Machine Learning

Ravi Kumar Tiwari

14 June 2016

```
library(caret)
library(rpart.plot)
library(rattle)
library(calibrate)
library(randomForest)
library(e1071)
library(class)
library(knitr)
library(party)
```

Decision Tree Example

Problem Description

Given a data set that contains some observation and corresponding class label, can a machine learning algorithm be trained to determine the class label of any data set (not necessarily the data that was used for training) from its observation

Solution using decision tree

```
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Create data partition

```
set.seed(100)
inTrain <- createDataPartition(iris$Species, p = 0.6, list = FALSE)
trainData <- iris[inTrain,]
testData <- iris[-inTrain,]
```

Build a decision tree model and use it for prediction on test data set

```
treeModel <- train(Species ~ ., data = trainData, method = "rpart")
preClass <- predict(treeModel, newdata = testData)
cMatrix <- confusionMatrix(preClass, testData$Species)
cMatrix$table
```

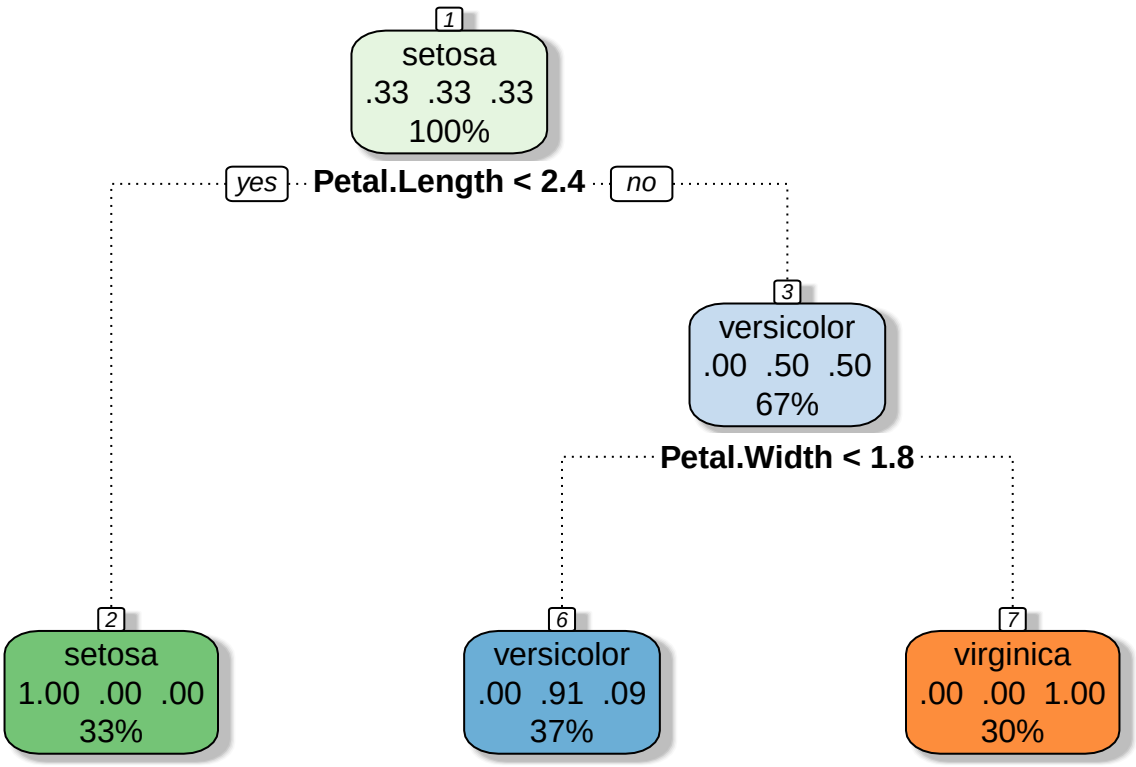
```
##             Reference
## Prediction   setosa versicolor virginica
##   setosa         20          0         0
##   versicolor      0         19         2
##   virginica       0          1        18
```

Look at what are the important variables

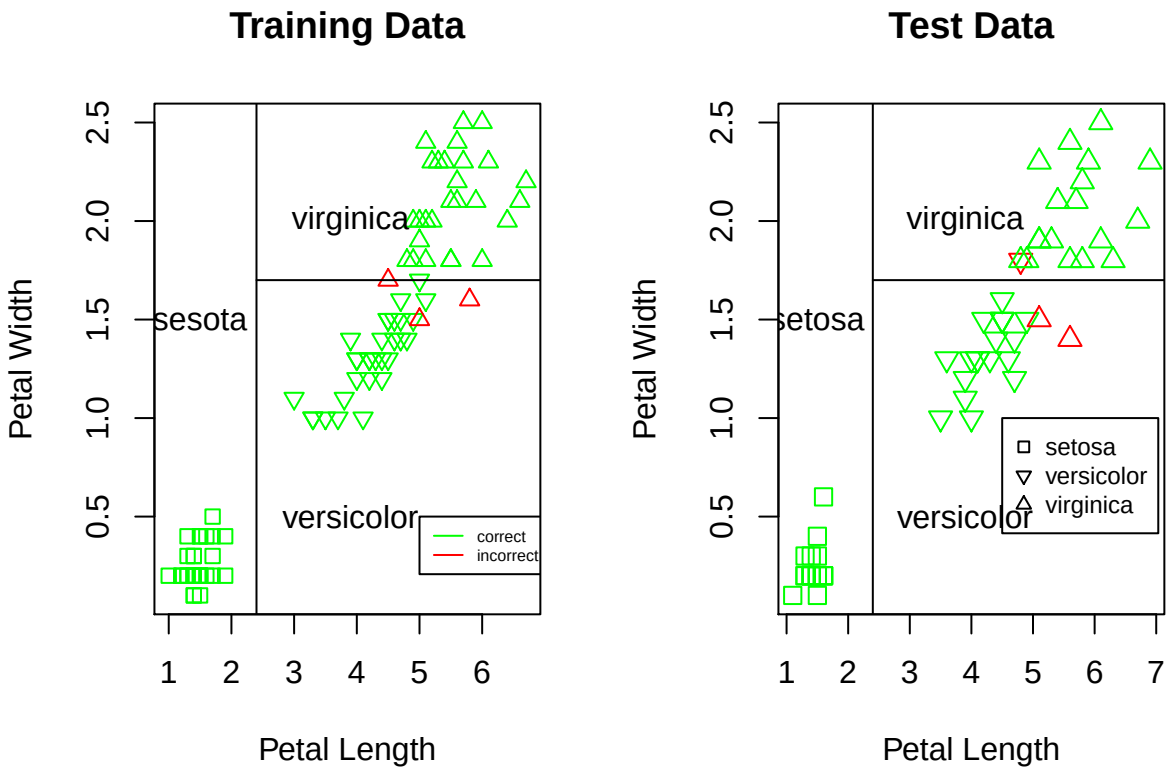
```
varImp(treeModel)
```

```
## rpart variable importance
##
##              Overall
## Petal.Width   100.00
## Petal.Length   89.53
## Sepal.Length   18.24
## Sepal.Width     0.00
```

Visualization of the decision tree

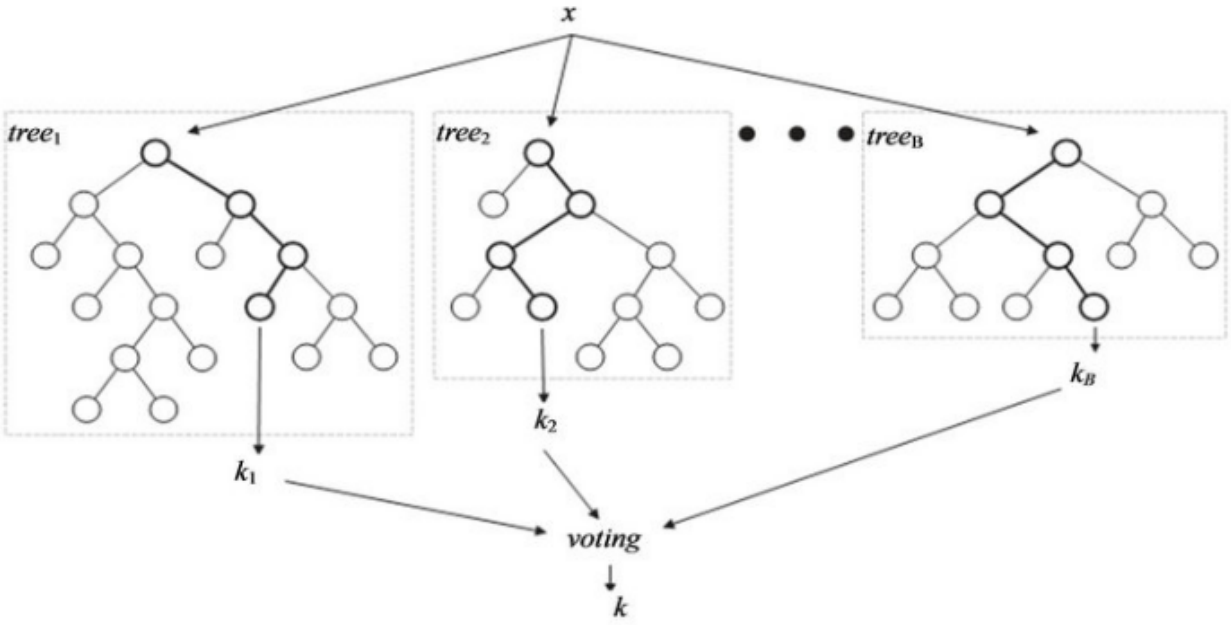


Visualization of the decision tree

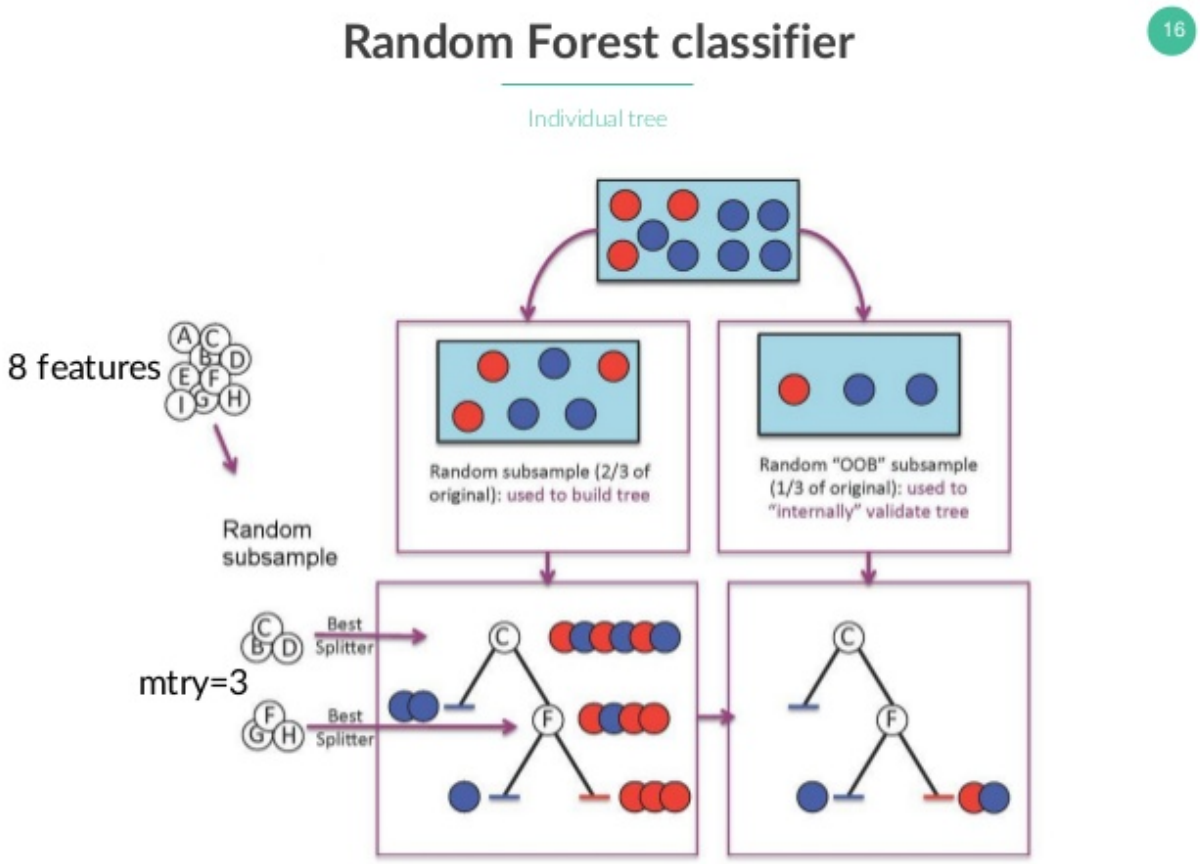


random Forest

ntree



mtry



random forest example

```
set.seed(100)
inTrain <- createDataPartition(iris[,5], p = 0.6, list=FALSE)
trainData <- iris[inTrain,]
testData <- iris[-inTrain, 1:4]
testClass <- iris[-inTrain, 5]

rfModel <- randomForest(Species ~ ., data=iris, mtry=3, ntree=10)
predClass <- predict(rfModel, newdata = testData)
table(predClass, testClass)
```

```
##             testClass
## predClass    setosa versicolor virginica
##   setosa         20          0         0
##   versicolor      0         20         0
##   virginica       0          0        20
```

knn2

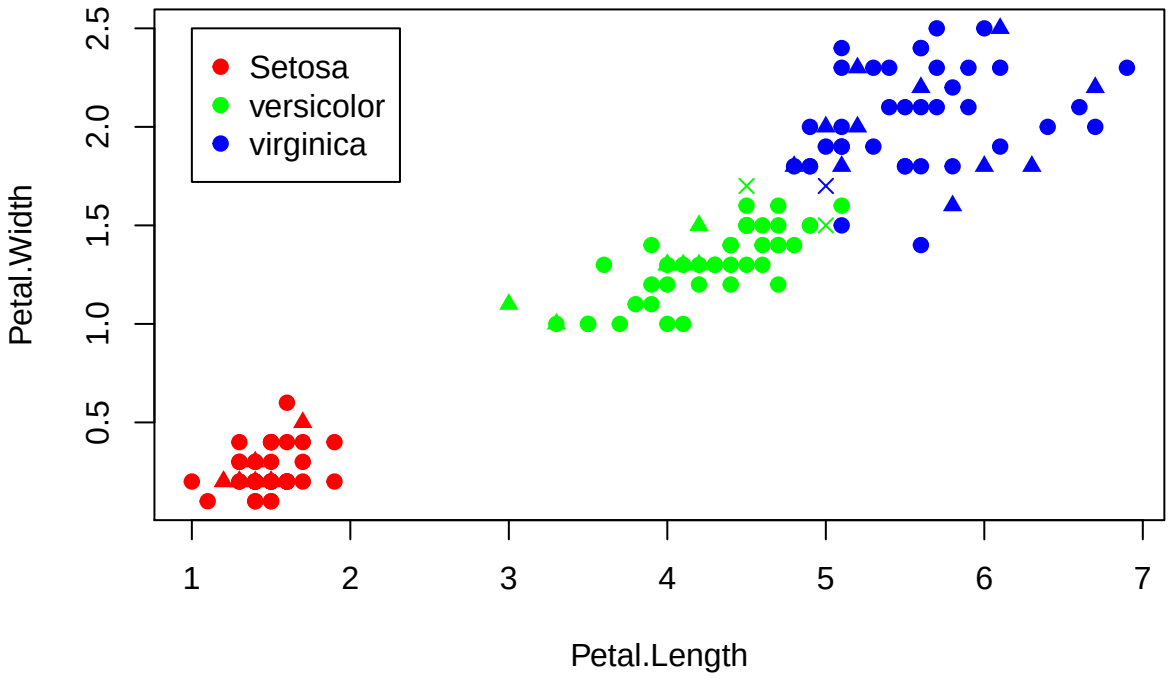
```
myIris <- iris[,3:5]
head(myIris)
```

```
##   Petal.Length Petal.Width Species
## 1          1.4         0.2  setosa
## 2          1.4         0.2  setosa
## 3          1.3         0.2  setosa
## 4          1.5         0.2  setosa
## 5          1.4         0.2  setosa
## 6          1.7         0.4  setosa
```

```
nI <- nrow(myIris)
ind <- sample(1:nI, 0.8*nI)
trainData <- myIris[ind, 1:2]
trainClass <- myIris[ind, 3]
testData <- myIris[-ind, 1:2]
testClass <- myIris[-ind, 3]
preClass <- knn(trainData, testData, cl = trainClass, k = 2)
table(preClass, testClass)
```

```
##             testClass
## preClass     setosa versicolor virginica
##   setosa          9          0         0
##   versicolor      0          7         2
##   virginica       0          1        11
```

```
color <- ifelse(trainClass=="setosa", "red", ifelse(trainClass=="versicolor", "green",
                "blue"))

plot(trainData$Petal.Length, trainData$Petal.Width, pch = 19, col = color,
     xlab = "Petal.Length", ylab = "Petal.Width")
legend(x = 1, y = 2.5, legend = c("Setosa", "versicolor", "virginica"),
       col = c("red", "green", "blue"), pch = 19)

color <- ifelse(preClass=="setosa", "red", ifelse(preClass=="versicolor", "green",
                "blue"))

pType = ifelse(preClass == testClass, 17, 4)
points(testData$Petal.Length, testData$Petal.Width, pch = pType, col = color)
```


Setosa
versicolor
virginica
Petal.Width
Petal.Length
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7

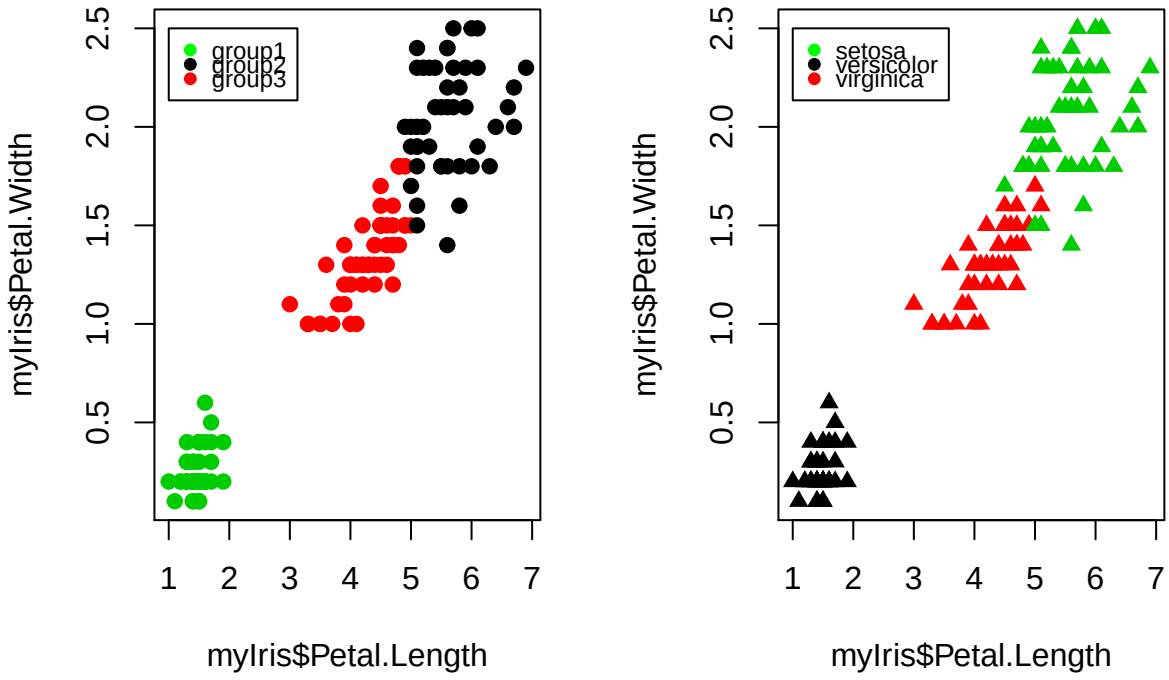
clustering example

kmeans clustering

```
myIris <- iris[3:4]
group <- iris$Species
predGroup <- kmeans(myIris, centers = 3)
predGroupC <- ifelse(predGroup$cluster==2, "setosa", ifelse(predGroup$cluster==3,
                                                            "versicolor", "virginnica"))
predGroupC <- factor(predGroupC)
table(predGroupC, group)
```

```
##             group
## predGroupC   setosa versicolor virginica
##   setosa          0          2        44
##   versicolor      0         48         6
##   virginnica     50          0         0
```

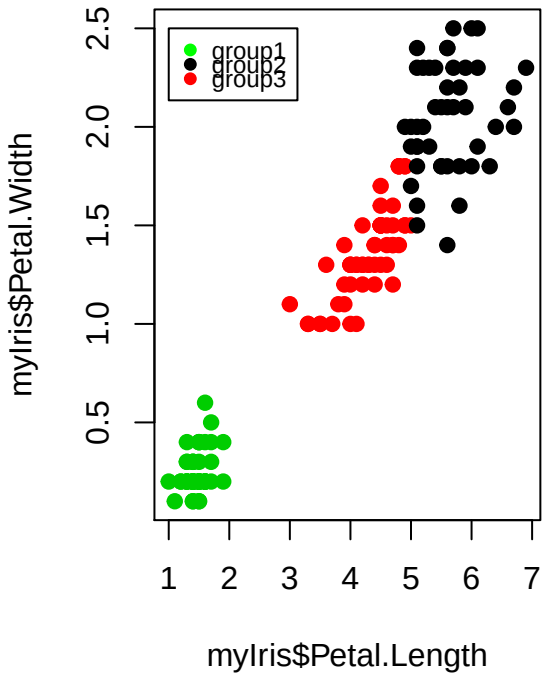
```
par(mfrow = c(1,2))
plot(myIris$Petal.Length, myIris$Petal.Width, pch = 19, col = predGroupC)
legend(x=1,y=2.5, legend = c("group1", "group2", "group3"),
       col = c("green", "black", "red"), pch = 19, y.intersp=0.5, cex = 0.75)

plot(myIris$Petal.Length, myIris$Petal.Width, pch = 17, col = group)
legend(x=1,y=2.5, legend = c("setosa", "versicolor", "virginica"),
       col = c("green", "black", "red"), pch = 19, y.intersp=0.5, cex = 0.75)
```

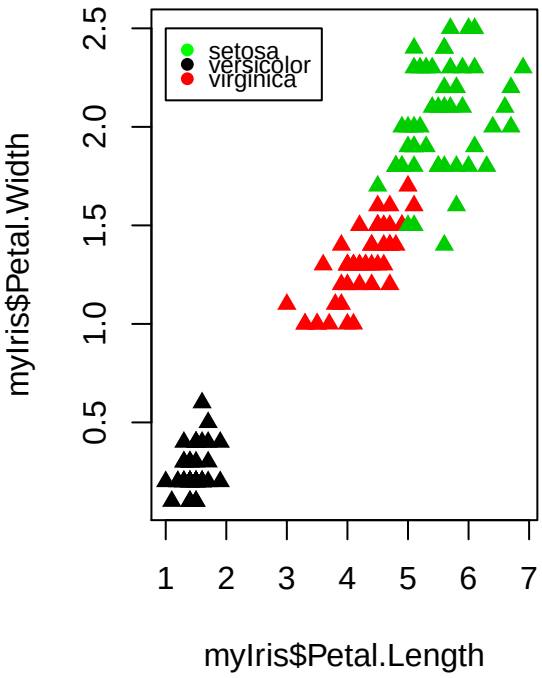



```
par(mfrow = c(1,1))
```

hierarchichal clustering

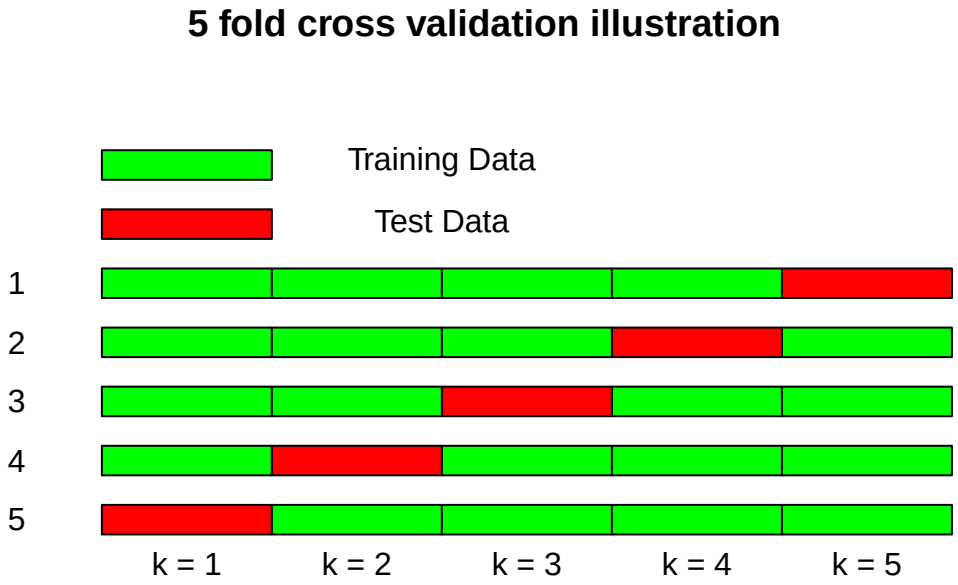
```
group <- iris$Species
disM <- dist(myIris)
irisClust <- hclust(disM)
clusters <- cutree(irisClust, k = 3)

clusters <- ifelse(clusters==1, "setosa", ifelse(clusters==2,
                                                 "virginnica", "versicolor"))
clusters <- factor(clusters)


par(mfrow = c(1,2))
plot(myIris$Petal.Length, myIris$Petal.Width, pch = 19, col = predGroupC)
legend(x=1,y=2.5, legend = c("group1", "group2", "group3"),
       col = c("green", "black", "red"), pch = 19, y.intersp=0.5, cex = 0.75)

plot(myIris$Petal.Length, myIris$Petal.Width, pch = 17, col = group)
legend(x=1,y=2.5, legend = c("setosa", "versicolor", "virginica"),
       col = c("green", "black", "red"), pch = 19, y.intersp=0.5, cex = 0.75)
```






```
par(mfrow = c(1,1))
```

Cross-validation



```
#rfModel <- randomForest(Species ~ . , data = trainData, ntree = 3)
```

knn

```
head(trees)
```

```
##   Girth Height Volume
## 1   8.3     70   10.3
## 2   8.6     65   10.3
## 3   8.8     63   10.2
## 4  10.5     72   16.4
## 5  10.7     81   18.8
## 6  10.8     83   19.7
```

```
set.seed(100)
index <- sample(nrow(iris), 0.6*nrow(iris))
p <- knn(iris[index, 1:4], iris[-index, 1:4], iris[index, 5], 1)
#data.frame(iris[-index, 5], p)
table(iris[-index, 5], p)
```

```
##             p
##              setosa versicolor virginica
##   setosa         24          0         0
##   versicolor      0         16         1
##   virginica       0          3        16
```

```
## show cross validation
## show parameter selection
## show visualization
```

knn2

```
head(trees)
```

```
##   Girth Height Volume
## 1   8.3     70   10.3
## 2   8.6     65   10.3
## 3   8.8     63   10.2
## 4  10.5     72   16.4
## 5  10.7     81   18.8
## 6  10.8     83   19.7
```

```
set.seed(100)
index <- sample(nrow(trees), 0.6*nrow(trees))
p <- knn(trees[index, 1:2], trees[-index, 1:2], iris[index, 3], 4)
#data.frame(trees[-index, 3], p)
```

baye's theorem

```
#head(Titanic)
```

svm

```
#svmModel <- svm()
```