Red Wine Quality

Damiano Pincolini

2023-02-25

INTRODUCTION

This work is the second task of the final Capstone Course which is the last step of the Data Science Programme provided by HarvardX on the Exd platform.

Project goal

The main aim I have set is to understand if and how chemical and physical wine attributes affect its perceived quality. I an not willing to evaluate (or even judge) grades reliability, since I want to take into account that a panel of expert have provided them and each wine got a given number of marks, which, to me, makes each single overall wine mark balanced and affordable enough. What I would like to understand is that there is one or more feature that can lead me to reasonably predict that a particular wine is good or not according to any specific "technical" information.

To reach this goal, I will make an exploratory analysis in order to understand how the outcome (quality) behaves and check if there is correlation between quality and other features.

Based on the main result of this explorative data analysis, I will try to understand if all features are somehow relavant to my scope or if the dataset can be reduced in terms of attributes.

At that point I will train and test some Machine Learning models to find out if there is a better way to predict wine quality base on some specific elements. Since I am working with a dataset that includes the final outcome (quality), the model I am going to find out will be supervised and since the outcome is a numerical variable, my task comes into regression perimeter. My aim is to use the caret package and try the most common and widespread models such as linear regression, decision tree, random forest, kNN and XGBoost. I have experienced severe difficulties with neural network in terms of PC resource usage (probably due to the small RAM of my laptop) so I decided to skip this algorithm.

Once I have trained and tested ML regression models, I will evaluate them. Sticking to caret package I want to consider RMSE and Rsquared as main KPIs for each model.

Dataset description

According to the documentation, the content of the dataset I will work on (see references) is made of several inputs including objective tests (e.g. PH values) and of an output which is the grade provided by wine experts based on sensory evaluation (median of at least 3 evaluations). Each expert graded the wine quality between 0 (very bad) and 10 (very excellent).

The dataset has 1599 red wine instances, each of which has got 11 attributes and an output (the grade from wine experts).

Input variables (based on physicochemical tests) are: 1. Fixed acidity (tartaric acid - g / dmˆ3). 2. Volatile acidity (acetic acid - g / dmˆ3): the amount of acetic acid in wine, which at too high of levels can lead to an

unpleasant, vinegar taste. 3. Citric acid (g / dmˆ3): found in small quantities, citric acid can add 'freshness' and flavor to wines. 4. Residual sugar (g / dmˆ3): the amount of sugar remaining after fermentation stops, it's rare to find wines with less than 1 gram/liter and wines with greater than 45 grams/liter are considered sweet. 5. Chlorides (sodium chloride (g / dmˆ3): the amount of salt in the wine. 6. Free sulfur dioxide (mg / dmˆ3): the free form of SO2 exists in equilibrium between molecular SO2 (as a dissolved gas) and bisulfite ion; it prevents microbial growth and the oxidation of wine. 7. Total sulfur dioxide (mg / dmˆ3): amount of free and bound forms of S02; in low concentrations, SO2 is mostly undetectable in wine, but at free SO2 concentrations over 50 ppm, SO2 becomes evident in the nose and taste of wine. 8. Density (g / cmˆ3): the density of water is close to that of water depending on the percent alcohol and sugar content. 9. pH: describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic); most wines are between 3-4 on the pH scale. 10. sulphates: a wine additive which can contribute to sulfur dioxide gas (S02) levels, which acts as an antimicrobial and antioxidant. 10. Sulphates (potassium sulphate - g / dm3). 11. Alcohol (% by volume): the percent alcohol content of the wine.

Output variable (based on sensory data): 1. Quality (score between 0 and 10).

Loading libraries

```
library(readr)
library(tidyverse)
library(moments)
library(GGally)
library(caret)
library(MASS)
library(stats)
library(rpart.plot) # for decision tree plots
library("xgboost") # for X Gradient Boosting model
```

Loading data

IMPORTANT NOTE:
1. The dataset is contained in the wineQualityReds csv file which can be downloaded on the following **github repository**: https://github.com/damianopincolini/Red-Wine-Quality
2. As an alternative, since it is not allowed to download any dataset from Kaggle without providing account credentials to other graders, in the reference section, the first link allows to land on the kaggle's page where it is possibile to download the dataset.

```
wines <- read_csv("wineQualityReds.csv")
colnames(wines)[1] <- "wineId"
```

1. EXPLORATORY DATA ANALYSIS

I want to have a first glance to the variables, its format and the structure of the dataset.

```
head(wines)
```

```
## # A tibble: 6 x 13
##   wineId fixed.a~1 volat~2 citri~3 resid~4 chlor~5 free.~6 total~7 density    pH
##    <dbl>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <dbl>
```

```
## 1      1       7.4    0.7     0        1.9   0.076      11      34   0.998  3.51
## 2      2       7.8    0.88    0        2.6   0.098      25      67   0.997  3.2
## 3      3       7.8    0.76    0.04     2.3   0.092      15      54   0.997  3.26
## 4      4      11.2    0.28    0.56     1.9   0.075      17      60   0.998  3.16
## 5      5       7.4    0.7     0        1.9   0.076      11      34   0.998  3.51
## 6      6       7.4    0.66    0        1.8   0.075      13      40   0.998  3.51
## # ... with 3 more variables: sulphates <dbl>, alcohol <dbl>, quality <dbl>, and
## #   abbreviated variable names 1: fixed.acidity, 2: volatile.acidity,
## #   3: citric.acid, 4: residual.sugar, 5: chlorides, 6: free.sulfur.dioxide,
## #   7: total.sulfur.dioxide
```

```
colnames(wines)
```

```
##  [1] "wineId"               "fixed.acidity"        "volatile.acidity"
##  [4] "citric.acid"          "residual.sugar"       "chlorides"
##  [7] "free.sulfur.dioxide"  "total.sulfur.dioxide" "density"
## [10] "pH"                   "sulphates"            "alcohol"
## [13] "quality"
```

```
str(wines)
```

```
## spc_tbl_ [1,599 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ wineId              : num [1:1599] 1 2 3 4 5 6 7 8 9 10 ...
##  $ fixed.acidity       : num [1:1599] 7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
##  $ volatile.acidity    : num [1:1599] 0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
##  $ citric.acid         : num [1:1599] 0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
##  $ residual.sugar      : num [1:1599] 1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
##  $ chlorides           : num [1:1599] 0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ..
##  $ free.sulfur.dioxide : num [1:1599] 11 25 15 17 11 13 15 15 9 17 ...
##  $ total.sulfur.dioxide: num [1:1599] 34 67 54 60 34 40 59 21 18 102 ...
##  $ density             : num [1:1599] 0.998 0.997 0.997 0.998 0.998 ...
##  $ pH                  : num [1:1599] 3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
##  $ sulphates           : num [1:1599] 0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
##  $ alcohol             : num [1:1599] 9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
##  $ quality             : num [1:1599] 5 5 5 6 5 5 5 7 7 5 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ...1 = col_double(),
##   ..   fixed.acidity = col_double(),
##   ..   volatile.acidity = col_double(),
##   ..   citric.acid = col_double(),
##   ..   residual.sugar = col_double(),
##   ..   chlorides = col_double(),
##   ..   free.sulfur.dioxide = col_double(),
##   ..   total.sulfur.dioxide = col_double(),
##   ..   density = col_double(),
##   ..   pH = col_double(),
##   ..   sulphates = col_double(),
##   ..   alcohol = col_double(),
##   ..   quality = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
summary(wines)
```

```
##      wineId       fixed.acidity   volatile.acidity  citric.acid
##  Min.   :   1.0   Min.   : 4.60   Min.   :0.1200   Min.   :0.000
##  1st Qu.: 400.5   1st Qu.: 7.10   1st Qu.:0.3900   1st Qu.:0.090
##  Median : 800.0   Median : 7.90   Median :0.5200   Median :0.260
##  Mean   : 800.0   Mean   : 8.32   Mean   :0.5278   Mean   :0.271
##  3rd Qu.:1199.5   3rd Qu.: 9.20   3rd Qu.:0.6400   3rd Qu.:0.420
##  Max.   :1599.0   Max.   :15.90   Max.   :1.5800   Max.   :1.000
##  residual.sugar     chlorides       free.sulfur.dioxide total.sulfur.dioxide
##  Min.   : 0.900   Min.   :0.01200   Min.   : 1.00       Min.   :  6.00
##  1st Qu.: 1.900   1st Qu.:0.07000   1st Qu.: 7.00       1st Qu.: 22.00
##  Median : 2.200   Median :0.07900   Median :14.00       Median : 38.00
##  Mean   : 2.539   Mean   :0.08747   Mean   :15.87       Mean   : 46.47
##  3rd Qu.: 2.600   3rd Qu.:0.09000   3rd Qu.:21.00       3rd Qu.: 62.00
##  Max.   :15.500   Max.   :0.61100   Max.   :72.00       Max.   :289.00
##     density             pH          sulphates         alcohol
##  Min.   :0.9901   Min.   :2.740   Min.   :0.3300   Min.   : 8.40
##  1st Qu.:0.9956   1st Qu.:3.210   1st Qu.:0.5500   1st Qu.: 9.50
##  Median :0.9968   Median :3.310   Median :0.6200   Median :10.20
##  Mean   :0.9967   Mean   :3.311   Mean   :0.6581   Mean   :10.42
##  3rd Qu.:0.9978   3rd Qu.:3.400   3rd Qu.:0.7300   3rd Qu.:11.10
##  Max.   :1.0037   Max.   :4.010   Max.   :2.0000   Max.   :14.90
##     quality
##  Min.   :3.000
##  1st Qu.:5.000
##  Median :6.000
##  Mean   :5.636
##  3rd Qu.:6.000
##  Max.   :8.000
```

Univariate analysis of the output (quality)

The first main goal of this section is to well understand how the outcome I want to investigate (quality) behaves.

Five numbers by Tukey

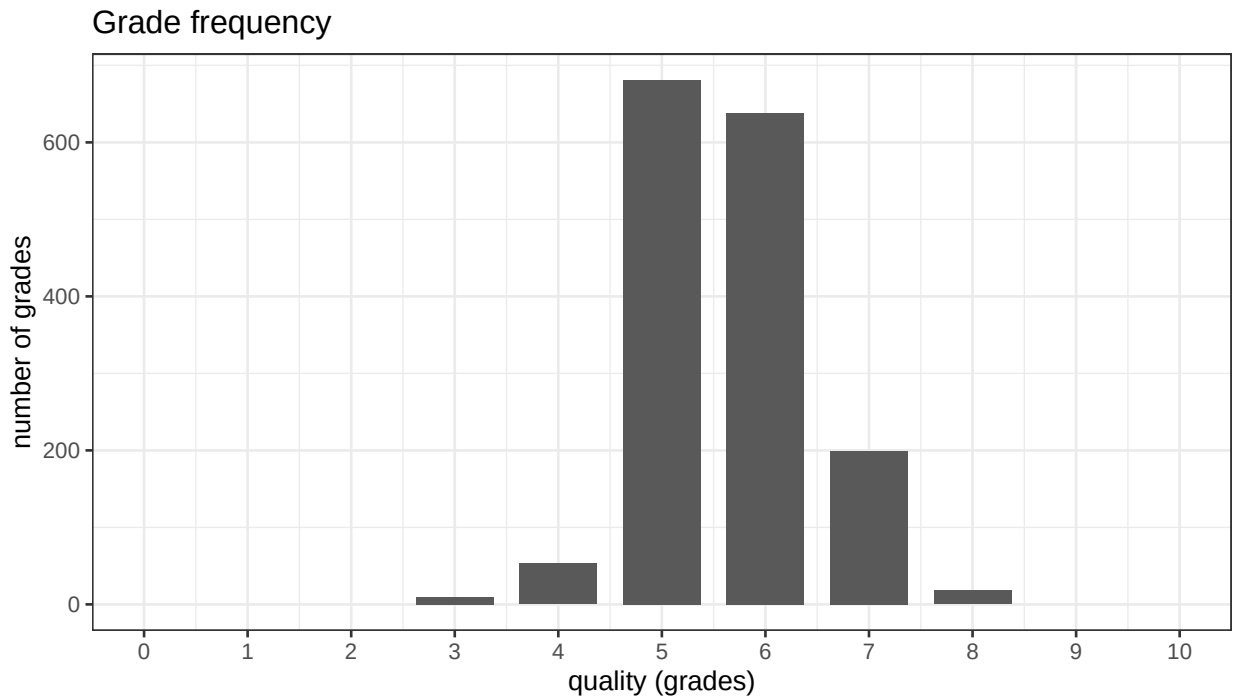
```
summary(wines$quality)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   5.000   6.000   5.636   6.000   8.000
```

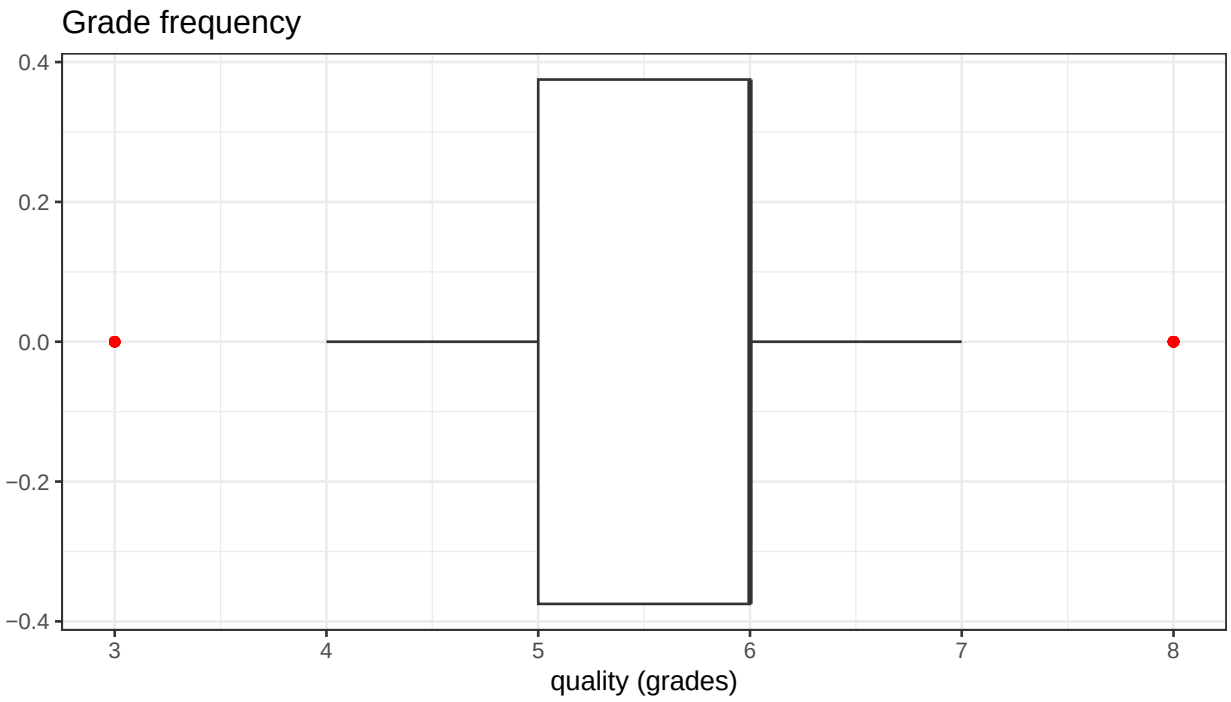
Distribution graphic analysis

Let's start with a graphic analysis

```
ggplot(wines, aes(quality))+
  geom_bar(width=0.75)+
  scale_x_continuous(n.breaks=10,limits=c(0,10))+
  coord_cartesian(xlim=c(0,10))+
  labs(x = "quality (grades)",
       y = "number of grades",
       title ="Grade frequency")+
  theme_bw()
```




```
ggplot(wines, aes(quality))+
  geom_boxplot(outlier.colour="red", orientation="y")+
  labs(x = "quality (grades)",
       title ="Grade frequency")+
  theme_bw()
```



Skweness and kurtosis

I want to measure skewness and kurtosis index:

```
skewness(wines$quality)
```

```
## [1] 0.2175972
```

It is visible a slightly positive skewness (0.217).

```
kurtosis(wines$quality)
```

```
## [1] 3.292031
```

With a 3.29 value, the distribution is slightly leptokurtic, but it does not seems a major issue.

Normality tests

Let's take some test on the distribution of quality variable from the moments package.

```
jarque.test(wines$quality)
```

```
##
##  Jarque-Bera Normality Test
##
## data:  wines$quality
## JB = 18.3, p-value = 0.0001062
## alternative hypothesis: greater
```

In the Jarque-Bera normality test the null hypothesis is that the variable has a skewness and kurtosis that matches a normal distribution, while the alternative hypothesis is that the dataset has a skewness and kurtosis that does not match a normal distribution. Since the p-value is 0.0001062 which is less than 0.05, we can not reject the alternative hypothesis.

```
agostino.test(wines$quality)
```

```
##
##  D'Agostino skewness test
##
## data:  wines$quality
## skew = 0.2176, z = 3.5283, p-value = 0.0004182
## alternative hypothesis: data have a skewness
```

The D'Agostino skewness test returns a p-value of 0.0004182 is lower than 0.05 so we can accept the alternative hypotesis (skewness).

```
anscombe.test(wines$quality)
```

```
##
##  Anscombe-Glynn kurtosis test
##
## data:  wines$quality
## kurt = 3.2920, z = 2.1939, p-value = 0.02824
## alternative hypothesis: kurtosis is not equal to 3
```

The anscombe kurtosis test returns a p-value of 0.028, lower than 0.05, that leads me to accept the alternative hypotesis (kurtosis).

It seems we are facing a non-normal distribution of red wine quality grades.

Is there some issue with outliers? As we have already seen, the first quantile corresponds to grade 5 and the third quantile corresponds to grade 6. So IQR is simply 1.

The limit for lower outliers is:

```
as.vector(summary(wines$quality)[2] - 1.5*IQR(wines$quality))
```

```
## [1] 3.5
```

The limit for higher outliers is:

```
as.vector(summary(wines$quality)[5] + 1.5*IQR(wines$quality))
```

```
## [1] 7.5
```

It appears that the minimun and the maximum grade (respectively 3 and 5) are supposed to be considered as outliers. In the dataset documentation each wine rate is based on sensory evaluation which is the median of at least 3 evaluations. Due to this information, I don't want to lose the information of 3 and 8 grades because the evaluation process seems pretty solid since it takes into account more than a single grade (a least 3); in my opinion this can be considered as a good tip to exclude the presence of misleading information.

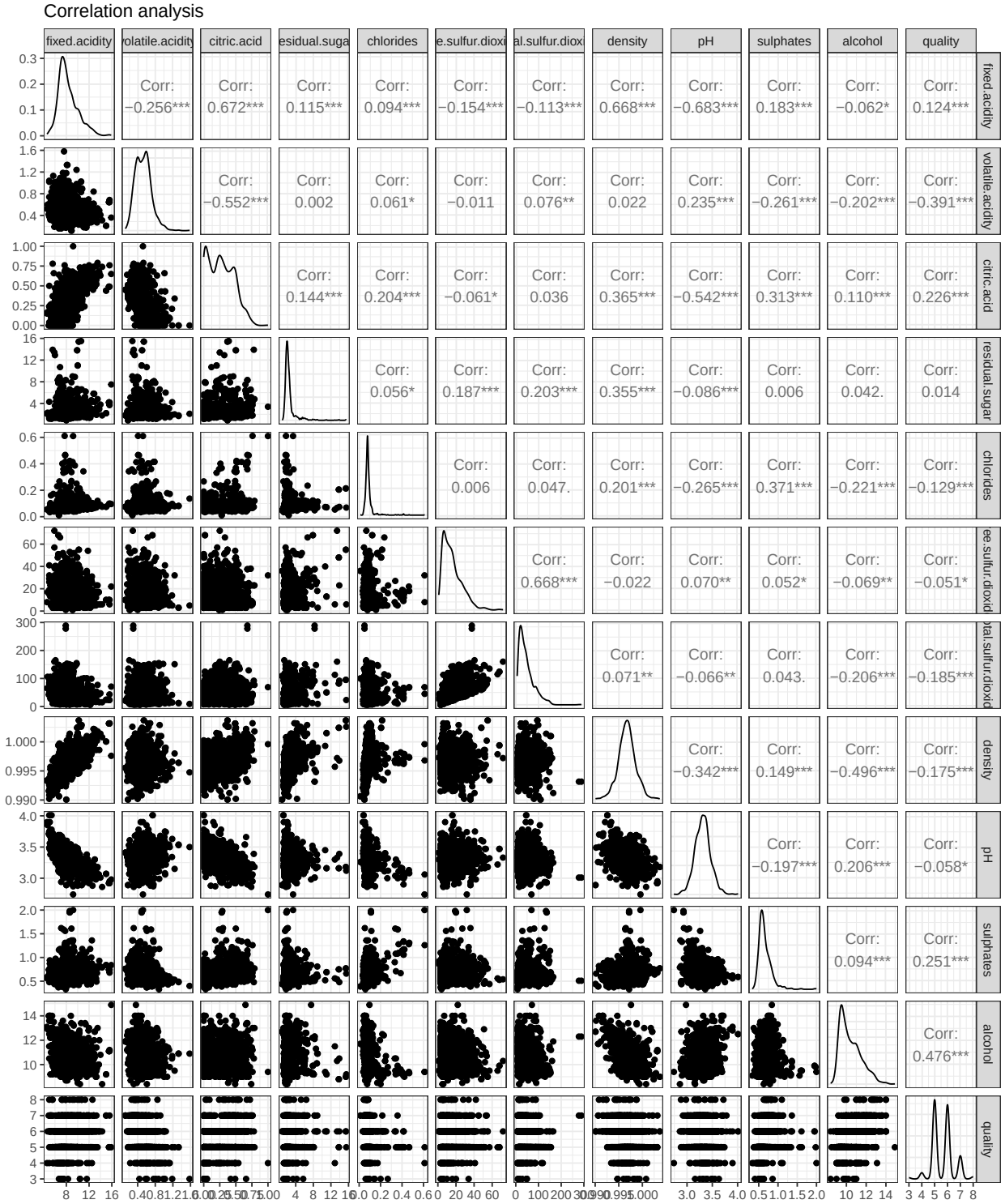
Correlation

Once I have had a first look at the output variable, I want to focus on correlations between variabiles, paying a special attention to quality, which the one I want to understand if is predictable throught a ML model I want to train. I want to detect if there are specific correlations either between quality and other features and between features themselves.

First of all, I need to cancel the wineId column which is a simple progressive counter that does not represent a feature for quality wine prediction.

```
wines2 <- wines %>%
  dplyr::select(2:13)

ggpairs(wines2, title="Correlation analysis", proportions="auto")+
  theme_bw()
```



According to the general rule for which correlation is classified as low, strong and medium if its index absolute value is respectively lower than 0.3, higher than 0.7 and between the two, three elements arise: 1. The features most correlated (respectively in a negative and a positive way) with quality are volatile acidity (-0.391) and alcohol (0.476). 2. Volatile acidity appears to be well positively correlated to citric acid (-0.552). 3. Alcohol is well negatively correlated with density (-0.496).

According to what's above, I will focus on volatile acidity and alcohol (the two most important feature since

I have spotted a direct correlation with quality).

The value of correlation does not appear, in both case, extremely significant. According to this insight, the question that arise is if (before how) the features in this dataset can really give back a model efficiently and steadily able to predict quality wine based on some of its features.

Dataset reduction and analysis of the selected features

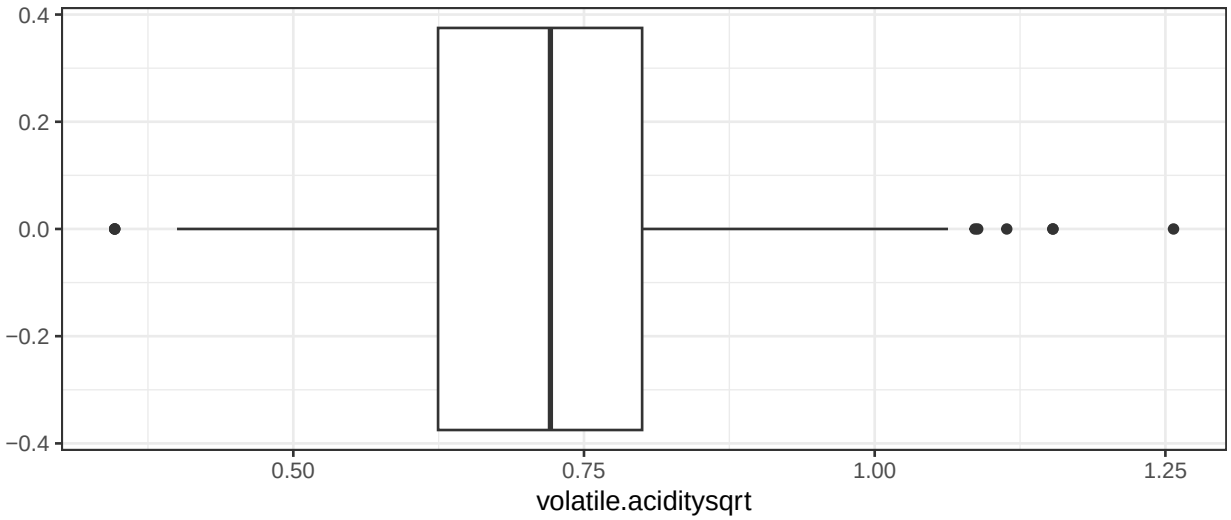
The aim is here to focus on the features that seems to have a fairly good correlation with quality, thus managing the "curse of dimensionality".

In order to reduce the number of features, I want to rescale the dataset values so to carry on the rest of this work, from this point, directly on the new values. During the explorative data analysis, I have spotted both quite a relevant differences in scale amongst features (including the ones I want to keep) and a moderate positive skewness as far as quality is concerner; for these reasons, I will rescale variable throught squared root transformation.
After that, I will check that transformation has not influenced the correlation previously estimated.

```
wines2sqrt <- wines2 %>%
  transmute(volatile.aciditysqrt = sqrt(volatile.acidity),
            alcoholsqrt = sqrt(alcohol),
            qualitysqrt = sqrt(quality)) %>%
  dplyr::select(volatile.aciditysqrt, alcoholsqrt, qualitysqrt)
```

Volatile acidity

```
wines2sqrt %>% ggplot(aes(volatile.aciditysqrt))+
  geom_boxplot()+
  theme_bw()
```




```
skewness(wines2sqrt$volatile.aciditysqrt)
```

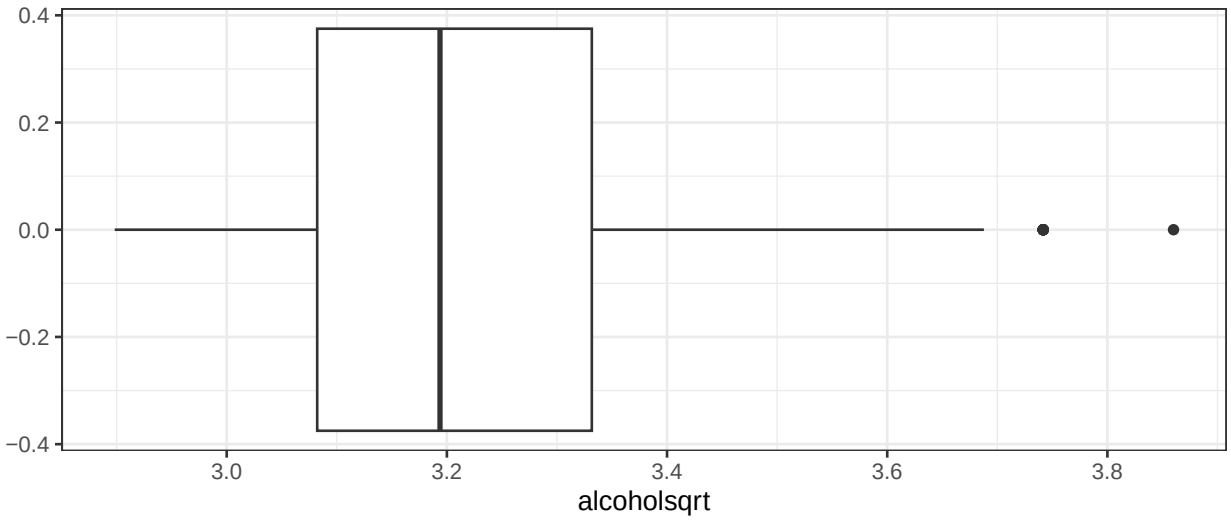
```
## [1] 0.1082536
```

```
kurtosis(wines2sqrt$volatile.acidityсqrt)
```

```
## [1] 3.127039
```

Alcohol

```
wines2sqrt %>% ggplot(aes(alcoholsqrt))+
  geom_boxplot()+
  theme_bw()
```




```
skewness(wines2sqrt$alcoholsqrt)
```

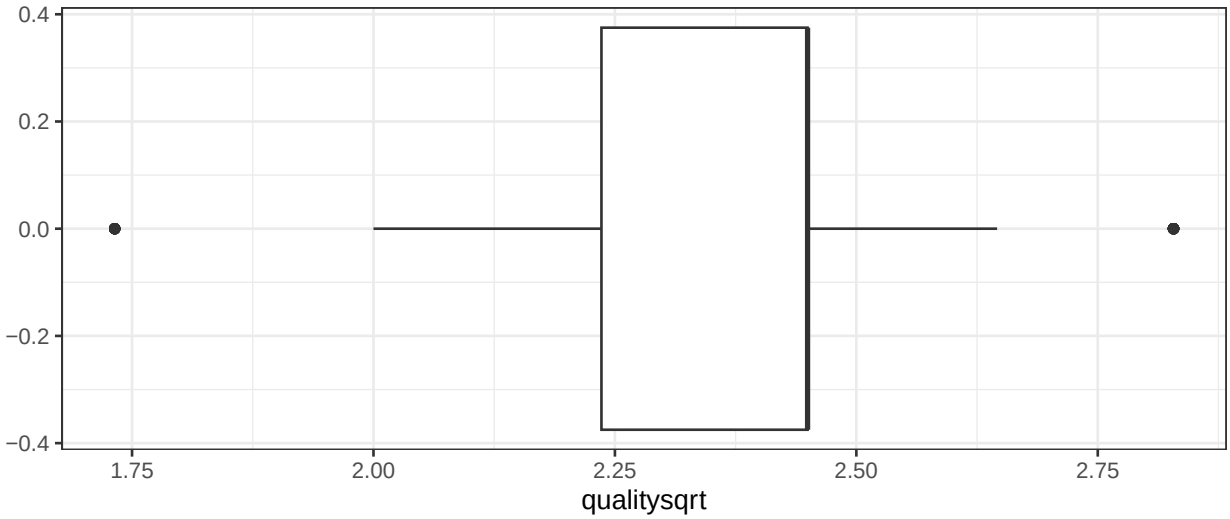
```
## [1] 0.7580388
```

```
kurtosis(wines2sqrt$alcoholsqrt)
```

```
## [1] 2.93023
```

Quality

```
wines2sqrt %>% ggplot(aes(qualitysqrt))+
  geom_boxplot()+
  theme_bw()
```




```
skewness(wines2sqrt$qualitysqrt)
```

```
## [1] -0.05011032
```

```
kurtosis(wines2sqrt$qualitysqrt)
```

```
## [1] 3.578785
```

```
cor.test(x=wines2$volatile.acidity, y=wines2$quality, method="pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  wines2$volatile.acidity and wines2$quality
## t = -16.954, df = 1597, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.4313210 -0.3482032
## sample estimates:
##        cor
## -0.3905578
```

```
cor.test(x=wines2$alcohol, y=wines2$quality, method="pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  wines2$alcohol and wines2$quality
## t = 21.639, df = 1597, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4373540 0.5132081
## sample estimates:
##       cor
## 0.4761663
```

After squared root transformation I can detect what follows: 1. quality skeness has significantly reduced (from 0.217 to -0.05), 2. quality kurtosis has slightly increased (from 3.29 to 3.58), 3. correlation between quality and both alcohol and volatile acidity has not changed (0.476 and -0.391 respectively).

2. MODELING

Pre-processing

So far, data have already been transformed (via squared root transformation), the number of features have already been reduced and outliers have been kept for the reasons explained above. For these reasons, EDA has already incorporated a pre-processing phase that at this point I assume I can take for done.

Modeling

Data partitioning

The starting point of modeling activity is the dataset called wines2sqrt, which contains both outcomes (quality) and two features (alcohol and volatile acidity). I want to split the dataset in the two following items: 1. wineQuality, a vector containing quality marks. 2. wineFeatures, a matrix of two numeric features.

```
wineQuality <- wines2sqrt$qualitysqrt

wineFeatures <- wines2sqrt %>%
  dplyr::select(c(1:2)) %>%
  as.matrix()
```

After setting the seed to 1, I proceed with data partition splitting both wineQuality vector and wineFeatures matrix into a 20% test set and 80% train.

```
set.seed(1, sample.kind = "Rounding")

test_index <- createDataPartition(wineQuality, times = 1, p = 0.2, list = FALSE)

wineFeatureTest <- wineFeatures[test_index,]

wineQualityTest <- wineQuality[test_index]

wineFeatureTrain <- wineFeatures[-test_index,]

wineQualityTrain <- wineQuality[-test_index]
```

I check that the training and test wineQuality datasets have similar proportions of marks.

```
mean(wineQuality)
```

```
## [1] 2.367901
```

```
mean(wineQualityTest)
```

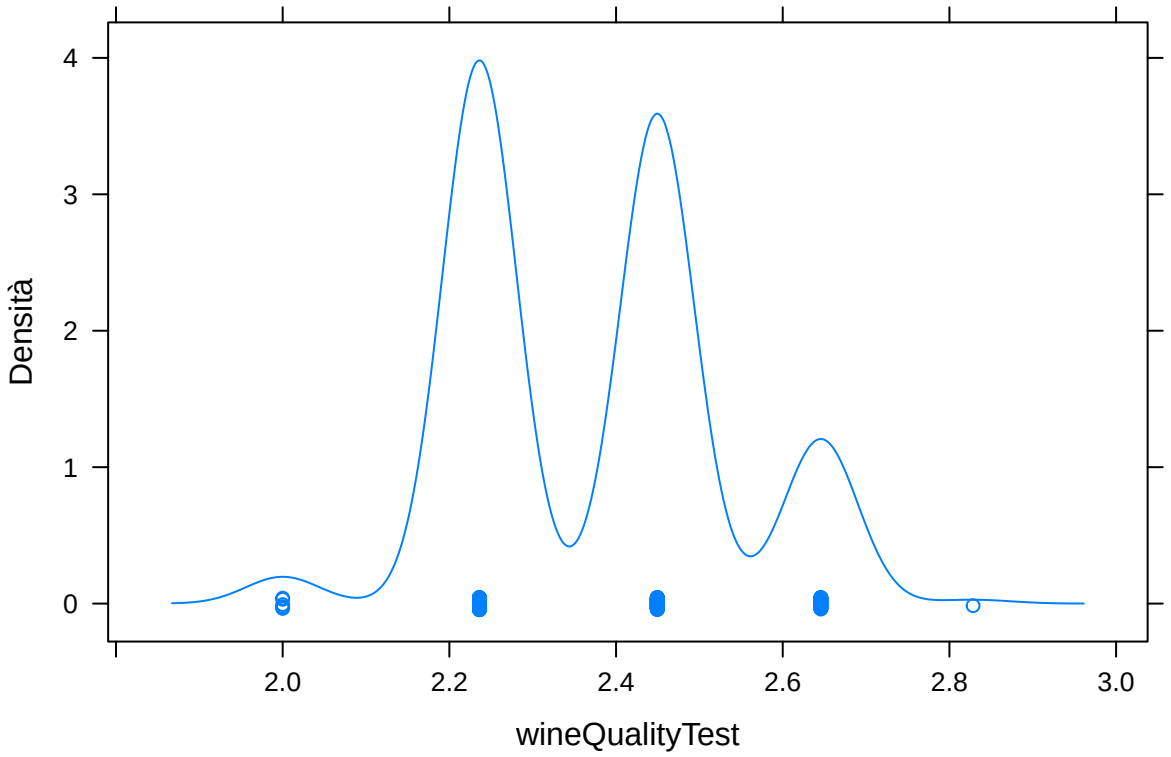
```
## [1] 2.372748
```

```
mean(wineQualityTrain)
```

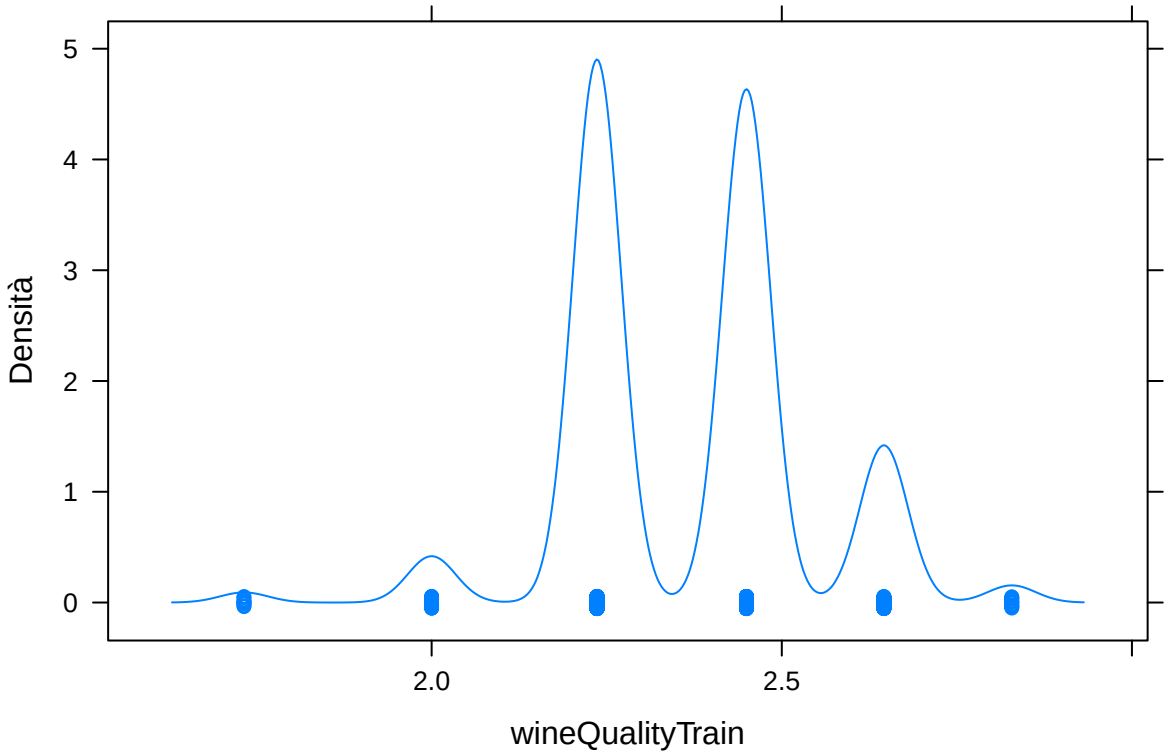
```
## [1] 2.366683
```

Everything seems fine: the average mark in the original, train and test tests is 2.37 in all of the three cases.

```
densityplot(wineQualityTest)
```




```
densityplot(wineQualityTrain)
```




```
wineQualityTest %>%
  as_tibble() %>%
  group_by(value) %>%
  summarize(numMarks=n()) %>%
  mutate(percMark=numMarks/sum(numMarks)*100)
```

```
## # A tibble: 5 x 3
##   value numMarks percMark
##   <dbl>    <int>    <dbl>
## 1  2           7    2.18
## 2  2.24      142   44.2
## 3  2.45      128   39.9
## 4  2.65       43   13.4
## 5  2.83        1    0.312
```

```
wineQualityTrain %>%
  as_tibble() %>%
  group_by(value) %>%
  summarize(numMarks=n()) %>%
  mutate(percMark=numMarks/sum(numMarks)*100)
```

```
## # A tibble: 6 x 3
##   value numMarks percMark
##   <dbl>    <int>    <dbl>
```

```
## 1  1.73       10    0.782
## 2  2          46    3.60
## 3  2.24      539   42.2
## 4  2.45      510   39.9
## 5  2.65      156   12.2
## 6  2.83       17    1.33
```

Distribution of test and train set is similar with highs and lows at the same points as it is visibile from the two plot above. The 2.24 value appears for the 42/44 per cent of the times; the 2.45 value for the 39.9 per cent of the times, the 2.65 value has 13.4/12.2 per cent of frequency. The marginal (transformed) marks (1.73, 2 and 2.83) show slightly meaningful differences but we can deduce that data partitioning has not produced unbalanced train and test dataset.

Training

I want to try the following algorithms, by creating a prediction model and evaluating their accuracy: linear regression, decision tree, random forest, XG boost and k-nearest neighbors. As said in introduction section, during the development of this work, I have set aside neural network algorithm due to its extreme slowness. Another choice I have made after some trials was to keep the default bootstrapping method which use has non brought any tangible advantage in terms of model performance.

Linear regression

```
train_lm <- train(x = wineFeatureTrain,
                  y = wineQualityTrain,
                  method = "lm")

train_lm
```

Decision Tree

```
train_d.tree <- train(x = wineFeatureTrain,
                y = wineQualityTrain,
                method = "rpart")

train_d.tree
```

Random forest model

```
train_rf <- train(x = wineFeatureTrain,
                  y = wineQualityTrain,
                  method = "ranger")

train_rf
```

XG Boost

```
train_xg.boost <- train(x = wineFeatureTrain,
                        y = wineQualityTrain,
                        method = "xgbTree")
train_xg.boost
```

K-Nearest Neighbor

```
train_knn <- train(x = wineFeatureTrain,
                   y = wineQualityTrain,
                   method = "knn")

train_knn
```

Testing: prediction and performance metrics

Always sticking to caret functions, I move to the test phase. For each model I will predict outcomes based on the features of test set and then, with postSample() command, I will get the most used KPI: RMSE, R squared and MAE which will be discussed in the following sections of this report.

Linear regression

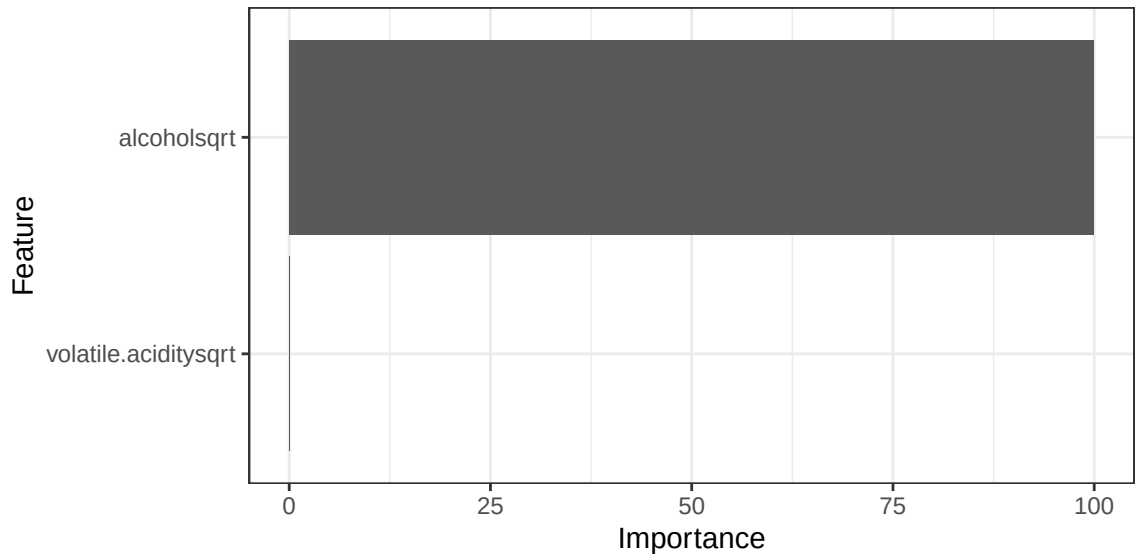
```
hat_lm <- predict(train_lm, wineFeatureTest)

Perf_lm <- postResample(pred=hat_lm, obs=wineQualityTest)

Perf_lm
```

```
##      RMSE  Rsquared       MAE
## 0.1372697 0.2315709 0.1110814
```

```
ggplot(varImp(train_lm))+
  theme_bw()
```



Decision Tree

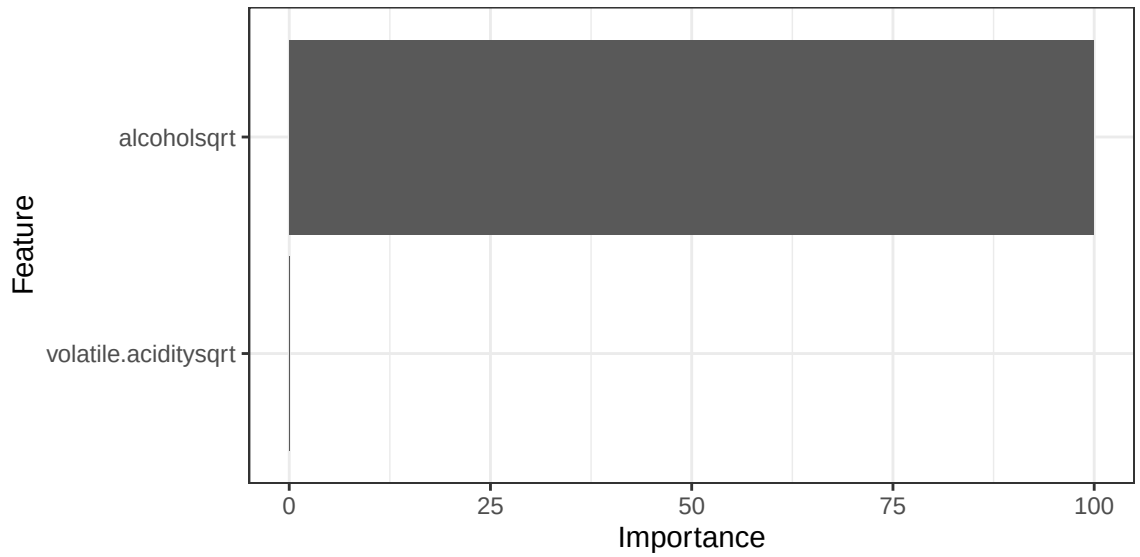
```
hat_d.tree <- predict(train_d.tree, wineFeatureTest)

Perf_d.tree <- postResample(pred=hat_d.tree, obs=wineQualityTest)

Perf_d.tree
```

```
##      RMSE  Rsquared       MAE
## 0.1454822 0.1545609 0.1187413
```

```
ggplot(varImp(train_d.tree))+
  theme_bw()
```



Random forest

```
hat_rf <- predict(train_rf, wineFeatureTest)

Perf_rf <- postResample(pred=hat_rf, obs=wineQualityTest)

Perf_rf
```

```
##      RMSE  Rsquared       MAE
## 0.1353865 0.2725238 0.1057372
```

XG Boost

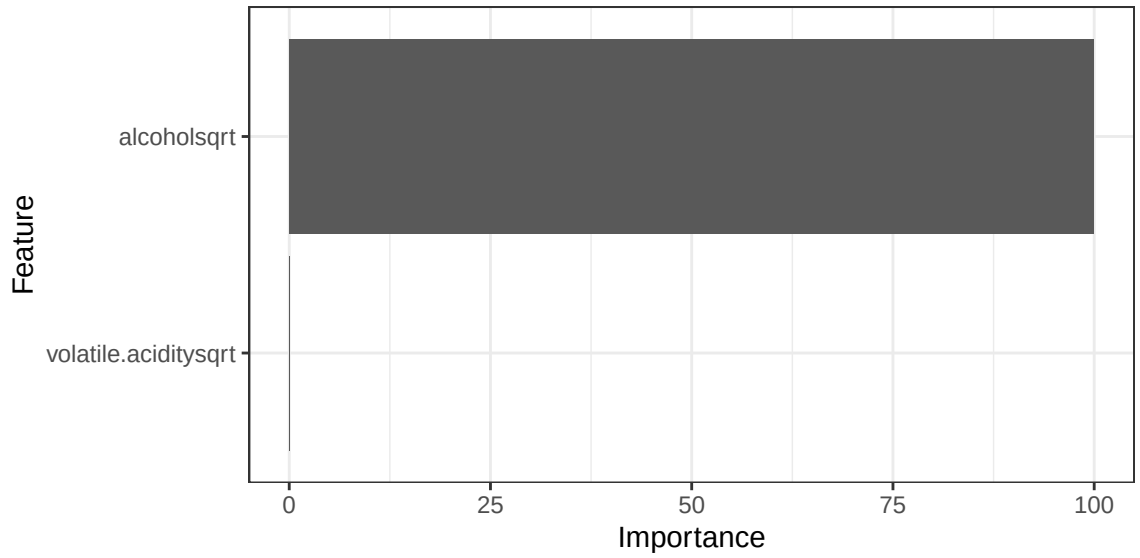
```
hat_xg.boost <- predict(train_xg.boost, wineFeatureTest)

Perf_xg.boost <- postResample(pred=hat_xg.boost, obs=wineQualityTest)

Perf_xg.boost
```

```
##      RMSE  Rsquared       MAE
## 0.1360305 0.2428273 0.1103146
```

```
ggplot(varImp(train_xg.boost))+
  theme_bw()
```



K-Nearest Neighbor

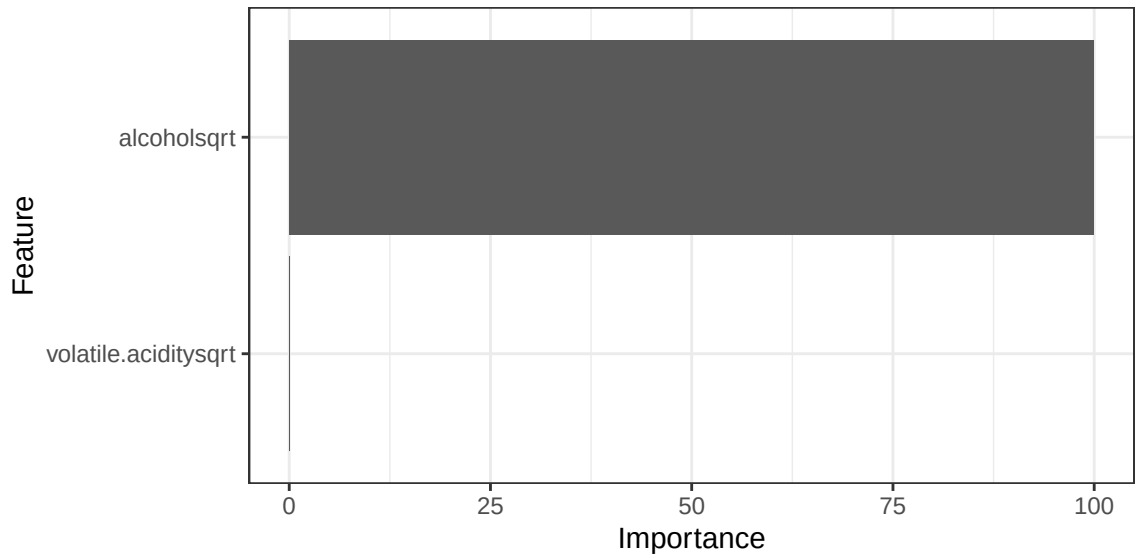
```
hat_knn <- predict(train_knn, wineFeatureTest)

Perf_knn <- postResample(pred=hat_knn, obs=wineQualityTest)

Perf_knn
```

```
##      RMSE  Rsquared       MAE
## 0.1405343 0.2185444 0.1116959
```

```
ggplot(varImp(train_knn))+
  theme_bw()
```



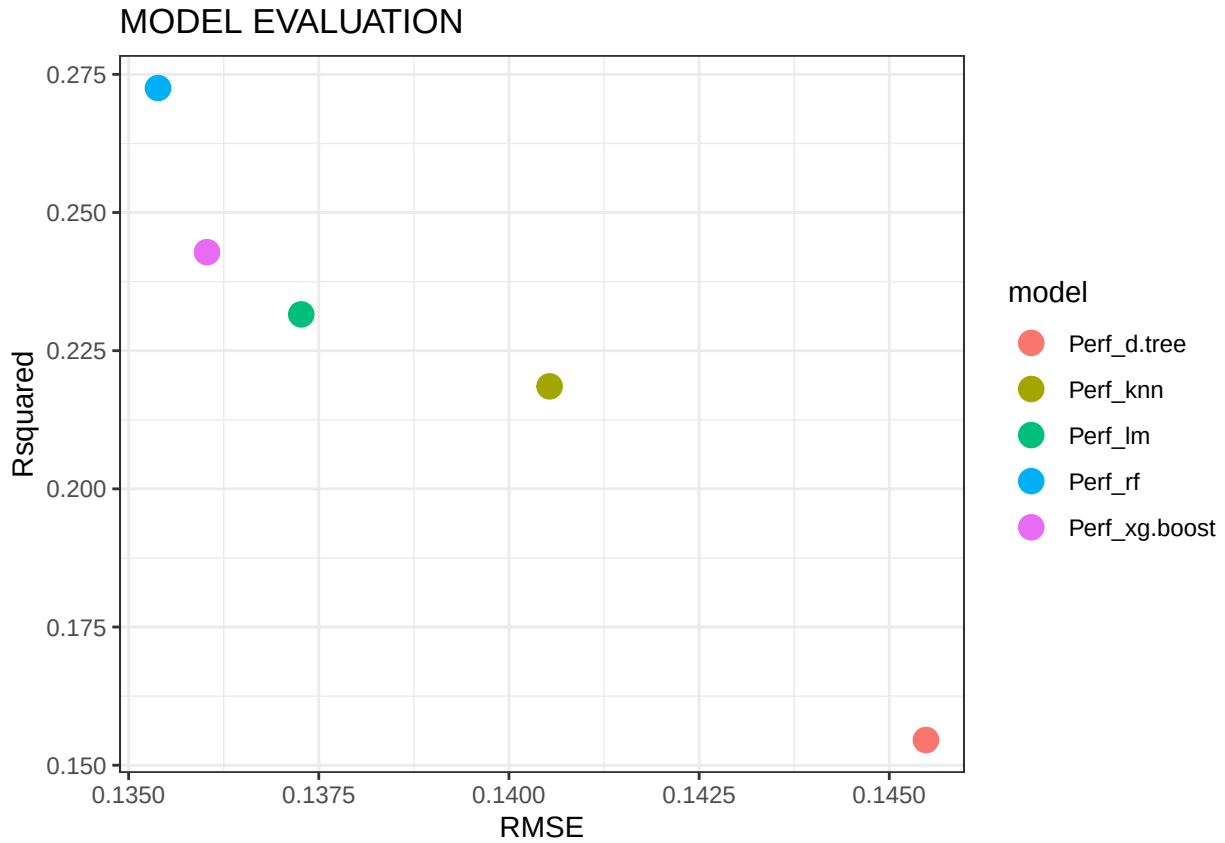
3. MODEL PERFORMANCES EVALUATION

The goal is to gather the performances of the models I have tried in order to find out which is more suitable to predict wine quality.

```
Perf_overview <- rbind(Perf_lm, Perf_d.tree, Perf_rf, Perf_xg.boost, Perf_knn)

Perf_overview2 <- Perf_overview %>%
  as_tibble %>%
  cbind(model = row.names(Perf_overview), .)

Perf_overview2 %>% ggplot(aes(x=RMSE, y=Rsquared, color=model))+
  geom_point(size=4)+
  labs(x = "RMSE", y = "Rsquared",
       title ="MODEL EVALUATION")+
  theme_bw()
```



As said in the introduction section, I am interested in understanding models performance using RMSE and R squared index. Even if provided by the postResample function of the caret package, I will not take into account the MAE since it provides a measure that I find "unconfortable" to interpet due to the fact that it is expressed in a different scale than the values of the observed entities. It may be useful to briefly recap the meaning of the two KPI I am going to use: - RMSE is a metric that tells us how far apart the predicted values are from the observed values in a dataset, on average. The lower the RMSE, the better a model fits a dataset. - R squared is a metric that tells us the proportion of the variance in the response variable of a regression model that can be explained by the predictor variables. This value ranges from 0 to 1. The higher the R squared value, the better a model fits a dataset.

As far as our models are concerned, we can visualize an inverse relation between RMSE and the R squared.

Generally speaking the level of R squared is not very satisfying: it remains pretty low closer to 0 and far from 1, the value that certify the best performance. On the other hand, it seems that the RMSE gives us a good feedback. Since it uses the same scale of the observed values and since the average quality rate (after transformation) of the entire dataset was 2.37, an RMSE of 0.145 (I use the worst to be safer) represents the 6.1% which can be considered quite a limited gap.

4 SUMMARY

Conclusion

The aim of the project was to understand (after getting a deeper knowledge of the dataset) if a regression was able to predict the quality of red wine according to some physical and chemical features and eventually which model could deliver best results. The main points that have arisen are the following:
1. Despite a large number of features provided in the original dataset, only very few of them show some kind of moderate correlation.
2. The machine learning regression task has been based on the question whether it was possible to create a model able to predict wine quality depending on some features, specifically volatile acidity and alcohol, which (in the EDA phase) seemed to be the most correlated variable to quality.
3. Data transformation was needed due to both a skewness of the distribution of the outcome (quality) and relevant differences in variables scale.
4. Different regression algorithms have been tried, trained and test; a few have been set aside because of the high amount of RAM needed (and, as a consequence, for the dramatic decrease of computational speed).
5. Generally speaking, every model used (linear regression, decision tree, random forest, xg boost and knn) have given back a good response in terms of RMSE, which means that predicted values seem to "fall" pretty close to the observed ones.
6. On the other hand, the R squared, that shows the proportion of the variance in the quality variable that can be explained by the two chosen features, is not satisfying with values closer to 0 rather than 1.

Limitations and future work

1. Only a limited number of algorithms has been trained and tested and varImp() function has been used (apart from random forest model when an error I have not been able to fix occured) without a deep analysis of its feedback.
2. A potential improvement could be the creation of an ensamble model.
3. Finally, it could be interesting a switch of prospective in terms of project aim: from prediction to clustering. In this case, an unsupervised machine learning method would replace the supervised approach of the regression.

REFERENCES

- Dataset "Redwinequality" Kaggle **link**: https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009/code?datasetId=4458&language=R.
- P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis, "Modeling wine preferences by data mining from physicochemical properties". "Decision Support Systems", Elsevier (http://dx.doi.org/10.1016/j.dss.2009.05.016)
- Caret package documentation: https://cran.r-project.org/web/packages/caret/caret.pdf