iris

Yatharth Malik

January 3, 2017

```
library(caTools)
library(caret)
library(rpart)
library(rpart.plot)
library(randomForest)
library(VGAM)
```

```
iris = read.csv("iris.csv",header = F)

names = c("sepal_length","sepal_width", "petal_length","petal_width","class")

names(iris) = names          #Assigning names to data frame
```

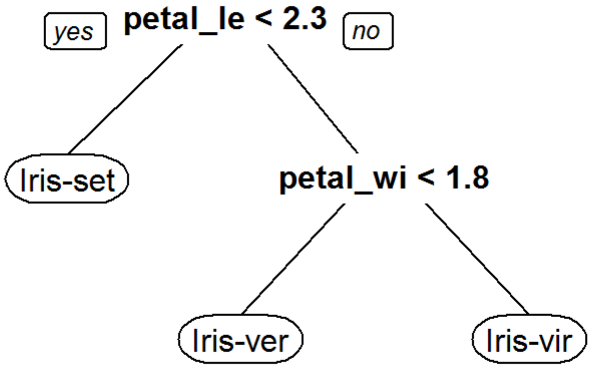
To create training and testing data set,we’ll be using library caTools.Here,65% of original dataset will act as training dataset while remaining 35% will be testing dataset.

```
split = sample.split(iris$class,SplitRatio = 0.65)

train = subset(iris,split == T)
test = subset(iris,split == F)
```

1. Decision Trees

```
tree = rpart(class ~ . ,data = train,method = "class")
prp(tree)
```



```
preds = predict(tree, newdata = test,type = "class")
```

To create confusion matrix

```
table(test$class,preds)
```

```
##                    preds
##                     Iris-setosa Iris-versicolor Iris-virginica
##   Iris-setosa              18               0              0
##   Iris-versicolor           0              18              0
##   Iris-virginica            0               3             15
```

Accuracy of model is given by

```
((18+16+16)/nrow(test))*100
```

```
## [1] 92.59259
```

2. Random Forest

```
rf = randomForest(class~ . ,data = train)
preds1 = predict(rf,newdata = test)
```

To create confusion matrix

```
table(test$class,preds1)
```

```
##                  preds1
##                   Iris-setosa Iris-versicolor Iris-virginica
##   Iris-setosa              18               0              0
##   Iris-versicolor           0              18              0
##   Iris-virginica            0               2             16
```

Accuracy of model is given by

```
((18+16+18)/(nrow(test)))*100
```

```
## [1] 96.2963
```

3. Logisitic Regression

Usually logistic regression is used for binary classification,but we’ll be using library (“VGAM”) for multivariate classification

```
fit = vglm(class ~ . ,data = train ,family = "multinomial")
probs = predict(fit,newdata =test, type = "response")
predictions = apply(probs,1,which.max)
predictions[which(predictions=="1")] = levels(test$class)[1]
predictions[which(predictions=="2")] = levels(test$class)[2]
predictions[which(predictions=="3")] = levels(test$class)[3]
```

To create confusion matrix

```
table(test$class,predictions)
```

```
##                  predictions
##                   Iris-setosa Iris-versicolor Iris-virginica
##   Iris-setosa              18               0              0
##   Iris-versicolor           0              18              0
##   Iris-virginica            0               1             17
```

Accuracy of model is calculated as follows

```
((18+17+18)/nrow(test))*100
```

```
## [1] 98.14815
```