Recursive Partitioning / Decision Trees using CARET

Saqib Ali

May 16th, 2017

Load the data

```
data.joined <- readRDS(file="/home/saqib/ml_at_berkeley/CSX460/04-logistic-regression/04-exercise-nycfli
```

Add a categorical variable for arr_delay >= 22 minutes. It is called arrival_delayed

```
data.joined$arrival_delayed <- factor(ifelse(data.joined$arr_delay >= 22, 1,0))
```

Filter out rows with NAs for arrival_delayed

```
data.joined <- data.joined  %>% filter(!is.na(arrival_delayed))

#data.joined <- data.joined[, speed:=NULL]
```

Split the Dataset in Training and Test datasets

```
data.joined.training <- sample_frac(data.joined, .75)
data.joined.testing <- sample_frac(data.joined, .5)
```

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.

Train the Linear Regression Model using CARET

```
# Your Work Here

#data.joined <- data.joined[, speed:=NULL]


#lapply(data.joined, levels)
#(l <- sapply(data.joined, function(x) is.factor(x)))
```

```
fitControl <- trainControl(method = "cv", number = 2)

glmFit <- train(arrival_delayed ~ dep_delay + dest + origin + year + month + day + hour + sched_dep_tim
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Model Performance

```
pred <- as.vector(ifelse(predict(glmFit, newdata=data.joined, type="prob")[,"1"]<.5, 0, 1))
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
probsTest <- predict(glmFit, data.joined.testing, type = "prob")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
pred      <- factor( ifelse(probsTest[, "1"] > 0.5, "1", "0") )
confusionMatrix(pred, data.joined.testing$arrival_delayed)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction     0     1
##          0 93442  6522
##          1  1940 14918
## 
##                Accuracy : 0.9276
##                  95% CI : (0.9261, 0.929)
##     No Information Rate : 0.8165
##     P-Value [Acc > NIR] : < 2.2e-16
## 
##                   Kappa : 0.7365
##  Mcnemar's Test P-Value : < 2.2e-16
## 
##             Sensitivity : 0.9797
##             Specificity : 0.6958
##          Pos Pred Value : 0.9348
##          Neg Pred Value : 0.8849
##              Prevalence : 0.8165
##          Detection Rate : 0.7999
##    Detection Prevalence : 0.8557
##       Balanced Accuracy : 0.8377
## 
##        'Positive' Class : 0
```

```
##
```

Plot the ROC Curve

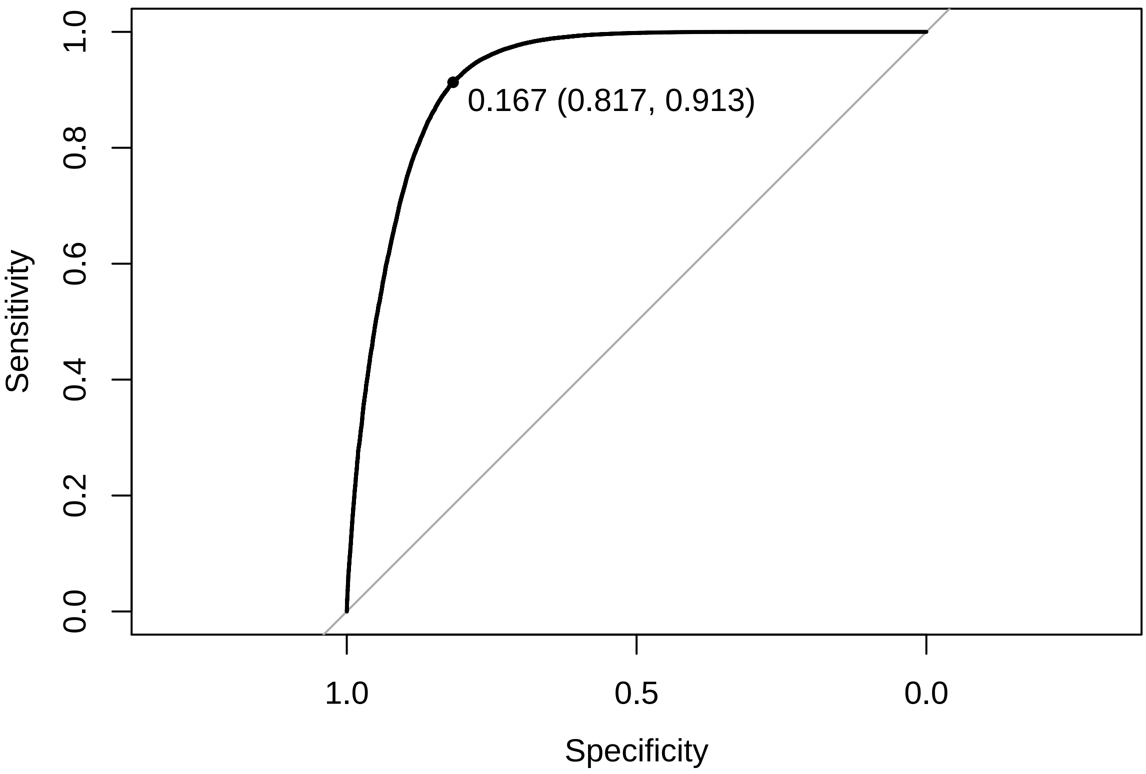
```
library(pROC)
```

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

##
## Attaching package: 'pROC'

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

```
probsTrain <- predict(glmFit, data.joined.training, type = "prob")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
rocCurve   <- roc(response = data.joined.training$arrival_delayed,
                      predictor = probsTrain[, "1"],
                      levels = rev(levels(data.joined.training$arrival_delayed)))
plot(rocCurve, print.thres = "best")
```



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

```
# Your Work Here

fitControl <- trainControl(method = "cv", number = 2)

rpartFit <- train(arrival_delayed ~ dep_delay + dest + origin + year + month + day + hour + sched_dep_ti
```

Model Performance

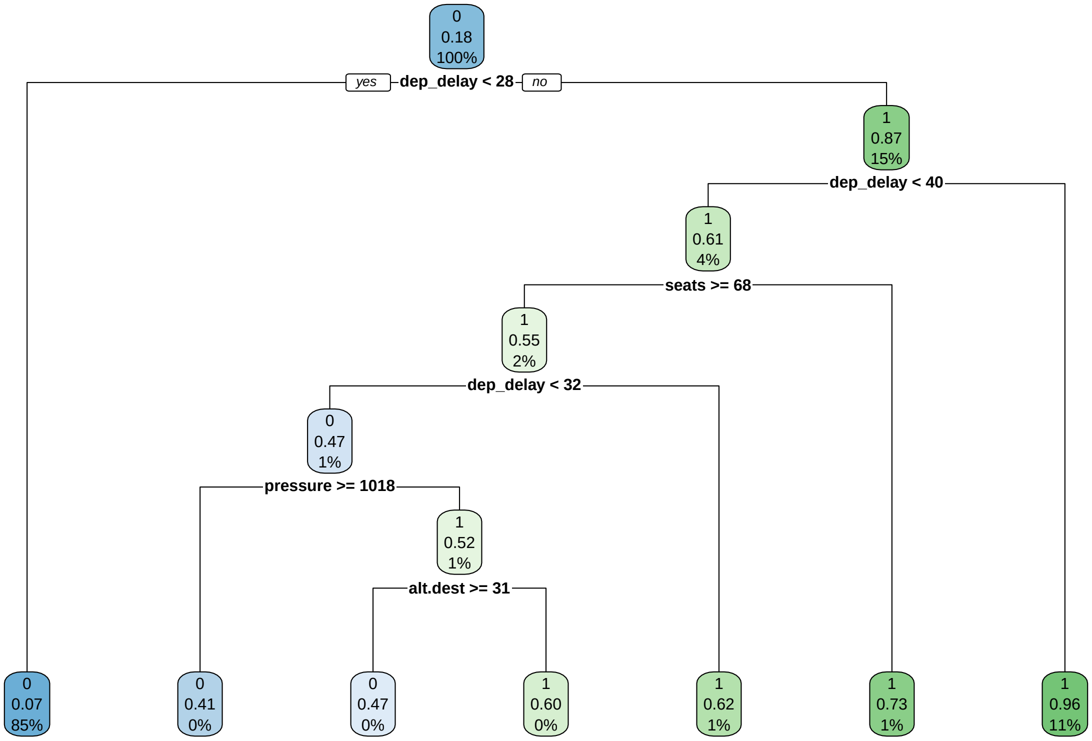
```
pred <- as.vector(ifelse(predict(rpartFit, newdata=data.joined, type="prob")[,"1"]<.5, 0, 1))
probsTest <- predict(rpartFit, data.joined.testing, type = "prob")
pred      <- factor( ifelse(probsTest[, "1"] > 0.5, "1", "0") )
confusionMatrix(pred, data.joined.testing$arrival_delayed)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction     0     1
##          0 93880  7017
##          1  1502 14423
##
##                Accuracy : 0.9271
##                  95% CI : (0.9256, 0.9286)
##     No Information Rate : 0.8165
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7297
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9843
##             Specificity : 0.6727
##          Pos Pred Value : 0.9305
##          Neg Pred Value : 0.9057
##              Prevalence : 0.8165
##          Detection Rate : 0.8036
##    Detection Prevalence : 0.8637
##       Balanced Accuracy : 0.8285
##
##        'Positive' Class : 0
##
```

Decision Tree Plot

```
library(rpart.plot)
```

```r
rpart.plot(rpartFit$finalModel)
```



Questions:

- Discuss the difference between the models and why you would use one model over the other?

Logistic Regression performed better than Decision Trees.