CPSC 375 Homework 7

Kenn Son, Hamid Suda, Vivian Truong

4/5/2022

```
library(class)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

1. Consider the toy dataset below which shows if 4 subjects have diabetes or not, along with two diagnostic measurements.

Preg	BP	HasDiabetes	Preg.Norm	BP.Norm	Euc.OG	Euc.Norm
2	74	No	0.5	1.0	4	0.5385165
3	58	Yes	1.0	0.2	12.04159	0.781025
2	58	Yes	0.5	0.2	12	.6
1	54	No	0.0	0.0	16.03122	0.9433981
2	70	?	0.5	0.8	0	0

a) Which variable is the "Class" variable?
HasDiabetes is the "Class" variable

b) Normalize the Preg and BP values by scaling the minimum-maximum range of each column to 0-1. Fill in the empty columns in the table.

```
bp <- c(74,58,58,54,70)
preg <- c(2,3,2,1,2)
normalize <- function(x) { return ((x-min(x)) / (max(x)-min(x)) )}
normalize(bp)
```

```
## [1] 1.0 0.2 0.2 0.0 0.8
```

```
normalize(preg)
```

```
## [1] 0.5 1.0 0.5 0.0 0.5
```

c) Predict whether a subject with Preg=2, BP=70 will have diabetes using the 1-NN algorithm and

i) Using Euclidean distance on the original variables

According the the Euclidean distance on the original variables the subject doesn't have diabetes (HasDiabetes = No , 1-NN = Row 1)

ii) Using Euclidean distance on the normalized variables

According the the Euclidean distance on the normalized variables the subject doesn't have diabetes (HasDiabetes = No , 1-NN = Row 1)

For each of these cases, give the nearest distance, nearest neighbor (e.g., "Row 1" or "Row 2"), and prediction.

2. The pima-indians-diabetes-resampled.csv file on Canvas contains records indicating whether the subjects have diabetes or not, along with certain diagnostic measurements. All subjects are of Pima Indian heritage and this dataset is called the Pima Indian Diabetes Database. The goal is to see if it is possible to predict if a subject has diabetes given some of the diagnostic measurements. (Note: this problem is an extension of the classwork assignment; R code from the class is also posted on Canvas.)

a) Read the data file [code]

```
setwd("~/Documents/CPSC-375")
pih <- read.csv("pima-indians-diabetes-resampled.csv")
```

b) What does "Preg" represent in the dataset? (2-3 sentences. Search for the Pima Indian Diabetes Database online and read up on its background.)

"Preg" represents the number of times of pregnacy the patient has had.

Backgroud: All patients are female and 21 years old of pima indian heritage.

c) 0 values in the Glucose column indicate missing values. Remove rows which contain missing values in the Glucose column. You should have 763 rows. [code]

```
pih.no_gluc <- pih %>% filter(Glucose > 0)
pih.no_gluc %>% nrow()
```

```
## [1] 763
```

d) Create three new columns/variables which are the normalized versions of Preg, Pedigree, and Glucose columns, scaling the minimum-maximum range of each column to 0-1 (you can use the code developed in class). [code]

```
pih <- pih.no_gluc %>% mutate(Preg.Norm=normalize(Preg),
                      Pedigree.Norm=normalize(Pedigree),
                      Glucose.Norm=normalize(Glucose))
```

e) Split the dataset into train and test datasets with the first 500 rows for training, and the remaining rows for test. Do NOT randomly sample the data (though resampling is usually done, this hw problem does not use this step for ease of grading).

```
trainindex <- 1:500
```

f) Train and test a k-nearest neighbor classifier with the dataset. Consider only the normalized Preg and Pedigree columns. Set k=1. What is the error rate (number of misclassifications)? [code, error rate]

```
trainfeatures <- pih[trainindex, c(10,11)] #using Normalized Preg and Pedigree
trainlabels <- pih[trainindex, 9]
testindex <-setdiff(1:nrow(pih), trainindex)
testfeatures <- pih[testindex, c(10,11)]
testlabels <- pih[testindex, 9]
predicted <- knn(train = trainfeatures, test = testfeatures, cl=trainlabels, k=1)
## table(testlabels, predicted)
##           predicted
## testlabels   0   1
##          0 120  50
##          1  55  38
```

$$(\text{FP} + \text{FN})/\text{ALL} = (55{+}50)/263 = 0.3992395 = 39.92\%$$

g) Repeat part (f) but consider the normalized Preg, Pedigree, and Glucose columns. Set k=1. What is the error rate? Will the error rate always decrease with a larger number of features? Why or why not: answer in 2-3 sentences? [code, error rate, answer]

```
trainfeatures <- pih[trainindex, c(10,11,12)] #using Normalized Preg and Pedigree
trainlabels <- pih[trainindex, 9]
testindex <-setdiff(1:nrow(pih), trainindex)
testfeatures <- pih[testindex, c(10,11,12)]
testlabels <- pih[testindex, 9]
predicted <- knn(train = trainfeatures, test = testfeatures, cl=trainlabels, k=1)
table(testlabels, predicted)
```

```
##           predicted
## testlabels   0   1
##          0 128  42
##          1  42  51
```

$$(\text{FP} + \text{FN})/\text{ALL} = (42{+}42)/263 = 0.3193916 = 31.94\%$$

The more trained features you apply the error rate will decrease. This is because the test feature will be able to compare to more features leading to more accurate predictions or lower error rate.

h) Repeat part (g) but set k=5. What is the error rate? [code, error rate]

```
trainfeatures <- pih[trainindex, c(10,11,12)] #using Normalized Preg, Pedigree and Glucose
trainlabels <- pih[trainindex, 9]
testindex <-setdiff(1:nrow(pih), trainindex)
testfeatures <- pih[testindex, c(10,11,12)]
testlabels <- pih[testindex, 9]
predicted <- knn(train = trainfeatures, test = testfeatures, cl=trainlabels, k=5)
table(testlabels, predicted)
```

```
##           predicted
## testlabels   0   1
##          0 149  21
##          1  42  51
```

$$(\text{FP} + \text{FN})/\text{ALL} = (42+21)/263 = 0.2395437 = 23.95\%$$

i) Repeat part (h) but set k=11. What is the error rate? Considering your observations from (g)-(i), which is the best value for k? [code, error rate, answer]

```
trainfeatures <- pih[trainindex, c(10,11,12)] #using Normalized Preg, Pedigree and Glucose
trainlabels <- pih[trainindex, 9]
testindex <-setdiff(1:nrow(pih), trainindex)
testfeatures <- pih[testindex, c(10,11,12)]
testlabels <- pih[testindex, 9]
predicted <- knn(train = trainfeatures, test = testfeatures, cl=trainlabels, k=11)
table(testlabels, predicted)
```

```
##           predicted
## testlabels   0   1
##          0 154  16
##          1  42  51
```

$$(\text{FP} + \text{FN})/\text{ALL} = (42+16)/263 = 0.2205323 = 22.05\%$$

We think that the best value for k is 11 since the error rate was lower then the other k-values and we think its due to the fact it has more neighbors to compare with.