Multi-class Classification with Decision Trees

Juan Luis Polo, Student, Illinois Institute of Technology

```
library(rpart)
library(caret)
```

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

```
## Loading required package: ggplot2
```

```
library(rpart.plot)

rm(list=ls())
```

Load the Iris dataset amd split into 80-20 (test/train). The dataset has 150 observations; 120 are used for training and 30 for testing.

```
data(iris)
set.seed(100)
index <- sample(1:nrow(iris), size=0.2*nrow(iris))
test <- iris[index, ]
train <- iris[-index, ]
```

Let's see the real class distribution in the test dataset

```
table(test$Species)
```

```
##
##     setosa versicolor  virginica
##         10         10         10
```

Build the model with all predictor variables and predict on test data. Show confusion matrix

```
model <- rpart(Species ~ ., method="class", data=train)
pred <- predict(model, test, type="class")
```

When we display the confusion matrix, you will note $>$ 2 classes

```
confusionMatrix(pred, test[, 5])
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   setosa versicolor virginica
##   setosa         10          0         0
##   versicolor      0         10         1
##   virginica       0          0         9
##
## Overall Statistics
##
##                Accuracy : 0.9667
```

```
##                  95% CI : (0.8278, 0.9992)
##     No Information Rate : 0.3333
##     P-Value [Acc > NIR] : 2.963e-13
##
##                   Kappa : 0.95
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: setosa Class: versicolor Class: virginica
## Sensitivity                 1.0000            1.0000           0.9000
## Specificity                 1.0000            0.9500           1.0000
## Pos Pred Value              1.0000            0.9091           1.0000
## Neg Pred Value              1.0000            1.0000           0.9524
## Prevalence                  0.3333            0.3333           0.3333
## Detection Rate              0.3333            0.3333           0.3000
## Detection Prevalence        0.3333            0.3667           0.3000
## Balanced Accuracy           1.0000            0.9750           0.9500
```

One vs.All Class SETOSA

```
setosa.train.species <- rep("pos", dim(train)[1])
setosa.train.species[train$Species!="setosa"] <- "neg"
setosa.train <- train
setosa.train$Species <- as.factor(setosa.train.species)

setosa.test.species <- rep("pos", dim(test)[1])
setosa.test.species[test$Species!="setosa"] <- "neg"
setosa.test <- test
setosa.test$Species <- as.factor(setosa.test.species)
```

It is needed to balance the attribute Species before creating the model:

```
cat("Number of positive class values: ", length(which(setosa.train$Species=="pos")), "\n", fill = T)
```

```
## Number of positive class values:  40
```

```
cat("Number of negative class values: ", length(which(setosa.train$Species=="neg")), "\n", fill = T)
```

```
## Number of negative class values:  80
```

Undersampling:

```
set.seed(1122)
setosa.sample.neg <- sample( which(setosa.train$Species=="neg"),
                             length(which(setosa.train$Species=="pos")) )
setosa.train <- setosa.train[c(which(setosa.train$Species=="pos"), setosa.sample.neg),]
```

Creating the model for setosa:

```
setosa.model <- rpart(Species ~ ., method="class", data=setosa.train)
setosa.pred <- predict(setosa.model, setosa.test, type="class")
```

Plotting Sensitivity, Specificity and Precision for the binary model:

```
round(confusionMatrix(setosa.pred, setosa.test[,5])$byClass["Sensitivity"], 2)
```

```
## Sensitivity
##           1
```

```
round(confusionMatrix(setosa.pred, setosa.test[,5])$byClass["Specificity"], 2)
```

```
## Specificity
##           1
```

```
round(confusionMatrix(setosa.pred, setosa.test[,5])$byClass["Pos Pred Value"], 2)
```

```
## Pos Pred Value
##              1
```

The confusion matrix for this binary model:

```
confusionMatrix(setosa.pred, setosa.test[,5])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction neg pos
##        neg  20   0
##        pos   0  10
##
##                Accuracy : 1
##                  95% CI : (0.8843, 1)
##     No Information Rate : 0.6667
##     P-Value [Acc > NIR] : 5.215e-06
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.6667
##          Detection Rate : 0.6667
##    Detection Prevalence : 0.6667
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : neg
##
```

One vs.All Class VIRGINICA

```
virginica.train.species <- rep("pos", dim(train)[1])
virginica.train.species[train$Species!="virginica"] <- "neg"
virginica.train <- train
virginica.train$Species <- as.factor(virginica.train.species)

virginica.test.species <- rep("pos", dim(test)[1])
virginica.test.species[test$Species!="virginica"] <- "neg"
virginica.test <- test
virginica.test$Species <- as.factor(virginica.test.species)
```

It is needed to balance the attribute Species before creating the model:

```
cat("Number of positive class values: ", length(which(virginica.train$Species=="pos")), "\n", fill = T)
```

```
## Number of positive class values:  40
```

```
cat("Number of negative class values: ", length(which(virginica.train$Species=="neg")), "\n", fill = T)
```

```
## Number of negative class values:  80
```

Undersampling:

```
set.seed(1122)
virginica.sample.neg <- sample( which(virginica.train$Species=="neg"),
                                length(which(virginica.train$Species=="pos")) )
virginica.train <- virginica.train[c(which(virginica.train$Species=="pos"), virginica.sample.neg),]
```

Creating the model for setosa:

```
virginica.model <- rpart(Species ~ ., method="class", data=virginica.train)
virginica.pred <- predict(virginica.model, setosa.test, type="class")
```

Plotting Sensitivity, Specificity and Precision for the binary model:

```
round(confusionMatrix(virginica.pred, virginica.test[,5])$byClass["Sensitivity"], 2)
```

```
## Sensitivity
##           1
```

```
round(confusionMatrix(virginica.pred, virginica.test[,5])$byClass["Specificity"], 2)
```

```
## Specificity
##         0.9
```

```
round(confusionMatrix(virginica.pred, virginica.test[,5])$byClass["Pos Pred Value"], 2)
```

```
## Pos Pred Value
##           0.95
```

The confusion matrix for this binary model:

```
confusionMatrix(virginica.pred, virginica.test[,5])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction neg pos
##        neg  20   1
##        pos   0   9
##
##                Accuracy : 0.9667
##                  95% CI : (0.8278, 0.9992)
##     No Information Rate : 0.6667
##     P-Value [Acc > NIR] : 8.344e-05
##
##                   Kappa : 0.9231
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 1.0000
##             Specificity : 0.9000
```

```
##          Pos Pred Value : 0.9524
##          Neg Pred Value : 1.0000
##              Prevalence : 0.6667
##          Detection Rate : 0.6667
##    Detection Prevalence : 0.7000
##       Balanced Accuracy : 0.9500
##
##        'Positive' Class : neg
##
```

One vs.All Class VERSICOLOR

```
versicolor.train.species <- rep("pos", dim(train)[1])
versicolor.train.species[train$Species!="versicolor"] <- "neg"
versicolor.train <- train
versicolor.train$Species <- as.factor(versicolor.train.species)

versicolor.test.species <- rep("pos", dim(test)[1])
versicolor.test.species[test$Species!="versicolor"] <- "neg"
versicolor.test <- test
versicolor.test$Species <- as.factor(versicolor.test.species)
```

It is needed to balance the attribute Species before creating the model:

```
cat("Number of positive class values: ", length(which(versicolor.train$Species=="pos")), "\n", fill = T
```

```
## Number of positive class values:  40
```

```
cat("Number of negative class values: ", length(which(versicolor.train$Species=="neg")), "\n", fill = T
```

```
## Number of negative class values:  80
```

Undersampling:

```
set.seed(1122)
versicolor.sample.neg <- sample( which(versicolor.train$Species=="neg"),
                                 length(which(versicolor.train$Species=="pos")) )
versicolor.train <- versicolor.train[c(which(versicolor.train$Species=="pos"),
                                       versicolor.sample.neg),]
```

Creating the model for setosa:

```
versicolor.model <- rpart(Species ~ ., method="class", data=versicolor.train)
versicolor.pred <- predict(versicolor.model, versicolor.test, type="class")
```

Plotting Sensitivity, Specificity and Precision for the binary model:

```
round(confusionMatrix(versicolor.pred, versicolor.test[,5])$byClass["Sensitivity"], 2)
```

```
## Sensitivity
##        0.95
```

```
round(confusionMatrix(versicolor.pred, versicolor.test[,5])$byClass["Specificity"], 2)
```

```
## Specificity
##           1
```

```
round(confusionMatrix(versicolor.pred, versicolor.test[,5])$byClass["Pos Pred Value"], 2)
```

```
## Pos Pred Value
```

```
##              1
```

The confusion matrix for this binary model:

```
confusionMatrix(versicolor.pred, versicolor.test[,5])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction neg pos
##        neg  19   0
##        pos   1  10
##
##                Accuracy : 0.9667
##                  95% CI : (0.8278, 0.9992)
##     No Information Rate : 0.6667
##     P-Value [Acc > NIR] : 8.344e-05
##
##                   Kappa : 0.9268
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9500
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9091
##              Prevalence : 0.6667
##          Detection Rate : 0.6333
##    Detection Prevalence : 0.6333
##       Balanced Accuracy : 0.9750
##
##        'Positive' Class : neg
##
```