**Solution Description**

In this solution, I implemented a **stacking ensemble approach** to predict sales for the BigMart dataset. The idea was to combine the strengths of tree-based models and neural networks, and then use a meta-model to learn the optimal way of blending their predictions.

**Preprocessing & Feature Engineering**

* Missing values for Item\_Weight were imputed using the median per Item\_Type.
* Missing values for Outlet\_Size were filled using the most frequent category (mode) within each Outlet\_Type.
* Inconsistent values in Item\_Fat\_Content were cleaned and unified.
* New features such as Outlet\_Age (how many years old the outlet was in 2013) and Item\_Category (first two letters of Item\_Identifier) were created.
* Categorical variables were encoded with LabelEncoder, and features were standardized for use with neural networks.

**Base Models**  
I trained a diverse set of models:

* RandomForestRegressor
* XGBoostRegressor
* LightGBMRegressor
* Two simple feed-forward neural networks, one using **Swish activation** and the other using **ELU activation**, with dropout regularization.

Each model was trained in a **5-fold cross-validation** setup to generate out-of-fold predictions, which were then used as inputs for the meta-model.

**Meta-Model (Ridge Regression)**  
The meta-model is a **Ridge Regression** trained on the out-of-fold predictions from the base models. Ridge is a simple but effective choice since it balances model weights while controlling overfitting. The final predictions are generated by applying the trained Ridge model to the averaged test predictions of the base models.

**Results**  
This approach gave me a stable **Leaderboard score of ~1149.**, placing me around rank ~680. While not the absolute lowest error, the model is consistent, interpretable, and demonstrates the value of blending diverse learners in a stacking ensemble.

A screenshot of a computer

AI-generated content may be incorrect.