HDL Cholesterol Prediction Report

Goal and Metric

The goal of this project is to predict HDL cholesterol (LBDHDD_outcome) for each row in the provided test dataset. Submissions are evaluated using Root Mean Squared Error (RMSE) on the hidden test labels.

Data and Preprocessing

The training dataset (train.csv) contains the target column LBDHDD_outcome along with the feature variables. The test dataset (test.csv) contains the same feature variables but does not include the target. I defined y = LBDHDD_outcome and used all remaining columns as features X. If an index-style column such as Unnamed: 0 was present, it was removed. I also removed 3 useless variables with the same unique responses. Some visualizations that help with EDA were also included in the notebook.

Modeling Approach

I trained several regression models to establish a performance baseline and select a final model:

1. Mean baseline model (predicts the mean of y_train for every observation)
2. Ridge Regression (linear model with L2 regularization and feature scaling)
3. Random Forest Regressor
4. XGBoost Regressor
5. CatBoost Regressor

I first compared models using an 80/20 train-validation split to understand relative performance. Based on this comparison, XGBoost and CatBoost were the strongest candidates, so I evaluated them more rigorously using 5-fold cross-validation.

Validation Strategy and Final Model Selection

To select the final model, I used 5-fold cross-validation with shuffling (random_state = 42) and RMSE scoring. The cross-validation results were:

- XGBoost: mean RMSE = 4.754 (std = 0.219)
- CatBoost: mean RMSE = 4.732 (std = 0.184)

CatBoost achieved the lowest mean RMSE and slightly lower variability, so it was selected as the final model.

Final Model and Submission File

The final model was a CatBoostRegressor with the following settings: iterations = 1000, learning_rate = 0.03, depth = 6, l2_leaf_reg = 3, loss_function = RMSE, random_seed = 42. I trained this model on all available training data (all rows in train.csv), then generated predictions for each row in test.csv in the same row order. I saved the predictions to a file named pred.csv with exactly one column named pred, and the number of prediction rows matches the number of rows in the test dataset.