Classification-Practice-1—Test-data—diabetes-NEW.R

user

2022-05-05

library(readr)

## Warning: package 'readr' was built under R version 4.1.3

testss <- read\_csv("diabetes.csv")

## Rows: 768 Columns: 9  
## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## dbl (9): Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, D...  
##   
## i Use `spec()` to retrieve the full column specification for this data.  
## i Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

names(testss)

## [1] "Pregnancies" "Glucose"   
## [3] "BloodPressure" "SkinThickness"   
## [5] "Insulin" "BMI"   
## [7] "DiabetesPedigreeFunction" "Age"   
## [9] "Outcome"

head(testss)

## # A tibble: 6 x 9  
## Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigre~  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 6 148 72 35 0 33.6 0.627  
## 2 1 85 66 29 0 26.6 0.351  
## 3 8 183 64 0 0 23.3 0.672  
## 4 1 89 66 23 94 28.1 0.167  
## 5 0 137 40 35 168 43.1 2.29   
## 6 5 116 74 0 0 25.6 0.201  
## # ... with 2 more variables: Age <dbl>, Outcome <dbl>

tail(testss)

## # A tibble: 6 x 9  
## Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigre~  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 9 89 62 0 0 22.5 0.142  
## 2 10 101 76 48 180 32.9 0.171  
## 3 2 122 70 27 0 36.8 0.34   
## 4 5 121 72 23 112 26.2 0.245  
## 5 1 126 60 0 0 30.1 0.349  
## 6 1 93 70 31 0 30.4 0.315  
## # ... with 2 more variables: Age <dbl>, Outcome <dbl>

dim(testss)

## [1] 768 9

library(caTools)

## Warning: package 'caTools' was built under R version 4.1.3

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.1.3

##   
## Attaching package: 'dplyr'  
##   
## The following objects are masked from 'package:stats':  
##   
## filter, lag  
##   
## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(ggplot2)   
library(caret)

## Warning: package 'caret' was built under R version 4.1.3

## Loading required package: lattice

## Warning: package 'lattice' was built under R version 4.1.3

library(class)

## Warning: package 'class' was built under R version 4.1.3

library(corrplot)

## Warning: package 'corrplot' was built under R version 4.1.2

## corrplot 0.92 loaded

summary(testss)

## Pregnancies Glucose BloodPressure SkinThickness   
## Min. : 0.000 Min. : 0.0 Min. : 0.00 Min. : 0.00   
## 1st Qu.: 1.000 1st Qu.: 99.0 1st Qu.: 62.00 1st Qu.: 0.00   
## Median : 3.000 Median :117.0 Median : 72.00 Median :23.00   
## Mean : 3.845 Mean :120.9 Mean : 69.11 Mean :20.54   
## 3rd Qu.: 6.000 3rd Qu.:140.2 3rd Qu.: 80.00 3rd Qu.:32.00   
## Max. :17.000 Max. :199.0 Max. :122.00 Max. :99.00   
## Insulin BMI DiabetesPedigreeFunction Age   
## Min. : 0.0 Min. : 0.00 Min. :0.0780 Min. :21.00   
## 1st Qu.: 0.0 1st Qu.:27.30 1st Qu.:0.2437 1st Qu.:24.00   
## Median : 30.5 Median :32.00 Median :0.3725 Median :29.00   
## Mean : 79.8 Mean :31.99 Mean :0.4719 Mean :33.24   
## 3rd Qu.:127.2 3rd Qu.:36.60 3rd Qu.:0.6262 3rd Qu.:41.00   
## Max. :846.0 Max. :67.10 Max. :2.4200 Max. :81.00   
## Outcome   
## Min. :0.000   
## 1st Qu.:0.000   
## Median :0.000   
## Mean :0.349   
## 3rd Qu.:1.000   
## Max. :1.000

str(testss)

## spec\_tbl\_df [768 x 9] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ Pregnancies : num [1:768] 6 1 8 1 0 5 3 10 2 8 ...  
## $ Glucose : num [1:768] 148 85 183 89 137 116 78 115 197 125 ...  
## $ BloodPressure : num [1:768] 72 66 64 66 40 74 50 0 70 96 ...  
## $ SkinThickness : num [1:768] 35 29 0 23 35 0 32 0 45 0 ...  
## $ Insulin : num [1:768] 0 0 0 94 168 0 88 0 543 0 ...  
## $ BMI : num [1:768] 33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...  
## $ DiabetesPedigreeFunction: num [1:768] 0.627 0.351 0.672 0.167 2.288 ...  
## $ Age : num [1:768] 50 31 32 21 33 30 26 29 53 54 ...  
## $ Outcome : num [1:768] 1 0 1 0 1 0 1 0 1 1 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. Pregnancies = col\_double(),  
## .. Glucose = col\_double(),  
## .. BloodPressure = col\_double(),  
## .. SkinThickness = col\_double(),  
## .. Insulin = col\_double(),  
## .. BMI = col\_double(),  
## .. DiabetesPedigreeFunction = col\_double(),  
## .. Age = col\_double(),  
## .. Outcome = col\_double()  
## .. )  
## - attr(\*, "problems")=<externalptr>

standard.features = scale(testss[,1:8])  
  
data = cbind(standard.features,testss[9])  
  
head(data)

## Pregnancies Glucose BloodPressure SkinThickness Insulin BMI  
## 1 0.6395305 0.8477713 0.1495433 0.9066791 -0.6924393 0.2038799  
## 2 -0.8443348 -1.1226647 -0.1604412 0.5305558 -0.6924393 -0.6839762  
## 3 1.2330766 1.9424580 -0.2637694 -1.2873733 -0.6924393 -1.1025370  
## 4 -0.8443348 -0.9975577 -0.1604412 0.1544326 0.1232213 -0.4937213  
## 5 -1.1411079 0.5037269 -1.5037073 0.9066791 0.7653372 1.4088275  
## 6 0.3427574 -0.1530851 0.2528715 -1.2873733 -0.6924393 -0.8108128  
## DiabetesPedigreeFunction Age Outcome  
## 1 0.4681869 1.42506672 1  
## 2 -0.3648230 -0.19054773 0  
## 3 0.6040037 -0.10551539 1  
## 4 -0.9201630 -1.04087112 0  
## 5 5.4813370 -0.02048305 1  
## 6 -0.8175458 -0.27558007 0

summary(data)

## Pregnancies Glucose BloodPressure SkinThickness   
## Min. :-1.1411 Min. :-3.7812 Min. :-3.5703 Min. :-1.2874   
## 1st Qu.:-0.8443 1st Qu.:-0.6848 1st Qu.:-0.3671 1st Qu.:-1.2874   
## Median :-0.2508 Median :-0.1218 Median : 0.1495 Median : 0.1544   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000   
## 3rd Qu.: 0.6395 3rd Qu.: 0.6054 3rd Qu.: 0.5629 3rd Qu.: 0.7186   
## Max. : 3.9040 Max. : 2.4429 Max. : 2.7327 Max. : 4.9187   
## Insulin BMI DiabetesPedigreeFunction  
## Min. :-0.6924 Min. :-4.057829 Min. :-1.1888   
## 1st Qu.:-0.6924 1st Qu.:-0.595191 1st Qu.:-0.6885   
## Median :-0.4278 Median : 0.000941 Median :-0.2999   
## Mean : 0.0000 Mean : 0.000000 Mean : 0.0000   
## 3rd Qu.: 0.4117 3rd Qu.: 0.584390 3rd Qu.: 0.4659   
## Max. : 6.6485 Max. : 4.452906 Max. : 5.8797   
## Age Outcome   
## Min. :-1.0409 Min. :0.000   
## 1st Qu.:-0.7858 1st Qu.:0.000   
## Median :-0.3606 Median :0.000   
## Mean : 0.0000 Mean :0.349   
## 3rd Qu.: 0.6598 3rd Qu.:1.000   
## Max. : 4.0611 Max. :1.000

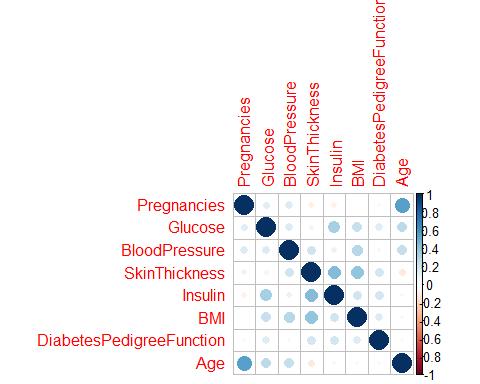
dim(data)

## [1] 768 9

anyNA(data)

## [1] FALSE

corrplot(cor(data[,-9]))



set.seed(550)  
  
sample = sample.split(data$Outcome, SplitRatio = 0.70)  
  
train = subset(data, sample==TRUE)  
dim(train)

## [1] 538 9

test = subset(data, sample==FALSE)  
dim(test)

## [1] 230 9

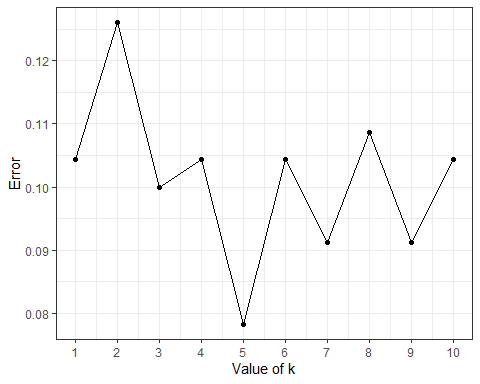
predicted.type = knn(train[1:9], test[1:9], train$Outcome, k=1)  
  
error = mean(predicted.type!=test$Outcome)  
  
confusionMatrix(predicted.type, as.factor(test$Outcome))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 139 13  
## 1 11 67  
##   
## Accuracy : 0.8957   
## 95% CI : (0.8487, 0.932)  
## No Information Rate : 0.6522   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.7687   
##   
## Mcnemar's Test P-Value : 0.8383   
##   
## Sensitivity : 0.9267   
## Specificity : 0.8375   
## Pos Pred Value : 0.9145   
## Neg Pred Value : 0.8590   
## Prevalence : 0.6522   
## Detection Rate : 0.6043   
## Detection Prevalence : 0.6609   
## Balanced Accuracy : 0.8821   
##   
## 'Positive' Class : 0   
##

#The above results reveal that our model achieved an accuracy of 89.57%.  
  
predicted.type = NULL  
error.rate = NULL  
  
for (i in 1:10){  
 predicted.type = knn(train[1:9], test[1:9], train$Outcome, k=i)  
 error.rate[i] = mean(predicted.type!=test$Outcome)  
}  
  
knn.error = as.data.frame(cbind(k=1:10, error.type=error.rate))  
knn.error

## k error.type  
## 1 1 0.10434783  
## 2 2 0.12608696  
## 3 3 0.10000000  
## 4 4 0.10434783  
## 5 5 0.07826087  
## 6 6 0.10434783  
## 7 7 0.09130435  
## 8 8 0.10869565  
## 9 9 0.09130435  
## 10 10 0.10434783

ggplot(knn.error, aes(k,error.type))+  
 geom\_point()+  
 geom\_line()+  
 scale\_x\_continuous(breaks = 1:10)+  
 theme\_bw()+  
 xlab("Value of k")+  
 ylab("Error")



#The above plot reveals that error is lowest when k=5  
  
predicted.type = knn(train[1:9], test[1:9], train$Outcome, k=5)  
  
error = mean(predicted.type!=test$Outcome)  
error

## [1] 0.07826087

confusionMatrix(predicted.type, as.factor(test$Outcome))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 143 11  
## 1 7 69  
##   
## Accuracy : 0.9217   
## 95% CI : (0.8791, 0.953)  
## No Information Rate : 0.6522   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.8255   
##   
## Mcnemar's Test P-Value : 0.4795   
##   
## Sensitivity : 0.9533   
## Specificity : 0.8625   
## Pos Pred Value : 0.9286   
## Neg Pred Value : 0.9079   
## Prevalence : 0.6522   
## Detection Rate : 0.6217   
## Detection Prevalence : 0.6696   
## Balanced Accuracy : 0.9079   
##   
## 'Positive' Class : 0   
##

#Above Model gave us an accuracy of 92.17 %.