Solution Approach of BigMart Sales Prediction Using XGBoost :

1️⃣ Data Understanding

The dataset consists of two files:

✅ **Train.csv** → Contains sales data with target values (Item_Outlet_Sales).

✅ **Test.csv** → Contains the same features but **without** sales values (this is what we predict).

🔹 Key Features in the Dataset:

- **Product Information**: Item_Weight, Item_Fat_Content, Item_Type, Item_MRP
- **Store Information**: Outlet_Size, Outlet_Location_Type, Outlet_Type, Outlet_Establishment_Year
- **Target Variable**: Item_Outlet_Sales (only in the training set)

2️⃣ Data Preprocessing & Feature Engineering

🔹 Handling Missing Values

- **Item_Weight** → Replaced missing values with the **mean**.
- **Outlet_Size** → Filled missing values with **'Medium'** (most frequent category).

🔹 Standardizing Categorical Data

- **Inconsistent Labels** in Item_Fat_Content (low fat, LF, reg) were fixed.

◇ Creating a New Feature

- **num_years** → We calculated how many years have passed since each store was established.

◇ Encoding Categorical Variables

- We converted categorical variables into numeric format using **One-Hot Encoding.**
- The same encoding was applied to the test set to ensure both datasets have the **same feature structure**.

3 Model Selection & Training

We used the **XGBoost Regressor**, which is a powerful gradient boosting model.

◇ Splitting Data

Before training, we split the dataset into **training and validation sets.**

◇ Training XGBoost

We trained the model with optimized hyperparameters for better accuracy.

4 Model Evaluation

After training, we evaluated performance using **RMSE (Root Mean Squared Error) and R^2 Score.**

- **Lower RMSE** → Better predictions.
- **Higher R^2** → Model explains more variance in the data.

5️⃣ Solving the Negative Predictions Issue

During testing, we found that **XGBoost can sometimes predict negative sales**, which is not valid.
To fix this, we used **post-processing** to ensure all predictions are non-negative.

6️⃣ Generating Final Predictions

Once predictions were corrected, we saved them in a CSV file for submission.

7️⃣ Visualizing Predictions

We plotted the **distribution of predicted sales** to ensure they look reasonable.

🎯 Conclusion

This project successfully builds an **XGBoost-based sales prediction model** for **BigMart stores**. The model captures key relationships between product attributes and sales trends, and produces **realistic predictions**.