**Big Mart Sales Prediction: Project Approach and Experimentation**

**Objective:** To accurately predict the sales of various products across different Big Mart outlets, thereby providing actionable insights to optimize inventory and marketing strategies.

**1. The Thought Process: Strategy & Problem Framing**

The core problem is a **regression task**: predicting a continuous target variable (**Item\_Outlet\_Sales**). The strategy centers on leveraging machine learning to model the complex relationships between item characteristics, store properties, and historical sales.

|  |  |  |
| --- | --- | --- |
| **Phase** | **Core Goal** | **Rationale** |
| Understand | Analyze the data (items, outlets, sales). | Identify data types, missing values, and initial feature interactions (e.g., higher sales in bigger stores). |
| Transform | Engineer features and handle dirty data. | ML models require clean, numerical data; creating new features Item\_type\_Combined (Food, Drinks, NC) can capture more information than raw features. |
| Model | Select and train robust regression models. | Start with simple models (Linear Regression, Decision Tree) as baselines, then progress to advanced ensemble models (Gradient Boosting) for higher accuracy. |
| Evaluate | Measure performance using appropriate metrics. | RMSE (Root Mean Squared Error) is the primary metric, as it penalizes large errors heavily and is in the same units as the target variable (sales). |

**2. Experimentation Steps: The Iterative Process**

The project followed an iterative process, starting with a baseline model and progressively improving performance through data cleaning, feature engineering, and model tuning.

**A. Data Preprocessing and Baseline**

1. **Exploratory Data Analysis (EDA):**
   * **Univariate Analysis:** Checked distributions of Item\_Weight and Item\_Outlet\_Sales.
   * **Bivariate Analysis:** Noticed Outlet\_Size and Outlet\_Type correlate strongly with sales.
   * **Initial Findings:** Identified significant missing values in Item\_Weight and Outlet\_Size.
2. **Handling Missing Data:**
   * **Item\_Weight:** Imputed missing values using the **mean weight** *for that specific Item\_Identifier*. If the item was new, used the overall mean.
   * **Outlet\_Size:** Imputed based on the mode in the Outlet\_Type. For example, if a store was in Tier 1 city and Supermarket Type 1, mode was calculated for that group and imputed for missing Outlet\_Size.
3. **Fixing Zero Visibility:**

EDA reveals that minimum value for Item\_Visibility is 0 which makes no logical sense for any item. Hence we need to impute it with the mean Item\_Visibility of the Item\_Group/Outlet

1. **Baseline Model:**
   * **Model:** Linear Regression or Decision Tree
   * **Result:** Established a baseline RMSE, around 1200-1300, to measure future improvements against.

**B. Feature Engineering**

|  |  |  |
| --- | --- | --- |
| **New Feature** | **Creation Logic** | **Rationale** |
| Item\_Type\_Combined | Grouped 16 Item\_Type categories into 3 major groups: Food, Drinks, and Non-Consumable. | Reduced high cardinality and captured inherent relationship (e.g., all food items behave similarly). |
| Outlet\_Age | Calculated by subtracting the Outlet\_Establishment\_Year from the current year. | Older stores often have a more established customer base. |
| Item\_Fat\_Content | Consolidated misspellings (low fat, LF) into uniform categories (Low Fat, Regular). | Ensured consistency for categorical encoding. |

**C. Data Transformation and Encoding**

1. **Target Variable Transformation:** Applied a **log transformation** () to the Item\_Outlet\_Sales target variable.
   * **Reason:** The sales distribution was right-skewed; log transformation normalizes it, which often improves the performance of linear models.
2. **Categorical Encoding:** Converted all processed categorical variables (Outlet\_Type, Item\_Type\_Combined, etc.) into numerical features using **One-Hot Encoding**.

**D. Model Experimentation and Tuning**

A systematic, incremental approach was used to find the best-performing model.

1. **Simple Models:** Tested and to handle multicollinearity caused by One-Hot Encoding.
2. **Ensemble Models:**
   * **Random Forest:** A robust first ensemble test.
   * **Gradient Boosting Machine (GBM) / XGBoost:** The best-performing class of models for this type of structured data.
3. **Hyperparameter Tuning:** Performed **Cross-Validated Grid Search** (GridSearchCV) on the best model to optimize key parameters (e.g., max\_depth, n\_estimators, learning\_rate).
4. **Final Selection:** The model with tuned parameters consistently yielded the lowest RMSE, achieving a final score of approximately (a significant improvement over the baseline).

|  |  |  |
| --- | --- | --- |
| **Experimentation Step** | **Impact on RMSE** | **Key Takeaway** |
| Baseline Model | ∼1250 | Established floor baseline performance. |
| Feature Engineering (Age, Groups) | ∼1180 | Domain knowledge is key for performance gains. |
| XGBoost (Tuned) | ∼1100 | Optimally tuned XGBoost models excel on structured data. |