Assignment 5

2023-02-19

### Load Packages Needed for Both Questions

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.4.1 ✔ purrr 1.0.1  
## ✔ tibble 3.1.8 ✔ dplyr 1.1.0  
## ✔ tidyr 1.3.0 ✔ stringr 1.5.0  
## ✔ readr 2.1.4 ✔ forcats 1.0.0  
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(janitor)

##   
## Attaching package: 'janitor'  
##   
## The following objects are masked from 'package:stats':  
##   
## chisq.test, fisher.test

library(yardstick)

## For binary classification, the first factor level is assumed to be the event.  
## Use the argument `event\_level = "second"` to alter this as needed.  
##   
## Attaching package: 'yardstick'  
##   
## The following object is masked from 'package:readr':  
##   
## spec

library(stats)  
library(factoextra)

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(cluster)  
library(caret)

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following objects are masked from 'package:yardstick':  
##   
## precision, recall, sensitivity, specificity  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

## Assignment Instructions:

**Goal: You want to predict current alcohol consumption but it is expensive and time-consuming to administer all of the behavioral testing that produces the personality scores. You will conduct a reproducible analysis to build and test classification models using regularized logistic regression and traditional logistic regression.**

Address the following:

You should create and compare three different models:

* A model that chooses alpha and lambda via cross-validation using all of the features
* A model that uses all the features and traditional logistic regression
* A lasso model using all of the features

You should tune and compare the performance of all three models within the training set using cross-validation and then decide which model you would choose as your final model. Provide justification for your choice.

Apply your final model in the test set and report your final evaluation metrics.

Produce a shareable report of your analysis and results using R Markdown. \*\*\*

### Step 1: Load and Prepare Dataset

set.seed(123)  
  
#Loading in Data  
a5.data <- read.csv("~/Desktop/Nimish/Columbia MPH/4. Spring 2023/Machine Learning in Epi/Assignments/Assignment 5/alcohol\_use.csv") %>%   
 clean\_names()  
  
#Convert all categorical variables to factors   
a5.data <- a5.data %>%   
 mutate(alc\_consumption = as.factor(alc\_consumption))  
  
#Strip ID  
a5.data$x <- NULL  
  
#Remove Missing Variables  
a5.data <- na.omit(a5.data)  
  
#Check Data Structure  
str(a5.data)

## 'data.frame': 1885 obs. of 8 variables:  
## $ neurotocism\_score : num 1.72 1.84 1.604 1.72 0.521 ...  
## $ extroversion\_score : num 0.322 -0.948 -0.948 0.322 -1.232 ...  
## $ openness\_score : num -1.119 -0.9763 1.2403 1.6565 -0.0193 ...  
## $ agreeableness\_score : num -0.4532 -0.917 -0.917 -0.1549 -0.0173 ...  
## $ conscientiousness\_score: num -2.42 -1.01 -1.39 -1.78 -2.18 ...  
## $ impulsiveness\_score : num 2.9 2.9 2.9 2.9 1.86 ...  
## $ sens\_seeking\_score : num -0.216 1.922 1.922 0.401 0.401 ...  
## $ alc\_consumption : Factor w/ 2 levels "CurrentUse","NotCurrentUse": 1 1 1 1 1 1 1 1 1 1 ...

#Finding correlated predictors  
a5.data.numeric <- a5.data %>% dplyr::select(where(is.numeric))  
correlations<-cor(a5.data.numeric, use="complete.obs")  
high.correlations<-findCorrelation(correlations, cutoff=0.4) #None Found  
  
a5.data.lowcor<-a5.data[,-high.correlations]  
  
#Centering and Scaling  
set.up.preprocess<-preProcess(a5.data.lowcor, method=c("center", "scale"))  
  
#Output pre-processed values  
transformed.vals<-predict(set.up.preprocess, a5.data.lowcor)

### Step 2: Data Partitioning

set.seed(123)  
  
train.index<-createDataPartition(transformed.vals$alc\_consumption, p=0.7, list=FALSE)  
  
a5.data.train<-transformed.vals[train.index,]  
a5.data.test<-transformed.vals[-train.index,]  
  
  
#Construct k-folds in your data  
train.folds<-createFolds(transformed.vals$alc\_consumption, k=10, list=FALSE)

### Step 3: Training the Model using different methods:

#### 1. A Model that chooses **Alpha and Lambda** via cross-validation using all of the features

set.seed(123)  
  
#Training an Elastic Net Model  
  
en.model<- train(alc\_consumption ~., data = a5.data.train,   
 method = "glmnet",  
 trControl = trainControl("cv", number = 10),   
 preProc=c("center", "scale"),  
 tuneLength=10  
 )  
  
#Print the values of alpha and lambda that gave best prediction  
en.model$bestTune

## alpha lambda  
## 36 0.4 0.2578427

#Accuracy  
en.model$results$Accuracy[36]

## [1] 0.8515577

# Alpha Lambda Accuracy  
# 0.4 0.2578427450 0.8515577

#### 2. A model that uses **all the features** and **traditional regression**.

set.seed(123)  
  
#Training a Logistic Regression Model  
log.model <- glm(alc\_consumption ~ ., data = a5.data.train,   
 family = "binomial")  
  
#Setting a cross validation control  
cv\_results <- trainControl(method = "cv", number = 10)  
  
#Training a logistic regression model with cross validation  
cv.log.model <- train(alc\_consumption ~ ., data = a5.data.train,  
 method = "glm",  
 family = "binomial",  
 trControl = cv\_results,  
 preProc=c("center", "scale"))  
  
  
cv.log.model$results

## parameter Accuracy Kappa AccuracySD KappaSD  
## 1 none 0.7826045 0.5643039 0.03754273 0.07404442

# Parameters Accuracy   
# None 0.7826045

#### 3. A lasso model that uses **all the features**.

set.seed(123)  
  
#Create grid to search lambda  
lambda<-10^seq(-3,3, length=100) #Creating lambda grid  
  
#Training a Lasso Model  
lasso.model <- train(alc\_consumption ~., data=a5.data.train,   
 method="glmnet",   
 trControl=trainControl("cv", number=10),   
 preProc=c("center", "scale"),   
 tuneGrid=expand.grid(alpha=1, lambda=lambda)  
)  
  
  
lasso.model$bestTune

## alpha lambda  
## 40 1 0.231013

lasso.model$results$Accuracy[40]

## [1] 0.8515577

# Alpha Lambda Accuracy  
# 1 0.231013 0.8515577

To compare the three models, I created examined the accuracies for each model based on the cross-validation within the training set.

1. Elastic Net Model Accuracy: **0.8515577**
2. Traditional Logistic Regression Model Accuracy: **0.7826045**
3. Lasso Model Accuracy: **0.8515577**

The lasso model and the elastic GLM modeel have higher accuracy than the traditional logistic regression model, so we can discard the logistic regression model.

Since both the models have the same accuracy, the Lasso Model is perhaps the best model for predicting the current alcohol use prediction, since the Lasso model is itself a specialized form of the elastic GLM model that is geared towards feature selection and helps us select the best features.

### Step 4: Final Model Evaluation for the Lasso Model

set.seed(123)  
  
lasso.prediction.final <- predict(lasso.model,newdata = a5.data.test)  
lasso.confusion.matrix <- confusionMatrix(lasso.prediction.final, a5.data.test$alc\_consumption, positive = "CurrentUse")  
  
postResample(lasso.prediction.final,a5.data.test$alc\_consumption)

## Accuracy Kappa   
## 0.8548673 0.7028109

lasso.confusion.matrix

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction CurrentUse NotCurrentUse  
## CurrentUse 301 82  
## NotCurrentUse 0 182  
##   
## Accuracy : 0.8549   
## 95% CI : (0.8231, 0.8829)  
## No Information Rate : 0.5327   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.7028   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 1.0000   
## Specificity : 0.6894   
## Pos Pred Value : 0.7859   
## Neg Pred Value : 1.0000   
## Prevalence : 0.5327   
## Detection Rate : 0.5327   
## Detection Prevalence : 0.6779   
## Balanced Accuracy : 0.8447   
##   
## 'Positive' Class : CurrentUse   
##

Model Evalutation Metrics:

**Accuracy : 0.8549 (95% CI: 0.8231, 0.8829)**

**Sensitivity : 1.0000**

**Specificity : 0.6894**

**Balanced Accuracy : 0.8447**

**What research questions could this analysis either a) directly address or b) indirectly help to address by providing information that could be used in subsequent analyses? Limit this response to no more than 1 paragraph. Be sure to use complete sentences.**

Answer:

Based on the above analysis, we can directly address research questions related to **prediction** and **feature selection** which help us predict the risk of a patient’s alcohol consumption behavior based on the optimum features that are a part of our model. This analysis helps us to build and compare different models that can be used to predict risk of alcohol consumption, and choose the most parsimonious and accurate models that are most effective in predicting the outcome variable. We can adapt this analysis to feature selection in a granular way by comparing models with subsets of all features.