UNIVERSITÀ DEGLI STUDI DI CATANIA

DIPARTIMENTO DI ECONOMIA E IMPRESA

CORSO DI LAUREA IN DATA SCIENCE FOR MANAGEMENT

Carla Perrone

Statistical learning report:

white wine quality dataset analysis

Prof. S. Ingrassia

ANNO ACCADEMICO 2020- 2021

1 Introduction

This report is based on the **wine quality data set** available at the link:
`https://archive.ics.uci.edu/ml/datasets/Wine+Quality`.

Two datasets are included related to red and white variants of the Portuguese "Vinho Verde" wine. For more details, consult the reference [1]. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.).

In this report we focus our attention on the **white wine data set**.
In the original data set there are seven values for the **target variable "quality"**, integer numbers ranging from 3 to 9. The target variable has been reclassified in five values: the values 3 and 4 have been put together in class "4", the values 8 and 9 have been put together in class "8".

The **aim** of this report is to fit a model in order to predict if a wine is "good" or not, in other words in order to predict human wine taste preferences.
Actually, wine certification is generally assessed by both physicochemical and sensory tests. But, as Cortez et al. [1] observe "Physicochemical laboratory tests routinely used to characterize wine, while sensory tests rely mainly on human experts. It should be stressed that taste is the least understood of the human senses thus wine classification is a difficult task. Moreover, the relationships between the physicochemical and sensory analysis are complex and still not fully understood. Statistical models have made it possible to collect, store and process massive, often highly complex datasets. All this data hold valuable information such as trends and patterns". This implies that a statistical model can be useful to support oenologists and improve wine production.

Some questions need to be answered:

- is there a relationship beetween quality and physiochemical properties?
- What are the physiochemical properties more associated with the output variable and so that are useful in predicting the wine quality?
- What is the best model, in terms of accuracy, we can use to predict the wine quality?

First of all, let's see more in detail the structure of the dataset and check if there are or not missing values.

The **input variables** are:

- fixed acidity (g/dm^3): nonvolatile acids (this means that they do not evaporate readily);
- volatile acidity (g/dm^3): acidic elements of a wine that are gaseous;
- citric acid (g/dm^3): an acid that can add 'freshness' and flavor to wines;
- residual sugar (g/dm^3): the amount of sugar remaining after fermentation stops;
- chlorides (g/dm^3): the amount of salt in the wine;
- free sulfur dioxide (mg/dm^3): the free form of SO_2. It prevents microbial growth and the oxidation of wine;
- total sulfure dioxide (mg/dm^3): amount of free and bound forms of SO_2;
- density (g/cm^3): density of the wine;

- ph: describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic);
- sulphates (g/dm^3): a wine additive which can contribute to sulfur dioxide gas (SO_2) levels, which acts as an antimicrobial and antioxidant;
- alcohol (% by volume): the percent alcohol content of the wine.

They are all **numerical and continuos**. Actually in the structure of the dataset there is also another variable, the first one, that we have not included in the previous list. Indeed it refears to the index of the wine, therefore, it is not useful for the analysis and we remove it from the dataset.

The **output variable** is **"quality"** which is, instead, **numerical and discrete**.

There aren't missing values.

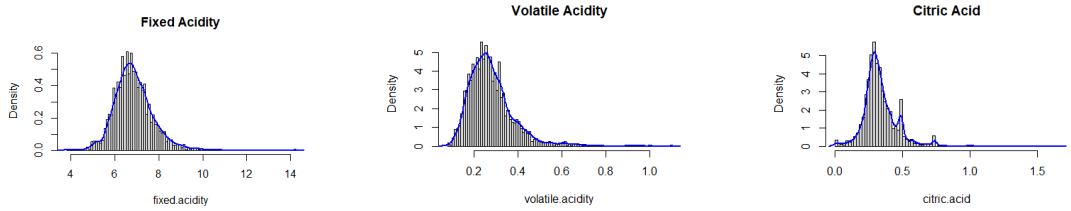
To achieve the aim of this report, the data set has been divided into three datasets: the train data (about 60% of the units of the original dataset), the validation data (about 20% of the units of the original dataset) and the test data (about 20% of the units of the original dataset).
After a preliminary univariate analysis, the dataset has been splitted into two groups: a wine with a value of quality less or equal than 5 has been classified as "bad wine" (low quality), on the contrary a wine with a value of quality more than 5 has been classified as "good wine" (high quality). The relationship between each physiochemical property and quality, low or high, has been investigated. Then, three different models have been fitted using the training dataset: a logistic regression model, a classification trees model and a neural networks model. Then, after a comparison between the three models, the best one has been chosen and tested on the validation dataset.

2 Descriptive analysis

2.1 Univariate analysis

The training dataset contains 2937 observations of the variables we have described in the Introduction 1.

We start doing a univariate analysis to discover the features of each variable and to check if there are outliers. Figures 1 shows the histogram and the density line of each input variable.



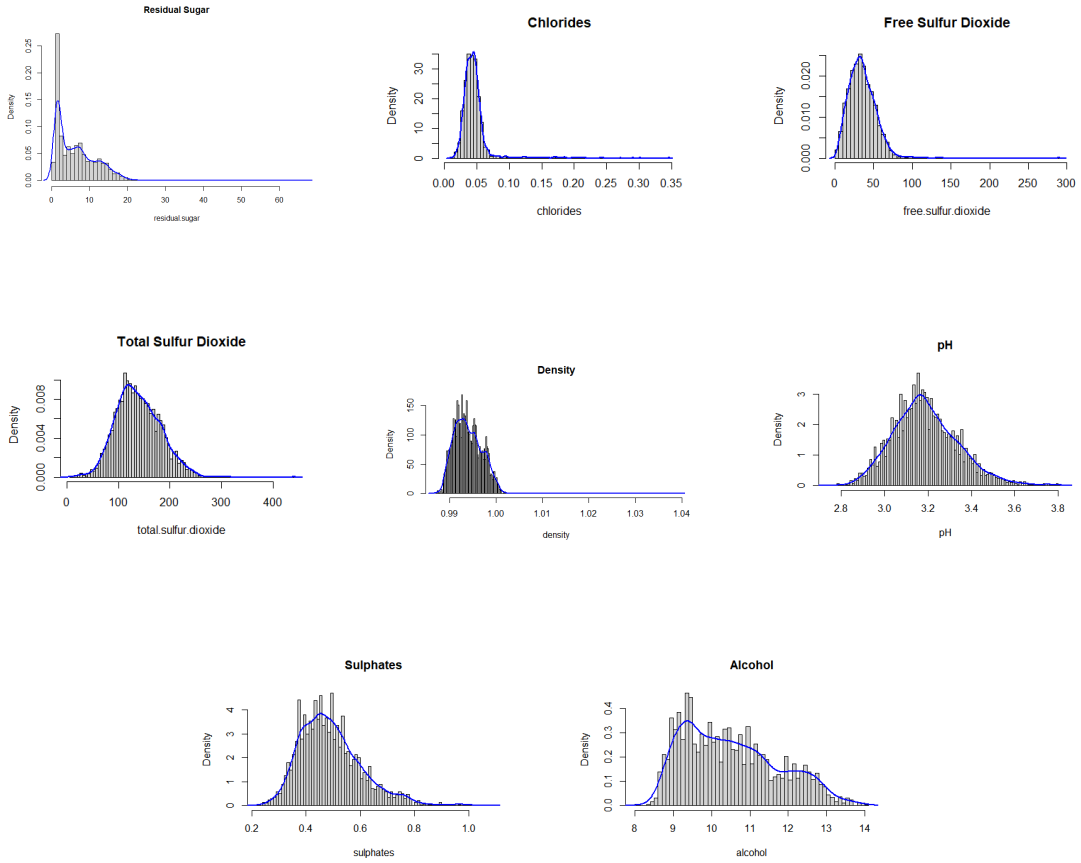


Figure 1: Histogram and density line of each input variable

Summary statistics are presented in Table 1 (more details about the meaning of these summary statistics are given in the Appendix, paragraph 6.1):

Variable	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	Range	Sd	Cv	Skewness
Fixed Acidity	3.80	6.30	6.80	6.86	7.30	14.20	10.40	0.85	12.44	0.72
Volatile Acidity	0.08	0.21	0.26	0.28	0.32	1.10	1.02	0.10	36.71	1.70
Citric Acid	0	0.27	0.31	0.33	0.39	1.66	1.66	0.12	36.20	1.48
Residual Sugar	0.60	1.70	5.10	6.32	9.70	65.80	65.20	5.09	80.50	1.29
Chlorides	0.0090	0.036	0.0430	0.0459	0.0500	0.3460	0.34	0.023	50.24	5.30
Free Sulfur Dioxide	2.00	23.00	34.00	35.42	46.00	289.00	287	17.24	48.69	1.73
Total Sulfur Dioxide	9.0	109.0	135.0	138.6	167.0	440.0	431	42.77	30.85	0.38
Density	0.9871	0.9917	0.9937	0.9940	0.9960	1.0390	0.05187	0.003	0.304	1.41
Ph	2.74	3.10	3.18	3.19	3.28	3.82	1.08	0.15	4.73	0.50
Sulphates	0.22	0.41	0.48	0.49	0.55	1.08	0.86	0.11	23.28	0.93
Alcohol	8.00	9.50	10.40	10.53	11.40	14.05	6.05	1.23	11.68	0.50

Table 1: Summary statistics of the input variables

Looking at the previous table and at the distributions of the input variables, we can deduce the main features of each one:

- the distribution of fixed acidity has a tail on the right side so it is right skewed, but the skewness, which gives a measure of symmetry, is low because its value is 0.72, this means

that the asimmetry is weak. Third quartile, 7.3, shows that 75% of the wines have a fixed acidity value in range $[3.8, 7.3]\, g/dm^3$. However, the remaining 25% of the wines have a fixed acidity value in range $(7.3, 14.2]\, g/dm^3$.

- The distribution of volatile acidity has a tail on the right side so it is right skewed and the skewness value is 1.70. Third quartile, 0.32, shows that 75% of the wines have a volatile acidity value in range $[0.08, 0.32]\, g/dm^3$. However, the remaining 25% of the wines have a volatile acidity value in range $(0.32, 1.1]\, g/dm^3$.

- The distribution of citric acid has a tail on the right side so it is right skewed and the skewness value is 1.48. Third quartile, 0.39, shows that 75% of the wines have a citric acid value in range $[0, 0.39]\, g/dm^3$. However, the remaining 25% of the wines have a citric acid value in range $(0.39, 1.66]\, g/dm^3$. Furthermore we can see that the distribution of citric acid has three peaks that suggest three main classes, that is low, medium and high quality.

- The distribution of residual sugar has a tail on the right side so it is right skewed and the skewness value is 1.29. Third quartile, 9.7, shows that 75% of the wines have a residual sugar value in range $[0.6, 9.7]\, g/dm^3$. However, the remaining 25% of the wines have a chlorides value in range $(9.7, 65.8]\, g/dm^3$. As in the previous plot, we can see that the distribution of residual sugar has three peaks that suggest three main classes, that is low, medium and high quality.

- The distribution of chlorides has a tail on the right side so it is right skewed; furthermore the skewness is high because its value is 5.30, this means that the asimmetry is strong. Third quartile, 0.05, shows that 75% of the wines have a volatile acidity value in range $[0.009, 0.05]\, g/dm^3$. However, the remaining 25% of the wines have a chlorides value in range $(0.05, 0.3460]\, g/dm^3$.

- The distribution of free sulfur dioxide has a tail on the right side so it is right skewed and the skewness value is 1.73. Third quartile, 46, shows that 75% of the wines have a free sulfur dioxide value in range $[2, 46]\, mg/dm^3$. However, the remaining 25% of the wines have a free sulfur dioxide value in range $(46, 289]\, mg/dm^3$.

- The distribution of total sulfur dioxide has a tail on the right side so it is right skewed but the skewness is low because its value is 0.38, this means that the asimmetry is weak. Third quartile, 167.0, shows that 75% of the wines have a total sulfur dioxide value in range $[9, 167]\, mg/dm^3$. However, the remaining 25% of the wines have a total sulfur dioxide value in range $(167, 440]\, mg/dm^3$. We can see that the distribution of total sulfur dioxide has three peaks that suggest three main classes, that is low, medium and high quality.

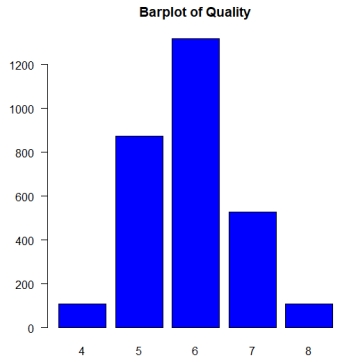
- The distribution of density has a tail on the right side so it is right skewed and the skewness value is 1.41. Third quartile, 0.9960, shows that 75% of the wines have a density value in range $[0.9871, 0.9960]\, g/cm^3$. However, the remaining 25% of the wines have a density value in range $(0.9960, 1.0390]\, g/cm^3$. We can see that the distribution of density has three peaks that suggest three main classes, that is low, medium and high quality.

- The distribution of ph has a tail on the right side so it is right skewed but the skewness is low because its value is 0.50, this means that the asimmetry is weak. Third quartile, 3.280, shows that 75% of the wines have a ph value in range $[2.740, 3.280]$. However, the remaining 25% of the wines have a ph value in range $(3.280, 3.820]$.

- The distribution of sulphates has a tail on the right side so it is right skewed but the skewness is low because its value is 0.93, this means that the asimmetry is weak. Third quartile, 0.55, shows that 75% of the wines have a sulphates value in range $[0.22, 0.55]\, g/dm^3$. However, the remaining 25% of the wines have a sulphates value in range $(0.55, 1.08]\, g/dm^3$.

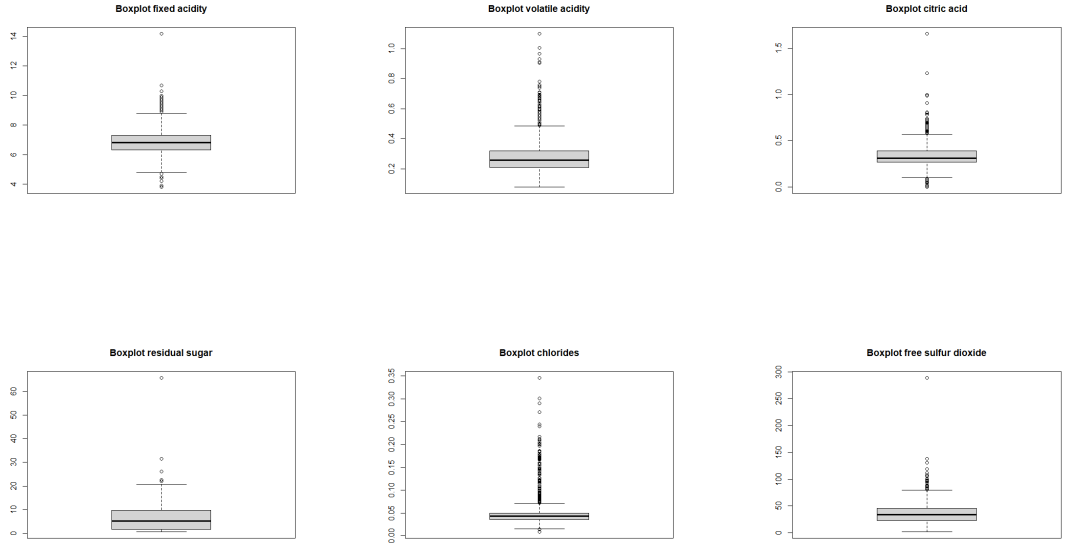
- The distribution of alcohol is asymmetric, precisely it is right skewed but the skewness is low because its value is 0.50, this means that the asimmetry is weak. Third quartile, 11.40, shows that 75% of the wines have an alcohol value in range $[8, 11.40]$. However, the remaining 25% of the wines have an alcohol value in range $(11.40, 14.05]$. We can see that the distribution of alcohol has three peaks that could be interpreted as three main classes, that is low, medium and high quality.

Finally we have a look at the barplot of quality, which is the output variable, and to its summary statistics.



Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
4.000	5.000	6.000	5.882	6.000	8.000

Third quartile, 6, shows that 75% of the wines have a value quality equal to 4, 5, or 6. However, the remaining 25% of the wines have a quality value equal to 7 or 8. We can also see that the most number of wines have a quality equal to 6.
Now for each input variable we check if there are outliers or not. Figure 2 shows the boxplot of each input variable.



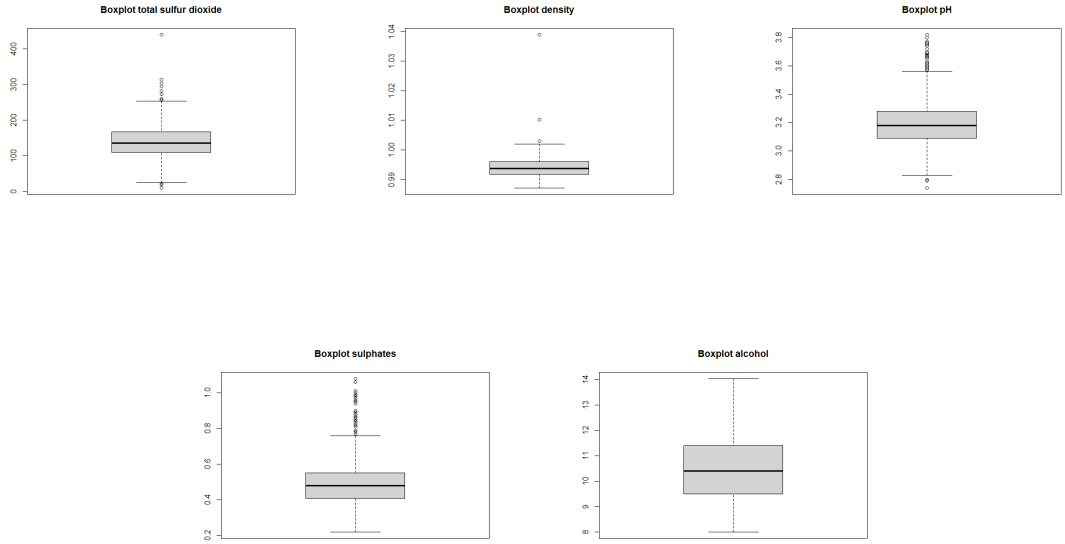
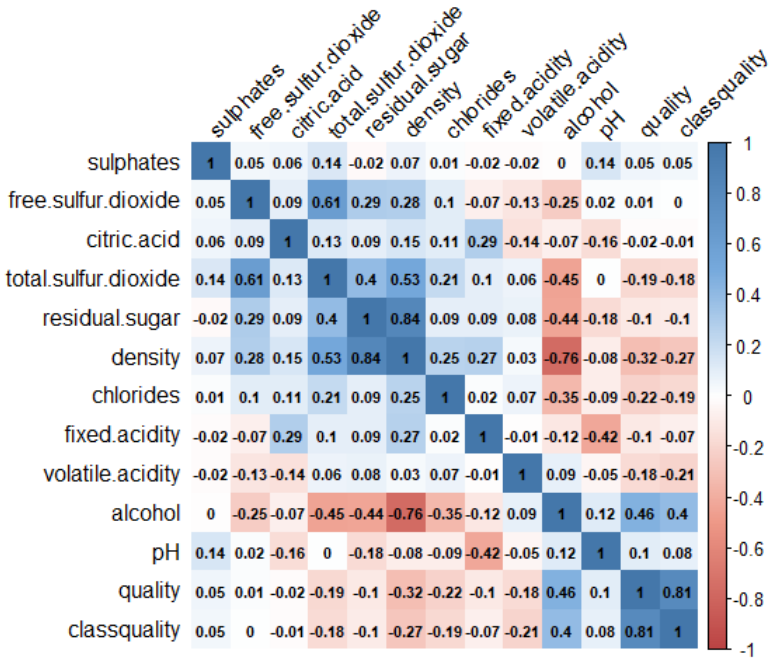


Figure 2: Boxplot of each input variable

- Looking at the boxplots we can see that all variables, except alcohol, contains outliers.
- Fixed acidity, volatile acidity, citric acid, chlorides, free sulfur dioxide, pH and sulphates have a lot of outliers.
- On the contrary, residual sugar, total sulfur dioxide and density have a small number of outliers.

2.2 Bivariate analysis

We transform, as we said in the introduction, the quality variable into a binary one dividing the observations into two classes: we assign an observation to the class number 0 if the quality of that observation is 4 or 5; on the contrary, we assign it to the class with number 1 if its quality goes from 6 to 8. We call this new variable "class quality". Let's see the correlation matrix:



	sulphates	free.sulfur.dioxide	citric.acid	total.sulfur.dioxide	residual.sugar	density	chlorides	fixed.acidity	volatile.acidity	alcohol	pH	quality	classquality
sulphates	1	0.05	0.06	0.14	-0.02	0.07	0.01	-0.02	-0.02	0	0.14	0.05	0.05
free.sulfur.dioxide	0.05	1	0.09	0.61	0.29	0.28	0.1	-0.07	-0.13	-0.25	0.02	0.01	0
citric.acid	0.06	0.09	1	0.13	0.09	0.15	0.11	0.29	-0.14	-0.07	-0.16	-0.02	-0.01
total.sulfur.dioxide	0.14	0.61	0.13	1	0.4	0.53	0.21	0.1	0.06	-0.45	0	-0.19	-0.18
residual.sugar	-0.02	0.29	0.09	0.4	1	0.84	0.09	0.09	0.08	-0.44	-0.18	-0.1	-0.1
density	0.07	0.28	0.15	0.53	0.84	1	0.25	0.27	0.03	-0.76	-0.08	-0.32	-0.27
chlorides	0.01	0.1	0.11	0.21	0.09	0.25	1	0.02	0.07	-0.35	-0.09	-0.22	-0.19
fixed.acidity	-0.02	-0.07	0.29	0.1	0.09	0.27	0.02	1	-0.01	-0.12	-0.42	-0.1	-0.07
volatile.acidity	-0.02	-0.13	-0.14	0.06	0.08	0.03	0.07	-0.01	1	0.09	-0.05	-0.18	-0.21
alcohol	0	-0.25	-0.07	-0.45	-0.44	-0.76	-0.35	-0.12	0.09	1	0.12	0.46	0.4
pH	0.14	0.02	-0.16	0	-0.18	-0.08	-0.09	-0.42	-0.05	0.12	1	0.1	0.08
quality	0.05	0.01	-0.02	-0.19	-0.1	-0.32	-0.22	-0.1	-0.18	0.46	0.1	1	0.81
classquality	0.05	0	-0.01	-0.18	-0.1	-0.27	-0.19	-0.07	-0.21	0.4	0.08	0.81	1

We are interesting in the correlation between class quality and all the others variables (except quality of course). From the matrix above we can deduce that:

- There is no linear correlation between class quality and free sulfur dioxide.
- There is a positive linear correlation between class quality and the variables pH, sulphates, alcohol. To be precise, the linear correlation with pH and sulphates is very small and the correlation with alcohol is higher. Positive linear correlation between two variables means that an increase (or decrease) of one of them is associated with an increase (or decrease) of the other one.

  If we wanted to interpret these results, we could say that we expected a correlation between class quality and alcohol because "alcohol content affects a wine's body. Precisely, a wine with higher alcohol content will have a fuller, richer body, while a lower alcohol wine will taste lighter and more delicate on the palate"[1].
- There is a negative linear correlation between class quality and the variables fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, total sulfur dioxide and density. To be precise, the linear correlation with citric acid, fixed acidity and residual sugar is very small and the correlation with volatile acidity, chlorides, total sulfur dioxide and density is a little bit higher. Negative linear correlation between two variables means that an increase

[1] https://www.masterclass.com/articles/learn-about-alcohol-content-in-wine-highest-to-lowest-abv-wines#3-ways-alcohol-content-affects-the-taste-of-wine

(or decrease) of one of them is associated with a decrease (or increase) of the other one.

If we wanted to interpret these results, we could say that we expected a negative (linear) correlation between class quality and volatile acidity because high values of volatile acidity lead to an unpleasant, vinegar taste.
We expected also a correlation between classquality and residual sugar because the latter determines the sweetness of wine.
Finally, we could imagine that there was a correlation between classquality and total sulfur dioxide because, for its antioxidant effect, it prevents the alteration of aromas and of taste wine.

Let's see the conditional distributions on values of the variable "class quality" trying to characterize the distributions of high quality wine and low quality wine. Figure 3 and Figure 4 show, for each input variable, and for each interval in which we have suddivided the values of that input variable, the proportion of wine with high quality (light blue color) and with low quality (red color).

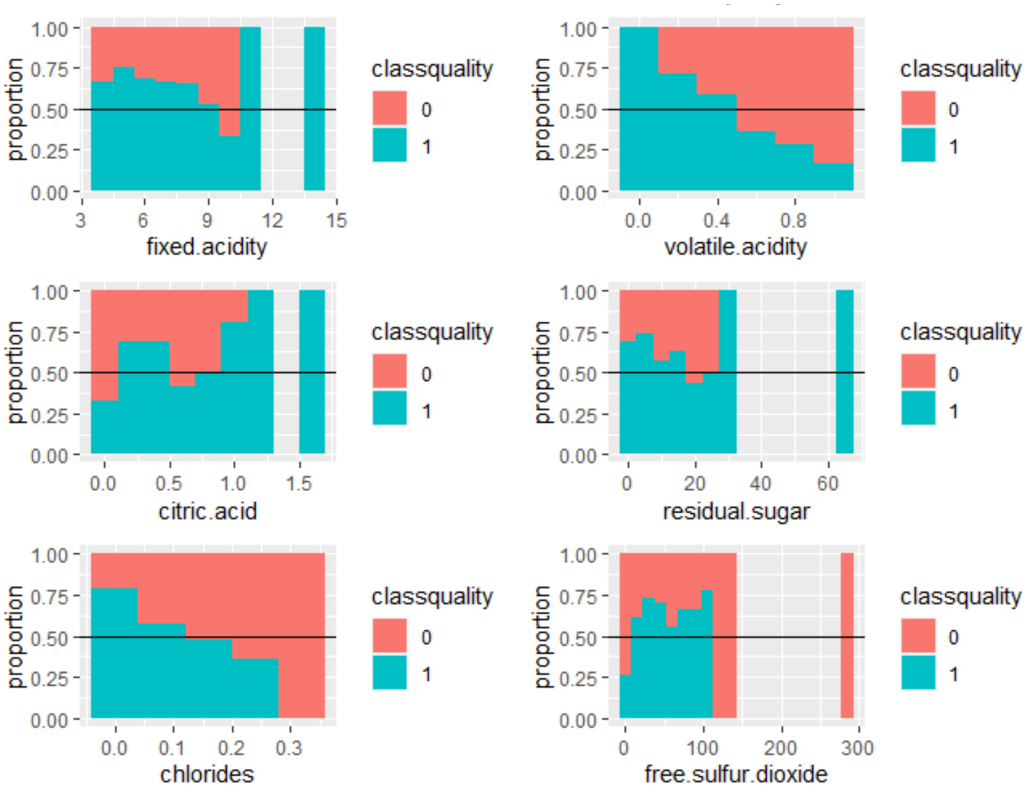


Figure 3: Conditional distributions of the first 6 input variables on values of classquality. For each input variable, the conditional distribution gives the proportion of high and low quality wine with respect to each interval in which the values of the input variable are suddivided.

High quality wine is statistically characterized as follows:

- fixed acidity less than 9.5. We do not take into account the last two bins because in these bins there are few data (see Appendix, paragraph 6.2).
- Volatile acidity between 0.1 and 0.5. We do not take into account the first bin because in this bins there are few data (see Appendix, paragraph 6.2).

- Citric acid between 0.1 and 0.5. We do not take into account the last three bins because in these bins there are few data (see Appendix, paragraph 6.2).
- Residual sugar less than 17.5. We do not take into account the last two bins because in these bins there are few data (see Appendix, paragraph 6.2).
- Chlorides less than 0.12.
- Free sulfure dioxide between 7.5 and 112.

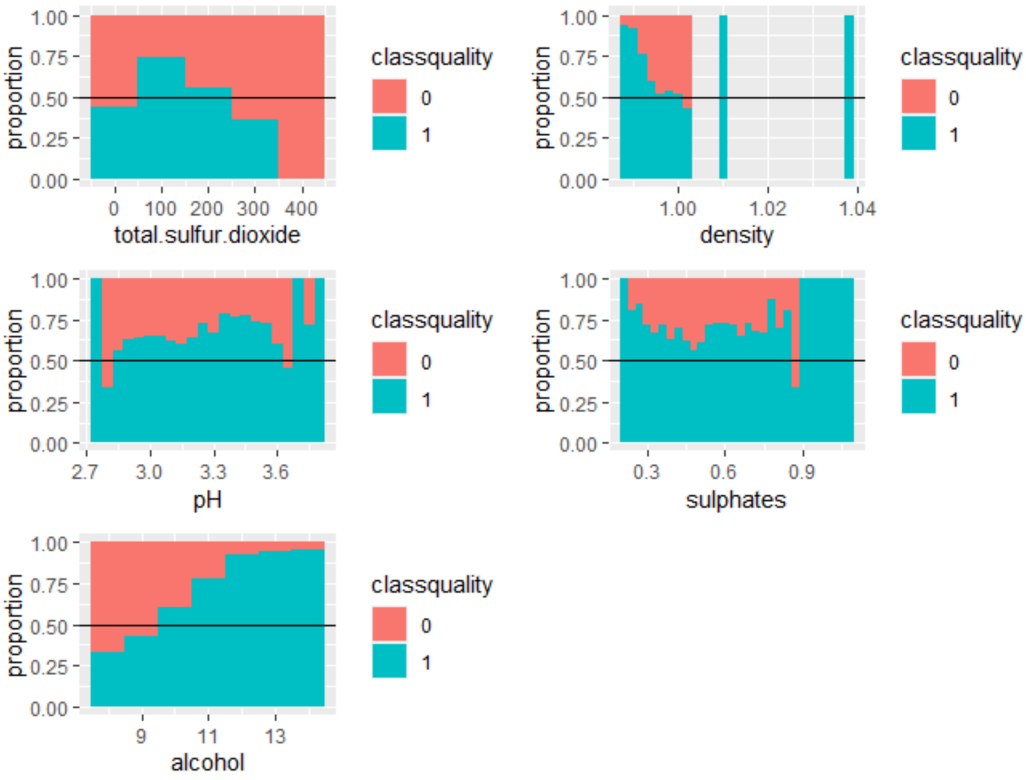


Figure 4: Conditional distributions of the first 6 input variables on values of classquality. For each input variable, the conditional distribution gives the proportion of high and low quality wine with respect to each interval in which the values of the input variable are suddivided.

- total sulfure dioxide between 50 and 250,
- density less than 0.997. We do not take into account the last two bins because in these bins there are few data (see Appendix, paragraph 6.2),
- ph values between 2.83 and 3.62. We do not take into account the first bin and the last bins, because in these bins there are few data (see Appendix, paragraph 6.2).
- sulphates between 0.0225 and 0.855. We do not take into account the first bin and the last bins, starting from about 0.9, because in these bins there are few data (see Appendix, paragraph 6.2).
- alcohol greater than 9.5.

3 Model of the train data

3.1 First approach: logistic regression

We start fitting a multiple logistic regression model to the training dataset. The task is to obtain a model involving only the variables that are associated with the response, which in this case study is the one we have called "class quality". This task is called "variable selection". Ideally, we would like to perform variable selection by trying out a lot of different models, each containing a different subset of predictors. In practice they would be too many and this implies that we have to use a different method: the "stepwise regression". It is a method of fitting regression models in which the choice of predictive variables is carried out by an automatic procedure. In each step, a variable is considered for addition to or subtraction from the set of explanatory variables based on some prespecified criterion. Here we use the Akaike Information Criterion (AIC). According to this criterion, the best model is the one with the smallest AIC.
The "stepwise regression" includes three different approaches: the forward selection, the backward selection and the mixed selection, which is a combination of the two previous ones. Here we choose the mixed selection. We start with a model including all the variables as predictors. Then, at each step, the procedure considers the statistical consequences of dropping variables that are included in the current model and also of adding those variables that were previously dropped. So, a variable might be dropped in one of the steps, then added again in another step and so on. The process goes on until neither adding nor dropping variables the model improves.

Let's see the first step of the algorithm:

```
> logiReg = step(glm(classquality~.-quality, data = data_train1, family = binomial),
+
+             direction = "both") # we are using all the variables except quality
Start:  AIC=2966.23
classquality ~ (fixed.acidity + volatile.acidity + citric.acid +
    residual.sugar + chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
    density + pH + sulphates + alcohol + quality) - quality

                      Df Deviance    AIC
- chlorides            1   2942.2 2964.2
- citric.acid          1   2942.5 2964.5
- fixed.acidity        1   2943.4 2965.4
- total.sulfur.dioxide 1   2943.6 2965.6
<none>                     2942.2 2966.2
- pH                   1   2947.4 2969.4
- free.sulfur.dioxide  1   2951.5 2973.5
- density              1   2952.6 2974.6
- sulphates            1   2955.1 2977.1
- residual.sugar       1   2971.4 2993.4
- alcohol              1   2983.9 3005.9
- volatile.acidity     1   3094.9 3116.9

Step:  AIC=2964.23
```

The initial model is the one containing all the variables as predictors and it has an AIC equal to 2966.23.
Looking at the first row we can say that if we removed the variable chlorides as a predictor and we refitted the model, the AIC would become 2964.2 which is a bit lower than the start AIC, the one related to the model with all predictors, whose value is 2966.23. As the variables are ordered in ascending order according to the value of the corresponding AIC, we can conclude that we have the highest decrease in AIC if we drop the variable chlorides. Nevertheless, this improvement is not so much because the two values of AIC are very similar to each other. This means that, in practice, the two models are equivalent in terms of AIC but the new one has a lower number of variables and this means that it is more parsimonious. Therefore we can conclude that the model

without chlorides as a predictor is better than the full model.
Now the algorithm restarts from the model that does not contain the variables chlorides and the result of the second step is the following:

```
classquality ~ fixed.acidity + volatile.acidity + citric.acid +
    residual.sugar + free.sulfur.dioxide + total.sulfur.dioxide +
    density + pH + sulphates + alcohol

                      Df Deviance    AIC
- citric.acid          1   2942.5 2962.5
- fixed.acidity        1   2943.5 2963.5
- total.sulfur.dioxide 1   2943.6 2963.6
<none>                     2942.2 2964.2
+ chlorides            1   2942.2 2966.2
- pH                   1   2947.5 2967.5
- free.sulfur.dioxide  1   2951.6 2971.6
- density              1   2952.9 2972.9
- sulphates            1   2955.1 2975.1
- residual.sugar       1   2972.7 2992.7
- alcohol              1   2984.0 3004.0
- volatile.acidity     1   3096.0 3116.0

Step:  AIC=2962.49
```

Reasoning as we did for the first step, we conclude that if we drop the variable citric acid the model becomes better in terms of AIC because the AIC decreases from 2964.2 to 2962.5. Despite this, we notice, as we did before, that the decrease of AIC is not so much. This means that, in practice, according to AIC, the two models are equivalent but the new one has a lower number of variables and this means that it is more parsimonious. Therefore we can conclude that the model without both chlorides and citric acid as predictors is the best. Now the algorithm restarts from the model that does not contain the variables chlorides and citric acid and the result of the third step is the following:

```
classquality ~ fixed.acidity + volatile.acidity + residual.sugar +
    free.sulfur.dioxide + total.sulfur.dioxide + density + pH +
    sulphates + alcohol

                      Df Deviance    AIC
- fixed.acidity        1   2943.6 2961.6
- total.sulfur.dioxide 1   2943.8 2961.8
<none>                     2942.5 2962.5
+ citric.acid          1   2942.2 2964.2
+ chlorides            1   2942.5 2964.5
- pH                   1   2948.2 2966.2
- free.sulfur.dioxide  1   2951.7 2969.7
- density              1   2953.6 2971.6
- sulphates            1   2955.2 2973.2
- residual.sugar       1   2973.6 2991.6
- alcohol              1   2984.0 3002.0
- volatile.acidity     1   3097.6 3115.6

Step:  AIC=2961.64
```

We remark two aspects: firstly we observe that if we added to the current model again the variable citric acid or the variable chlorides, we would not have an improvement on the current model because its AIC (2962.49) is a bit lower than the one related both to citric acid and chlorides (respectively AIC= 2964.2 and AIC=2964.5) and in addition the current model is more parsimonius. Secondly, as we did before, we look at the first row, that now is related to the variable fixed acidity and we conclude that if we drop this variable from the current model, the AIC decreases from 2962.5 to 2961.6. This means that, in practice, according to AIC, the two

models are equivalent but the new one has a lower number of variables and this means that it is more parsimonious. Therefore we can conclude that the model without chlorides, citric acid and also fixed acidity as predictors is the best.
Now the algorithm restarts from the model that does not contain the variables chlorides, citric acid and fixed acidity and the result of the third step is the following:

```
classquality ~ volatile.acidity + residual.sugar + free.sulfur.dioxide +
    total.sulfur.dioxide + density + pH + sulphates + alcohol

                       Df Deviance    AIC
- total.sulfur.dioxide  1   2945.1 2961.1
<none>                      2943.6 2961.6
+ fixed.acidity         1   2942.5 2962.5
+ citric.acid           1   2943.5 2963.5
+ chlorides             1   2943.6 2963.6
- pH                    1   2948.7 2964.7
- free.sulfur.dioxide   1   2952.6 2968.6
- sulphates             1   2955.5 2971.5
- density               1   2956.4 2972.4
- residual.sugar        1   2983.9 2999.9
- alcohol               1   3051.7 3067.7
- volatile.acidity      1   3106.8 3122.8

Step:  AIC=2961.14
```

We can see that removing the variable total sulfur dioxide the AIC decreases from 2961.6 to 2961.1. This means that, in practice, according to AIC, the two models are equivalent but the new one has a lower number of variables and this means that it is more parsimonious. Therefore we can conclude that the model without chlorides, citric acid fixed acidity and also total sulfur dioxide as predictors is the best. So the algorithm restarts from the model that does not contain the variables chlorides, citric acid, fixed acidity and total sulfur dioxide and the result of the third step is the following:

```
classquality ~ volatile.acidity + residual.sugar + free.sulfur.dioxide +
    density + pH + sulphates + alcohol

                       Df Deviance    AIC
<none>                      2945.1 2961.1
+ total.sulfur.dioxide  1   2943.6 2961.6
+ fixed.acidity         1   2943.8 2961.8
+ citric.acid           1   2945.0 2963.0
+ chlorides             1   2945.1 2963.1
- pH                    1   2950.4 2964.4
- free.sulfur.dioxide   1   2953.2 2967.2
- sulphates             1   2956.2 2970.2
- density               1   2961.1 2975.1
- residual.sugar        1   2989.3 3003.3
- alcohol               1   3051.9 3065.9
- volatile.acidity      1   3121.5 3135.5
```

We observe that adding to the current model one of the variables we have removed in the previous steps (see the first four rows above) or removing one of the other variables (see the fifth row and the following ones) we always obtain a model that has an AIC greater than the one related to the current one.
This means that be have obtained the final model: it contains all the variables as predictors except total sulfur dioxide, fixed acidity, citric acid and chlorides.

The following table shows the coefficient estimates and related information of the fitted model. We recall that we are using the predictors volatile acidity, residual sugar, free sulfur dioxide,

density, pH, sulphates and alcohol in order to predict the probability that a wine has a class quality equal to 1 (high quality wine).

```
> summary(logiReg)

Call:
glm(formula = classquality ~ volatile.acidity + residual.sugar +
    free.sulfur.dioxide + density + pH + sulphates + alcohol,
    family = binomial, data = data_train1)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.1910  -0.8858   0.4334   0.7976   2.5244

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)          1.919e+02  5.489e+01   3.496 0.000471 ***
volatile.acidity    -6.324e+00  5.116e-01 -12.363  < 2e-16 ***
residual.sugar       1.458e-01  2.289e-02   6.368 1.91e-10 ***
free.sulfur.dioxide  8.067e-03  2.896e-03   2.786 0.005337 **
density             -2.041e+02  5.500e+01  -3.710 0.000207 ***
pH                   7.472e-01  3.282e-01   2.276 0.022820 *
sulphates            1.473e+00  4.494e-01   3.278 0.001046 **
alcohol              8.839e-01  8.498e-02  10.400  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3744.4  on 2936  degrees of freedom
Residual deviance: 2945.1  on 2929  degrees of freedom
AIC: 2961.1

Number of Fisher Scoring iterations: 5
```

In particular, the values contained in the **“Estimate”** column represent the increase in the log odds of class quality, the response variable, associated with a one-unit increase in each predictor value when all the other predictor values are fixed. The log odds is the logarithm of the ratio between the probability that the wine is good and the probability that the wine is not good. In practice, if we consider the coefficient estimate associated to the first variable, volatile acidity, we can see that it is about -6.3; the negative value indicates that an increase in volatile acidity is associated with a decrease in the probability of having a good wine. To be precise, a one-unit increase in volatile acidity is associated with a decrease in the log odds of class quality by 6.3 units.
Also for the variable density we can say a similar thing, as the associated coefficient estimate is negative: an increase in density is associated with a decrease in the probability of having a good wine.
On the contrary, in correspondance to the other variables we have positive coefficient estimates; this means that an increase in residual sugar, or free sulfur dioxide, or pH, or sulphates, or alcohol, is associated with an increase in the probability of having a good wine.
The estimated intercept is typically not of interest; its main purpose is to adjust the average fitted probabilities to the proportion of ones in the data.

The **p-value** associated with each variable (last column of the table), is a measure of the accuracy of the coefficient estimates. Since all the p-values are very small we conclude that there is a strong association between each predictor and probability of having an high quality wine.

Once the coefficients have been estimated, we have to investigate the model adequacy. The

goodness of fit of the logistic regression model can be assessed by the deviance. As we ca see above, R reports two forms of deviance: the null deviance and the residual deviance. The first one shows how well the response variable is predicted by a model that includes only the intercept (grand mean); the second one, instead, shows how well the response is predicted by the model when the predictors are included. If the Null Deviance is really small, it means that the Null Model explains the data pretty well. Likewise with the Residual Deviance.
In this case, we have a value of 3744.4 on 2936 degrees of freedom. Including the independent variables decreased the deviance to 2945.1 points on 2929 degrees of freedom, a significant reduction in deviance. The Residual Deviance has reduced by 799.3 with a loss of seven degrees of freedom (because the predictors are seven).

The algorithm, as we said, has chosen the model with the lowest AIC. This is equivalent to say that it has tried to maximize the log-likelihood function. In fact the AIC is related to the opposite of the log-likelihood. The maximization ot the log-likelihood function occurs through some numerical optimization algorithm like the **Fisher scoring**. In this case we see that Fisher's Scoring Algorithm needed 5 iterations to perform the fit.

We can compute the confusion matrix which is a contigency table containing the information about actual and predicted classifications. If we choose 0.5 as threshold we have:

		True values		**Total**
		Bad	***Good***	
Predicted values	***Bad***	499	244	743
	Good	484	1710	2194
Total		983	1954	2937

Table 2: Confusion matrix related to the training dataset and the logistic regression model.

The sum of the diagonal elements of the matrix gives the number of right classified units otherwise the sum of the off diagonal elements gives the number of wrong classified units. If we consider the sum of the off diagonal elements, we divide it by the number of the total observations and we multiply by 100, we obtain the **misclassification error** that is the percentage of the misclassified units. In this case the misclassification error is about equal to 24.79%. In other words, this model has an accuracy of 76.21%. Despite this, if we look the column "True values", we can see, firstly, that the model has misclassificated about the 50% of bad wines, which is an high percentage! Secondly, we can see that the model has misclassificated about the 12.5% of good wines, which is not an high percentage. We conclude that the model seems to be not so good because it misclassifies bad wines.

Actually there are two kind of misclassifications: some misclassifications, called false positive, will result from incorrectly assigning a good wine to the low quality class and others, called false negative, will result from incorrectly assigning a bad wine to the high quality class. So we compute the **true positive rate** (TPR) and the **false positive rate** (FPR) to characterize the performance of a classifier. The TPR is the percentage of real good wines that are individuated; the FPR is the percentage of real bad wines that are wrongly classified as high quality wines. In this case, if we choose 0.5 as threshold, we obtain $TPR = 87.51$ and $FPR = 49.24$.

The Receiver Operating Characteristic (ROC) curve is used to assess the accuracy of a continuous measurement for predicting a binary outcome. We need the **ROC curve** because, actually, the TPR and the FPR depend on a fixed cutoff c. Since the cutoff is not usually fixed in advance,

we can plot the TPR(c) against the FPR(c) for all possible values of c. This is exactly what the ROC curve is. The overall performance of a classifier, summarized over all possible thresholds, is given by the area under the ROC curve (AUC). An ideal ROC curve will hug the top left corner, so the larger the AUC the better the classifier. In this case we have $AUC = 0.80$. Since the ideal case is having $AUC = 1$, we can conclude, as we expected, that the fitted model is not a good model.

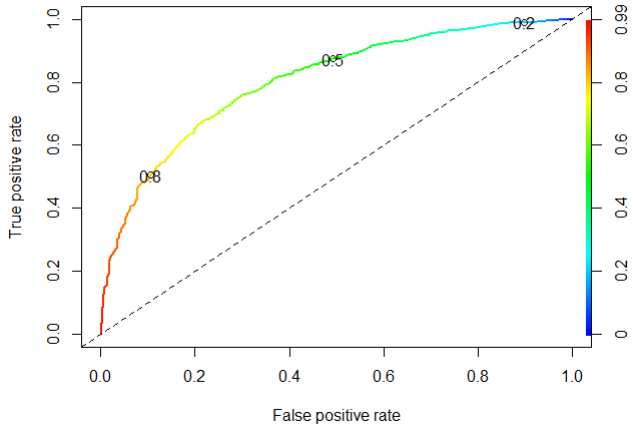


Figure 5: ROC curve for the logistic regression model.

So far we have fitted the model on the training data. Now we compute the model on the validation dataset in order to discover how much the model has learned.
The validation set contains 983 observations of 13 variables. The first one refears to the index of the wine so then, in order to do the analysis, we will remove it.
We then compute the confusion matrix, as we did before. If we choose 0.5 as threshold we have:

		True values		**Total**
		Bad	***Good***	
Predicted values	***Bad***	175	93	268
	Good	155	560	715
Total		330	653	983

Table 3: Confusion matrix related to the validation dataset and the logistic regression model.

Now the misclassification error is about equal to 25.23% which is similar to 24.79%, that is the one we obtained before in correspondance to the training dataset. If we look to the column "True values" , as we did before, we obtain the same proportions and we conclude that the model misclassifies bad wines.

3.2 Second approach: classification trees

A **classification tree** is a structural mapping of binary decisions that lead to a decision about the class of an object. A classification tree is composed of branches that represent attributes,

while the leaves represent decisions. In practice, the decision process starts at the root and follows the branches until a leaf is reached. In essence, the algorithm iteratively selects the attribute and value that can split a set of samples into two groups, minimizing the variability within each subgroup while maximizing the contrast between the groups.

We start fitting a classification tree model based to the training dataset. The classification tree obtained is reported in Figure 6.

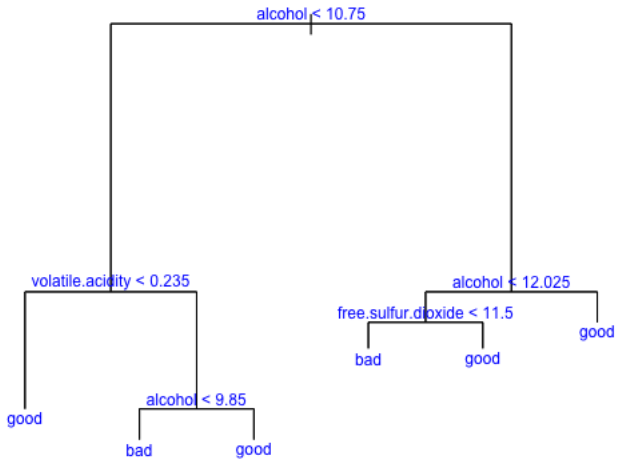


Figure 6: Classification tree

How can the tree be useful in order to predict if a wine is "good" or not? We start from the root of the tree; if the alcohol value is less than 10.75, we follow the left branch until we reach the decision node. Otherwise, we follow the right branch until we reach another decision node. In both cases, once we reach a decision node, we continue following the right or the left branch until a leaf is reached. For example, suppose we follow always the left branch: if a wine has an alcohol value less than 10.75 and a volatile acidity less than 0.235, then the classification tree classifies it as a "good wine".
In this case, we can see that the variables used in tree construction are alcohol, volatile acidity and sulfur dioxide. Furthermore the misclassification errore rate (included in the "summary" function) is equal about to 24.79%. More details can be found in the Appendix, paragraph 6.3.
Now we evaluate the performance of this model on the validation dataset. We obtain the following confusion matrix:

		True values		Total
		Bad	***Good***	
Predicted values	***Bad***	172	99	271
	Good	158	554	712
Total		330	653	983

Table 4: Confusion matrix related to the validation dataset and the classification tree model.

We deduce that this model leads to correct predictions for around 73.86% of the locations in the validation data set. In other words, it misclassifies the 26.14% of the observations. If we look to the column "True values" we can see that the model misclassifies bad wines, as the 47% of bad wines are classified as good wines. This means that this model is not good.

3.2.1 Cross-validation and pruning

One of the questions that arises in a decision tree algorithm is the optimal size of the final tree. A tree that is too large risks overfitting the training data and poorly generalizing to new samples. A small tree might not capture important structural information about the sample space. A common strategy it the following one: after we have obtained a tree (for example growing the tree until each node contains a small number of instances), we use pruning to remove nodes that do not provide additional information.

Pruning is a data compression technique that reduces the size of decision trees by removing sections of the tree that are redundant to classify instances. Pruning reduces the complexity of the final classifier because it reduces the size of the tree but it should not reduce predictive accuracy! In order to find the best pruned tree, that is the best subtree of the initial tree, we use the **cross-validation** approach because with this approach we can estimate the accuracy of a subtree and choose the one having the highest accuracy.
The cross validation approach leads to obtain the plot showed in Figure 7:

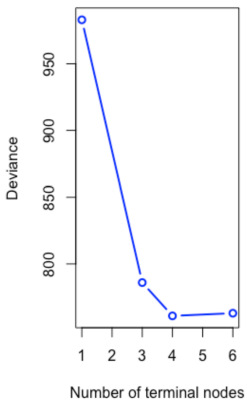


Figure 7: Value of deviance in correspondance to a number of terminal nodes going from 1 to 6. The optimal number of nodes is 4.

The plot shows how Deviance (y-axis) changes when the number of the terminal nodes (x-axis),

that is the number of leaves, changes. In this case, the optimal number of leaves, is 4. We prune the tree in such a way it has only 4 terminal nodes and we obtain the following tree:

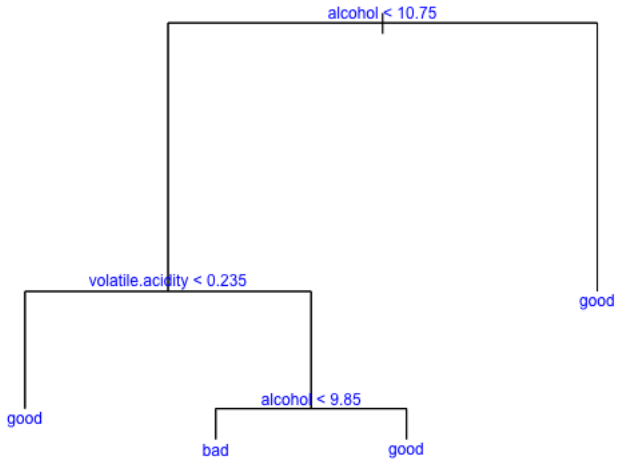


Figure 8: Pruned tree

We can see that the variables used in tree construction are alcohol and volatile acidity. Furthermore the misclassification errore rate (included in the "summary" function) is equal about to 25%. More details can be found in the Appendix, paragraph 6.4.
Now we evaluate the performance of this model on the validation dataset. We obtain the following confusion matrix:

		True values		**Total**
		Bad	***Good***	
Predicted values	***Bad***	162	92	254
	Good	168	561	729
Total		330	653	983

Table 5: Confusion matrix related to the validation dataset and the pruned tree model.

We deduce that this model leads to correct predictions for around 73.56% of the locations in the validation data set. In other words, it misclassifies the 26.44% of the observations. If we look to the column "True values" we can see that the model misclassifies about the 50% of bad wines. This means that this model, as the previous ones, is not a good model but the pruning process has produced a more interpretable tree.

3.2.2 Ensemble methods: bagging, random forest, boosting

There is a drowback in using just one tree: a decision tree suffer from high variance. The idea to overcome this problem is to combine several decision trees to produce better predictive performance than using a single decision tree. Procedures like this are called **Ensemble methods**.

- **BAGGing** (or Bootstrap AGGregation) is one of the ensemble methods and so it can be used to reduce the variance of a statistical learning method. The idea of **BAGGing** is to create several subsets of data from training sample chosen randomly with replacement. Each collection of subset data is used to train their decision trees. As a result, we end up with an ensemble of different models. Average of all the predictions from different trees are used which is more robust than a single decision tree.

  If we fit a model on the training dataset using bagging, we obtain an error of about 18%, to be precise 17.91% (details can be found in the Appendix).
  We have to highlight that bagging improves prediction accuracy at the expense of interpretability. This means that the collection of bagged trees is much more difficult to interpret than a single tree. However, one can obtain an overall summary of the importance of each predictor. Here we use the mean decrease accuracy index and the mean decrease Gini index to find the most important predictors (see paragraph 6.5 for an explanation of these two indices). The following figure shows only graphically the importance of each predictor according to the two indices cited above, but more details can be found in the Appendix, paragraph 6.5:

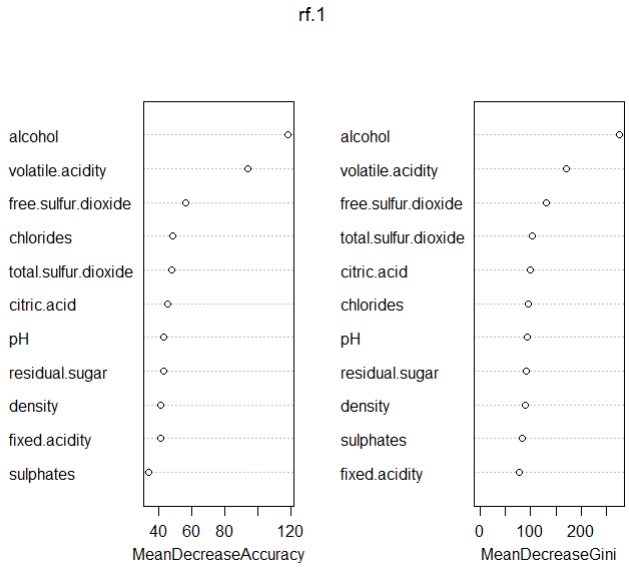


Figure 9: On the left: mean decrease accuracy value of each input variable according to the bagging model. On the rigth: mean decrease Gini value of each input variable according to the bagging model.

The left plot shows the value of the Mean decrease accuracy related to each of the predictors. The right one shows the value of the Mean decrease Gini related to each of the predictors.
To choose the most important variables we have to look to the ones with a big value of mean decrease accuracy or mean decreas Gini and to the gaps between points. In this case

if we look to the points from the left to the right in the left plot, we can see an evident big gap between volatile acidity and free sulfur dioxide. This means that volatile acidity and alcohol have values of mean decrease accuracy which are much bigger than the others. Therefore we deduce that they are the most important variables. Then if we do the same thing in the right plot we can see a big gap between alcohol and volatile acidity, then we can deduce that alcohol is the most important predictor according to the mean decrease Gini.

Now we evaluate the performance of this model on the validation dataset. We obtain the following confusion matrix:

		True values		**Total**
		Bad	***Good***	
Predicted values	***Bad***	236	65	301
	Good	94	588	682
Total		330	653	983

Table 6: Confusion matrix related to the validation dataset and the bagging model.

We deduce that this model leads to correct predictions for around 84% of the locations in the validation data set. In other words, it misclassifies the 16% of the observations.
If we look to the column "True values" we deduce that this model classifies incorretly the 28% of bad wines and the 10% of good wines. These percentage are lower than the ones obtained with the previous models.

These results we have obtained show that an ensamble method, bagging in this case, can improve the accuracy of the final model.

- Another ensamble method is the one called **Random forest**. Random forest is a substantial modification of bagging that builds a large collection of de-correlated trees, and then averages them. Indeed we used bagging because the variance of the tree obtained doing the average of the trees is lower than the variance of just one tree. However, the variance obtained averaging trees depends also on the correlation of the trees. So if we want to improve the variance reduction of bagging, we have to reduce the correlation between the trees. This is the idea of the random forest model.
  But how this approach is able to construct de-correlated trees? Random forest, as bagging does, takes the random subset of data; but, in addition, it takes the random selection of the predictors rather than using all predictors to grow trees, as instead bagging does.

  If we fit a model on the training dataset using random forest, we obtain an error almost equal to the one of the previous model, that is 17.67%.
  Figure 10 shows only graphically the importance of each predictor, but more numerical details can be found in the Appendix, paragraph 6.6:

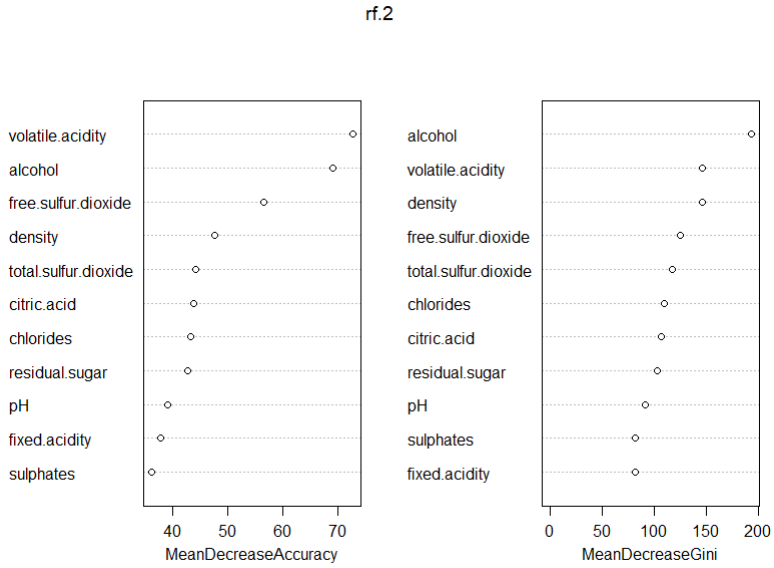


Figure 10: On the left: mean decrease accuray value of each input variable according to the random forest model. On the right: mean decrease Gini value of each input variable according to the random forest model.

To choose the most important variables we follow the same procedure of the previous model. In this case if we look to the points from the left to the right in the left plot, we can see an evident big gap between alcohol and free sulfur dioxide. This means that volatile acidity and alcohol have values of mean decrease accuracy which are much bigger than the others. Therefore we deduce that they are the most important variables. Then if we do the same thing in the right plot we can see a big gap between alcohol and volatile acidity, then we can deduce that alcohol is the most important predictor according to the mean decrease Gini.
Now we evaluate the performance of this model on the validation dataset. We obtain the following confusion matrix:

		True values		**Total**
		Bad	***Good***	
Predicted values	***Bad***	230	61	291
	Good	100	592	692
Total		330	653	983

Table 7: Confusion matrix related to the validation dataset and the random forest model.

We deduce that this model, as the previous one, leads to correct predictions for around 84% of the locations in the validation data set. In other words, it misclassifies the 16% of the observations.
If we look to the column "True values" we deduce that this model classifies incorrectly the 30% of bad wines and the 9% of good wines. We conclude that this model is similar to the bagging tree.

- **Boosting**

Boosting is another approach for improving the predictions resulting from a decision tree. Bagging works more or less like boosting Training dataset, but the trees are geown sequentially. To be more precisely, each tree is grown using information from previously grown trees and, at every step, the goal is to solve for net error from the prior tree. If we fit this model using the train dataset, the confusion matrix we obtain is the following:

		True values		**Total**
		Bad	***Good***	
Predicted values	***Bad***	703	161	864
	Good	280	1793	2073
Total		983	1954	2937

Table 8: Confusion matrix related to the training dataset and the boosting model.

In this case the misclassification error is about equal to 15%. In other words, this model has an accuracy of 85%, which is higher than those related to the models we have fitted previously. In addition, if we look the column "True values", we can see, firstly, that the model has misclassificated about the 28% of bad wines, which is a percentage lower than those of the previous models! Secondly, we can see that the model has misclassificated about the 8% of good wines, which is not an high percentage. We conclude that the model seems to be good.

Figure 6.7 shows only graphically the relative influence of each variable (for the numerical values of the relative influence of each variable, see the Appendix, paragraph 6.7):

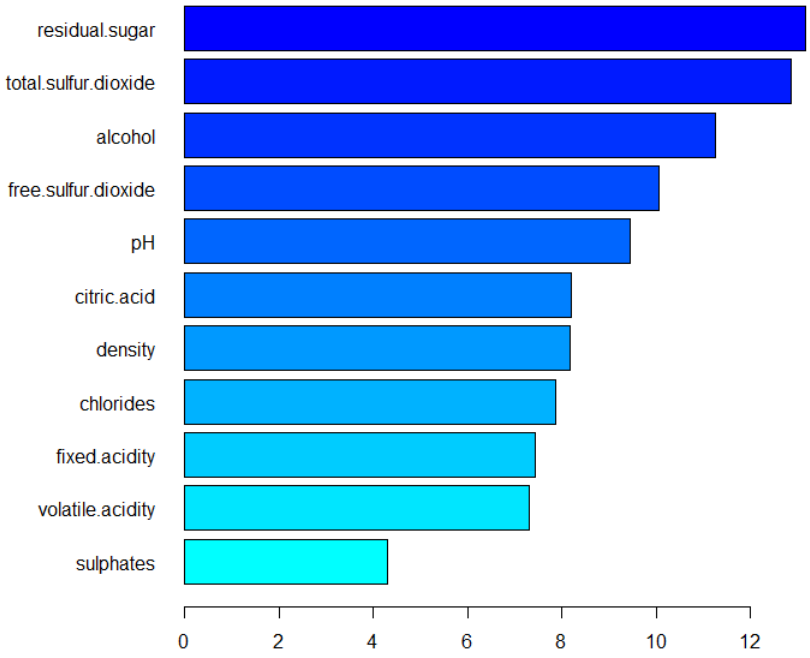


Figure 11: Relative influence of each input variable according to the boosting model.

Looking at this plot, we deduce that the most important variables are residual sugar and total

sulfur dioxide.

Now we evaluate the performance of this model on the validation dataset. We obtain the following confusion matrix:

		True values		**Total**
		Bad	***Good***	
Predicted values	***Bad***	193	96	289
	Good	137	557	694
Total		330	653	983

Table 9: Confusion matrix related to the validation dataset and the boosting model.

The misclassification error is about equal to 24%. In other words, this model has an accuracy of 76%. If we look the column "True values", we can see, firstly, that the model has misclassificated about the 40% of bad wines, which is an high percentage! Secondly we can see that the model has misclassificated about the 14% of good wines, which is not an high percentage. We conclude that this model, although it seemed to be good looking at the confusion matrix related to the training dataset, when tested on the validation dataset, we conclude that it is not so good because it misclassifies an high percentage of bad wines.

3.3 Third approach: neural networks

The last approach is neural network. We modelled 5 different neural networks with just one hidden layer but with a number of neurons going from 3 to 7. The number of repetitions we chose for each neural network's training is 3.
Looking at the misclassification error of the 5 models, we can say that the best model is the one with 6 neurons, as its misclassification error, 23%, is the lowest. Figure 12 shows this neural network.

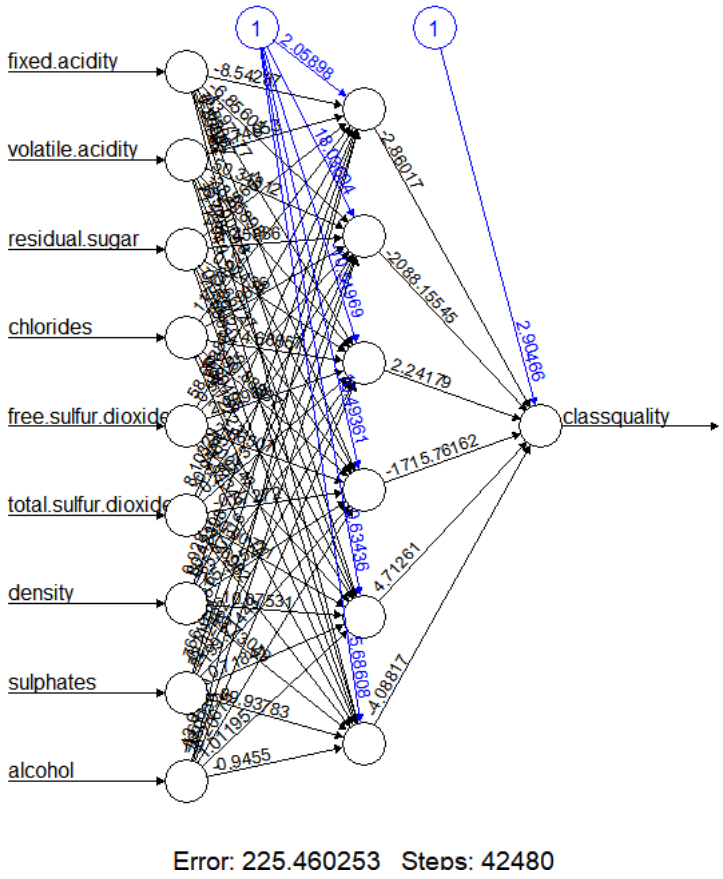


Figure 12: Neural network with six neurons and one hidden layer.

The confusion matrix we obtain after testing this neural network on the validation dataset is the following:

		True values		Total
		Bad	***Good***	
Predicted values	***Bad***	202	89	300
	Good	128	555	683
Total		330	653	983

Table 10: Confusion matrix related to the validation dataset and the neural network model.

If we look to the column "True values" we can see that this model misclassifies almost the 40% of bad wines, which is an high percentage, so it is not a good model.
According to the value of the misclassification error, this model has the lowest one, as we said. Actually, if we look at the other neural network models we have obtained, we can see that the correspondant misclassification errors are not so different. Therefore, in the Appendix, paragraph 6.8, we have added the misclassification error of the other neural networks and the correspondant confusion matrix in order to make a comparison with the Table 10.

4 Comparison of the results and choice of the best model

After we have fitted on the data different models, we have to chose the best one according to the correspondant value of misclassification error. Therefore, we summarize the results we have obtained so far:

- Logistic regression model: misclassification error equal to 25.23%. The model misclassifies about the 50% of bad wines.
- Trees
  - Classification tree (without pruning): misclassification error equal to 26.14%. The model misclassifies the 47% of bad wines.
  - Classification tree with pruning: misclassification error equal to 26.44%. The model misclassifies about the 50% of bad wines.
  - Ensemble methods
    - Bagging: misclassification error equal to 16%. The model misclassifies the 28% of bad wines.
    - Random forest: misclassification error equal to 16%. The model misclassifies the 30% of bad wines.
    - Boosting: misclassification error equal to 24%. The model misclassifies the 40% of bad wines.
- Neural network model: misclassification error equal to 23%. The model misclassifies an high percentage of bad wines.

We can deduce that the best models are the ones obtained with bagging and with random forest as the other ones misclassify an high percentage of bad wines. The selected two models are very similar as they misclassify the 28% and 30% of bad wines respectively. As far as it concerned the number of good wines misclassified we can see, if we go back, that they are 10% and 9% respectively. Therefore, from a mathematical point of view these two models are equivalent. The choice of one of them depends if oenologists prefer having an higher percentage of bad wines misclassified or an higher percentage of good wines misclassified.
We test both these two models on the test dataset.

5 Prediction target values

The test dataset contains 978 observations of all the variables we have already described in the Introduction 1 except for quality. Indeed, the quality of these observations must be predicted.
The quality of each wine is predicted considering the bagging and the random forest models we have selected previously.
The results are reported in the following two tables:

Predicted values *(*bagging)	
Bad	***Good***
305	673

Table 11: Predicted qualities of the wines contained in the test dataset using the bagging model.

Predicted values (random forest)	
Bad	***Good***
289	689

Table 12: Predicted qualities of the wines contained in the test dataset using the random forest model.

The following last table shows the comparison between the results obtained with the two previous models (bagging and random forest):

		Random forest model		**Total**
		Bad	***Good***	
Bagging model	***Bad***	279	26	305
	Good	10	663	673
Total		289	689	978

Table 13: Confusion matrix related to the test dataset and the two selected best models.

The two selected models lead to different results. As we said before, oenologists must choose one of them because from a mathematical point of view they are equivalent.

6 Appendix

6.1 Table 1

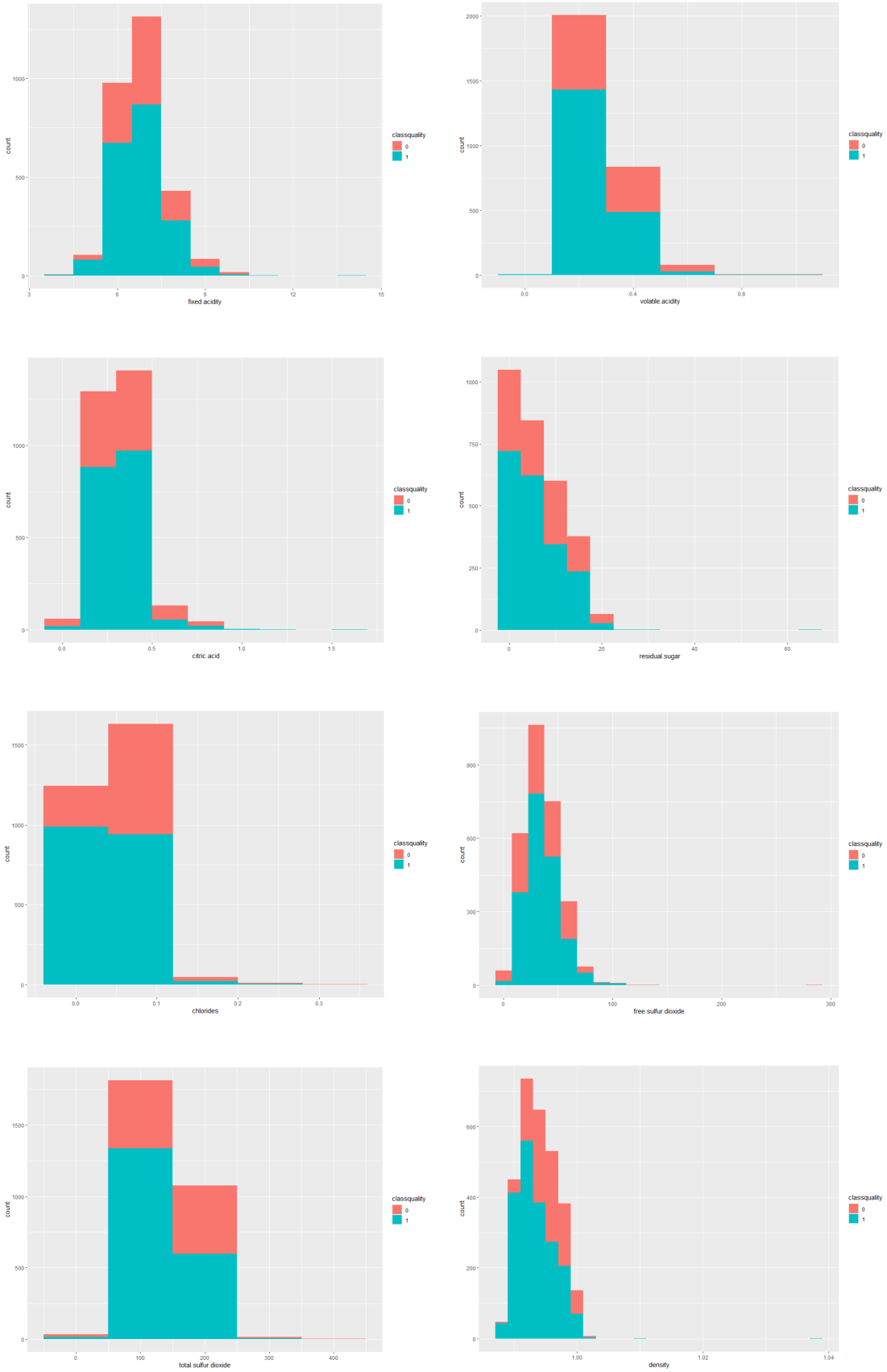
In this paragraph we precise the meaning of the summary statistics presented in Table 1.

- Minimum (Min.): minimum value.
- First quartile (1st Qu.): the point such that the 25% of the data is below it.
- Median: the middle value.
- Mean: the average value.
- Third quartile (3rd Qu.): the point such that the 75% of the data is below it.
- Maximum (Max.): maximum value.
- Range: biggest value minus the smallest value, it gives the full spread of the data.
- Standard deviation (Sd): the square root of the variance, which represents the average squared deviation from the mean. The standard deviation measures the average deviation of the values, in the data, from the mean value.
- Coefficient of variation (Cv): standard deviation divided by the mean so it is a measure of relative variability.
- Skewness: is a measure of symmetry. It is defined as $\frac{E[(x-\bar{x})^3]}{\sigma^3}$ where σ is the standard deviation and $\bar{x}$ is the mean of the values.

A note about variance, standard deviation and coefficient of variation. The variance is completely uninterpretable because it doesn't use the same units as the data, this is the reason why standard deviation is preferred. Indeed standard deviation is expressed in the same units as the data. A drowback of the standard deviation is that we can get no information by comparing standard deviation relatad to different variables if the values of the variables are completely different. The solution is to use the coefficient of variation.

6.2 Figures 3 and 4

In this paragraph we discuss the results reported in Figures 3 and 4. Indeed, in these latter figures, for some input variables, there are intervals in which the 100% of wines have high quality. We have to pay attention because this fact does not imply necessarily that those value of the input variable characterize high quality wines, as the number of observations that fall in those intervals could be very low. Therefore, we have to check the number of observations in these intervals. In order to do this, we look at the conditional distributions of each input variable on values of the variable "class quality", as we did before, but without computing the proportion, in each interval, of wine with high quality and low quality and just counting the number of observations in each interval. Figure 13 show these new conditional distributions.


count
fixed.acidity
volatile.acidity
citric.acid
residual.sugar
chlorides
free.sulfur.dioxide
total.sulfur.dioxide
density
classquality
0
1

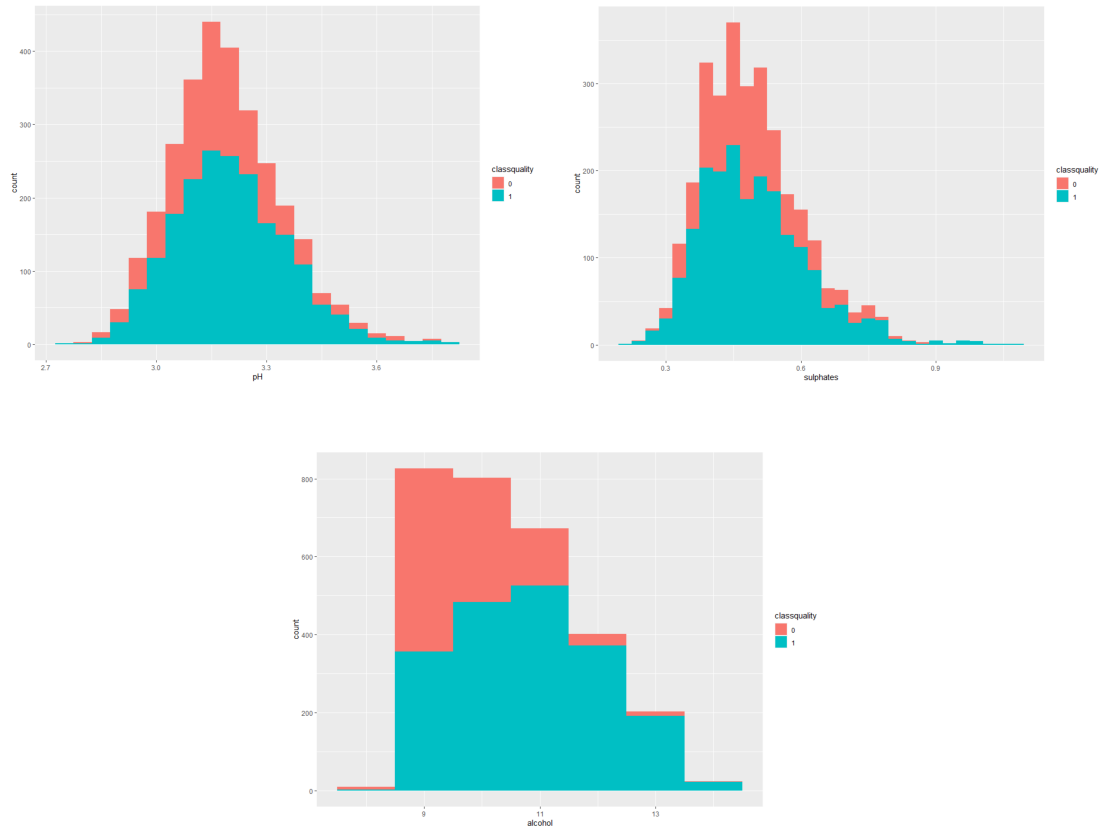


Figure 13: Conditional distributions of the input variables on values of classquality. For each input variable, the conditional distribution gives the counting of high and low quality wine with respect to each interval in which the values of the input variable are suddivided.

Looking at the Figure 13 we can deduce the conclusions reported in paragraph 2.2 about what are the values of the input variables that characterize wine quality. Indeed, we recall that in that paragraph, in particular, we did not take into account some intervals related to the input variables fixed acidity, citric acid, residual sugar, density, sulphates.

6.3 Figure 6

In this paragraph we give details about the classification tree of Figure 6. The summary of this tree is the following:

```
Classification tree:
tree(formula = classquality ~ . - quality, data = data_train1)
Variables actually used in tree construction:
[1] "alcohol"            "volatile.acidity"    "free.sulfur.dioxide"
Number of terminal nodes:  6
Residual mean deviance:  1.022 = 2995 / 2931
Misclassification error rate: 0.2479 = 728 / 2937
```

In particular we can see that the tree has 6 terminal nodes and that the variables used in the tree construction are alcohol, volatile acidity and free sulfur dioxide.

Here we can see further informations about this tree:

```
node), split, n, deviance, yval, (yprob)
      * denotes terminal node

 1) root 2937 3744.00 good ( 0.33470 0.66530 )
   2) alcohol < 10.75 1772 2452.00 good ( 0.47404 0.52596 )
     4) volatile.acidity < 0.235 659  764.90 good ( 0.26707 0.73293 ) *
     5) volatile.acidity > 0.235 1113 1501.00 bad ( 0.59659 0.40341 )
      10) alcohol < 9.85 716  904.90 bad ( 0.67318 0.32682 ) *
      11) alcohol > 9.85 397  547.60 good ( 0.45844 0.54156 ) *
   3) alcohol > 10.75 1165  867.60 good ( 0.12275 0.87725 )
     6) alcohol < 12.025 733  669.60 good ( 0.17053 0.82947 )
      12) free.sulfur.dioxide < 11.5 45   61.29 bad ( 0.57778 0.42222 ) *
      13) free.sulfur.dioxide > 11.5 688  566.90 good ( 0.14390 0.85610 ) *
     7) alcohol > 12.025 432  149.60 good ( 0.04167 0.95833 ) *
```

The first line of the output gives us the key to reading the results:

- node): node number;
- split: subdivision criterion;
- n: number of observations;
- deviance: the value of the Gini index or entropy;
- yval: the fitted value at the node, that is the majority class in the case of classification trees (as in our case);
- (yprob): fitted probabilities for each response level. In this case there are two probabilities, the first one represents the probability that an observation has a value of classquality equal to 0 (which means bad wine) and the second one represents the probability that an observation has a value of classquality equal to 1 (which means good wine).
- The asterisk denotes terminal node, in this case the terminal nodes are volatile acidity, alcohol and free sulfur dioxide.

6.4 Figure 8

In this paragraph we give details about the pruned tree of Figure 8. The summary of this tree is the following:

```
Classification tree:
snip.tree(tree = tree.datatrain1, nodes = 3L)
Variables actually used in tree construction:
[1] "alcohol"          "volatile.acidity"
Number of terminal nodes:  4
Residual mean deviance:  1.052 = 3085 / 2933
Misclassification error rate: 0.2503 = 735 / 2937
```

In particular we can see that the tree has 4 terminal nodes and that the variables used in the tree construction are alcohol and volatile acidity.

Here we can see further informations about this tree:

```
node), split, n, deviance, yval, (yprob)
      * denotes terminal node

 1) root 2937 3744.0 good ( 0.3347 0.6653 )
   2) alcohol < 10.75 1772 2452.0 good ( 0.4740 0.5260 )
     4) volatile.acidity < 0.235 659  764.9 good ( 0.2671 0.7329 ) *
     5) volatile.acidity > 0.235 1113 1501.0 bad ( 0.5966 0.4034 )
      10) alcohol < 9.85 716  904.9 bad ( 0.6732 0.3268 ) *
      11) alcohol > 9.85 397  547.6 good ( 0.4584 0.5416 ) *
   3) alcohol > 10.75 1165  867.6 good ( 0.1227 0.8773 ) *
```

The meaning of the output variables is given in the paragraph 6.3.

6.5 Figure 9

In this paragraph we explain what is the meaning of the mean decrease accuracy index and the mean decrease Gini index and we give more details than the ones we gave in Figure 9 that refears to the tree obtained with bagging.
The indices used to measure the importance of each variable are the mean decrease accuracy index and the mean decrease Gini one. The Mean Decrease Accuracy plot expresses how much accuracy the model losses by excluding each variable. The more the accuracy suffers, the more important the variable is for the successful classification. The variables are presented from descending importance. The mean decrease in Gini coefficient is a measure of how each variable contributes to the homogeneity of the nodes and leaves in the resulting random forest. The higher the value of mean decrease accuracy or mean decrease Gini score, the higher the importance of the variable in the model.
As far as it concerned Figure 9, the summary of this tree is the following:

```
> rf.1=randomForest(classquality ~ . - quality, data = data_train1, mtry=11,importance=TRUE)
> rf.1

Call:
 randomForest(formula = classquality ~ . - quality, data = data_train1,      mtry = 11, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 11

        OOB estimate of  error rate: 17.91%
Confusion matrix:
     bad good class.error
bad  667  316   0.3214649
good 210 1744   0.1074719
```

- The argument mtry= 11 indicates that all 11 predictors should be considered for each split of the trees in other words, that bagging should be done.
- Once we set the option importance = TRUE (by default, it is importance = FALSE), using the importance() function, we can view the importance of each variable.
- The number of trees considered, by default, is 500.
- At the end of the summary we have the error associated to the model and the confusion matrix. We have already discussed both of them in the paragraph 3.2.2.

As we said, the importance() function gives informations about the importance of each variable. The output of this function is the following:

```
> importance(rf.1)
                        bad     good MeanDecreaseAccuracy MeanDecreaseGini
fixed.acidity        28.90557 31.22442             41.40916         77.48368
volatile.acidity     70.23546 66.09764             93.72084        170.91480
citric.acid          32.85148 35.73400             45.63194         98.68750
residual.sugar       23.03137 32.80903             43.11074         90.85996
chlorides            34.89084 31.22576             48.63470         94.81703
free.sulfur.dioxide  42.20548 36.78935             56.45784        130.09473
total.sulfur.dioxide 31.45493 30.28234             47.69937        104.09985
density              24.60249 27.98283             41.44166         89.46817
pH                   32.82490 30.86329             43.21884         94.04446
sulphates            24.56876 26.48816             34.26081         83.05445
alcohol              80.18910 64.66847            118.15504        274.61384
```

The last two columns give the numerical values of mean decrease accuracy and mean decrease Gini for each variable. We recall that Figure 9 gives informations about these two indexes but only in a graphically way, without highlighting the numerical values.

6.6 Figure 10

In this paragraph we give more details than the ones we gave in Figure 10 that refears to the random forest tree. The summary of this tree is the following.

```
> rf.2=randomForest(classquality ~ . - quality, data = data_train1, importance=TRUE)
> rf.2

Call:
 randomForest(formula = classquality ~ . - quality, data = data_train1,      importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 17.67%
Confusion matrix:
     bad good class.error
bad  660  323   0.3285860
good 196 1758   0.1003071
```

The meaning of all the outputs above has been already given in the paragraph 6.5 (the previous one). We remark just a difference: the number of variables tried at each split, in the tree obtained with bagging was 11, instead in one we have obtained with random forest tree is 3. Indeed, bagging uses all the predictors in each split, on the contrary, by default, the randomForest function uses $\lfloor x \rfloor$ variables when building a random forest of classification trees, where with x we have indicated the total number of variables.

As we already said, the importance() function gives informations about the importance of each variable. The output of this function is the following:

```
> importance(rf.2)
                        bad     good MeanDecreaseAccuracy MeanDecreaseGini
fixed.acidity        27.24882 28.14469             37.76738         82.35927
volatile.acidity     55.01740 54.96044             72.72864        146.96633
citric.acid          34.75944 32.99392             43.89105        107.39377
residual.sugar       24.03538 31.87946             42.74234        103.26195
chlorides            35.11455 30.06505             43.25503        110.33433
free.sulfur.dioxide  42.82684 35.33923             56.46269        125.62119
total.sulfur.dioxide 31.94779 29.69388             44.13824        117.30743
density              28.68697 32.79830             47.63056        146.82911
pH                   30.04250 27.87547             39.18957         91.75071
sulphates            27.47558 24.27779             36.16057         82.43694
alcohol              66.14119 43.04238             69.12639        193.92542
```

6.7 Figure 11

In this paragraph we give details related to the tree obtained with boosting. Indeed, we recall that Figure 11 gives informations about the relative influence of each input variable but only in a graphically way, without highlighting the numerical values. Therefore, here, with the summary function we can see the numerical values of each relative influence.

```
> summary(wine.boost,method = relative.influence, las = 1)
                                    var   rel.inf
residual.sugar             residual.sugar 13.171994
total.sulfur.dioxide total.sulfur.dioxide 12.864345
alcohol                         alcohol 11.241819
free.sulfur.dioxide   free.sulfur.dioxide 10.056544
pH                                   pH  9.435412
citric.acid                 citric.acid  8.183029
density                         density  8.176674
chlorides                     chlorides  7.852494
fixed.acidity             fixed.acidity  7.438923
volatile.acidity       volatile.acidity  7.290900
sulphates                     sulphates  4.287865
```

6.8 Table 10

In this paragraph we find the misclassification errors and confusion matrices of the neural networks we have modeled, except those of the neural network we have already discussed in paragraph 3.3, that is the best one. The scope is to underline that actually all these neural networks are very similar to each other.

The neural network with 1 hidden layer and 3 neurons has a misclassification error equal to 25% and its confusion matrix is reported in Table 14.

		True values		**Total**
		Bad	***Good***	
Predicted values	***Bad***	178	94	272
	Good	152	559	711
Total		330	653	983

Table 14: Confusion matrix related to the validation dataset and the neural network model with 1 hidden layer and 3 neurons.

The neural network with 1 hidden layer and 4 neurons has a misclassification error equal to 23% and its confusion matrix is reported in Table 15.

		True values		**Total**
		Bad	***Good***	
Predicted values	***Bad***	198	95	293
	Good	132	558	690
Total		330	653	983

Table 15: Confusion matrix related to the validation dataset and the neural network model with 1 hidden layer and 4 neurons.

The neural network with 1 hidden layer and 5 neurons has a misclassification error equal to 26% and its confusion matrix is reported in Table 16.

		True values		**Total**
		Bad	***Good***	
Predicted values	***Bad***	187	111	298
	Good	143	542	685
Total		330	653	983

Table 16: Confusion matrix related to the validation dataset and the neural network model with 1 hidden layer and 5 neurons.

The neural network with 1 hidden layer and 7 neurons has a misclassification error equal to 26% and its confusion matrix is reported in Table 17.

		True values		**Total**
		Bad	***Good***	
Predicted values	***Bad***	170	95	265
	Good	160	558	718
Total		330	653	983

Table 17: Confusion matrix related to the validation dataset and the neural network model with 1 hidden layer and 7 neurons.

References

[1] Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, José Reis, (2009) Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems* **pp.** 547–553.