Machine Learning with groups

Ana Real

June 4, 2019

Read data

```
library(xlsx)
library(ggpubr)
```

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

```
## Loading required package: magrittr
```

```
library(DescTools)
library(xtable)
library(caret)
```

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

```
##
## Attaching package: 'caret'
```

```
## The following objects are masked from 'package:DescTools':
##
##     MAE, RMSE
```

```
library(DMwR)
```

```
## Loading required package: grid
```

```
library(ggcorrplot)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(pROC)
```

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

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

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

```
data <- read.xlsx2("../final.xlsx", sheetIndex = 1)

for(i in c(1,2,4:10)){
    data[,i] <- as.numeric(as.character(data[,i]))
}

colnames(data) <- c("Year", "Rank", "City", "Overall", "Rankings", "Student.Mix", "Desirability", "Empl

# Adding variable for top universities
```

```
data$Top <- data$Rank

for (i in 1:376){
    if (data$Rank[i] <= 10){
        data$Top[i] <- "Top10"
    } else if (data$Rank[i] > 10 & data$Rank[i] <= 20){
        data$Top[i] <- "Top20"
    } else if (data$Rank[i] > 20 & data$Rank[i] <= 30){
        data$Top[i] <- "Top30"
    } else if (data$Rank[i] > 30 & data$Rank[i] <= 40){
        data$Top[i] <- "Top40"
    } else if (data$Rank[i] > 40 & data$Rank[i] <= 50){
        data$Top[i] <- "Top50"
    } else if (data$Rank[i] > 50 & data$Rank[i] <= 60){
        data$Top[i] <- "Top60"
    } else if (data$Rank[i] > 60 & data$Rank[i] <= 70){
        data$Top[i] <- "Top70"
    } else if (data$Rank[i] > 70 & data$Rank[i] <= 80){
        data$Top[i] <- "Top80"
    } else if (data$Rank[i] > 80 & data$Rank[i] <= 90){
        data$Top[i] <- "Top90"
    } else {
        data$Top[i] <- "Top100"
    }
}

data$Top <- as.factor(data$Top)
str(data)
```

```
## 'data.frame':    376 obs. of  11 variables:
##  $ Year             : num  2018 2018 2018 2018 2018 ...
##  $ Rank             : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ City             : Factor w/ 110 levels "Aberdeen","Adelaide",..: 50 99 56 62 76 65 12 110 96 89
##  $ Overall          : num  482 479 476 467 463 461 457 454 453 449 ...
##  $ Rankings         : num  100 84 68 57 93 54 49 63 64 93 ...
##  $ Student.Mix      : num  92 55 100 94 80 74 75 83 97 67 ...
##  $ Desirability     : num  80 97 91 89 80 89 88 94 95 67 ...
##  $ Employer.Activity: num  93 100 86 80 88 78 80 90 84 92 ...
##  $ Affordability    : num  25 54 33 47 38 67 71 42 23 44 ...
##  $ Student.View     : num  92 89 98 100 84 99 94 82 90 86 ...
##  $ Top              : Factor w/ 10 levels "Top10","Top100",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
cat2numt <- function(x){
    x <- as.character(x)
    for (i in 1:108){
        if (x[i]=="Top10"){
            x[i] <- 10
        } else if (x[i]=="Top20"){
            x[i] <- 20
        } else if (x[i]=="Top30"){
            x[i] <- 30
        } else if (x[i]=="Top40"){
            x[i] <- 40
        } else if (x[i]=="Top50"){
            x[i] <- 50
        } else if (x[i]=="Top60"){
            x[i] <- 60
        } else if (x[i]=="Top70"){
            x[i] <- 70
        } else if (x[i]=="Top80"){
            x[i] <- 80
        } else if (x[i]=="Top90"){
            x[i] <- 90
        } else {
            x[i] <- 100
        }
    }
    x <- as.numeric(x)
    x
}
```

Preparing data

```
set.seed(112)
inTrain <- createDataPartition(y=data$Top, times = 1, p = 0.7, list = FALSE)
training <- data[inTrain,]
testing <- data[-inTrain,]
dim(training)
```

```
## [1] 264  11
```

```
dim(testing)
```

```
## [1] 112  11
```

```
train_control <- trainControl(method="repeatedcv", number=10, repeats=3)
```

Decision Trees

```
trees <- train(Top~Student.View+Employer.Activity+Desirability+Rankings+Student.Mix+Affordability+facto
pred_trees <- predict(trees,testing)
confusionMatrix(pred_trees, testing$Top)$table
```

```
##           Reference
## Prediction Top10 Top100 Top20 Top30 Top40 Top50 Top60 Top70 Top80 Top90
##     Top10     12      0    10     3     2     0     0     0     0     0
##     Top100     0      0     0     0     0     0     0     0     0     0
##     Top20      1      1     5     7     6     4     4     1     1     0
##     Top30      0      0     0     0     0     0     0     0     0     0
##     Top40      0      0     0     0     0     0     0     0     0     0
##     Top50      2      5     0     5     7    11     5     8     6     6
##     Top60      0      0     0     0     0     0     0     0     0     0
##     Top70      0      0     0     0     0     0     0     0     0     0
##     Top80      0      0     0     0     0     0     0     0     0     0
##     Top90      0      0     0     0     0     0     0     0     0     0
```

```
confusionMatrix(pred_trees, testing$Top)$overall[1]
```

```
## Accuracy
##     0.25
```

```
print(trees)
```

```
## CART
##
## 264 samples
##   7 predictors
##  10 classes: 'Top10', 'Top100', 'Top20', 'Top30', 'Top40', 'Top50', 'Top60', 'Top70', 'Top80', 'Top9
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 238, 239, 235, 239, 237, 237, ...
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa
##   0.05240175  0.2222431  0.10296947
##   0.06550218  0.2092403  0.08837407
##   0.09606987  0.1699610  0.04991102
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.05240175.
```

```
testing_Top <- cat2numt(testing$Top)
```

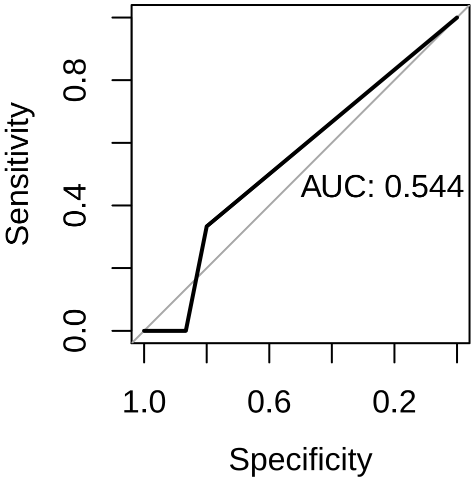
```
## Warning in cat2numt(testing$Top): NAs introduced by coercion
```

```
# Trees
pred_trees <- cat2numt(pred_trees)
```

```
## Warning in cat2numt(pred_trees): NAs introduced by coercion
```

```
# Testing set
plot.roc(testing_Top, pred_trees, print.auc=TRUE)
```

```
## Warning in roc.default(x, predictor, plot = TRUE, ...): 'response' has
## more than two levels. Consider setting 'levels' explicitly or using
## 'multiclass.roc' instead
```



Random Forest

```
randf <- train(Top~Student.View+Employer.Activity+Desirability+Rankings+Student.Mix+Affordability+facto
pred_randf <- predict(randf,testing)
confusionMatrix(pred_randf, testing$Top)$table
```

```
##           Reference
## Prediction Top10 Top100 Top20 Top30 Top40 Top50 Top60 Top70 Top80 Top90
##     Top10     13      0     3     1     1     0     0     0     0     0
##     Top100     0      4     0     0     0     0     0     0     2     4
##     Top20      2      0    12     4     0     0     0     0     0     0
##     Top30      0      0     0     7     5     1     0     0     0     0
##     Top40      0      0     0     3     5     7     1     0     0     0
##     Top50      0      0     0     0     4     6     4     0     0     0
##     Top60      0      0     0     0     0     1     2     0     1     0
##     Top70      0      0     0     0     0     0     1     4     3     0
##     Top80      0      1     0     0     0     0     1     4     1     1
##     Top90      0      1     0     0     0     0     0     1     0     1
```

```
confusionMatrix(pred_randf, testing$Top)$overall[1]
```

```
##  Accuracy
## 0.4910714
```

```
print(randf)
```

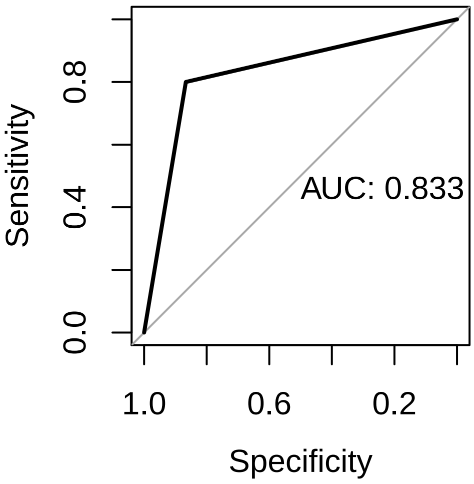
```
## Random Forest
##
## 264 samples
##   7 predictors
##  10 classes: 'Top10', 'Top100', 'Top20', 'Top30', 'Top40', 'Top50', 'Top60', 'Top70', 'Top80', 'Top9
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 238, 238, 236, 238, 238, 237, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.4190824  0.3458949
##    6    0.4464026  0.3769584
##   10    0.4226397  0.3502275
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 6.
```

```
pred_randf <- cat2numt(pred_randf)
```

```
## Warning in cat2numt(pred_randf): NAs introduced by coercion
```

```
plot.roc(testing_Top, pred_randf, print.auc=TRUE)
```

```
## Warning in roc.default(x, predictor, plot = TRUE, ...): 'response' has
## more than two levels. Consider setting 'levels' explicitly or using
## 'multiclass.roc' instead
```



Support Vector Machine Linear

```
svml <- train(Top~Student.View+Employer.Activity+Desirability+Rankings+Student.Mix+Affordability+factor
pred_svml <- predict(svml,testing)
confusionMatrix(pred_svml, testing$Top)$table
```

```
##           Reference
## Prediction Top10 Top100 Top20 Top30 Top40 Top50 Top60 Top70 Top80 Top90
##     Top10     15      0     1     0     0     0     0     0     0     0
##     Top100     0      4     0     0     0     0     0     0     0     3
##     Top20      0      0    12     1     0     0     0     0     0     0
##     Top30      0      0     2    10     3     0     0     0     0     0
##     Top40      0      0     0     4     9     6     0     0     0     0
##     Top50      0      0     0     0     3     8     3     0     0     0
##     Top60      0      0     0     0     0     1     3     2     1     0
##     Top70      0      0     0     0     0     0     2     4     1     0
##     Top80      0      0     0     0     0     0     1     3     5     1
##     Top90      0      2     0     0     0     0     0     0     0     2
```

```
confusionMatrix(pred_svml, testing$Top)$overall[1]
```

```
##  Accuracy
## 0.6428571
```

```
print(svml)
```

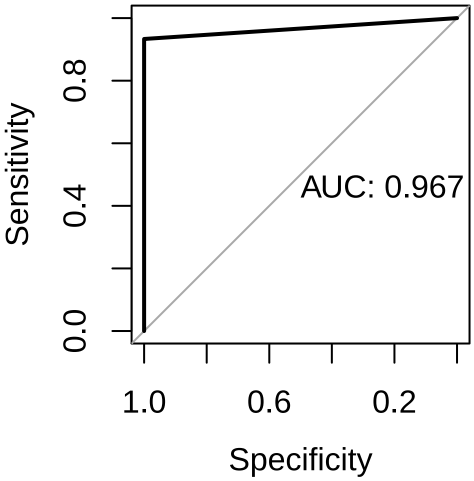
```
## Support Vector Machines with Linear Kernel
##
## 264 samples
##   7 predictors
##  10 classes: 'Top10', 'Top100', 'Top20', 'Top30', 'Top40', 'Top50', 'Top60', 'Top70', 'Top80', 'Top9
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 236, 237, 238, 238, 241, 235, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.6232555  0.5753398
##
## Tuning parameter 'C' was held constant at a value of 1
```

```
pred_svml <- cat2numt(pred_svml)
```

```
## Warning in cat2numt(pred_svml): NAs introduced by coercion
```

```
plot.roc(testing_Top, pred_svml, print.auc=TRUE)
```

```
## Warning in roc.default(x, predictor, plot = TRUE, ...): 'response' has
## more than two levels. Consider setting 'levels' explicitly or using
## 'multiclass.roc' instead
```



Support Vector Machine Radial

```
svmr <- train(Top~Student.View+Employer.Activity+Desirability+Rankings+Student.Mix+Affordability+factor
pred_svmr <- predict(svmr,testing)
confusionMatrix(pred_svmr, testing$Top)$table
```

```
##           Reference
## Prediction Top10 Top100 Top20 Top30 Top40 Top50 Top60 Top70 Top80 Top90
##     Top10     13      0     3     0     0     0     0     0     0     0
##     Top100     0      0     0     0     0     0     0     0     0     2
##     Top20      2      0    11     4     0     0     0     0     0     0
##     Top30      0      0     1     7     5     4     0     0     0     0
##     Top40      0      0     0     3     7     4     1     0     0     0
##     Top50      0      0     0     1     3     7     5     1     0     0
##     Top60      0      0     0     0     0     0     0     2     1     0
##     Top70      0      0     0     0     0     0     2     2     2     0
##     Top80      0      2     0     0     0     0     1     3     3     4
##     Top90      0      4     0     0     0     0     0     1     1     0
```

```
confusionMatrix(pred_svmr, testing$Top)$overall[1]
```

```
##  Accuracy
## 0.4464286
```

```
print(svmr)
```

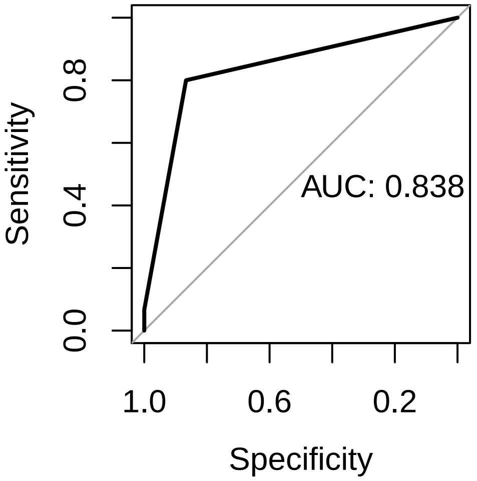
```
## Support Vector Machines with Radial Basis Function Kernel
##
## 264 samples
##   7 predictors
##  10 classes: 'Top10', 'Top100', 'Top20', 'Top30', 'Top40', 'Top50', 'Top60', 'Top70', 'Top80', 'Top9
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 240, 240, 237, 234, 237, 239, ...
## Resampling results across tuning parameters:
##
##   C     Accuracy   Kappa
##   0.25  0.2623707  0.1563522
##   0.50  0.3170469  0.2245263
##   1.00  0.4045582  0.3280646
##
## Tuning parameter 'sigma' was held constant at a value of 0.07752212
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were sigma = 0.07752212 and C = 1.
```

```
pred_svmr <- cat2numt(pred_svmr)
```

```
## Warning in cat2numt(pred_svmr): NAs introduced by coercion
```

```
plot.roc(testing_Top, pred_svmr, print.auc=TRUE)
```

```
## Warning in roc.default(x, predictor, plot = TRUE, ...): 'response' has
## more than two levels. Consider setting 'levels' explicitly or using
## 'multiclass.roc' instead
```



Neural Networks

```
# Neural Networks
nbc <- train(Top~Student.View+Employer.Activity+Desirability+Rankings+Student.Mix+Affordability+factor(
pred_nbc <- predict(nbc,testing)
confusionMatrix(pred_nbc, testing$Top)$table
```

```
##              Reference
## Prediction Top10 Top100 Top20 Top30 Top40 Top50 Top60 Top70 Top80 Top90
##     Top10      8      0     9     6     3     1     1     0     0     0
##     Top100     0      0     0     0     0     0     0     0     3     0
##     Top20      0      0     1     1     1     0     0     0     0     0
##     Top30      2      0     1     5     6     4     2     0     0     0
##     Top40      5      0     3     0     3     2     0     0     0     0
##     Top50      0      2     1     2     2     5     1     1     2     1
##     Top60      0      0     0     1     0     2     1     2     1     1
##     Top70      0      0     0     0     0     1     1     1     0     0
##     Top80      0      3     0     0     0     0     2     5     1     2
##     Top90      0      1     0     0     0     0     1     0     0     2
```

```
confusionMatrix(pred_nbc, testing$Top)$overall[1]
```

```
##  Accuracy
## 0.2410714
```

```
print(nbc)
```

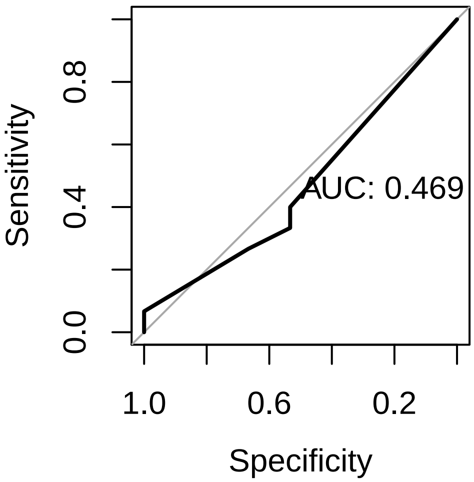
```
## Neural Network
##
## 264 samples
##   7 predictors
##  10 classes: 'Top10', 'Top100', 'Top20', 'Top30', 'Top40', 'Top50', 'Top60', 'Top70', 'Top80', 'Top9
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 236, 236, 238, 237, 238, 239, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.2524572  0.1536616
##
## Tuning parameter 'size' was held constant at a value of 10
##
## Tuning parameter 'decay' was held constant at a value of 0.1
```

```
pred_nbc <- cat2numt(pred_nbc)
```

```
## Warning in cat2numt(pred_nbc): NAs introduced by coercion
```

```
plot.roc(testing_Top, pred_nbc, print.auc=TRUE)
```

```
## Warning in roc.default(x, predictor, plot = TRUE, ...): 'response' has
## more than two levels. Consider setting 'levels' explicitly or using
## 'multiclass.roc' instead
```


AUC: 0.469
Sensitivity
0.0
0.4
0.8
1.0
0.6
0.2
Specificity