Caret / Recursive Partitioning

Vani P jyothy

5/13/2017

Exercise 1: caret/logistic regression (5 points)

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

```
##       tailnum year                     type                manufacturer
##    1:  N10156 2004 Fixed wing multi engine                     EMBRAER
##    2:  N102UW 1998 Fixed wing multi engine            AIRBUS INDUSTRIE
##    3:  N103US 1999 Fixed wing multi engine            AIRBUS INDUSTRIE
##    4:  N104UW 1999 Fixed wing multi engine            AIRBUS INDUSTRIE
##    5:  N10575 2002 Fixed wing multi engine                     EMBRAER
##   ---
## 3318:  N997AT 2002 Fixed wing multi engine                      BOEING
## 3319:  N997DL 1992 Fixed wing multi engine MCDONNELL DOUGLAS AIRCRAFT CO
## 3320:  N998AT 2002 Fixed wing multi engine                      BOEING
## 3321:  N998DL 1992 Fixed wing multi engine MCDONNELL DOUGLAS CORPORATION
## 3322:  N999DN 1992 Fixed wing multi engine MCDONNELL DOUGLAS CORPORATION
##           model engines seats speed    engine
##    1: EMB-145XR       2    55    NA Turbo-fan
##    2:  A320-214       2   182    NA Turbo-fan
##    3:  A320-214       2   182    NA Turbo-fan
##    4:  A320-214       2   182    NA Turbo-fan
##    5: EMB-145LR       2    55    NA Turbo-fan
##   ---
## 3318:   717-200       2   100    NA Turbo-fan
## 3319:     MD-88       2   142    NA Turbo-fan
## 3320:   717-200       2   100    NA Turbo-fan
## 3321:     MD-88       2   142    NA Turbo-jet
## 3322:     MD-88       2   142    NA Turbo-jet
```

Data set used after doing the necessary preprocessing and only selecting the required fiels

```
str(Aviation1)
```

```
## 'data.frame':    327346 obs. of  15 variables:
##  $ arr_delay     : int  11 20 33 -18 -25 12 19 -14 -8 8 ...
##  $ arr_delay_22  : num  1 1 0 1 1 1 1 1 1 1 ...
##  $ visib         : num  NA NA NA NA 10 NA 10 10 10 10 ...
##  $ precip        : num  NA NA NA NA 0 NA 0 0 0 0 ...
##  $ humid         : num  NA NA NA NA 57.3 ...
##  $ temp          : num  NA NA NA NA 39.9 ...
##  $ dep_delay     : int  2 4 2 -1 -6 -4 -5 -3 -3 -2 ...
##  $ arr_time      : int  830 850 923 1004 812 740 913 709 838 753 ...
##  $ carrier       : chr  "UA" "UA" "AA" "B6" ...
##  $ origin        : chr  "EWR" "LGA" "JFK" "JFK" ...
```

```
## $ dest          : chr  "IAH" "IAH" "MIA" "BQN" ...
## $ air_time      : int  227 227 160 183 116 150 158 53 140 138 ...
## $ sched_arr_time: int  819 830 850 1022 837 728 854 723 846 745 ...
## $ month         : int  1 1 1 1 1 1 1 1 1 1 ...
## $ distance      : int  1400 1416 1089 1576 762 719 1065 229 944 733 ...
```

Splitting the data into tesing and training data set

```r
set.seed(1)
Train <- createDataPartition(Aviation1$arr_delay_22, p=0.6, list=FALSE,times=1)
training <- Aviation1[ Train, ]
testing <- Aviation1[ -Train, ]

#Converting the output variable to a factor variable
training$arr_delay_22=factor(training$arr_delay_22)
testing$arr_delay_22=factor(testing$arr_delay_22)
```

Building the logistic regression model using the Caret package

```r
fit=train(arr_delay_22 ~ dep_delay+carrier+air_time+visib+precip+temp+humid,data=training, method="glm"
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

- Calculate the training or apparent performance of the model.

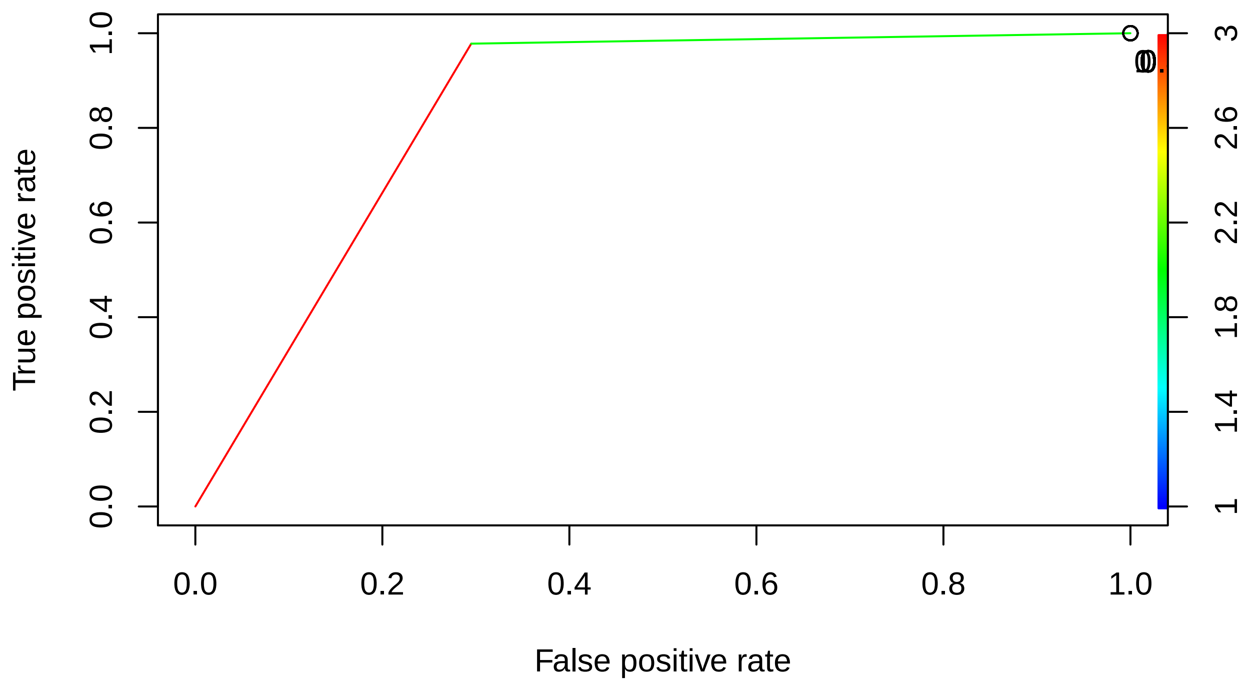
```r
#Removing the NA values in the testing data
testing1=na.omit(testing)
predicttest=predict(fit,newdata = testing1)
confusionMatrix(predicttest,testing1$arr_delay_22)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction      0      1
##          0  17713   2318
##          1   7414 103019
##
##                Accuracy : 0.9254
##                  95% CI : (0.924, 0.9268)
##     No Information Rate : 0.8074
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7401
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.7049
##             Specificity : 0.9780
##          Pos Pred Value : 0.8843
##          Neg Pred Value : 0.9329
##              Prevalence : 0.1926
##          Detection Rate : 0.1358
##    Detection Prevalence : 0.1535
##       Balanced Accuracy : 0.8415
##
##        'Positive' Class : 0
```

##

- Create a ROC Curve for your model

```
#predicttest1=predict(fit,newdata = testing1,type="response")
rocpred=prediction(as.numeric(predicttest),testing1$arr_delay_22)
rocperf=performance(rocpred,"tpr","fpr")
plot(rocperf,colorize=TRUE,print.cutoffs.at=seq(0,1,.1),text.adj=c(-0.2,1.7))
```



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:

Using rpart to train a model.

```
flight_tree=rpart(arr_delay_22 ~ air_time+visib+precip+temp+humid+dep_delay+carrier,data=training,metho
```

Using caret to train the model

```
fit_tree_caret=train(arr_delay_22 ~ dep_delay+carrier+air_time+visib+precip+temp+humid,data=training, me
```

- An articulation of the the problem your are

The problem here is to make a decision tree to predict the arrival delay of the flights data.Model is trained,so that it will predict whether a particular flight will have delay >22 min

- An naive model

Our Naive model is the most frequent outcome.That is there wont be any delay greater than 22 min

- An unbiased calculation of the performance metric

Testing the model using the testing data set using rpart function

```
predictflight=predict(flight_tree,newdata = testing1,type="class")

# Calculating the accuracy of the model

table(testing1$arr_delay_22,predictflight)
```

```
##    predictflight
##          0      1
##   0  17833   7294
##   1   2554 102783
```

```
#Accuracy of the prediction

(17833+102783)/(17833+102783+7294+2554)
```

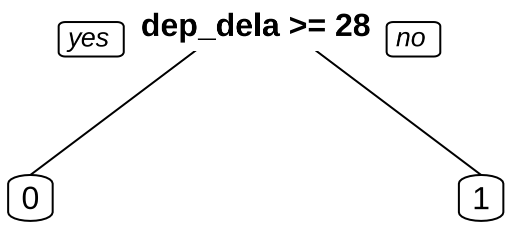
```
## [1] 0.9245156
```

```
#Accuracy of the Naive model(most frequent outcome,which is that there is no delay >22 min)
(2554+102783)/(17833+102783+7294+2554)
```

```
## [1] 0.8074028
```

- A plot of your model – (the actual tree; there are several ways to do this)

```
prp(flight_tree)
```



- A discussion of your model

The accuracy of the NAive model is 80%,where as the accuracy of the model that was trained by decision tree is 92%.Since it beats the accuracy of the Naive model,the model build is fairly good.

Testing the model using the testing data set using Caet function

```
predictflight_caret_tree=predict(fit_tree_caret,newdata = testing1)

# Calculating the accuracy of the model

table(testing1$arr_delay_22,predictflight_caret_tree)
```

```
##    predictflight_caret_tree
##          0      1
##   0  16948   8179
##   1   1657 103680
```

```
#Accuracy of the prediction

(16948+103680)/(16948+103680+8179+1657)
```

```
## [1] 0.9246076
```

```
#Accuracy of the Naive model(most frequent outcome,which is that there is no delay >22 min)
(1657+103680)/(16948+103680+8179+1657)
```

```
## [1] 0.8074028
```

Questions:

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

The accuracy is almost same using caret or rpart

- How might you produce an ROC type curve for the *rpart* model?

```
predictROC=predict(flight_tree,newdata = testing1)

head(predictROC)
```

```
##             0         1
## 7  0.06601362 0.9339864
## 8  0.06601362 0.9339864
## 9  0.06601362 0.9339864
## 10 0.06601362 0.9339864
## 11 0.06601362 0.9339864
## 12 0.06601362 0.9339864
```

```
predRoctree=prediction(predictROC[,2],testing1$arr_delay_22)

perftree=performance(predRoctree,"tpr","fpr")

plot(perftree)
```