Data Mining Final Project

Bruno Marović

January 2021

1 Data set overview and preprocessing

The data set used in this project is called the Wine Quality Data Set. The data we actually got is separated into 2 different data sets, each of them representing either white or red wine. Taking into the account the ease of use for the data set we decided to bind the two sets into one, while adding and additional attribute named *type*, which will then represent the color/type of the wine.

	fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide	total.sulfur.dioxide	density
1	7.4	0.70	0.00	1.9	0.076	11	34	0.9978
2	7.8	0.88	0.00	2.6	0.098	25	67	0.9968
3	7.8	0.76	0.04	2.3	0.092	15	54	0.9970
4	11.2	0.28	0.56	1.9	0.075	17	60	0.9980
5	7.4	0.70	0.00	1.9	0.076	11	34	0.9978
6	7.4	0.66	0.00	1.8	0.075	13	40	0.9978

	pH	sulphates	alcohol	quality	type
1	3.51	0.56	9.4	5	red
2	3.20	0.68	9.8	5	red
3	3.26	0.65	9.8	5	red
4	3.16	0.58	9.8	6	red
5	3.51	0.56	9.4	5	red
6	3.51	0.56	9.4	5	red

Figure 1: First 5 rows of the data set

From 1 we can see that the data set contains 6497 points of 13 attributes: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, quality and the type of the wine, which we added manually. We then created a plot which shows the correlations between every pair of attributes, as seen in 2. We can see that not many pairs have a high correlation between them, but we can isolate density and alcohol content, as well as the total and free sulfur dioxide concentration as the pairs that are the most positively correlated, and alcohol and density as the one which is the most negative.

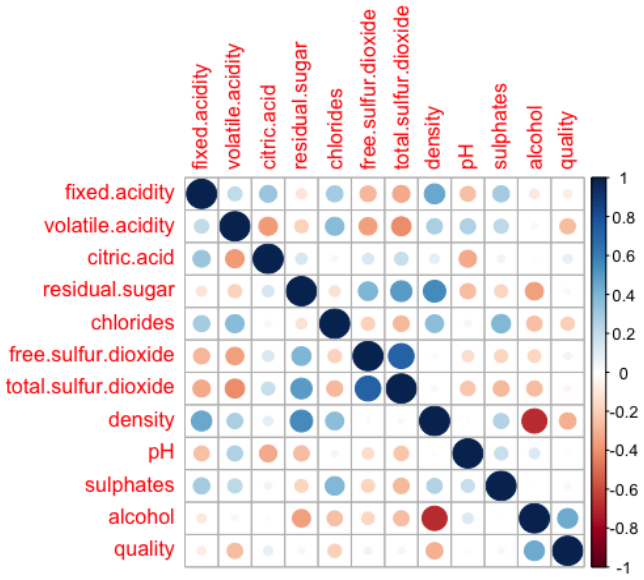


Figure 2: Correlation plot

We had to check if we have some missing values in the data set, which could eventually cause the models to be less accurate, and could also cause problems for some algorithms in the model training. Fortunately, we didn't have any missing values in the set.

As we have 13 different attributes, which can be a lot if we want to visualise it, and also if we want good algorithm performance we need to check if we can perform dimensionality reduction, because some of the attributes could prove to be more relevant than others. That's why we decided to apply principal component analysis on the data.

```
Importance of components:
                          PC1      PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10
Standard deviation     58.0698 11.98563 4.13097 1.32231 1.10726 0.69455 0.17520 0.14193 0.1209 0.1019
Proportion of Variance  0.9536  0.04062 0.00483 0.00049 0.00035 0.00014 0.00001 0.00001 0.0000 0.0000
Cumulative Proportion   0.9536  0.99418 0.99900 0.99950 0.99984 0.99998 0.99999 0.99999 1.0000 1.0000
                          PC11      PC12
Standard deviation     0.02786 0.0007505
Proportion of Variance 0.00000 0.0000000
Cumulative Proportion  1.00000 1.0000000
```

Figure 3: Summary of the PCA

We can see that the cumulative proportion of the variance of the first 2 principal components is 0.99418, which means that those two components explain 99,48% of the variance in the data. This can be visualized better by plotting a cumulative variance plot, as it can be seen in 4. There we can see that there is no point in taking more than 3 principal components because if we take more we will get a very small difference in the explained variance, but we have an additional attribute to process, which will certainly make algorithms slower. However, we won't be using the data set with the principal components in every part of our projects, as it will become obvious later in the report.

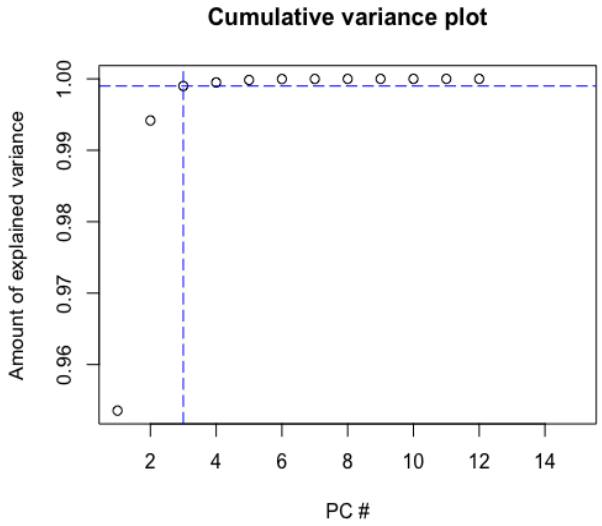


Figure 4: Summary of the PCA

2 Clustering

First problem we dealt with was the attempt to cluster the data to see if we can separate different kinds of wines in the set. As we are talking about a unsupervised approach we removed the wine type attribute from the data set so it doesn't provide unrealistically good results. Main goal of clustering is to find more information about the structure of the data set.

There are many method that can be used for clustering such as: representative based technique, hierarchical clustering techniques, grid and density based technique and others. However, not every technique is suitable for every problem, so we had to decide which of these is the best option. In the end we decided to use a representative based technique, k-medoids to be precise.

The reason we didn't choose k-means, which is usually the go-to algorithm for this kinds of problems is that unlike k-means, k-medoids chooses actual data points as centers (medoids or exemplars), and thereby allows for greater interpretability of the cluster centers than in k-means. And also because k-medoids minimizes a sum of pairwise dissimilarities instead of a sum of squared Euclidean distances, it is more robust to noise and outliers than k-means.

Once we had chosen the algorithm we were going to use for this problem, we needed to find the optimal hyper parameters, in this case that was the parameter k which determines how many medoids are we going to use, which also tells us how many clusters are we going to end up with.

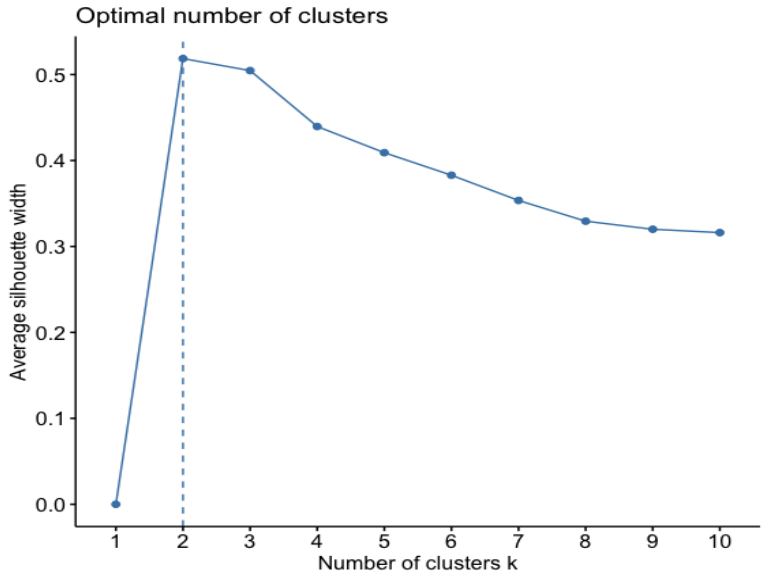
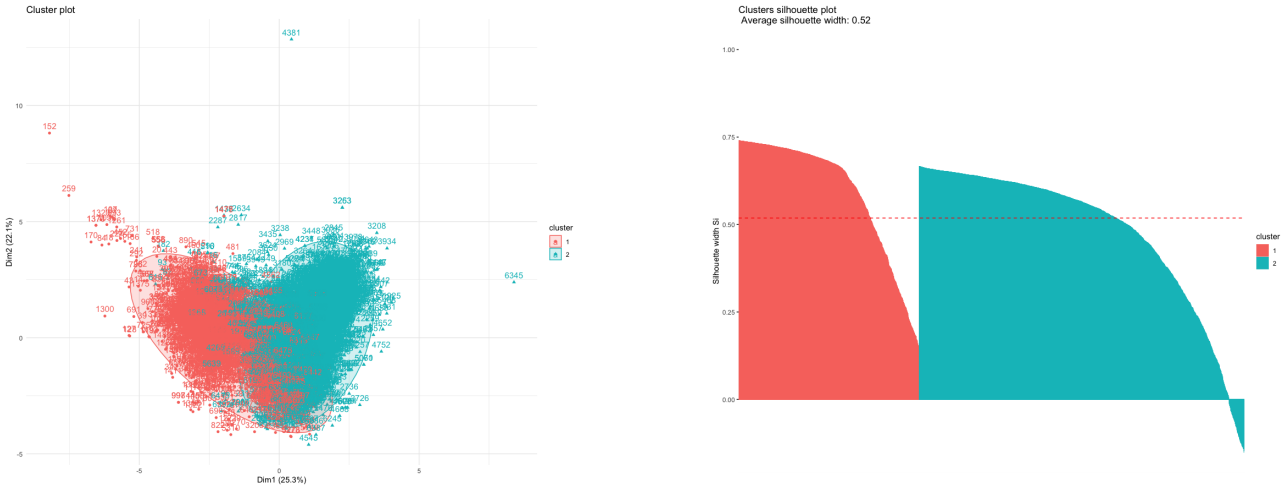


Figure 5: Choosing the optimal number of clusters

To do that we used the average silhouette width to determine which value of the parameter k has the largest value. In 5 it can be seen that, not counting k equal to 1, we have the largest average width k being 2, and after that we have a decline in the average width as we increase the value of the parameter k.

After we decided that we could start the algorithm because we had every parameter that we needed. The results that the algorithm gave us can be seen in 6.



(a) Cluster plot

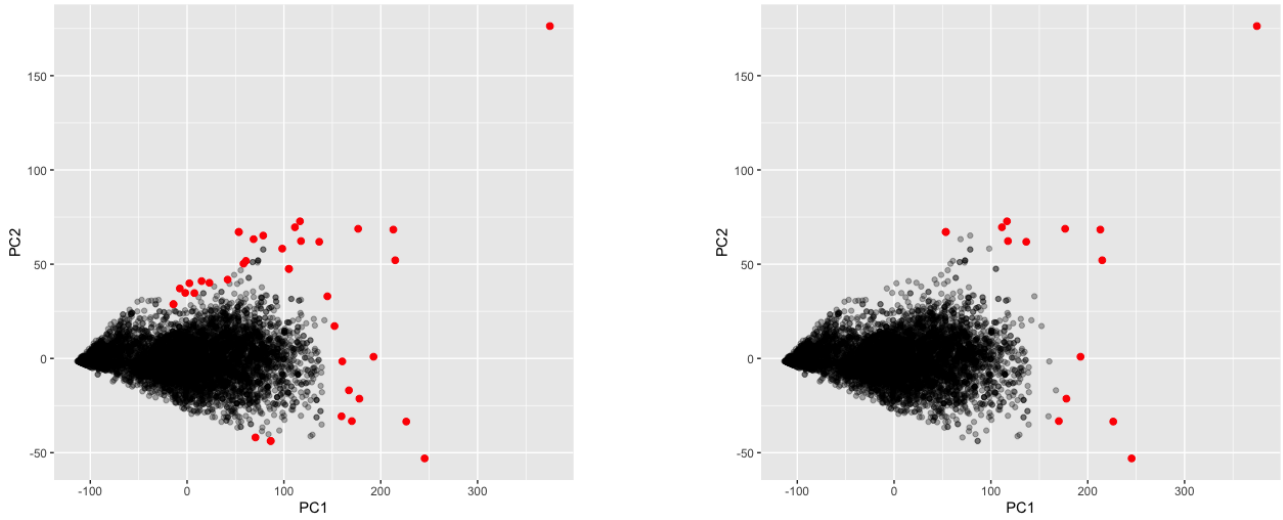
(b) Silhouette plot

Figure 6: Result of the PAM algorithm with the k parameter equal to 2

It can be seen from the silhouette plot that the results provide a solid clustering performance with only a fraction of the second class having a negative silhouette coeficient. The results seen in 6 are not done with a PCA reduced data set, but we tried to do it and the silhouette coefficients were almost identical.

3 Outlier detection

Next problem we faced was to try and implement detection for the outliers found in the data set. Outlier analysis can be seen as a complementary problem to clustering. As we had good enough results in our clustering problem we decided that it was worth it to try to find outliers in the data set. The method we decided to use was by using the local outlier factor (LOF). It is based on a concept of a local density, where locality is given by k nearest neighbors, whose distance is used to estimate the density. After we compute the local outlier factor for every data point, we can then set a threshold above which every data point LOF value will be classified as an outlier of the data set.



(a) LOF threshold at 1.8

(b) LOF threshold at 2.3

Figure 7: Result of the outlier detection by LOF thresholding

In this implementation we decided to use the PCA reduced data set because it explains 99% of the variability and makes it easier to spot a data point which deviates from the rest of the data. After we calculated the local outlier factor for every single point in the data set, we then tried to find a threshold, In 11 we see two attempts, with first being the threshold at 1.8 and the second one at 2.3. The the first threshold captures more points which may or may not be a real outlier, whereas the second one caputures the more seperated ones.

	fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide
1925	7.5	0.270	0.31	5.80	0.057	131.0
2259	6.8	0.290	0.16	1.40	0.038	122.5
3017	8.6	0.550	0.35	15.55	0.057	35.5
3531	7.1	0.490	0.22	2.00	0.047	146.5
3727	9.1	0.330	0.38	1.70	0.062	50.5
4254	6.9	0.400	0.22	5.95	0.081	76.0
4650	6.2	0.255	0.24	1.70	0.039	138.5
6345	6.1	0.260	0.25	2.90	0.047	289.0

	total.sulfur.dioxide	density	pH	sulphates	alcohol	quality
1925	313.0	0.99460	3.18	0.59	10.5	5
2259	234.5	0.99220	3.15	0.47	10.0	4
3017	366.5	1.00010	3.04	0.63	11.0	3
3531	307.5	0.99240	3.24	0.37	11.0	3
3727	344.0	0.99580	3.10	0.70	9.5	5
4254	303.0	0.99705	3.40	0.57	9.4	5
4650	272.0	0.99452	3.53	0.53	9.6	4
6345	440.0	0.99314	3.44	0.64	10.5	3

Figure 8: Top 8 outliers in the data set according to their LOF

In 8 we can see 8 data points with the highest local outlier factor in the data set. As we can see those points are all lower quality wines, with the lowest quality going down to 3 out of 10.

4 Classification

Problem with which we're dealing with in this section is classification of the wine. Classification is the process of finding or discovering a model or function which helps in separating the data into multiple categorical classes i.e. discrete values. In our case we are trying to classify the data points into one of two classes, and the classes are representing the type of wine, red or white. In this case we are talking about supervised learning algorithms, which means we will need our *type* attribute back in the data set.

First thing we had to do was generate a train and test data sets from the original data set. We used a 70-30 split, which means that we sampled 70% of the data for the train set, and 30% for the test set. After that we scaled the data to allow for a better algorithm performance.

4.1 Decision tree classifier

The first algorithm we used was decision tree learning, Recursive Partitioning and Regression Tree, to be more precise. Good thing about using the decision tree is that it provides a very good visualization which can show us the whole decision process, instead of having a black box like in some other models, as it can be seen in 9.

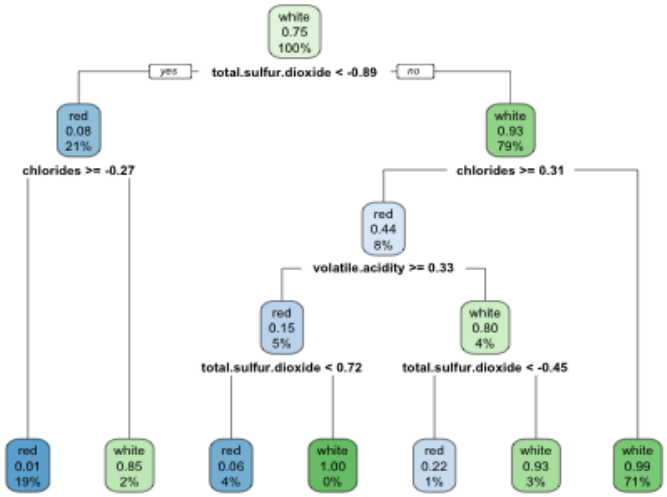


Figure 9: Model of the trained decision tree

4.2 Naïve Bayes classifier

The second algorithm we used was the so called Naïve Bayes classifier learning, which is an example of a probabilistic classifier based on applying Bayes' theorem with strong (naïve) independence assumptions between the features.

4.3 Logistic regression

The last algorithm we're using for classification is the logistic regression, it's a statistical model that in its basic form uses a logistic function to model a binary dependent variable. It is one of the most basic algorithms, but sometimes it can have better results than some more complex ones.

4.4 Results

	Decision tree	Naïve Bayes classifier	Logistic regression
Accuracy	98.59%	97.09%	99.59%

Table 1: Accuracy of the used methods

As we can see, all of the methods provide good results, but the logistic regression model has the best accuracy for this specific problem, but we might give the advantage to the decision tree classifier if we want a better understanding of how we got to that certain solution.

5 Regression

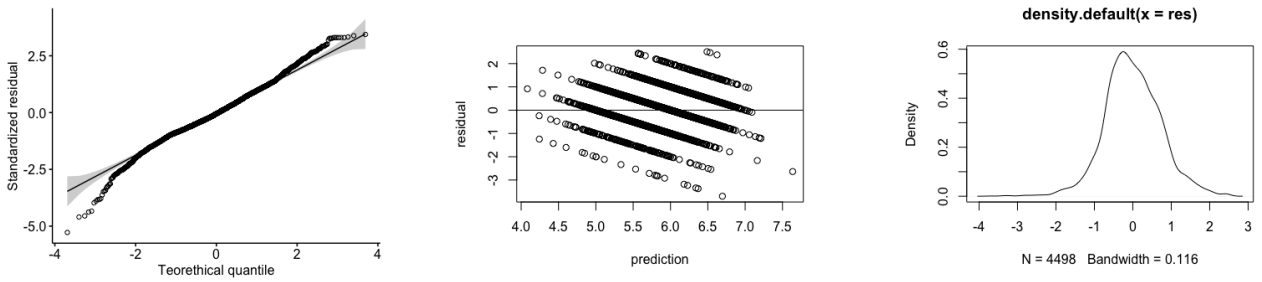
The last problem we are dealing with in this project is the problem of predicting the quality of the wine for the given attribute values. Process of building such a model is called regression analysis. We will be using the most common form of regression analysis called linear regression and since we have more than 1 explanatory variable it's called multiple linear regression. The goal is to find the coefficients of a linear function which minimises the sum of squared errors between the line and the data points.

First we generated a model using all attributes as explanatory variables. In 10 it can be seen that the p-values for the estimated parameters *citric.acid* and *chlorides* is greater than 0.05, which means they aren't statistically significant. So we tried constructing a model without using them, an we received more or less the same residual standard error: 0.7314, but we are using less attributes to predict the values. That residual standard error tells us that on average the prediction misses the real value 0.7314 quality points which are measured on a scale from 1 to 10. So taking that into consideration it's a pretty good predictor.

```
Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)           1.053e+02  1.636e+01   6.436 1.35e-10 ***
fixed.acidity         8.447e-02  1.865e-02   4.528 6.10e-06 ***
volatile.acidity     -1.475e+00  9.622e-02 -15.334  < 2e-16 ***
citric.acid          -1.411e-01  9.624e-02  -1.466    0.143
residual.sugar        6.372e-02  6.975e-03   9.136  < 2e-16 ***
chlorides            -6.348e-01  3.858e-01  -1.645    0.100 .
free.sulfur.dioxide   4.929e-03  9.159e-04   5.381 7.79e-08 ***
total.sulfur.dioxide -1.553e-03  3.828e-04  -4.059 5.02e-05 ***
density              -1.042e+02  1.660e+01  -6.278 3.75e-10 ***
pH                    4.332e-01  1.081e-01   4.007 6.25e-05 ***
sulphates             7.589e-01  8.950e-02   8.479  < 2e-16 ***
alcohol               2.160e-01  2.103e-02  10.271  < 2e-16 ***
classwhite           -3.215e-01  6.592e-02  -4.878 1.11e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 10: Summary of the coefficients of the model

After training the model we wanted to analyse the residuals. The first thing we did was that we made a quantile-quantile plot to see if the residuals have a normal distribution. By the looks of it the residuals have approximately normal distribution. We also made the residuals plot and the density plot to show how they are distributed. As it can be seen the residuals appear slightly ordered and not random, because they look like they are in stripes. The reason for that is that the *quality* attribute is originally discrete, but we decided to make a linear regression model which is continuous.



(a) Quantile - Quantile plot (b) Residuals plot (c) Density plot

Figure 11: Residuals analysis

6 Sources

- Charu C. Aggarwal, Data Mining: The Textbook, Springer, May 2015
- Sven Nomm, lectures from the course Data Mining, Department of Software Science, Tallinn University of Technology, 2020
- Paulo Cortez, University of Minho, Guimarães, Portugal, http://www3.dsi.uminho.pt/pcortez A. Cerdeira, F. Almeida, T. Matos and J. Reis, Viticulture Commission of the Vinho Verde Region(CVRVV), Porto, Portugal @2009
- Probability and Statistics for Engineers and Scientists, 9th Edition, Walpole, Myers, Myers, Ye