Finding A Baseline Random Forest model

Random Forest models have several hyperparameters that can be adjusted to affect the accuracy of the model being trained. Determining what values of these hyperparameters optimizes the model results can be difficult. Scikit-learn provides access to several hyperparameters. These 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 Scikit-learn hyperparameters considered are summarized in Table 1.

Table 1: Summary of Random Forest Hyperparameters

|  |  |  |
| --- | --- | --- |
| 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. |

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 2. Also included in the table are the three percentages of the training data used to train the models. The combinations of these hyperparameters were examined in a brute force manor. Each variation of a hyperparameter was combined with each possibility of other the hyperparameters. With this method there were 5,184 models to train. {talk about the n\_E = 150 } ?

Table 2: Hyperparameter Values

|  |  |
| --- | --- |
| Hyperparameter | Values |
| 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 |

Training the 5,184 possible models was accomplished using four systems with varying capabilities, and existing workloads. They ran either Mac OSX, Debian, or Ubuntu. Table 3 summarizes the systems used, and presents the time each ran the best model in. The Random Forest regressors were configured in a way that four parallel jobs were used to build trees in the forest. Statistics were collected on each of the models trained, but the models themselves were not saved at this time. The statistics recorded were the time to train each model, the time to predict the values for a given year, the mean difference and its variance, the mean absolute difference and its variance, and the median.

The work done by each computer was tracked via a csv file located in a git repository. This 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. After each model was trained, the csv data was updated with the model’s statistics, and checked in with the git repository. During this step next set of Random Forest parameters was also acquired.

A problem arose with merge conflicts in the csv file after the first few days of running, and a step to correct this had to be developed. Merge conflicts were corrected by removing any duplicate rows, keeping completed rows over unrun or in progress rows with the same parameters, and deleting any of the auto generated merge conflict text. This step occurred right before the final resync that push new stats to the remote repository. This updated utility was deployed to only one of the systems, and the problem was corrected.

Table 3: Training Systems Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Computer Name | OS | CPU | Memory | Time to run best model |
| Bristlecone | Ubuntu 18.04 | 3.2 GHz  Intel Core i7 (12) | 16 GB  1333 MHz  DDR3 |  |
| Finwhale | Debian 10 | 3.2 GHz  Intel Core i7 (12) | 16 GB  1333 MHz  DDR3 |  |
| Ocotal | Ubuntu 18.04 | 2.4 GHz  Intel Xeon (16) | 24 GB  1067 MHZ |  |
| Chickadee | Mac OSX 10.12 | 2.5 GHz Intel Core i7 (8) | 16 GB  1600 MHz DDR3 |  |

The bulk training of the random forest models was run over the period of about a month before the training programs were externally stopped. A combined training time of 125 days 17:38:38 was completed during this period. The order for training took the number of estimators into account before any other hyperparameter. Therefor all 10 and 50 estimator models were run during the testing period while only ~83% of all 100 estimator models were completed before the training was stopped.. Figures 1 and 2 show how the model training times vary color coded by number of estimators.

The accuracy of the models was measured two main statistics the mean difference (MD) from the original model, and the mean absolute difference (MAD) from the original model. Both of these values have to be used because the MD values all stay close to zero due to many positive and negative values being used to calculate the mean. The Mean Absolute Difference corrects this by taking the absolute value of the differences before calculating the mean there for giving a measure of how far off the random forest model is predicting. Figures 1 and 2 show this phenomenon graphically. In figure 1 the points plotted all show all numbers near zero while figure 2 shows numbers that

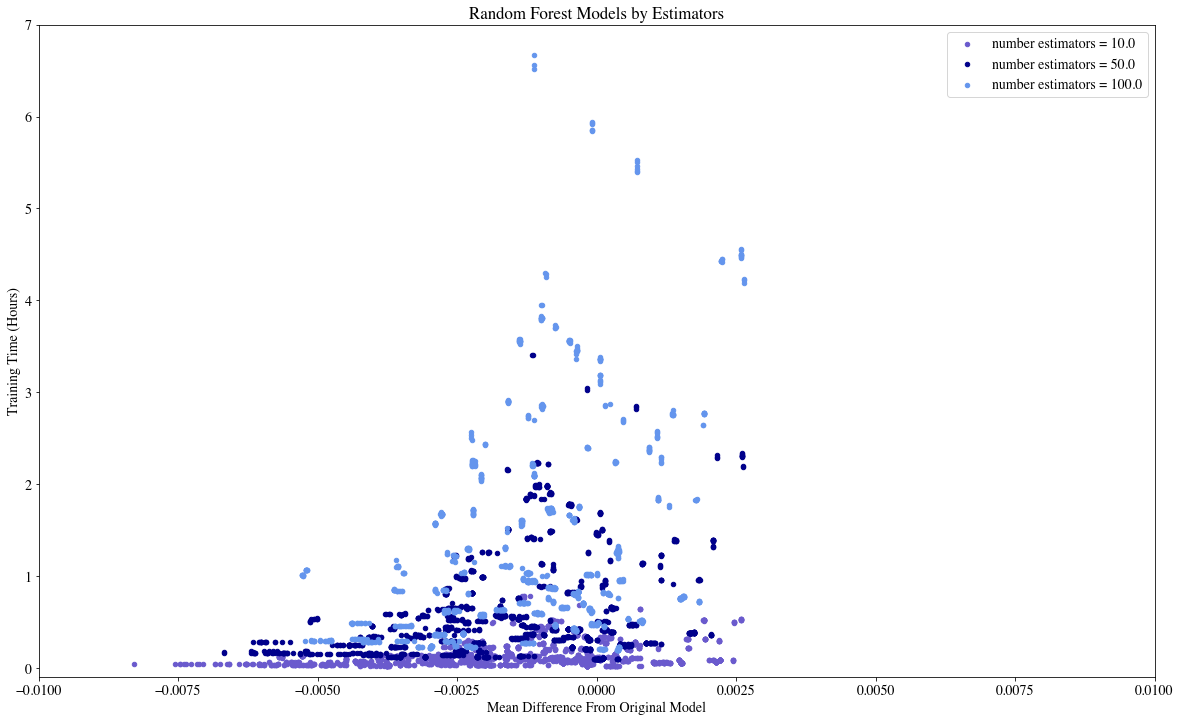


Figure 1: Mean Difference

A screen shot of a computer

Description automatically generated

Figure 2: Mean Absolute Difference

The best model was chosen by sorting the models by the mean of their absolute differences from the original model. This measure was used as it gives a better indication of how far off the Random Forest Model Predictions are. High accuracy model have low MAD values. A selection of top preforming models is presented in table (MADT). These models all had similar accuracy when applied to the study area for various years in the training period as shown in figure (FTOP). The ‘best’ model was selected because it had the lowest MAD value, other models in table (MADT) and figure (FTOP) were selected semi randomly. The hyperparameters from best model, now called the baseline Random Forest model, were used as the base of all the sensitivity analysis performed.