Data Mining

G. Holt Williams

6/10/2020

# Introduction

The purpose of this report is to practice data minning techniques in R, including classifying and clustering methods. The project is broken down into two main sections. The first being an exploratory analysis of the data set chosen, then followed by data minning of the dataset

# Data Minning Assignment

## Load Librairies

To begin the analysis the requisite packages need to be loaded into the workspace.

library(tidyverse)  
library(printr)  
library(psych)  
library(ggcorrplot)

## Data Processing

Next the chosen dataset needs to be loaded. For this assignment the UCI Machien Learning Repository’s Wine Quality data set was chosen (<http://archive.ics.uci.edu/ml/datasets/Wine+Quality>). This dataset was chosen for its large sample size and due to the preprocessing UCI applies to limit additional cleaning steps. The code to load the dataset and clean it is as follows:

# load data Set  
red<-read.csv("winequality-red.csv", sep=";")  
white<-read.csv("winequality-white.csv", sep=";")  
## Add type variable  
red$type<-"red"  
white$type<-"white"  
#Combine  
wine<-rbind(red,white)  
wine$type<-as.factor(wine$type)  
table(wine$quality)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 30 | 216 | 2138 | 2836 | 1079 | 193 | 5 |

## Make Quality Factor  
wine$qualityFact<-as.factor(wine$quality)  
wine$qualityFact<-fct\_collapse(wine$qualityFact, Bad = c("3", "4"), Med = c("5","6","7"), Great=c("8","9"))  
table(wine$qualityFact)

|  |  |  |
| --- | --- | --- |
| Bad | Med | Great |
| 246 | 6053 | 198 |

table(wine$quality)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 30 | 216 | 2138 | 2836 | 1079 | 193 | 5 |

The above code adds two variables to the dataset: type and quality as a factor. Type is just if the wine is red or white, while quality as a factor categorizes the quality rankings into 3 sets: Bad, Medium, and Good. This will be useful for the prediction capability later in the assignment.

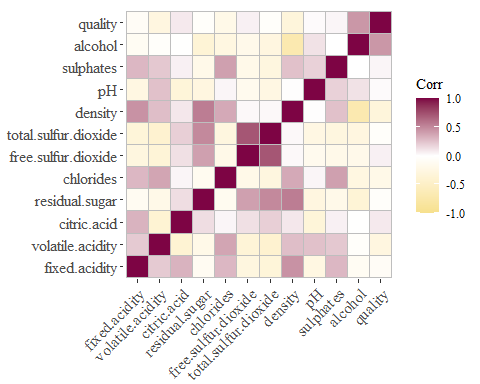
## Tables and Plots

wine%>%describe(quant=c(.25,.75),omit=TRUE)%>%select(n, sd,mean, median, min, max, Q0.25, Q0.75)%>%round(digits=2)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | n | sd | mean | median | min | max | Q0.25 | Q0.75 |
| fixed.acidity | 6497 | 1.30 | 7.22 | 7.00 | 3.80 | 15.90 | 6.40 | 7.70 |
| volatile.acidity | 6497 | 0.16 | 0.34 | 0.29 | 0.08 | 1.58 | 0.23 | 0.40 |
| citric.acid | 6497 | 0.15 | 0.32 | 0.31 | 0.00 | 1.66 | 0.25 | 0.39 |
| residual.sugar | 6497 | 4.76 | 5.44 | 3.00 | 0.60 | 65.80 | 1.80 | 8.10 |
| chlorides | 6497 | 0.04 | 0.06 | 0.05 | 0.01 | 0.61 | 0.04 | 0.06 |
| free.sulfur.dioxide | 6497 | 17.75 | 30.53 | 29.00 | 1.00 | 289.00 | 17.00 | 41.00 |
| total.sulfur.dioxide | 6497 | 56.52 | 115.74 | 118.00 | 6.00 | 440.00 | 77.00 | 156.00 |
| density | 6497 | 0.00 | 0.99 | 0.99 | 0.99 | 1.04 | 0.99 | 1.00 |
| pH | 6497 | 0.16 | 3.22 | 3.21 | 2.72 | 4.01 | 3.11 | 3.32 |
| sulphates | 6497 | 0.15 | 0.53 | 0.51 | 0.22 | 2.00 | 0.43 | 0.60 |
| alcohol | 6497 | 1.19 | 10.49 | 10.30 | 8.00 | 14.90 | 9.50 | 11.30 |
| quality | 6497 | 0.87 | 5.82 | 6.00 | 3.00 | 9.00 | 5.00 | 6.00 |

Each of these variables, except quality, are the inherent characteristics of the wine gatherred through scientific tests. The quality variable is an average of a panel of experts rating the wine. There are a total of 6497 wines in the dataset, broken down into 1599 red wines and 4898 white wines. Below is a plot to show the correlation of each variable.

ggcorrplot(round(cor(wine[1:12]),2), ggtheme=theme\_tufte(),colors = c("#F6E08C", "white", "#7D0444"))

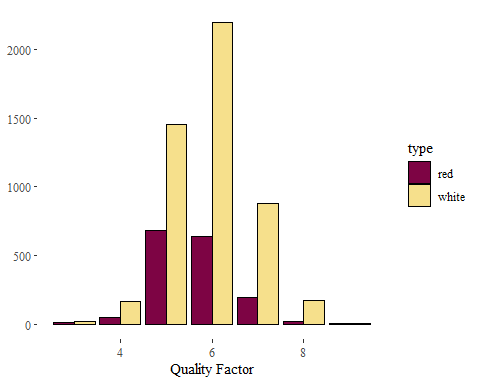


Many of the independant variables here are correlated, but interstingly only alcohol has a higher than average correlation with quality at 0.44. Some other noteworthy points is the interaction of density, residual sugar, and alcohol (content). As alcohol goes down, the residual sugar increases,a dn density increases. This mimics a simple understanding of the wine making process of yeast eating the sugar to produce alcohol and making it less dense. Because this assignment looks partially at prediction by segmentign the data into test and train, correlated variables will not be removed. If instead the outcome was to intrepret the model, it would make sense to remove highly correlated variables.

table(wine$type,wine$qualityFact)

|  |  |  |  |
| --- | --- | --- | --- |
| / | Bad | Med | Great |
| red | 63 | 1518 | 18 |
| white | 183 | 4535 | 180 |

wine%>%ggplot()+  
 geom\_bar(aes(x=quality, fill=type),color="black", position=position\_dodge())+  
 theme\_tufte()+  
 scale\_fill\_manual(values = c("red"="#7D0444","white"="#F6E08C"))+  
 xlab("Quality Factor")+  
 ylab(NULL)



Here we can see the majority of quality ratings are between 5 and 7 under the Medium category, and this holds true for both red and white wines.

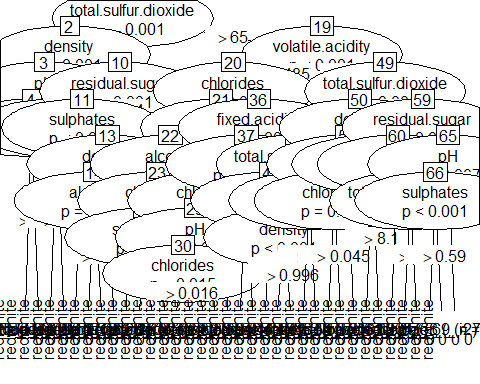
## Data Minning

library(factoextra)  
library(NbClust)  
library(party)  
library(fpc)

### Classification

First data mining technique employed is classification, where the outcome is to try and classify the dataset as either Red or White wine. Both the quality and quality factor variable have been removed as they would be considerred outcome variables, and in testing had no outcome in the predictions. The dataset has been split into both training and testing to practice prediction using decision trees.

set.seed(1234)  
ind <- sample(2, nrow(wine), replace=T, prob=c(0.7, 0.3))  
wine.train <- wine[ind==1,-c(12,14)]  
wine.test <- wine[ind==2, -c(12,14)]  
  
wine.form<- type ~ .  
wine.ctree <- ctree(wine.form,data=wine.train)  
plot(wine.ctree)



Next, the model developed above is used to predict the test set type of wine .

pred<- predict(wine.ctree, newdata=wine.test)  
table(pred, wine.test$type)

|  |  |  |
| --- | --- | --- |
| pred/ | red | white |
| red | 428 | 16 |
| white | 30 | 1444 |

Overall, the model was rather succesful only failing to correctly classify 46 of 1918 total predictions to the correct wine type.

Retrying the analysis to predict the quality factor variable using the data\_split function from tidymodels to stratify the sample across the quality factor variable produces the following prediction results.

library(tidymodels)

## Warning: package 'tidymodels' was built under R version 3.6.3

## -- Attaching packages ---------------------------------------------------------------------------------------------- tidymodels 0.1.0 --

## v dials 0.0.6 v rsample 0.0.6   
## v infer 0.5.1 v tune 0.1.0   
## v parsnip 0.1.1 v workflows 0.1.1   
## v recipes 0.1.10 v yardstick 0.0.6

## Warning: package 'dials' was built under R version 3.6.3

## Warning: package 'infer' was built under R version 3.6.3

## Warning: package 'parsnip' was built under R version 3.6.3

## Warning: package 'recipes' was built under R version 3.6.3

## Warning: package 'rsample' was built under R version 3.6.3

## Warning: package 'tune' was built under R version 3.6.3

## Warning: package 'workflows' was built under R version 3.6.3

## Warning: package 'yardstick' was built under R version 3.6.3

## -- Conflicts ------------------------------------------------------------------------------------------------- tidymodels\_conflicts() --  
## x psych::%+%() masks ggplot2::%+%()  
## x scales::alpha() masks psych::alpha(), ggplot2::alpha()  
## x gridExtra::combine() masks dplyr::combine()  
## x scales::discard() masks purrr::discard()  
## x dplyr::filter() masks stats::filter()  
## x parsnip::fit() masks party::fit(), modeltools::fit()  
## x recipes::fixed() masks stringr::fixed()  
## x dplyr::lag() masks stats::lag()  
## x dials::margin() masks ggplot2::margin()  
## x tune::parameters() masks dials::parameters(), modeltools::parameters()  
## x yardstick::spec() masks readr::spec()  
## x recipes::step() masks stats::step()

set.seed(1234)  
data\_split <- initial\_split(wine[-c(12,13)], strat=qualityFact)  
train\_data <- training(data\_split)  
test\_data <- testing(data\_split)  
wine.formqf<- qualityFact ~ .  
wine.ctreeqf <- ctree(wine.formqf,data=train\_data)  
pred<- predict(wine.ctreeqf, newdata=test\_data)  
table(pred, test\_data$qualityFact)

|  |  |  |  |
| --- | --- | --- | --- |
| pred/ | Bad | Med | Great |
| Bad | 3 | 3 | 0 |
| Med | 57 | 1513 | 44 |
| Great | 0 | 2 | 2 |

Overall, the outcome is not very predictive and suggests the issue is that the majority of the expert’s opinions on the wine is that they are average. The small percentage of Great and Bad reviews is creating difficulties in predicting those values. This may be due to the actual quality being based on interactions of the independant variables, versus the variables themselves.

### Kmeans Clustering

set.seed(1234)  
wine2<- wine[,c(1:11,13)]  
wine2$type<- NULL  
  
wine.kmeans<- kmeans(wine2,2)  
table(wine$type, wine.kmeans$cluster)

|  |  |  |
| --- | --- | --- |
| / | 1 | 2 |
| red | 1514 | 85 |
| white | 1294 | 3604 |

Unlike the classification model, the kmeans model of clustering is not particularly predictive for the type of wine. It groups the majority of the red wines in group one, but it also includes 26% of the white wines in group one. Group two contains 97% white wines in its grouping. Retrying the analysis on both quality and quality factor both produce similar results to the classification model above, with little to no useful groupings of quality.

## Kmeans QualityFact  
wine\_qf<- wine[,c(1:11,14)]  
wine\_qf$qualityFact<- NULL  
wine.kmeansqf<- kmeans(wine\_qf,3)  
table(wine$qualityFact, wine.kmeansqf$cluster)

|  |  |  |  |
| --- | --- | --- | --- |
| / | 1 | 2 | 3 |
| Bad | 85 | 96 | 65 |
| Med | 2716 | 1497 | 1840 |
| Great | 124 | 25 | 49 |

## Kmeans Quality  
wine\_q<- wine[,c(1:12)]  
wine\_q$quality<- NULL  
wine.kmeansq<- kmeans(wine\_q,7)  
table(wine$quality, wine.kmeansq$cluster)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| / | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 3 | 3 | 3 | 2 | 6 | 2 | 4 | 10 |
| 4 | 44 | 35 | 8 | 8 | 23 | 52 | 46 |
| 5 | 324 | 367 | 238 | 172 | 373 | 300 | 364 |
| 6 | 672 | 346 | 360 | 158 | 483 | 361 | 456 |
| 7 | 321 | 84 | 117 | 19 | 246 | 129 | 163 |
| 8 | 63 | 9 | 40 | 6 | 37 | 25 | 13 |
| 9 | 1 | 0 | 0 | 0 | 3 | 1 | 0 |

This continues to suggest that transformations of variables or interaction variables might be useful for this type of prediction.

### Density Based Clustering

This section utilizes a density based clustering model to try and predict the same three variables.

## Density Based Clustering  
library(fpc)  
library(dbscan)

##   
## Attaching package: 'dbscan'

## The following object is masked from 'package:fpc':  
##   
## dbscan

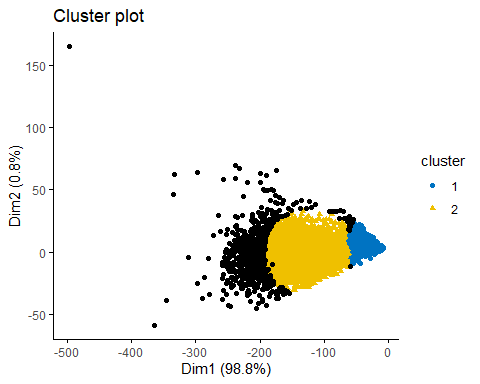
# Group Type  
wine3<- wine[,c(1:12)]  
ds<- dbscan(wine3, eps=20, MinPts = 800)  
table(ds$cluster, wine$type)

|  |  |  |
| --- | --- | --- |
| / | red | white |
| 0 | 37 | 840 |
| 1 | 1140 | 84 |
| 2 | 422 | 3974 |

Using a density based clustering leads to an intersting breakdown of clustering. It clusters the majority of red wines in group one with few white wines, and it groups the majority of white wines in group two. With further optimization it might be possible to increase the clustering saught after.

## Density Based Clustering  
fviz\_cluster(ds, data = wine3, stand = FALSE,  
 ellipse = FALSE, show.clust.cent = FALSE,  
 geom = "point",palette = "jco", ggtheme = theme\_classic())

## Registered S3 methods overwritten by 'car':  
## method from  
## influence.merMod lme4  
## cooks.distance.influence.merMod lme4  
## dfbeta.influence.merMod lme4  
## dfbetas.influence.merMod lme4



Visually this shows all of group one on the right side in blue with group two being just to the left in yellow. This suggests that manipulating where the two groups are touching might produce more optimal results. Quality and Quality Factor are not included as the density clustering did not produce any results that were accurate, even after changing parameters of dbscan to try and optimize the clustering.

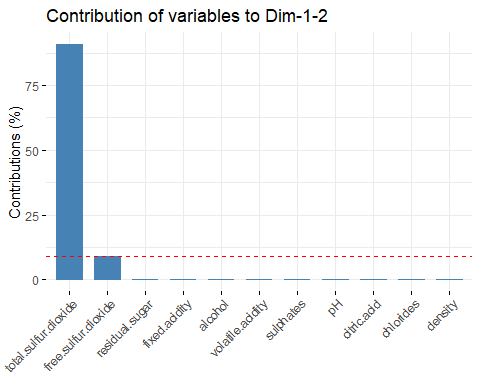
## PCA and Clustering

The poor prediction capabilites of the above sections is most likely due to the interactions of the variables having the most influence on the ratings of the reviewers. To try and rememdy that a Principal Component Analysis will be used along with the decision tree, kmeans, and density clustering data minning techniques. To shorten the length of the final product limited output and discussion will be had unless noteworthy predictions occur. ### PCA Code

set.seed(1234)  
winePCA<-princomp(wine[c(1:11)])  
summary(winePCA)

## Importance of components:  
## Comp.1 Comp.2 Comp.3 Comp.4  
## Standard deviation 58.0653717 11.98421157 4.130503621 1.2805617443  
## Proportion of Variance 0.9537583 0.04062775 0.004826251 0.0004638792  
## Cumulative Proportion 0.9537583 0.99438601 0.999212258 0.9996761371  
## Comp.5 Comp.6 Comp.7 Comp.8  
## Standard deviation 1.0327183086 1.771237e-01 0.1446292672 1.210800e-01  
## Proportion of Variance 0.0003016947 8.874769e-06 0.0000059172 4.147142e-06  
## Cumulative Proportion 0.9999778317 9.999867e-01 0.9999926237 9.999968e-01  
## Comp.9 Comp.10 Comp.11  
## Standard deviation 1.031413e-01 2.786559e-02 7.515958e-04  
## Proportion of Variance 3.009325e-06 2.196547e-07 1.597984e-10  
## Cumulative Proportion 9.999998e-01 1.000000e+00 1.000000e+00

fviz\_contrib(winePCA,choice="var", axes=1:2)



df<-wine  
df$pca1<-winePCA$scores[,1]  
df$pca2<-winePCA$scores[,2]  
df$pca3<-winePCA$scores[,3]

This graph shows that the majority of variance comes from the total.sulfur.dioxide variable.

### PCA Decision Tree

set.seed(1234)  
ind <- sample(2, nrow(df), replace=T, prob=c(0.7, 0.3))  
wine.train\_pca <- df[ind==1,-c(12,14)]  
wine.test\_pca <- df[ind==2, -c(12,14)]  
wine.form\_pca<- type ~ total.sulfur.dioxide  
wine.ctree\_pca <- ctree(wine.form\_pca,data=wine.train\_pca)  
pred\_pca<- predict(wine.ctree\_pca, newdata=wine.test\_pca)  
table(pred\_pca, wine.test\_pca$type)

|  |  |  |
| --- | --- | --- |
| pred\_pca/ | red | white |
| red | 357 | 39 |
| white | 101 | 1421 |

Interrestingly, the prediction just using Total Sulfur Dioxide is really close to the previous decision tree model.

set.seed(1234)  
ind <- sample(2, nrow(df), replace=T, prob=c(0.7, 0.3))  
wine.train\_pca <- df[ind==1,-c(7,12,14,15,16,17)]  
wine.test\_pca <- df[ind==2, -c(7,12,14,15,16,17)]  
wine.form\_pca<- type ~ .   
wine.ctree\_pca <- ctree(wine.form\_pca,data=wine.train\_pca)  
pred\_pca<- predict(wine.ctree\_pca, newdata=wine.test\_pca)  
table(pred\_pca, wine.test\_pca$type)

|  |  |  |
| --- | --- | --- |
| pred\_pca/ | red | white |
| red | 428 | 30 |
| white | 30 | 1430 |

When running the model without this variable the predictions are more accurate than either of the two previous decision tree models

set.seed(1234)  
ind <- sample(2, nrow(df), replace=T, prob=c(0.7, 0.3))  
wine.train\_pca <- df[ind==1,-c(7,13,12,15,16,17)]  
wine.test\_pca <- df[ind==2, -c(7,13,12,15,16,17)]  
wine.form\_pca<- qualityFact ~ .   
wine.ctree\_pca <- ctree(wine.form\_pca,data=wine.train\_pca)  
pred\_pca<- predict(wine.ctree\_pca, newdata=wine.test\_pca)  
table(pred\_pca, wine.test\_pca$qualityFact)

|  |  |  |  |
| --- | --- | --- | --- |
| pred\_pca/ | Bad | Med | Great |
| Bad | 2 | 7 | 0 |
| Med | 62 | 1785 | 62 |
| Great | 0 | 0 | 0 |

Although predictions were better for the type of wine, predicting quality factor is still a challenge.

### PCA Kmeans

set.seed(1234)  
wine\_pca<- df[,c(1:6,8:11,13)]  
wine\_pca$type<- NULL  
wine.kmeans<- kmeans(wine\_pca,2)  
table(wine$type, wine.kmeans$cluster)

|  |  |  |
| --- | --- | --- |
| / | 1 | 2 |
| red | 1489 | 110 |
| white | 2370 | 2528 |

Removing total sulfur dioxide did not produce any more optimal results and suggests the data may have an extra dimension that is not contained in the dataset. Running Kmeans with the same variables produces similar results to before and it is likley that reviewer’s personal taste is highly subjective and its unlikely to be predicted.

## Kmeans QualityFact  
set.seed(1234)  
wine\_qf<- df[,c(1:6,8:11,14)]  
wine\_qf$qualityFact<- NULL  
wine.kmeansqf<- kmeans(wine\_qf,3)  
table(wine$qualityFact, wine.kmeansqf$cluster)

|  |  |  |  |
| --- | --- | --- | --- |
| / | 1 | 2 | 3 |
| Bad | 160 | 60 | 26 |
| Med | 2254 | 2491 | 1308 |
| Great | 45 | 109 | 44 |

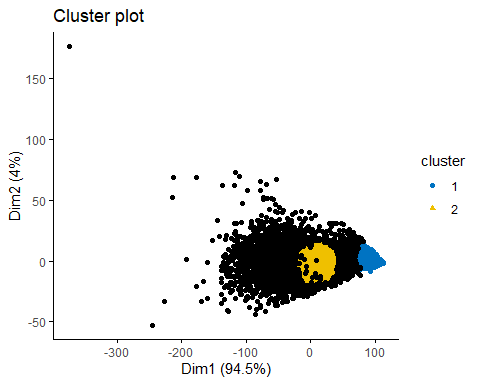
## Kmeans Quality  
wine\_q<- df[,c(1:6,8:11,12)]  
wine\_q$quality<- NULL  
wine.kmeansq<- kmeans(wine\_q,7)  
table(wine$quality, wine.kmeansq$cluster)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| / | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 3 | 12 | 1 | 1 | 6 | 4 | 1 | 5 |
| 4 | 92 | 8 | 38 | 19 | 2 | 11 | 46 |
| 5 | 419 | 361 | 400 | 323 | 5 | 192 | 438 |
| 6 | 371 | 440 | 686 | 594 | 9 | 185 | 551 |
| 7 | 145 | 135 | 282 | 294 | 2 | 43 | 178 |
| 8 | 14 | 33 | 47 | 57 | 4 | 8 | 30 |
| 9 | 0 | 1 | 4 | 0 | 0 | 0 | 0 |

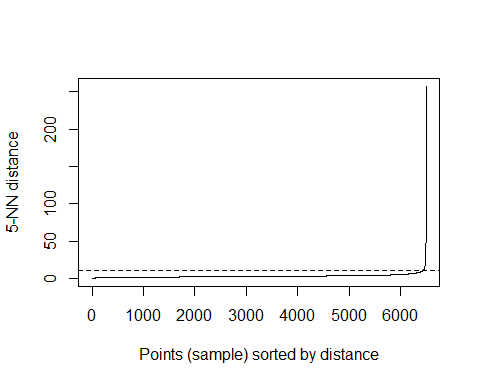
# Group Type  
wine3<- df[,c(12,15:17)]  
ds<- dbscan(wine3, eps=10, MinPts = 320)  
table(ds$cluster, wine$type)

|  |  |  |
| --- | --- | --- |
| / | red | white |
| 0 | 440 | 2852 |
| 1 | 938 | 35 |
| 2 | 221 | 2011 |

fviz\_cluster(ds, data = wine3, stand = FALSE,  
 ellipse = FALSE, show.clust.cent = FALSE,  
 geom = "point",palette = "jco", ggtheme = theme\_classic())



dbscan::kNNdistplot(wine3, k = 5)  
abline(h = 10, lty = 2)



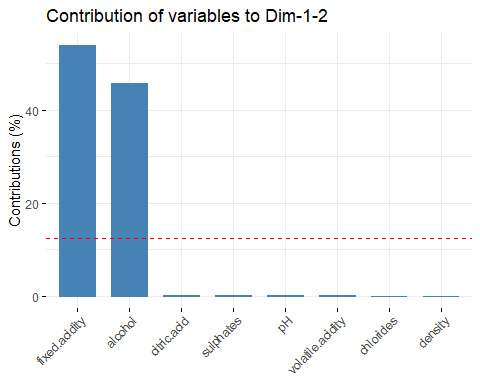
Using the PCA components the density clustering model has simialr outputs to the previous models.

To further test the large influence of total sulfur dioxide, the next section removes it from the PCA code to test if that alters the predictive capability

set.seed(1234)  
winePCA<-princomp(wine[c(1:3,5,8:11)])  
summary(winePCA)

## Importance of components:  
## Comp.1 Comp.2 Comp.3 Comp.4  
## Standard deviation 1.324462 1.1646061 0.2054862 0.145678630  
## Proportion of Variance 0.548142 0.4238112 0.0131941 0.006631411  
## Cumulative Proportion 0.548142 0.9719532 0.9851473 0.991778689  
## Comp.5 Comp.6 Comp.7 Comp.8  
## Standard deviation 0.121504503 0.103579161 0.0285581928 1.674648e-03  
## Proportion of Variance 0.004613165 0.003352425 0.0002548447 8.763177e-07  
## Cumulative Proportion 0.996391854 0.999744279 0.9999991237 1.000000e+00

fviz\_contrib(winePCA,choice="var", axes=1:2)



df2<-wine[c(1:3,5,8:11,13)]  
df2$pca1<-winePCA$scores[,1]  
df2$pca2<-winePCA$scores[,2]  
df2$pca3<-winePCA$scores[,3]  
colnames(df2)

## [1] "fixed.acidity" "volatile.acidity" "citric.acid"   
## [4] "chlorides" "density" "pH"   
## [7] "sulphates" "alcohol" "type"   
## [10] "pca1" "pca2" "pca3"

# Decision tree  
set.seed(1234)  
ind <- sample(2, nrow(df2), replace=T, prob=c(0.7, 0.3))  
wine.train\_pca <- df2[ind==1,-c(1:8)]  
wine.test\_pca <- df2[ind==2, -c(1:8)]  
wine.form\_pca<- type ~ .  
wine.ctree\_pca <- ctree(wine.form\_pca,data=wine.train\_pca)  
pred\_pca<- predict(wine.ctree\_pca, newdata=wine.test\_pca)  
table(pred\_pca, wine.test\_pca$type)

|  |  |  |
| --- | --- | --- |
| pred\_pca/ | red | white |
| red | 421 | 56 |
| white | 37 | 1404 |

# Kmeans  
wine\_pca<- df2[,c(9:12)]  
wine\_pca$type<- NULL  
wine.kmeans<- kmeans(wine\_pca,2)  
table(wine$type, wine.kmeans$cluster)

|  |  |  |
| --- | --- | --- |
| / | 1 | 2 |
| red | 1201 | 398 |
| white | 2747 | 2151 |

# Density  
wine3<- df2[,c(10:12)]  
ds<- dbscan(wine3, eps=0.5, MinPts = 400)

## Warning in dbscan(wine3, eps = 0.5, MinPts = 400): converting argument  
## MinPts (fpc) to minPts (dbscan)!

table(ds$cluster, wine$type)

|  |  |  |
| --- | --- | --- |
| / | red | white |
| 0 | 1012 | 1855 |
| 1 | 587 | 3043 |

Similar results were found as the previous models.

## Conclusion

When using data minning techniques to predict the type of wine, the decision tree and kmeans methods were relatively succesful. This was not the case for the density based clustering used. When trying to predict quality or quality as a factor, every model was unsuccesful at sussing out a pattern for the values on either end of the scale as it would label almsot all values in the 5 to 7 or medium range of quality values. Using PCA to remove variables or using the components it outputs did not improve results.

### References

P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.