Caret / Recursive Partitioning

Abhishek Chaturvedi

05/15/2017

Exercise 1: caret/logistic regression (5 points)

Rebuild your logistic regression model from the previous week, this time using the `caret` package.

- Calculate the training or apparent performance of the model.
- Calculate an unbiased measure of performance
- Create a ROC Curve for your model

Show all work.

```
# Your Work Here
library('data.table')
flightsDataJoined <- readRDS("flightsDataJoined.rds")

y <- "arr_delay"
xs <- c('humid','dep_time', 'sched_dep_time','sched_arr_time','dep_delay','origin')
yx <- flightsDataJoined[,c(y,xs),with=FALSE]
yx <- na.omit(yx)

set.seed(333)
inTraining <- createDataPartition(na.omit(yx[,arr_delay]), p = .75, list = FALSE)
training <- yx[inTraining, ]
testing <- yx[-inTraining, ]

set.seed(3333)
lmModel1 <- train(arr_delay ~ ., data = training, method = "lm")

library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
#performance testing using the model
yhat <- predict(lmModel1,testing,type="raw")
postResample(pred = yhat, obs = testing$arr_delay)
```

```
##       RMSE   Rsquared
## 16.6561162  0.7979787
```

Exercise 2: caret/rpart (5 points)

Using the `caret` and `rpart` packages, create a **classification** model for flight delays using your NYC FLight data. Your solution should include:

- The use of `caret` and `rpart` to train a model.
- An articulation of the the problem your are
- An naive model
- An unbiased calculation of the performance metric
- A plot of your model – (the actual tree; there are several ways to do this)
- A discussion of your model

Show and describe all work

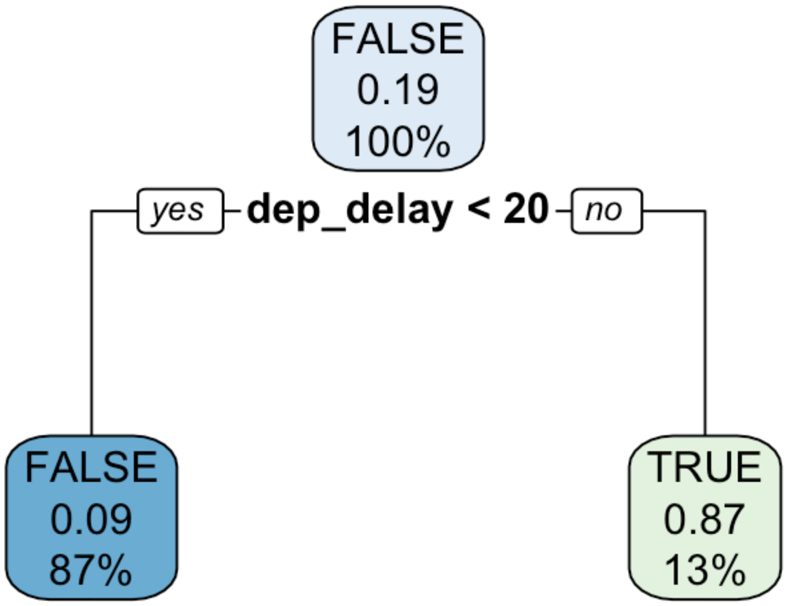
```
# Creating a classification model for flights which has arr_delay more than 15 mins

library("rpart")
library("rpart.plot")

# adding an extea column gt15 to the data with binary values if delay is more than 15
mins
yx <- within(yx, gt15 <- arr_delay>=15)

set.seed(333)
inTraining <- createDataPartition(na.omit(yx[,arr_delay]), p = .75, list = FALSE)
trainingData <- yx[inTraining, ]
testingData <- yx[-inTraining, ]

treeModel <- rpart(gt15 ~ humid + dep_time + sched_dep_time + sched_arr_time + dep_de
lay + origin, data = trainingData, method = "class")
rpart.plot(treeModel)
```



```
treeModel
```

```
## n= 23501
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 23501 4470 FALSE (0.80979533 0.19020467)
##   2) dep_delay< 20.5 20407 1773 FALSE (0.91311805 0.08688195) *
##   3) dep_delay>=20.5 3094  397 TRUE (0.12831286 0.87168714) *
```

```
yhat <- predict(treeModel,testingData,type="class")

confusionMatrix(data = yhat, reference = testingData$gt15)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction FALSE TRUE
##      FALSE  6190  632
##      TRUE    156  854
## 
##                Accuracy : 0.8994
##                  95% CI : (0.8925, 0.906)
##     No Information Rate : 0.8103
##     P-Value [Acc > NIR] : < 2.2e-16
## 
##                   Kappa : 0.627
##  Mcnemar's Test P-Value : < 2.2e-16
## 
##             Sensitivity : 0.9754
##             Specificity : 0.5747
##          Pos Pred Value : 0.9074
##          Neg Pred Value : 0.8455
##              Prevalence : 0.8103
##          Detection Rate : 0.7903
##    Detection Prevalence : 0.8710
##       Balanced Accuracy : 0.7751
## 
##        'Positive' Class : FALSE
## 
```

```
# the model is depending on dep_delay if its more than 20 the arr_delay will be more
than 15 mins have a probability is 0.87

library("pROC")
library("ROCR")
```

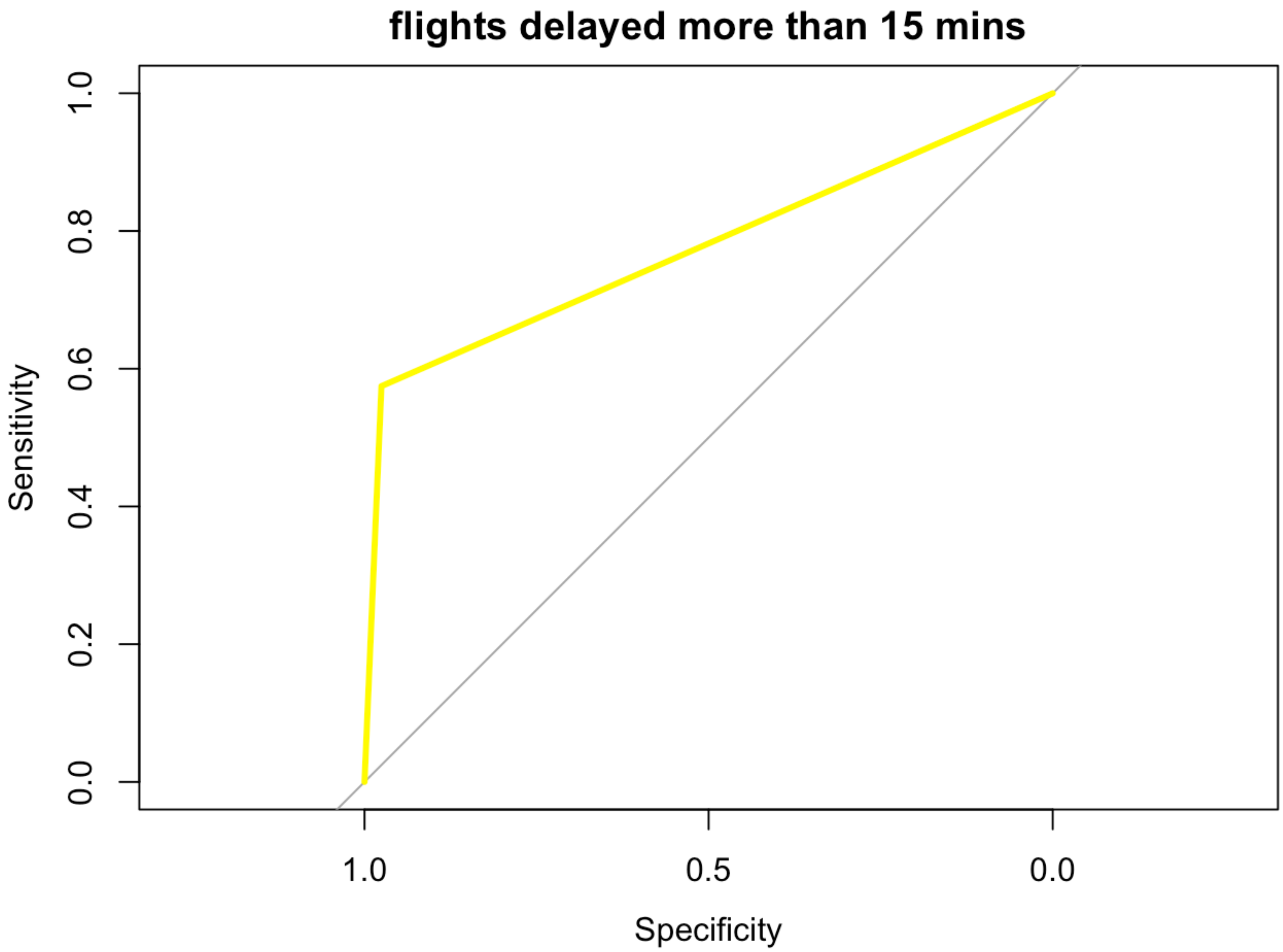
```
## Loading required package: gplots
```

```
## 
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
## 
##     lowess
```

```
gt15Predicted <- predict(treeModel,testingData,type="vector")

plot(roc(testingData$gt15, gt15Predicted, direction="<"),
     col="yellow", lwd=3, main="flights delayed more than 15 mins")
```



Questions:

- Discuss the difference between the models and why you would use one model over the other?
- How might you produce an ROC type curve for the *rpart* model?