Exercise 14

Taniya Adhikari"

10/26/2020

**Fit a logistic regression model to the binary-classifier-data.csv dataset from the previous assignment.**

library(tinytex)  
classifier\_df <- read.csv("data/binary-classifier-data.csv")  
model.1 = glm(label ~ . , family = binomial, data = classifier\_df)  
summary(model.1)

##   
## Call:  
## glm(formula = label ~ ., family = binomial, data = classifier\_df)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3728 -1.1697 -0.9575 1.1646 1.3989   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.424809 0.117224 3.624 0.00029 \*\*\*  
## x -0.002571 0.001823 -1.411 0.15836   
## y -0.007956 0.001869 -4.257 2.07e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 2075.8 on 1497 degrees of freedom  
## Residual deviance: 2052.1 on 1495 degrees of freedom  
## AIC: 2058.1  
##   
## Number of Fisher Scoring iterations: 4

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

**Answer a**

library(knitr)  
nrow(classifier\_df)

## [1] 1498

classifier\_df$predicted = predict(model.1, newdata=classifier\_df, type="response")  
kable(table(classifier\_df$label, classifier\_df$predicted> 0.5))

|  |  |  |
| --- | --- | --- |
|  | FALSE | TRUE |
| 0 | 429 | 338 |
| 1 | 286 | 445 |

The Model isn’t very accurate. Out of 715 predicted probability less than 0.5, 40% of those values are labeled as 1 which is quite high. Similarly, predicted probability greater than 0.5 has 43% of values labeled as 0.

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

**Answer b**

library(class)  
knn.data <- sample(1:nrow(classifier\_df),size=nrow(classifier\_df)\*0.7,replace = FALSE)  
  
## the normalization function is created  
nor <-function(x) { (x -min(x))/(max(x)-min(x)) }  
  
## Run nomalization for predictor variables.  
classifier\_norm <- as.data.frame(lapply(classifier\_df[,c(2,3)], nor))  
summary(classifier\_norm)

## x y   
## Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.2275 1st Qu.:0.2274   
## Median :0.4278 Median :0.4386   
## Mean :0.4580 Mean :0.4421   
## 3rd Qu.:0.6522 3rd Qu.:0.6556   
## Max. :1.0000 Max. :1.0000

train.data <- classifier\_norm[knn.data,] # 70% training data  
  
test.data <- classifier\_norm[-knn.data,] # remaining 30% test data  
  
## the creating dataframes for "credibility"  
  
train.labels <- classifier\_df[knn.data,1]  
  
test.labels <-classifier\_df[-knn.data,1]  
  
  
k = round(sqrt(NROW(train.labels)), digits = 0)  
  
model.knn = knn(train=train.data, test=test.data, cl=train.labels, k=k)  
  
## create confusion matrix  
tab <- table(model.knn,test.labels)

## this function divides the correct predictions by total number of predictions that tell us how accurate teh model is.  
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) \* 100}  
accuracy(tab)

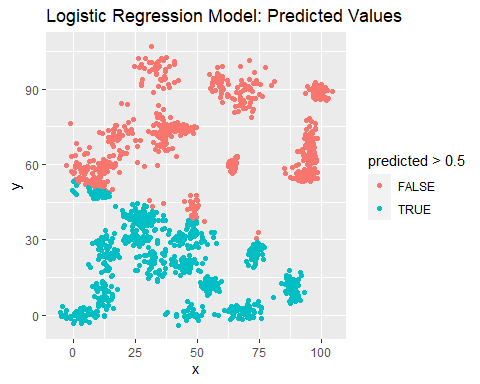
## [1] 97.77778

This KNN Model is much better at predicting values than logistic model. with accuracy of 98%.

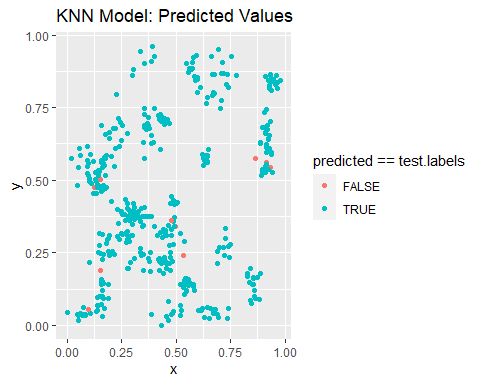
**c. Why is the accuracy of the logistic regression classifier different from that of the nearest neighbors?**

**Answer c**

library(ggplot2)  
## Logistic Model Plot  
ggplot(data = classifier\_df, aes(x = x, y = y, color = predicted > 0.5))+  
 geom\_point() + ggtitle("Logistic Regression Model: Predicted Values")



## Knn Model PLot  
test.data$predicted <- model.knn  
ggplot(data = test.data, aes(x = x, y = y, color = predicted == test.labels)) +   
 geom\_point() +   
 ggtitle("KNN Model: Predicted Values")



In the above Plot of Logistic Regression of Predicted Values, we see that data is randomly dispersed into multiple clusters. Plot doesn’t seem to show any linear relationship. instead True and False values are clustered together. The two variables alone doesn’t seem to work well in this model. When the data is non-linear like this one, knn models works best because it trains the model first and then it predicts the value after which helps minimize the error. The more we train the model with the data the better the prediction it gets.

**Referrences**

1. <https://www.edureka.co/blog/knn-algorithm-in-r/>
2. <https://rstudio-pubs-static.s3.amazonaws.com/506235_848f078b245f4fe885cea65f1528ad79.html#fitting-a-knn-model>
3. <http://www.sthda.com/english/articles/36-classification-methods-essentials/151-logistic-regression-essentials-in-r/#making-predictions>