Untitled

Deepali

1/24/2021

R Markdown

```r
#  Q-2.2-1)...The files credit_card_data.txt (without headers) and credit_card_data-headers.txt (with h




setwd("C:/Users/rash0/OneDrive/Documents/Georgia Tech/Georgia Tech Masters/ISYE6501/FA_SP_hw1/data 2.2")
credit_card_data.headers<-read.delim("credit_card_data-headers.txt", header=TRUE)

library(kernlab)
```

```
## Warning: package 'kernlab' was built under R version 4.0.3
```

```r
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.0.3
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.3
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.0.3
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:kernlab':
##
##     alpha
```

```r
library(ggplot2)
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.3

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v tibble  3.0.5     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
## v purrr   0.3.4

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

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

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

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

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

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

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

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x ggplot2::alpha() masks kernlab::alpha()
## x purrr::cross()   masks kernlab::cross()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x purrr::lift()    masks caret::lift()
```

```r
library(caTools)
```

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

```r
View(credit_card_data.headers)
credits<-credit_card_data.headers
head(credits)
```

```
##   A1    A2    A3   A8 A9 A10 A11 A12 A14 A15 R1
## 1  1 30.83 0.000 1.25  1   0   1   1 202   0  1
## 2  0 58.67 4.460 3.04  1   0   6   1  43 560  1
## 3  0 24.50 0.500 1.50  1   1   0   1 280 824  1
## 4  1 27.83 1.540 3.75  1   0   5   0 100   3  1
## 5  1 20.17 5.625 1.71  1   1   0   1 120   0  1
## 6  1 32.08 4.000 2.50  1   1   0   0 360   0  1
```

```r
str(credits)
```

```
## 'data.frame':    654 obs. of  11 variables:
##  $ A1 : int  1 0 0 1 1 1 1 0 1 1 ...
##  $ A2 : num  30.8 58.7 24.5 27.8 20.2 ...
##  $ A3 : num  0 4.46 0.5 1.54 5.62 ...
##  $ A8 : num  1.25 3.04 1.5 3.75 1.71 ...
##  $ A9 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ A10: int  0 0 1 0 1 1 1 1 1 1 ...
##  $ A11: int  1 6 0 5 0 0 0 0 0 0 ...
##  $ A12: int  1 1 1 0 1 0 0 1 1 0 ...
##  $ A14: int  202 43 280 100 120 360 164 80 180 52 ...
##  $ A15: int  0 560 824 3 0 0 31285 1349 314 1442 ...
##  $ R1 : int  1 1 1 1 1 1 1 1 1 1 ...
```

```r
#Here we are setting the

credits$R1=factor(credits$R1,level=c(0,1))
split=sample.split(credits$R1,SplitRatio=0.70)
train_credits<-subset(credits,split==TRUE)
head(train_credits)
```

```
##   A1    A2     A3   A8 A9 A10 A11 A12 A14   A15 R1
## 1  1 30.83  0.000 1.25  1   0   1   1 202     0  1
## 2  0 58.67  4.460 3.04  1   0   6   1  43   560  1
## 4  1 27.83  1.540 3.75  1   0   5   0 100     3  1
## 5  1 20.17  5.625 1.71  1   1   0   1 120     0  1
## 7  1 33.17  1.040 6.50  1   1   0   0 164 31285  1
## 8  0 22.92 11.585 0.04  1   1   0   1  80  1349  1
```

```r
test_credits<-subset(credits,split==FALSE)
head(test_credits)
```

```
##    A1    A2    A3    A8 A9 A10 A11 A12 A14  A15 R1
## 3   0 24.50 0.500 1.500  1   1   0   1 280  824  1
## 6   1 32.08 4.000 2.500  1   1   0   0 360    0  1
## 10  1 42.50 4.915 3.165  1   1   0   0  52 1442  1
## 12  1 29.92 1.835 4.335  1   1   0   1 260  200  1
## 16  1 36.67 4.415 0.250  1   0  10   0 320    0  1
## 20  0 19.17 8.585 0.750  1   0   7   1  96    0  1
```

```r
#scaling is done to keep the dataset range from 0 to 1

train_credits[-11]=scale(train_credits[-11])
test_credits[-11]=scale(test_credits[-11])

#Case 1:using svm model fitting the testing set for value of C value as 100000

sm_fit1<-ksvm(R1~.,data=train_credits,kernel="vanilladot",type="C-svc",C=100000,scale=TRUE)
```

```
##  Setting default kernel parameters
```

```
a1<-colSums(sm_fit1@xmatrix[[1]] * sm_fit1@coef[[1]])
a0<- -sm_fit1@b
a0
```

```
## [1] 0.04900558
```

```
summary(sm_fit1)
```

```
## Length  Class   Mode
##      1   ksvm     S4
```

```
pred<-predict(sm_fit1,newdata=test_credits[-11])
pred
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
##  [75] 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [112] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1
## [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [186] 0 0 0 0 0 0 0 0 0 0 0
## Levels: 0 1
```

```
#the accuracy of the model is 86%
sum(pred ==test_credits[,11])/nrow(test_credits)
```

```
## [1] 0.9030612
```

```
#case 2

sm_fit2<-ksvm(R1~.,data=train_credits,kernel="vanilladot",type="C-svc",C=1,scale=TRUE)
```

```
##  Setting default kernel parameters
```

```
a1<-colSums(sm_fit2@xmatrix[[1]] * sm_fit2@coef[[1]])
a0<- -sm_fit2@b
a0
```

```
## [1] 0.08733115
```

```
summary(sm_fit2)
```

```
## Length  Class   Mode
##      1   ksvm     S4
```

```
pred<-predict(sm_fit2,newdata=test_credits[-11])

pred
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
##  [75] 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [112] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1
## [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [186] 0 0 0 0 0 0 0 0 0 0 0
## Levels: 0 1
```

```
#the accuracy of the model is 88% for case 2
sum(pred ==test_credits[,11])/nrow(test_credits)
```

```
## [1] 0.8979592
```

```
#Case 3:

sm_fit3<-ksvm(R1~.,data=train_credits,kernel="vanilladot",type="C-svc",C=100000000,scale=TRUE)
```

```
##  Setting default kernel parameters
```

```
a1<-colSums(sm_fit3@xmatrix[[1]] * sm_fit3@coef[[1]])
a0<- -sm_fit3@b
a0
```

```
## [1] 2.900058
```

```
summary(sm_fit3)
```

```
## Length  Class   Mode
##      1   ksvm     S4
```

```
pred<-predict(sm_fit3,newdata=test_credits[-11])

pred
```

```
##   [1] 0 1 0 0 1 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 1 0 1 1 1 0 1 0 0
##  [38] 1 0 1 1 1 1 1 1 0 1 1 1 0 0 1 0 1 1 1 1 1 1 0 0 1 1 0 0 1 0 0 0 0 1 1 0 0
##  [75] 1 0 0 0 1 1 0 0 1 0 1 1 0 0 0 1 0 0 0 0 1 1 0 1 1 0 1 1 0 0 1 0 0 0 0 0 0
## [112] 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 0 1 1 1 0 1 1 1 1 1 0
## [149] 0 1 1 1 0 1 1 0 0 1 1 0 1 0 0 1 0 1 1 0 0 0 1 1 1 1 0 0 0 1 1 0 1 1 0 0 1
## [186] 0 0 1 1 0 1 1 0 0 0 0
## Levels: 0 1
```

```
#the accuracy of the model is 76% for case 2
sum(pred ==test_credits[,11])/nrow(test_credits)
```

```
## [1] 0.5816327
```

```r
#Conclusion_ for my model As the C value is increasing ,the accuracy is decreasing as due to the high v

#Q 2.2-2)3. Using the k-nearest-neighbors classification function kknn contained in the R knn package,

credit_new<-credits[,-11]
view(credit_new)
#normalization is the type of standardized scaling done to ensure that all value are from 0 to 1.
data_norm<-function(x){((x-min(x))/(max(x)-min(x)))}
credit_norm<-as.data.frame(lapply(credit_new,data_norm))
summary(credit_norm[,1:10])
```

```
##        A1               A2               A3                A8
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:0.1328   1st Qu.:0.03714   1st Qu.:0.00579
##  Median :1.0000   Median :0.2212   Median :0.10196   Median :0.03509
##  Mean   :0.6896   Mean   :0.2681   Mean   :0.17252   Mean   :0.07866
##  3rd Qu.:1.0000   3rd Qu.:0.3684   3rd Qu.:0.26562   3rd Qu.:0.09175
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.00000   Max.   :1.00000
##        A9              A10               A11               A12
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000
##  Median :1.0000   Median :1.0000   Median :0.00000   Median :1.0000
##  Mean   :0.5352   Mean   :0.5612   Mean   :0.03729   Mean   :0.5382
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.04478   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.00000   Max.   :1.0000
##       A14                A15
##  Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.03537   1st Qu.:0.00000
##  Median :0.08000   Median :0.00005
##  Mean   :0.09004   Mean   :0.01013
##  3rd Qu.:0.13550   3rd Qu.:0.00399
##  Max.   :1.00000   Max.   :1.00000
```

```r
#creating the training dataset
credit_train<-credit_norm[1:200,]
view(credit_train)
credit_test<-credit_norm[201:654,]
view(credit_test)
library(class)
library(kernlab)
library(caret)
#prediting for 454 test data and training for 200 values,I am checking for squareroot of total number o

KNN_25<-knn(train=credit_train,test=credit_test,cl=credit_new[1:200,1],k=25)
tablecm25<-table(KNN_25,credit_new[201:654,1])
summary(tablecm25)
```

```
## Number of cases in table: 454
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 454, df = 1, p-value = 9.719e-101
```

```r
confusionMatrix(tablecm25)
```

```
## Confusion Matrix and Statistics
##
##
## KNN_25   0   1
##      0 136   0
##      1   0 318
##
##                Accuracy : 1
##                  95% CI : (0.9919, 1)
##     No Information Rate : 0.7004
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.2996
##          Detection Rate : 0.2996
##    Detection Prevalence : 0.2996
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : 0
##
```

```r
#Case 1 :for k=25 ,I am getting around 100 percent accuracy and it  is trying to overfit the model

#CAse 2: for k=50 ,I am getting around 99 percent accuracy for the same training and testing datasets.

KNN_50<-knn(train=credit_train,test=credit_test,cl=credit_new[1:200,1],k=50)
tablecm50<-table(KNN_50,credit_new[201:654,1])
summary(tablecm50)
```

```
## Number of cases in table: 454
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 439.8, df = 1, p-value = 1.175e-97
```

```r
confusionMatrix(tablecm50)
```

```
## Confusion Matrix and Statistics
##
##
## KNN_50   0   1
##      0 133   0
##      1   3 318
##
```

```
##                Accuracy : 0.9934
##                  95% CI : (0.9808, 0.9986)
##     No Information Rate : 0.7004
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9842
##
##  Mcnemar's Test P-Value : 0.2482
##
##             Sensitivity : 0.9779
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9907
##              Prevalence : 0.2996
##          Detection Rate : 0.2930
##    Detection Prevalence : 0.2930
##       Balanced Accuracy : 0.9890
##
##        'Positive' Class : 0
##
```

```
#case3:for k=100 ,I am getting around 94 percent accuracy for the same training and testing datasets.
KNN_100<-knn(train=credit_train,test=credit_test,cl=credit_new[1:200,1],k=100)
tablecm100<-table(KNN_100,credit_new[201:654,1])
summary(tablecm100)
```

```
## Number of cases in table: 454
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 331.4, df = 1, p-value = 4.875e-74
```

```
confusionMatrix(tablecm100)
```

```
## Confusion Matrix and Statistics
##
##
## KNN_100   0   1
##       0 108   0
##       1  28 318
##
##                Accuracy : 0.9383
##                  95% CI : (0.9121, 0.9586)
##     No Information Rate : 0.7004
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8438
##
##  Mcnemar's Test P-Value : 3.352e-07
##
##             Sensitivity : 0.7941
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9191
```

```
##              Prevalence : 0.2996
##          Detection Rate : 0.2379
##    Detection Prevalence : 0.2379
##       Balanced Accuracy : 0.8971
##
##        'Positive' Class : 0
##
```

```
#case4:or k=150 , percent accuracy is 74% for the same training and testing datasets.

KNN_150<-knn(train=credit_train,test=credit_test,cl=credit_new[1:200,1],k=150)
tablecm150<-table(KNN_150,credit_new[201:654,1])
summary(tablecm150)
```

```
## Number of cases in table: 454
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = NaN, df = 1, p-value = NA
##  Chi-squared approximation may be incorrect
```

```
confusionMatrix(tablecm150)
```

```
## Confusion Matrix and Statistics
##
##
## KNN_150   0   1
##       0   0   0
##       1 136 318
##
##                Accuracy : 0.7004
##                  95% CI : (0.656, 0.7423)
##     No Information Rate : 0.7004
##     P-Value [Acc > NIR] : 0.5231
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.0000
##             Specificity : 1.0000
##          Pos Pred Value :    NaN
##          Neg Pred Value : 0.7004
##              Prevalence : 0.2996
##          Detection Rate : 0.0000
##    Detection Prevalence : 0.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 0
##
```

```
#So for my datset the good value of k is 100 for which the accuracy is 94% whch does not allow the mode
```

"'