Big Mart Sales Prediction Hackathon – Project Summary

Objective –

- Goal: To Predict product sales across Big Mart outlets using historical data
- Problem Type: Regression (predicting continuous sales values)
- Success Metric: Root Mean Squared Error (RMSE)

Data Understanding & Challenges

The dataset contained 8,523 training rows and 5,681 test rows with 12 features describing products (weight, fat content, visibility, type, price) and outlets (size, location, type, establishment year).
Key challenges included:

- **Missing values:** 1,463 missing Item_Weight and 2,410 missing Outlet_Size.
- **Inconsistent labels:** Item_Fat_Content had messy categories ("LF," "low fat," "reg").
- **Invalid values:** 879 items had Item_Visibility = 0, which can't be true.
- **large number of distinct values:** 1,559 unique Item_Identifier categories made the feature too granular.

Initial Approach & Correction

Initially, I combined the train and test data to ensure all cleaning and feature engineering steps were applied consistently. While this approach helped in aligning preprocessing, it introduced **data leakage**, since imputation rules were influenced by test data. Recognizing this issue, I reprocessed the datasets separately – fitting imputations and transformations only on the training data and then applying them to the test set.

After resolving this, I performed a thorough examination of data quality, created new features, and simplified categories (e.g., grouping product identifiers, standardizing fat content, deriving outlet age). This correction ensured that preprocessing was both accurate and implementable to unseen data.

Data Cleaning & Preprocessing

- Imputed missing Item_Weight by average weight per product (Item_Identifier), as a fallback I added another step to impute by Item_Type.
- Filled missing Outlet_Size based on the most frequent size for that Outlet_Type.
- Standardized inconsistent Item_Fat_Content labels e.g LF and low fat to Low Fat, reg to Regular.
- Replaced zero visibility values with mean visibility per product.
- Converted categorical variables to factors.

Feature Engineering

To reduce overfitting and improve model performance, several derived features were added:

- **Item_Type_Combined:** Grouped 16 product categories into 3 groups (Food, Drinks, Non-Consumables).
- **Item_Fat_Content_Clean:** Fixed inconsistencies and marked non-consumables as "Non-Edible."
- **Outlet_Age:** Derived as (2013 – Establishment Year).
  Multiple feature subsets (Set3–Set8) were tested, with **Set8** yielding the best results.

Model Development & Evaluation

I designed a model selection workflow to benchmark multiple algorithms. Six models were tested using 5-fold cross-validation:

- Linear Regression (baseline), Ridge, Lasso
- Decision Tree, Random Forest
- XGBoost (gradient boosting)

XGBoost consistently outperformed the others, achieving an RMSE of ~1,015, compared to ~1,200 for linear baselines.

Final Model & Deployment

- Retrained **XGBoost** on the full training set using the best-performing feature set.
- Tuned hyperparameters (nrounds, eta, max_depth, gamma) for optimal performance.
- Adjusted predictions with pmax(pred, 0) to avoid negative sales outputs.
- Used the retrained model for final test predictions.