:

Used for numerical computations.

Efficient handling of arrays and mathematical operations.

Pandas:

Utilized for loading, cleaning, and manipulating data.

Converts raw data into structured, tabular formats suitable for analysis.

Matplotlib and Seaborn:

Used for data visualization.

Generates histograms, boxplots, scatterplots, and heatmaps to understand data distributions and relationships between features.

Scikit-learn:

Main library for machine learning model development and evaluation.

Handles:

Data preprocessing (label encoding, scaling, train-test splitting)

Implementation of regression models: Linear, Ridge, Polynomial

Metrics used MSE, RMSE, R2

XGBoost:

Specialized library for gradient boosting algorithms.

Used to train the final model that gave the best prediction performance.

Django (Web Framework):

Facilitates web application development using Python.

Handles backend logic, form handling, and routing of user inputs and prediction outputs.

HTML:

Used to create and structure the frontend user interface.

Displays input forms and results in a user-friendly format.

SQL Integration:

Supports CRUD operations (Create, Read, Update, Delete).

Stores user inputs and predicted outputs in a database, improving interactivity and usability.

Web Scraping:

Implemented in the current version.

Project is structured in a way that allows for future integration of real-time data collection via web scraping.

3.3 : EDA

Categorical Features

1. Item_Fat_Content
   - There's some inconsistency in labels (Low Fat, Regular)
   - Low Fat items dominate the dataset, indicating preference or stock pattern.
2. Item_Type
   - A wide variety of item categories (like dairy, snacks, meat, etc.).
   - Helps in grouping for sales trends.
3. Outlet_Size
   - Many outlets have missing or unspecified sizes (nan or blank).
   - Medium-sized outlets are the most common among the specified ones.
4. Outlet_Location_Type
   - Three types of locations: Tier 1, 2, and 3.
   - Tier 3 outlets appear to be the most frequent.
5. Outlet_Type
   - Types include: Grocery Store, Supermarket Type1/2/3.
   - Supermarket Type1 is the most common, while Type3 is rare.
6. Item_Identifier:
   - A unique code for each item.
   - Can be used to extract item category based on the prefix (FD, DR, NC).
7. Outlet_Identifier:
   - A unique code for each outlet.
   - Can be used directly or the numeric suffix can be extracted to create a more general outlet identifier.

Numerical Features

1. Item_Weight
   - Normally distributed with some outliers.
   - Need imputation for missing values.
2. Item_Visibility
   - Right-skewed distribution.
   - Many values are near zero—possibly invalid entries.
3. Item_MRP
   - Bimodal or multimodal distribution—may represent different pricing segments.
   - Useful for sales prediction.
4. Item_Outlet_Sales
   - Heavily right-skewed—most items have low sales, a few with very high sales.
   - Could benefit from log transformation in modeling.

Correlation Heatmap of numerical features:

- **Item_MRP** has strongest positive correlation of **+0.57** with sales, suggests that products with higher MRP tend to have higher sales—likely due to perceived value or brand influence..
- **Item_Visibility** has the weak negative correlation of **-0.13** with sales. Could indicate visibility outliers or that items with higher visibility don't necessarily sell more. May need cleaning or transformation. Item_Visibility need outlier treatment
- **Outlet_Establishment_Year** has a Very weak negative correlation of **-0.05** with sales. Older outlets might be slightly less performant, but not significant. Could be more useful in encoded or bucketed form
- **Item_Weight** with practically **no correlation** with sales. Unlikely to impact sales directly.

Boxplot: Sales by Item Type

- All item types have a wide range of sales, **with similar medians**, but some have more outliers on the higher side — especially Fruits and Vegetables, Dairy, Snack Foods, Household, etc.
- Seafood appears to have **slightly higher median** sales among all — possibly a premium category.
- Health and Hygiene and Frozen Foods also show healthy sales, suggesting consumer interest.

Boxplot: Sales by Outlet Type

- **Supermarket Type 3** has the **highest median and maximum sales**, making it the most profitable outlet type.
- **Grocery Store** has the **lowest median and range**, indicating a relatively underperforming channel. • **Supermarket Type 1 and 2** perform moderately but still better than Grocery Stores.

3.4 : Data Preprocessing

Shape: (8523, 12) (rows, columns) **Missing Values:**

- **Item_Weight**: 1463 missing values filled with **median**.
- **Outlet_Size**: 2410 missing values filled with **mode**.
- **Item_Visibility**: 526 zero values replaced with **median**.

Column Dropping:

- **Dropped**: 'Item_Identifier', 'Outlet_Identifier' (these were considered unnecessary for analysis).

Normalization:

- **Normalized Item_Fat_Content**:
    - ['Low Fat', 'Regular', 'low fat', 'LF', 'reg'] ➔ ['Low Fat', 'Regular'] **Encoding:**
- **Item_Fat_Content**: Encoded into 2 classes (Low Fat, Regular).
- **Item_Type**: Encoded into 16 classes (categories for item types).
- **Outlet_Size**: Encoded into 3 classes (Small, Medium, Large).
- **Outlet_Location_Type**: Encoded into 3 classes (Rural, Semiurban, Urban). • **Outlet_Type**: Encoded into 4 classes (Supermarket Type1, Type2, Type3, Grocery Store).

Correlation Heatmap:

Correlation Among all Features

Strong Correlations

1. Outlet_Size and Outlet_Location_Type**: -0.61**
   → Smaller outlet sizes are more common in certain location types.
2. Item_MRP and Item_Outlet_Sales: **0.57**
   → Higher MRP items are generally associated with higher sales (this is also correlation with the target variable).

Moderate Correlations

1. Outlet_Type and Item_Outlet_Sales**: 0.40**
   → Type of outlet moderately impacts sales.
2. Outlet_Type and Outlet_Location_Type**: 0.47**
   → Certain outlet types are more common in specific locations.

Weak or Negligible Correlations (|r| < 0.2)

- Most other feature pairs show **very weak or no correlation**, such as:
    - Item_Weight with any other feature.
    - Item_Visibility with most features (except weak negative with Outlet_Type: 0.18).
    - Item_Type, Item_Fat_Content, and Outlet_Establishment_Year also show low correlation with others.

Figure 1.1

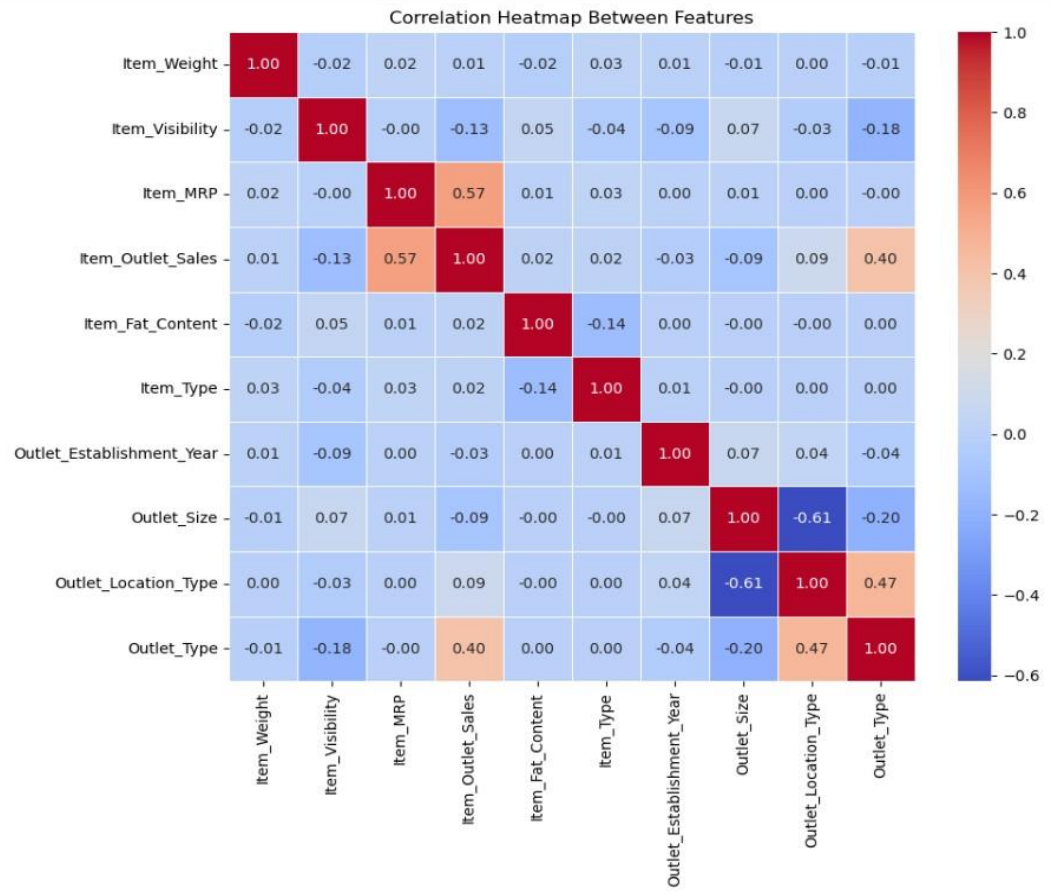
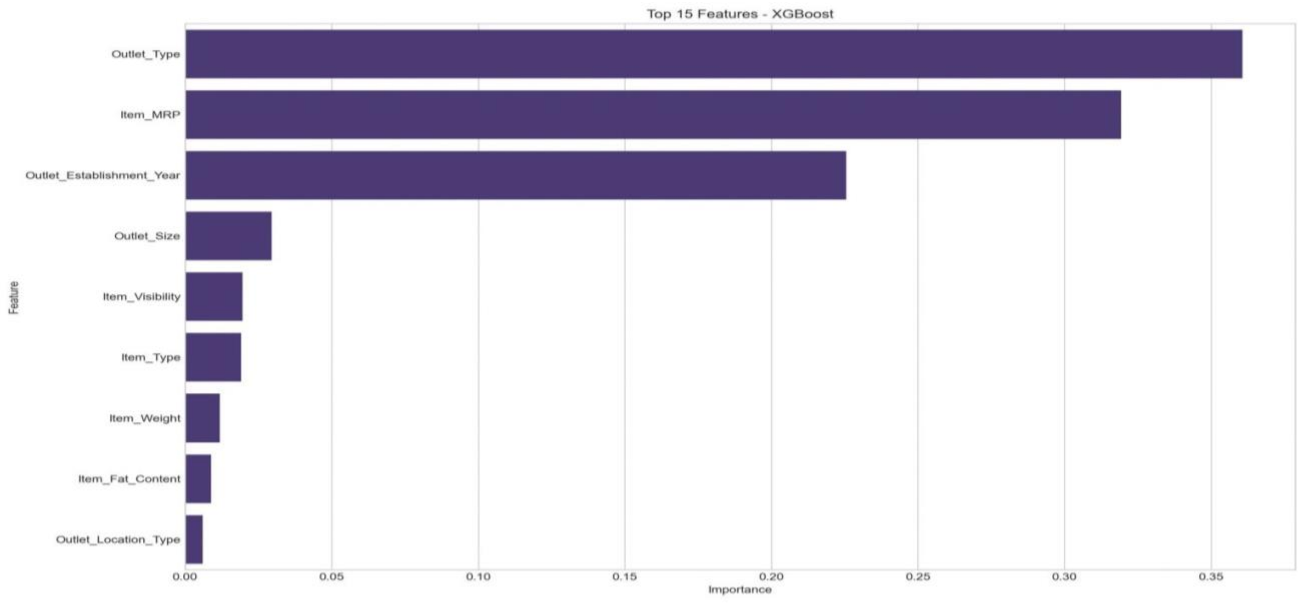


Figure 1.2 Feature Importance:



- Outlet_Type: The type of outlet (e.g., supermarket vs. grocery) is the single most influential predictor of sales (0.361).
- Item_MRP: The item's maximum retail price strongly drives sales, with higher-priced items selling more (0.319).
- Outlet_Establishment_Year: The age of the outlet influences sales patterns, making establishment year important (0.225).
- Outlet_Size: Larger outlets moderately boost sales compared to smaller ones (0.029).
- Item_Visibility: How prominently an item is displayed has a small yet meaningful effect on its sales (0.020).
- Item_Type: The category of the item (e.g., dairy, snacks) slightly affects sales performance (0.019).
- Item_Weight: The weight of the item contributes minimally to predicting sales (0.012).
- Item_Fat_Content: Whether an item is low-fat or regular has a minor impact on sales (0.009)
- Outlet_Location_Type: The urban/semiurban/rural classification of the outlet plays a small predictive role (0.006).

Figure 1.3: Numerical Feature Distribution Graphs

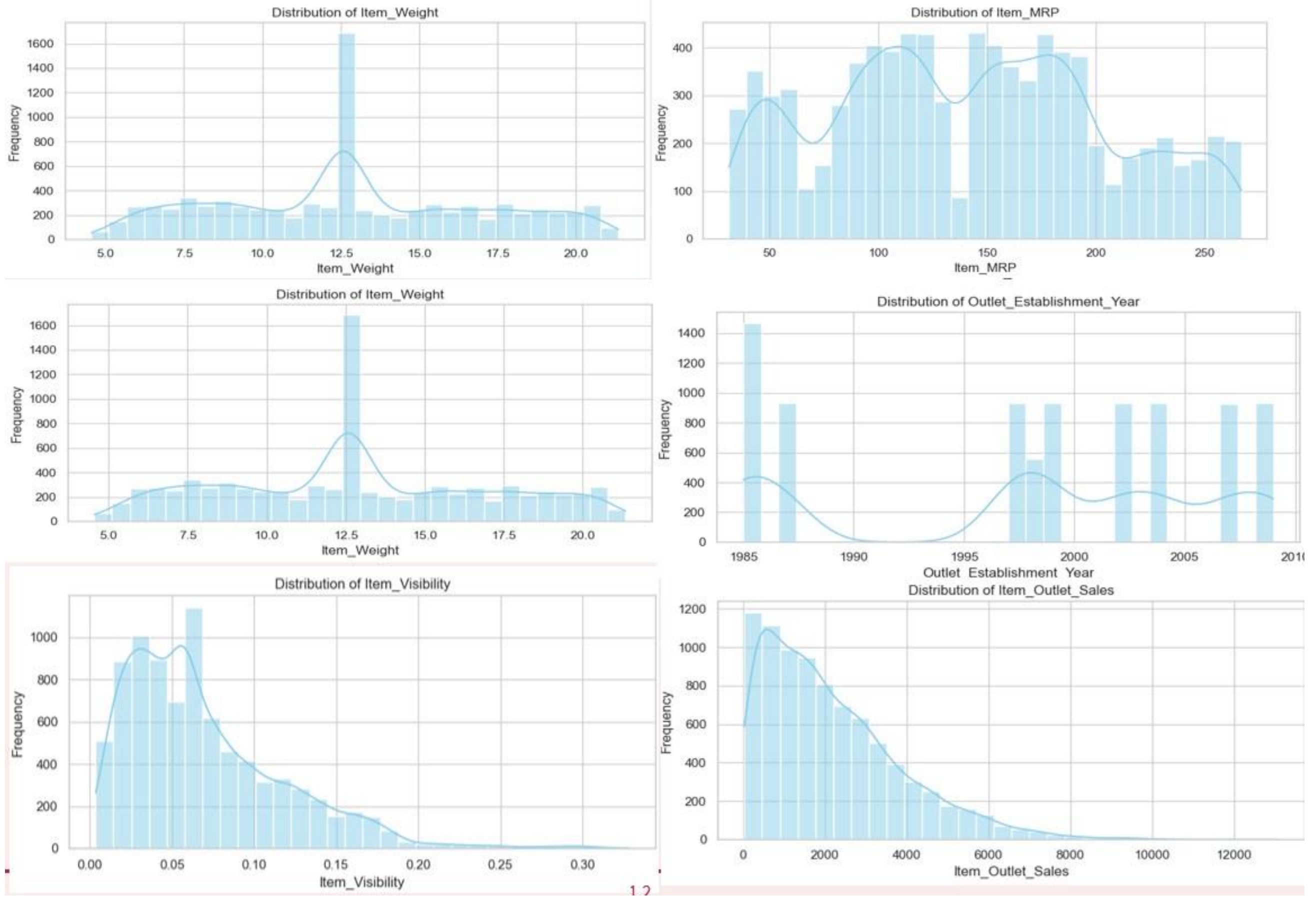
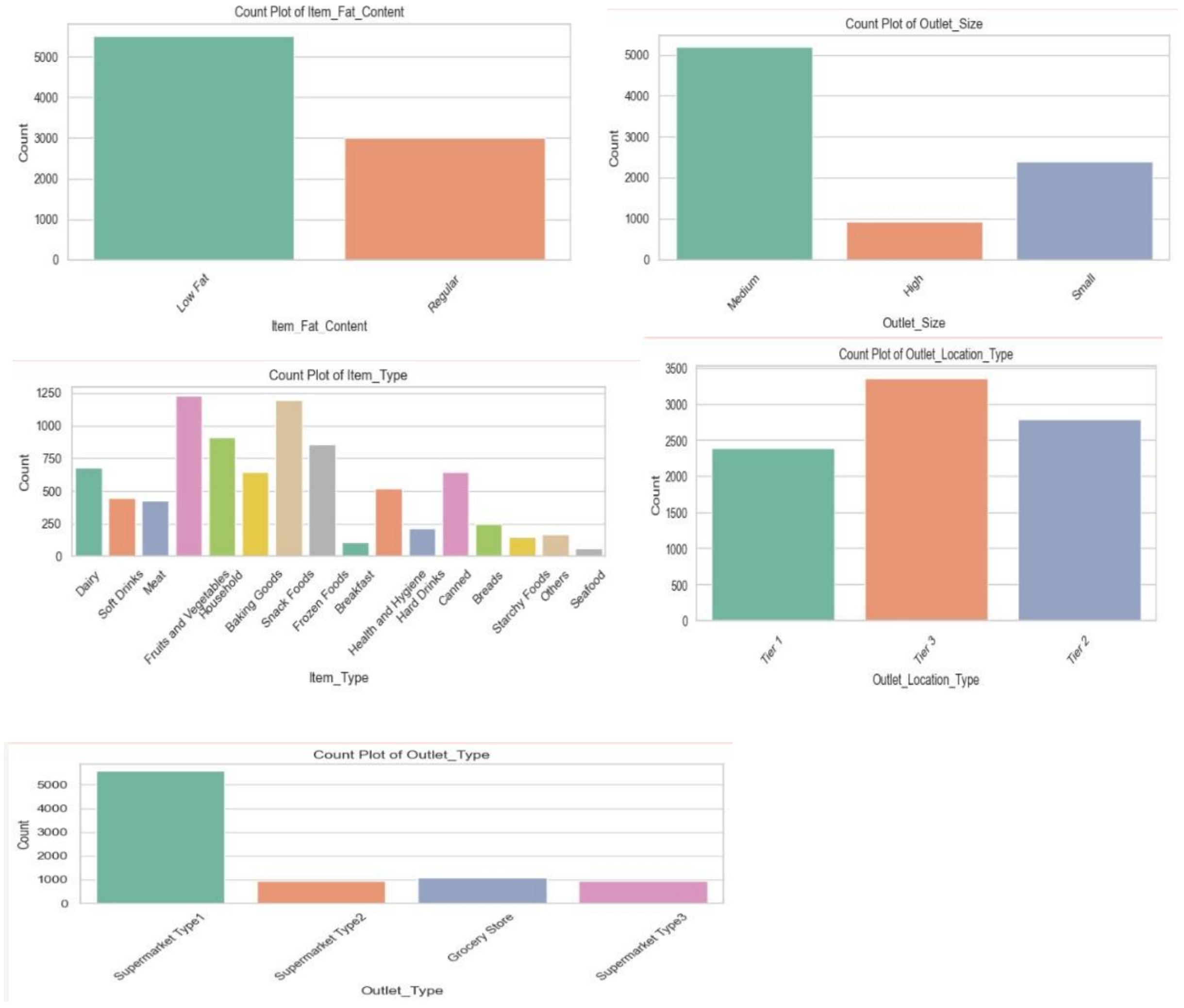


Figure 1.4 Categorical Features Distribution Graphs:



3.6 : Model building:

Feature Engineering (FE)

- **Handling Missing Data**: Fill missing values in Item_Weight and Outlet_Size using the median and mode, respectively.
- **Handling Zero Values**: Replace zero values in Item_Visibility with the median.
- **Encoding Categorical Variables**: Encode Item_Fat_Content, Item_Type, Outlet_Size, Outlet_Location_Type, and Outlet_Type using one-hot encoding or label encoding.
- **Dropping Irrelevant Features**: Drop Item_Identifier and Outlet_Identifier as they don't contribute to the model.

Model Selection & Justification

XGBoost:

- A highly efficient gradient boosting model known for its speed and performance, excelling in handling complex, non-linear relationships and large datasets.
- **Performance**: Best overall performance with the lowest **MSE**, **RMSE**, and **MAE**, and a strong **R^2 = 0.6176**.

MLP Regressor:

- A type of neural network for regression tasks, capable of learning non-linear relationships through multiple layers of nodes. It can perform well but may require significant data preprocessing and tuning.
- **Performance**: Slightly worse than XGBoost with an **RMSE = 1027.67** and **R^2 = 0.6114**.

Gradient Boosting:

- An ensemble method that builds weak learners (typically decision trees) sequentially, with each learner focusing on the errors of the previous one. It performs well but can be sensitive to overfitting if not tuned properly.
- **Performance**: Comparable to MLP Regressor, with very similar metrics but slightly worse performance.

Decision Tree:

- A simple yet powerful model that splits the data into branches based on feature values, providing a clear decision path. It's prone to overfitting but can be pruned to improve generalization.
- **Performance**: Slightly worse than Gradient Boosting wih **R^2 = 0.6094**.

Random Forest:

- An ensemble of decision trees, where each tree is trained on a random subset of the data and features. It reduces overfitting compared to a single decision tree and is robust to noise in the data.
- **Performance**: Slightly worse than Decision Tree with **R^2 = 0.6043**.

LightGBM:

- A gradient boosting framework designed for efficiency and scalability, particularly on large datasets. It uses histogram-based learning to speed up training while maintaining high accuracy.
- **Performance**: Similar to Random Forest, with a slightly higher **MSE** and a marginally lower **R^2**.

Lasso Regression:

- A linear regression model that applies L1 regularization, encouraging sparsity in the model by forcing some coefficients to zero. It is useful for feature selection but struggles with complex relationships.
- **Performance**: Larger errors with **R^2 = 0.5247**.

Linear Regression:

- A simple statistical model that assumes a linear relationship between the input features and the target. It's easy to interpret but limited when the relationships are non-linear.
- **Performance**: Almost identical to **Lasso Regression**, with similar performance metrics.

SVR (Support Vector Regression):

- A type of Support Vector Machine (SVM) used for regression tasks, aiming to find a function that deviates from the true values by less than a specified margin. It works well with non-linear relationships but is sensitive to hyperparameters.
- **Performance**: Performs worse with a significantly higher **MSE** and a lower **R^2 = 0.5130**.

Polynomial Regression:

- A form of linear regression where the relationship between the independent and dependent variables is modeled as an nth-degree polynomial. It's prone to overfitting, especially with high-degree polynomials.
- **Performance**: The least effective model with the lowest **R^2 = 0.3565**.

Figure 1.5: R – Squared Score Graph:

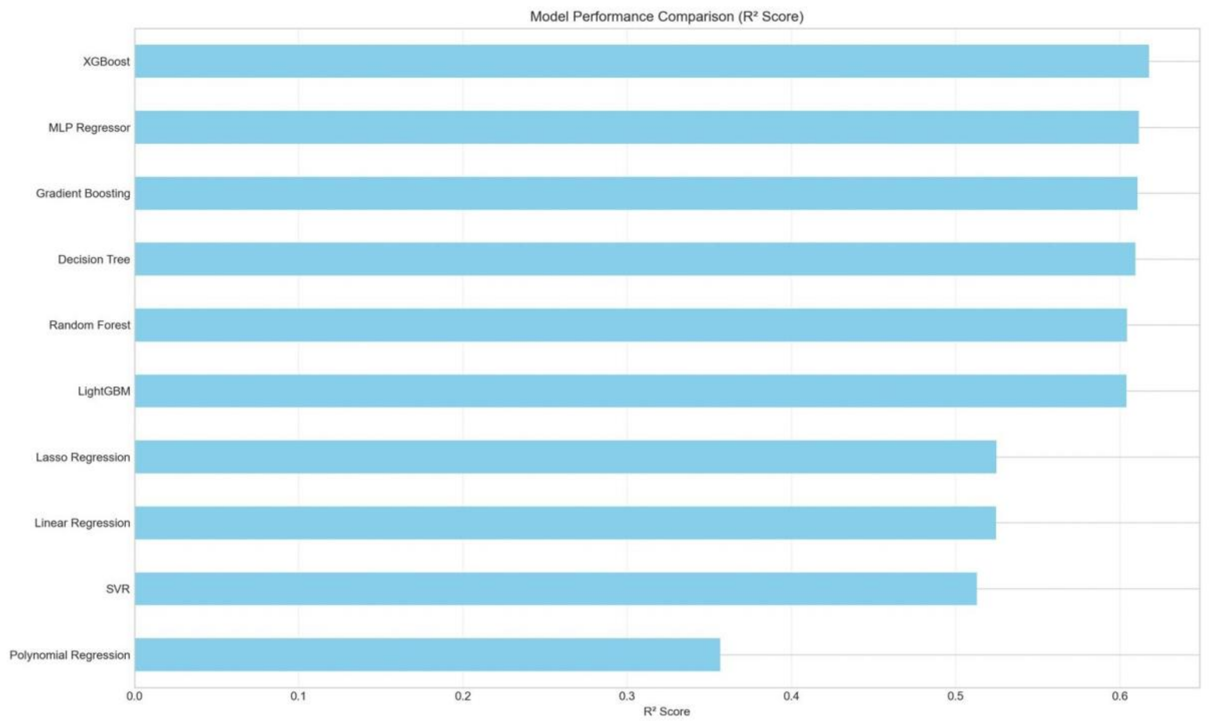


Figure 1.6 RMSE Graph:

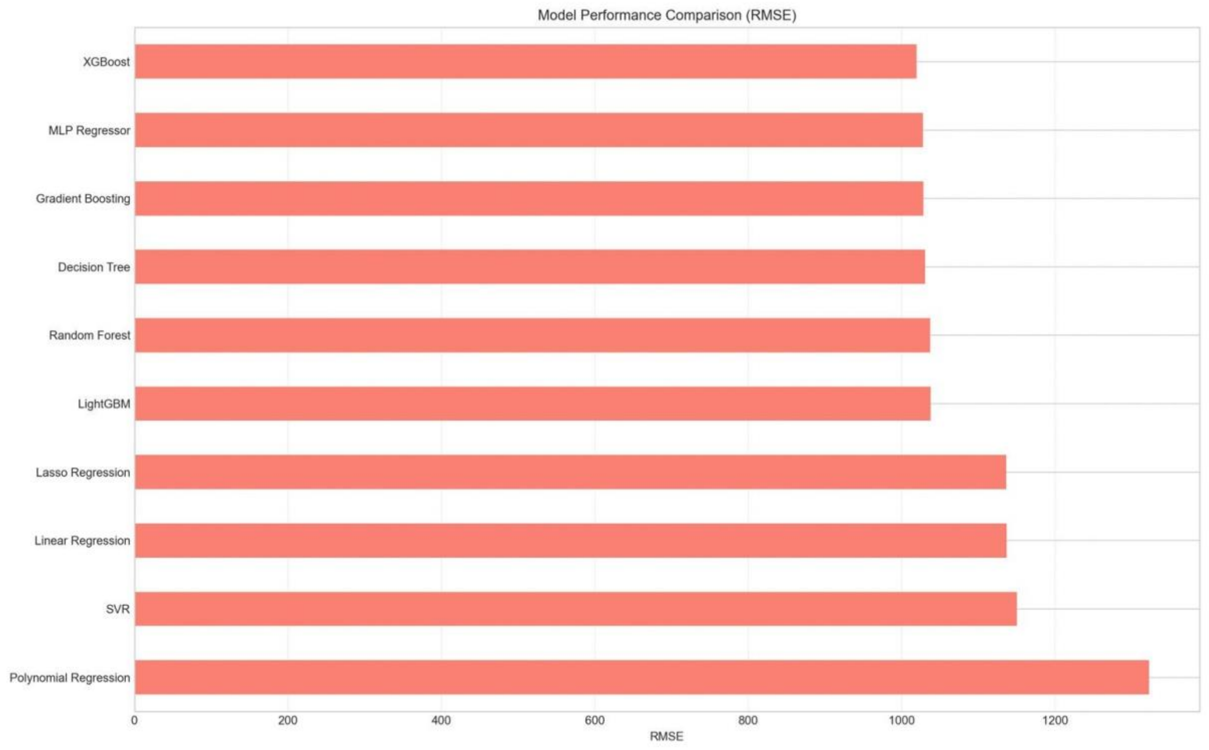


Table 1.1: Model evaluation table

Model	MSE	RMSE	MAE	R²
XGBoost	1.04E+06	1019.454035	716.815724	0.617624
MLP Regressor	1.06E+06	1027.674896	727.198786	0.611432
Gradient Boosting	1.06E+06	1028.59025	721.367656	0.61074
Decision Tree	1.06E+06	1030.36526	723.6076	0.609395
Random Forest	1.08E+06	1037.053622	726.473076	0.604308
LightGBM	1.08E+06	1037.63825	746.174559	0.603861
Lasso Regression	1.29E+06	1136.61236	855.143867	0.524687

Linear Regression	1.29E+06	1136.713497	855.38129	0.524602
SVR	1.32E+06	1150.526308	847.748442	0.512978
Polynomial Regression	1.75E+06	1322.489524	978.353348	0.356513

WHY XGBOOST?
XGBoost outperforms all other models, achieving the best trade-off between error metrics and explanatory power. Models like MLP Regressor and Gradient Boosting follow closely, while more traditional models like Lasso Regression and Polynomial Regression are less effective for this problem.

Model Implementation

- Data Splitting: Split the data into training and test sets (80%,20%). • Model Training: Train the XGBoost regressor using the training data.

Model Evaluation

- Metrics Used:
- Mean Absolute Error (MAE): Measures average errors between predicted and actual values.
- Mean Squared Error (MSE): Penalizes larger errors more heavily.
- R^2 Score: Indicates the proportion of variance in the target variable explained by the model.

4 . Implementation and Results:

Frontend: HTML, CSS (via Django templates)
Backend: Python, Django
Database: SQLite (default Django DB)
Machine Learning: XGBoost, Scikit-learn, Pandas
Deployment Ready: Model saved as .pkl, easily portable
Web page is created using Django, SQLite & Machine Learning Integration:

Django handles:

- Web routing and view logic
- HTML rendering via templates
- Form and file handling for user input
- Integration with ML model (.pkl) for predictions

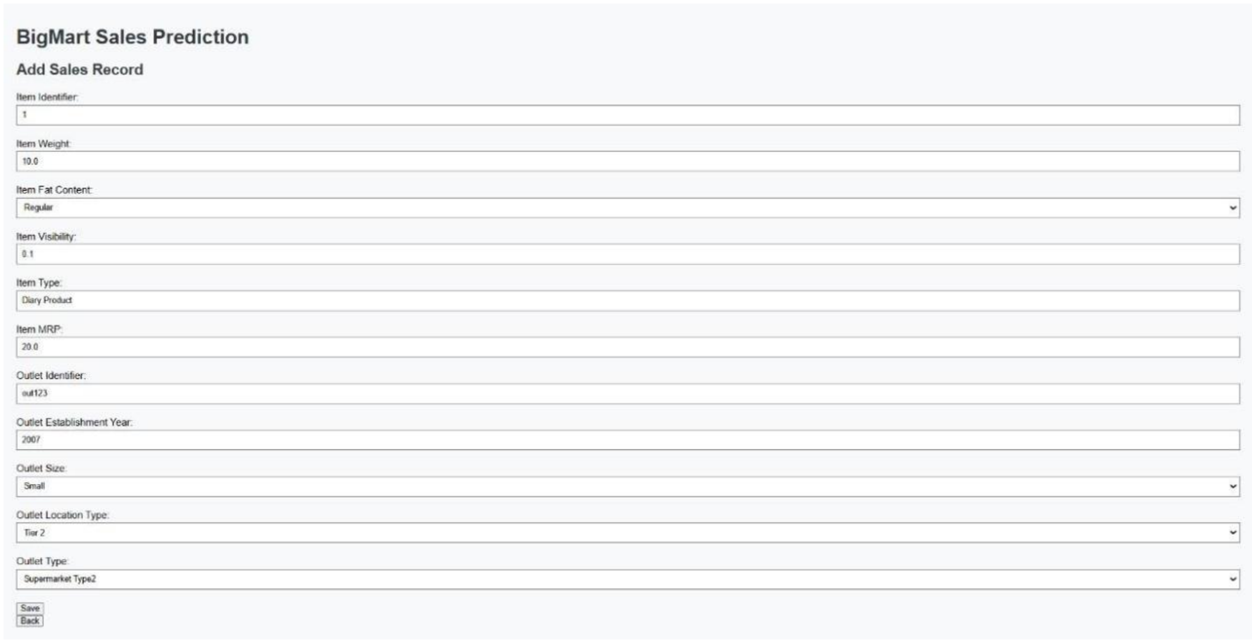
SQLite handles:

- Data storage using db.sqlite3
- Lightweight, file-based relational database
- Managed through Django's ORM (Object-Relational Mapper)
- Used to store user data, logs, or results if needed

Machine Learning integration:

- Trained using XGBoost Regressor
- Model serialized with joblib as xgb_model.pkl
- Auto-loaded by Django during file upload
- Makes predictions in real-time on uploaded data

Figure1.7:



BigMart Sales Prediction

Add Sales Record

Item Identifier:	1
Item Weight:	10.0
Item Fat Content:	Regular
Item Visibility:	0.1
Item Type:	Diary Product
Item MRP:	20.0
Outlet Identifier:	out123
Outlet Establishment Year:	2007
Outlet Size:	Small
Outlet Location Type:	Tier 2
Outlet Type:	Supermarket Type2

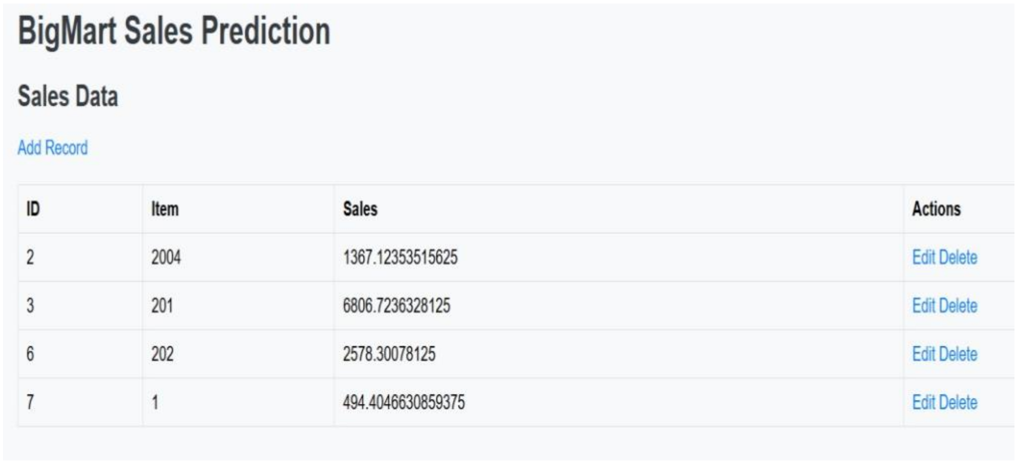
This clean and user-friendly interface for the BigMart Sales Prediction system provides an intuitive form for inputting detailed product and outlet information.

This form allows users to enter new data into the system so that the model can use it for predicting sales

Action Buttons:

- Save – Submits the entered information to the backend or database. • Back – Returns to the previous page without saving.

Figure 1.8:



BigMart Sales Prediction

Sales Data

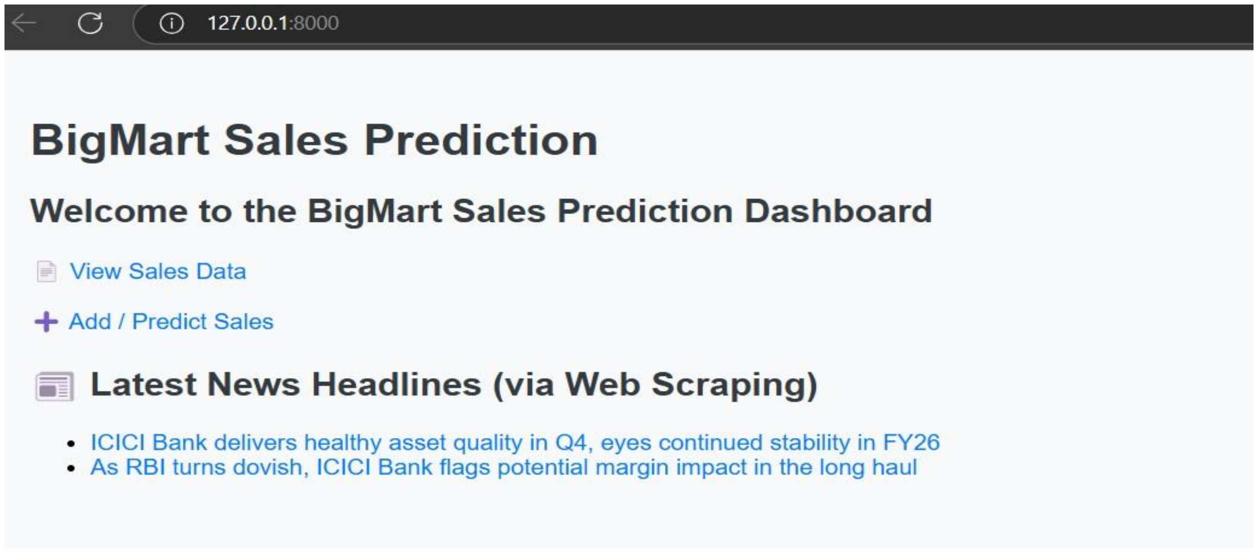
Add Record

ID	Item	Sales	Actions
2	2004	1367.12353515625	Edit Delete
3	201	6806.7236328125	Edit Delete
6	202	2578.30078125	Edit Delete
7	1	494.4046630859375	Edit Delete

This page acts as a **data dashboard**, showing all item entries and their predicted sales values.
Features:

- Add Record – A clickable link at the top that navigates the user back to the data entry form (as seen in the first picture), allowing the addition of new item records.
- Edit/Delete Functionality – Users can update or remove existing records, making it easy to manage data in the system.

Figure 1.9: Django Admin panel:



BigMart Sales Prediction

Welcome to the BigMart Sales Prediction Dashboard

View Sales Data

Add / Predict Sales

Latest News Headlines (via Web Scraping)

- ICICI Bank delivers healthy asset quality in Q4, eyes continued stability in FY26
- As RBI turns dovish, ICICI Bank flags potential margin impact in the long haul

Web scraping is the automated process of extracting data from websites. It is widely used to collect information such as product prices, news articles, stock data, or social media content from publicly available web pages.

BeautifulSoup – Parses and extracts data from HTML/XML documents.

requests – Sends HTTP requests to fetch webpage content.

5. Conclusion:

This project successfully integrates a full-stack machine learning application using the Django web framework and a pre-trained XGBoost regression model for sales prediction. The application allows users to upload CSV datasets, which are automatically preprocessed and passed to the ML model for generating real-time predictions. The results are then rendered in a user-friendly format on the web interface.

Django plays a crucial role in handling routing, frontend rendering, and backend logic, while SQLite provides lightweight and efficient database support for data persistence. The Django Admin Panel adds additional convenience by enabling secure management of users and data records.

Alongside this, web scraping techniques such as BeautifulSoup and Selenium were discussed as potential methods for collecting external data for machine learning or analytics purposes.

Overall, the system showcases the practical application of machine learning in a web environment and demonstrates how open-source technologies like Django, SQLite, and XGBoost can be seamlessly combined to deliver an intelligent, scalable, and interactive data-driven application.

6.REFERENCES:

[1]https://ieeexplore.ieee.org/document/10544274

[2]https://ieeexplore.ieee.org/document/10777132

[3]https://ieeexplore.ieee.org/document/9067927

7.List of Publications based on this research work

1.Ching Wu Chu and Guoqiang Peter Zhang, “A comparative study of linear and nonlinear models for aggregate retails sales forecasting”, Int. Journal Production Economics, vol. 86, pp. 217- 231, 2003

2. K. Punam, R. Pamula, and P. K. Jain, “A two-level statistical model for big mart sales prediction,” in 2018 International Conference on Computing, Power and Communication Technologies (GUCON). IEEE, 2018, pp. 617–620.

3. Zone-Ching Lin, Wen-Jang Wu, “Multiple LinearRegression Analysis of the Overlay Accuracy Model Zone”, IEEE Trans. on Semiconductor Manufacturing, vol. 12, no. 2, pp. 229 – 237, May 1999.

4. O. Ajao Isaac, A. Abdullahi Adedeji, I. Raji Ismail, “Polynomial Regression Model of Making Cost Prediction In Mixed Cost Analysis”, Int. Journal on Mathematical Theory and Modeling, vol. 2, no. 2, pp. 14 – 23, 2012.

5. G. Behera and N. Nain, “A comparative study of big mart sales prediction,” in *Proceedings of International Conference on Computer Vision and Image Processing*. Springer, 2019.