1 Feature Engineering and Data Preprocessing

The data was first checked if there were any missing values. Missing values leads to a biased analysis, distorted feature relationships, and reduced statistical power.

Each feature was checked for correlation with the target value and removed if it correlated value resulted in a value less than 0.1. This removes irrelevant features to simplify the model, prevent overfitting, and increase interpretability and computational efficiency.

The data was centered and normalized. This scales the data to be more comparable, algorithms to converge more efficiently, and avoid dominance of certain features with a larger scale that the other values.

The variance of the data was calculated, but no features were removed. The values of were above all similar in value, so no low values were able to be picked out reasonably.

2 Machine Learning Model/Algorithm

The XGBRegressor algorithm was chosen since it had the lowest RMSE value and that is the metric being used to demonstrate model performance. Each model was trained with the given data and returned an RMSE. The XGBRegressor had lowest value.

3 Hyperparameter Optimization

Using grid search, the hyperparameters chosen were a learning rate of 0.1 with 100 estimators.

First, Identify the hyperparameters of the algorithm that you want to tune. Specify the range or set of values that you want to explore for each hyperparameter. For XGBoost, we used learning rate, maximum depth, and number of estimators.

Determine the evaluation metric that you want to use to compare different combinations of hyperparameters. We used RMSE.

Utilize a grid search algorithm (GridSearchCV‘ in scikit-learn) that systematically searches through the defined hyperparameter grid. It trains and evaluates the model for each combination of hyperparameters using cross-validation.

Once the grid search is complete, identify the combination of hyperparameters that yielded the best performance based on the chosen scoring metric. This combination is the optimal set of hyperparameters.

With the best hyperparameters determined, train the final model using the entire training dataset and these optimized hyperparameters.