Richard Gresham
CPSC 375-01
November 3, 2022

Homework #7

Prepare your answers as a **single PDF file**.
Group work: You may work in groups of 1-3. Include all group member names in the PDF file. Only one person in the group should submit to Canvas.
Due: check on Canvas.

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**
2	74	No	.5	1.0
3	58	Yes	1.0	.2
2	58	Yes	.5	.2
1	54	No	.0	.0
2	70	?	.5	.8

For reference I created a dataframe called toy to do the calculations as seen below. toy <- data.frame(Preg = c(2,3,2,1,2), BP = c(74,58,58,54,70), HasDiabetes = c("No", "Yes", "Yes", "No", NA))

a. Which variable is the “Class” variable?
   The class/label variable here would be HasDiabetes.
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.
   > toy <- toy %>% mutate(Preg.Norm = (Preg - min(Preg))/ (max(Preg) - min(Preg)))
   > toy <- toy %>% mutate(BP.Norm = (BP - min(BP))/ (max(BP) - min(BP)))
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
      (2,70) from new Subject to all other ones find shortest distance.
      1. Sqrt((2 – 2)^2 + (70 – 74)^2) = sqrt((0)^2 + (-4)^2) = sqrt(16) = +4
      2. Sqrt((2 – 3)^2 + (70 – 58)^2) = sqrt((-1)^2 + (12)^2) = sqrt(1 + 144) = sqrt(145) = 12.04159
      3. Sqrt((2 – 2)^2 + (70 – 58)^2) = sqrt((0)^2 + (12)^2) = sqrt(144) = + 12
      4. Sqrt((2 – 1)^2 + (70 – 54)^2) = sqrt((1)^2 + (16)^2) = sqrt(1 + 256) = 16.0312

      Since k = 1 we only look at the closest Euclidean distance which is Row 1 at 4. Thus, the new subject is predicted to not have Diabetes.

ii. Using Euclidean distance on the normalized variables

1. Sqrt((.5 - .5) ^ 2 + (.8 – 1.0)^2) = Sqrt((0)^2 + (-.2)^2)) = Sqrt(.04) = +.2
2. Sqrt((.5 – 1.0)^2 + (.8 - .2)^2) = Sqrt((-.5)^2 + (.6)^2) = Sqrt(.25 + .36) = Sqrt(.61) = .78102496
3. Sqrt((.5 - .5)^2 + (.8 - .2)^2) = Sqrt((0)^2 + (.6)^2 = Sqrt(.36) = .6
4. Sqrt((.5 - .0)^2 + (.8 - .0)^2) = sqrt((.5)^2 + (.8)^2 = Sqrt(.25 + .64) = .943398

   For k = 1 of our normalized variables, the closest one is still Row1 at .2. Thus, the new subject is predicted to not have diabetes.

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[1]. 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]
   pima <- 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 in the pima-indians-diabetes-resampled.csv essentially means the number of times pregnant. Which in this research document on diabetes could be included because of a possible linkage of diabetes and pregnancy, which can be used to determine the onset of diabetes within 5 years. All patients here are female and at least 21 years old of Pimaindians descent.
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] pima <- pima %>% filter(Glucose != 0) # this Filters out all rows where Glucose = 0.
   nrow(pima)
   output: 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]
   normalize function: normalize <- function(x) { return ((x- min(x))/(max(x)-min(x)))}
   pima <- pima %>% mutate(Glu.norm = normalize(Glucose))
   pima <- pima %>% mutate(Preg.norm = normalize(Preg))
   pima <- pima %>% mutate(Ped.norm = normalize(Pedigree))

[1] https://github.com/jbrownlee/Datasets/blob/master/pima-indians-diabetes.names

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

testindex <- 501:763
```

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]

```
trainf <- pima[trainindex, c("Preg.norm", "Ped.norm")]
trainl <- pima[trainindex, "HasDiabetes"]
testf <- pima[testindex, c("Preg.norm", "Ped.norm")]
testl <- pima[testindex, "HasDiabetes"]
prediction2_hasdiabetes <- knn(train = trainf, test = testf, cl = trainl[[1]], k = 1)
table(testl[[1]], prediction2_hasdiabetes)
prediction2_hasdiabetes
     0   1
  0 120  50
  1  56  37
```

Error rate = 1- accuracy or (FP + FN) / ALL

Error rate = (56 + 50) / (120 + 50 + 56 + 37) = [1] 0.4030418

Thus, the error rate is 40.30418%

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] trainf <- pima[trainindex, c("Preg.norm", "Ped.norm", "Glu.norm")]

```
trainl <- pima[trainindex, "HasDiabetes"]
testf <- pima[testindex, c("Preg.norm", "Ped.norm", "Glu.norm")]
testl <- pima[testindex, "HasDiabetes"]
prediction2_hasdiabetes <- knn(train = trainf, test = testf, cl = trainl[[1]], k = 1)
table(testl[[1]], prediction2_hasdiabetes)
prediction2_hasdiabetes
     0   1
  0 128  42
  1  42  51
```

Error rate = (42 + 42) / (128 + 42 + 42 + 51) = .3193916

Thus, the error rate is 31.93916%

While it may look like a good thing that the error rate will always decrease with a larger number of features, it’s important to note that too many features can be a bad thing. If too many features are added it can lead to overfitting which makes our model specifically fit to our data, which is very negative if used on another dataset.

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

Since we are just repeating I shall only repost the prediction.

```
prediction2_hasdiabetes <- knn(train = trainf, test = testf, cl = trainl[[1]], k = 5)
table(testl[[1]], prediction2_hasdiabetes)
   prediction2_hasdiabetes
      0   1
  0 149  21
  1  42  51
```

Error rate = (42 + 21) / (149 + 42 + 21 + 51) = [1] 0.2395437

Thus, our error rate is 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]

```
prediction2_hasdiabetes <- knn(train = trainf, test = testf, cl = trainl[[1]], k = 11)
table(testl[[1]], prediction2_hasdiabetes)
prediction2_hasdiabetes
      0   1
  0 154  16
  1  42  51
```

Error rate = (42 + 16) / (154 + 42 + 16 + 51) = [1] 0.2205323

Thus, our error rate is 22.05323%

When k = 11 is our best value since the error rate is the lowest.