AlunosUFCG

Rodolfo Viana

13-06-2015

Durante os últimos anos a Universidade Federal de Campina Grande observa um número alto de evasão por parte dos alunos. Tentando entender os motivos dessa evasão analisamos uma amostra contendo dados importante. A nossa amostra tem os seguintes atributos:

1. MATRICULA: identificador do aluno
2. PERIODO: identificador do período letivo da universidade (ano.semestre)
3. COD_CURSO: identificador do curso
4. CURSO: nome do curso. Cada curso tem seu COD_CURSO
5. CODIGO: identificador da disciplina que o aluno cursou no periodo
6. DISCIPLINA: nome da disciplina referente que o aluno cursou no periodo.
7. CODIGO: Cada disciplina tem seu
8. CREDITOS: numero de créditos referente a disciplina
9. DEPARTAMENTO: departamento que ofertou a disciplina
10. MEDIA: média do aluno na disciplina (0 a 10). Alunos reprovados por falta numa disciplina recebem 0 e alunos que trancaram a disciplina recebem NA.
11. STATUS: Aprovado, Reprovado Por Falta, Trancado ou Reprovado. Se refere ao estado final do aluno na disciplina
12. PERIODO_INGRESSO: período letivo da universidade em que o aluno ingressou no curso.
13. PERIODO_RELATIVO: número de períodos que o aluno está matriculado na universidade. “1” refere-se ao aluno em seu primeiro periodo, “5” refere-se ao aluno no quinto período.
14. COD_EVASAO: identificador de evasão do aluno. “0” significa que o aluno continuou ativo na universidade no período seguinte e “1” significa que o aluno desistiu do curso nesse período e não voltou a se matricular no seguinte.

O nosso objetivo é construir um modelo de classificação que nos diga se o aluno irá evadir ou não. Para o nosso primeiro modelo vamos classificar apenas para os alunos que tem período relativo 1.

```
library(dplyr)

arquivo <- read.csv("~/Projetos/DataAnalysis/Assignment5/training_sem_acento.csv")

#Transformação para factor
arquivo$COD_EVASAO <- as.factor(arquivo$COD_EVASAO)
arquivo$DEPARTAMENTO <- as.factor(arquivo$DEPARTAMENTO)
arquivo$COD_CURSO <- as.factor(arquivo$COD_CURSO)
arquivo$MEDIA <- as.character(arquivo$MEDIA)
arquivo$MEDIA[is.na(arquivo$MEDIA)] <- -10
arquivo$MEDIA <- as.numeric(arquivo$MEDIA)

#Missing
levels(arquivo$DEPARTAMENTO)[1] = NA
levels(arquivo$DISCIPLINA)[1] = NA
levels(arquivo$SITUACAO)[1] = NA
```

Antes de criar o modelo é importante dividir o arquivo original em treino e teste (75% treinamento, 25% teste), para assim verificar o F-measure e saber se um modelo criado é melhor do que o modelo anterior.

```
#Primeiro periodo
set.seed(12345)
primeiro_periodo <- filter(arquivo, PERIODO_RELATIVO == 1)
primeiro_periodo <- primeiro_periodo[order(runif(nrow(primeiro_periodo))), ]

#Divisao de treino e teste
treino <- primeiro_periodo[1:round(0.75*nrow(primeiro_periodo)), ]
test <- primeiro_periodo[round(0.75*nrow(primeiro_periodo)):nrow(primeiro_perio
do), ]
```

Podemos notar que a proporção entre evasão e não evasão se manteve parecida após a divisão de treino e teste.

```
prop.table(table(primeiro_periodo$COD_EVASAO))
```

```
##
##         0         1
## 0.8936799 0.1063201
```

```
prop.table(table(treino$COD_EVASAO))
```

```
##
##         0         1
## 0.8924985 0.1075015
```

```
prop.table(table(test$COD_EVASAO))
```

```
##
##         0         1
## 0.8972542 0.1027458
```

Inicialmente vamos criar um modelo considerando apenas o período letivo da universidade, código da disciplina cursada, departamento e a situação.

```
library("C50")

model <- C5.0(treino[,c(5,7,9,11)], treino$COD_EVASAO)
model
```

```
##
## Call:
## C5.0.default(x = treino[, c(5, 7, 9, 11)], y = treino$COD_EVASAO)
##
## Classification Tree
## Number of samples: 10158
## Number of predictors: 4
##
## Tree size: 3
##
## Non-standard options: attempt to group attributes
```

```
summary(model)
```

```
##
## Call:
## C5.0.default(x = treino[, c(5, 7, 9, 11)], y = treino$COD_EVASAO)
##
##
## C5.0 [Release 2.07 GPL Edition]      Thu Jun 18 00:59:04 2015
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 10158 cases (5 attributes) from undefined.data
##
## Decision tree:
##
## SITUACAO in {0,Aprovado,Reprovado}: 0 (8767/297)
## SITUACAO in {Reprovado por Falta,Trancado}:
## :...PERIODO <= 2006.1: 0 (357/116)
##     PERIODO > 2006.1: 1 (1034/355)
##
##
## Evaluation on training data (10158 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##       3  768( 7.6%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##    8711   355    (a): class 0
##     413   679    (b): class 1
##
##
##  Attribute usage:
##
##  100.00% SITUACAO
##   13.69% PERIODO
##
##
## Time: 0.3 secs
```

```
pred <- predict(model, test[,c(5,7,9,11)])

true_eva <- test$COD_EVASAO == 1
table(pred, true_eva)
```

```
##     true_eva
## pred FALSE TRUE
##    0  2907  125
##    1   132  223
```

Podemos notar que o atributo departamento não tem importância para a criação desse modelo, assim como código da disciplina. Para o primeiro modelo temos: F-measure = 0.61

Tentando melhorar esse valor do F-measure, vamos criamos um novo modelo adicionando o cod_curso e média

```
model <- C5.0(treino[,c(3,5,7,9,10,11)], treino$COD_EVASAO)
model
```

```
##
## Call:
## C5.0.default(x = treino[, c(3, 5, 7, 9, 10, 11)], y = treino$COD_EVASAO)
##
## Classification Tree
## Number of samples: 10158
## Number of predictors: 6
##
## Tree size: 3
##
## Non-standard options: attempt to group attributes
```

```
summary(model)
```

```
##
## Call:
## C5.0.default(x = treino[, c(3, 5, 7, 9, 10, 11)], y = treino$COD_EVASAO)
##
##
## C5.0 [Release 2.07 GPL Edition]      Thu Jun 18 00:59:09 2015
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 10158 cases (7 attributes) from undefined.data
##
## Decision tree:
##
## SITUACAO in {0,Aprovado,Reprovado}: 0 (8767/297)
## SITUACAO in {Reprovado por Falta,Trancado}:
## :...PERIODO <= 2006.1: 0 (357/116)
##     PERIODO > 2006.1: 1 (1034/355)
##
##
## Evaluation on training data (10158 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##       3  768( 7.6%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##    8711   355    (a): class 0
##     413   679    (b): class 1
##
##
##  Attribute usage:
##
##  100.00% SITUACAO
##   13.69% PERIODO
##
##
## Time: 0.4 secs
```

```
pred <- predict(model, test[,c(3,5,7,9,10,11)])

true_eva <- test$COD_EVASAO == 1
table(pred, true_eva)
```

```
##     true_eva
## pred FALSE TRUE
##    0  2907  125
##    1   132  223
```

É possivel notar que mesmo adicionando novas variavéis o F-measure não aumentou. Por essa razão criamos novos atributos:

1. Média geral do período do aluno
2. Quantidade de cadeiras aprovadas
3. Quantidade de reprovações
4. Quantidade de reprovações por falta

```
#Criando novos atributos para o treino
treino2 <- treino
treino_group <- group_by(treino, MATRICULA)

media <- summarise(treino_group, mean(MEDIA))
names(media) <- c("MATRICULA", "MEDIATOTAL")

reprovacao <- summarise(group_by(filter(treino, SITUACAO=="Reprovado"), MATRICU
LA), n())
names(reprovacao) <- c("MATRICULA", "REPROVACAO")

aprovado <- summarise(group_by(filter(treino, SITUACAO=="Aprovado"), MATRICUL
A), n())
names(aprovado) <- c("MATRICULA", "APROVADO")

reprovado_falta <- summarise(group_by(filter(treino, SITUACAO=="Reprovado por F
alta"), MATRICULA), n())
names(reprovado_falta) <- c("MATRICULA", "REPROVADOFALTA")

treino2 <- merge(treino2, media, by = "MATRICULA", all = TRUE)
treino2 <- merge(treino2, reprovacao, by = "MATRICULA", all = TRUE)
treino2 <- merge(treino2, aprovado, by = "MATRICULA", all = TRUE)
treino2 <- merge(treino2, reprovado_falta, by = "MATRICULA", all = TRUE)

#Criando novos atributos para o teste
test2 <- test
test_group <- group_by(test, MATRICULA)

media <- summarise(test_group, mean(MEDIA))
names(media) <- c("MATRICULA", "MEDIATOTAL")
test2 <- merge(test2, media, by = "MATRICULA", all = TRUE)

reprovacao <- summarise(group_by(filter(test, SITUACAO=="Reprovado"), MATRICUL
A), n())
names(reprovacao) <- c("MATRICULA", "REPROVACAO")
test2 <- merge(test2, reprovacao, by = "MATRICULA", all = TRUE)

aprovado <- summarise(group_by(filter(test, SITUACAO=="Aprovado"), MATRICULA),
n())
names(aprovado) <- c("MATRICULA", "APROVADO")
test2 <- merge(test2, aprovado, by = "MATRICULA", all = TRUE)

reprovado_falta <- summarise(group_by(filter(test, SITUACAO=="Reprovado por Fal
ta"), MATRICULA), n())
names(reprovado_falta) <- c("MATRICULA", "REPROVADOFALTA")
test2 <- merge(test2, reprovado_falta, by = "MATRICULA", all = TRUE)
```

Agora com as 4 novas colunas criadas temos mais dados para analisar e ajudar no classificador. Porém ao longo do processo percebenos que a matrix de confusão da nossa versão final é pior da versão com sem a quantidade de aprovações e reprovações por falta. Observe:

```
model <- C5.0(treino2[,c(3,5,7,9,10,11,15,16,17,18)], treino2$COD_EVASAO)
pred <- predict(model, test2[,c(3,5,7,9,10,11,15,16,17,18)])

true_eva <- test2$COD_EVASAO == 1
table(pred, true_eva)
```

```
##     true_eva
## pred FALSE TRUE
##    0  2903  112
##    1   136  236
```

Modelo com matrix de confusão melhor:

```
model <- C5.0(treino2[,c(3,5,7,9,10,11,15,16)], treino2$COD_EVASAO)
pred <- predict(model, test2[,c(3,5,7,9,10,11,15,16)])

true_eva <- test2$COD_EVASAO == 1
table(pred, true_eva)
```

```
##     true_eva
## pred FALSE TRUE
##    0  2912  103
##    1   127  245
```

Esse foi o modelo que foi usado para fazer a classificação e submissão para o kaggle.