MLHW5\_JF

February 11, 2022

# QI: You should create and compare three different models. Remember to remove the ID variable as you do not want to include that in your analysis.

set.seed(100)  
  
alc = read.csv("/Users/judyfordjuoh/Desktop/Machine Learning/ML\_hw5/alcohol\_use.csv")  
  
#Strip off ID Variable  
alc <- alc[,2:9]  
  
#tidyverse way to create data partition (70/30)  
#training.data<-chr$life\_exp %>% createDataPartition(p=0.7, list=F)  
train.indices <- createDataPartition(y = alc$alc\_consumption,p = 0.7,list = FALSE)  
train.data <- alc[train.indices, ]  
test.data <- alc[-train.indices, ]

#REGULARIZED REGRESSION: ELASTIC NET  
set.seed(123)  
  
en.model<- train(  
 alc\_consumption ~., data = train.data, 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 %>% knitr::kable()

|  |  |  |
| --- | --- | --- |
|  | alpha | lambda |
| 54 | 0.6 | 0.2593237 |

#Print all of the options examined. Bc this is a logistic regression we are using the Accuracy. If it was linear regression it would be MSE/RMSE.  
en.model$results %>% knitr::kable()

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| alpha | lambda | Accuracy | Kappa | AccuracySD | KappaSD |
| 0.1 | 0.0003197 | 0.8068250 | 0.6116530 | 0.0210466 | 0.0417630 |
| 0.1 | 0.0007386 | 0.8068250 | 0.6116530 | 0.0210466 | 0.0417630 |
| 0.1 | 0.0017062 | 0.8068250 | 0.6116530 | 0.0210466 | 0.0417630 |
| 0.1 | 0.0039415 | 0.8053040 | 0.6085983 | 0.0216521 | 0.0429586 |
| 0.1 | 0.0091054 | 0.8038003 | 0.6056692 | 0.0227859 | 0.0450955 |
| 0.1 | 0.0210345 | 0.8068537 | 0.6115103 | 0.0236047 | 0.0466990 |
| 0.1 | 0.0485925 | 0.8046213 | 0.6067560 | 0.0276522 | 0.0550164 |
| 0.1 | 0.1122550 | 0.8045868 | 0.6064424 | 0.0239720 | 0.0477174 |
| 0.1 | 0.2593237 | 0.8023313 | 0.6008127 | 0.0250768 | 0.0503857 |
| 0.2 | 0.0003197 | 0.8068250 | 0.6116530 | 0.0210466 | 0.0417630 |
| 0.2 | 0.0007386 | 0.8068250 | 0.6116530 | 0.0210466 | 0.0417630 |
| 0.2 | 0.0017062 | 0.8068250 | 0.6116530 | 0.0210466 | 0.0417630 |
| 0.2 | 0.0039415 | 0.8060616 | 0.6100985 | 0.0216916 | 0.0428928 |
| 0.2 | 0.0091054 | 0.8045636 | 0.6071656 | 0.0222547 | 0.0439656 |
| 0.2 | 0.0210345 | 0.8060903 | 0.6100048 | 0.0222515 | 0.0440854 |
| 0.2 | 0.0485925 | 0.8053672 | 0.6084313 | 0.0289024 | 0.0577240 |
| 0.2 | 0.1122550 | 0.8031004 | 0.6034113 | 0.0263353 | 0.0528628 |
| 0.2 | 0.2593237 | 0.8106304 | 0.6172030 | 0.0205248 | 0.0413541 |
| 0.3 | 0.0003197 | 0.8068250 | 0.6116530 | 0.0210466 | 0.0417630 |
| 0.3 | 0.0007386 | 0.8068250 | 0.6116530 | 0.0210466 | 0.0417630 |
| 0.3 | 0.0017062 | 0.8068250 | 0.6116530 | 0.0210466 | 0.0417630 |
| 0.3 | 0.0039415 | 0.8060616 | 0.6100985 | 0.0216916 | 0.0428928 |
| 0.3 | 0.0091054 | 0.8053097 | 0.6087154 | 0.0232720 | 0.0459896 |
| 0.3 | 0.0210345 | 0.8068364 | 0.6117964 | 0.0226643 | 0.0447603 |
| 0.3 | 0.0485925 | 0.8068824 | 0.6114045 | 0.0282761 | 0.0564700 |
| 0.3 | 0.1122550 | 0.8068711 | 0.6106969 | 0.0232363 | 0.0469137 |
| 0.3 | 0.2593237 | 0.8091498 | 0.6142102 | 0.0266929 | 0.0536112 |
| 0.4 | 0.0003197 | 0.8060674 | 0.6101672 | 0.0204163 | 0.0404696 |
| 0.4 | 0.0007386 | 0.8060674 | 0.6101672 | 0.0204163 | 0.0404696 |
| 0.4 | 0.0017062 | 0.8053040 | 0.6085964 | 0.0207450 | 0.0410856 |
| 0.4 | 0.0039415 | 0.8060616 | 0.6101014 | 0.0228373 | 0.0451965 |
| 0.4 | 0.0091054 | 0.8045578 | 0.6073219 | 0.0228789 | 0.0451379 |
| 0.4 | 0.0210345 | 0.8053212 | 0.6088279 | 0.0231666 | 0.0457426 |
| 0.4 | 0.0485925 | 0.8060903 | 0.6097916 | 0.0276153 | 0.0555893 |
| 0.4 | 0.1122550 | 0.8106361 | 0.6182559 | 0.0241782 | 0.0486815 |
| 0.4 | 0.2593237 | 0.8258284 | 0.6463792 | 0.0307475 | 0.0622587 |
| 0.5 | 0.0003197 | 0.8053098 | 0.6086844 | 0.0209864 | 0.0415868 |
| 0.5 | 0.0007386 | 0.8053098 | 0.6086844 | 0.0209864 | 0.0415868 |
| 0.5 | 0.0017062 | 0.8045465 | 0.6071135 | 0.0212762 | 0.0421250 |
| 0.5 | 0.0039415 | 0.8053040 | 0.6086156 | 0.0222289 | 0.0439458 |
| 0.5 | 0.0091054 | 0.8045578 | 0.6073219 | 0.0228789 | 0.0451379 |
| 0.5 | 0.0210345 | 0.8038003 | 0.6057729 | 0.0251790 | 0.0498949 |
| 0.5 | 0.0485925 | 0.8053270 | 0.6083918 | 0.0281463 | 0.0565584 |
| 0.5 | 0.1122550 | 0.8083747 | 0.6135741 | 0.0218871 | 0.0440979 |
| 0.5 | 0.2593237 | 0.8326582 | 0.6594468 | 0.0315844 | 0.0640640 |
| 0.6 | 0.0003197 | 0.8053098 | 0.6086844 | 0.0209864 | 0.0415868 |
| 0.6 | 0.0007386 | 0.8053098 | 0.6086844 | 0.0209864 | 0.0415868 |
| 0.6 | 0.0017062 | 0.8045465 | 0.6071135 | 0.0212762 | 0.0421250 |
| 0.6 | 0.0039415 | 0.8060674 | 0.6101029 | 0.0219224 | 0.0433862 |
| 0.6 | 0.0091054 | 0.8045578 | 0.6073219 | 0.0228789 | 0.0451379 |
| 0.6 | 0.0210345 | 0.8030426 | 0.6043610 | 0.0265603 | 0.0528394 |
| 0.6 | 0.0485925 | 0.8030484 | 0.6039380 | 0.0281647 | 0.0565485 |
| 0.6 | 0.1122550 | 0.8053615 | 0.6082486 | 0.0325065 | 0.0648413 |
| 0.6 | 0.2593237 | 0.8538419 | 0.7006269 | 0.0278156 | 0.0576231 |
| 0.7 | 0.0003197 | 0.8053098 | 0.6086844 | 0.0209864 | 0.0415868 |
| 0.7 | 0.0007386 | 0.8053098 | 0.6086844 | 0.0209864 | 0.0415868 |
| 0.7 | 0.0017062 | 0.8045465 | 0.6071135 | 0.0212762 | 0.0421250 |
| 0.7 | 0.0039415 | 0.8060731 | 0.6101893 | 0.0209787 | 0.0414685 |
| 0.7 | 0.0091054 | 0.8053269 | 0.6087490 | 0.0231145 | 0.0456319 |
| 0.7 | 0.0210345 | 0.8053039 | 0.6086946 | 0.0254601 | 0.0507333 |
| 0.7 | 0.0485925 | 0.8007814 | 0.5994973 | 0.0294649 | 0.0590639 |
| 0.7 | 0.1122550 | 0.8007871 | 0.5992247 | 0.0304650 | 0.0606293 |
| 0.7 | 0.2593237 | 0.8538419 | 0.7006269 | 0.0278156 | 0.0576231 |
| 0.8 | 0.0003197 | 0.8060674 | 0.6101782 | 0.0216296 | 0.0428337 |
| 0.8 | 0.0007386 | 0.8060674 | 0.6101782 | 0.0216296 | 0.0428337 |
| 0.8 | 0.0017062 | 0.8053040 | 0.6086074 | 0.0219401 | 0.0434166 |
| 0.8 | 0.0039415 | 0.8075825 | 0.6132293 | 0.0219219 | 0.0433788 |
| 0.8 | 0.0091054 | 0.8075997 | 0.6133599 | 0.0229124 | 0.0452608 |
| 0.8 | 0.0210345 | 0.8037830 | 0.6055525 | 0.0245960 | 0.0490087 |
| 0.8 | 0.0485925 | 0.7985085 | 0.5950645 | 0.0276828 | 0.0551632 |
| 0.8 | 0.1122550 | 0.7894230 | 0.5779258 | 0.0375200 | 0.0742009 |
| 0.8 | 0.2593237 | 0.8538419 | 0.7006269 | 0.0278156 | 0.0576231 |
| 0.9 | 0.0003197 | 0.8060674 | 0.6101782 | 0.0216296 | 0.0428337 |
| 0.9 | 0.0007386 | 0.8060674 | 0.6101782 | 0.0216296 | 0.0428337 |
| 0.9 | 0.0017062 | 0.8053040 | 0.6086074 | 0.0219401 | 0.0434166 |
| 0.9 | 0.0039415 | 0.8075825 | 0.6132293 | 0.0219219 | 0.0433788 |
| 0.9 | 0.0091054 | 0.8060845 | 0.6103175 | 0.0240113 | 0.0475010 |
| 0.9 | 0.0210345 | 0.8045578 | 0.6071984 | 0.0223577 | 0.0447568 |
| 0.9 | 0.0485925 | 0.7939514 | 0.5863981 | 0.0286211 | 0.0569848 |
| 0.9 | 0.1122550 | 0.7742596 | 0.5491437 | 0.0358564 | 0.0700477 |
| 0.9 | 0.2593237 | 0.8538419 | 0.7006269 | 0.0278156 | 0.0576231 |
| 1.0 | 0.0003197 | 0.8060674 | 0.6101782 | 0.0216296 | 0.0428337 |
| 1.0 | 0.0007386 | 0.8060674 | 0.6101782 | 0.0216296 | 0.0428337 |
| 1.0 | 0.0017062 | 0.8053040 | 0.6086074 | 0.0219401 | 0.0434166 |
| 1.0 | 0.0039415 | 0.8060673 | 0.6101868 | 0.0222262 | 0.0439950 |
| 1.0 | 0.0091054 | 0.8038060 | 0.6057795 | 0.0224855 | 0.0443863 |
| 1.0 | 0.0210345 | 0.8053154 | 0.6088712 | 0.0245981 | 0.0489503 |
| 1.0 | 0.0485925 | 0.7894173 | 0.5781329 | 0.0291637 | 0.0572677 |
| 1.0 | 0.1122550 | 0.7742596 | 0.5492459 | 0.0358564 | 0.0698728 |
| 1.0 | 0.2593237 | 0.7249250 | 0.4258622 | 0.0604689 | 0.1293269 |

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

## 8 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) -0.136371600  
## neurotocism\_score .   
## extroversion\_score .   
## openness\_score .   
## agreeableness\_score .   
## conscientiousness\_score .   
## impulsiveness\_score -0.419003058  
## sens\_seeking\_score -0.000957449

# Make predictions in test set  
en.pred = en.model %>% predict(test.data)  
  
#Caret doesn't work if your binary response is of character/logical so you have to rename the levels as factors.  
test.data = test.data %>%  
mutate(alc\_consumption = as.factor(alc\_consumption))  
  
# Model prediction performance  
postResample(en.pred,test.data$alc\_consumption) %>% knitr::kable()

|  |  |
| --- | --- |
|  | x |
| Accuracy | 0.8495575 |
| Kappa | 0.6917186 |

#LOGISTIC REGRESSION

#LASSO

# Q2

#Q2 Task: 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.

# Q3

#Q3 Task: Apply your final model in the test set and report your final evaluation metrics.

# Q4

#Q4 Task: Produce a shareable report of your analysis and results using R Markdown.

# Q5

#Q5 Task: 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.

Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.