February 14, 2018

Assignment 2

Sanchit Singhal

INF 397 – Statistical Analysis and Learning w/ Prof. Varun Rai

Spring 2018

The University of Texas at Austin

library(class)  
  
# Read CSV from working directory into R  
  
MyData <- read.csv(file="redwine.csv", header=TRUE, sep=",")  
  
### Question 2  
  
## a.   
  
# Numerical summary of data  
  
str(MyData)

## 'data.frame': 1599 obs. of 12 variables:  
## $ fixed.acidity : num 7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...  
## $ volatile.acidity : num 0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...  
## $ citric.acid : num 0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...  
## $ residual.sugar : num 1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...  
## $ chlorides : num 0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ...  
## $ free.sulfur.dioxide : num 11 25 15 17 11 13 15 15 9 17 ...  
## $ total.sulfur.dioxide: num 34 67 54 60 34 40 59 21 18 102 ...  
## $ density : num 0.998 0.997 0.997 0.998 0.998 ...  
## $ pH : num 3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...  
## $ sulphates : num 0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...  
## $ alcohol : num 9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...  
## $ quality : int 5 5 5 6 5 5 5 7 7 5 ...

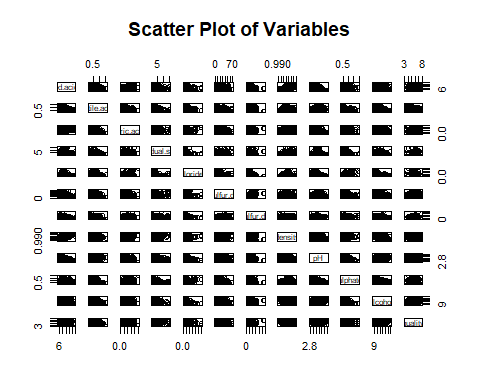
library(psych)  
describe(MyData)

## vars n mean sd median trimmed mad min  
## fixed.acidity 1 1599 8.32 1.74 7.90 8.15 1.48 4.60  
## volatile.acidity 2 1599 0.53 0.18 0.52 0.52 0.18 0.12  
## citric.acid 3 1599 0.27 0.19 0.26 0.26 0.25 0.00  
## residual.sugar 4 1599 2.54 1.41 2.20 2.26 0.44 0.90  
## chlorides 5 1599 0.09 0.05 0.08 0.08 0.01 0.01  
## free.sulfur.dioxide 6 1599 15.87 10.46 14.00 14.58 10.38 1.00  
## total.sulfur.dioxide 7 1599 46.47 32.90 38.00 41.84 26.69 6.00  
## density 8 1599 1.00 0.00 1.00 1.00 0.00 0.99  
## pH 9 1599 3.31 0.15 3.31 3.31 0.15 2.74  
## sulphates 10 1599 0.66 0.17 0.62 0.64 0.12 0.33  
## alcohol 11 1599 10.42 1.07 10.20 10.31 1.04 8.40  
## quality 12 1599 5.64 0.81 6.00 5.59 1.48 3.00  
## max range skew kurtosis se  
## fixed.acidity 15.90 11.30 0.98 1.12 0.04  
## volatile.acidity 1.58 1.46 0.67 1.21 0.00  
## citric.acid 1.00 1.00 0.32 -0.79 0.00  
## residual.sugar 15.50 14.60 4.53 28.49 0.04  
## chlorides 0.61 0.60 5.67 41.53 0.00  
## free.sulfur.dioxide 72.00 71.00 1.25 2.01 0.26  
## total.sulfur.dioxide 289.00 283.00 1.51 3.79 0.82  
## density 1.00 0.01 0.07 0.92 0.00  
## pH 4.01 1.27 0.19 0.80 0.00  
## sulphates 2.00 1.67 2.42 11.66 0.00  
## alcohol 14.90 6.50 0.86 0.19 0.03  
## quality 8.00 5.00 0.22 0.29 0.02

summary(MyData)

## fixed.acidity volatile.acidity citric.acid residual.sugar   
## Min. : 4.60 Min. :0.1200 Min. :0.000 Min. : 0.900   
## 1st Qu.: 7.10 1st Qu.:0.3900 1st Qu.:0.090 1st Qu.: 1.900   
## Median : 7.90 Median :0.5200 Median :0.260 Median : 2.200   
## Mean : 8.32 Mean :0.5278 Mean :0.271 Mean : 2.539   
## 3rd Qu.: 9.20 3rd Qu.:0.6400 3rd Qu.:0.420 3rd Qu.: 2.600   
## Max. :15.90 Max. :1.5800 Max. :1.000 Max. :15.500   
## chlorides free.sulfur.dioxide total.sulfur.dioxide  
## Min. :0.01200 Min. : 1.00 Min. : 6.00   
## 1st Qu.:0.07000 1st Qu.: 7.00 1st Qu.: 22.00   
## Median :0.07900 Median :14.00 Median : 38.00   
## Mean :0.08747 Mean :15.87 Mean : 46.47   
## 3rd Qu.:0.09000 3rd Qu.:21.00 3rd Qu.: 62.00   
## Max. :0.61100 Max. :72.00 Max. :289.00   
## density pH sulphates alcohol   
## Min. :0.9901 Min. :2.740 Min. :0.3300 Min. : 8.40   
## 1st Qu.:0.9956 1st Qu.:3.210 1st Qu.:0.5500 1st Qu.: 9.50   
## Median :0.9968 Median :3.310 Median :0.6200 Median :10.20   
## Mean :0.9967 Mean :3.311 Mean :0.6581 Mean :10.42   
## 3rd Qu.:0.9978 3rd Qu.:3.400 3rd Qu.:0.7300 3rd Qu.:11.10   
## Max. :1.0037 Max. :4.010 Max. :2.0000 Max. :14.90   
## quality   
## Min. :3.000   
## 1st Qu.:5.000   
## Median :6.000   
## Mean :5.636   
## 3rd Qu.:6.000   
## Max. :8.000

# Graphical summary of data  
  
pairs(MyData, main="Scatter Plot of Variables")



# Correlation of attributes  
  
z <- cor(MyData)   
z[lower.tri(z,diag=TRUE)]=NA   
z=as.data.frame(as.table(z))   
z=na.omit(z)   
z=z[order(-abs(z$Freq)),]   
head(z, n=10)

## Var1 Var2 Freq  
## 97 fixed.acidity pH -0.6829782  
## 25 fixed.acidity citric.acid 0.6717034  
## 85 fixed.acidity density 0.6680473  
## 78 free.sulfur.dioxide total.sulfur.dioxide 0.6676665  
## 26 volatile.acidity citric.acid -0.5524957  
## 99 citric.acid pH -0.5419041  
## 128 density alcohol -0.4961798  
## 143 alcohol quality 0.4761663  
## 134 volatile.acidity quality -0.3905578  
## 113 chlorides sulphates 0.3712605

# After exploring the dataset, I generated a correlation matrix to understand  
 # how the attributes are related to each other. I created a list of these  
 # relationships, sorted them, and printed out the ones with the largest absolute  
 # values. Obviously, a correlation of 1 would be the highest signify a perfect,  
 # positive correlation whereas a -1 would signify a perfect, negative  
 # correlation.  
  
 # The highest correlation patterns in the data seem to be between fixed.acidity  
 # & pH, fixed.acidity % citric.acid, fixed.acidity & density, and  
 # freesulfur.dioxide & total.sulfur.dioxide. All of them have roughly a 0.68  
 # correlation - only fixed.acidity and pH have an inverse relationship. Although  
 # these correlations are not extremely high, they indicate that there is some  
 # dependency of variables on each other which is potentially harmful to us when  
 # building a model.  
  
## b.   
  
# Create binary variable final\_quality using mean  
  
MyData$final\_quality <- with(ifelse(quality>mean(quality), 1, 0), data=MyData)  
  
  
# Creating dataset without original quality attribute  
  
myvars <- names(MyData) %in% c("quality")  
fulldataset <- MyData[!myvars]  
head(fulldataset, n=1)

## fixed.acidity volatile.acidity citric.acid residual.sugar chlorides  
## 1 7.4 0.7 0 1.9 0.076  
## free.sulfur.dioxide total.sulfur.dioxide density pH sulphates alcohol  
## 1 11 34 0.9978 3.51 0.56 9.4  
## final\_quality  
## 1 0

# Splitting data into train and test  
  
set.seed(1)  
rows <- sample(x=nrow(fulldataset), size=.80\*nrow(fulldataset))  
trainset <- fulldataset[rows, ]  
testset <- fulldataset[-rows, ]  
  
# Logistic Regression  
  
glm.fit <- glm(final\_quality ~ fixed.acidity+volatile.acidity+citric.acid+residual.sugar+chlorides+free.sulfur.dioxide+total.sulfur.dioxide+density+pH+sulphates+alcohol, data=trainset, family=binomial)  
summary(glm.fit)

##   
## Call:  
## glm(formula = final\_quality ~ fixed.acidity + volatile.acidity +   
## citric.acid + residual.sugar + chlorides + free.sulfur.dioxide +   
## total.sulfur.dioxide + density + pH + sulphates + alcohol,   
## family = binomial, data = trainset)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.3285 -0.8640 0.3094 0.8477 2.2338   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 19.150545 85.590617 0.224 0.82296   
## fixed.acidity 0.111649 0.105894 1.054 0.29173   
## volatile.acidity -2.844256 0.534364 -5.323 1.02e-07 \*\*\*  
## citric.acid -0.988874 0.630212 -1.569 0.11662   
## residual.sugar 0.006617 0.061182 0.108 0.91388   
## chlorides -4.933251 1.738512 -2.838 0.00454 \*\*   
## free.sulfur.dioxide 0.016142 0.009139 1.766 0.07734 .   
## total.sulfur.dioxide -0.014461 0.003119 -4.636 3.55e-06 \*\*\*  
## density -25.095517 87.419908 -0.287 0.77406   
## pH -0.921976 0.793994 -1.161 0.24557   
## sulphates 2.567119 0.499801 5.136 2.80e-07 \*\*\*  
## alcohol 0.872373 0.113699 7.673 1.68e-14 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1770.7 on 1278 degrees of freedom  
## Residual deviance: 1352.0 on 1267 degrees of freedom  
## AIC: 1376  
##   
## Number of Fisher Scoring iterations: 4

# Atleast 4 of the predictors appear to be statistically significant. alcohol,  
 # sulphates, total.sulfur.dioxide, and volative.acidity all have very small p  
 # values meaning they have the largest impact on the response variable of final  
 # quality.  
  
  
## c.   
  
# Confusion Matrix  
  
glm.probs <- predict(glm.fit, testset, type="response")  
glm.preds <- ifelse(glm.probs>0.5, 1, 0)  
confusion\_matrix\_glm <- table(testset$final\_quality,glm.preds)  
print(confusion\_matrix\_glm)

## glm.preds  
## 0 1  
## 0 105 27  
## 1 45 143

# Fraction of Correct Predictions  
  
Correct\_Predictions\_fraction = (confusion\_matrix\_glm[1,1]+confusion\_matrix\_glm[2,2])/sum(confusion\_matrix\_glm)  
sprintf("Overall Fraction of Correct Predictions are: %f",Correct\_Predictions\_fraction)

## [1] "Overall Fraction of Correct Predictions are: 0.775000"

# The confusion tells us about the performance of the model. There were True  
 # Negatives (105) and True Positives (143) and, as can be seen above, constitute  
 # about 77.5% of results. The rest were misclassified - so about 22.5%. 27  
 # points were False Positives and 45 were False Negatives. This tells us our  
 # logistic regression model is wrong about one fourth of the time and tends to  
 # be too conservative. It is wrongly classifying good wine (in class 1 that have  
 # quality above the mean) as bad wine (in class 0 with quality below mean) more  
 # than it is classifying bad wine as good wine (although that is happening a  
 # fair bit as well).  
  
## d.   
  
variables <- which(names(fulldataset)%in%c("fixed.acidity","volatile.acidity","citric.acid","residual.sugar","chlorides","free.sulfur.dioxide","total.sulfur.dioxide","density","pH","sulphates","alcohol"))  
  
test\_error <- data.frame("k"=1:11)  
  
set.seed(1)  
for(k in 1:11)  
 {  
 knn.pred <- knn(train=trainset[, variables], test=testset[, variables], cl=trainset$final\_quality, k=k)  
 test\_error$error[k]= round(sum(knn.pred!=testset$final\_quality)/nrow(testset)\*100,2)  
 }  
  
print(test\_error)

## k error  
## 1 1 25.62  
## 2 2 35.00  
## 3 3 34.06  
## 4 4 37.19  
## 5 5 36.25  
## 6 6 37.50  
## 7 7 32.81  
## 8 8 32.50  
## 9 9 34.06  
## 10 10 34.69  
## 11 11 31.87

# As seen above, the model with the lowest test error is when k=1 with an error  
# of approximately 25 and therefore can be concluded to be performing the best  
# on this dataset.  
  
  
### Question 3  
  
## a.   
  
# Split data into training and test - 80/20   
  
set.seed(1)  
  
rows <- sample(x=nrow(fulldataset), size=0.8\*nrow(fulldataset))  
trainset <- fulldataset[rows, ]  
testset <- fulldataset[-rows, ]  
  
## b.   
  
# LDA  
  
library (MASS)

## Warning: package 'MASS' was built under R version 3.4.3

lda.fit <- lda(final\_quality ~ fixed.acidity+volatile.acidity+citric.acid+residual.sugar+chlorides+free.sulfur.dioxide+total.sulfur.dioxide+density+pH+sulphates+alcohol, data=trainset)  
lda.pred <- predict(lda.fit, testset)  
confusion\_matrix\_lda <- table(testset$final\_quality, lda.pred$class)  
print(confusion\_matrix\_lda)

##   
## 0 1  
## 0 105 27  
## 1 47 141

test\_error\_lda <- sum(lda.pred$class!=testset$final\_quality)/nrow(testset)  
sprintf("The test error for LDA is: %f", test\_error\_lda)

## [1] "The test error for LDA is: 0.231250"

## c.   
  
# QDA  
  
qda.fit <- qda(final\_quality ~ fixed.acidity+volatile.acidity+citric.acid+residual.sugar+chlorides+free.sulfur.dioxide+total.sulfur.dioxide+density+pH+sulphates+alcohol, data=trainset)  
qda.pred <- predict(qda.fit, testset)  
confusion\_matrix\_qda <- table(testset$final\_quality, qda.pred$class)  
print(confusion\_matrix\_qda)

##   
## 0 1  
## 0 90 42  
## 1 36 152

test\_error\_qda <- sum(qda.pred$class!=testset$final\_quality)/nrow(testset)  
sprintf("The test error for QDA is: %f", test\_error\_qda)

## [1] "The test error for QDA is: 0.243750"

**SUMMARY OF SECTION 4.5 – COMPARISON OF CLASSIFICATION METHODS**

Logistic regression, LDA, QDA, and kNN are all different classification methods that each have their strengths and weaknesses – understanding the scenarios in which they are most useful can help us build more accurate models. Logistic regression and LDA both create linear decision boundaries (only difference between the two is their fitting procedures) and therefore when the true decision boundary is linear in form, these methods perform well. kNN, on the other hand, is completely non-parametric (does not assume shape), and hence performs better when the true boundary is highly-nonlinear. The disadvantage with kNN is that we cannot infer anything about the individual predictors and their impact. QDA falls somewhere in between the two: able to fit a wider range of shapes than linear methods but not as flexible as kNN. By making assumptions about the data, it is able to perform better at lower number of training examples than kNN. There are several in between states as well – such as transformations of the predictors – that can be performed to move between these four main types of classification methods. It is important to realize the benefits and shortcomings of each method so that we can apply the correct one when the problem needs it.