final189

Kalyani Cauwenberghs

3/7/2020

Part 1

Instructions: Compile a collection of tips for best data presentation, including the illustration and R script for each TODO: what does she mean for R script and illustration? resolved one should boxplot next to each other –> provide sample code how that's done use Rmd, show R code and plot. #Part 2 Instructions: For the final project, look back at all the analysis approaches you have used throughout the quarter. Consider HealthGen as the outcome, grouped into Excellent/Vgood, versus Good/Fair/Poor. (In the dataset dat.Rda, the HealthGen variable is assigned value 1 if its original value is Excellent/Vgood and is assigned value 0 otherwise.) Consider all other variables as potential predictors. Develop a comprehensive and reproducible analysis report, to explore the relationship between these variables and the outcome. Pay attention to (but not limited to) the following:

1.)

Missing data: do not remove observations with any missing data from the start; after screening you might reduce to a smaller set of variables, therefore remove fewer observations at that point. Also you may consider removing variables with too much missing.

```
load("dat (1).rda")
#remove columns with more than 310 missing
#TODO not hard
#TODO: get rid of categories with 0
dat<-dat[,-which(colSums(is.na(dat)) > 1457)]
#first combine categories:
#put mexican under hispanic
dat$Race1[which(dat$Race1=="Mexican")]<-"Hispanic"
#split education into above or below high school grad
levels(dat$Education)<-c(levels(dat$Education), "HS_or_less", "college_or_more")
dat$Education[which(dat$Education=="8th Grade" |dat$Education=="9 - 11th Grade"|dat$Education=="High Sc
dat$Education[which(dat$Education=="Some College" |dat$Education=="College Grad")]<-"college_or_more"
#split marital status into the following: (live partner, married), (divorced,separated), (widowed, neve
levels(dat$MaritalStatus)<-c(levels(dat$MaritalStatus), "have_SO")
dat$MaritalStatus[c(which(dat$MaritalStatus=="LivePartner"), which(dat$MaritalStatus=="Married"))]<-"ha
#below mean vs above median income (55000)
levels(dat$HHIncome)<-c(levels(dat$HHIncome), "below_med", "above_med")
below_med<-levels(dat$HHIncome)[1:9]
above_med<-levels(dat$HHIncome)[10:12]
dat$HHIncome[which(dat$HHIncome %in% below_med)]<-"below_med"
dat$HHIncome[which(dat$HHIncome %in% above_med)]<-"above_med"

#BMI: combine underweight with normal
```

```r
levels(dat$BMI_WHO)<-c("12.0_to_24.9",levels(dat$BMI_WHO))
dat$BMI_WHO[c(which(dat$BMI_WHO == "12.0_18.5"),which(dat$BMI_WHO == "18.5_to_24.9"))]<-"12.0_to_24.9"
#depressed: combine several with most
levels(dat$Depressed)<-c(levels(dat$Depressed), "Lots")
dat$Depressed[c(which(dat$Depressed=="Several"),which(dat$Depressed=="Most"))]<-"Lots"
#comp hrs day: categories: (0,1) (2,+)
levels(dat$CompHrsDay)<-c(levels(dat$CompHrsDay), "one_or_less", "two_or_more")
one_or_less<-levels(dat$CompHrsDay)[1:3]
two_or_more<-levels(dat$CompHrsDay)[4:7]
dat$CompHrsDay[which(dat$CompHrsDay %in% one_or_less)]<-"one_or_less"
dat$CompHrsDay[which(dat$CompHrsDay %in% two_or_more)]<-"two_or_more"

#TV hrs day: categories: (0,1) (2,+)
levels(dat$TVHrsDay)<-c(levels(dat$TVHrsDay), "one_or_less", "two_or_more")
dat$TVHrsDay[which(dat$TVHrsDay %in% one_or_less)]<-"one_or_less"
dat$TVHrsDay[which(dat$TVHrsDay %in% two_or_more)]<-"two_or_more"

#sex orient: hetero vs other
levels(dat$SexOrientation)<-c(levels(dat$SexOrientation), "Other")
dat$SexOrientation[c(which(dat$SexOrientation=="Bisexual"),which(dat$SexOrientation=="Homosexual"))]<-"

for(i in 1:ncol(dat)){
  if(class(dat[,i])=="factor"){
    print(colnames(dat)[i])
    print(table(dat[,i])/length(dat[,i]))
  }
}
```

```
## [1] "Gender"
##
##    female      male
## 0.4858067 0.5141933
## [1] "Race1"
##
##      Black   Hispanic    Mexican      White      Other
## 0.12145167 0.16349263 0.00000000 0.62378728 0.09126842
## [1] "Education"
##
##       8th Grade  9 - 11th Grade     High School    Some College    College Grad
##       0.0000000       0.0000000       0.0000000       0.0000000       0.0000000
##      HS_or_less college_or_more
##       0.3248293       0.6320517
## [1] "MaritalStatus"
##
##     Divorced  LivePartner      Married NeverMarried    Separated      Widowed
##   0.09306504   0.00000000   0.00000000   0.25044916   0.02587136   0.01042041
##      have_SO
##   0.57707510
## [1] "HHIncome"
##
##      0-4999   5000-9999 10000-14999 15000-19999 20000-24999 25000-34999
##   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
## 35000-44999 45000-54999 55000-64999 65000-74999 75000-99999  more 99999
##   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
```

```
##   below_med   above_med
##   0.4958678   0.4236436
## [1] "BMI_WHO"
##
## 12.0_to_24.9    12.0_18.5 18.5_to_24.9 25.0_to_29.9    30.0_plus
##    0.6496586    0.0000000    0.0000000    0.3417176    0.0000000
## [1] "HealthGen"
##
##         0         1
## 0.4858067 0.4028027
## [1] "Depressed"
##
##      None   Several      Most      Lots
## 0.6981674 0.0000000 0.0000000 0.1882860
## [1] "SleepTrouble"
##
##       No      Yes
## 0.745239 0.254761
## [1] "TVHrsDay"
##
##       0_hrs   0_to_1_hr        1_hr        2_hr        3_hr        4_hr
##   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
##   More_4_hr one_or_less two_or_more
##   0.0000000   0.3661516   0.6338484
## [1] "CompHrsDay"
##
##       0_hrs   0_to_1_hr        1_hr        2_hr        3_hr        4_hr
##   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
##   More_4_hr one_or_less two_or_more
##   0.0000000   0.6881064   0.3118936
## [1] "RegularMarij"
##
##        No       Yes
## 0.6313331 0.2464966
## [1] "SexOrientation"
##
##     Bisexual Heterosexual   Homosexual        Other
##   0.00000000   0.82285304   0.00000000   0.04024434

## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE

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

## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] FALSE
## [1] TRUE
```

```
## [1] TRUE
## [1] FALSE
```

```
## [1] "our final screened variables:"
```

```
##  [1] "Gender"        "Age"           "Race1"         "Education"
##  [5] "MaritalStatus" "HHIncome"      "BMI_WHO"       "Depressed"
##  [9] "SleepTrouble"  "TVHrsDay"      "AlcoholYear"   "RegularMarij"
```

2.)

Include "Table 1"

3.)

After univariate screening, building a multiple logistic regression model to predict the general health outcome of very good or excellent versus otherwise. State clearly your criteria at each step in the narrative.

```
## [1] "formula:"
```

```
## [1] "HealthGen~Gender+Age+Race1+Education+MaritalStatus+HHIncome+BMI_WHO+Depressed+SleepTrouble+TVHrs
```

```
## [1] "Final model:"
```

```
##               (Intercept)                 Gendermale                        Age
##              0.6961432830              -0.2767995696              -0.0204925283
##             Race1Hispanic                 Race1White                 Race1Other
##              -0.2342713978               0.3478541928              -0.1296240937
##  Educationcollege_or_more MaritalStatusNeverMarried    MaritalStatusSeparated
##               0.4100974152               0.0908903428               0.5321557992
##      MaritalStatusWidowed      MaritalStatushave_SO         HHIncomeabove_med
##              -0.5840308163               0.1606541425               0.6669789987
##       BMI_WHO25.0_to_29.9             DepressedLots            SleepTroubleYes
##              -1.2247368361              -0.8769732214              -0.7668729896
##       TVHrsDaytwo_or_more               AlcoholYear            RegularMarijYes
##              -0.0680462167               0.0003411239              -0.1920124381
```

TODO: each caegorical variable has different categories. when doing univariate logistic regression of a categorical variable, glm splits it into its categories and treats each category as a variable. so the regression ends up not being univariate. I assume the way to solve this is to make a variable out of each category of each categorical variable whose value is 1 if the observation belongs to that category and 0 if not. this would take lots of time. 1.) for screening, do we have to do this? 2.) for univariate regression in forward stepwise selection, do we have to do this?

after viewing the presentations, it looks like the second group made the continuous variables into binary. should we do this?

actually, another solution is to do backwards stepwise selection with all the variables and wipe out the variables with p>0.05.

4.)

Assess the predictability of the model by computing the (generalized) R-squared and the area under the ROC curve (AUC), as well as the cross-validated AUC.

```
## [1] 1972
```

```
## [1] 1972
```

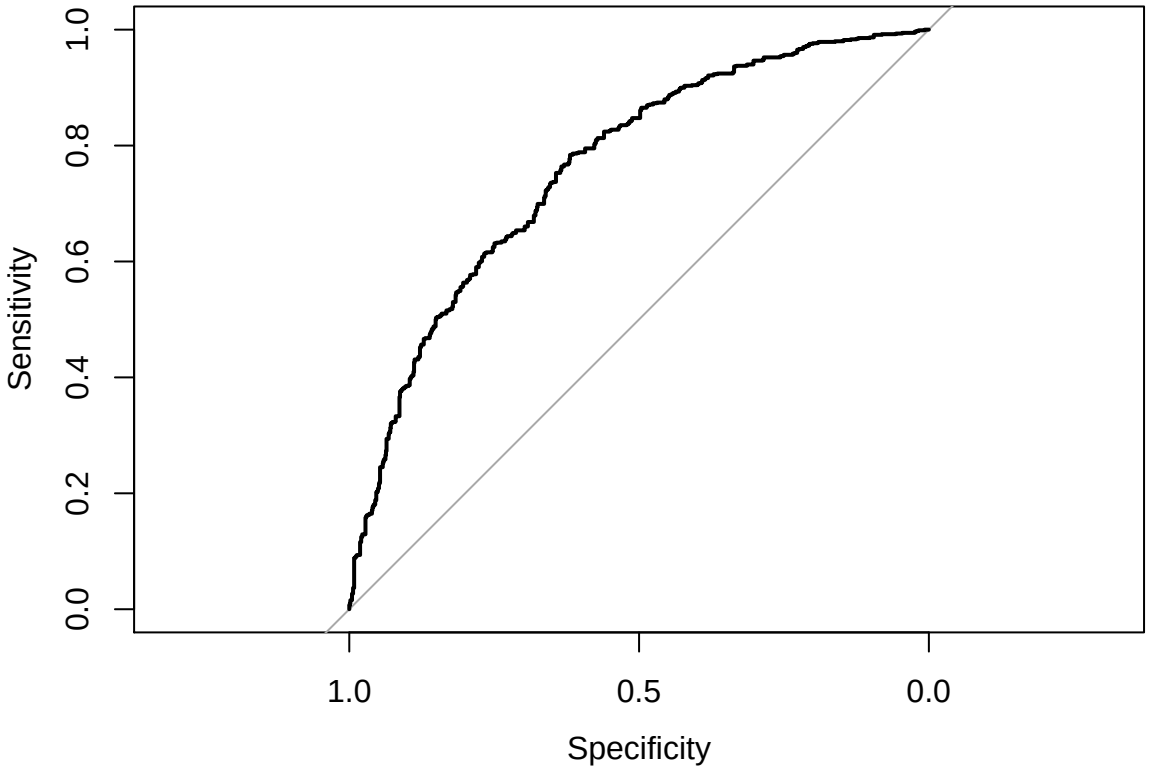
```
## [1] 1972
```

Find generalized R-squared:

```
## ##generalized R-squared:  0.1975666
```

ROC and AUC

```
## [1] "Plotting ROC of our selected model"
```




```
## [1] "AUC of our selected model"
```

```
## Area under the curve: 0.7618
```

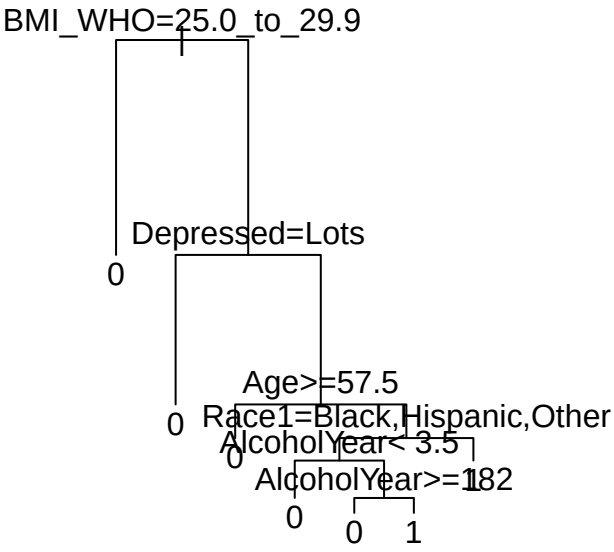
Cross-validated ROC and AUC

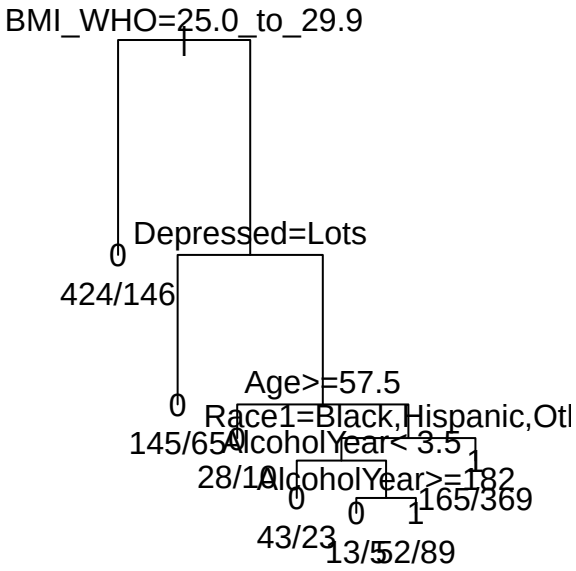
```
## [1] "final AUC:"
```

```
## [1] 0.751405
```

5.)

Use the variables that have passed the univariate screening, to build a classification tree. Describe clearly how you arrive at the final tree. Compute the error rate of your classification tree.






```
## null device
##           1
```

```
##
## Classification tree:
## rpart(formula = formula, data = dat, subset = train)
##
## Variables actually used in tree construction:
## [1] Age         AlcoholYear BMI_WHO     Depressed   Race1
##
## Root node error: 707/1577 = 0.44832
##
## n= 1577
##
##         CP nsplit rel error  xerror     xstd
## 1 0.162659      0   1.00000 1.00000 0.027934
## 2 0.113154      1   0.83734 0.83734 0.027198
## 3 0.025460      2   0.72419 0.72419 0.026301
## 4 0.014144      3   0.69873 0.69873 0.026052
## 5 0.011315      5   0.67044 0.69873 0.026052
## 6 0.010000      6   0.65912 0.69165 0.025980
```

6.)

Discuss any limitations in the analysis.

Bonus

Explore random forest on the data above.