**Build a random forest for the ‘iris’ data.**

**Ans:**

> data("iris")

> View(iris)

> summary(iris)

Sepal.Length Sepal.Width Petal.Length Petal.Width Species

Min. :4.300 Min. :2.000 Min. :1.000 Min. :0.100 setosa :50

1st Qu.:5.100 1st Qu.:2.800 1st Qu.:1.600 1st Qu.:0.300 versicolor:50

Median :5.800 Median :3.000 Median :4.350 Median :1.300 virginica :50

Mean :5.843 Mean :3.057 Mean :3.758 Mean :1.199

3rd Qu.:6.400 3rd Qu.:3.300 3rd Qu.:5.100 3rd Qu.:1.800

Max. :7.900 Max. :4.400 Max. :6.900 Max. :2.500

> str(iris)

'data.frame': 150 obs. of 5 variables:

$ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...

$ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...

$ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...

$ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...

$ Species : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...

> attach(iris)

**Splitting Data into train and test**

> s <- createDataPartition(Species,p=0.7,list = F)

> train\_iris <-iris[s,]

> test\_iris <- iris[-s,]

**Building random forest model**

> iris\_model <- randomForest(Species~.,data = train\_iris,importance = T)

> iris\_model

Call:

randomForest(formula = Species ~ ., data = train\_iris, importance = T)

Type of random forest: classification

Number of trees: 500

No. of variables tried at each split: 2

OOB estimate of error rate: 2.86%

Confusion matrix:

setosa versicolor virginica class.error

setosa 35 0 0 0.00000000

versicolor 0 34 1 0.02857143

virginica 0 2 33 0.05714286

**Prediction and accuracy based on train data**

> pred\_tain\_iris <-predict(iris\_model,train\_iris)

> mean(pred\_tain\_iris==train\_iris$Species)

[1] 1

> confusionMatrix(pred\_tain\_iris,train\_iris$Species)

Confusion Matrix and Statistics

Reference

Prediction setosa versicolor virginica

setosa 35 0 0

versicolor 0 35 0

virginica 0 0 35

Overall Statistics

Accuracy : 1

95% CI : (0.9655, 1)

No Information Rate : 0.3333

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 1

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: setosa Class: versicolor Class: virginica

Sensitivity 1.0000 1.0000 1.0000

Specificity 1.0000 1.0000 1.0000

Pos Pred Value 1.0000 1.0000 1.0000

Neg Pred Value 1.0000 1.0000 1.0000

Prevalence 0.3333 0.3333 0.3333

Detection Rate 0.3333 0.3333 0.3333

Detection Prevalence 0.3333 0.3333 0.3333

Balanced Accuracy 1.0000 1.0000 1.0000

**Prediction and accuracy based on test data**

> pred\_test\_iris <- predict(iris\_model,test\_iris)

> mean(pred\_test\_iris==test\_iris$Species)

[1] 0.9333333

> confusionMatrix(pred\_test\_iris,test\_iris$Species)

Confusion Matrix and Statistics

Reference

Prediction setosa versicolor virginica

setosa 15 0 0

versicolor 0 12 0

virginica 0 3 15

Overall Statistics

Accuracy : 0.9333

95% CI : (0.8173, 0.986)

No Information Rate : 0.3333

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.9

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: setosa Class: versicolor Class: virginica

Sensitivity 1.0000 0.8000 1.0000

Specificity 1.0000 1.0000 0.9000

Pos Pred Value 1.0000 1.0000 0.8333

Neg Pred Value 1.0000 0.9091 1.0000

Prevalence 0.3333 0.3333 0.3333

Detection Rate 0.3333 0.2667 0.3333

Detection Prevalence 0.3333 0.2667 0.4000

Balanced Accuracy 1.0000 0.9000 0.9500

**Variable importance**

> importance(iris\_model)

setosa versicolor virginica MeanDecreaseAccuracy MeanDecreaseGini

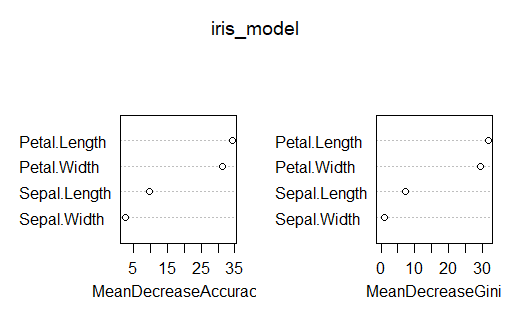
Sepal.Length 7.853311 6.305529 6.345820 9.858760 7.1404749

Sepal.Width 2.936403 2.040790 -2.220313 2.516242 0.8959233

Petal.Length 22.244105 33.705700 24.500412 34.290497 31.7920852

Petal.Width 22.842531 29.625870 20.977781 31.281465 29.4708119

> varImpPlot(iris\_model)



**So From Above Observations, we can say Petal length is the most significant variable**

**Bagging**

> accboost <- c()

>for(i in 3:10){

set.seed(100)

boost <- createDataPartition(Species,p=0.80,list = F)

train\_boost <- iris[boost,]

test\_boost <- iris[-boost,]

boost\_model <- randomForest(Species~.,data = train\_boost,mtry=i)

pred\_boost <- predict(boost\_model,test\_boost)

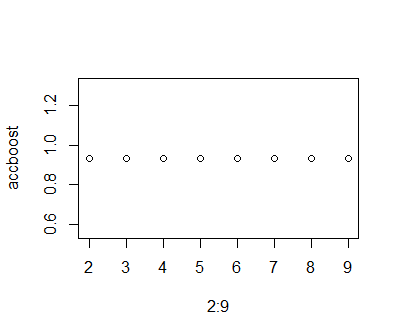
accboost[i-2]=mean(pred\_boost==test\_boost$Species)

}

> accboost

[1] 0.9333333 0.9333333 0.9333333 0.9333333 0.9333333 0.9333333 0.9333333 0.9333333

> plot(2:9,accboost)



**From above observation we can say, for different mtry accuracy is similar i.e. 0.9333333**

**Final model from mtry plot**

> finalmodel <- randomForest(Species~.,data = train\_iris,mtry=4)

> finalmodel

Call:

randomForest(formula = Species ~ ., data = train\_iris, mtry = 4)

Type of random forest: classification

Number of trees: 500

No. of variables tried at each split: 4

OOB estimate of error rate: 2.86%

Confusion matrix:

setosa versicolor virginica class.error

setosa 35 0 0 0.00000000

versicolor 0 34 1 0.02857143

virginica 0 2 33 0.05714286

> mean(predict(finalmodel,test\_iris)==test\_iris$Species)

[1] 0.9333333

**Our Final Model will have 93.33% Accuracy.**