# Big Mart Sales Prediction

## 1. Introduction

The Big Mart Sales Prediction project aims to forecast sales for products across various Big Mart outlets. By using machine learning algorithms, we predict Item Outlet Sales based on item and outlet attributes. Accurate sales forecasting helps Big Mart manage inventory, plan marketing strategies, and improve supply chain operations.

## 2. Dataset Overview

The project utilizes two datasets: Train.csv (with sales numbers) and Test.csv (without sales numbers). Key features include Item\_Identifier, Item\_Weight, Item\_Fat\_Content, Item\_Visibility, Item\_Type, Outlet\_Identifier, Outlet\_Size, Outlet\_Location\_Type, Outlet\_Type, and Item\_Outlet\_Sales (target variable).

## 3. Data Preprocessing

### 3.1 Handling Missing Values

1. Item Weight Imputation (Group-wise Mean Imputation):  
Missing values in the Item\_Weight column were filled using the mean weight within each Item\_Type group. This preserves internal category-specific characteristics, enhancing model learning.

2. Outlet Size Imputation (Group-wise Mode Imputation):  
Missing Outlet\_Size values were filled based on the most frequent size observed within each Outlet\_Type category. This maintains logical consistency and realistic data relationships, improving generalization and model robustness.

### 3.2 Standardizing Categories

Inconsistent entries in Item\_Fat\_Content were standardized by unifying similar labels (e.g., 'low fat', 'LF' to 'Low Fat').

### 3.3 Feature Engineering

Several additional features were created to enhance model performance:  
- Item\_Visibility\_MeanRatio: Ratio of an item's visibility compared to its average visibility.  
- Outlet\_Years: Store age calculated from Outlet\_Establishment\_Year.  
- Item\_Category: Broad product classification (Food, Drinks, Non-Consumable) extracted from Item\_Identifier.

### 3.4 Encoding Categorical Variables

Categorical features were transformed using label encoding and one-hot encoding to prepare them for machine learning models.

### 3.5 Target Variable Transformation

The Item\_Outlet\_Sales variable was log-transformed using np.log1p() to normalize its distribution and stabilize variance, leading to improved model performance.

## 4. Model Building

After preprocessing, the dataset was split into training and testing sets. Multiple regression models were trained and evaluated:  
- Linear Regression  
- Decision Tree Regressor  
- Random Forest Regressor  
- LightGBM Regressor  
- XGBoost Regressor

## 5. Model Evaluation

When deciding on the best model, it is essential to evaluate based on test performance, as it reflects the model's ability to generalize to unseen data. Below is a comparison of model performances based on R² scores and Root Mean Squared Error (RMSE):

Random Forest:  
- Train R² Score: 0.7481  
- Test R² Score: 0.7259  
- Test RMSE: 0.5402  
  
LightGBM:  
- Train R² Score: 0.8060  
- Test R² Score: 0.7198  
- Test RMSE: 0.5462  
  
XGBoost:  
- Train R² Score: 0.8972  
- Test R² Score: 0.6875  
- Test RMSE: 0.5768

### Key Observations:

- Random Forest achieved the best test R² and RMSE, suggesting excellent generalization.  
- LightGBM performed closely but with slightly inferior generalization.  
- XGBoost showed signs of overfitting, despite high training performance.

### Overfitting Analysis:

- XGBoost overfits significantly, capturing noise.  
- LightGBM shows moderate overfitting.  
- Random Forest balances train and test performance best, minimizing overfitting.

### Model Selection Recommendation:

Considering generalization performance and overfitting risk, Random Forest is recommended as the final model.

## 6. Model Interpretation (SHAP Analysis) on the training dataset

A screen shot of a graph

AI-generated content may be incorrect.

SHAP (SHapley Additive exPlanations) analysis was performed to interpret feature contributions to the Random Forest model's predictions.

### 6.1 SHAP Summary Plot Insights:

- Outlet\_Type emerged as the most influential feature.  
- Item\_MRP also had a strong positive impact, where higher prices often related to higher predicted sales.  
- Outlet\_Age and Outlet\_Establishment\_Year significantly contributed, suggesting outlet maturity influences sales.  
- Specific outlet identifiers like OUT027, OUT018, and OUT019 influenced performance, indicating store-specific behaviors.  
- Item-related factors like weight, visibility ratios, and item categories played moderate roles.

### 6.2 Color Interpretation:

The color gradient (blue to pink) represents feature value magnitudes. Higher values generally shift predictions up or down depending on the feature’s impact.

### 6.3 Final Interpretation:

Outlet characteristics dominate predictive strength, followed by product MRP and select visibility features. Understanding feature impacts allows Big Mart to align marketing and operational efforts based on feature importance insights.

## 7. Results and Conclusion

Extensive feature engineering, thoughtful preprocessing, and robust model evaluation led to Random Forest emerging as the best model. The model, along with SHAP-based insights, provides actionable strategies for inventory planning, targeted promotions, and operational improvements.

## 6. Hyperparameter Tuning (Random Forest Regressor)

Hyperparameter tuning for Random Forest was performed to optimize model performance and minimize overfitting.  
  
\*\*Key Hyperparameters Tuned:\*\*  
- \*\*n\_estimators=100\*\*: Defines the number of trees in the forest. Increasing the number of trees improves model stability and accuracy.  
- \*\*max\_depth=None\*\*: Trees were allowed to expand fully unless limited by other stopping criteria, capturing complex relationships.  
- \*\*random\_state=42\*\*: Ensured reproducibility of results by controlling randomness.  
  
This configuration provided a robust model that generalized well to unseen data without significant overfitting. The ensemble approach of Random Forest effectively reduced variance and captured non-linear patterns in the dataset.