BigMart Sales Prediction: Approach & Experimentation

Initial Exploration & Problem Understanding

My first step was understanding the data through exploratory analysis. I noticed several issues: ~18% of Item_Weight values were missing, ~28% of Outlet_Size values were missing, and some Item_Visibility values were suspiciously 0. The target variable (Item_Outlet_Sales) had a right-skewed distribution with values ranging from ~0 to ~13K.

Data Cleaning & Preprocessing

I addressed several data quality issues:

- Standardized Item_Fat_Content categories (consolidating 'LF', 'low fat' into 'Low Fat' and 'reg' into 'Regular')
- Imputed missing Item_Weight using average weight per Item_Identifier
- Filled missing Outlet_Size values based on most common size for each Outlet_Type
- Replaced 0 visibility values with mean visibility for that Item_Type (as 0 visibility is unrealistic)

Feature Engineering

This was a crucial part of my approach. I created multiple features to capture different aspects:

- Outlet_Age from Outlet_Establishment_Year (2013 - Year)
- Item_Type_Category from Item_Identifier (first 2 chars - FD/Food, DR/Drinks, NC/Non-Consumable)
- Item_MRP_Category based on price distribution analysis (Low/Medium/High/Very High)
- Price_Segment using quantile binning (10 segments)

- Item_Visibility_Ratio relative to mean visibility per Item_Type
- Item_MRP_Percentile within each Item_Type
- Log_Item_MRP and Item_MRP_Scaled transformations
- Store_Item_Count and Item_Type_Ratio to capture store assortment patterns
- Outlet_Quality_Score as a composite feature
- Various interaction terms between key variables

Modeling & Evaluation

I experimented with multiple modeling approaches, evaluated using RMSE on a 75/25 train/validation split:

1. **Baseline Models**: Simple mean prediction (baseline) and linear models (Linear, Ridge, Lasso regression)
2. **Tree-based Models**: Decision Tree, Random Forest, Gradient Boosting
3. **Advanced Gradient Boosting**: LightGBM, XGBoost
4. **Deep Learning**: A multi-layer neural network with batch normalization and dropout
5. **Ensemble Methods**: Both voting and stacking ensembles of top-performing models

Among individual models, XGBoost consistently performed best, followed closely by LightGBM. The neural network showed promise but required more tuning. My final solution used a weighted ensemble (XGBoost 50%, LightGBM 30%, Neural Net 20%).

Key Insights

- Item_MRP showed the strongest correlation with sales
- Outlet_Type was a critical predictor (Supermarket Type 3 had highest average sales)
- The store establishment year had less impact than expected
- Item visibility showed a weaker relationship with sales than anticipated

Future Improvements

There are several avenues for further enhancement:

- More granular feature engineering for Item_Identifier patterns
- Time-series aspects if seasonal data becomes available
- Advanced hyperparameter tuning via Bayesian optimization
- Deeper exploration of Neural Network architectures (RNNs, attention mechanisms)
- Additional external data sources (if available) like local demographics or competition data
- Ensemble diversity enhancement with additional model types
- More sophisticated handling of categorical variables using entity embeddings

The models developed should help BigMart understand key sales drivers and make informed business decisions.