



Assignment 1

Submitted by Maryam Heidari

ALY 6020- Predictive Analytics- CRN: 20770

Northeastern University

Date of Submission: 24 February 2020

Introduction:

In this assignment, we are going to get familiar with KNN method which is one of the simplest methods of classification.

Part A:

In the part one, I am going to use the example in the chapter 3 of Machine Learning with R (Lantz) and follow the five steps. In this example I used the Wisconsin Breast Cancer Diagnostic dataset from the UCI.

The result of each patient is in the M column and you can see that in the below table:

To avoid any misunderstanding, I renamed the B and M with "Benign", "Malignant". Now, the summary table of this column is:

```
Benign Malignant
62.9 37.1
```

Since I have different parameters with different scales, I need to normalize the data set.

```
# normalize these features
normalize <- function(x) {
   return ((x - min(x)) / (max(x) - min(x)))
}
wbcd_n <- as.data.frame(lapply(wbcd[3:32], normalize))</pre>
```

After normalizing the data set, I divided it to two data sets which are train and test, so I can use the train one to make a KNN model and use the test one to evaluate the accuracy of the model.

And also, I need to store these class labels in factor vectors, to generate the KNN model.

```
wbcd_train <- wbcd_n[1:469, ]
wbcd_test <- wbcd_n[470:568, ]
wbcd_train_labels <- wbcd[1:469, 2]
wbcd_test_labels <- wbcd[470:568, 2]</pre>
```

Now, I can make a model. As my train data includes 469 instances, I used k = 21, an odd number roughly equal to the square root of 469. With a two-category outcome, using an odd number eliminates the chance of ending with a tie vote.

Then I used the CrossTable() function to evaluate how good is my model.

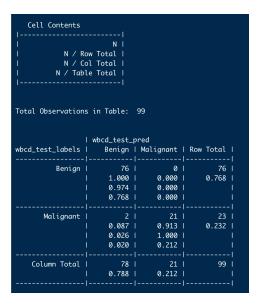
```
library(gmodels)
CrossTable(x = wbcd_test_labels, y = wbcd_test_pred, prop.chisq=FALSE)
```

And the result is:

```
Cell Contents
                       N I
            N / Row Total |
            N / Col Total |
         N / Table Total |
Total Observations in Table: 99
                   wbcd_test_pred
wbcd_test_labels |
                      Benign | Malignant | Row Total |
          Benign I
                          76 I
                                      a
                                                  76
                       1.000 I
                                   0.000 I
                                               0.768
                       0.974 I
                                   0.000
                       0.768 I
                                   0.000 |
                                     21
       Malignant |
                           2 1
                                                  23 I
                       0.087 I
                                   0.913 I
                                               0.232
                       0.026 I
                                   1.000 I
                       0.020 I
                                   0.212
    Column Total |
                          78 I
                                      21
                                                  99 |
                       0.788 I
                                   0.212
```

As you can see, the percentages of true negative results is 76%, the percentages of true positive results is 23%, the percentages of the false positive is 21%, and the percentages of false negative results is 2%.

The desire is to make false negative close to zero, so to improve the model, I standardize the data set, and repeat the steps and the result is:



The result is as same as the last table, and unfortunately it does not improve.

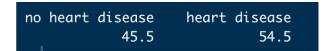
Part B:

In this part, we need to repeat all of the steps for new data set. I find a data set about the heart disease.

The result is in the target column, the table of this column is:

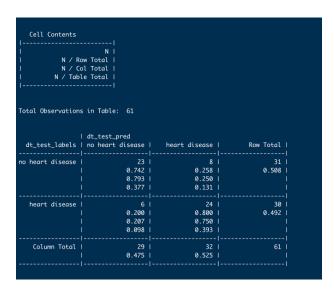


I renamed them by "no heart disease" and "heart disease" to avoid misunderstanding, and the table of parentages is:



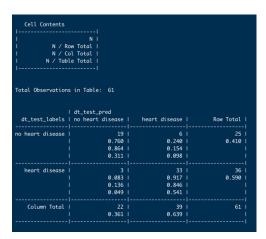
I normalized, divided the data set to the train and test and follow the steps.

As my train data includes 303 instances, I used k = 17, an odd number roughly equal to the square root of 303. With a two-category outcome, using an odd number eliminates the chance of ending with a tie vote.



As you can see, the percentages of true negative results is 23%, the percentages of true positive results is 8%, the percentages of the false positive is 24%, and the percentages of false negative results is 6%.

The desire is to make false negative close to zero, so to improve the model, I standardize the data set, and repeat the steps and the result is:



After standardizing, the percentage of false negative changed from 6% to 3% which is so much better.

Then I changed the K from 17 to 21, And the result is:

Cell Contents			
	N I		
I N/Ro	ow Total		
	ol Total I		
I N / Tabl	e Total I		
Total Observations	in Tables C1		
Total Observations	s in lable: 61		
	dt_test_pred		
		heart disease l	Row Total
		heart disease 6	Row Total 25
dt_test_labels	no heart disease		
dt_test_labels	no heart disease 	 6	 25
dt_test_labels	no heart disease 	 6 0.240	 25
dt_test_labels 	no heart disease 		25 0.410
dt_test_labels	no heart disease 	 6 0.240 0.150	 25
dt_test_labels 	no heart disease 		25 25 0.410
dt_test_labels 	no heart disease 		25 25 0.410
dt_test_labels 	no heart disease 	6 0.240 0.150 0.098	25 0.410
dt_test_labels 	no heart disease 		25 25 0.410

the percentage of false negative changed from 3% to 2% which is so much better. But after that even by increasing the K, I do not see any improvement.

Conclusion:

I learned about the KNN method, how it, how to choose your K number, and how I can improve it.

References:

Ch. 3 of Machine Learning with R (Lantz), in pp. 75-87

http://archive.ics.uci.edu/ml

Kaggle.com