Prediction - Comparing Trees

Xinyue Tan

2/11/2018

In this project, I will model student data using three flavors of tree algorithm: CART, C4.5 and C5.0. I will be using these algorithms to attempt to predict which students drop out of courses. Many universities have a problem with students over-enrolling in courses at the beginning of semester and then dropping most of them as they make decisions about which classes to attend. This makes it difficult to plan for the semester and allocate resources. However, schools don't want to restrict the choice of their students. One solution is to create predictions of which students are likley to drop out of which courses and use these predictions to inform semester planning.

In this project, I will be using the tree algorithms to build models of which students are likely to drop out of which classes.

Software

In order to generate our models, I will need several packages. The first package I install is caret (https://cran.r-project.org/web/packages/caret/index.html).

There are many prediction packages available and they all have slightly different syntax. caret is a package that brings all the different algorithms under one hood using the same syntax.

I will also be accessing an algorithm from the Weka suite (https://www.cs.waikato.ac.nz/~ml/weka/). Weka is a collection of machine learning algorithms that have been implemented in Java and made freely available by the University of Waikato in New Zealand. To access these algorithms, I first install both the Java Runtime Environment (JRE) and Java Development Kit (http://www.oracle.com/technetwork/java/javase/downloads/jre9-downloads-3848532.html) on my machine. Then, I install the RWeka (https://cran.r-project.org/web/packages/RWeka/index.html) package within R.

(Issue 1: failure to install RWeka/RWekajars, paste "sudo R CMD javareconf" into terminal and try to install again)

**Weka requires Java and Java causes problems.

The last package I need is C50 (https://cran.r-project.org/web/packages/C50/index.html).

Data

The data comes from a university registrar's office. The code book for the variables are available in the file code-book.txt, including the variables and their definitions.

Upload the drop-out.csv data into R as a data frame.

```
#install.packages("caret")
#install.packages("Weka Suite")
#install.packages("Java Runtime Environment (JRE) and Java Development Kit")
#install.packages("rJava")
#install.packages("RWeka")
#install.packages("C50")

D1 <- read.csv("drop-out.csv", header = TRUE)
```

Separate data set into a training set and a test set. Randomly select 25% of the students to be the test data set and leave the remaining 75% for the training data set. (Hint: each row represents an answer, not a single student.)

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# sample: randomly choose from the dataframe
sample<-data.frame(sample(unique(D1$student_id),511))

# Rename
names(sample)<-c("student_id")

train1 <- inner_join(D1, sample, by = "student_id")
test1<- anti_join(D1,sample, by = "student_id")
```

For this project, I will be predicting the student level variable “complete”.

Visualize the relationships between the chosen variables as a scatterplot matrix. Save your image as a .pdf named scatterplot_matrix.pdf. Based on this visualization do you see any patterns of interest? Why or why not?

CART Trees

I used the rpart package (https://cran.r-project.org/web/packages/rpart/rpart.pdf) to generate CART tree

models.

Construct a classification tree that predicts complete using the caret package.

Train the CART Trees model

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.5.3
```

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

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

```
## Warning: package 'ggplot2' was built under R version 3.5.3
```

```
# Remove the student_id variable that we do not want to use in the model
TRAIN2 <- train1[,c(2:10)]
# c means select; [row, column]


#k- fold cross validation: the original sample is randomly partitioned into k equal si
zed subsamples. Of the k subsamples, a single subsample is retained as the validation
data for testing the model, and the remaining subsamples are used as training data. Th
e cross-validation process is then repeated k times, with each of the k subsamples use
d exactly once as the validation data. The k results can then be averaged to produce
a single estimation. The advantage of this method over repeated random sub-sampling (s
ee below) is that all observations are used for both training and validation, and eac
h observation is used for validation exactly once.

#Define the control elements we would like to use
ctrl <- trainControl(method = "repeatedcv", #Tell caret to perform 10-fold cross valid
ation; 10-fold cross validation is commonly used
                repeats = 3, #Tell caret to repeat each fold three times
                classProbs = TRUE, #Calculate class probabilities for ROC calculation
                summaryFunction = twoClassSummary)

#Define the model
cartFit <- train(complete ~ ., #Define which variable to predict
                data = TRAIN2, #Define the data set to train the model on
                trControl = ctrl, #Tell caret the control elements
                method = "rpart", #Define the model type; rpart: tree chooses the opti
mal fit at each leaf, not the overall best fit for the data-> therefore, there is a da
nger of overfitting (tree is too specific to training data to be able to predict new d
ata, need to stop the tree at a certain number of nodes or prune)

                metric = "ROC", #Tell caret to calculate the ROC curve; ROC is the sho
rtcut of receiver operating characteristic-> goal: select possibly optimal models and
to discard suboptimal ones independly from the cost context or the class distributio
n. ROC is related to cost/benefit analysis of diagonostic decision making.
                preProc = c("center", "scale")) #Center and scale the data to minimiz
e the

#Check the results
cartFit
```

```
## CART
##
## 4502 samples
##    8 predictor
##    2 classes: 'no', 'yes'
##
## Pre-processing: centered (8), scaled (8)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 4052, 4052, 4051, 4052, 4052, 4051, ...
## Resampling results across tuning parameters:
##
##   cp          ROC        Sens       Spec
##   0.01152074  0.8840000  0.6563633  0.9965625
##   0.03725038  0.8248546  0.5995733  0.9978125
##   0.57066052  0.6320513  0.2641026  1.0000000
##
## ROC was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.01152074.
```

```
#Plot ROC against complexity
pdf("cartfit.pdf")
plot(cartFit)
dev.off()
```

```
## png
##   2
```

Analyze the important model attribues of the tree and determine the success of the student performance model

1. Model attributes mentioned in the code: *The cp is the “minimum benefit” that a split must add to the tree, which means in this model, if the split doesn’t yield 0.0096, rpart()does not add it. Thus, the final value is at ROC=0.8974, and the cp=0.0096; It is not a successful model since there is a danger of overfitting (e.g.: tree is too specific to training data to be able to predict new data, need to stop the tree at a certain number of nodes or prune).
2. Analyze the plot and underlying information: *The plot represents the relationship between complexity parameter (e.g.:used to control the size of the decision tree and to select the optimal tree size) and ROC (e.g.: cross-validated accuracy). The more complex the decision tree is, the more accurate the prediciton is .

Test the CART Trees model

Now predict results from the test data.

```
#Remove the student_id variable that we do not want to use in the model
TEST2 <- test1[,c(2:10)]

#Generate prediction using previously trained model
cartClasses <- predict(cartFit, newdata = TEST2)

#Generate model statistics
confusionMatrix(data = cartClasses, TEST2$complete)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no  275   6
##        yes 147 931
##
##                Accuracy : 0.8874
##                  95% CI : (0.8694, 0.9037)
##     No Information Rate : 0.6895
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7105
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.6517
##             Specificity : 0.9936
##          Pos Pred Value : 0.9786
##          Neg Pred Value : 0.8636
##              Prevalence : 0.3105
##          Detection Rate : 0.2024
##    Detection Prevalence : 0.2068
##       Balanced Accuracy : 0.8226
##
##        'Positive' Class : no
##
```

Analyze the important model attribues of the tree and determine the success of the student performance model

*The accuracy is 0.8894, which means that the decision tree can predict 88.9% of the test data correctly. The p-value is small, which indicates the result rejects the hypothesis and is statistically significant.

C4.5-Type Trees

I will now repeat the same prediction but using a different tree-based algorithm called J48 (). J48 is a

Java implementation of the C4.5 decsion tree algorithm of Quinlan (1993) ().

Train C4.5-Type Trees model

Train the J48 model on the same training data and examine the results.

```
ctrl<-trainControl(method="repeatedcv", repeats=3,classProbs = TRUE,summaryFunction =
twoClassSummary)
J48fit<-train(complete ~.,data=TRAIN2, method="J48",preProc=c("center","scale"), trCon
trol=ctrl,metric="ROC" )

J48fit
```

```
## C4.5-like Trees
##
## 4502 samples
##    8 predictor
##    2 classes: 'no', 'yes'
##
## Pre-processing: centered (8), scaled (8)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 4052, 4051, 4052, 4052, 4052, 4052, ...
## Resampling results across tuning parameters:
##
##   C      M  ROC        Sens       Spec
##   0.010  1  0.8380984  0.6651008  0.9965625
##   0.010  2  0.8380984  0.6651008  0.9965625
##   0.010  3  0.8380984  0.6651008  0.9965625
##   0.255  1  0.8490087  0.6679096  0.9940625
##   0.255  2  0.8482787  0.6697005  0.9943750
##   0.255  3  0.8467879  0.6686827  0.9946875
##   0.500  1  0.9003782  0.6842885  0.9801042
##   0.500  2  0.9028295  0.6837757  0.9802083
##   0.500  3  0.9051850  0.6832648  0.9822917
##
## ROC was used to select the optimal model using the largest value.
## The final values used for the model were C = 0.5 and M = 3.
```

```
pdf("J48fit.pdf")
plot(J48fit)
dev.off()
```

```
## png
##   2
```

Analyze the important model attribues of the tree and determine the success of the student

performance model

1. Model attributes mentioned in the code: *Data preprocessing manipulates data before training them by centering and scaling. Trcontrol defines how this funciton acts. Metric which is a measurement of the model performance is ROC. The final values used for the model were C = 0.5 and M = 3, which has the highest validity among all of the alternatives. It performs well on the training set since it has over 90% of the accuracy.
2. Analyze the plot and underlying information: *It represents the relationship between condidence threshold and ROC validity. When condidence threshold reach the turning point, the validity grows rapidly.

Test C4.5-Type Trees model

Now test new J48 model by predicting the test data and generating model fit statistics.

```
J48classes<-predict(J48fit, newdata=TEST2)
confusionMatrix(data=J48classes,TEST2$complete)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no  286  20
##        yes 136 917
##
##                Accuracy : 0.8852
##                  95% CI : (0.8671, 0.9017)
##     No Information Rate : 0.6895
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.71
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.6777
##             Specificity : 0.9787
##          Pos Pred Value : 0.9346
##          Neg Pred Value : 0.8708
##              Prevalence : 0.3105
##          Detection Rate : 0.2104
##    Detection Prevalence : 0.2252
##       Balanced Accuracy : 0.8282
##
##        'Positive' Class : no
##
```

Analyze the important model attribues of the tree and determine the success of the student

performance model

*The accuracy is 0.8869, which is high. The p-value is low, which means that the prediction model is statistically significant.

Compare the models

caret allows us to compare all three models at once.

```
resamps <- resamples(list(cart = cartFit, jfoureight = J48fit))
summary(resamps)
```

```
##
## Call:
## summary.resamples(object = resamps)
##
## Models: cart, jfoureight
## Number of resamples: 30
##
## ROC
##                 Min.   1st Qu.    Median     Mean   3rd Qu.      Max. NA's
## cart       0.8413942 0.8684886 0.8822269 0.884000 0.8996875 0.9303435    0
## jfoureight 0.8348798 0.8932437 0.9089724 0.905185 0.9177897 0.9459135    0
##
## Sens
##                 Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## cart       0.5538462 0.6259542 0.6538462 0.6563633 0.6884615 0.7862595
## jfoureight 0.6076923 0.6480769 0.6807692 0.6832648 0.7178802 0.7615385
##            NA's
## cart          0
## jfoureight    0
##
## Spec
##                Min.   1st Qu.   Median      Mean 3rd Qu.    Max. NA's
## cart       0.990625 0.9945313 0.996875 0.9965625  1.0000 1.00000    0
## jfoureight 0.956250 0.9781250 0.984375 0.9822917  0.9875 0.99375    0
```

Analyze model summary and determine the best model:

*J48 is the optimal prediction model. Since the ROC of J48 is slightly better than Cart tree model.

Which variables (features) within the chosen model are important, do these features provide insights that may be useful in solving the problem of students dropping out of courses?

From quanlitative perspectives, variables such as “course taken” and “online” are important. When students choose more courses than they can handle, they are highly likely to drop the course. Another explanation is that they choose more courses on purpose at the first place and drop the ones they do

not like. For students who are not familiar with the teaching mode of online course, students may not expect to read study materials by themselves and self-learning. Thus, those students might end up dropping the course.