Decision Trees - Predicting Flower types

V2 Maestros

Contents

Problem Statement

The input data is the iris dataset. It contains recordings of information about flower samples. For each sample, the petal and sepal length and width are recorded along with the type of the flower. We need to use this dataset to build a decision tree model that can predict the type of flower based on the petal and sepal information.

Techniques Used

1. Decision Trees - C5.0
2. Training and Testing
3. Confusion Matrix

Data Engineering & Analysis

```
setwd("C:/Personal/V2Maestros/Modules/Machine Learning Algorithms/Decision Trees")

iris_data <- iris

str(iris_data)
```

Loading and understanding the dataset

```
## '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 ...
```

```
summary(iris_data)
```

```
##   Sepal.Length   Sepal.Width    Petal.Length   Petal.Width
##  Min.   :4.30   Min.   :2.00   Min.   :1.00   Min.   :0.1
##  1st Qu.:5.10   1st Qu.:2.80   1st Qu.:1.60   1st Qu.:0.3
##  Median :5.80   Median :3.00   Median :4.35   Median :1.3
##  Mean   :5.84   Mean   :3.06   Mean   :3.76   Mean   :1.2
##  3rd Qu.:6.40   3rd Qu.:3.30   3rd Qu.:5.10   3rd Qu.:1.8
##  Max.   :7.90   Max.   :4.40   Max.   :6.90   Max.   :2.5
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

```
head(iris_data)
```

```
##   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
```

Data Cleansing

1. The ranges of values in each of the variables (columns) look ok without any kind of outliers
2. There is equal distribution of the three classes - setosa, versicolor and virginia

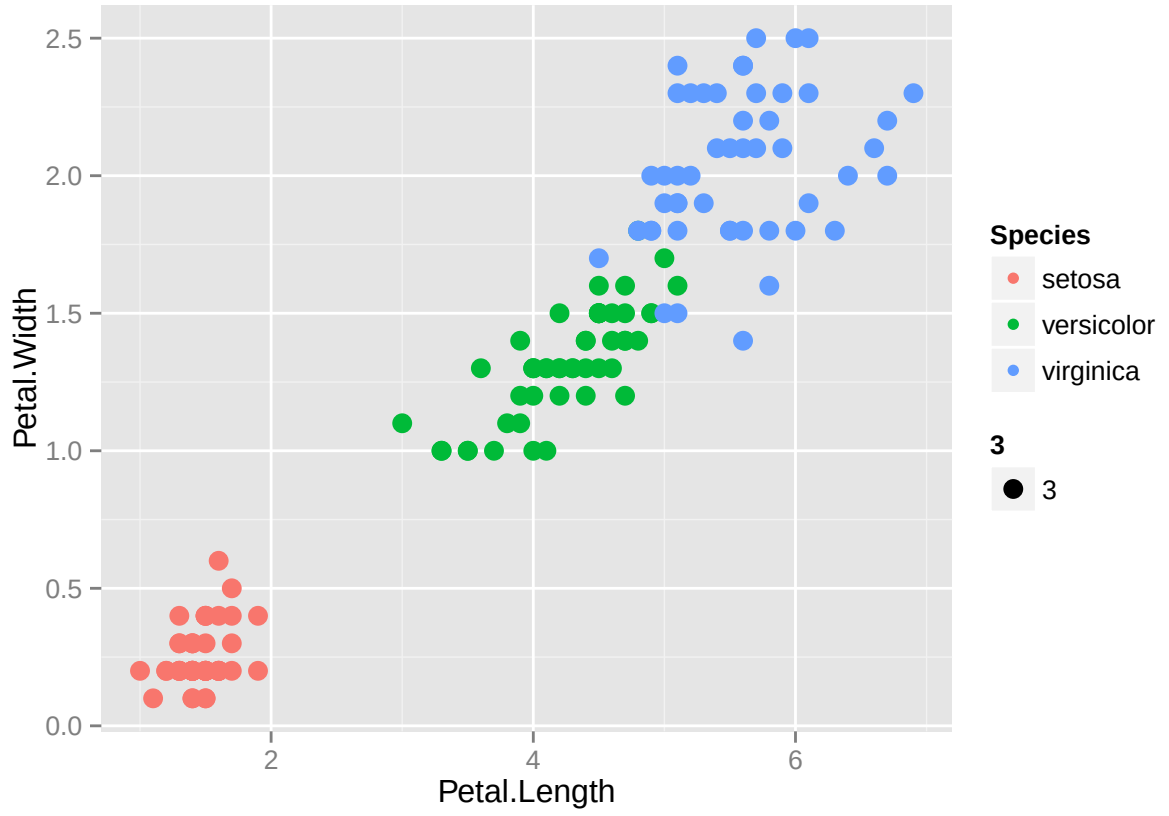
No cleansing required

```
library(ggplot2)
```

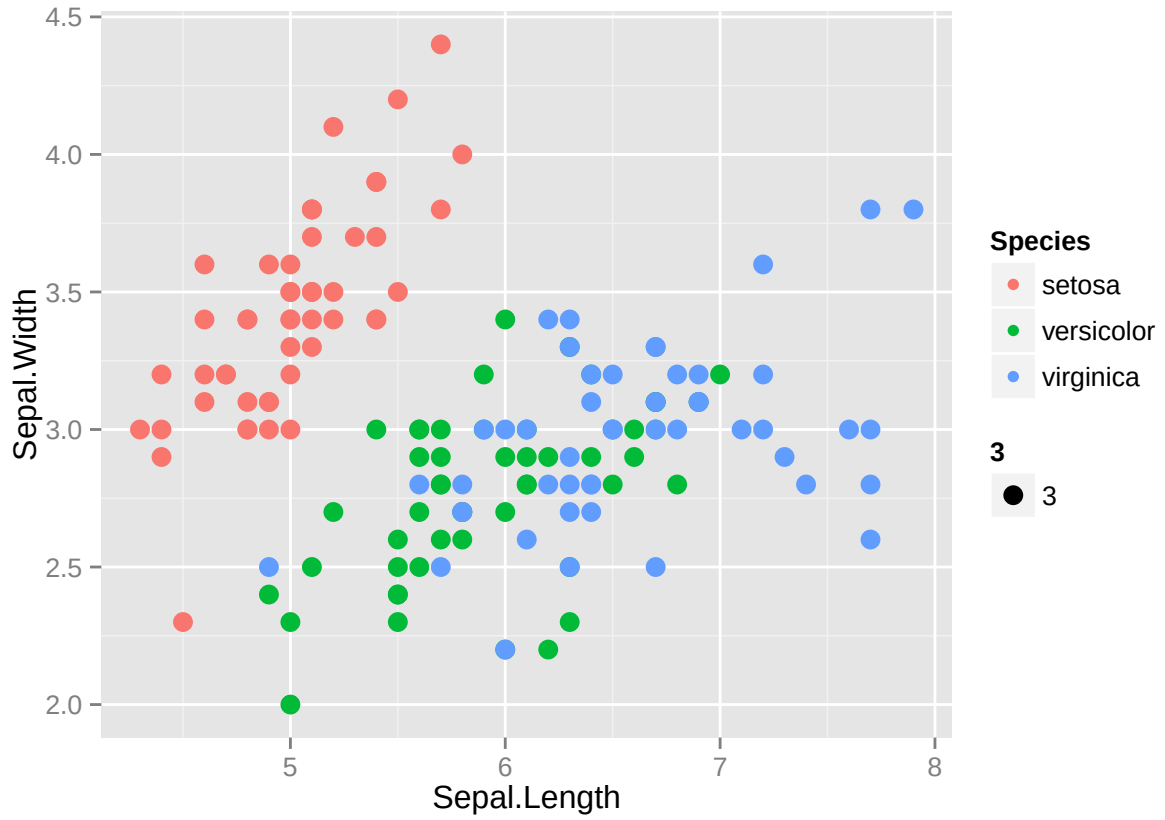
Exploratory Data Analysis

```
## Warning: package 'ggplot2' was built under R version 3.1.1
```

```
qplot(Petal.Length, Petal.Width, data=iris_data, colour=Species, size=3)
```

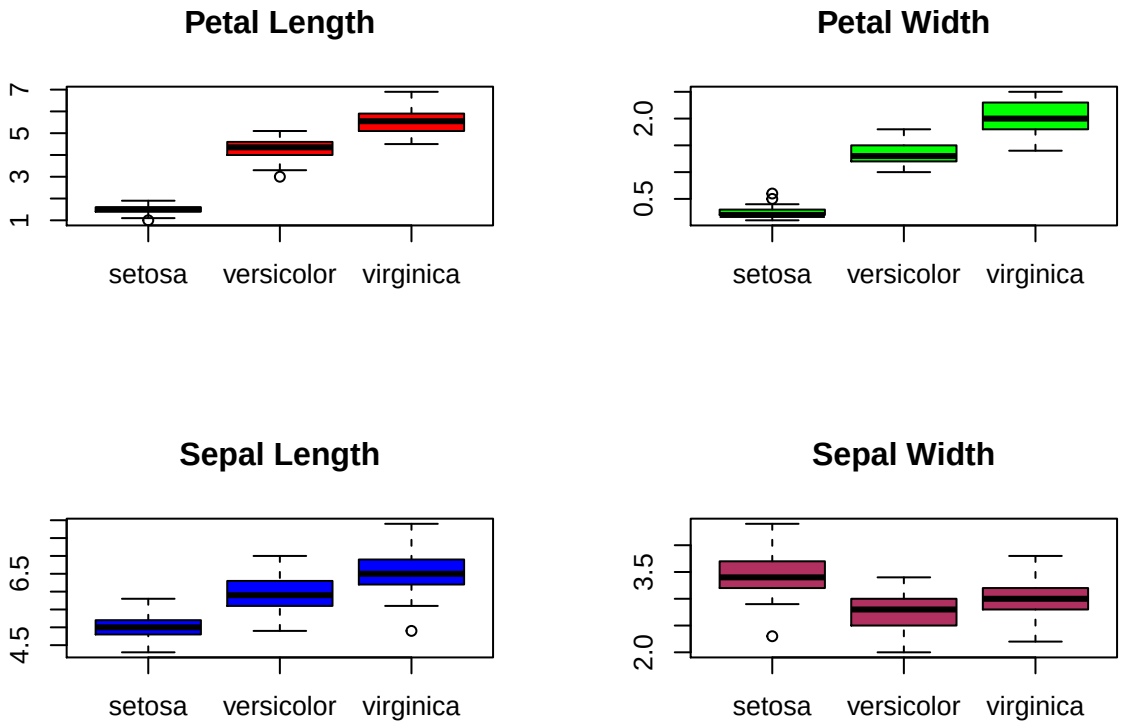



```
qplot(Sepal.Length, Sepal.Width, data=iris_data, colour=Species, size=3)
```




```
par(mfrow=c(2,2))

boxplot( Petal.Length ~ Species, data=iris_data,col="red")
title("Petal Length")
boxplot( Petal.Width ~ Species, data=iris_data,col="green")
title("Petal Width")
boxplot( Sepal.Length ~ Species, data=iris_data,col="blue")
title("Sepal Length")
boxplot( Sepal.Width ~ Species, data=iris_data,col="maroon")
title("Sepal Width")
```



All 3 except Sepal Width seem to bring the significant differenciation between the 3 classes

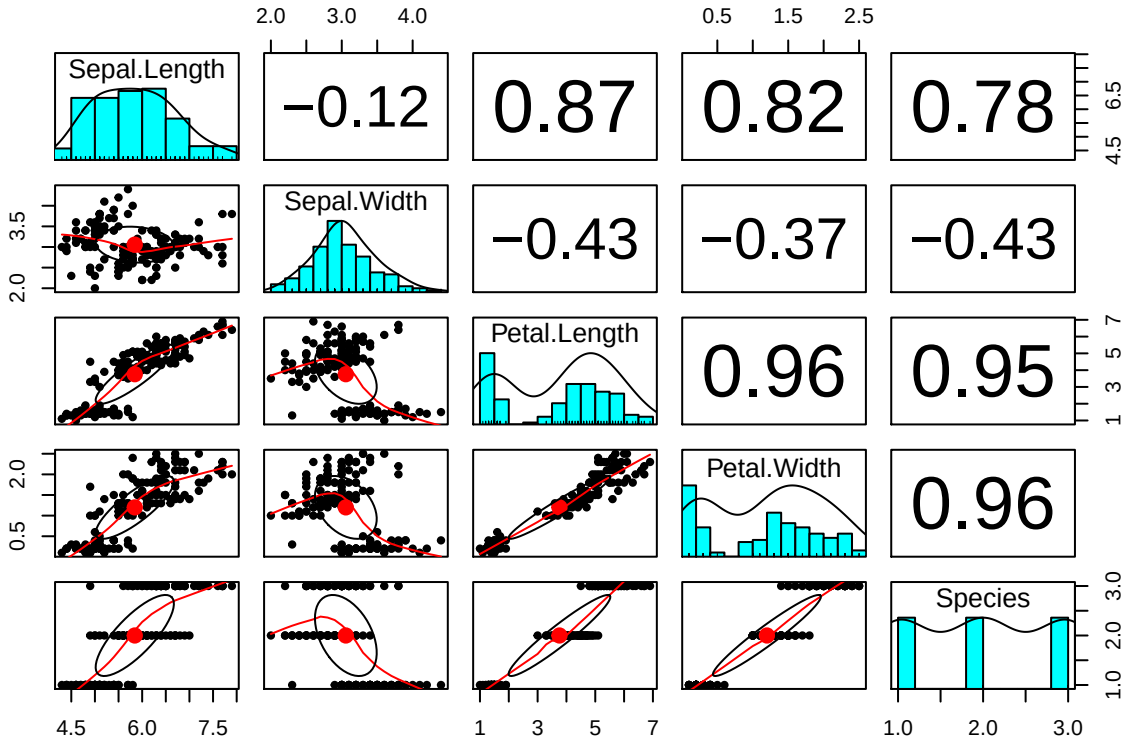
```
library(psych)
```

Correlations

```
## Warning: package 'psych' was built under R version 3.1.1

##
## Attaching package: 'psych'
##
## The following object is masked from 'package:ggplot2':
##
##     %+%
```

```
pairs.panels(iris_data)
```



The correlation co-efficients confirm the findings of the Exploratory Data Analysis.

Modeling & Prediction

Split Training and Testing Split training and testing datasets in the ratio of 70-30

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.1.1
```

```
## Loading required package: lattice
```

```
inTrain <- createDataPartition(y=iris_data$Species ,p=0.7,list=FALSE)
training <- iris_data[inTrain,]
testing <- iris_data[-inTrain,]
dim(training);dim(testing)
```

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

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

```
table(training$Species); table(testing$Species)
```

```
##
##     setosa versicolor  virginica
##         35         35         35
```

```
##
##     setosa versicolor  virginica
##         15         15         15
```

Model Building Build model based on the training data

```
library(C50)
```

```
## Warning: package 'C50' was built under R version 3.1.1
```

```
model <- C5.0(training[-5], training$Species)
summary(model)
```

```
##
## Call:
## C5.0.default(x = training[-5], y = training$Species)
##
##
## C5.0 [Release 2.07 GPL Edition]      Tue Feb 10 10:19:04 2015
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 105 cases (5 attributes) from undefined.data
##
## Decision tree:
##
## Petal.Length <= 1.9: setosa (35)
## Petal.Length > 1.9:
## :...Petal.Width > 1.7: virginica (30)
##     Petal.Width <= 1.7:
##     :...Petal.Length <= 4.9: versicolor (34/1)
##         Petal.Length > 4.9: virginica (6/2)
##
##
## Evaluation on training data (105 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##       4    3( 2.9%)   <<
##
##
##     (a)   (b)   (c)    <-classified as
##    ----  ----  ----
##      35                (a): class setosa
##            33     2    (b): class versicolor
```

```
##             1    34    (c): class virginica
##
##
##  Attribute usage:
##
##  100.00% Petal.Length
##   66.67% Petal.Width
##
##
## Time: 0.0 secs
```

The model clearly shows how the decision tree looks like. This is one of the advantages of decision trees.

Testing

Now let us predict the class for each sample in the test data. Then compare the prediction with the actual value of the class.

```
library(caret)
predicted <- predict(model, testing)
table(predicted)
```

```
## predicted
##     setosa versicolor  virginica
##         15         14         16
```

```
confusionMatrix(predicted, testing$Species)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   setosa versicolor virginica
##   setosa         15          0         0
##   versicolor      0         14         0
##   virginica       0          1        15
##
## Overall Statistics
##
##                Accuracy : 0.978
##                  95% CI : (0.882, 0.999)
##     No Information Rate : 0.333
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.967
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: setosa Class: versicolor Class: virginica
## Sensitivity                  1.000            0.933           1.000
## Specificity                  1.000            1.000           0.967
## Pos Pred Value               1.000            1.000           0.938
```

```
## Neg Pred Value               1.000            0.968           1.000
## Prevalence                   0.333            0.333           0.333
## Detection Rate               0.333            0.311           0.333
## Detection Prevalence         0.333            0.311           0.356
## Balanced Accuracy            1.000            0.967           0.983
```

The model shows very high accuracy. The reason why the accuracy is so high is because, the data itself has very strong signals (seperation between the classes). Sepal.Length and Sepal.Width have very high correlations and they are used in the decision tree. In order to see how the tree will behave if it only had Sepal.Length and Sepal.Width, let us remove that data and see how accurate the tree is.

```
#get only Sepal Length, width and species
sub_data <- iris_data[, c(1,2,5)]
summary(sub_data)
```

```
##   Sepal.Length   Sepal.Width         Species
##  Min.   :4.30   Min.   :2.00   setosa    :50
##  1st Qu.:5.10   1st Qu.:2.80   versicolor:50
##  Median :5.80   Median :3.00   virginica :50
##  Mean   :5.84   Mean   :3.06
##  3rd Qu.:6.40   3rd Qu.:3.30
##  Max.   :7.90   Max.   :4.40
```

```
inTrain <- createDataPartition(y=sub_data$Species ,p=0.7,list=FALSE)
training <- sub_data[inTrain,]
testing <- sub_data[-inTrain,]

model <- C5.0(training[-3], training$Species)
summary(model)
```

```
##
## Call:
## C5.0.default(x = training[-3], y = training$Species)
##
##
## C5.0 [Release 2.07 GPL Edition]      Tue Feb 10 10:19:04 2015
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 105 cases (3 attributes) from undefined.data
##
## Decision tree:
##
## Sepal.Length <= 5.4: setosa (37/5)
## Sepal.Length > 5.4:
## :...Sepal.Width > 3.6: setosa (5/2)
##     Sepal.Width <= 3.6:
##     :...Sepal.Length <= 7: versicolor (56/25)
##         Sepal.Length > 7: virginica (7)
##
##
## Evaluation on training data (105 cases):
```

```
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##       4   32(30.5%)   <<
##
##
##     (a)   (b)   (c)    <-classified as
##    ----  ----  ----
##      35                (a): class setosa
##       4    31          (b): class versicolor
##       3    25     7    (c): class virginica
##
##
##  Attribute usage:
##
##  100.00% Sepal.Length
##   64.76% Sepal.Width
##
##
## Time: 0.0 secs
```

```
predicted <- predict(model, testing)
confusionMatrix(predicted, testing$Species)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   setosa versicolor virginica
##   setosa         14          2         0
##   versicolor      1         13        12
##   virginica       0          0         3
##
## Overall Statistics
##
##                Accuracy : 0.667
##                  95% CI : (0.51, 0.8)
##     No Information Rate : 0.333
##     P-Value [Acc > NIR] : 5e-06
##
##                   Kappa : 0.5
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: setosa Class: versicolor Class: virginica
## Sensitivity                  0.933            0.867           0.2000
## Specificity                  0.933            0.567           1.0000
## Pos Pred Value               0.875            0.500           1.0000
## Neg Pred Value               0.966            0.895           0.7143
## Prevalence                   0.333            0.333           0.3333
## Detection Rate               0.311            0.289           0.0667
## Detection Prevalence         0.356            0.578           0.0667
```

```
## Balanced Accuracy            0.933             0.717           0.6000
```

You will notice that the decision tree itself has become more complex and the accuracy dropped significantly

Conclusions

. Irrespective of the algorithm used, we need high correlations between the predictor and target variables for good predictions

---