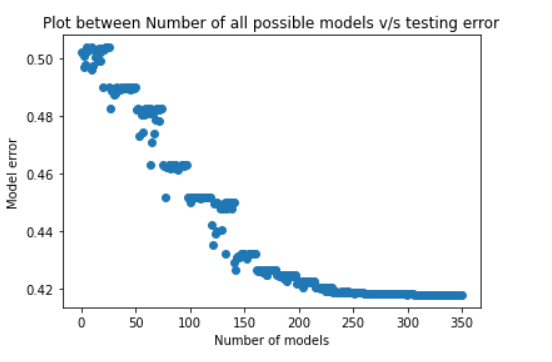
**GREEDY FORWARD FEATURE SELECTION**

A greedy feature selection is the one in which an algorithm will either select the best features one by one (forward selection) or remove the worst feature one by one (backward selection).

Feature selection is one of the key steps in training the most optimal model in order to achieve higher computational efficiency while training the model, and also reduce the the generalization error of the model by removing irrelevant features or noise

Greedy forward selection is a popular technique for feature subset selection. The main advantage of this approach is its simplicity and generally low run-time in small feature spaces.



In the greedy forward feature selection, the plot between all possible methods vs testing error is shown above. Here the model error keeps decreasing with the increasing number of models.

**Steps for greedy forward feature selection:**

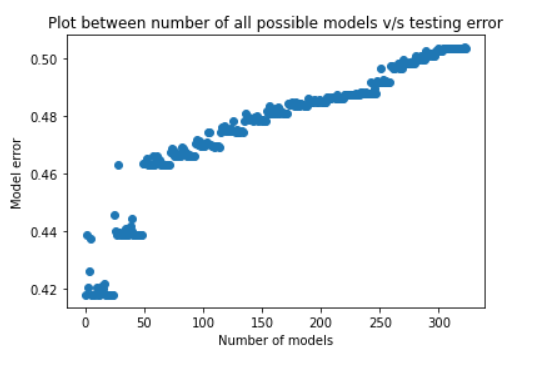
* Train n models using each feature(n) individually and check the performance.
* Choose the variable which gives the best performance
* Repeat the process and add one variable at a time
* Variable producing the highest improvement is retained
* Repeat the entire process until there is no significant improvement in the model’s performance

**GREEDY BACKWARD FEATURE SELECTION**

The sequential backward selection approach seeks to increase computational efficiency and decrease generalization error by decreasing the dimensionality of the initial feature subspace from N to K-features with a minimum drop in model performance.

To get to the list of K-features, the main concept is to successively delete features from the given features list, which consists of N features. The component that results in the least performance loss is removed at each level.

The algorithm used to find features is a combinatorial search, where a subset of features is chosen from a combination and given a score before being compared to other subsets.



In the greedy backward feature selection, the plot between all possible methods vs testing error is shown above. Here the model error keeps increasing with the increasing number of models.