Assignment 5

Megan Marziali

Feb 10, 2021

## Part 1: Set-up

### Problem set-up

##### Loading required packages

The following code chunk loads the required packages for the assignment.

library(tidyverse)   
library(caret)  
library(glmnet)  
  
set.seed(100)

##### Loading data into environment and cleaning

alc\_data =   
 read.csv("./data/alcohol\_use.csv", na = c("", ".", "NA", ".d", ".r")) %>%   
 janitor::clean\_names() %>%  
 mutate(  
 alc\_consumption = as.factor(alc\_consumption)  
 ) %>%   
 select(-x)  
  
summary(alc\_data)

To clean the data, I stripped off the ID variable and recoded the alcohol consumption variable as a factor. The complete dataset includes 1885 observations. No missing observations are present within the data.

##### Data partitioning

train.indices = createDataPartition(y = alc\_data$alc\_consumption,p = 0.7,list = FALSE)  
  
training = alc\_data[train.indices,]  
testing = alc\_data[-train.indices,]  
  
#Store outcome   
alc\_train = training$alc\_consumption  
alc\_test = testing$alc\_consumption  
  
#Model.matrix shortcut to removing outcome variable from matrix  
x\_train = model.matrix(alc\_consumption~., training)[,-1]  
x\_test = model.matrix(alc\_consumption~., testing)[,-1]

## Problem 1

### 1.1: Choosing alpha and lambda via cross-validation

model.1 = train(  
 alc\_consumption ~., data = training, method = "glmnet",  
 trControl = trainControl("cv", number = 10),  
 tuneLength = 10  
 )  
  
param = model.1$bestTune  
  
# Fitting model  
model\_1 = glmnet(x\_train, alc\_train, family = binomial, alpha = param$alpha, lambda = param$lambda)  
coef(model\_1)

## 8 x 1 sparse Matrix of class "dgCMatrix"  
## s0  
## (Intercept) -0.1360322012  
## neurotocism\_score .   
## extroversion\_score .   
## openness\_score .   
## agreeableness\_score .   
## conscientiousness\_score .   
## impulsiveness\_score -0.4395131944  
## sens\_seeking\_score -0.0008956011

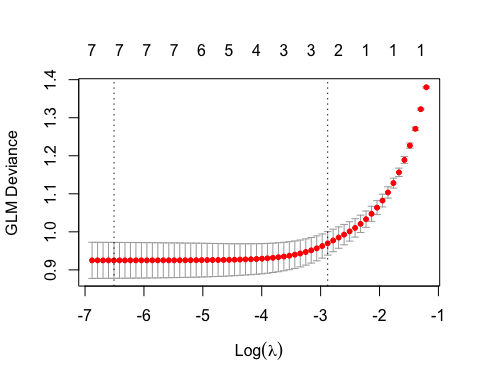
### 1.2: Model that uses logistic regression

model\_2 = glm(alc\_consumption ~ . ,family = binomial(link = 'logit'),data = training)  
summary(model\_2)

##   
## Call:  
## glm(formula = alc\_consumption ~ ., family = binomial(link = "logit"),   
## data = training)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9015 -0.7541 -0.2224 0.6597 3.4494   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.22334 0.07214 -3.096 0.00196 \*\*   
## neurotocism\_score -0.14892 0.08461 -1.760 0.07842 .   
## extroversion\_score -0.41965 0.09050 -4.637 3.53e-06 \*\*\*  
## openness\_score -0.06617 0.08360 -0.791 0.42866   
## agreeableness\_score -0.08752 0.07951 -1.101 0.27101   
## conscientiousness\_score 0.03249 0.08410 0.386 0.69930   
## impulsiveness\_score -1.77505 0.12512 -14.186 < 2e-16 \*\*\*  
## sens\_seeking\_score -0.22608 0.10152 -2.227 0.02595 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1824.3 on 1319 degrees of freedom  
## Residual deviance: 1201.6 on 1312 degrees of freedom  
## AIC: 1217.6  
##   
## Number of Fisher Scoring iterations: 5

### 1.3: LASSO model using all features

model\_3 = cv.glmnet(x\_train, alc\_train, alpha = 1, standardize = TRUE, family = binomial)  
  
plot(model\_3)



model\_3$lambda.min

## [1] 0.001490796

model\_3$lambda.1se

## [1] 0.05612751

model\_3\_final = glmnet(x\_train, alc\_train, alpha = 1, lambda = model\_3$lambda.1se, family = binomial)  
coef(model\_3\_final)

## 8 x 1 sparse Matrix of class "dgCMatrix"  
## s0  
## (Intercept) -0.17559353  
## neurotocism\_score .   
## extroversion\_score -0.04389756  
## openness\_score .   
## agreeableness\_score .   
## conscientiousness\_score .   
## impulsiveness\_score -1.27339635  
## sens\_seeking\_score -0.06516229

## Part 2: Evaluating Model Performance

### 2.1: Alpha and lambda cross-validation model

fitted.results = predict(model\_1, x\_test, type = 'response')   
fitted.results.p = ifelse(fitted.results > 0.5,1,0)  
  
testing.outcome = (as.numeric(testing$alc\_consumption) - 1)  
  
misClasificError = mean(fitted.results.p != testing.outcome, na.rm = T)  
print(paste('Accuracy Model 1',1 - misClasificError))

## [1] "Accuracy Model 1 0.849557522123894"

The accuracy of model 1 is 0.85.

### 2.2: Evaluating logistic regression model

fitted.results.2 = predict(model\_2, testing, type = 'response')   
fitted.results.p.2 = ifelse(fitted.results.2 > 0.5,1,0)  
  
testing.outcome.2 = (as.numeric(testing$alc\_consumption) - 1)  
  
misClasificError.2 = mean(fitted.results.p.2 != testing.outcome.2, na.rm = T)  
print(paste('Accuracy Model 2',1 - misClasificError.2))

## [1] "Accuracy Model 2 0.8"

The accuracy of model 2 is 0.80.

### 2.3: Evaluating LASSO model

fitted.results.3 = predict(model\_3\_final, x\_test, type = 'response')   
fitted.results.p.3 = ifelse(fitted.results.3 > 0.5, 1, 0)  
  
testing.outcome.3 = (as.numeric(testing$alc\_consumption) - 1)  
  
misClasificError.3 = mean(fitted.results.p.3 != testing.outcome.3, na.rm = T)  
print(paste('Accuracy Model 3',1 - misClasificError.3))

## [1] "Accuracy Model 3 0.782300884955752"

The accuracy of model 3 is 0.78.

Given the accuracies presented above across models, model 1 would be the best choice as it has the highest accuracy.

## Part 4: Research Questions

This analysis could aid future analyses in multiple ways. Firstly, this analysis outlines an approach for determining predictors of alcohol consumption. This analysis could aid in answering questions regarding variables to intervene on to reduce alcohol consumption, if the research question of interest was to predict variables related to alcohol consumption. Indirectly, this analysis provides insight into a methodological approach to answering the research question. Were someone to want to work with this data, and answer different research questions (i.e. using a different exposure or outcome), they could use this analysis to guide their own in terms of necessary analytic steps.