Assignment3

Harnoor Rangi

2017-08-10

# Reading Url into R

winequality<-read.csv(url("http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white.csv" ), header = TRUE, sep = ";")  
attach(winequality)

# Question 1: Check data characteristics. Is there missing data? Prepare the data for analysis.

#Checking for missing values  
  
sapply(winequality, function(x) sum(is.na(x)))

## fixed.acidity volatile.acidity citric.acid   
## 0 0 0   
## residual.sugar chlorides free.sulfur.dioxide   
## 0 0 0   
## total.sulfur.dioxide density pH   
## 0 0 0   
## sulphates alcohol quality   
## 0 0 0

summary(winequality)

## fixed.acidity volatile.acidity citric.acid residual.sugar   
## Min. : 3.800 Min. :0.0800 Min. :0.0000 Min. : 0.600   
## 1st Qu.: 6.300 1st Qu.:0.2100 1st Qu.:0.2700 1st Qu.: 1.700   
## Median : 6.800 Median :0.2600 Median :0.3200 Median : 5.200   
## Mean : 6.855 Mean :0.2782 Mean :0.3342 Mean : 6.391   
## 3rd Qu.: 7.300 3rd Qu.:0.3200 3rd Qu.:0.3900 3rd Qu.: 9.900   
## Max. :14.200 Max. :1.1000 Max. :1.6600 Max. :65.800   
## chlorides free.sulfur.dioxide total.sulfur.dioxide  
## Min. :0.00900 Min. : 2.00 Min. : 9.0   
## 1st Qu.:0.03600 1st Qu.: 23.00 1st Qu.:108.0   
## Median :0.04300 Median : 34.00 Median :134.0   
## Mean :0.04577 Mean : 35.31 Mean :138.4   
## 3rd Qu.:0.05000 3rd Qu.: 46.00 3rd Qu.:167.0   
## Max. :0.34600 Max. :289.00 Max. :440.0   
## density pH sulphates alcohol   
## Min. :0.9871 Min. :2.720 Min. :0.2200 Min. : 8.00   
## 1st Qu.:0.9917 1st Qu.:3.090 1st Qu.:0.4100 1st Qu.: 9.50   
## Median :0.9937 Median :3.180 Median :0.4700 Median :10.40   
## Mean :0.9940 Mean :3.188 Mean :0.4898 Mean :10.51   
## 3rd Qu.:0.9961 3rd Qu.:3.280 3rd Qu.:0.5500 3rd Qu.:11.40   
## Max. :1.0390 Max. :3.820 Max. :1.0800 Max. :14.20   
## quality   
## Min. :3.000   
## 1st Qu.:5.000   
## Median :6.000   
## Mean :5.878   
## 3rd Qu.:6.000   
## Max. :9.000

sapply(winequality, class)

## fixed.acidity volatile.acidity citric.acid   
## "numeric" "numeric" "numeric"   
## residual.sugar chlorides free.sulfur.dioxide   
## "numeric" "numeric" "numeric"   
## total.sulfur.dioxide density pH   
## "numeric" "numeric" "numeric"   
## sulphates alcohol quality   
## "numeric" "numeric" "integer"

str(winequality)

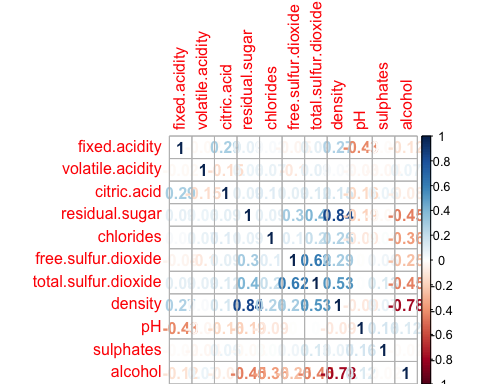
## 'data.frame': 4898 obs. of 12 variables:  
## $ fixed.acidity : num 7 6.3 8.1 7.2 7.2 8.1 6.2 7 6.3 8.1 ...  
## $ volatile.acidity : num 0.27 0.3 0.28 0.23 0.23 0.28 0.32 0.27 0.3 0.22 ...  
## $ citric.acid : num 0.36 0.34 0.4 0.32 0.32 0.4 0.16 0.36 0.34 0.43 ...  
## $ residual.sugar : num 20.7 1.6 6.9 8.5 8.5 6.9 7 20.7 1.6 1.5 ...  
## $ chlorides : num 0.045 0.049 0.05 0.058 0.058 0.05 0.045 0.045 0.049 0.044 ...  
## $ free.sulfur.dioxide : num 45 14 30 47 47 30 30 45 14 28 ...  
## $ total.sulfur.dioxide: num 170 132 97 186 186 97 136 170 132 129 ...  
## $ density : num 1.001 0.994 0.995 0.996 0.996 ...  
## $ pH : num 3 3.3 3.26 3.19 3.19 3.26 3.18 3 3.3 3.22 ...  
## $ sulphates : num 0.45 0.49 0.44 0.4 0.4 0.44 0.47 0.45 0.49 0.45 ...  
## $ alcohol : num 8.8 9.5 10.1 9.9 9.9 10.1 9.6 8.8 9.5 11 ...  
## $ quality : int 6 6 6 6 6 6 6 6 6 6 ...

# Question 2: What is the correlation between the attributes other than wine quality?

par(mfrow = c(1,1))  
cor.white <- cor(winequality[-c(12,13)])  
cor.white

## fixed.acidity volatile.acidity citric.acid  
## fixed.acidity 1.00000000 -0.02269729 0.28918070  
## volatile.acidity -0.02269729 1.00000000 -0.14947181  
## citric.acid 0.28918070 -0.14947181 1.00000000  
## residual.sugar 0.08902070 0.06428606 0.09421162  
## chlorides 0.02308564 0.07051157 0.11436445  
## free.sulfur.dioxide -0.04939586 -0.09701194 0.09407722  
## total.sulfur.dioxide 0.09106976 0.08926050 0.12113080  
## density 0.26533101 0.02711385 0.14950257  
## pH -0.42585829 -0.03191537 -0.16374821  
## sulphates -0.01714299 -0.03572815 0.06233094  
## alcohol -0.12088112 0.06771794 -0.07572873  
## residual.sugar chlorides free.sulfur.dioxide  
## fixed.acidity 0.08902070 0.02308564 -0.0493958591  
## volatile.acidity 0.06428606 0.07051157 -0.0970119393  
## citric.acid 0.09421162 0.11436445 0.0940772210  
## residual.sugar 1.00000000 0.08868454 0.2990983537  
## chlorides 0.08868454 1.00000000 0.1013923521  
## free.sulfur.dioxide 0.29909835 0.10139235 1.0000000000  
## total.sulfur.dioxide 0.40143931 0.19891030 0.6155009650  
## density 0.83896645 0.25721132 0.2942104109  
## pH -0.19413345 -0.09043946 -0.0006177961  
## sulphates -0.02666437 0.01676288 0.0592172458  
## alcohol -0.45063122 -0.36018871 -0.2501039415  
## total.sulfur.dioxide density pH  
## fixed.acidity 0.091069756 0.26533101 -0.4258582910  
## volatile.acidity 0.089260504 0.02711385 -0.0319153683  
## citric.acid 0.121130798 0.14950257 -0.1637482114  
## residual.sugar 0.401439311 0.83896645 -0.1941334540  
## chlorides 0.198910300 0.25721132 -0.0904394560  
## free.sulfur.dioxide 0.615500965 0.29421041 -0.0006177961  
## total.sulfur.dioxide 1.000000000 0.52988132 0.0023209718  
## density 0.529881324 1.00000000 -0.0935914935  
## pH 0.002320972 -0.09359149 1.0000000000  
## sulphates 0.134562367 0.07449315 0.1559514973  
## alcohol -0.448892102 -0.78013762 0.1214320987  
## sulphates alcohol  
## fixed.acidity -0.01714299 -0.12088112  
## volatile.acidity -0.03572815 0.06771794  
## citric.acid 0.06233094 -0.07572873  
## residual.sugar -0.02666437 -0.45063122  
## chlorides 0.01676288 -0.36018871  
## free.sulfur.dioxide 0.05921725 -0.25010394  
## total.sulfur.dioxide 0.13456237 -0.44889210  
## density 0.07449315 -0.78013762  
## pH 0.15595150 0.12143210  
## sulphates 1.00000000 -0.01743277  
## alcohol -0.01743277 1.00000000

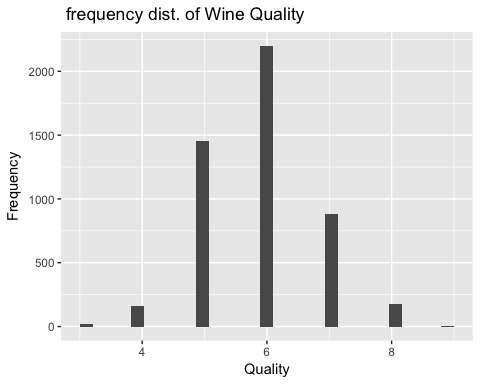
library(corrplot)  
corrplot(cor.white, method = 'number')



# Question 3: Graph the frequency distribution of wine quality.

library(ggplot2)  
qplot(winequality$quality, data=winequality, main=" frequency dist. of Wine Quality",xlab="Quality", ylab="Frequency")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



# Question 4: Reduce the levels of rating for quality to three levels as high, medium and low.

int <- cut(winequality$quality, c(2,5,6,9))  
levels(int) <- c("low", "medium", "high")  
label <- levels(int)  
winequality["labels"] <- NA  
winequality["labels"] <- int  
white <- winequality[-c(12)]

# Question 5: Normalize the data set.

w<-winequality[,c(12,13)]  
winequality<-winequality[,c(-12,-13)]  
  
doit <- function(x) {(x - min(x, na.rm=TRUE))/(max(x,na.rm=TRUE) -  
 min(x, na.rm=TRUE))}  
normed <- as.data.frame(lapply(winequality, doit))  
normed[,c(12,13)]<-w

# Question 6: Divide the data to training and testing groups.

dim(normed)

## [1] 4898 13

#Sample Indexes  
  
  
indexes <- sample(1:nrow(normed), size=0.7\*nrow(normed))  
  
# Split data  
  
  
train <- normed[indexes,]  
dim(train)

## [1] 3428 13

test <- normed[-indexes,]  
dim(test)

## [1] 1470 13

# Question 7: Use the KNN algorithm to predict the quality of wine using its attributes.

train\_labels <- train$labels  
test\_labels <- test$labels  
  
train.set\_new <- train[,c(-12,-13)]  
test.set\_new <- test[,c(-12,-13)]  
  
  
  
library("class")  
library("gmodels")  
knn\_prediction <- knn(train = train.set\_new, test = test.set\_new, cl = train\_labels, k = 5)  
cross <- CrossTable(x=test\_labels, y=knn\_prediction, prop.chisq=FALSE)

##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Row Total |  
## | N / Col Total |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 1470   
##   
##   
## | knn\_prediction   
## test\_labels | low | medium | high | Row Total |   
## -------------|-----------|-----------|-----------|-----------|  
## low | 278 | 175 | 21 | 474 |   
## | 0.586 | 0.369 | 0.044 | 0.322 |   
## | 0.619 | 0.240 | 0.072 | |   
## | 0.189 | 0.119 | 0.014 | |   
## -------------|-----------|-----------|-----------|-----------|  
## medium | 149 | 410 | 112 | 671 |   
## | 0.222 | 0.611 | 0.167 | 0.456 |   
## | 0.332 | 0.562 | 0.385 | |   
## | 0.101 | 0.279 | 0.076 | |   
## -------------|-----------|-----------|-----------|-----------|  
## high | 22 | 145 | 158 | 325 |   
## | 0.068 | 0.446 | 0.486 | 0.221 |   
## | 0.049 | 0.199 | 0.543 | |   
## | 0.015 | 0.099 | 0.107 | |   
## -------------|-----------|-----------|-----------|-----------|  
## Column Total | 449 | 730 | 291 | 1470 |   
## | 0.305 | 0.497 | 0.198 | |   
## -------------|-----------|-----------|-----------|-----------|  
##   
##

# Question 8: Evaluate the model performance.

eva <- data.frame(cross$t)  
evalKNN <- eva  
dim(evalKNN)

## [1] 9 3

x <- c(eva$Freq[1] + eva$Freq[5] + eva$Freq[9])  
y <- x/1470  
y

## [1] 0.5755102