Solving the BigMart Sales Prediction Problem

By Abhinav Mallick

1. Data Understanding and Cleaning:

- Combined train and test datasets for consistency in analysis and missing value treatment.
- Created a lookup table by grouping Item_Weight values according to Item_Identifier, ensuring that missing values were imputed using the weight of the corresponding item in the dataset.
- Filled missing Outlet_Size values using a CatBoost classification model trained only on correlated features.
- Corrected discrepancies in Item_Fat_Content and renamed a few columns for clarity.

2. Exploratory Data Analysis (EDA) and Feature Engineering:

- Univariate & Multivariate Analysis:
  - Examined distributions of numerical columns across all datasets (combined/train/test) to check if any significant differences exist amongst them.
  - Analyzed num feature distribution to check significant differences between datasets (combined/train/test)
  - Identified outliers and evaluated statistical characteristics of num. columns to inform transformations.
  - Found strong correlation between Item, Item_MRP, Outlet, Outlet_Type & Sales.
  - Plotted scatterplots between sales and numerical features with hue of Outlet_Type that displayed distinctive clustering effect suggesting it may be an influential feature as compared to other cat. features
- Item Distribution Analysis: Identified Items with fewer observations present in training dataset as compared to the test data suggesting that we may need to focus on these particular Items more during modelling.
- Feature Engineering: Created three new features out of the significant features inferred from the EDA such as Mean_Sales per Outlet_Type/Location_Type and Mean_MRP per Outlet_Type.
- Transformations: Applied Yeo-Johnson transformation and RobustScaler to normalize the numerical distributions and handle outliers. And One-Hot Encoded the categorical variables.

3. Model Building & Feature Selection:

- Baseline Models: Chose Linear Regression with Polynomial features allowing interaction between features to get benchmark scores and improved on it by adding regularization to reduce overfitting. Ridge Regression provided the best results followed by Lasso and then ElasticNet.
- Decision Tree Models: Trained diverse DTs including GradientBoostRegressor, XGBoost, LightGBM, ExtraTreesRegressor & Catboost and compared the results with minimum & maximum feature sets.
- Multi-Layer Perceptron: Trained a neural network and compared performance with min. & max. feature sets.
- Stacking Ensemble: Combined multiple models' predictions together with ElasticNet as the Meta model in order to leverage the advantages of each base models' performance together to obtain the final predictions. Used RandomSearchCV for hyperparameter tuning of each individual model.

4. Results:

- Individual DTs and the MLP neural net performed better when more features were added as inputs.
- Stacking Ensemble provided better results than individual DT models but started overfitting as more features were included. Therefore, the Stacking Ensemble with minimum features provided the best predictions overall and CatBoost & ETR had the highest contributions within it.
- The highest correlated features with Sales, also had the highest feature importance in all models.
- *Achieved a rank of #55 and a score of ~1141.87 (as of 10/2/25) on the public test dataset.*
- Criticisms:
  - *Separate approaches could have been explored for Items with less training data* instead of predicting it with the global prediction model that captured more general patterns across *all items*.
  - Advanced categorical encoding techniques could have been tested.
  - Models were evaluated with either extreme feature inclusion or exclusion. Therefore, more rigorous in-the-middle combinations could have been tested to utilize each model to its full potential.