CMTH 642 Data Analytics: Advanced Methods

Assignment 3 (10%)

[Daria Yip]

[DHA - Student#500721106]

```
library(RCurl) #getURL
u <- getURL("http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white.cs
whitewine <- read.csv(text = u, header = TRUE, sep = ";")
#This is a very important dataset to specify the separators,
#otherwise you will have a lot of observations in one column.
```

1. Import to R the following fiel: http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white.csv (The dataset is related to white Portuguese "Vinho Verde" wine. For more info: https://archive.ics.uci.edu/ml/datasets/Wine+Quality) (3 points)

```
str(whitewine)
```

2. Check the datatypes of the attributes. (3 points)

```
## '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 ...
```

```
#All the attributes are num type except quality which is int.
```

```r
sum(is.na(whitewine))
```

3. Are there any missing values in the dataset? (4 points)

```
## [1] 0
```

```r
#There are no missing values in the dataset.
```

```r
correlation <- cor(whitewine[-12]) #want to have correlation of
#every attribute other than Quality which is variable 12
correlation
```

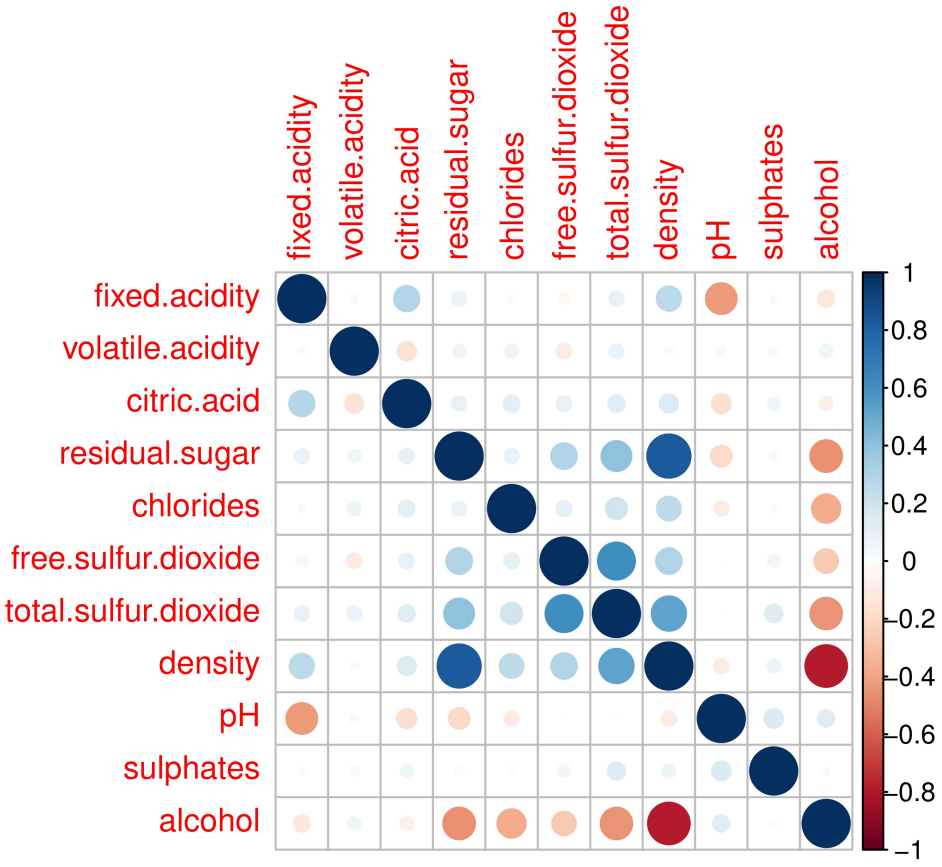
4. What is the correlation between the attributes other than Quality? (10 points)

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

```r
#Visualize the correlation
library(corrplot)
```

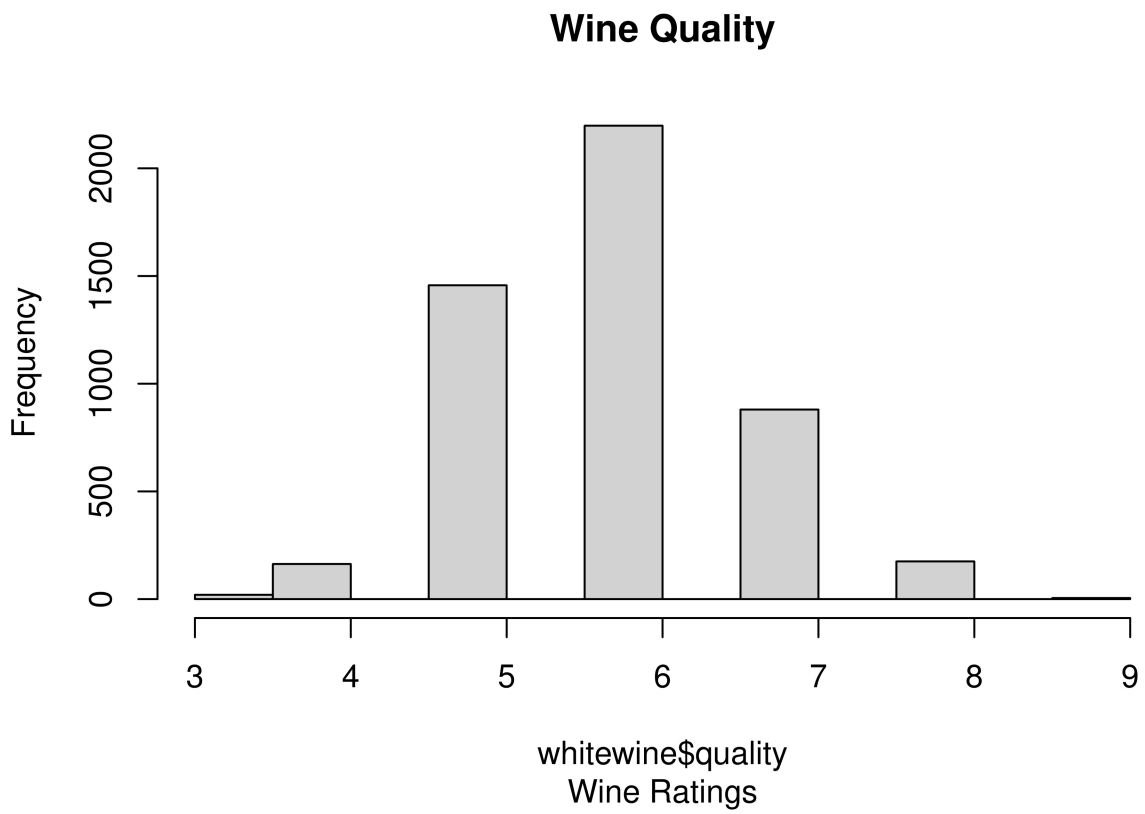
```
## corrplot 0.84 loaded
```

```r
corrplot(correlation)
```




```r
hist(whitewine$quality, main = "Wine Quality", sub = "Wine Ratings")
```

5. Graph the frequency distribution of wine quality by using Quality. (10 points)




```
whitewine$quality <- ifelse(whitewine$quality < 5, "Low", ifelse(whitewine$quality >= 7, "High", "Mediu
head(whitewine)
```

6. Reduce the levels of rating for quality to three levels as high, medium and low. Assign the levels of 3 and 4 to level 0; 5 and 6 to level 1; and 7,8 and 9 to level 2. (10 points)

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1           7.0             0.27        0.36           20.7     0.045
## 2           6.3             0.30        0.34            1.6     0.049
## 3           8.1             0.28        0.40            6.9     0.050
## 4           7.2             0.23        0.32            8.5     0.058
## 5           7.2             0.23        0.32            8.5     0.058
## 6           8.1             0.28        0.40            6.9     0.050
##   free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 1                  45                  170  1.0010 3.00      0.45     8.8
## 2                  14                  132  0.9940 3.30      0.49     9.5
## 3                  30                   97  0.9951 3.26      0.44    10.1
## 4                  47                  186  0.9956 3.19      0.40     9.9
## 5                  47                  186  0.9956 3.19      0.40     9.9
## 6                  30                   97  0.9951 3.26      0.44    10.1
##   quality
## 1  Medium
```

```
## 2  Medium
## 3  Medium
## 4  Medium
## 5  Medium
## 6  Medium
```

```r
normalize <- function(x){
  return ((x - min(x)) / (max(x) - min(x)))
}
```

```r
whitewinen <- as.data.frame(lapply(whitewine[1:11], normalize))
#normalize the dataframe with lapply and only variables 1:11
summary(whitewine[1])
```

7. Normalize the data set by using the following function: (12 points)

```
##  fixed.acidity
##  Min.   : 3.800
##  1st Qu.: 6.300
##  Median : 6.800
##  Mean   : 6.855
##  3rd Qu.: 7.300
##  Max.   :14.200
```

```r
#checking the first variable to see if the data has been
#normalized. We can see it has by the 0-1 scale.
```

```r
trainingtesting <- sample(nrow(whitewine), floor(nrow(whitewine) * 0.7)) #70% is the training set and w
#randomly with the "sample" function, floor will return a
#numeric vector with the largest integers
training <- whitewinen[trainingtesting,]
test <- whitewinen[-trainingtesting,] #testing will be that #which is not the training
```

8. Divide the dataset to training and test sets. (12 points)

```r
#install.packages("class")
library(class)
```

9. Use the KNN algorithm to predict the quality of wine using its attributes. (12 points)

```
## Warning: package 'class' was built under R version 4.0.2
```

```
training_labels <- whitewine[trainingtesting, 12]
test_labels <- whitewine[-trainingtesting, 12]
prediction <- knn(train = training, test = test, cl = training_labels, k = 10)
table(prediction)
```

```
## prediction
##   High    Low Medium
##    258      0   1212
```

```
confusionmatrix <- table(Actual = test_labels, Predicted = prediction)
confusionmatrix
```

10. Display the confusion matrix to evaluate the model performance. (12 points)

```
##         Predicted
## Actual   High Low Medium
##   High    131   0    189
##   Low       2   0     54
##   Medium  125   0    969
```

```
# install.packages("class")
#install.packages("gmodels")
library(gmodels)
```

11. Evaluate the model performance by computing Accuracy, Sensitivity and Specificity. (12 points)

```
## Warning: package 'gmodels' was built under R version 4.0.2
```

```
#CrossTable(x = test_labels, y = prediction, prop.chisq = F)
accuracy <- mean(test_labels == prediction)
accuracy
```

```
## [1] 0.7482993
```

```
error <- mean(test_labels != prediction)
error
```

```
## [1] 0.2517007
```

```
#install.packages("caret", dependencies = TRUE)
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.2
```

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

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

```
test_labels <- as.factor(test_labels) #to make the levels
#the same for the confusionMatrix function

confusionMatrix(prediction, test_labels, positive = "pos")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction High Low Medium
##     High    131   2    125
##     Low       0   0      0
##     Medium  189  54    969
##
## Overall Statistics
##
##                Accuracy : 0.7483
##                  95% CI : (0.7253, 0.7703)
##     No Information Rate : 0.7442
##     P-Value [Acc > NIR] : 0.3728
##
##                   Kappa : 0.2771
##
##  Mcnemar's Test P-Value : 6.836e-15
##
## Statistics by Class:
##
##                      Class: High Class: Low Class: Medium
## Sensitivity              0.40937     0.0000        0.8857
## Specificity              0.88957     1.0000        0.3537
## Pos Pred Value           0.50775        NaN        0.7995
## Neg Pred Value           0.84406     0.9619        0.5155
## Prevalence               0.21769     0.0381        0.7442
## Detection Rate           0.08912     0.0000        0.6592
## Detection Prevalence     0.17551     0.0000        0.8245
## Balanced Accuracy        0.64947     0.5000        0.6197
```

```
#The model below shows the stats of accuracy which is 78.91%.
#Sensitivity = TruePositives/(TruePositives + FalseNegatives)
#Specificity is the True Negatives proportion.
#Specificity = TrueNegatives/(TrueNegatives + FalseNegatives)
#Precision = TruePositives/(TruePositives + FalsePositives).
```

This is the end of Assignment 3

Ceni Babaoglu, PhD