

The Wine Vinho Verde

Andresa de Andrade

April 2, 2015

Contents

1	Problem Introduction	2
2	Exploratory Analysis	2
2.1	Sulfur Dioxide	3
2.2	Volatile Acidity	5
2.3	Alcohol	6
3	Correlation and PCA	8
3.1	Correlation Analysis	8
3.2	Principal Component Analysis	8
4	Classification Model	12
4.1	Machine Learning - 10 Cross Folder Validation	14
5	Comparison to other models	17
6	Conclusion	17
7	References	17
8	Appendix	18

1 Problem Introduction

The motivation of this project is to predict the quality of the wine based on the the following variables:

- Fixed Acidity;
- Volatile Acidity;
- Citric Acid;
- Residual Sugar;
- Chlorides;
- Free Sulfur Dioxide;
- Total Sulfur Dioxide
- Density
- pH;
- Sulfates;
- Alcohol;

The quality was based on sensor data. Due privacy and logistic issues there's no further information on how the data was collected or how many people had been used as Evaluators.

In order to predict the quality of the wine here presented, more than one methodology will be tested for learning purposes.

2 Exploratory Analysis

```
#reading data
data<-read.csv("winequality-red.csv", sep=";")

#Treating data names to look preety in our charts/analysis

data_names<-names(data)

simpleCap <- function(x) {

  s <- strsplit(x, " ")[[1]]
  paste(toupper(substring(s, 1,1)), substring(s, 2),
        sep="", collapse=" ")
}

data_names<-as.vector(sapply(data_names, gsub, pattern=".", replacement = " "))
data_names<-as.vector(sapply(data_names, simpleCap))
#####

num_variables = length(data)-1
```

It's interesting to understand if the data set has any outliers or even missing information. Because, if positive, it'd be necessary some treatment prior to the data exploration.

Based on histograms in the appendix section, it's possible to infer that the data set has some outliers affecting the model performance (and it has no missing values). For learning purposes, the data will be used as it's and the influence of outliers will be bring up later on this document.

This is how the output variable behaviors:

```
hist(data$quality, main = "Wine Ratings", xlab= "Rates", freq=F)
```

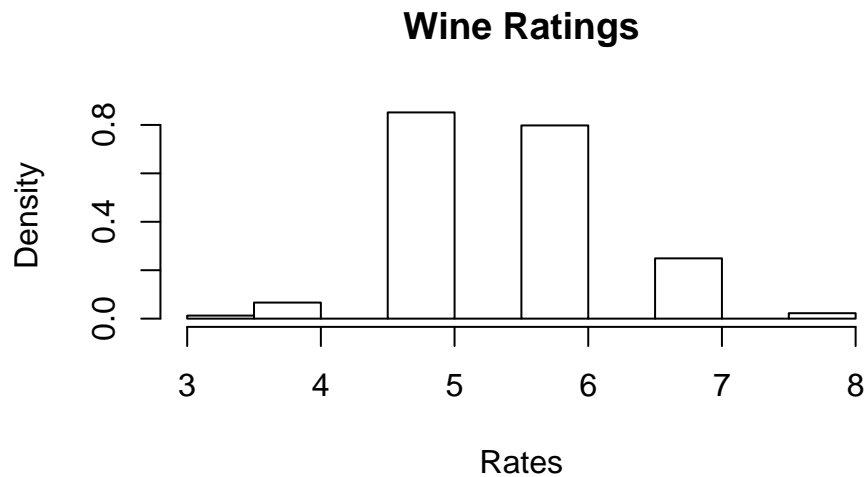


Figure 1: Histogram of Quality of Wine

It's noticeable that the majority of the grades are between 5 and 6 but it's also possible to observe that the rates are slightly skewed to 7 due the heavy tail.

The question is what is relevant for the wine to be highly rated.

To try to answer this question, it's necessary to go back to the (cor)relation between the grades and wine characteristics described in the data set.

2.1 Sulfur Dioxide

Let's start by the Sulfur Dioxide SO_2 . This substance has two main properties within the wine making process. First is to preserve the wine, preventing from oxidation. The second is to bind the acetaldehyde, since this last one has an unpleasant smell of bruised apple or rank sherry and it could affect the final taste of the wine. [1]

In the data set there are two variables dedicated to SO_2 , one is called free sulfur dioxide and is a natural result of fermentation. And the other is Total Sulfur Dioxide being the result of the sum of free SO_2 and the level manually added.

This is a good example of a pair of variable that might be correlated and could be aggregate in a single one.

```
#function that calculate the boxplot by rate for an specific variable using column name as arg
byrate_hist<-function(x, label){
  boxplot_string = ""
  x_label<-array()

  for(grade in 1:10){
    first_graph<-paste("data[data$quality==",grade, ",$",x)
    boxplot_string = paste(boxplot_string, first_graph, ",")
    x_label[grade]<-paste("rate", grade)
  }
}
```

```

first_string <-paste(boxplot_string, "main = '", label, "by Wine Rates'")

lim_sup<-eval(parse(text = paste("max(data$", x, ")*1.05")))
lim_inf<-eval(parse(text = paste("min(data$", x, ")*0.95")))

second_string = paste("boxplot(", first_string, ", ylim=c(", lim_inf, ",", lim_sup, ")", xlim=c(2.5,8.5))

test<-eval(parse(text =second_string))
axis(1, 1:10,x_label)
}

byrate_hist("total.sulfur.dioxide", "Total level of So2")

```

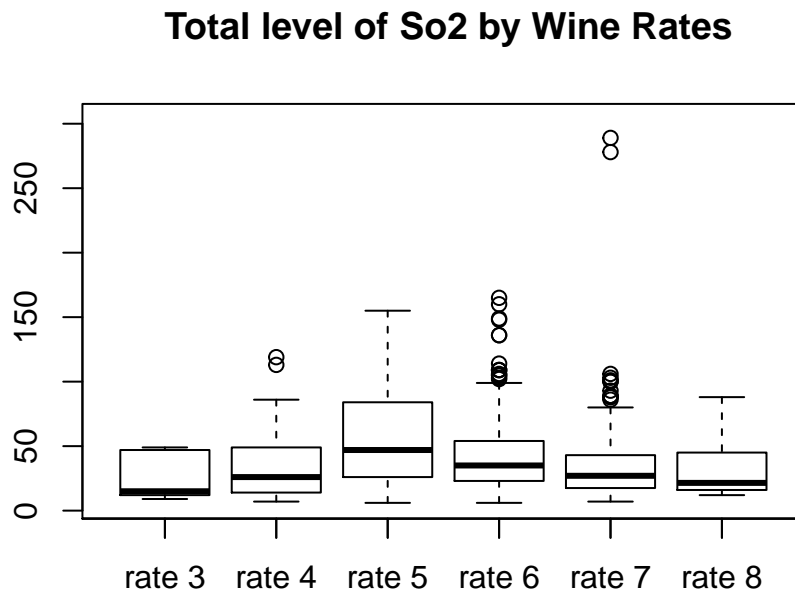


Figure 2: Box plot of Level of Sulfur Dioxide by rating

```

#function that creates the summary table for a specific variable using index as arg
library(knitr)
create_summary_table<-function (x) {
  median_alc<-aggregate(data[,x], list(data$quality), median)
  mean_alc<-aggregate(data[,x], list(data$quality), mean)
  min_alc<-aggregate(data[,x], list(data$quality), min)
  max_alc<-aggregate(data[,x], list(data$quality), max)
  sd_alc<-aggregate(data[,x], list(data$quality), sd)
  summary_table<-data.frame(median_alc, mean_alc[,2], min_alc[,2], max_alc[,2], sd_alc[,2])
  return(summary_table)
}

```

```
summary_table<-create_summary_table(7)
names(summary_table)<-c("Rates", "Mean", "Median", "Min", "Max", "Standard Deviation")
knitr::kable(summary_table, caption = "Total Sulfur Dioxide Summary")
```

Table 1: Total Sulfur Dioxide Summary

Rates	Mean	Median	Min	Max	Standard Deviation
3	15.0	24.90000	9	49	16.82888
4	26.0	36.24528	7	119	27.58337
5	47.0	56.51395	6	155	36.99312
6	35.0	40.86991	6	165	25.03825
7	27.0	35.02010	7	289	33.19121
8	21.5	33.44444	12	88	25.43324

Combining the Figure 2 and the Table 1 we can see that the range level of So_2 between 15 and 48 increases the likelihood of having high rates.

2.2 Volatile Acidity

From the correlation table it's also possible to see that Volatile Acidity is significant for the sensorial analysis of the wine.

```
byrate_hist("volatile.acidity", "Volatile Acidity")
```

Volatile Acidity by Wine Rates

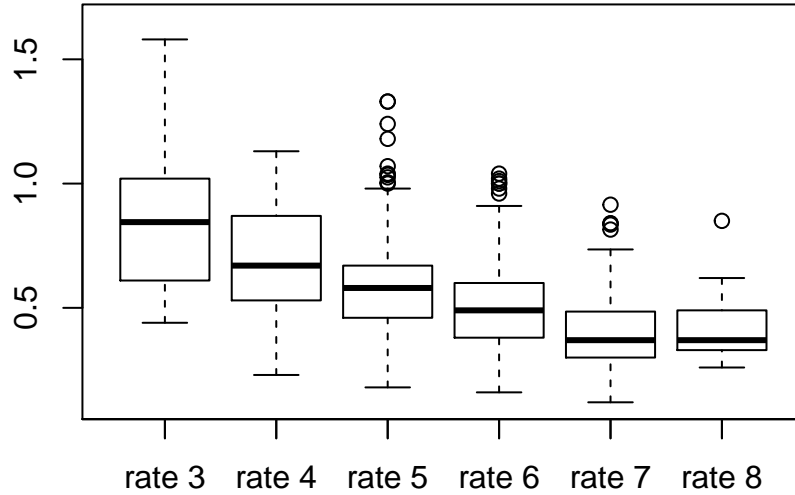


Figure 3: Level of Volatile Acidity

From Figure 2 and Table 2 it's possible to infer that the level of alcohol has a negative correlation to the wine rates, in another words, the lower the vol acidity the higher the rates.

```
summary_table_alc<-create_summary_table(2)
names(summary_table_alc)<-c("Rates", "Mean", "Median", "Min", "Max", "Standard Deviation")
knitr::kable(summary_table_alc, caption = "Vol Acidity Summary")
```

Table 2: Vol Acidity Summary

Rates	Mean	Median	Min	Max	Standard Deviation
3	0.845	0.8845000	0.44	1.580	0.3312556
4	0.670	0.6939623	0.23	1.130	0.2201100
5	0.580	0.5770411	0.18	1.330	0.1648012
6	0.490	0.4974843	0.16	1.040	0.1609623
7	0.370	0.4039196	0.12	0.915	0.1452244
8	0.370	0.4233333	0.26	0.850	0.1449138

2.3 Alcohol

From the appendix section the other relevant substance is alcohol, the graphic and table below describe the distribution of level of alcohol for different rates

```
byrate_hist("alcohol", "Total level of Alcohol")
```

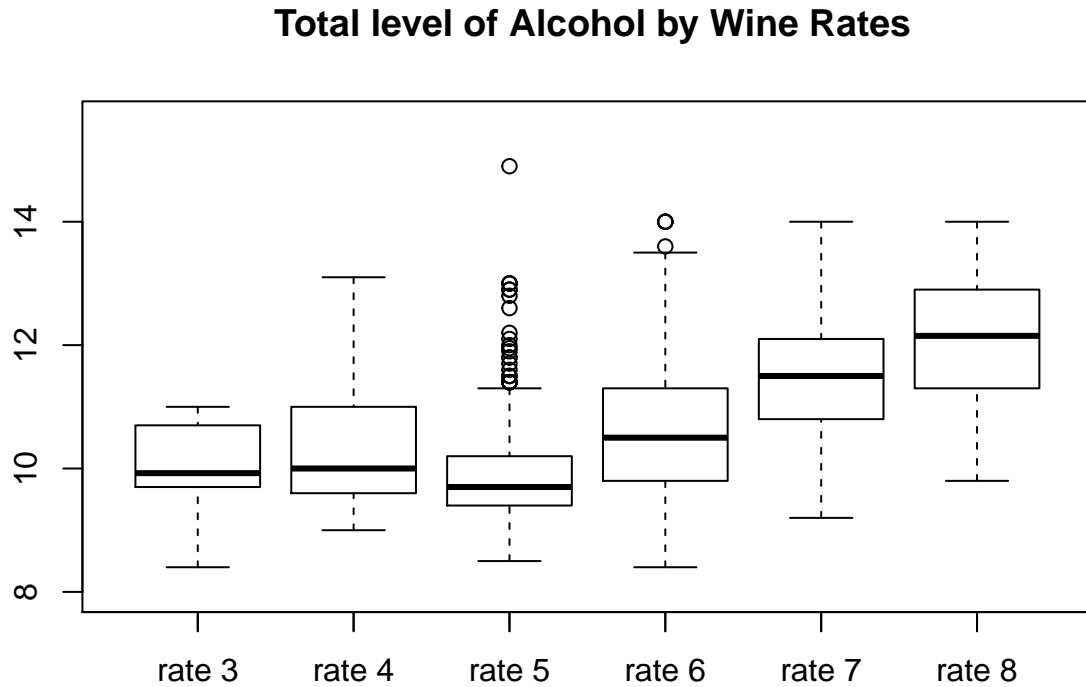


Figure 4: Total Level of alcohol

From Figure 3 and Table 3 it's possible to infer that the level of alcohol has a positive effect in the wine quality.

```
summary_table_alc<-create_summary_table(11)
names(summary_table_alc)<-c("Rates", "Mean", "Median", "Min", "Max", "Standard Deviation")
knitr::kable(summary_table_alc, caption = "Alcohol Summary")
```

Table 3: Alcohol Summary

Rates	Mean	Median	Min	Max	Standard Deviation
3	9.925	9.955000	8.4	11.0	0.8180091
4	10.000	10.265094	9.0	13.1	0.9347760
5	9.700	9.899706	8.5	14.9	0.7365210
6	10.500	10.629519	8.4	14.0	1.0496390
7	11.500	11.465913	9.2	14.0	0.9619330
8	12.150	12.094444	9.8	14.0	1.2240109

3 Correlation and PCA

3.1 Correlation Analysis

The Table 3 helps to confirm that Free SO₂ and Total SO₂ have a significant correlation between each other.

It also confirms that the most important/relevant for high rates is alcohol, and volatile acidity has a negative effect in the wine quality.

At this point, it's important to highlight that having a high correlation doesn't mean having a causality.

```
table_cor<-cor(data)
table_cor<-data.frame(table_cor)
names_cor<-c("Acid", "Vol.Acidity", "Citr.Acidity", "Sugar", "Chlor", "FreeSO2", "TotalSo2", "Den", "pH", "Su", "Alcohol", "Quality")
names(table_cor)<-names_cor
rownames(table_cor)<-names_cor
knitr::kable(table_cor, caption = "Correlation Matrix", digits = 2)
```

Table 4: Correlation Matrix

	Acid	Vol.Acidity	Citr.Acidity	Sugar	Chlor	FreeSO2	TotalSo2	Den	pH	Sulph	Alcohol	Quality
Acid	1.00	-0.26	0.67	0.11	0.09	-0.15	-0.11	0.67	-0.68	0.18	-0.06	0.12
Vol.Acidity	-0.26	1.00	-0.55	0.00	0.06	-0.01	0.08	0.02	0.23	-0.26	-0.20	-0.39
Citr.Acidity	0.67	-0.55	1.00	0.14	0.20	-0.06	0.04	0.36	-0.54	0.31	0.11	0.23
Sugar	0.11	0.00	0.14	1.00	0.06	0.19	0.20	0.36	-0.09	0.01	0.04	0.01
Chlor	0.09	0.06	0.20	0.06	1.00	0.01	0.05	0.20	-0.27	0.37	-0.22	-0.13
FreeSO2	-0.15	-0.01	-0.06	0.19	0.01	1.00	0.67	-0.02	0.07	0.05	-0.07	-0.05
TotalSo2	-0.11	0.08	0.04	0.20	0.05	0.67	1.00	0.07	-0.07	0.04	-0.21	-0.19
Den	0.67	0.02	0.36	0.36	0.20	-0.02	0.07	1.00	-0.34	0.15	-0.50	-0.17
pH	-0.68	0.23	-0.54	-0.09	-0.27	0.07	-0.07	-0.34	1.00	-0.20	0.21	-0.06
Sulph	0.18	-0.26	0.31	0.01	0.37	0.05	0.04	0.15	-0.20	1.00	0.09	0.25
Alcohol	-0.06	-0.20	0.11	0.04	-0.22	-0.07	-0.21	-0.50	0.21	0.09	1.00	0.48
Quality	0.12	-0.39	0.23	0.01	-0.13	-0.05	-0.19	-0.17	-0.06	0.25	0.48	1.00

we can see from the table above some interesting highlights. The first one is that we were right about the correlation between Free SO₂ and Total So₂ variables, hence we might combined them in one single variable.

3.2 Principal Component Analysis

The application of Principal Component only depends on the data covariance (or correlation) matrix, ie, it's not necessary to assume any distribution for the data. Which makes the methodology much easier to use since there's no assumption for the data distribution.[2]

The graphic below shows the main components for the explanation of the variance. The table is built to carry the heaviest proportion of the variance in the first components.

```
pca <- prcomp(data[,1:11])
plot(pca, type = "l", main = "Principal Component Analysis")
```


Principal Component Analysis

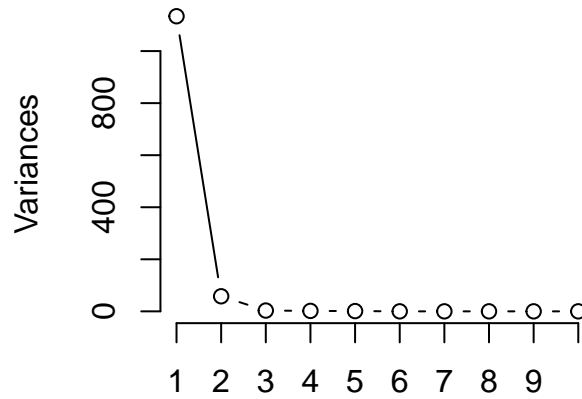


Figure 5: Proportion of Variance Explained by Principal Component

```
#summary(pca)
```

```
summary_pca<-summary(pca)

percent <- function(x, digits = 2, format = "f", ...) {
  paste0(formatC(100 * x, format = format, digits = digits, ...), "%")
}

t1<-percent(summary_pca$importance[2,])
t2<-percent(summary_pca$importance[3,])
variance_pca<-cbind(t1,t2)
colnames(variance_pca)<-c("PVE", "Accumulated PVE")
rnames<-array()
for(i in 1:11){
  rnames[i]<-paste("PC",i, sep = "")
}

rownames(variance_pca)<-rnames
variance_pca<-data.frame(variance_pca)

knitr::kable(variance_pca, caption = "Proportion of Variance Explained by Components")
```

Table 5: Proportion of Variance Explained by Components

	PVE	Accumulated.PVE
PC1	94.66%	94.66%
PC2	4.84%	99.50%
PC3	0.26%	99.75%
PC4	0.15%	99.91%

	PVE	Accumulated.PVE
PC5	0.09%	99.99%
PC6	0.00%	100.00%
PC7	0.00%	100.00%
PC8	0.00%	100.00%
PC9	0.00%	100.00%
PC10	0.00%	100.00%
PC11	0.00%	100.00%

The criteria to evaluate a good model is around 80 being good and greater than 99.5 being excellent, it's important to mention that as the variance, the PVE can variate with the problem. For this project we will consider the first 3 component which consolidates 99.7% of the variance of our variables. The reason that we reduced the number of variables from 11 to 3 is because the data set has a high correlation, therefore it doesn't vary or it has the same variation.

Besides reducing the number of variables for the model, the component also highlights any particular evidence for clusters.

```
eigenvalues<-round(pca$sdev,2)
names<-data_names[1:11]
eigenvalues<-cbind(names, eigenvalues)
eigenvalues<-data.frame(eigenvalues)
names(eigenvalues)<-c("a", "b")
eigenvalues<-eigenvalues[with(eigenvalues, order(b)),]
names(eigenvalues)<-c("Variables", "EigenValues")
knitr::kable(eigenvalues, caption = "Eigenvalues", digits=2)
```

Table 6: Eigenvalues

	Variables	EigenValues
11	Alcohol	0
10	Sulphates	0.04
9	PH	0.1
8	Density	0.11
7	Total Sulfur Dioxide	0.15
6	Free Sulfur Dioxide	0.2
5	Chlorides	1.02
4	Residual Sugar	1.35
3	Citric Acid	1.76
1	Fixed Acidity	33.67
2	Volatile Acidity	7.61

Based on eigenvalues we can observe that Acidity is more mixed within the rates in a way that most of the variability of the model are in these two variables.

Alcohol, Sulfate and Density for example are substances that are important to the final grade, since their variability is very important and they have a positive correlation to the increase of the rates.

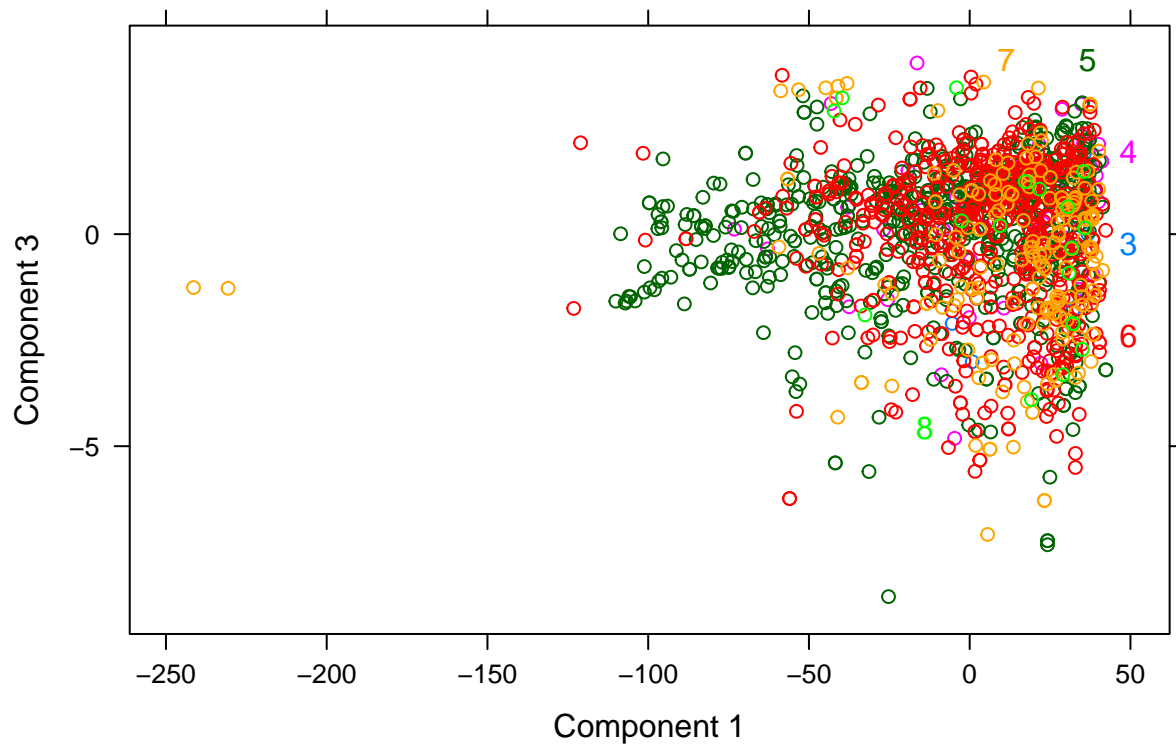
Plotting the first two components we have the following graph:

```
library(lattice)
library(directlabels)
```

```
## Loading required package: grid
## Loading required package: quadprog
```

```
t2<-cbind(pca$x[,1], pca$x[,3], data$quality)
direct.label(xyplot(t2[,2]~t2[,1],groups=t2[,3], ylab="Component 3", xlab = "Component 1", main = "Plot
```

Plot of the first and third Principal Component



We can see that the 5 rates is lower but also more expanded than the others.

4 Classification Model

The classification methodology that we'll be using in this document is the Logistic Regression. The reason we chose this model rather than the other classification methodologies is because this method shares the properties of all other memory-based classification methods - defer most of the processing of the training data point until after a query is made - and has some particular qualities like simplicity, capability of extrapolating and a known confidence interval. [3]

```
data_2<-data
data_2[,12]<-as.factor(data_2[,12])

formula_text <- paste(names(data_2[12]),"~",paste(names(data[1:11]), collapse="+"))
formula_raw <- as.formula(formula_text)
fit <- glm(formula_raw,data=data_2,family=binomial())
summary(fit)
```

```
##
## Call:
## glm(formula = formula_raw, family = binomial(), data = data_2)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.96787    0.00752    0.02341    0.05730    1.22446
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    494.89510   523.99352   0.944 0.344931
## fixed.acidity    -0.28791    0.67009  -0.430 0.667446
## volatile.acidity  -8.40765    2.50988  -3.350 0.000809 ***
## citric.acid      -3.70698    3.92708  -0.944 0.345195
## residual.sugar     0.14205    0.29387   0.483 0.628827
## chlorides       -13.03262    7.00680  -1.860 0.062886 .
## free.sulfur.dioxide -0.15367    0.08888  -1.729 0.083823 .
## total.sulfur.dioxide  0.09925    0.04981   1.992 0.046322 *
## density        -470.45027   533.58716  -0.882 0.377953
## pH              -8.01302    4.80305  -1.668 0.095253 .
## sulphates        2.69403    3.47425   0.775 0.438088
## alcohol          1.32310    0.77934   1.698 0.089563 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 121.428  on 1598  degrees of freedom
## Residual deviance:  69.165  on 1587  degrees of freedom
## AIC: 93.165
##
## Number of Fisher Scoring iterations: 10
```

The table above shows the scores for all variables. Based on the last column, it's possible to choose the relevant coefficients (with p-value < 0.10). So for the final model we have:

- volatility acidity;

- chlorides;
- free sulfur dioxide;
- pH;
- alcohol

Alcohol is the only compound that as higher the level the better. All the others affect the wine flavor/quality negatively. This way it's possible to predict the rate based on the coefficients above.

Now applying the same methodology to the PCA data set we have the following output:

```
data_pca<-cbind(pca$x[,1], pca$x[,2], pca$x[,3], data$quality)
data_pca<-data.frame(data_pca)
names(data_pca)<-c("comp1", "comp2", "comp3", "quality")
data_pca[,4]<-as.factor(data_pca[,4])

formula_text <- paste(names(data_pca[4]), "~", paste(names(data_pca[1:3]), collapse="+"))
formula <- as.formula(formula_text)
fit <- glm(formula, data=data_pca, family=binomial())
summary(fit)
```

```
##
## Call:
## glm(formula = formula, family = binomial(), data = data_pca)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.4683   0.0580   0.1033   0.1417   0.1990
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   5.76102    0.64052   8.994  <2e-16 ***
## comp1        -0.04768    0.02530  -1.885   0.0594 .
## comp2        -0.05441    0.07312  -0.744   0.4568
## comp3         0.01174    0.17916   0.066   0.9477
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 121.43  on 1598  degrees of freedom
## Residual deviance: 114.29  on 1595  degrees of freedom
## AIC: 122.29
##
## Number of Fisher Scoring iterations: 9
```

So Based on the table above only the the first component is relevant for the model, and as higher the value, the worse is the rating.

The only problem of having the pca model is the machine learning cost, because the algorithm would have to run the PCA and then using this “new data” we would have the logistic model. When we are dealing with huge data sets this becomes expensive.

4.1 Machine Learning - 10 Cross Folder Validation

Now we need to run the cross validation for 10 folders. Based on the data below we can see that our model is very accurate. We have one loop with 3 prediction error over 67%

But still very accurate.

```
library("caret")
```

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

```
set.seed(1)
folds <- createFolds(data$quality)
i=1
for (f in folds){
  train <- data[-f,]
  test <- data[f,]
  fit <- lm(formula_raw, data=train)
  test$scores <- predict(fit, type="response",
                        newdata=test)
  pred<-abs((round(test$scores)-test$quality)/test$quality)

  pred_table<-table(round(pred,2))
  print(paste("Prediction Table for validation", i))
  print(pred_table)

  #confusion matrix using 90% accuracy
  c<-table(factor(round(test$scores),levels=3:8), test$quality)
  print(paste("Confusion Matrix for validation", i))
  print(c)
  i=i+1
}
```

```
## [1] "Prediction Table for validation 1"
##
##      0 0.14 0.17  0.2 0.25 0.33  0.4  0.5
## 101  14  22  14   5   1   1   1
## [1] "Confusion Matrix for validation 1"
##
##      3  4  5  6  7  8
## 3  0  0  0  0  0  0
## 4  1  0  0  0  0  0
## 5  0  3 54 21  0  0
## 6  0  1 14 41 14  2
## 7  0  0  1  1  6  0
## 8  0  0  0  0  0  0
## [1] "Prediction Table for validation 2"
##
##      0 0.14 0.17  0.2 0.25  0.5
## 98  17  17  20   5   4
## [1] "Confusion Matrix for validation 2"
##
##      4  5  6  7  8
```

```

## 3 0 0 0 0 0
## 4 0 0 0 0 0
## 5 4 47 15 0 0
## 6 4 20 47 17 1
## 7 0 0 2 4 0
## 8 0 0 0 0 0
## [1] "Prediction Table for validation 3"
##
## 0 0.12 0.14 0.17 0.2 0.25 0.29 1
## 104 1 13 19 15 4 2 1
## [1] "Confusion Matrix for validation 3"
##
## 3 4 5 6 7 8
## 3 0 0 0 0 0 0
## 4 0 0 0 0 0 0
## 5 0 3 55 14 2 0
## 6 1 0 15 45 13 1
## 7 0 0 0 5 4 1
## 8 0 0 0 0 0 0
## [1] "Prediction Table for validation 4"
##
## 0 0.12 0.14 0.17 0.2 0.25 0.4 0.5
## 91 1 16 19 24 6 1 1
## [1] "Confusion Matrix for validation 4"
##
## 4 5 6 7 8
## 3 0 0 0 0 0
## 4 0 1 0 0 0
## 5 5 43 16 0 0
## 6 1 23 45 16 1
## 7 0 1 3 3 1
## 8 0 0 0 0 0
## [1] "Prediction Table for validation 5"
##
## 0 0.12 0.14 0.17 0.2 0.25 0.5
## 95 1 12 25 24 2 2
## [1] "Confusion Matrix for validation 5"
##
## 4 5 6 7 8
## 3 0 0 0 0 0
## 4 0 1 0 0 0
## 5 2 47 23 0 0
## 6 2 23 39 12 0
## 7 0 0 2 9 1
## 8 0 0 0 0 0
## [1] "Prediction Table for validation 6"
##
## 0 0.12 0.14 0.17 0.2 0.25 0.29 0.4 0.5 0.6
## 87 1 10 30 23 4 2 1 1 1
## [1] "Confusion Matrix for validation 6"
##
## 4 5 6 7 8
## 3 0 0 0 0 0
## 4 0 0 0 0 0

```

```

## 5 2 46 24 2 0
## 6 1 23 34 10 2
## 7 0 1 6 7 1
## 8 0 1 0 0 0
## [1] "Prediction Table for validation 7"
##
## 0 0.12 0.14 0.17 0.2 0.25 0.5 0.67
## 96 1 19 19 16 7 1 2
## [1] "Confusion Matrix for validation 7"
##
## 3 4 5 6 7 8
## 3 0 0 0 0 0 0
## 4 0 1 0 0 0 0
## 5 2 6 49 18 0 0
## 6 0 1 16 45 19 1
## 7 0 0 0 1 1 1
## 8 0 0 0 0 0 0
## [1] "Prediction Table for validation 8"
##
## 0 0.12 0.14 0.17 0.2 0.25 0.29 0.5 0.67
## 94 1 16 23 18 5 1 1 1
## [1] "Confusion Matrix for validation 8"
##
## 3 4 5 6 7 8
## 3 0 0 0 0 0 0
## 4 0 0 0 0 0 0
## 5 1 5 49 19 1 0
## 6 0 1 18 41 16 0
## 7 0 0 0 4 4 1
## 8 0 0 0 0 0 0
## [1] "Prediction Table for validation 9"
##
## 0 0.12 0.14 0.17 0.2 0.25 0.29 0.5 0.67
## 93 1 14 18 24 3 1 4 3
## [1] "Confusion Matrix for validation 9"
##
## 3 4 5 6 7 8
## 3 0 0 0 0 0 0
## 4 0 0 0 0 0 0
## 5 3 1 43 15 1 0
## 6 0 4 24 46 14 2
## 7 0 0 0 3 4 1
## 8 0 0 0 0 0 0
## [1] "Prediction Table for validation 10"
##
## 0 0.14 0.17 0.2 0.25 0.29 0.5 0.67
## 93 19 18 18 4 1 3 2
## [1] "Confusion Matrix for validation 10"
##
## 3 4 5 6 7 8
## 3 0 0 0 0 0 0
## 4 0 0 1 0 0 0
## 5 2 3 48 18 1 0
## 6 0 3 17 45 19 1

```



```
## 7 0 0 0 0 0 0
## 8 0 0 0 0 0 0
```

5 Comparison to other models

One of the final goals of this project is to compare with other models, in this case.

Comparing the methodology and the results in this document to the reference it's possible to see very similar results considering 90% accuracy since the methodology is the same. These are the main similarities and discrepancies from the model proposed here and the author methodology:

- in the reference the authors apply different fitting criteria in order to consider the data well fitted (p. 25). In this project it's showed only one criteria which is 90
- the authors applied 20 runs of a 5 cross folder validation, while in this project we have only 1 run for 10 cross folder validation. This could be affect the results since it's not as confused as it should be.
- the authors ignored grades 3 and 9 since they were very rare within the data set. This was a very good approached since the rates 9 and 3 hurt the model results being to hard to predict.

6 Conclusion

Based on the methodology applied above we can infer that the relevant compounds to predict the wine rates are:

- volatility acidity having a negative effect;
- chlorides having a negative effect;
- free sulfur dioxide having a negative effect;
- pH having a negative effect;
- alcohol having a positive effect

The Logistic Model predicts with an error lower than 25% 95% of the times that we ran the algorithm.

The rate is most common to be misplaced is 5 because the variability of the compounds is too high to have a distinct group.

The computational cost for the algorithm is relatively low making possible the automation and deployment into a quality assurance process.

7 References

References

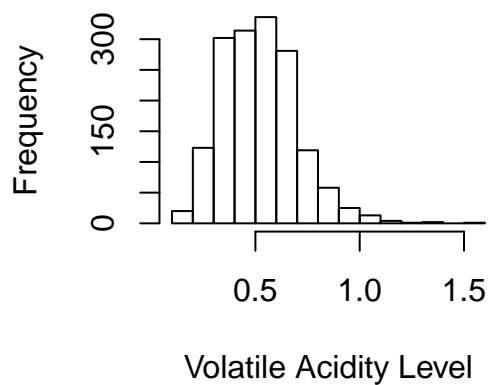
- [1] Gawel Richard (Unknown period). Retrieved from http://www.aromadictionary.com/articles/sulfurdioxide_article.html
- [2] Johnson and Wichern. (pag. 459 - 462) Retrieved from *Applied Multivariate Statistical Analysis*

- [3] Deng Kan. (1999. April 09) Retrieved from <https://www.cs.cmu.edu/~kdeng/thesis/logistic.pdf>
- [4] Pandell J Alexander. (1999) Retrieved from http://www.wineperspective.com/the_acidity_of_wine.htm
- [5] Paulo Cortez, Antonio Cerdeira and Fernando Almeida. (Unknown) Retrieved from <http://repositorium.sdum.uminho.pt/bitstream/1822/10029/1/wine5.pdf>

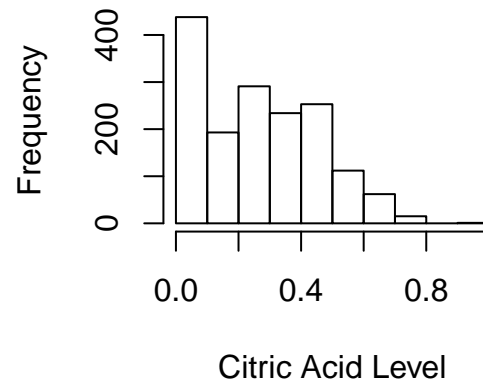
8 Appendix

Histogram of all Variables

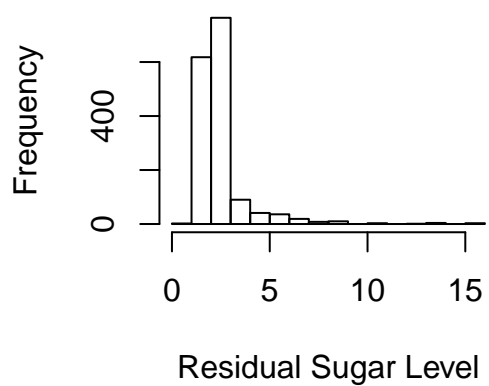
Volatile Acidity Histogram



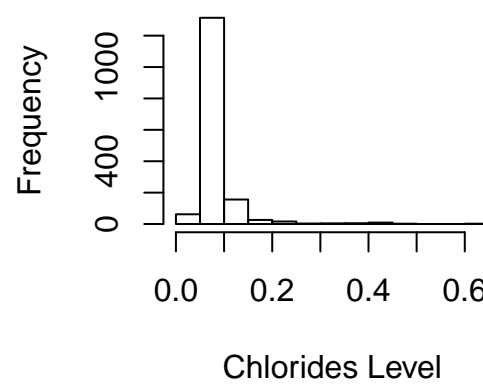
Citric Acid Histogram



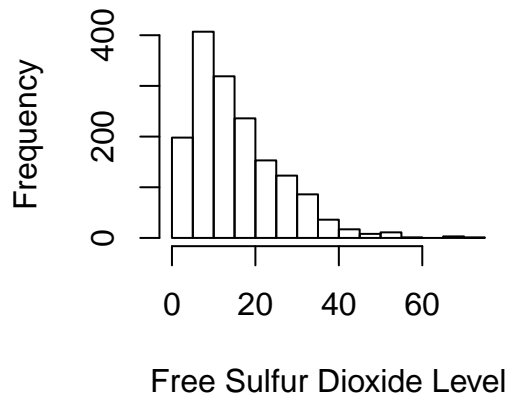
Residual Sugar Histogram



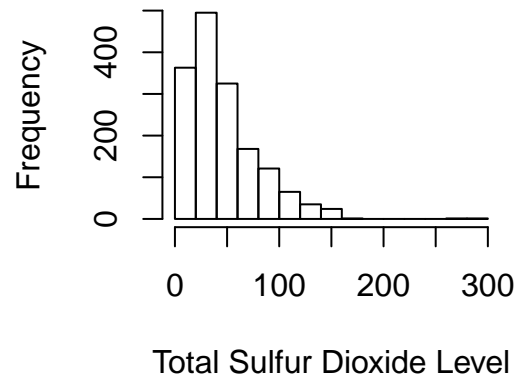
Chlorides Histogram



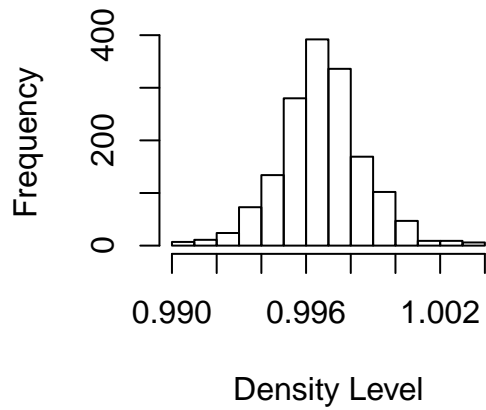
Free Sulfur Dioxide Histogram



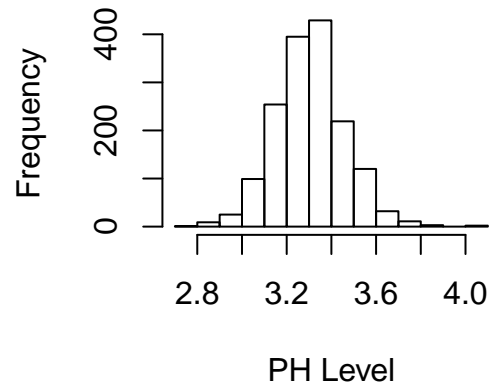
Total Sulfur Dioxide Histogram



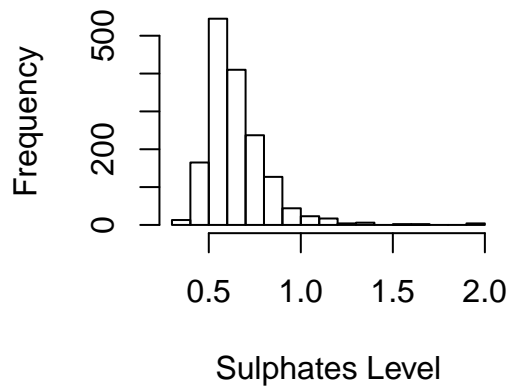
Density Histogram



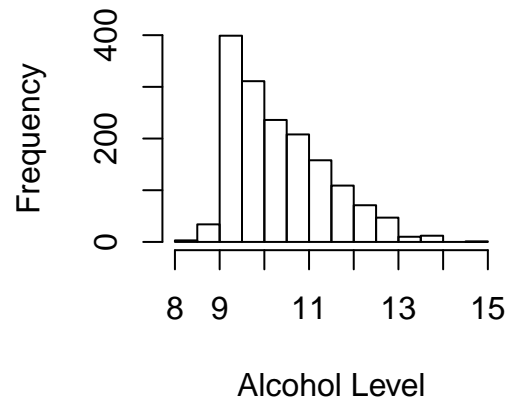
PH Histogram



Sulphates Histogram



Alcohol Histogram



Quality Histogram

