**Random Forest Classification**

(Basic description … advantage/ disadvantage)

**Methodology**

* Data handling

Data will be split to training (80 %) and testing (20%) set.

* Hyperparameter tuning
  + “mtry” parameter

It controls the number of randomly selected features that are used to determine the best split at each node. i.e. the level of fitting of the algothrim.

tuneRF() is used in tuning the parameter, ntree = 2000 is used for the tuning (higher the value, better the result, after testing the result become

Trial for different ntree value (higher the value, better the result)

ntree = 500

> model <- tuneRF(train[,1:6], train[,7] , mtryStart = 2, ntree= 500)

mtry = 2 OOB error = 3.97%

Searching left ...

mtry = 1 OOB error = 28.76%

-6.236364 0.05

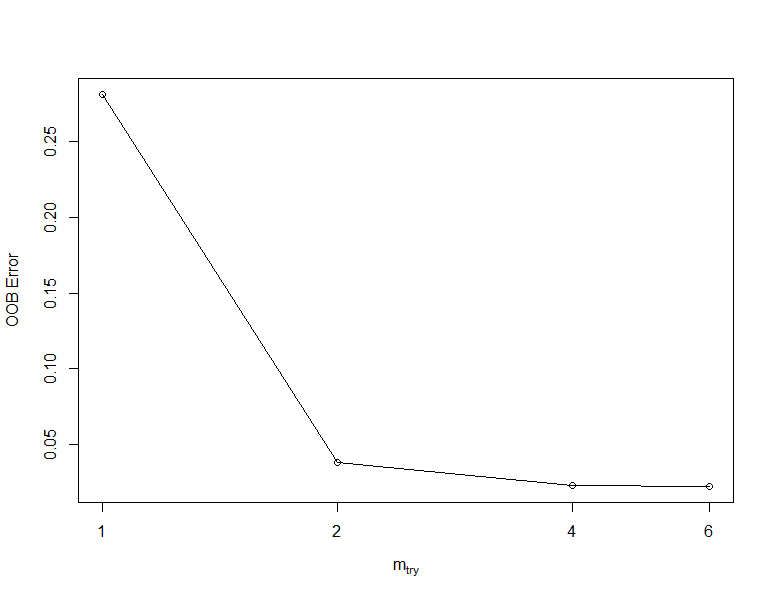
Searching right ...

mtry = 4 OOB error = 2.46%

0.3818182 0.05

mtry = 6 OOB error = 2.1%

0.1470588 0.05



ntree= 1000

> model <- tuneRF(train[,1:6], train[,7] , mtryStart = 2, ntree= 1000)

mtry = 2 OOB error = 3.83%

Searching left ...

mtry = 1 OOB error = 28.11%

-6.339623 0.05

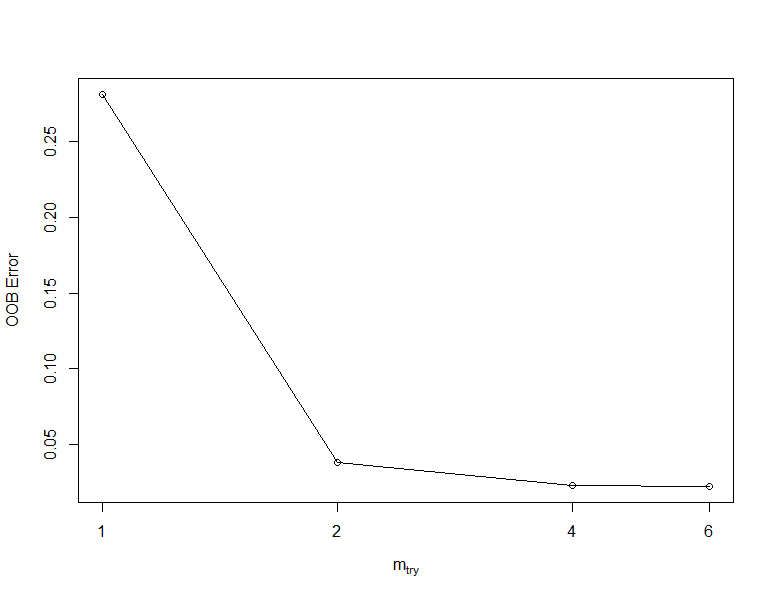
Searching right ...

mtry = 4 OOB error = 2.31%

0.3962264 0.05

mtry = 6 OOB error = 2.24%

0.03125 0.05



We see mtry = 6 still have the lowest OOB Error even in higher ntree number, so we pick it as the optimal value.

* + “nodesize” parameter

This parameter specifies the minimum size of leave nodes in the tree. Smaller “nodesize” value results higher predictive accuracy, but cause overfitting.

The different node size will be tested and AUC will calculated to find the optimal model.

(…. TBC)

**Detail description of the optimal model**

(…. TBC)