P8451\_HW5\_ML

Ruixi Li

2024-02-14

# Introduction

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. You will produce a shareable report that includes code, results and answers to questions using R Markdown.

# Data preparation

set.seed(123)  
alcohol\_use = read\_csv("alcohol\_use.csv")   
skimr::skim(alcohol\_use)

Data summary

|  |  |
| --- | --- |
| Name | alcohol\_use |
| Number of rows | 1885 |
| Number of columns | 9 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 1 |
| numeric | 8 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| --- | --- | --- | --- | --- | --- | --- | --- |
| alc\_consumption | 0 | 1 | 10 | 13 | 0 | 2 | 0 |

**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| …1 | 0 | 1 | 943.00 | 544.30 | 1.00 | 472.00 | 943.00 | 1414.00 | 1885.00 | ▇▇▇▇▇ |
| neurotocism\_score | 0 | 1 | 0.00 | 1.00 | -3.46 | -0.68 | 0.04 | 0.63 | 3.27 | ▁▃▇▅▁ |
| extroversion\_score | 0 | 1 | 0.00 | 1.00 | -3.27 | -0.70 | 0.00 | 0.64 | 3.27 | ▁▃▇▃▁ |
| openness\_score | 0 | 1 | 0.00 | 1.00 | -3.27 | -0.72 | -0.02 | 0.72 | 2.90 | ▁▃▇▆▁ |
| agreeableness\_score | 0 | 1 | 0.00 | 1.00 | -3.46 | -0.61 | -0.02 | 0.76 | 3.46 | ▁▃▇▃▁ |
| conscientiousness\_score | 0 | 1 | 0.00 | 1.00 | -3.46 | -0.65 | -0.01 | 0.58 | 3.46 | ▁▃▇▃▁ |
| impulsiveness\_score | 0 | 1 | 0.01 | 0.95 | -2.56 | -0.71 | -0.22 | 0.53 | 2.90 | ▁▆▇▃▁ |
| sens\_seeking\_score | 0 | 1 | 0.00 | 0.96 | -2.08 | -0.53 | 0.08 | 0.77 | 1.92 | ▂▇▇▇▅ |

alcohol\_use = alcohol\_use[,-1]  
alcohol\_use$alc\_consumption = as.factor(alcohol\_use$alc\_consumption)  
  
#tidyverse way to create data partition  
train.indices <- alcohol\_use %>%  
 pull(alc\_consumption) %>%  
 createDataPartition(p = 0.7, list = FALSE)  
  
train.data <- alcohol\_use %>%  
 slice(train.indices)  
  
test.data <- alcohol\_use %>%  
 slice(-train.indices)  
  
control = trainControl(method = "repeatedcv",   
 number = 10,  
 repeats = 5,  
 selectionFunction = "best")

# Model building

en.model<- train(  
 alc\_consumption ~.,   
 data = train.data,   
 method = "glmnet",  
 trControl = control,   
 preProc=c("center", "scale"),  
 tuneGrid = expand.grid(alpha = seq(0, 1, length = 21),   
 lambda = exp(seq(3, -3, length = 100)))  
 )  
# I chose tuneGrid here, because this command can be used for different scenarios,but using TuneLength, I have to look at all results and see which k is the best.   
  
#Print the values of alpha and lambda that gave best prediction  
en.model$bestTune

## alpha lambda  
## 1131 0.55 0.3067206

# Model coefficients  
coef(en.model$finalModel, en.model$bestTune$lambda)

## 8 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) -0.134172804  
## neurotocism\_score .   
## extroversion\_score .   
## openness\_score .   
## agreeableness\_score .   
## conscientiousness\_score .   
## impulsiveness\_score -0.337985809  
## sens\_seeking\_score -0.003384134

confusionMatrix(en.model)

## Cross-Validated (10 fold, repeated 5 times) Confusion Matrix   
##   
## (entries are percentual average cell counts across resamples)  
##   
## Reference  
## Prediction CurrentUse NotCurrentUse  
## CurrentUse 53.2 14.8  
## NotCurrentUse 0.0 31.9  
##   
## Accuracy (average) : 0.8515

# Define logistic regression model  
log.model <- train(  
 alc\_consumption ~ .,  
 data = train.data,  
 method = "glm",  
 trControl = control)  
  
confusionMatrix(log.model)

## Cross-Validated (10 fold, repeated 5 times) Confusion Matrix   
##   
## (entries are percentual average cell counts across resamples)  
##   
## Reference  
## Prediction CurrentUse NotCurrentUse  
## CurrentUse 43.1 10.4  
## NotCurrentUse 10.2 36.4  
##   
## Accuracy (average) : 0.7945

#Print all of the options examined  
log.model$results

## parameter Accuracy Kappa AccuracySD KappaSD  
## 1 none 0.7945351 0.5871796 0.03363552 0.06772167

# Model coefficients  
log.model$finalModel |> tbl\_regression()

## Table printed with {flextable}, not {gt}. Learn why at  
## https://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html  
## To suppress this message, include `message = FALSE` in the code chunk header.

| **Characteristic** | **log(OR)**1 | **95% CI**1 | **p-value** |
| --- | --- | --- | --- |
| neurotocism\_score | -0.16 | -0.32, 0.00 | 0.057 |
| extroversion\_score | -0.37 | -0.55, -0.20 | <0.001 |
| openness\_score | -0.06 | -0.22, 0.10 | 0.4 |
| agreeableness\_score | -0.11 | -0.27, 0.04 | 0.14 |
| conscientiousness\_score | 0.06 | -0.11, 0.24 | 0.5 |
| impulsiveness\_score | -1.7 | -2.0, -1.5 | <0.001 |
| sens\_seeking\_score | -0.23 | -0.43, -0.03 | 0.023 |
| 1OR = Odds Ratio, CI = Confidence Interval | | | |

summary(log.model)$coef

## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -0.19863180 0.07154542 -2.7763036 5.498084e-03  
## neurotocism\_score -0.15850882 0.08315895 -1.9060946 5.663794e-02  
## extroversion\_score -0.37253442 0.08955505 -4.1598371 3.184746e-05  
## openness\_score -0.06465730 0.08164346 -0.7919471 4.283915e-01  
## agreeableness\_score -0.11424657 0.07772235 -1.4699321 1.415802e-01  
## conscientiousness\_score 0.06382903 0.08737256 0.7305387 4.650610e-01  
## impulsiveness\_score -1.73910364 0.12488310 -13.9258525 4.412448e-44  
## sens\_seeking\_score -0.23073972 0.10167438 -2.2693988 2.324408e-02

set.seed(123)  
  
#Create grid to search lambda  
lambda<-10^seq(-3,3, length=100)  
  
#Note replacing tuneLength with tuneGrid  
la.model<-train(  
 alc\_consumption ~.,   
 data=train.data,   
 method="glmnet",   
 trControl=control,   
 preProc=c("center", "scale"),   
 tuneGrid=expand.grid(alpha=1, lambda=lambda)  
 )  
  
confusionMatrix(la.model)

## Cross-Validated (10 fold, repeated 5 times) Confusion Matrix   
##   
## (entries are percentual average cell counts across resamples)  
##   
## Reference  
## Prediction CurrentUse NotCurrentUse  
## CurrentUse 53.3 14.8  
## NotCurrentUse 0.0 31.9  
##   
## Accuracy (average) : 0.8515

#Print the values of lambda that gave best prediction  
la.model$bestTune

## alpha lambda  
## 40 1 0.231013

# Model coefficients  
coef(la.model$finalModel, la.model$bestTune$lambda)

## 8 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) -0.1329242  
## neurotocism\_score .   
## extroversion\_score .   
## openness\_score .   
## agreeableness\_score .   
## conscientiousness\_score .   
## impulsiveness\_score -0.2730990  
## sens\_seeking\_score .

# Model comparison

models = list(model1 = en.model, model2 = log.model, model3 = la.model)  
results <- resamples(models)  
summary(results)

##   
## Call:  
## summary.resamples(object = results)  
##   
## Models: model1, model2, model3   
## Number of resamples: 50   
##   
## Accuracy   
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  
## model1 0.7819549 0.8333333 0.8522929 0.8515362 0.8693182 0.9160305 0  
## model2 0.7099237 0.7727273 0.7946738 0.7945351 0.8192071 0.8636364 0  
## model3 0.8106061 0.8333333 0.8549618 0.8515324 0.8688372 0.9090909 0  
##   
## Kappa   
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  
## model1 0.5485193 0.6585136 0.6970757 0.6956761 0.7334320 0.8292856 0  
## model2 0.4170960 0.5455848 0.5862950 0.5871796 0.6362600 0.7247451 0  
## model3 0.6108491 0.6585136 0.7025929 0.6957773 0.7315977 0.8145633 0

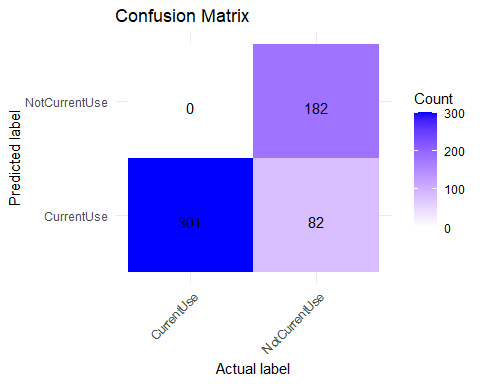
The accuracy for model3(logistic regression with lasso penalty) and model1(elasticnet) are the highest(0.8515), larger than that for model2(traditional logistic regression). I would choose model 3(lasso) as my final model because it has good accuracy and relative lower computational consumption and it is more interpretable compared with model1(elasticnet).

# Model evaluation

la.pred <- la.model %>%   
 predict(test.data)  
  
# Model prediction performance  
confusion\_matrix = confusionMatrix(la.pred,test.data$alc\_consumption)  
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   
##

#visualization  
cm = confusion\_matrix$table  
cm\_melted <- as.data.frame(as.table(cm))  
ggplot(data = cm\_melted, aes(x = Reference, y = Prediction, fill = Freq)) +  
 geom\_tile() +  
 geom\_text(aes(label = sprintf("%d", Freq)), vjust = 1) +  
 scale\_fill\_gradient(low = "white", high = "blue") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 labs(fill = "Count", title = "Confusion Matrix", x = "Actual label", y = "Predicted label")



* Since the outcome is binary, I used confusionMatrix to evaluate this model. Accuracy is about 85.5%, PPV is about 78.6%, sensitivity is 100%, specificty is about 68.9%. These metrics suggest that the model is highly accurate and specific but could improve in terms of specificity to reduce the number of false negatives.

# Research questions

* 1. this model can directly address the research question that which features most strongly predict recent alcohol consumption (currentuse/noncurrentuse), because lasso regression can select features. b) this model help address research question exploring whether interactions between different personality traits increase the predictive power of the model.