Classification

Leo Artoni, Riccardo Paolantoni

19/01/2023

Dataset and data cleaning

The Wine quality dataset presents 4898 observations (each one representing a different wine) and their characteristics. In the summary below we can observe that they are all numerical variables representing different parameters indicating different chemical properties in each wine.

First of all, we checked for NA values and only considered the distinct values within the dataset (eliminating duplicates). There were 0 NA values and removing duplicates reduced the number of observations to 3961. After plotting the data for the different variables we noticed the presence of some outliers, which we have removed using remove_percentile_outlier.

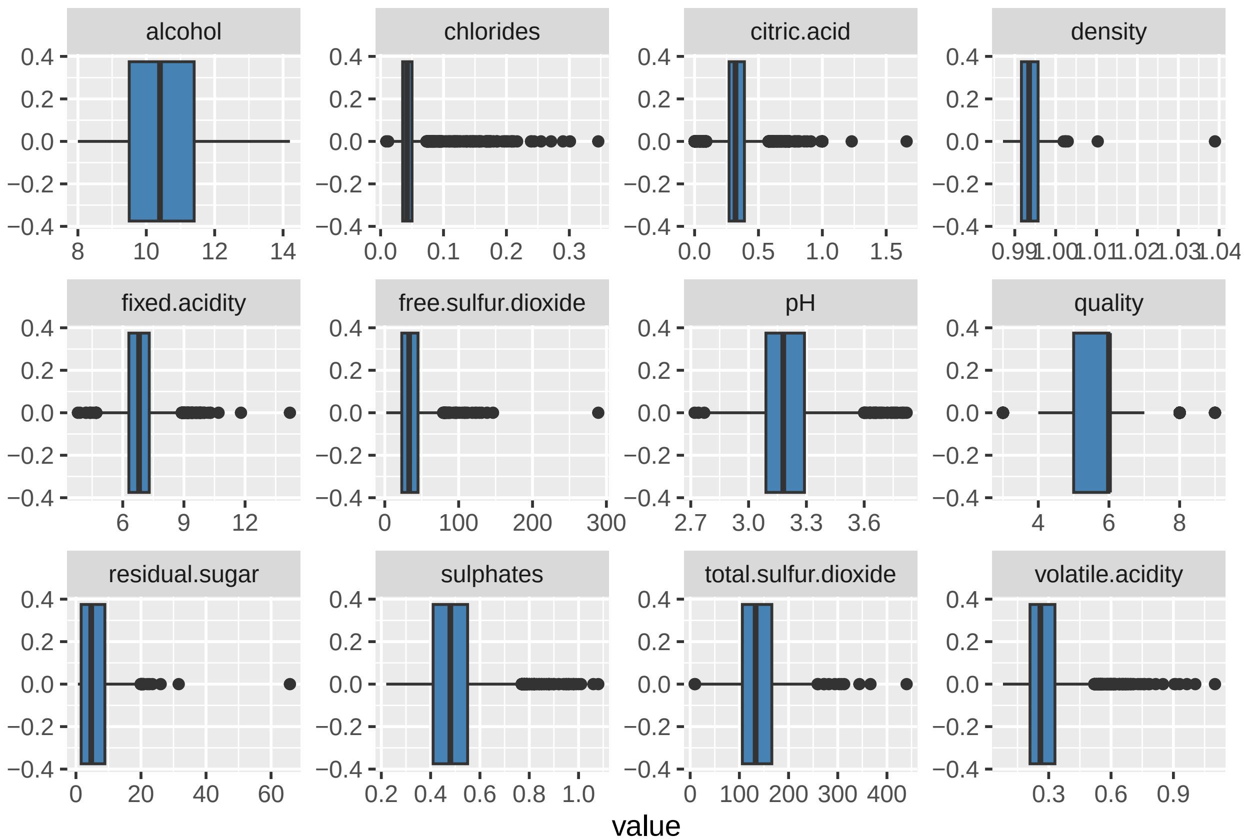
```
## 'data.frame':    4898 obs. of  12 variables:
##  $ fixed.acidity       : num  7 6.3 8.1 7.2 7.2 8.1 6.2 7 6.3 8.1 ...
##  $ volatile.acidity    : num  0.27 0.3 0.28 0.23 0.23 0.28 0.32 0.27 0.3 0.22 ...
##  $ citric.acid         : num  0.36 0.34 0.4 0.32 0.32 0.4 0.16 0.36 0.34 0.43 ...
##  $ residual.sugar      : num  20.7 1.6 6.9 8.5 8.5 6.9 7 20.7 1.6 1.5 ...
##  $ chlorides           : num  0.045 0.049 0.05 0.058 0.058 0.05 0.045 0.045 0.049 0.044 ...
##  $ free.sulfur.dioxide : num  45 14 30 47 47 30 30 45 14 28 ...
##  $ total.sulfur.dioxide: num  170 132 97 186 186 97 136 170 132 129 ...
##  $ density             : num  1.001 0.994 0.995 0.996 0.996 ...
##  $ pH                  : num  3 3.3 3.26 3.19 3.19 3.26 3.18 3 3.3 3.22 ...
##  $ sulphates           : num  0.45 0.49 0.44 0.4 0.4 0.44 0.47 0.45 0.49 0.45 ...
##  $ alcohol             : num  8.8 9.5 10.1 9.9 9.9 10.1 9.6 8.8 9.5 11 ...
##  $ quality             : int  6 6 6 6 6 6 6 6 6 6 ...
```

```
##  fixed.acidity    volatile.acidity  citric.acid     residual.sugar
##  Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600
##  1st Qu.: 6.300   1st Qu.:0.2100   1st Qu.:0.2700   1st Qu.: 1.700
##  Median : 6.800   Median :0.2600   Median :0.3200   Median : 5.200
##  Mean   : 6.855   Mean   :0.2782   Mean   :0.3342   Mean   : 6.391
##  3rd Qu.: 7.300   3rd Qu.:0.3200   3rd Qu.:0.3900   3rd Qu.: 9.900
##  Max.   :14.200   Max.   :1.1000   Max.   :1.6600   Max.   :65.800
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide    density
##  Min.   :0.00900   Min.   :  2.00      Min.   :  9.0        Min.   :0.9871
##  1st Qu.:0.03600   1st Qu.: 23.00      1st Qu.:108.0        1st Qu.:0.9917
##  Median :0.04300   Median : 34.00      Median :134.0        Median :0.9937
##  Mean   :0.04577   Mean   : 35.31      Mean   :138.4        Mean   :0.9940
##  3rd Qu.:0.05000   3rd Qu.: 46.00      3rd Qu.:167.0        3rd Qu.:0.9961
##  Max.   :0.34600   Max.   :289.00      Max.   :440.0        Max.   :1.0390
##        pH          sulphates         alcohol         quality
```

```
##  Min.   :2.720   Min.   :0.2200   Min.   : 8.00   Min.   :3.000
##  1st Qu.:3.090   1st Qu.:0.4100   1st Qu.: 9.50   1st Qu.:5.000
##  Median :3.180   Median :0.4700   Median :10.40   Median :6.000
##  Mean   :3.188   Mean   :0.4898   Mean   :10.51   Mean   :5.878
##  3rd Qu.:3.280   3rd Qu.:0.5500   3rd Qu.:11.40   3rd Qu.:6.000
##  Max.   :3.820   Max.   :1.0800   Max.   :14.20   Max.   :9.000
```

```
## [1] 0
```

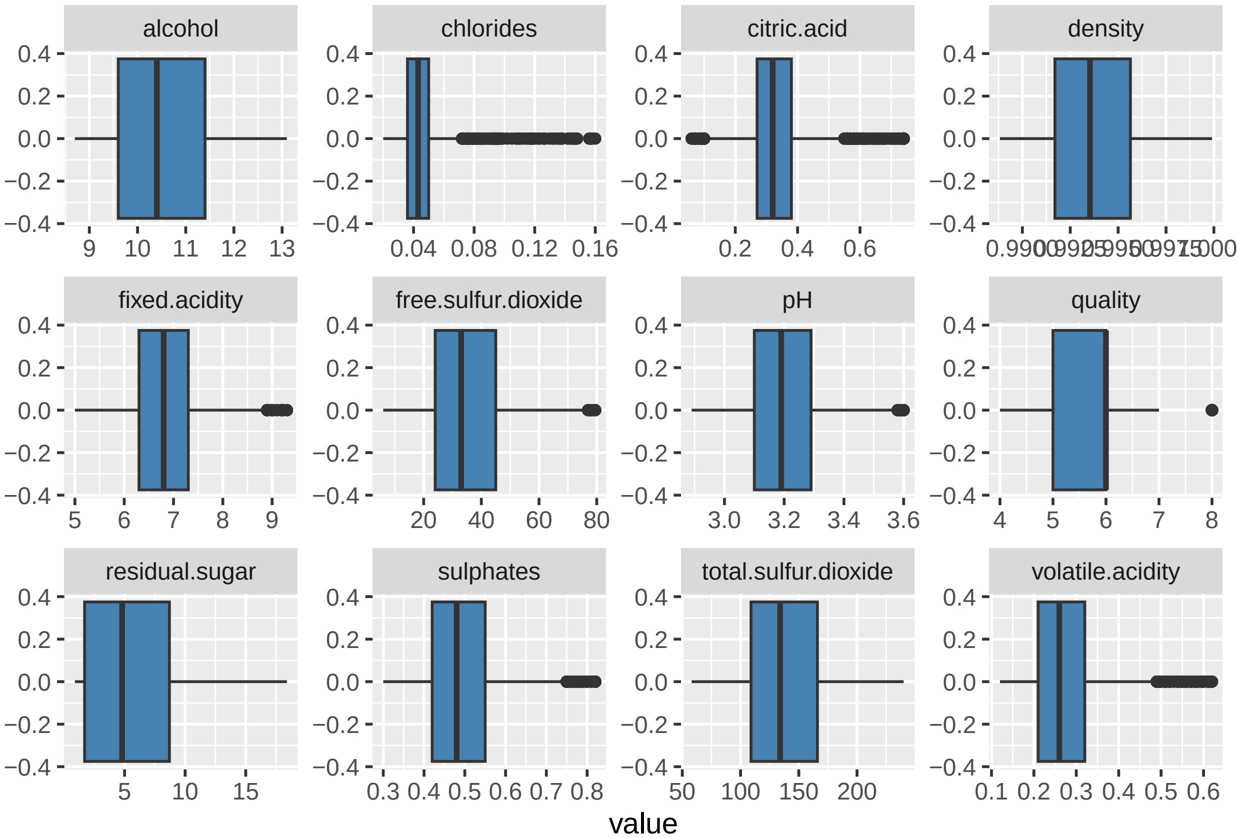
```
## [1] 937
```

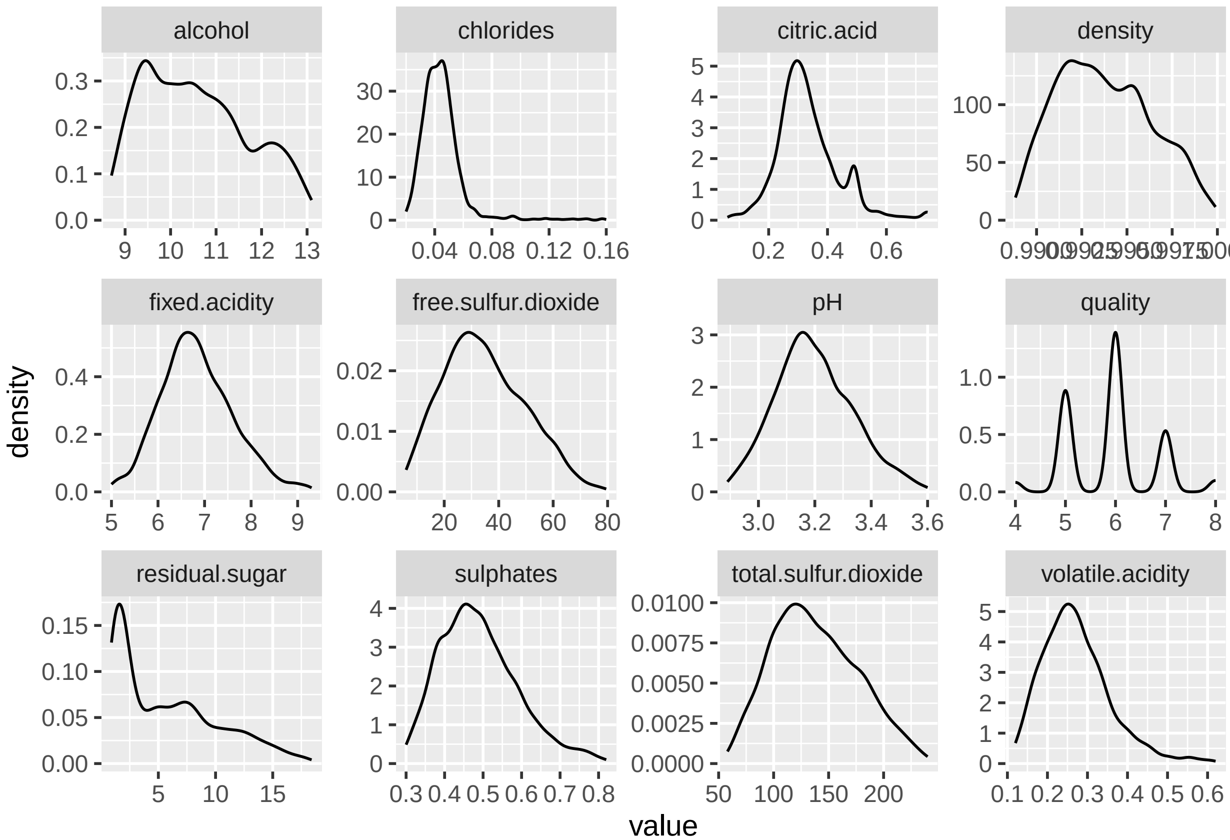


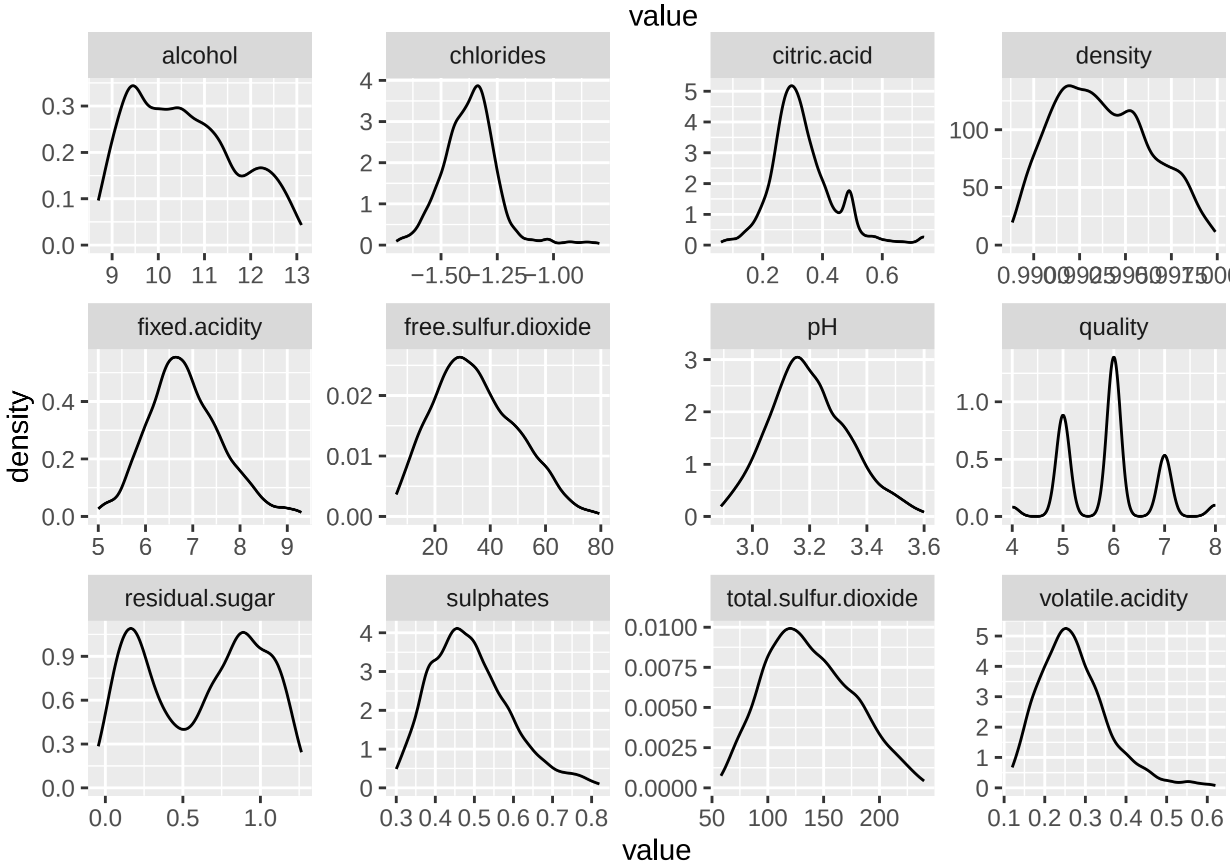

```
## [1] "remove_percentile_outlier: I start to filter categorical rare events"
## [1] "remove_percentile_outlier: dropped 67 row(s) that are rare event on fixed.acidity."
## [1] "remove_percentile_outlier: dropped 65 row(s) that are rare event on volatile.acidity."
## [1] "remove_percentile_outlier: dropped 56 row(s) that are rare event on citric.acid."
## [1] "remove_percentile_outlier: dropped 63 row(s) that are rare event on residual.sugar."
## [1] "remove_percentile_outlier: dropped 70 row(s) that are rare event on chlorides."
## [1] "remove_percentile_outlier: dropped 67 row(s) that are rare event on free.sulfur.dioxide."
## [1] "remove_percentile_outlier: dropped 68 row(s) that are rare event on total.sulfur.dioxide."
## [1] "remove_percentile_outlier: dropped 70 row(s) that are rare event on density."
## [1] "remove_percentile_outlier: dropped 58 row(s) that are rare event on pH."
## [1] "remove_percentile_outlier: dropped 62 row(s) that are rare event on sulphates."
## [1] "remove_percentile_outlier: dropped 52 row(s) that are rare event on alcohol."
## [1] "remove_percentile_outlier: dropped 10 row(s) that are rare event on quality."
## [1] "remove_percentile_outlier: 708 have been dropped. It took 0.01 seconds. "
```

Data Visualization

After removing the outliers we tried to understand the distribution of the dataset. We can see that the variables are normally distributed meaning that the set has been properly cleaned and is ready for the analysis.





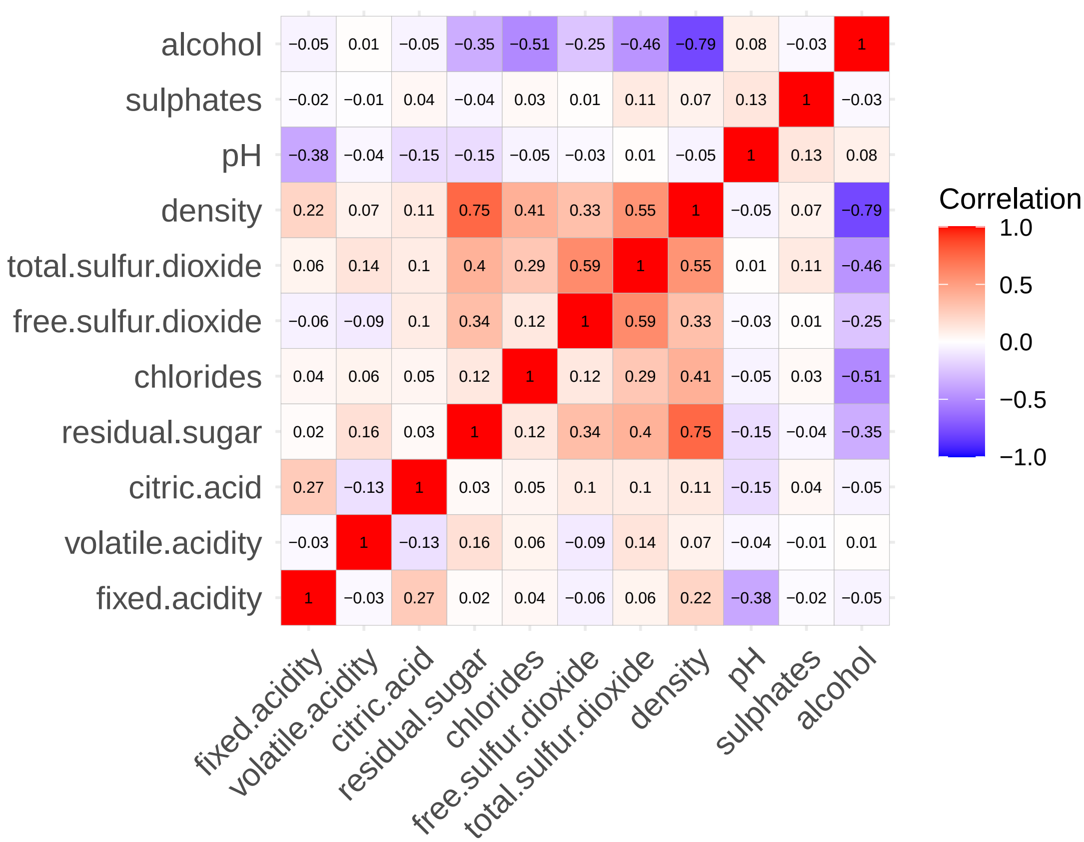


As one can see, some of the variables distributions are right-skewed (specifically, *residual.sugar* and *chlorides*), this is why we performed a log-transformation. This allowed to improve the distribution for *chlorides*, but not for *residual.sugar* (which we have decided to remove).

Classification

We then prepared the classification models. Our objective was to classify wines according to their characteristics. By using the variable *quality* we divided the wines into two categories: bad wines (score < 6) and good wines ($>= 6$). This transformation enabled us to transform the target variable into a bi-valued variable.

```
## quality
##  Bad Good
## 1052 2201
```




```
## Warning: 'funs()' was deprecated in dplyr 0.8.0.
## i Please use a list of either functions or lambdas:
##
## # Simple named list: list(mean = mean, median = median)
##
## # Auto named with 'tibble::lst()': tibble::lst(mean, median)
##
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
```

Next, we studied the correlation between the different variables, with the use of a correlation matrix, and removed the highly correlated ones (both positively and negatively correlated). From said matrix we have observed a high positive correlation between *density*, *residual.sugar*, *total.sulfur.dioxide* and *free.sulfur.dioxide* and a high negative correlation between *alcohol*, *density*. To avoid redundancy we have decided to remove the *density*, *residual.sugar* and *total.sulfur.dioxide* variables. This left us with 9 variables, 8 of which are predictors.

In order to properly evaluate the accuracy of the models, we decided to use a validation set approach. This is useful to compare the their respective performances (70 percent of the complete dataset was used as a training set while the remaining 30 percent was used for the test set).

Logistic regression

As a first approach, we chose to perform a logistic regression. In the model we used all of the 8 predictors. To specify that we are fitting a logistic regression we have set the family parameter of the glm function to binomial. After computing all the probabilities of the response variable, we assigned a "Good" value to all observations above the threshold (which we set to 0.6 because of a prevalence of "Good" wines over "Bad" ones), all others were defined as "Bad".

```
##
## Call:
## glm(formula = quality ~ ., family = binomial, data = data.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5444  -0.9007   0.4641   0.7980   2.2732
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)           0.96387    0.05554  17.354  < 2e-16 ***
## fixed.acidity        -0.12855    0.05600  -2.296   0.0217 *
## volatile.acidity     -0.50218    0.05435  -9.240  < 2e-16 ***
## citric.acid           0.01786    0.05160   0.346   0.7292
## chlorides            -0.11902    0.05952  -2.000   0.0456 *
## free.sulfur.dioxide   0.25158    0.05337   4.714 2.43e-06 ***
## pH                    0.07952    0.05730   1.388   0.1652
## sulphates             0.13541    0.05412   2.502   0.0124 *
## alcohol               1.11266    0.07230  15.389  < 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: 2873.1  on 2276  degrees of freedom
## Residual deviance: 2308.9  on 2268  degrees of freedom
## AIC: 2326.9
##
## Number of Fisher Scoring iterations: 4
```

```
##
## glm.pred Bad Good
##     Bad  207  132
##     Good 104  533
```

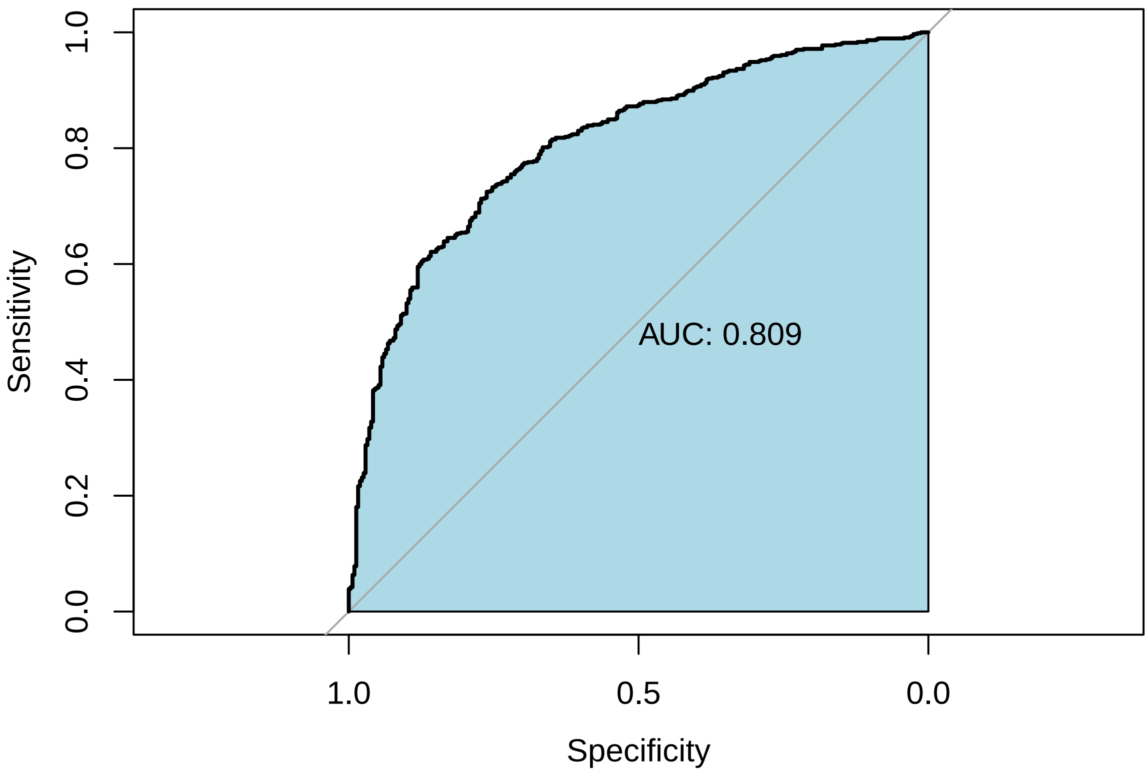
```
## [1] 0.7581967
```

As we can clearly see from the confusion matrix the model's accuracy is about 76%. It should also be noted that the summary describing the model also highlights the fact that *pH* and *citric.acid* have a high p-value, meaning they are not statistically significant (we also tried to fit a model excluding these variables, but this did not improve the results. For this reason we have decided not to include it in this analysis).

After the confusion matrix, we generated the ROC curve and the corresponding AUC.

```
## Setting levels: control = Bad, case = Good
```

```
## Setting direction: controls < cases
```



Lasso regression

Considering some of the variables do not appear to be highly significant, we decided to implement a Lasso regression. This model has the particularity of having a parameter called lambda, which impacts magnitude of the coefficients of the regression.

We started by converting the training set into a model matrix and obtained the optimal value of lambda (0.005) through cross-validation.

```
## [1] 0.00620809
```

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                               s0
## (Intercept)          0.78980735
## (Intercept)          .
```

```
## fixed.acidity       -0.08488134
## volatile.acidity    -0.39940369
## citric.acid          .
## chlorides           -0.53891089
## free.sulfur.dioxide  .
## pH                   0.11894913
## sulphates            0.06637211
```

```
##
## ytest  Bad Good
##   Bad  172  122
##   Good 139  543
```

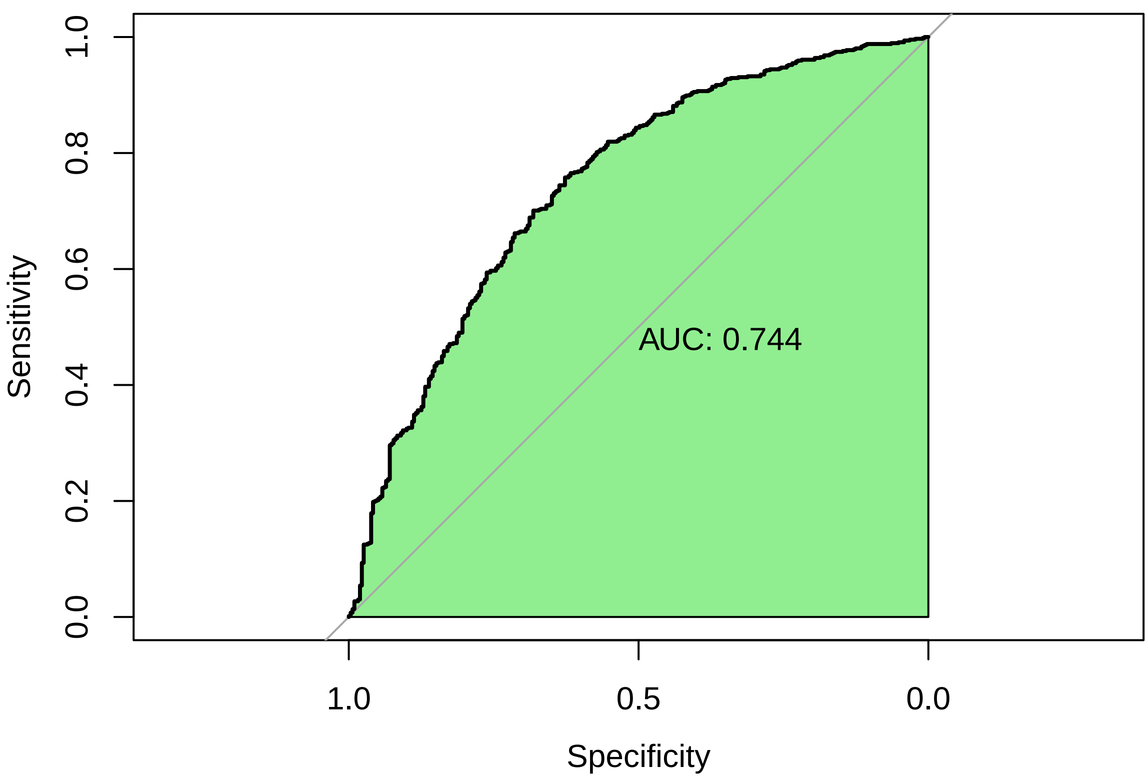
```
## [1] 0.732582
```

Following the lasso regression, the variables *citric.acid* and *free.sulfur.dioxide* were removed. Looking at the accuracy of this model (0.73) we observe that it is performing worse than the logistic regression.

```
## Setting levels: control = Bad, case = Good
```

```
## Warning in roc.default(response = data.valid$quality, predictor =
## probabilities, : Deprecated use a matrix as predictor. Unexpected results may be
## produced, please pass a numeric vector.
```

```
## Setting direction: controls < cases
```



Random forest

The third and final model is a Random forest which is an ensemble method. First of all we created a forest composed of 1000 trees. By default we use the square root of p (p = number of parameters) as the value of the mtry parameter when building a random forest of classification trees.

```
##
## Call:
##  randomForest(formula = quality ~ ., data = data.train, ntree = 1000,      mtry = 3, importance = T,
##                Type of random forest: classification
##                      Number of trees: 1000
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 22.97%
## Confusion matrix:
##      Bad Good class.error
## Bad  414  327   0.4412955
## Good 196 1340   0.1276042
```

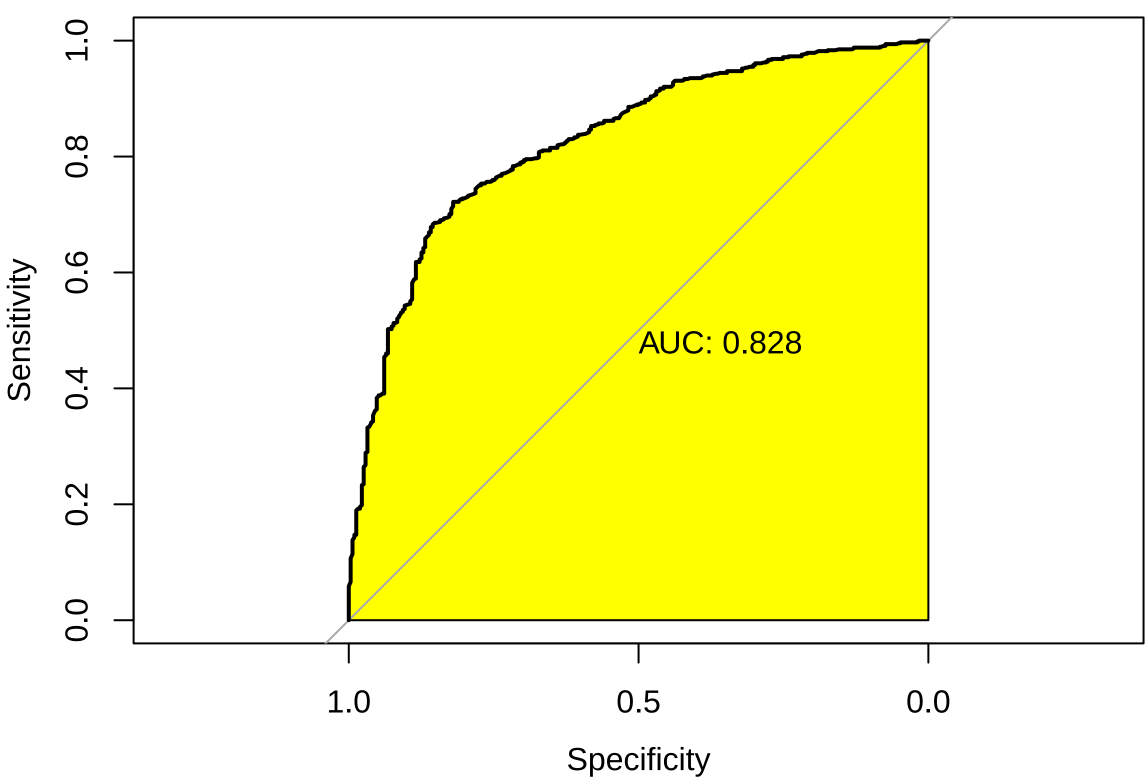
```
## [1] 0.7622951
```

We obtain an accuracy of approximately 77 percent, which is higher than the ones of the previous models (as we could have expected).

After the confusion matrix we generated the ROC curve and the corresponding AUC.

```
## Setting levels: control = Bad, case = Good
```

```
## Setting direction: controls > cases
```



Conclusions

We can conclude that the best results are obtained using the **Random Forest**. The most significant variables in order to determine the quality of a wine are the *volatile acidity* and the quantity of *chlorides*, this can be observed from the results of the Lasso regression.

Further research As an oter bit of food for thought, it would be interesting to perform statistical studies in order to determine which characteristics define whether a wine is red or white. This could be done by using classification methods. (#TeamRed or #TeamWhite)