STA141A

Fall 2019

Quilvio Hernandez - 914847032

Submit the assignment electronically through Canvas. Electronic submission must be in the form of a zip folder (with extension .zip, .7z, etc.) containing two files: (i) your answer (.pdf file); (ii) R codes used (.R file).

Honor Code: *"The codes and results derived by using these codes constitute my own work. I have consulted the following resources regarding this assignment:"*
https://stackoverflow.com/questions/15030910/randomly-sample-a-percentage-of-rows-within-a-data-frame
https://blog.exploratory.io/merging-two-data-frames-with-union-or-bind-rows-a55e79766d0

1. The data-set `lung.Rdata` contains three variables measured on 835 patients:
   - `biopsy`, a binary variable equal to 1 if the patient has a lung cancer, and 0 otherwise;
   - `smoke_years`, a continuous variable measuring the number of smoking years;
   - `second_hand_years`, a continuous variable measuring the number of second-hand smoking years.

   Set aside 20% of observations with `biopsy` equal to 1 and 0 respectively. Use that sub-sample as the test data set and use the remaining data as the training data set.

2. Use *Linear Discriminant Analysis (LDA)* for classifying the test data. Use `smoke_years` and `second_hand_years` as the predictor variables (or features).

   (a) Report the class-specific means of the predictor variables for the training data.
   The groups means are

	smoke_years	second_hand_years
0	3.976736	6.602909
1	10.280845	5.357525

   (b) Compute the *confusion matrix* for the test data, and the misclassification error rate.

	predicted	
true	0	1
0	156	0
1	3	8

   This is good for an error rate of 0.0179641 or 1.8%.

3. Use the *Logistic Regression* method, fitted to the training data, to classify the test data.

   (a) Fit a logistic regression model to the training data, using the variables `smoke_years` and `second_hand_years` as predictors.

   i. Obtain the estimates and their standard errors for the model parameters.
   Coefficients:

	Estimate Std.	Std. Error
(Intercept)	-11.555	2.0974
smoke_years	1.4834	0.2534
second_hand_years	-.02536	0.1198

   ii. Compute the *confusion matrix* for the test data, and the misclassification error rate.

	predicted	
true	0	1
0	156	0
1	2	9

This is good for an error rate of 0.011976 or 1.2%.

iii. Which is the most relevant predictor for the purpose of classification? Justify.
The most relevant predictor is smoking years. This is backed by its p-value of 4.8e-09, large coefficient, low standard error and high z value.

(b) Fit a logistic regression model to the training data, using the variable `smoke_years` as a one-dimensional predictor.

i. Obtain the estimates and their standard errors for the model parameters.
Coefficients:

	Estimate Std.	Std. Error
(Intercept)	-13.8292	2.0324
smoke_years	1.5903	0.2644

ii. Compute the *confusion matrix* for the test data, and the misclassification error rate.

	predicted	
true	0	1
0	156	0
1	2	9

This is good for an error rate of 0.0179641 or 1.8%.

iii. Compare the results with those in 3(a). Does your result in 3(b)(ii) support the answer to 3(a)(iii) ?
Yes. The confusion matrix is the same for both 3(b)(ii) and 3(a)(iii) suggesting that `smoke_years` is our most relevant predictor for the purpose of classification.

4. Use the *k Nearest Neighbors (kNN)* classification method to classify the test data, using only `smoke_years` as the predictor variable. Perform this analysis using k = 20 and k = 50. In each case, compute the *confusion matrix* for the test data, and the misclassification error rate. How do you explain the results?
The k=20 confusion matrix is:

	predicted	
true	0	1
0	156	0
1	2	9

With an error rate of 0.011976.

The k=50 confusion matrix is:

	predicted	
true	0	1
0	156	0
1	7	4

With an error rate of 0.0419162.

5. Write a very brief summary (maximum of 200 words) about the comparative performance of the three different classification methods for this data set.
All three classification methods appear to be good models for predicting our `biopsy` variable. Using kNN with k=20 and logistic regression appear to the best models of `biopsy` with an error rate of 1.2%. Our worse prediction was with kNN using k=50 which yielded an error rate of 4.2%. This shows that increasing k is not always the optimal choice. In fact, the optimal choice for k in this model is k=19. All three model managed to perfectly predict a patient not having lung cancer. As a result, all the variability in the confusion matrix came from the models power to predic whether or not the patient did have lung cancer. From our test set of 167 observations, kNN(k=20) and logistic regression (both one and two dimensional) had 2 errors, while LDA has 3 errors, and kNN(k=50) had 7 errors.

Appendix

```
knitr::opts_chunk$set(echo = FALSE)
library(dplyr)
library(MASS)
library(class)
library(caret)

setwd("/Users/quilviohernandez/Desktop/Fall 2019/STA 141A/Homeworks/HW3/")
lungdata <- read.table("lung.txt", sep = "\t", header = TRUE, fileEncoding = "UTF-8")

str(lungdata)
head(lungdata)
summary(lungdata)
set.seed(1116)
test <- lungdata %>% group_by(biopsy) %>% sample_frac(.2)
train <- setdiff(lungdata, test)
dim(test)
dim(train)
biop.lda = lda(biopsy ~ smoke_years + second_hand_years, train)
biop.lda
biop.lda.pred = predict(biop.lda, test)
biop.lda.conf = table(true = test$biopsy, predicted = biop.lda.pred$class)
biop.lda.conf
error_rate_lda = 3/167
biop.glm = glm(biopsy ~ smoke_years + second_hand_years, train, family = binomial)
summary(biop.glm)
biop.glm.pred.prob = predict(biop.glm, test, type = "response")
biop.glm.pred = as.numeric(biop.glm.pred.prob > 0.5)
biop.glm.conf = table(true = test$biopsy, predicted = biop.glm.pred)
biop.glm.conf

biop.glm_smoke = glm(biopsy ~ smoke_years, train, family = binomial)
summary(biop.glm_smoke)
biop.glm.pred.prob_smoke = predict(biop.glm, test, type = "response")
biop.glm.pred_smoke = as.numeric(biop.glm.pred.prob > 0.5)
biop.glm.conf_smoke = table(true = test$biopsy, predicted = biop.glm.pred)
biop.glm.conf_smoke
error_rate_glm = 2/167
biop.knn.20 = knn(
  train = train["smoke_years"],
  test = test["smoke_years"],
  cl = train$biopsy,
  k = 20)
biop.knn.20

biop.knn.50 = knn(
  train = train["smoke_years"],
  test = test["smoke_years"],
  cl = train$biopsy,
  k = 50)

biop.knn.conf.20 = table(true = test$biopsy, predicted = biop.knn.20)
```

```
biop.knn.conf.20
error_rate_20 = 2/167
error_rate_20

biop.knn.conf.50 = table(true = test$biopsy, predicted = biop.knn.50)
biop.knn.conf.50
error_rate_50 = 7/167
error_rate_50

trControl <- trainControl(method  = "cv",
                          number  = 5)

fit <- train(as.factor(biopsy) ~ smoke_years,
             method     = "knn",
             tuneGrid   = expand.grid(k = 1:50),
             trControl  = trControl,
             metric     = "Accuracy",
             data       = train
)
```