R Notebook

Diagnostic Metrics Course: HUDK 4050, Week 10

Author: Guotai Sun

Assignment: ICE7

Objectives:

At the end of this ICE, you will be able to:

1.Identify the correct model diagnostic metric(s) for performance

2.Implement at least one model diagnostic metric for a model you have built for ACA2.

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.5
```

```
## -- Attaching packages ---------------------------------------
tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
## Warning: package 'tibble' was built under R version 4.0.5
```

```
## Warning: package 'tidyr' was built under R version 4.0.5
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
train <- read_csv("aca2_dataset_training.csv")
```

```
##
## -- Column specification --------------------------------------------
------------
## cols(
##   UNIQUEID = col_double(),
##   SCHOOL = col_character(),
##   Class = col_character(),
##   GRADE = col_double(),
##   CODER = col_character(),
##   STUDENTID = col_double(),
##   Gender = col_double(),
##   OBSNUM = col_double(),
##   `totalobs-forsession` = col_double(),
##   Activity = col_character(),
##   ONTASK = col_character(),
##   TRANSITIONS = col_double(),
##   NumACTIVITIES = col_double(),
##   FORMATchanges = col_double(),
##   NumFORMATS = col_double(),
##   `Obsv/act` = col_double(),
##   `Transitions/Durations` = col_double(),
##   `Total Time` = col_double()
## )

train

## # A tibble: 22,184 x 18
##    UNIQUEID SCHOOL Class GRADE CODER STUDENTID Gender OBSNUM
`totalobs-forsessi~
##       <dbl> <chr>  <chr> <dbl> <chr>     <dbl>  <dbl>  <dbl>
<dbl>
##  1    14400 B      T9Q       0 Z        600865      0      1
1
##  2    14401 B      T9Q       0 Z        596466      0      1
1
##  3    14402 B      T9Q       0 Z        616590      0      1
2
##  4    14403 B      T9Q       0 Z        734358      1      1
3
##  5    14404 B      T9Q       0 Z        826308      1      1
4
##  6    14405 B      T9Q       0 Z        983650      0      1
5
##  7    14406 B      T9Q       0 Z        400753      1      1
6
##  8    14407 B      T9Q       0 Z        483575      1      1
7
##  9    14408 B      T9Q       0 Z        638337      0      1
8
## 10    14409 B      T9Q       0 Z        744115      1      1
```

```
9
## # ... with 22,174 more rows, and 9 more variables: Activity <chr>,
## #   ONTASK <chr>, TRANSITIONS <dbl>, NumACTIVITIES <dbl>,
FORMATchanges <dbl>,
## #   NumFORMATS <dbl>, Obsv/act <dbl>, Transitions/Durations <dbl>,
## #   Total Time <dbl>

table(train$ONTASK)

##
##     N     Y
##  7246 14938

summary(train)

##     UNIQUEID        SCHOOL             Class               GRADE
##  Min.   :14400   Length:22184       Length:22184       Min.   :0.000
##  1st Qu.:21277   Class :character   Class :character   1st Qu.:1.000
##  Median :28264   Mode  :character   Mode  :character   Median :2.000
##  Mean   :28257                                         Mean   :2.056
##  3rd Qu.:35231                                         3rd Qu.:4.000
##  Max.   :42130                                         Max.   :4.000
##     CODER             STUDENTID          Gender           OBSNUM
##  Length:22184       Min.   :  1123   Min.   :0.0000   Min.   : 1.000
##  Class :character   1st Qu.:264220   1st Qu.:0.0000   1st Qu.: 5.000
##  Mode  :character   Median :514301   Median :1.0000   Median : 9.000
##                     Mean   :506966   Mean   :0.5064   Mean   : 9.621
##                     3rd Qu.:743450   3rd Qu.:1.0000   3rd Qu.:14.000
##                     Max.   :999979   Max.   :1.0000   Max.   :32.000
##  totalobs-forsession   Activity            ONTASK
TRANSITIONS
##  Min.   :  1.0       Length:22184       Length:22184       Min.
:0.000
##  1st Qu.: 82.0       Class :character   Class :character   1st
Qu.:1.000
##  Median :165.0       Mode  :character   Mode  :character   Median
:2.000
##  Mean   :170.7                                             Mean
:2.383
##  3rd Qu.:248.0                                             3rd
Qu.:3.000
##  Max.   :511.0                                             Max.
:6.000
##  NumACTIVITIES   FORMATchanges     NumFORMATS       Obsv/act
##  Min.   :1.000   Min.   :0.000   Min.   :1.000   Min.   : 387.0
##  1st Qu.:2.000   1st Qu.:1.000   1st Qu.:2.000   1st Qu.: 721.2
##  Median :3.000   Median :1.000   Median :2.000   Median : 876.2
##  Mean   :3.383   Mean   :1.534   Mean   :2.534   Mean   : 973.5
##  3rd Qu.:4.000   3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.:1106.8
##  Max.   :7.000   Max.   :5.000   Max.   :6.000   Max.   :2735.0
##  Transitions/Durations   Total Time
```

```
##  Min.   :0.000000      Min.   :   0.0
##  1st Qu.:0.000839      1st Qu.: 252.0
##  Median :0.001513      Median : 586.5
##  Mean   :0.003159      Mean   : 774.6
##  3rd Qu.:0.003268      3rd Qu.:1121.0
##  Max.   :0.666667      Max.   :3554.0

trainD <- train %>% mutate(ONTASK_ON = as_factor(ONTASK))%>%
 select(ONTASK_ON,
TRANSITIONS,FORMATchanges,Gender,GRADE,`Obsv/act`,`Transitions/Duration
s`, `Total Time`,`totalobs-forsession` )

trainD

## # A tibble: 22,184 x 9
##    ONTASK_ON TRANSITIONS FORMATchanges Gender GRADE `Obsv/act`
`Transitions/Dur~
##    <fct>           <dbl>         <dbl>  <dbl> <dbl>      <dbl>
<dbl>
##  1 Y                   3             1      0     0       770.
0.00404
##  2 Y                   3             1      0     0       770.
0.00404
##  3 Y                   3             1      0     0       770.
0.00404
##  4 Y                   3             1      1     0       770.
0.00404
##  5 Y                   3             1      1     0       770.
0.00404
##  6 Y                   3             1      0     0       770.
0.00404
##  7 Y                   3             1      1     0       770.
0.00404
##  8 Y                   3             1      1     0       770.
0.00404
##  9 Y                   3             1      0     0       770.
0.00404
## 10 N                   3             1      1     0       770.
0.00404
## # ... with 22,174 more rows, and 2 more variables: Total Time <dbl>,
## #   totalobs-forsession <dbl>

#Decision Tree
library(party)

## Loading required package: grid

## Loading required package: mvtnorm

## Loading required package: modeltools
```

```
## Loading required package: stats4

## Loading required package: strucchange

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

##
## Attaching package: 'strucchange'

## The following object is masked from 'package:stringr':
##
##     boundary
```

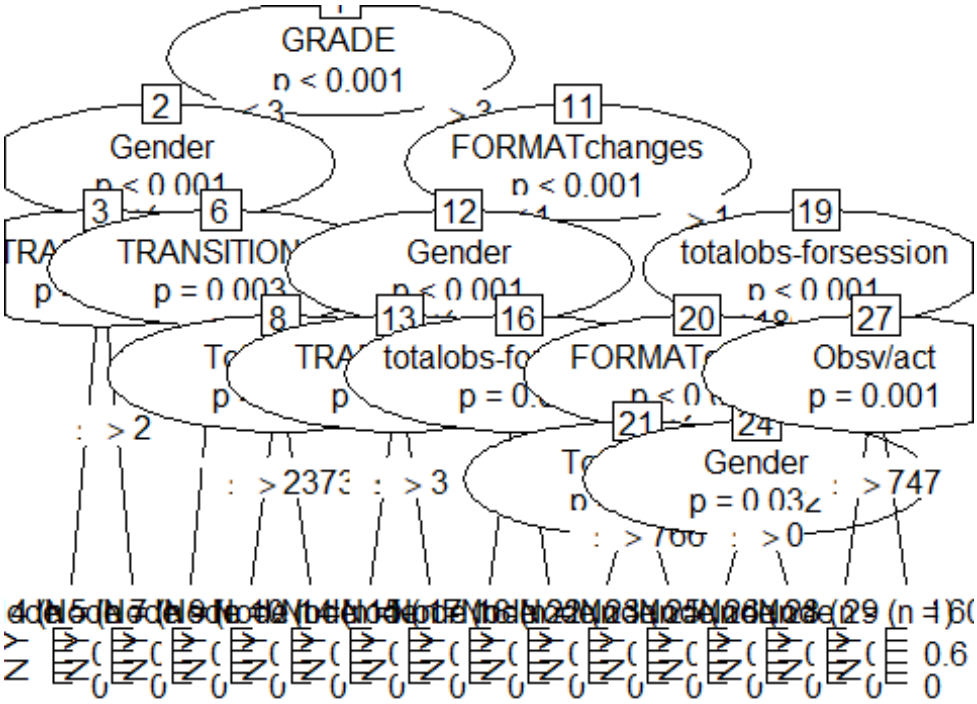
```
trainTree <- ctree(
 ONTASK_ON ~ TRANSITIONS + FORMATchanges + Gender + GRADE +
`Obsv/act`+`Transitions/Durations`+`Total Time`+`totalobs-forsession`,
 data = trainD)

print(trainTree)
```

```
##
##   Conditional inference tree with 15 terminal nodes
##
## Response:  ONTASK_ON
## Inputs:  TRANSITIONS, FORMATchanges, Gender, GRADE, Obsv/act,
Transitions/Durations, Total Time, totalobs-forsession
## Number of observations:  22184
##
## 1) GRADE <= 3; criterion = 1, statistic = 53.869
##   2) Gender <= 0; criterion = 1, statistic = 25.407
##     3) TRANSITIONS <= 2; criterion = 0.994, statistic = 11.41
##       4)*  weights = 4166
##     3) TRANSITIONS > 2
##       5)*  weights = 3539
##   2) Gender > 0
##     6) TRANSITIONS <= 2; criterion = 0.997, statistic = 12.397
##       7)*  weights = 4304
##     6) TRANSITIONS > 2
##       8) Total Time <= 2373; criterion = 0.961, statistic = 7.917
##         9)*  weights = 3589
##       8) Total Time > 2373
##         10)*  weights = 129
## 1) GRADE > 3
```

```
##   11) FORMATchanges <= 1; criterion = 1, statistic = 38.192
##     12) Gender <= 0; criterion = 1, statistic = 33.411
##       13) TRANSITIONS <= 3; criterion = 1, statistic = 22.713
##         14)*  weights = 2023
##       13) TRANSITIONS > 3
##         15)*  weights = 377
##     12) Gender > 0
##       16) totalobs-forsession <= 59; criterion = 0.96, statistic =
7.847
##         17)*  weights = 359
##       16) totalobs-forsession > 59
##         18)*  weights = 1781
##   11) FORMATchanges > 1
##     19) totalobs-forsession <= 186; criterion = 1, statistic =
21.884
##       20) FORMATchanges <= 2; criterion = 1, statistic = 17.186
##         21) Total Time <= 766; criterion = 0.998, statistic = 13.426
##           22)*  weights = 604
##         21) Total Time > 766
##           23)*  weights = 151
##       20) FORMATchanges > 2
##         24) Gender <= 0; criterion = 0.968, statistic = 8.234
##           25)*  weights = 111
##         24) Gender > 0
##           26)*  weights = 187
##     19) totalobs-forsession > 186
##       27) Obsv/act <= 747; criterion = 0.999, statistic = 14.63
##         28)*  weights = 261
##       27) Obsv/act > 747
##         29)*  weights = 603
```

```
plot(trainTree)
```




```
#Naive Bayes
library(e1071)

trainNB <- naiveBayes(
 ONTASK_ON ~ TRANSITIONS + FORMATchanges + Gender + GRADE +
`Obsv/act`+`Transitions/Durations`+`Total Time`+`totalobs-forsession`,
 data = trainD)

print(trainNB)

##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##         Y         N
## 0.6733682 0.3266318
##
## Conditional probabilities:
##    TRANSITIONS
## Y       [,1]     [,2]
##   Y 2.412706 1.329310
##   N 2.322109 1.283445
##
```

```
##    FORMATchanges
## Y       [,1]     [,2]
##   Y 1.539229 1.240604
##   N 1.524151 1.205993
##
##    Gender
## Y        [,1]      [,2]
##   Y 0.5222921 0.4995195
##   N 0.4737786 0.4993464
##
##    GRADE
## Y       [,1]     [,2]
##   Y 2.004485 1.499385
##   N 2.162434 1.505628
##
##    Obsv/act
## Y       [,1]     [,2]
##   Y 971.2960 465.1450
##   N 978.1209 431.8459
##
##    Transitions/Durations
## Y          [,1]        [,2]
##   Y 0.003141409 0.007782243
##   N 0.003196415 0.014161842
##
##    Total Time
## Y       [,1]     [,2]
##   Y 773.2423 671.0709
##   N 777.3617 652.8731
##
##    totalobs-forsession
## Y       [,1]     [,2]
##   Y 171.0404 106.6437
##   N 169.9786 104.0082
```

```
#K-fold cross validation for Decision Tree
library(caret)
```

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

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

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
set.seed(125)

train_control <- trainControl(method = "cv",
                              number = 10)
```

```r
tree_fit <- train(factor(ONTASK_ON) ~., data = trainD,
                  method = "ctree",
                  trControl = train_control)

print(tree_fit)
```

```
## Conditional Inference Tree
##
## 22184 samples
##     8 predictor
##     2 classes: 'Y', 'N'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 19966, 19966, 19965, 19965, 19966, 19966,
...
## Resampling results across tuning parameters:
##
##   mincriterion  Accuracy   Kappa
##   0.01          0.6603417  0.08645091
##   0.50          0.6780111  0.05932712
##   0.99          0.6746306  0.02760443
##
## Accuracy was used to select the optimal model using the largest
value.
## The final value used for the model was mincriterion = 0.5.
```

```r
#K-fold cross validation for Naive Bayes
set.seed(100)

trctrl <- trainControl(method = "cv", number = 10,
savePredictions=TRUE)

nb_fit <- train(factor(ONTASK_ON) ~., data = trainD, method =
"naive_bayes", trControl=trctrl, tuneLength = 0)

print(nb_fit)
```

```
## Naive Bayes
##
## 22184 samples
##     8 predictor
##     2 classes: 'Y', 'N'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 19965, 19966, 19966, 19966, 19966, 19966,
...
## Resampling results across tuning parameters:
```

```
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.6640826  -8.631760e-03
##    TRUE      0.6730076  -5.437745e-05
##
## Tuning parameter 'laplace' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest
value.
## The final values used for the model were laplace = 0, usekernel =
TRUE
##  and adjust = 1.
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.