R Notebook

Classification Course: HUDK 4050, Week 6

Author: Guotai Sun

Assignment: ICE4

Objectives:

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

implement a binary logistic regression model to train a classifier

implement a decision tree model to train a classifier

implement a Naive Bayes model to train a classifier

report model performance on a validation dataset

Classification is among the most important areas of machine learning, and there are a lot of implementations. By the end of this ICE, you'll have learned about classification in general and the basics of logistic regression, decision tree, and Naive Bayes in particular, as well as their implementation in R.

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*.

```
#Logistic Regression
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
```

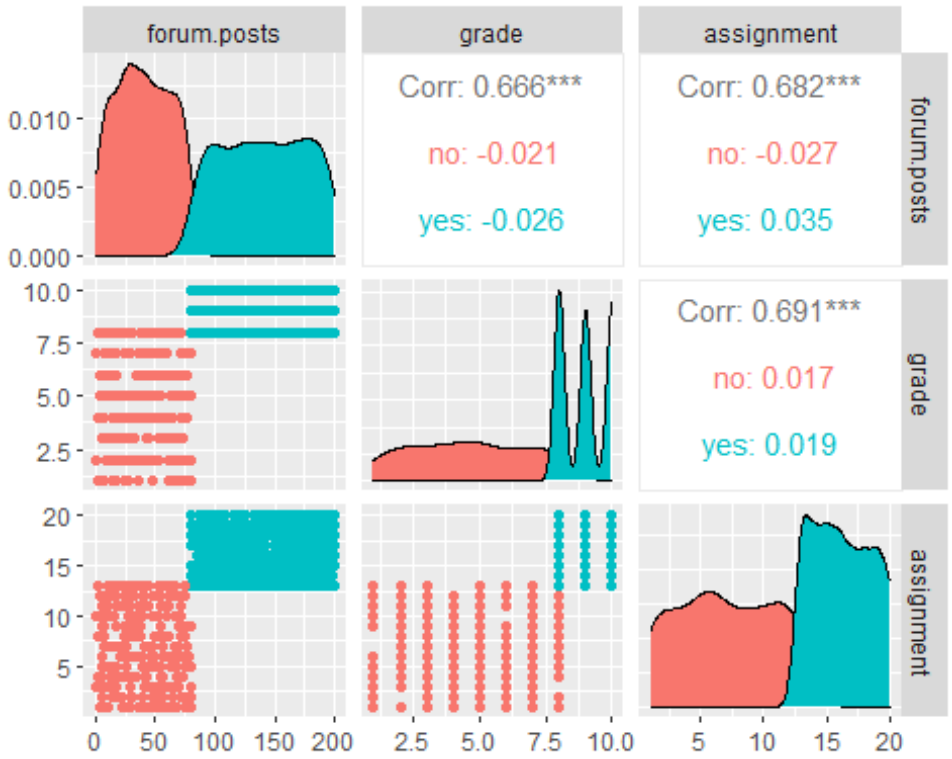
```
## Warning: package 'tibble' was built under R version 4.0.5
```

```
## 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()

ICE4 <- read_csv("ICE4_Data.csv")

##
## -- Column specification --------------------------------------------
------------
## cols(
##   certified = col_character(),
##   forum.posts = col_double(),
##   grade = col_double(),
##   assignment = col_double()
## )

ICE4

## # A tibble: 1,000 x 4
##    certified forum.posts grade assignment
##    <chr>           <dbl> <dbl>      <dbl>
##  1 no                  7     3          9
##  2 no                  7     4          1
##  3 yes               191     8         19
##  4 yes               130    10         18
##  5 yes               135     8         18
##  6 no                 24     2         11
##  7 yes               188    10         14
##  8 no                 51     4          2
##  9 no                 26     2          5
## 10 no                 40     2         13
## # ... with 990 more rows

table(ICE4$certified)

##
##  no yes
## 275 725

summary(ICE4)

##   certified          forum.posts         grade          assignment
##  Length:1000        Min.   :  1.00   Min.   : 1.000   Min.   : 1.00
##  Class :character   1st Qu.: 72.75   1st Qu.: 8.000   1st Qu.:12.00
##  Mode  :character   Median :118.50   Median : 8.000   Median :15.00
##                     Mean   :113.11   Mean   : 7.765   Mean   :13.69
##                     3rd Qu.:160.00   3rd Qu.: 9.000   3rd Qu.:17.00
##                     Max.   :200.00   Max.   :10.000   Max.   :20.00

#install.packages("GGally")
library(GGally)
```

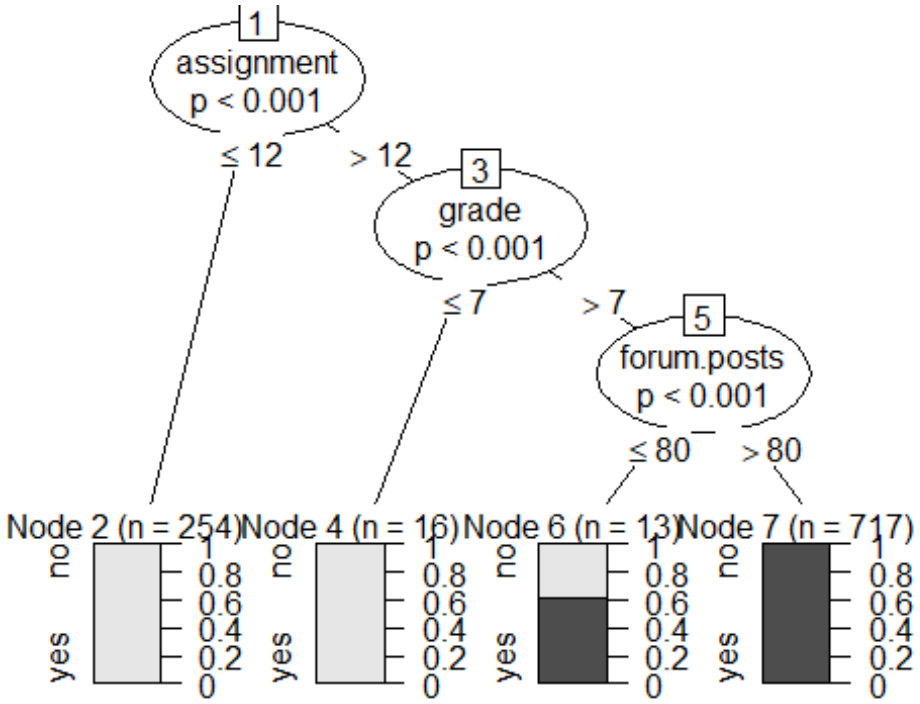
```
## Warning: package 'GGally' was built under R version 4.0.5

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

ggpairs(ICE4, columns = 2:4, ggplot2::aes(colour=certified))
```




```
ICE4 <- ICE4 %>% mutate(certified_yes = as_factor(certified)) %>%
  select(certified_yes, forum.posts, grade, assignment)
ICE4

## # A tibble: 1,000 x 4
##    certified_yes forum.posts grade assignment
##    <fct>               <dbl> <dbl>      <dbl>
##  1 no                      7     3          9
##  2 no                      7     4          1
##  3 yes                   191     8         19
##  4 yes                   130    10         18
##  5 yes                   135     8         18
##  6 no                     24     2         11
##  7 yes                   188    10         14
##  8 no                     51     4          2
##  9 no                     26     2          5
## 10 no                     40     2         13
## # ... with 990 more rows
```

```
logitModel <- glm(certified_yes ~ forum.posts + grade + assignment,
data = ICE4, family = "binomial")

## Warning: glm.fit: algorithm did not converge

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

summary(logitModel)

##
## Call:
## glm(formula = certified_yes ~ forum.posts + grade + assignment,
##     family = "binomial", data = ICE4)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -2.948e-04  -2.000e-08   2.000e-08   2.000e-08   3.709e-04
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -328.561  39360.821  -0.008    0.993
## forum.posts     2.573    285.837   0.009    0.993
## grade           5.480   5193.765   0.001    0.999
## assignment      7.339   2112.383   0.003    0.997
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1.1763e+03  on 999  degrees of freedom
## Residual deviance: 2.5003e-07  on 996  degrees of freedom
## AIC: 8
##
## Number of Fisher Scoring iterations: 25

#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

ICE4Tree <- ctree(
  certified_yes ~ forum.posts + grade + assignment,
  data = ICE4)

print(ICE4Tree)

##
##   Conditional inference tree with 4 terminal nodes
##
## Response:  certified_yes
## Inputs:  forum.posts, grade, assignment
## Number of observations:  1000
##
## 1) assignment <= 12; criterion = 1, statistic = 689.719
##   2)*  weights = 254
## 1) assignment > 12
##   3) grade <= 7; criterion = 1, statistic = 244.485
##     4)*  weights = 16
##   3) grade > 7
##     5) forum.posts <= 80; criterion = 1, statistic = 39.938
##       6)*  weights = 13
##     5) forum.posts > 80
##       7)*  weights = 717

plot(ICE4Tree)
```




```
#Naive Bayes

library(e1071)

ICE4NB <- naiveBayes(
  certified_yes ~ forum.posts + grade + assignment,
  data = ICE4)
ICE4NB

##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##    no   yes
## 0.275 0.725
##
## Conditional probabilities:
##      forum.posts
## Y         [,1]     [,2]
##   no   40.2400 22.44280
##   yes 140.7462 34.94066
##
##      grade
```

```
## Y         [,1]      [,2]
##   no  4.574545 2.1892420
##   yes 8.975172 0.8273244
##
##      assignment
## Y          [,1]     [,2]
##   no   6.934545 3.791472
##   yes 16.256552 2.300096
```

```
certified_pred_NB <- predict(ICE4NB, ICE4[,2:4])

performance = ICE4$certified_yes == certified_pred_NB
cat('The accuracy is', sum(performance)/length(performance)*100, '%')
```

```
## The accuracy is 99.8 %
```

```
#Model Evaluation

set.seed(123)

sample_size <- floor(0.8*nrow(ICE4))

picked <- sample(seq_len(nrow(ICE4)),size = sample_size)

training_ICE4 <- ICE4[picked,]

testing_ICE4 <- ICE4[-picked,]

ICE4Logit <- glm(certified_yes ~ forum.posts + grade + assignment, data
= training_ICE4, family = "binomial")
```

```
## Warning: glm.fit: algorithm did not converge

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

```
ICE4Tree <- ctree(certified_yes ~ forum.posts + grade + assignment,
data = ICE4)

probabilities <- predict(ICE4Logit, testing_ICE4[,2:4], type =
"response")
certified_pred_logit <- ifelse(probabilities > 0.5, "yes", "no")

certified_pred_tree <- predict(ICE4Tree, testing_ICE4[,2:4])

logitCM <- table(testing_ICE4$certified_yes,certified_pred_logit)
logitCM
```

```
##      certified_pred_logit
##        no yes
```

```
##   no   59   1
##   yes   0 140
```

```
treeCM <- table(testing_ICE4$certified_yes,certified_pred_tree)
treeCM
```

```
##      certified_pred_tree
##        no yes
##   no   60   0
##   yes   0 140
```

```
library(caret)
```

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

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

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

```
logitAccuracy <- confusionMatrix(logitCM)$overall["Accuracy"]
cat('The accuracy for the logistic regression model is',
logitAccuracy*100, '%')
```

```
## The accuracy for the logistic regression model is 99.5 %
```

```
logitAccuracy <- confusionMatrix(treeCM)$overall["Accuracy"]
cat('The accuracy for the tree regression model is', logitAccuracy*100,
'%')
```

```
## The accuracy for the tree regression model is 100 %
```

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.