Universitat Politècnica De Catalunya

Machine Learning Course

Wine Quality

Authors:
Raul Lorenzo Villagrasa
Ferran Noguera Vall
Alaa Cheaib

Supervisor:
Prof. Marta

June 2020

Contents

1 Brief description and goal of the work

The data used in this project was provided by the UCI Machine Learning Repository. Two datasets are included, containing information related to red and white vinho verde wine samples from the north of Portugal.

The main goal of this research is to model the wine according to the pshyco-chemical tests in order to help wine producers enhancing wine quality without having to spend money adding chemicals which do not improve their final product.

1.1 Data available

Number of Instances: Red wine - 1599; White wine - 4898.
Number of Attributes: 11 + output attribute.

Input variables (based on physico-chemical tests):

1. Fixed acidity
2. Volatile acidity
3. Citric acid
4. Residual sugar
5. Chlorides
6. Free sulfur dioxide
7. Total sulfur dioxide
8. Density
9. pH
10. Sulphates
11. Alcohol
12. Quality (score between 0 and 10 which is an output variable)

Note: Several of the attributes may be correlated, thus it makes sense to apply some sort of feature selection. After reading the data in the "csv" files and examining the data using the head function, we see that there are 2 types of variables, categorical and continuous.

2 Related previous work

Those are some of the previous work related to this kind of research. Some of them are from last year (2019) so this is a topic which is currently being explored.

P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.
Link: https://www.sciencedirect.com/science/article/pii/S0167923609001377?via%3Dihub

Hu, Gongzhu, et al. "Classification of wine quality with imbalanced data." 2016 IEEE International Conference on Industrial Technology (ICIT). IEEE, 2016.
Link: https://ieeexplore.ieee.org/abstract/document/7475021

RAJU, RINSON. "DATA ANALYTICS OF WINE QUALITY IN RSTUDIO." Journal of the Gujarat Research Society 21.14s (2019): 362-368.
Link: http://www.gujaratresearchsociety.in/index.php/JGRS/article/view/1375

Agyemang, Perpetual O. Modeling the preference of wine quality using logistic regression techniques based on physicochemical properties. Diss. Youngstown State University, 2010.
Link: https://etd.ohiolink.edu/pg_10?0::NO:10:P10_ACCESSION_NUM:ysu1298055470

Lingfeng, Zhang, Feng Feng, and Huang Heng. "Wine quality identification based on data mining research." 2017 12th International Conference on Computer Science and Education (ICCSE). IEEE, 2017.
Link: https://ieeexplore.ieee.org/abstract/document/8085517

3 Data exploration

3.1 Pre-processing

3.1.1 Combining the two datasets

We have two datasets, one for the red wine and the other for white wine. We have decided to combine those two data sets together and add a new column called "wine_type" to differentiate between red and white wine. In R, we called this dataset "AllWineData".

```
> summary(AllWineData)
 fixed.acidity    volatile.acidity  citric.acid     residual.sugar     chlorides       free.sulfur.dioxide
 Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600   Min.   :0.00900   Min.   :  1.00
 1st Qu.: 6.400   1st Qu.:0.2300   1st Qu.:0.2500   1st Qu.: 1.800   1st Qu.:0.03800   1st Qu.: 17.00
 Median : 7.000   Median :0.2900   Median :0.3100   Median : 3.000   Median :0.04700   Median : 29.00
 Mean   : 7.215   Mean   :0.3397   Mean   :0.3186   Mean   : 5.443   Mean   :0.05603   Mean   : 30.53
 3rd Qu.: 7.700   3rd Qu.:0.4000   3rd Qu.:0.3900   3rd Qu.: 8.100   3rd Qu.:0.06500   3rd Qu.: 41.00
 Max.   :15.900   Max.   :1.5800   Max.   :1.6600   Max.   :65.800   Max.   :0.61100   Max.   :289.00
 total.sulfur.dioxide    density             pH          sulphates         alcohol         quality      wine_type
 Min.   :  6.0        Min.   :0.9871   Min.   :2.720   Min.   :0.2200   Min.   : 8.00   Min.   :3.000   red  :1599
 1st Qu.: 77.0        1st Qu.:0.9923   1st Qu.:3.110   1st Qu.:0.4300   1st Qu.: 9.50   1st Qu.:5.000   white:4898
 Median :118.0        Median :0.9949   Median :3.210   Median :0.5100   Median :10.30   Median :6.000
 Mean   :115.7        Mean   :0.9947   Mean   :3.219   Mean   :0.5313   Mean   :10.49   Mean   :5.818
 3rd Qu.:156.0        3rd Qu.:0.9970   3rd Qu.:3.320   3rd Qu.:0.6000   3rd Qu.:11.30   3rd Qu.:6.000
 Max.   :440.0        Max.   :1.0390   Max.   :4.010   Max.   :2.0000   Max.   :14.90   Max.   :9.000
```

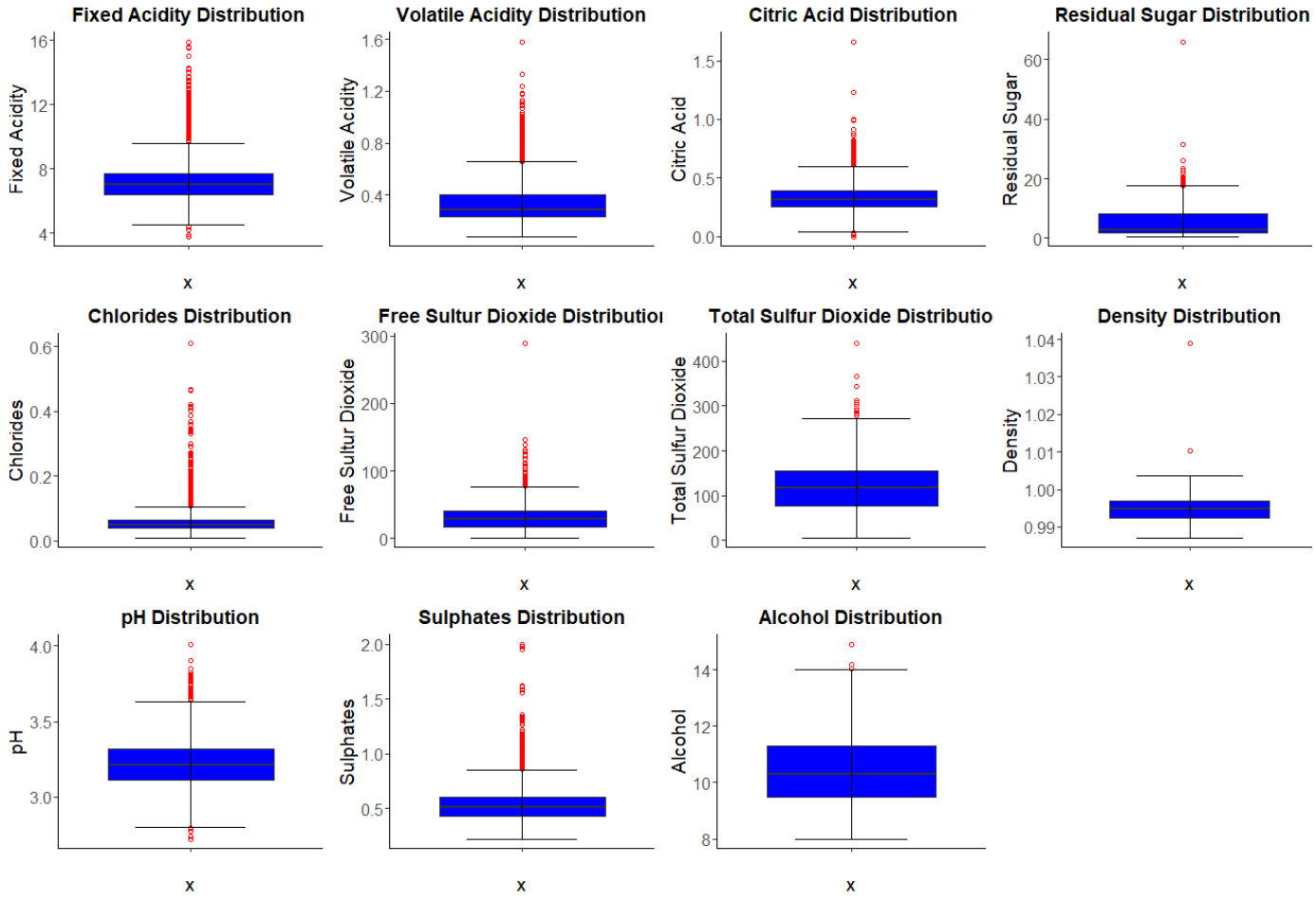
After combining both datasets another variable that could be extracted was the type of the wine, but this would only help us for the metadata rather than as a predictor this is why we decided to ignore it and not use it.

3.1.2 Missing Values

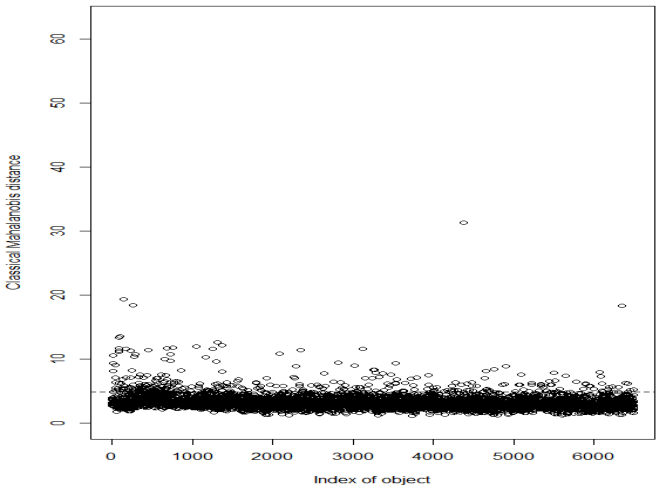
We applied functions to detect missing values on our dataset and turns out it does **NOT** have any hence no imputation methods for them are needed.

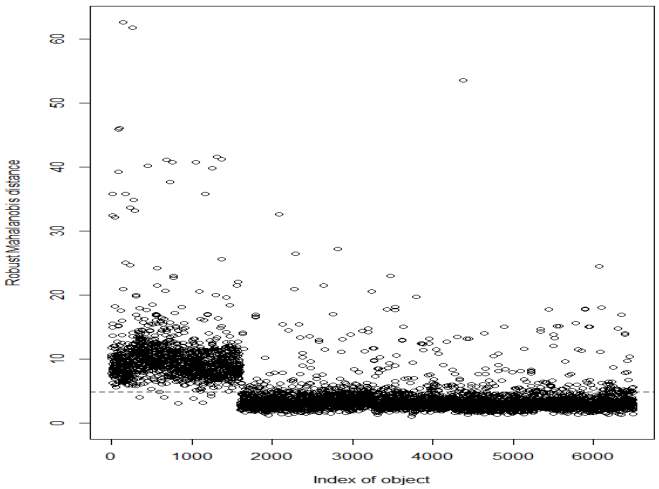
3.1.3 Outliers

Univariate outlier detection is a method used to detect outliers by using box plots. If a value exceeds the box plot boundary line, it is considered as an outlier. We observe that almost all of them contain some outliers, in some cases they contain a lot of them:

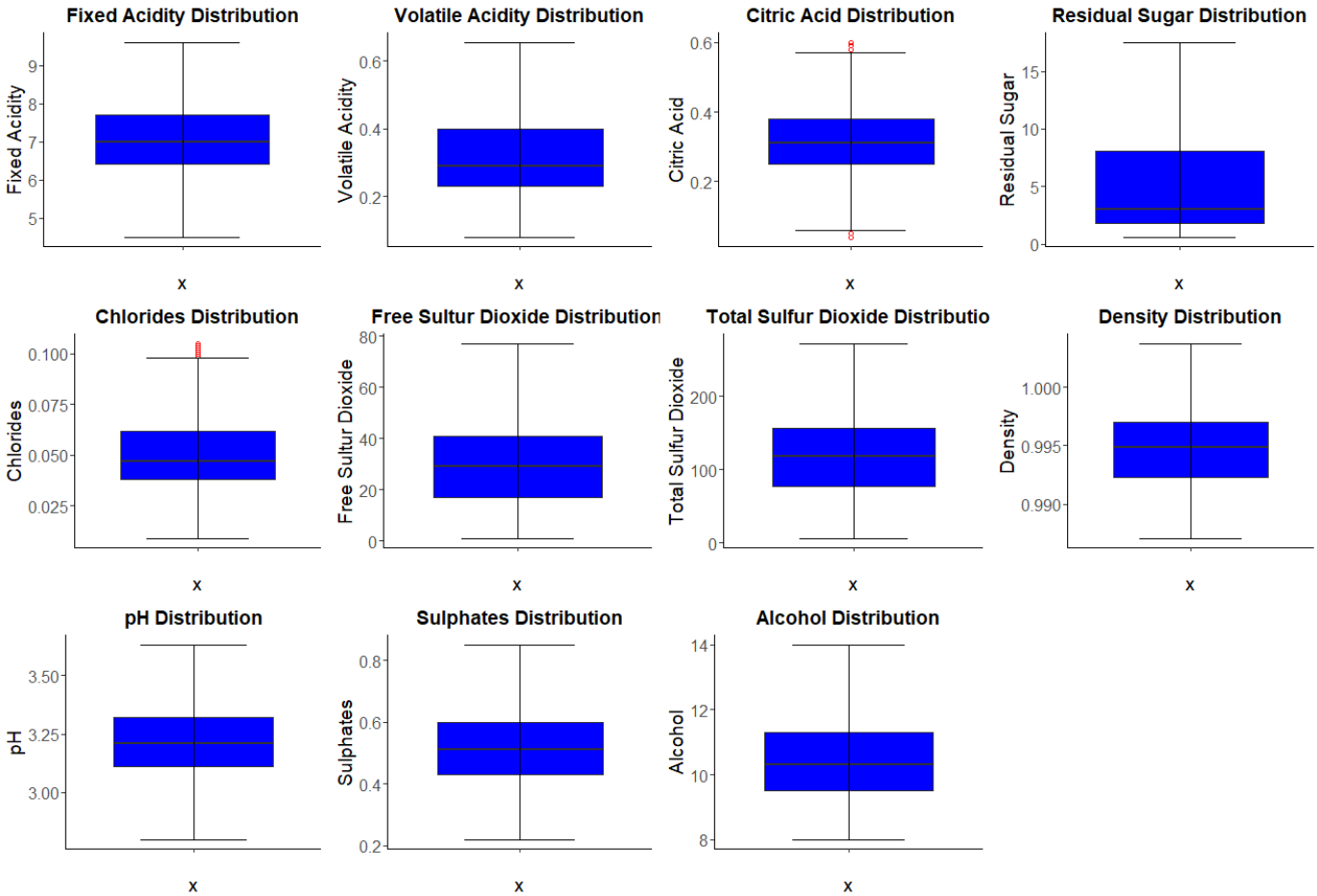


In **Multivariate Outlier Detection**, we used the Mahalanobis distance in comparison to the distribution of the whole data to see if there is an outlier. We can see how many points are considered outliers with each distance given a cutoff point with confidence level % 97.5.





There are several ways of treating outliers, the easiest and fastest is to simly delete them, this was not considered in our case because we would have lost a lot of information as outliers represent a great quantity in our dataset, taking this into consideration we decided to treat them by making all of them as NA and then imputing them with "missForest". After all is done, the following is the result:



You will notice that there are still a small amount of outliers in *Citric Acid* and *Chlorides*. Even if you apply "missForest" these outliers still exist so we decided to exclude them while doing the rest of the functions.

3.2 Feature selection/extraction

In this section, we used a method called "Boruta" which is a feature ranking and selection algorithm based on random forests algorithm. The advantage with Boruta is that it clearly decides if a variable is important or not and helps to select variables that are statistically significant. So according to this method, all the attributes are important for this analysis therefore, none of them will be excluded.

	meanImp	decision
total.sulfur.dioxide	67.09940	Confirmed
chlorides	60.08930	Confirmed
volatile.acidity	37.66625	Confirmed
sulphates	34.84242	Confirmed
density	33.51176	Confirmed
residual.sugar	27.70731	Confirmed
fixed.acidity	25.77956	Confirmed
pH	25.65097	Confirmed
free.sulfur.dioxide	24.67140	Confirmed
alcohol	22.61120	Confirmed
citric.acid	20.05181	Confirmed
quality	10.50988	Confirmed

3.3 Splitting of data into Train and Test Data

We have split the data into two sets: Training Data and Test Data. Our method splits 70% of the data selected randomly into training set and the remaining 30% sample into test data set.

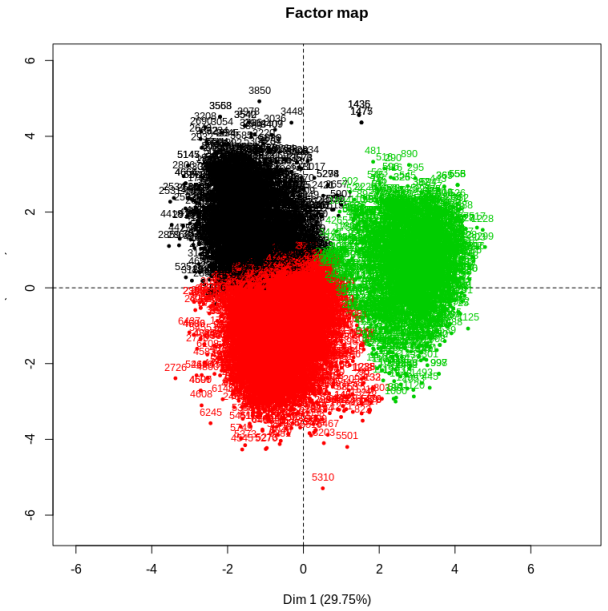
3.4 Visualization and Clustering

To perform the visualization and clustering methods we first performed a principal component analysis in order to reduce the dimensions of the dataset and being able to represent it on a 2 dimensional space.

After performing the PCA we plotted the individuals with the first two principal components and labeled each wine according to its quality group in order to see how well represented they are.

It can be seen on the image that normal and good wines are clearly splitted but bad and excellent ones aren't, this probably going to give us problems when trying make predictions for this last classes.

To perform the clustering we used the "HCPC" method from "FactomineR" library, the clustering obtained is the following:



The method automatically defines three clusters, probably one of the splits is due to the wine type whereas the other for normal and good wines, probably black and red wines according to the visualization picture.

4 Modeling methods and validation protocol

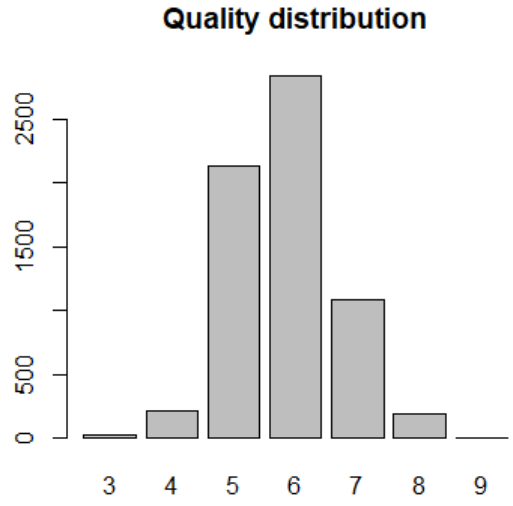
The main objective, as stated earlier, is to create a model to predict with its best parameterization, the quality of the wine. This will allow vineyards to save money and tie using taste testers to evaluate the wines' quality.
First, we are going to choose those variables which are important to determine the quality. Then, we will try to predict the quality using the variables below.

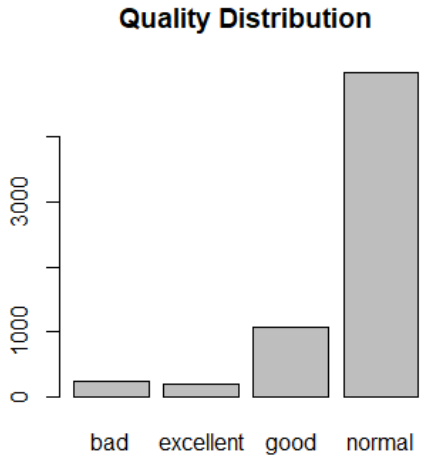
- fixed.acidity
- volatile.acidity
- citric.acid
- residual.sugar
- chlorides
- free.sulfur.dioxide
- total.sulfur.dioxide
- density
- pH
- sulphates
- alcohol

Throughout the course, we have studied some algorithms in-depth to perform models and predictions, we are going to compare these algorithms to build different models to make our predictions in order to get the best possible model for this dataset.
First, we take a look at the distribution of the wine qualities. As we can see, there are a lot of wines with a quality of 6 and 5 as compared to the others. The data set description states, there are a lot more normal wines than excellent or poor ones. For this purpose, we are going to classify the wines into bad, normal, good and excellent.

Punctuation criteria:

- Bad: quality < 5
- Normal: $5 \leq$ quality < 7
- Good: $7 \leq$ quality < 8
- Excellent: quality ≥ 8





We can't make sure that the model is going to have the preferred accuracy and variance, so we have to assure that the accuracy of the prediction of the model is legit. This assurance is called validation in which we validate our model. To evaluate the performance, we need to test it on some data that is not yet been seen and based on the performance of the model, we say if it's under-fitted, over-fitted or it's well-fitted. **Cross validation** is a approach that is used to evaluate the effectiveness of the model where the data is split into training/test. A known method is called **K-folds cross validation** which widely results in a less biased model in comparison to other methods. It's used for two main purposes, to tune hyper parameters and to better evaluate the performance of a model.

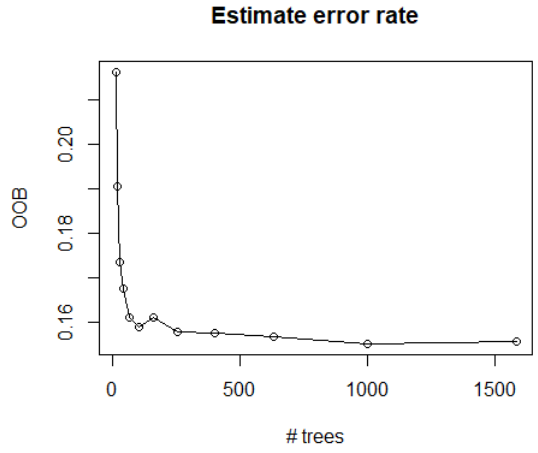
The number of folds (**K**) is usually determined by the number of instances contained in your dataset, which is 6497 individuals in our case. Larger K means less bias towards overestimating the true expected error; however, sometimes higher variance and higher running time. Also higher K give more samples to estimate a more accurate confidence interval on our estimate. Anyhow, our choice of **K is 10**. We split our data into 10 subsets, and we take one subset from the data and treat it as the validation set for the model. After that we keep the other 9 subsets for training the model, so that the average for all the 10 trails to get the effective readiness of the model and each subset out of the 10 subsets will be in the validation set at least once. This reduced the error induced by the bias and it also reduced the variance as each of the subsets is actually used in the validation as well.
We have decided to use the 70% of data for training and the 30% for testing.

4.1 Random Forest

The next step is to decide the parameters that we can control in order to get better results. To find out what is the best number of trees, we are going to fit different models with the same training data

set but changing the number of trees. Finally, with the obtained results, we will compare the OOB estimate of error rate.

Number of trees: 10, 16, 25, 40, 63, 100, 158, 251, 398, 631, 1000, 1585.



ntrees	OOB
10	0.216138328530259
16	0.190591338755771
25	0.173444713123763
40	0.167509342712684
63	0.160914486700374
100	0.158936029896681
158	0.161134315234117
251	0.157836887227962
398	0.157397230160475
631	0.156737744559244
1000	0.155198944823038
1585	0.155638601890525

Looking at the results, from 251 trees, we are getting a similar error, so we can conclude that any number of trees greater than 251 will be correct. As the speed is similar in all cases, we are going to choose the minimal OOB, in this case, 1000 trees.

Final parameterization:

- ntree = 1000
- No. of vars at each split=3
- proximity = TRUE
- importance = TRUE
- keep.forest = TRUE
- training data = 70 %
- testing data = 30 %

Once the model is done, we are going to check the importance of the variables. Basically, the graph below gives us the variables which are important for predicting the quality of the wine. Based on Gini plot, we can see that the alcohol is the most important variable, followed by density. All the other variables are approximately of the equal importance. So finally, we can conclude that alcohol and density will have an important role in order to predict the final score of the wine quality.

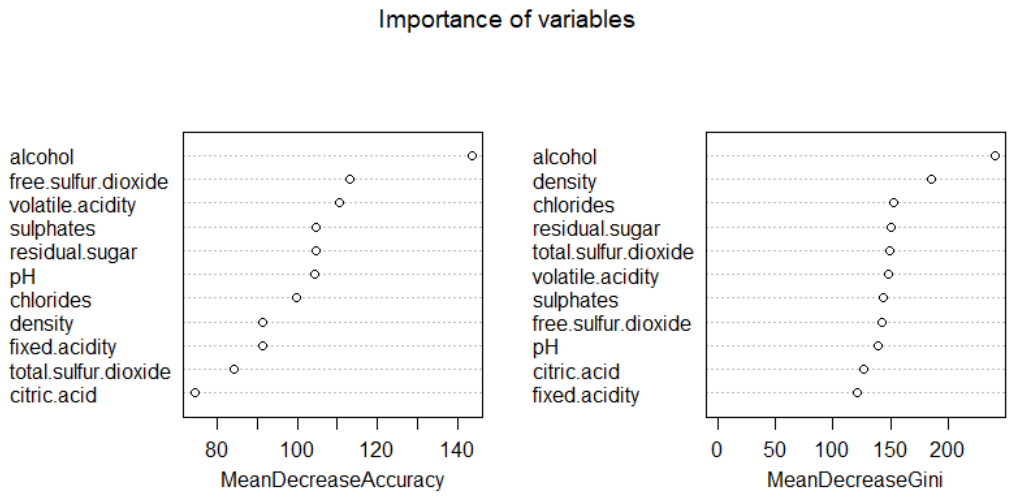


Figure 1: Random Forest - Importance of variables

In the graph below, it is interesting to see that total.sulfur.dioxide with density have been chosen a lot even though they are not as important as alcohol, as we see in the previous image.

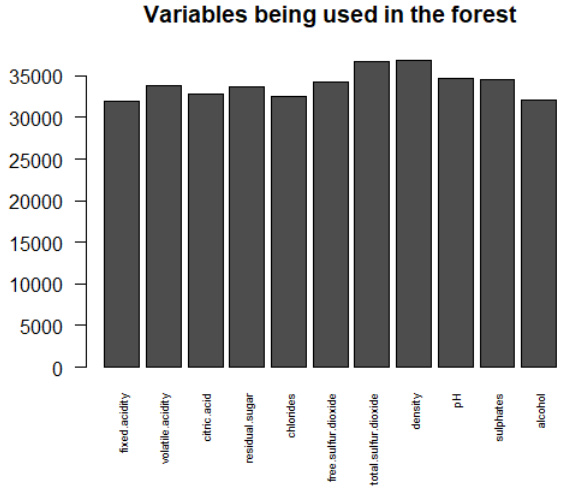


Figure 2: Total counts variables used Random forest

Now, to measure the relevance of our prediction, we are going to calculate the fraction of relevant instances among the retrieved instances (precision) and the fraction of total amount of relevant instances that were actually retrieved (recall). Also, we will calculate the general error and the accuracy of our predictor.

Final results:
- Accuracy: 84.14%
- Error: 15.86%

	bad	normal	good	excellent
precision	0.80	0.8505	0.7660	0.8947
recall	0.0533	0.9738	0.4970	0.3617

4.2 Multinomial Logistic Regression

First of all, we are going to decide which parameters are significance. So we are going to perform a Multinomial Logistic Regression with all the parameters as a important and after we will check the coefficients and we will perform a z-test. In this way, we will get the significance of them and we will be able to choose which ones are the best for our model.

```
          (Intercept) fixed.acidity volatile.acidity citric.acid residual.sugar chlorides free.sulfur.dioxide
excellent           0     0.5629552     2.561986e-09   0.3532274   2.057197e-05         0        1.357048e-11
good                0     0.1055158     0.000000e+00   0.5752074   1.474578e-04         0        2.686740e-14
normal              0     0.1305116     5.218338e-08   0.7453949   6.180883e-01         0        2.667067e-11
          total.sulfur.dioxide density          pH    sulphates      alcohol
excellent         2.800249e-03       0 0.159055440 9.660553e-06 0.000000e+00
good              2.377109e-06       0 0.254217425 1.568079e-12 0.000000e+00
normal            3.421548e-02       0 0.003943013 3.410622e-05 2.417819e-05
```

Figure 3: two tailed z-test scores (p-values)

From the image above, considering as a p-value reference 0.05, we have conclude that the following variables will be the most significance in order to perform a model to predict the quality of the wine..

- volatile.acidity
- residual.sugar
- free.sulfur.dioxide
- total.sulfur.dioxide
- density
- sulphates
- alcohol

In order to interpret the results from the model, we have chosen a reference value equal to *bad*. The reason for choosing this value as a reference is because we will want to know what variables to increment are giving more quality.

Once the model is fitted, we are going to calculate the relative risk, this parameter is the ratio of the probability of choosing one outcome category over the probability of choosing the baseline category. It can be obtained by the exponentiation of the coefficients from the model.

```
          (Intercept) volatile.acidity residual.sugar free.sulfur.dioxide total.sulfur.dioxide      density
excellent 1.971872e+00     0.0003976403       1.144119           1.084017           0.9890642 3.010541e-08
good      7.946801e+09     0.0007008972       1.083417           1.072457           0.9880640 9.754466e-16
normal    4.415359e-20     0.0400218202       1.005350           1.059028           0.9940140 1.676368e+19
          sulphates  alcohol
excellent 267.99860 3.713934
good      962.01149 2.787995
normal     61.01797 1.194180
```

Figure 4: Relative risk

Looking at the image above, we see that the relative risk ratio for a one-unit increase in the variables volatile.acidity, total.sulfur.dioxide and density are insignificant for being in excellent, good or normal vs. bad, here a value of 1 represents that there is no change. Otherwise, for a one-unit increase in the variable residual.sugar, free.sulfur.dioxide, sulphates and alcohol are very significant

for being in excellent, good or normal vs. bad. Here a value of 1 represents totally different quality of wine. Now, to measure the relevance of our prediction, we will calculate the general error and the accuracy of our predictor.

Final results:
- Accuracy: 77.93%
- Error: 22.07%

4.3 Neural Network

The neural network is very powerful and trending algorithm for classification tasks therefore it was decided to use it. Our particular artificial neural network would be composed by a single hidden layer.

First, we are going to decide the parameters that we can control in order to get better results. After that, we are going to fit different models with the same training data set but changing the hyperparameters decay and maxite, and the number of nodes for the hidden layer. Finally, with the obtained results, we will compare the performance.

In order to train our models, we're going to use a *grid* of model parameters and trains using a given re-sampling method, in this case, we'll be using 10x10 CV. All combinations are evaluated and the best one is chosen and used to construct a final model, which is refit using the whole training set.

We have chosen that the maximum number of iterations will be 1000.

Decay: 0.01, 0.01584893, 0.02511886, 0.03981072, 0.06309573, 0.10, 0.15848932, 0.25118864, 0.39810717, 0.63095734, 1.00

Nodes: 10, 20, 30, 40, 50, 60

The best tune for our model is given by a decay of 0.01 (the minimum) and a total of nodes of 60. Although the system gives us the best model with 60 nodes, we have chosen 30 for two main reasons. It takes too long in terms of time to fit the model and in terms of errors, we are getting the minimum error predicting the dataset used for training.
Now, to measure the relevance of our prediction, we will calculate the general error and the accuracy of our predictor.

Final results:
- Accuracy: 77.82%
- Error: 22.18%

4.4 Support Vector Machines

Finally, we decided to implement support vector machines (SVM) for the classification task. Different SVM models with different kernels and hyperparameters (cost and gamma) have been tested in order to extract the best model that represents our dataset. The cross-validation error has been used to state which is the best model.

The kernels tried have been: linear, quadratic, cubic and Gaussian RBF. In each of them several costs, hyperparameter that penalizes the miss-classifications, have been tested and, finally, different gamma.

Cost: 0.01 ,0.1 ,1 ,10 ,100 ,1000

Gamma: 0.125, 0.25, 0.5, 1, 2, 4, 8, 16

The best model found is quadratic with cost 0.1 and gamma 0.125.

Final results:
- Accuracy: 77.31%
- Error: 22.69%

5 Results obtained

In this section, we are going to compare and understand the results obtained during the modeling section.

Algorithm	Error	Accuracy
Random Forest	**15.86%**	**84.14%**
Multinomial Logistic Regression	22.07%	77.93%
Artificial Neural Network	22.18%	77.82%
Support Vector Machine	22.69%	77.31%

6 Final model chosen and an estimation of its generalization performance

Looking at the results, we see that the best algorithm for this dataset in order to predict the quality wine is Random Forest, with a 84.14% of accuracy. As we see, ANN is the one of the worst methods, this is because a Neural Network require much more data, in this project we got a defined sample of data, we are not working with a data stream. Although, Neural Network has a high visibility and is a trending algorithm used in machine learning, in this project doesn't make sense to use it because we are limited for the amount of data and the hardware where we are executing the code.

Support vector machine is the method delivering worst results, our assumptions are that some of the target variables are very close therefore they mostly fall into the margin created by the algorithm and it miss-classifies them.

ANN, SVM and Multinomial Logistic Regression are getting similar results. For this project, using a Multinomial Logistic Regression is not the best option, for mostly the same reason as SVM is not the best reason, because our dataset is not linearly separable, the assumption of linearity between the dependent variable and the independent variable.

Finally, we are getting good results with Random Forest because we don't have to define a hyperparameter like the others, this will increment the performance, as we have defined the optimal number of trees, we are avoiding over-fitting problems and finally, this algorithm has good ability to handle imbalanced data.

For this reasons, for this project the best algorithm is Random Forest with the following parametrization:

- ntree = 1000
- No. of vars at each split=3
- proximity = TRUE
- importance = TRUE
- keep.forest = TRUE

7 Scientific and personal conclusions

On a personal level, this research has helped us to understand different classification methods and how to apply them in R. It has been difficult to select what method to apply and how to tune the hyperparameters to increase their performance.

On a scientific level it can be observed that, as expected, chemicals effect the quality of the wine and it can also be extracted the properties that have a bigger impact. Excellent and bad wines are very difficult to predict, it can't be said for sure if this is because our distribution of the wine quality is underrepresented for this particular classes (which probably is) or that it is easier to predict normal wines but harder to tell if the quality ones.

8 Possible extensions and known limitations

One of the big limitations this research has is its dataset since most of the wines have a 5-6 quality therefore the bad, good and excellent wines are underrepresented hence it is very difficult to predict them wisely.

Taking the previous paragraph into consideration the immediate extension this project would need is to fed more extreme wines (bad, good and excellent) to the dataset or to reconsider the distribution of the values of the target. This would probably increase a lot the accuracy, specially feeding more data since methods that are not working as expected would perform much better.