Explanation - ABB Senior Data Scientist Coding Assignment

- The approach to the problem starts with an emphasis on feature engineering and data cleaning. This way, the effort required to select a good model and fine-tuning it to achieve an amazing final score will be minimal, simply due to the fact that all the dirty work has been done during feature engineering.
- Another area of focus is to eliminate any zeroes in the training set. Zeros add unnecessary complications to calculations, as dividing or multiplying anything by zero does not achieve any useful result.
- I also want to emphasize the usefulness of removing outliers. Now, some outliers are useful as they capture important information of how a feature can extend in values, but for the most part, outliers only hamper the learning and generalizing ability of a model, which should be the job of a machine learning model.
- This is an unconventional choice, but I've used a for loop that trains a model using a combination of features and removes the features that are not helping to improve the root mean squared error (rmse) score. This is the score used to evaluate other contestants in the hackathon as well.

Approach

1. **Data Cleaning:** I started by finding irregularities in the training set for categorical columns, and I found the column **Item_Fat_Content** contains duplicated elements such as 'LF' and 'low fat' which belongs to the main category 'Low Fat'. Similarly, 'reg' belongs to the main category 'Regular'.
2. **Data Cleaning (Imputing):** Since the column **Item_Weight** contains missing values, I impute the median values of the column to the missing values. Similarly, I impute the column **Outlet_Size** with the most frequent values (mode) of the column. Mode and median work better than the mean since they are not derived by using outliers. Mean is sensitive to outliers. Therefore, using median and mode helps achieve a better final rmse score.
3. **Outlier Removal:** Based on the statistical distance from the main data distribution with respect to the target variable (**Item_Outlet_Sales**) from the scatterplots, I've removed them and checked if the rmse score improves. I try looking for natural patterns in the scatterplot after removing the outliers.
4. **Feature Engineering:** I've added 3 interaction terms that help capture the essence of combined features that cannot be represented individually. It's a proven technique to help improve the performance of a model.
5. **Train and Test sets:** I've split the training set into a very standard 80-20 ratio with a seed in place so results can be reproduced. Instead of using *random_state = 42* (from the documentation), I've used *random_state = 2024* cause it's a more recent year.
6. **Modelling:** I've used XGBoost and Catboost models since they are very effective in achieving results for competitions like this one. Tree-based models work brilliantly at capturing non-linear relationships and feature interactions that help with the final prediction being very good.
7. **Evaluation:** Same as what they use in the competition to assess the final rank. Root Mean Squared Error (rmse). The mention of r2 score is also in the Jupyter Notebook, but that metric is of no significance in this competition. I've also used cross-validation scores to check if the model has any room for improvement.
8. **Fine-tuning:** I've used the optuna library for fine-tuning cause it takes very little time to finish fine-tuning and find the best parameters. Although the training set is quite small, the use of grid-search can also be made use of since it does not take much time to train with a small dataset.

Result: Achieved rmse = 1016.436, outperforming the current #1 leaderboard score (1127.717) by ~110 points.