Big Mart Sales Prediction Challenge

1. Executive Summary

The project goal was to build a robust regression model for Item_Outlet_Sales. The solution evolved from a simple baseline to a high-performance ensemble by addressing data quality through advanced model-based imputation (Random Forest), rigorous feature engineering, and a segmented modeling strategy. The final submission utilizes a weighted ensemble of XGBoost, LightGBM, and CatBoost, complemented by a segmented approach that models Grocery Stores and Supermarkets as distinct populations to capture their unique sales dynamics.

2. Advanced Data Preprocessing

Initial EDA on the 8,523 training and 5,681 test rows revealed significant missing values in Outlet_Size (~28%) and Item_Weight (~17%). Standard mean/mode imputation was rejected to prevent data distortion.

A. Model-Based Imputation Strategy

Instead of simple fills, I used machine learning to predict missing values based on intrinsic feature relationships:

- Outlet_Size (Multi-Class Classification):
  - **Logic:** Store size is physically constrained by its location and type.
  - **Method:** Trained a **Random Forest Classifier** using Outlet_Type and Outlet_Location_Type as predictors.
  - **Result:** The model accurately predicted missing sizes (e.g., inferring "Small" for Tier 2 Supermarkets) consistent with observed patterns.
- Item_Weight (Hybrid Approach):
  - **Stage 1 - Deterministic Lookup:** Created a mapping of Item_Identifier to Item_Weight from the training data. If an item appeared in another store with a known weight, that value was propagated.
  - **Stage 2 - Regression Prediction:** For remaining missing values (new/rare items), I exploited the high correlation between Item_MRP and weight in specific categories (e.g., Baking Goods). A **Random Forest Regressor** was trained to predict weight based on price, preserving the natural price-weight relationship better than a global mean.

B. Data Cleaning

- **Standardization:** Consolidated inconsistent labels in Item_Fat_Content (LF, low fat to Low Fat).
- **Logical Correction:** Created a Non-Edible category for "Non-Consumable" items (e.g., household goods) to remove logical inconsistencies where they had fat content.

3. Feature Engineering

Feature extraction focused on capturing store maturity and broad product categories:

- **Outlet_Years:** Transformed Establishment_Year into an age feature (2013 - Year) to reflect customer base maturity.
- **Item_Category:** Parsed Item_Identifier to create three broad buckets: **Food**, **Drinks**, and **Non-Consumables**.
- **Encoding:** Applied One-Hot Encoding to categorical variables. Crucially, Outlet_Identifier was retained to allow the model to learn the intrinsic baseline performance of specific high-volume stores (e.g., OUT027).

4. Modeling Strategy & Evolution

Phase 1: Baseline Gradient Boosting (XGBoost)

- **Validation:** Established a **5-Fold Cross-Validation** framework to ensure stability, achieving a baseline RMSE of ~1081.
- **Tuning:** Utilized RandomizedSearchCV to optimize n_estimators, max_depth, and learning_rate. A lower learning rate (0.05) with moderate depth (4-5) yielded the best generalization.

Phase 2: Multi-Model Ensemble

To reduce variance and overfitting, I introduced diverse gradient boosting implementations:

- **Models:** Added **LightGBM** (for leaf-wise growth speed) and **CatBoost** (for superior categorical handling).
- **Strategy:** Combined predictions using a weighted average (**40% XGBoost, 30% LightGBM, 30% CatBoost**).

5. Conclusion

The experimentation process highlighted that **Feature Engineering** (specifically handling Item_Fat_Content and Outlet_Size) and **Segmentation** were the biggest drivers of performance. While the Ensemble model provided robustness, the Segmentation strategy offered the best handling of outliers (e.g., Grocery Stores).

Final Recommendation: For the leaderboard submission, the **Weighted Ensemble** provides the safest, most robust score, while the **Segmented XGBoost** approach serves as a high-potential experimental submission to capture specific store nuances.