Amin Fesharaki

6/12/2022

```
library(caret)
library(pROC)
library(AppliedPredictiveModeling)
library(dplyr)
library(MASS)
library(glmnet)
```

Brodnjak-Vonina et al. (2005) develop a methodology for food laboratories to determine the type of oil from a sample. In their procedure, they used a gas chromatograph (an instrument that separates chemicals in a sample) to measure seven different fatty acids in an oil. These measurements would then be used to predict the type of oil in a food sample. To create their model, they used 96 samples of seven types of oils. These data can be found in the caret package using data(oil). The oil types are contained in a factor variable called oilType. The types are pumpkin (coded as A), sunflower (B), peanut (C), olive (D), soybean (E), rapeseed (F), and corn (G). We would like to use these data to build a model that predicts the type of oil-based on a sample's fatty acid percentages.

```
## Warning: package 'glmnet' was built under R version 4.1.2
```

```
data(oil)
```

a) Given the classification imbalance in hepatic injury status, describe how you would create a training and testing set.

```
cat("Check for NA values for each predictor","\n")
```

```
## Check for NA values for each predictor
```

```
sapply(fattyAcids, FUN = function(x) sum(is.na(x))) #No missing values
```

```
##   Palmitic    Stearic      Oleic   Linoleic  Linolenic Eicosanoic Eicosenoic 
##          0          0          0          0          0          0          0
```

```
#Count samples
cat("\nNumber of obs =", nrow(fattyAcids), "\nNumber of predictors =", as.numeric(ncol(fattyAcids)))
```

```
## 
## Number of obs = 96
## Number of predictors = 7
```

```
#Near Zero Variance Check
cat("\nNumber of Near Zero Variance predictors = ")
```

```
##
```

```
## Number of Near Zero Variance predictors =
```

```
nzv(fattyAcids)
```

```
## integer(0)
```

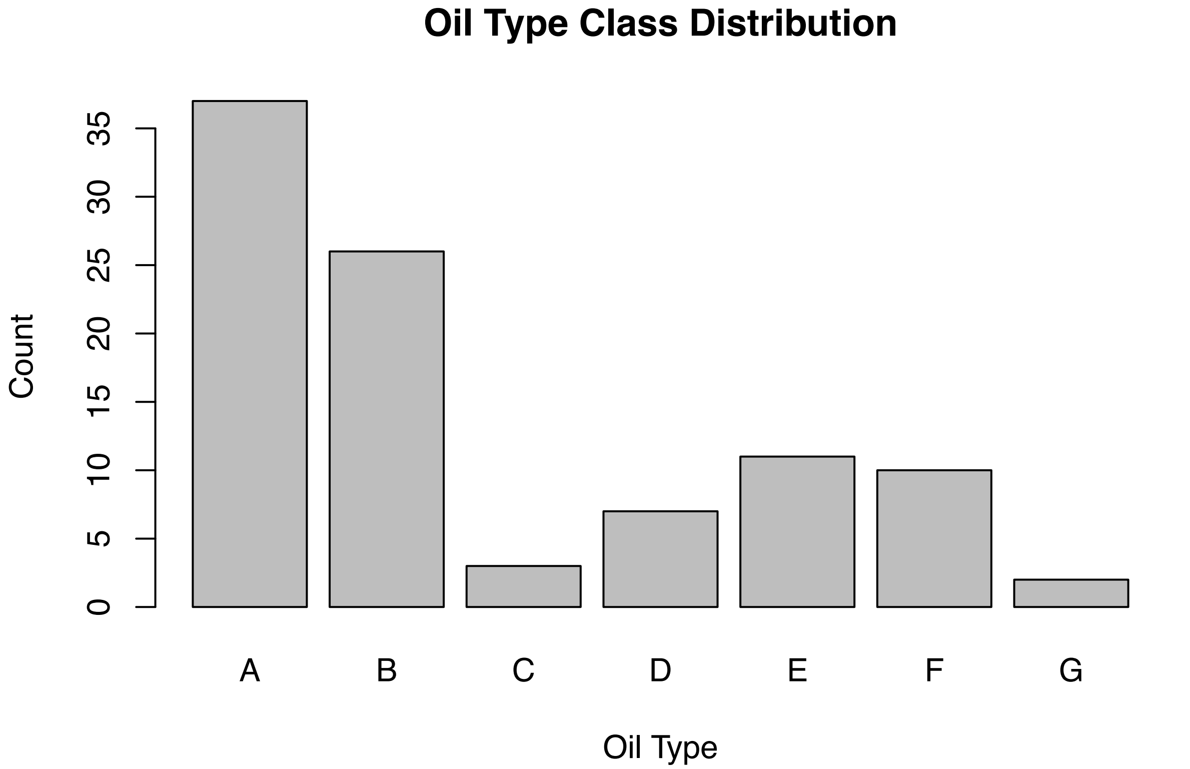
```
#Linear Combination Check
cat("\nCheck for Linear Combos in predictors data set = ")
```

```
##
## Check for Linear Combos in predictors data set =
```

```
findLinearCombos(fattyAcids)$remove
```

```
## NULL
```

```
#Correlations will not be removed since each predictor is an oil type we are tying to predict
#Observe Class Distribution
barplot(table(oilType), main ="Oil Type Class Distribution", xlab ="Oil Type", ylab="Count")
```



The amount of data is limited, therefore we will use the entire data for model building. Thus, the data set will not be split into test and training set. Splitting into the data will result in a very few samples for each class. Instead, cross validation will be used to asses the model.

```
fattyAcids$Type <- oilType
```

b) Which classification statistic would you choose to optimize for this exercise and why?

The optimal classification statistic for this data set would be **Accuracy**; we're trying to predict oil type, and the outcomes seem to share a similar impact (there is not one outcome considered worse to have unlike predictions in th health sector).

c) Build linear discriminant analysis, penalized multinomial regression, and Nearest shrunken centroids models to this data; which model performs best on these data? Which oil type does the model most accurately predict? Which oil type does the model least accurately predict?

```
#Control for all models
ctrl <- trainControl(method = "cv",
                    classProbs = TRUE,
                    savePredictions = TRUE)
```

```
#Pre Process
set.seed(2)
Process <- preProcess(fattyAcids, "center", "scale", "BoxCox")
data <- predict(Process, fattyAcids)
```

```

set.seed(2)
#Create Penalized model

ldaFit = lda(Type ~ Palmitic+Stearic+Oleic+Linoleic+Eicosanoic + Eicosenoic, data = data, cv= TRUE)

#Store Predictions
lda <- predict(ldaFit, data[,1:7])
LDA_pred <- lda$class
testResults <- data.frame(obs = data$Type,
                         LDA = LDA_pred)
table(LDA_pred, data$Type) #Coefficient Matrix
```

Linear Discriminant Analysis

```
##
## LDA_pred  A  B  C  D  E  F  G
##        A 33  0  0  0  0  0  0
##        B  2 26  0  0  0  0  0
##        C  0  0  3  1  0  0  0
##        D  0  0  0  6  0  0  0
##        E  2  0  0  0 11  0  0
##        F  0  0  0  0  0 10  0
##        G  0  0  0  0  0  0  2
```

```

#Create Penalized model

glmnGrid <- expand.grid(alpha = c(0,  .1,  .2, .4, .6, .8, 1),
                       lambda = seq(.01, .2, length = 20))
set.seed(2)
glmnFit <- train(x = data[,1:7],
                 y = data$Type,
                 method = "glmnet",
                 tuneGrid = glmnGrid,
                 preProc = c("center", "scale"),
                 metric = "Accuracy",
                 family = "multinomial",
                 trControl = ctrl)
```

```
testResults$glmnet <- predict(glmnFit, data[,1:7])

glmn_pred <- predict(glmnFit, data[,1:7])
table(glmn_pred, data$Type) #Coefficient Matrix
```

Penalized Regression

```
##
## glmn_pred  A  B  C  D  E  F  G
##         A 36  0  0  0  0  0  0
##         B  1 26  0  0  0  0  1
##         C  0  0  3  0  0  0  0
##         D  0  0  0  7  0  0  0
##         E  0  0  0  0 11  0  1
##         F  0  0  0  0  0 10  0
##         G  0  0  0  0  0  0  0
```

Nearest Shrunken Centroid

```
set.seed(2)
nscFit <- train(x = data[,1:7],
                y = data$Type,
                method = "pam",
                preProc = c("center","scale"),
                metric = "Accuracy",
                tuneGrid = data.frame(threshold = seq(0, 2, length = 50)),
                trControl = ctrl)
```

```
## 111Warning: a class contains only 1 sample1111Warning: a class contains only 1 sample1111
```

```
testResults$nscFit <- predict(nscFit, data[,1:7])
nsc_pred <- predict(nscFit, data[,1:7])
table(nsc_pred, data$Type) #Coefficient Matrix
```

```
##
## nsc_pred  A  B  C  D  E  F  G
##        A 35  0  0  0  0  0  0
##        B  2 26  0  0  0  0  0
##        C  0  0  3  0  0  0  0
##        D  0  0  0  7  0  0  0
##        E  0  0  0  0 11  0  0
##        F  0  0  0  0  0 10  0
##        G  0  0  0  0  0  0  2
```

Accuracy

```
#Model Accuracy
Model_Acurracy <- data.frame(lda = mean(LDA_pred == data$Type))
Model_Acurracy$penalized <- mean(glmn_pred == data$Type)
Model_Acurracy$nsc <- mean(nsc_pred == data$Type)
rownames(Model_Acurracy) = "Model Acuraccy"
Model_Acurracy
```

```
##                      lda penalized       nsc
## Model Acuraccy 0.9479167   0.96875 0.9791667
```

According to the accuracy table (as well as the coefficient matrix), the nearest shrunken neighbors resulted in the most accurate model to predict oil type with an accuracy of 97.9%. NSC classified two A oil types, but other than that, it has predicted every other sample with 100% accuracy. Linear discriminant analysis resulted in the lowest accuracy performance with an accuracy of 94.5%.