Approach Note — BigMart Sales Prediction

Objective:
Predict 2013 sales for 1559 products across 10 BigMart outlets using historical sales data and outlet/product attributes. The aim is to understand key drivers of sales and deliver an accurate, leakage-safe predictive model.

---

1. Understanding the Problem & Data

- **Data Scope:** 1559 products, 10 outlets, attributes like product weight, visibility, MRP, outlet size, location, and type.
- **Challenge:** Missing values due to reporting gaps; potential target leakage if store-level sales statistics are improperly engineered.
- **Target Variable:** Item_Outlet_Sales (continuous).

---

2. Data Preprocessing & Feature Engineering

The data from train and test sets is combined for consistent transformations. Key steps:

1. **Label Normalization:** Standardized categories (Item_Fat_Content harmonization).
2. **Missing Value Treatment:**
   - Item_Weight: Mean per product → global mean.
   - Item_Visibility: Zero replaced with type-wise mean.
   - Outlet_Size: Mode by Outlet_Type.
3. **Derived Features:**
   - **Time-based:** Outlet_Years = 2013 – establishment year.
   - **Categorical Reduction:** Item_Type_Combined (Food / Drinks / Non-Consumable).
   - **Binning:** Item_MRP_Bins, Item_Visibility_Bins.
   - **Ratios:** Price_Per_Weight, Visibility_MRP_Ratio.
4. **Aggregations:** Store-level item counts, store average sales (carefully handled to avoid leakage).

5. **Encoding:**
    - Ordinal encoding for ordered categories.
    - Label encoding for nominal categories.
    - One-hot encoding for outlet IDs.
6. **Frequency Features:** Item occurrence counts.

---

3. Model Selection

Chosen algorithms balance interpretability and power:

- **Tree-based Boosting Models:**
    - XGBoost (reg:tweedie)
    - LightGBM (regression/Tweedie)
    - CatBoost (Tweedie / RMSE loss)
- **Optional:** Tweedie Generalized Linear Model (TweedieRegressor).

---

4. Hyperparameter Optimization

- **Framework:** Optuna with TPE Sampler.
- **Cross-validation:** 5-fold within each tuning trial to avoid overfitting.
- **Parameters Tuned:** Depth, learning rate, regularization terms, subsampling rates, Tweedie variance power, etc.
- **Caching:** Best parameters stored/reloaded from JSON to avoid redundant tuning.

---

5. Ensemble Strategy

Multi-Stage Approach:

1. **Base Models:** Trained in **K-Fold** fashion for multiple seeds (default: 5 seeds × 10 folds).
2. **Out-of-Fold (OOF) Predictions:** Captured for each model to train a meta-learner.

3. **Meta-Learner:** Ridge regression (or Linear) on raw & engineered meta-features (raw predictions, squared terms, interactions, row-wise stats).
4. **Selection:** Per-seed comparison of meta-learner RMSE vs uniform average → choose better for that seed.
5. **Final Prediction:** Average chosen per-seed predictions.

---

6. Evaluation Metric

- **RMSE (Root Mean Squared Error)** chosen for consistency with continuous sales prediction.
- Both **uniform averaging** and **meta-learning** compared to ensure no over-complication.

---

7. Experimentation Insights

- **Feature engineering** significantly boosts model stability.
- **Tweedie loss** in tree-based models handles skewed, positive sales distribution better than plain RMSE.
- **Multi-seed averaging** reduces variance and smooths random splits' effects.
- **Meta-learner** often outperforms simple averaging, but is only selected when RMSE improvement is consistent.

---

8. Final Deployment

- Saved submission file with Item_Identifier, Outlet_Identifier, and predicted Item_Outlet_Sales.
- Generated a detailed run report: per-model OOF RMSE, final method chosen, uniform vs meta RMSE.

Summary:
This pipeline builds a leakage-safe, multi-model stacked ensemble with robust feature engineering, advanced hyperparameter tuning, and variance-reduction via multi-seed

cross-validation—delivering strong predictive performance while preserving interpretability of key sales drivers.