Homework10.R

harit

2020-10-28

rm(list=ls())  
cancerdata <- read.table("breast-cancer-wisconsin.data.txt",stringsAsFactors = FALSE,header= FALSE,  
 na.strings = "?", sep=",")  
  
# give meaningful names to the data   
  
colnames(cancerdata)<-c("ID", "Clump\_Thickness", "Cell\_Size",  
 "Cell\_Shape", "Marginal\_Adhesion", "Single\_Epith\_Cell\_Size",   
 "Bare\_Nuclei", "Bland\_Chromatin","Normal\_Nucleoli", "Mitoses", "Class")  
  
cancerdata$Class <- as.factor(cancerdata$Class)  
levels(cancerdata$Class) <-c(0,1)  
#Find the missing data  
head(cancerdata)

## ID Clump\_Thickness Cell\_Size Cell\_Shape Marginal\_Adhesion  
## 1 1000025 5 1 1 1  
## 2 1002945 5 4 4 5  
## 3 1015425 3 1 1 1  
## 4 1016277 6 8 8 1  
## 5 1017023 4 1 1 3  
## 6 1017122 8 10 10 8  
## Single\_Epith\_Cell\_Size Bare\_Nuclei Bland\_Chromatin Normal\_Nucleoli Mitoses  
## 1 2 1 3 1 1  
## 2 7 10 3 2 1  
## 3 2 2 3 1 1  
## 4 3 4 3 7 1  
## 5 2 1 3 1 1  
## 6 7 10 9 7 1  
## Class  
## 1 0  
## 2 0  
## 3 0  
## 4 0  
## 5 0  
## 6 1

summary(cancerdata)

## ID Clump\_Thickness Cell\_Size Cell\_Shape   
## Min. : 61634 Min. : 1.000 Min. : 1.000 Min. : 1.000   
## 1st Qu.: 870688 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 1.000   
## Median : 1171710 Median : 4.000 Median : 1.000 Median : 1.000   
## Mean : 1071704 Mean : 4.418 Mean : 3.134 Mean : 3.207   
## 3rd Qu.: 1238298 3rd Qu.: 6.000 3rd Qu.: 5.000 3rd Qu.: 5.000   
## Max. :13454352 Max. :10.000 Max. :10.000 Max. :10.000   
##   
## Marginal\_Adhesion Single\_Epith\_Cell\_Size Bare\_Nuclei Bland\_Chromatin   
## Min. : 1.000 Min. : 1.000 Min. : 1.000 Min. : 1.000   
## 1st Qu.: 1.000 1st Qu.: 2.000 1st Qu.: 1.000 1st Qu.: 2.000   
## Median : 1.000 Median : 2.000 Median : 1.000 Median : 3.000   
## Mean : 2.807 Mean : 3.216 Mean : 3.545 Mean : 3.438   
## 3rd Qu.: 4.000 3rd Qu.: 4.000 3rd Qu.: 6.000 3rd Qu.: 5.000   
## Max. :10.000 Max. :10.000 Max. :10.000 Max. :10.000   
## NA's :16   
## Normal\_Nucleoli Mitoses Class   
## Min. : 1.000 Min. : 1.000 0:458   
## 1st Qu.: 1.000 1st Qu.: 1.000 1:241   
## Median : 1.000 Median : 1.000   
## Mean : 2.867 Mean : 1.589   
## 3rd Qu.: 4.000 3rd Qu.: 1.000   
## Max. :10.000 Max. :10.000   
##

# Bare Nuclie column has 16 missing values  
  
#Divide the data into training and testing sets  
split\_data <- function(cancerdata)  
{  
 set.seed(499)  
 cancerdata <- subset(cancerdata,select=-c(ID))  
 samplesize <- floor(0.75 \* nrow(cancerdata))  
 Train\_indices <- sample(seq\_len(nrow(cancerdata)),size = samplesize)  
 trainingdata <- cancerdata[Train\_indices,]  
 testingdata <- cancerdata[-Train\_indices,]  
   
 return(list("Training" = trainingdata, "Test" = testingdata))  
}  
  
#Function to predict the best accuracy using  
#K nearest neighbour fuction which takes input as Training set and Test Data set and list of k values.  
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

my\_knn\_func <- function(TrainingSet, TestSet,list\_of\_K)  
{  
 Best\_Accuracy = 0; Best\_K = 0  
   
 for(k\_value in list\_of\_K)  
 {  
 knnmodel <- knn3(Class~., data = TrainingSet, k = k\_value)  
 prediction\_vals <- predict(knnmodel, TestSet[0:(length(TestSet)-1)], type= "class")  
 prediction\_accuracy <- round((sum(prediction\_vals == TestSet$Class)/length(TestSet$Class)),digits=3)  
   
 if(prediction\_accuracy > Best\_Accuracy)  
 {  
 Best\_K <- k\_value  
 Best\_Accuracy <- prediction\_accuracy  
 }  
 }  
 return(list("Best\_K" = Best\_K, "Best\_Accuracy"=Best\_Accuracy))  
}  
  
#Case 1: Remove all the missing rows and run the knn fuction to find the accuracy  
list\_of\_k <- seq(1,15)   
drop\_missing <- na.omit(cancerdata)  
GetSplitData <- split\_data(drop\_missing)  
GetK\_And\_Accuracy <- my\_knn\_func(GetSplitData$Training,GetSplitData$Test,list\_of\_k )  
print("Results after dropping Missing rows:")

## [1] "Results after dropping Missing rows:"

GetK\_And\_Accuracy$Best\_K

## [1] 5

GetK\_And\_Accuracy$Best\_Accuracy

## [1] 0.982

#14.1 1. Use the mean/mode imputation method to impute values for the missing data.  
  
# Impute the mean of the Bare\_Nuclei for the missing values   
cancerdata\_with\_mean <- cancerdata  
cancerdata\_with\_mean$Bare\_Nuclei[is.na(cancerdata\_with\_mean$Bare\_Nuclei)]<-mean(cancerdata\_with\_mean$Bare\_Nuclei,na.rm=TRUE)  
rm(GetSplitData)  
GetSplitData <- split\_data(cancerdata\_with\_mean)  
GetK\_And\_Accuracy <- my\_knn\_func(GetSplitData$Training,GetSplitData$Test,list\_of\_k )  
print("Results after imputing mean:")

## [1] "Results after imputing mean:"

GetK\_And\_Accuracy$Best\_K

## [1] 5

GetK\_And\_Accuracy$Best\_Accuracy

## [1] 0.977

# Impute the mode of Bare Nuclie  
cancerdata\_with\_mode <- cancerdata  
cancerdata\_with\_mode$Bare\_Nuclei[is.na(cancerdata\_with\_mode$Bare\_Nuclei)]<-mode(cancerdata\_with\_mode$Bare\_Nuclei)  
rm(GetSplitData)  
GetSplitData <- split\_data(cancerdata\_with\_mode)  
#GetK\_And\_Accuracy <- my\_knn\_func(GetSplitData$Training,GetSplitData$Test,list\_of\_k )  
print("Results after imputing mode:")

## [1] "Results after imputing mode:"

GetK\_And\_Accuracy$Best\_K

## [1] 5

GetK\_And\_Accuracy$Best\_Accuracy

## [1] 0.977

99.3

## [1] 99.3

#Use Regression to impute the values for the missing data  
library(mice)

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

##   
## Attaching package: 'mice'

## The following objects are masked from 'package:base':  
##   
## cbind, rbind

impute <- mice(cancerdata,method="norm.predict",m=1)

##   
## iter imp variable  
## 1 1 Bare\_Nuclei  
## 2 1 Bare\_Nuclei  
## 3 1 Bare\_Nuclei  
## 4 1 Bare\_Nuclei  
## 5 1 Bare\_Nuclei

data\_impute <- complete(impute)  
  
rm(GetSplitData)  
GetSplitData <- split\_data(data\_impute)  
GetK\_And\_Accuracy <- my\_knn\_func(GetSplitData$Training,GetSplitData$Test,list\_of\_k )  
print("Results after imputing Regression:")

## [1] "Results after imputing Regression:"

GetK\_And\_Accuracy$Best\_K

## [1] 5

GetK\_And\_Accuracy$Best\_Accuracy

## [1] 0.977

#use Regression with perturbation to impute the values for the missing data  
# and run knn  
  
impute <- mice(cancerdata,method="norm.nob",m=1)

##   
## iter imp variable  
## 1 1 Bare\_Nuclei  
## 2 1 Bare\_Nuclei  
## 3 1 Bare\_Nuclei  
## 4 1 Bare\_Nuclei  
## 5 1 Bare\_Nuclei

data\_impute <- complete(impute)  
  
rm(GetSplitData)  
GetSplitData <- split\_data(data\_impute)  
GetK\_And\_Accuracy <- my\_knn\_func(GetSplitData$Training,GetSplitData$Test,list\_of\_k )  
print("Results after imputing Regression:")

## [1] "Results after imputing Regression:"

GetK\_And\_Accuracy$Best\_K

## [1] 3

GetK\_And\_Accuracy$Best\_Accuracy

## [1] 0.977