Practical Machine Learning - Week3

Saul Lugo

January 22, 2016

Predicting with Trees

The basic algorithm for predicting with trees is the following:

1) Start with all the variables in one group
2) Find the variable/split that best separates the outcomes
3) Divide the data into two groups ("leaves") on that split
4) Within each split, find the variable that best separates the outcome
5) Continue until the groups are too small or sufficiently pure

Example with the Iris dataset

```
data(iris)
library(ggplot2)
library(caret)
names(iris)
```

```
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"
## [5] "Species"
```

```
table(iris$Species)
```

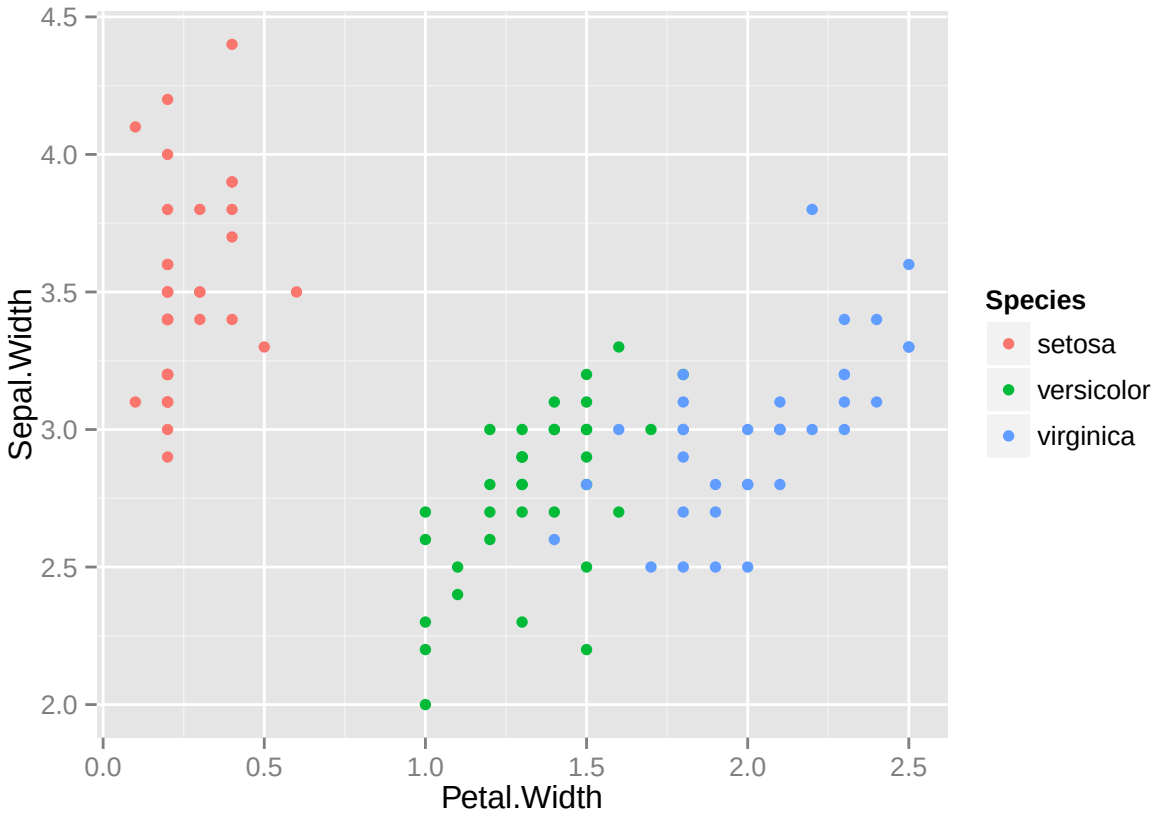
```
##
##     setosa versicolor  virginica
##         50         50         50
```

```
#Create the data partition
inTrain <- createDataPartition(y=iris$Species,p=0.7,list=FALSE)
training <- iris[inTrain,]
testing <- iris[-inTrain,]
dim(training); dim(testing)
```

```
## [1] 105   5
```

```
## [1] 45  5
```

```
#Exploring the data
qplot(Petal.Width,Sepal.Width,colour=Species,data=training)
```




```
#Training the model (a tree model)
modFit <- train(Species ~ ., method="rpart", data = training)
print(modFit$finalModel)
```

```
## n= 105
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 105 70 setosa (0.3333333 0.3333333 0.3333333)
##   2) Petal.Length< 2.6 35  0 setosa (1.0000000 0.0000000 0.0000000) *
##   3) Petal.Length>=2.6 70 35 versicolor (0.0000000 0.5000000 0.5000000)
##     6) Petal.Length< 4.75 32  1 versicolor (0.0000000 0.9687500 0.0312500) *
##     7) Petal.Length>=4.75 38  4 virginica (0.0000000 0.1052632 0.8947368) *
```