Finding the best Random Forest Model.

Random Forest models have several hyperparameters that can be adjusted to affect the accuracy of the model being trained (get sources). We used the random forest regressor in scikit-learn (0.20.3). Determining what values of these hyperparameters optimizes the model results can be difficult. Hyperparameters include the number of trees, the maximum depth of each tree, and the maximum depth of each tree. Other parameters influence how the trees are split at each decision node. The hyperparameters we considered are summarized in table (HPT)

|  |  |  |
| --- | --- | --- |
| Hyperparameter |  | Description |
| n\_estimators | e | Number of trees in the forest. |
| max\_depth | md | Maximum depth of each tree. |
| min\_samples\_split | mss | Number of samples required to split a node. |
| min\_samples\_leaf | msl | Number of samples required at each leaf. |
| max\_featuers | mf | Maximum number of features considered at each split. |
| max\_leaf\_nodes | mln | Maximum leaf nodes of each tree. |

HPT summary of scikit learn rf parameters

In order to determine which combination of hyperparameters creates the best random forest model we looked at various combinations of parameters. The values of each hyperparameter considered are shown in table VCT. We also look at three percentages of the total data as the training data. The combinations of these hyperparameters were examined in a brute force manor. Each variation in a hyperparameter was combined with each possibility of other hyperparameters. This means there were 5,184 models to train.

|  |  |
| --- | --- |
| Hyperparameter | Values used |
| n\_estimators | 10, 50, 100 |
| max\_depth | 12, 25, 60, 100 |
| min\_samples\_split | 2, 5, 10 |
| min\_samples\_leaf | 1, 2, 4, 8 |
| max\_featuers | AUTO, SQRT, LOG2 |
| max\_leaf\_nodes | 1,000, 5,000, 10,000, 50,000 |
| percent of training data | 25, 50, 75 |

VCT Values considered

The work of training these 5,184 possible models was accomplished using four computers with varying capabilities, and existing workloads. The computers were running either mac OSX or ubuntu. Table CPUT Summarizes the computers used. The work being done by each computer was tracked via a CSV file located in a git repository. The list initially contained the parameters to be used, a column to track the progress of the training, and empty columns for the desired statistics to be collected. An example of this format is included in the appendix. As the computers completed each of the model trails they checked in with the git repository. The csv was updated with the statistics and other metadata for the trained random forest, and the next set of random forest parameters was acquired. Finally, any merge conflicts in the csv file were corrected by removing any duplicate rows, keeping completed rows over unrun or in progress rows if the names were the same, and deleting any of the auto generated merge conflict text. The csv file was then resynced with the remote repository and the next Random Forest model was run.

The bulk training of the random forest models was run over the period of about a month, and a total training time of 125 days 17:38:38.230825 was completed.

The order of training the models took the number of estimators into account before any other hyperparameter. All 10 and 50 estimator models were run during the testing period while ~83% of all 100 estimator models were completed, due to the brute force training being terminated before full set of parameters could be run.

The best model was chosen by sorting the models by the mean of their absolute differences from the original model.

# note to self this is not the correct formula