RMarkdown Assignment # 14 - Week 08

Pushkar Chougule

Oct 23rd 2020

## Note - Remove example code and comments before submitting assignment. Producing a professional R Markdown document is the goal.

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

##Read the Binary csv dataset and use summary() on the dataframe.

binary\_csv <- read.csv('binary-classifier-data.csv', header = TRUE)  
  
str(binary\_csv)

## 'data.frame': 1498 obs. of 3 variables:  
## $ label: int 0 0 0 0 0 0 0 0 0 0 ...  
## $ x : num 70.9 75 73.8 66.4 69.1 ...  
## $ y : num 83.2 87.9 92.2 81.1 84.5 ...

summary(binary\_csv)

## label x y   
## Min. :0.000 Min. : -5.20 Min. : -4.019   
## 1st Qu.:0.000 1st Qu.: 19.77 1st Qu.: 21.207   
## Median :0.000 Median : 41.76 Median : 44.632   
## Mean :0.488 Mean : 45.07 Mean : 45.011   
## 3rd Qu.:1.000 3rd Qu.: 66.39 3rd Qu.: 68.698   
## Max. :1.000 Max. :104.58 Max. :106.896

##Fit a binary logistic regression model to the data set

library(caTools)

## Warning: package 'caTools' was built under R version 4.0.3

split <- sample.split(binary\_csv, SplitRatio = 0.8)  
split

## [1] FALSE TRUE TRUE

bin\_train <- subset(binary\_csv, split == "TRUE")  
bin\_test <- subset(binary\_csv, split == "FALSE")  
  
  
log\_reg\_binary <- glm(label ~ ., data = bin\_train, family = "binomial")  
  
summary(log\_reg\_binary)

##   
## Call:  
## glm(formula = label ~ ., family = "binomial", data = bin\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3658 -1.1672 -0.9614 1.1650 1.4004   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.415200 0.143553 2.892 0.003824 \*\*   
## x -0.002376 0.002237 -1.062 0.288036   
## y -0.007953 0.002302 -3.456 0.000549 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1382.9 on 997 degrees of freedom  
## Residual deviance: 1367.6 on 995 degrees of freedom  
## AIC: 1373.6  
##   
## Number of Fisher Scoring iterations: 4

num\_rows <- nrow(binary\_csv)  
num\_rows

## [1] 1498

k\_train <- round(sqrt(nrow(bin\_train)))  
k\_train

## [1] 32

k\_test <- round(sqrt(nrow(bin\_test)))  
k\_test

## [1] 22

**a. What is the accuracy of the logistic regression classifier?**

result <- predict(log\_reg\_binary, bin\_test, type="response")  
  
confusion\_matrix <- table(Actual\_Value = bin\_test$label, Predicted\_Value = result > 0.5)  
confusion\_matrix

## Predicted\_Value  
## Actual\_Value FALSE TRUE  
## 0 143 113  
## 1 98 146

#Accuracy calculation based on confusion matrix  
accuracy = (confusion\_matrix[[1,1]] + confusion\_matrix[[2,2]])/sum(confusion\_matrix) \* 100  
accuracy

## [1] 57.8

**Answer a.**

The above results indicate, based on the original dataset, that the model accuracy is around 58%. There are many False positives and false negatives outcomes from the model. So, this model does not produced very accurate results.

**b. How does the accuracy of the logistic regression classifier compare to the nearest neighbors algorithm?**

Based on the informative link in reference section from <https://www.analyticsvidhya.com>, using square root values as k values for model predictions and accuracy calculations. And using nearest odd numbers as well

since k\_train is 32 (even number), we will use k value as odd numbers closer to 32 such as 31 and 33. Also, we will perform calculations based on the k\_test value of 22 such as values of 21 and 23. Finally we will calculate the model accuracy based on k value of based on total number of rows in the dataset, values such as 38 and 39.

library(class)  
  
cat("calculations based on the square root of total number of rows in test dataset\n")

## calculations based on the square root of total number of rows in test dataset

cat("\n")

knn.out1 <- knn(train=bin\_train, test=bin\_test, cl=bin\_train$label, k=21)  
knn.out2 <- knn(train=bin\_train, test=bin\_test, cl=bin\_train$label, k=23)  
knn.out3 <- knn(train=bin\_train, test=bin\_test, cl=bin\_train$label, k=k\_test)  
  
accuracy\_knn1 <- sum(bin\_test$label == knn.out1)/nrow(bin\_test) \* 100  
accuracy\_knn1

## [1] 98.2

accuracy\_knn2 <- sum(bin\_test$label == knn.out2)/nrow(bin\_test) \* 100  
accuracy\_knn2

## [1] 98.4

accuracy\_knn3 <- sum(bin\_test$label == knn.out3)/nrow(bin\_test) \* 100  
accuracy\_knn3

## [1] 98.4

cat("\n")

cat("calculations based on the square root of total number of rows in train dataset\n")

## calculations based on the square root of total number of rows in train dataset

cat("\n")

knn.out4 <- knn(train=bin\_train, test=bin\_test, cl=bin\_train$label, k=31)  
knn.out5 <- knn(train=bin\_train, test=bin\_test, cl=bin\_train$label, k=33)  
knn.out6 <- knn(train=bin\_train, test=bin\_test, cl=bin\_train$label, k=k\_train)  
  
accuracy\_knn4 <- sum(bin\_test$label == knn.out4)/nrow(bin\_test) \* 100  
accuracy\_knn4

## [1] 97.8

accuracy\_knn5 <- sum(bin\_test$label == knn.out5)/nrow(bin\_test) \* 100  
accuracy\_knn5

## [1] 97.8

accuracy\_knn6 <- sum(bin\_test$label == knn.out6)/nrow(bin\_test) \* 100  
accuracy\_knn6

## [1] 97.8

cat("\n")

cat("calculations based on the square root of total number of rows in full dataset\n")

## calculations based on the square root of total number of rows in full dataset

cat("\n")

knn.out7 <- knn(train=bin\_train, test=bin\_test, cl=bin\_train$label, k=38)  
knn.out8 <- knn(train=bin\_train, test=bin\_test, cl=bin\_train$label, k=39)  
  
accuracy\_knn7 <- sum(bin\_test$label == knn.out7)/nrow(bin\_test) \* 100  
accuracy\_knn7

## [1] 97.6

accuracy\_knn8 <- sum(bin\_test$label == knn.out8)/nrow(bin\_test) \* 100  
accuracy\_knn8

## [1] 97.4

**Answer b.**

As we notice above, K nearest neighbors algorithm produces model with accuracy closer to 98%, which is very good level of accuracy. And this accuracy level is very higher than the logistic regression model’s accuracy

**c. Why is the accuracy of the logistic regression classifier different from that of the nearest neighbors?**

**Answer c.**

Logistic regression model uses generalized linear modelng technique using maximum likelyhood methd. It tries to predict the results/outcome based on the probability value ranging between 0 and 1. But this probability prediction may not be very reliable way, since we use threshold of greater than 0.5 for TRUE / FALSE outcome and we might miss out on the boundary values in predictions. As we noticed eariler, this method produced many false positive and false negative outcomes, which indicates the lower accuracy.

Nearest Neighbor method, however, is a instance based and non-parametric test. It uses training instances as “knowledge” for the prediction phase. It also does not make any assumption of distribution of the data samples. So, based on the data clusters formed during the training phase, it evaluates the test sample provided and predicts the outcome as binary results in testing dataset i.e. in our case, label variable values of 0 and 1. These steps are based on nearness of test dataset points in comparison with existing clusters values. This is makes is highly accurate, as the outcome predictions are based on the available data observations rather than with on a model based on assumption of relationship between variables.

## References

[https://www.analyticsvidhya.com/blog/2015/08/learning-concept-knn-algorithms-programming/#](https://www.analyticsvidhya.com/blog/2015/08/learning-concept-knn-algorithms-programming/):~:text=Unhesitatingly%2C%20using%20kNN%20Algorithm.,points%20into%20well%20defined%20groups

<https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/>