CENG 480

PROJECT REPORT

Çankaya University, Computer Engineering Department, Turkey

24 May 2020

201611047 İrem ÖZTÜRK

201711006 Mert BAYRAMUSTA

Abstract

Wine is an important part of human culture; a large segment sees winemaking as art. As there are many types of wine, there are many factors that affect the quality of the wine. In this paper, we conducted a study on the general factors affecting the quality of the Red Wine and how much they affect it. The dataset is considered with red wine samples. Three supervised learning methods were applied, under a computationally efficient procedure that performs simultaneous variable and model selection. The models that have been used were neural networks, decision trees, and kNN. These methods are useful for understanding how the quality of red wine is affected by the factors from the physicochemical aspects of sensory preference. Furthermore, this paper can support the wine experts especially the red wine expert evaluations, and ultimately improve the product that is provided with improved quality. In addition, similar concepts can be used to help increase the appreciation of the target audience by modeling consumer tastes from wine markets.

Contents

1. Introduction

In this project, we used Orange and RStudio to see the results. That is why our report includes Orange version's report and RStudio version's report. Our purpose is to display the quality of red wine that is affected by the factors from the physicochemical aspects of sensory preference.

2. Related Work

The three papers that we found uses the same datasets with us, they prefered to use the data mining approach. They used multiple regression and neural network methods for sensitivity analysis.

The other one of the papers that we have found uses the climate effects on wine quality. They tried to make understandable patterns to use the pH on the grape, the type of fertilizer, etc.

The final one and one of the interesting papers that we have found is about how berries affect wine quality. Berries from Machine-, Cane- and Spur-pruned vines were sampled for analysis of berry size and berry phenolic composition.

All those searches used unsupervised techniques as a first step, after that they used supervised learning techniques to see their search's accuracy.

3. Methodology

3.1 Dataset Information

The two datasets are related to a red variant of the Portuguese "Vinho Verde" wine. 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.).

These datasets can be viewed as classification or regression tasks. The classes are ordered and not balanced (e.g. there are many more normal wines than excellent or poor ones). Outlier detection algorithms could be used to detect the few excellent or poor wines. Also, we are not sure if all input variables are relevant. So it could be interesting to test feature selection methods.

3.1.1Attribute Information:

Input variables (based on physicochemical 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
Output variable (based on sensory data):
12 - quality (score between 0 and 10)

3.2 ORANGE (Part I)

3.2.1 Preprocessing

As a first step, we download the dataset which is already in Orange because it is one of the most popular datasets. We create a data table to see the dataset more open and be able to pre-process.

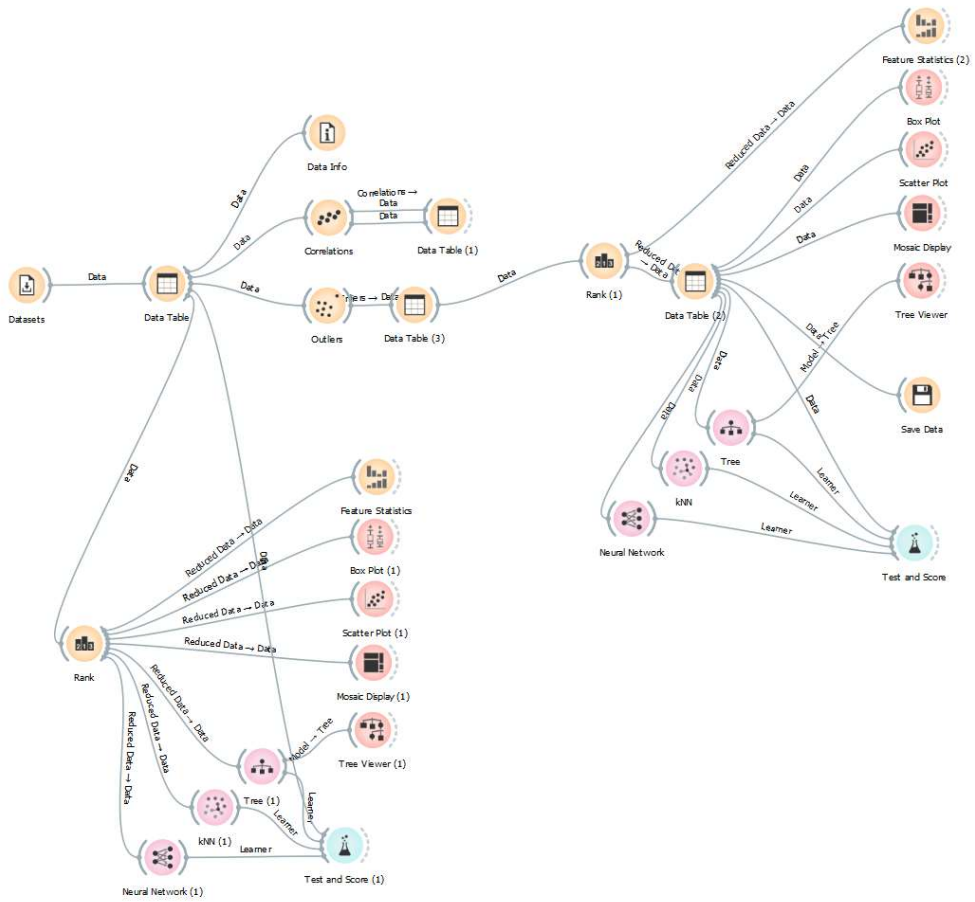


Figure 1:General schema for better understanding

Before removing outliers, we checked the rank to see the most effective attributes for creating a better learning path.

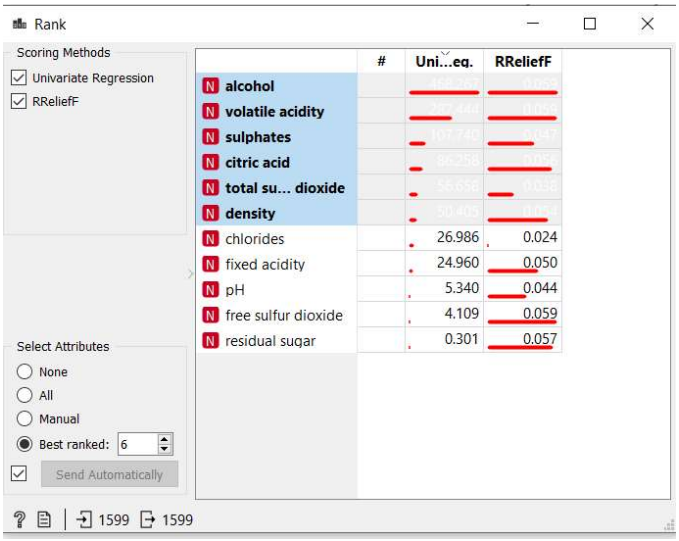


Figure 2:Rank of non-cleared data version

After made the firm the attributes that we have to pay attention more, we created feature statistics, box plot, scatter plot and mosaic plot to compare. This decision gave us the viewpoint to see how important to clean our data. Also, we process our non-clear data for kNN, Neural Network, and Decision Tree and put the results on the Test&Score. You can see the results on the below.

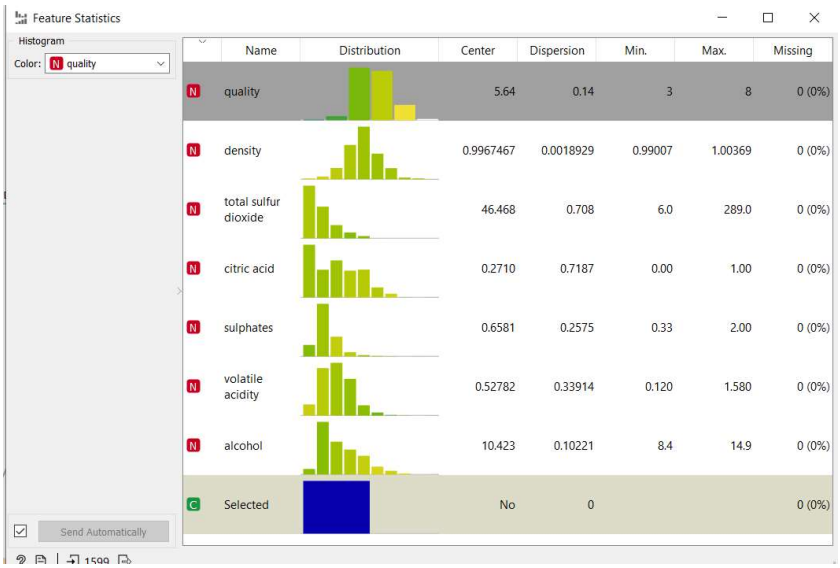


Figure 3:Feature Statistics of non-cleared data version

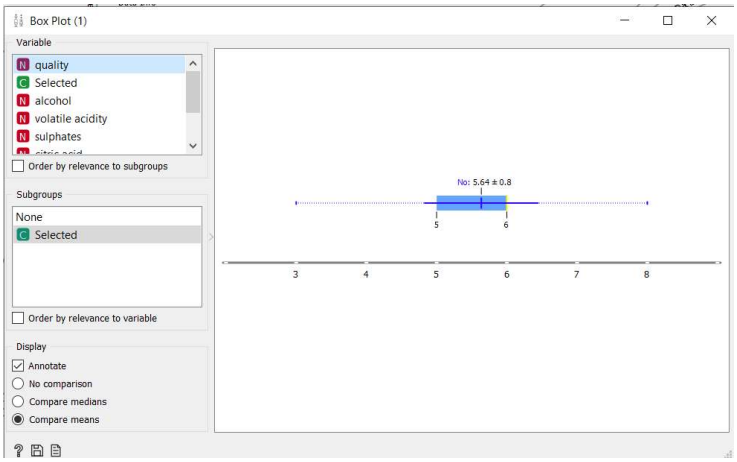


Figure 4:Quality's box plot of non-cleared version

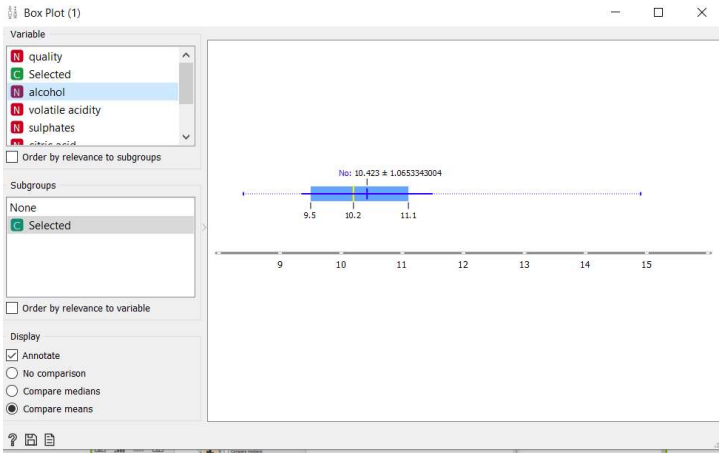


Figure 5: Alcohol's Box Plot of non-cleared data version

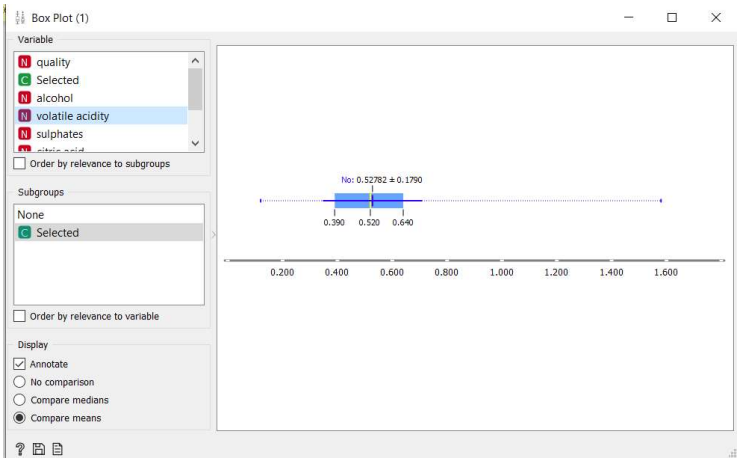


Figure 6:Volatile Acid's box plot of non-cleared version

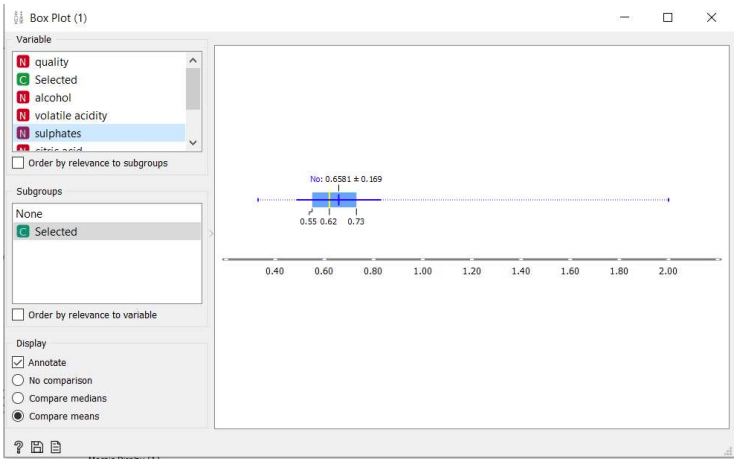


Figure 7:Sulphates' box plot of non-cleared version

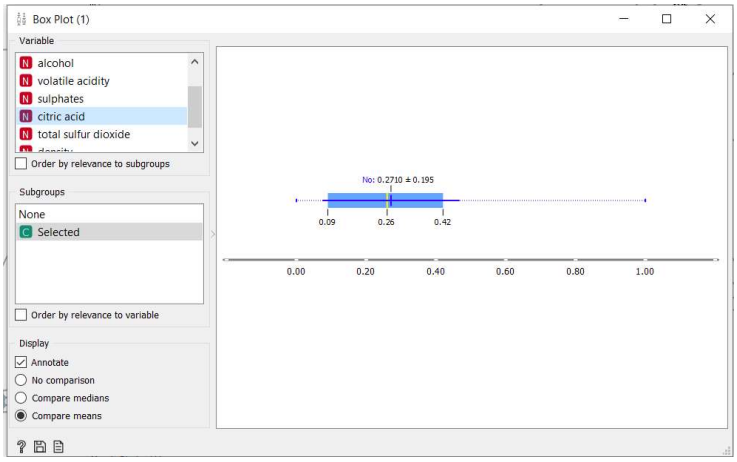


Figure 8:Citric Acid's box plot of non-cleared version

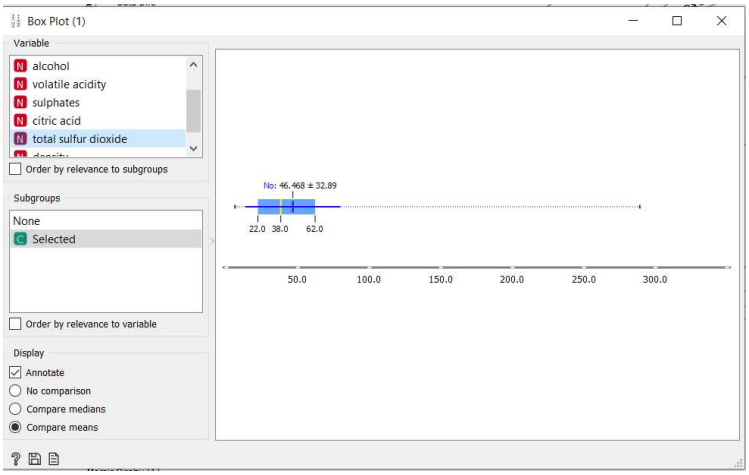


Figure 9:Total Sulfur Dioxide's box plot of non-cleared version

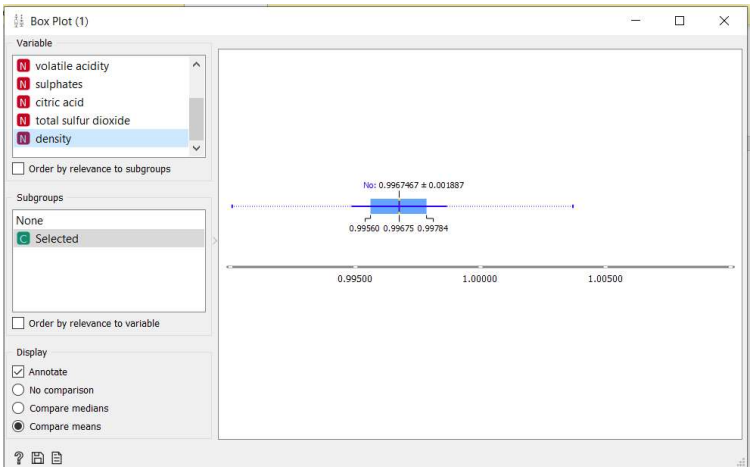


Figure 10:Density's box plot of non-cleared version

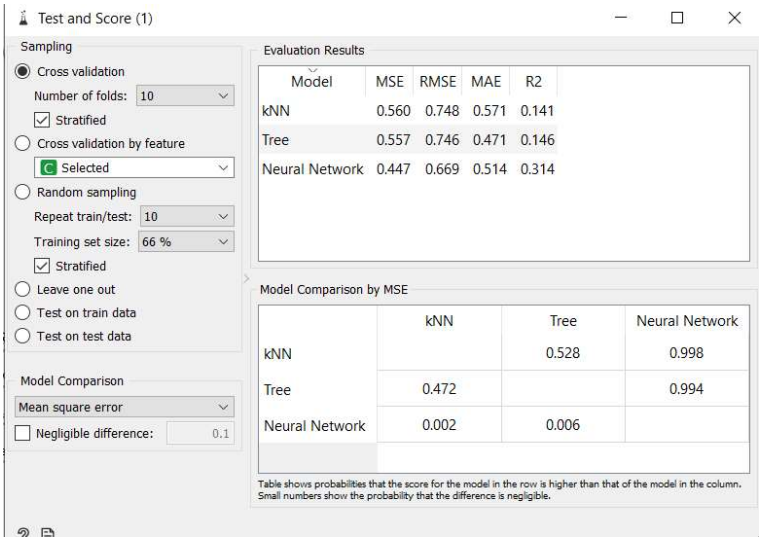


Figure 11:Test&Score of non-cleared version

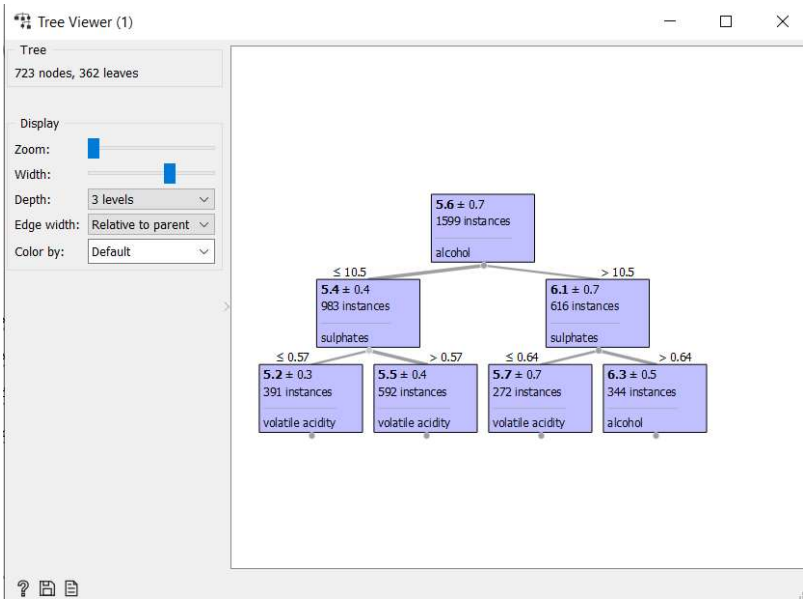


Figure 12:The general view of the Decision Tree of non-cleared data

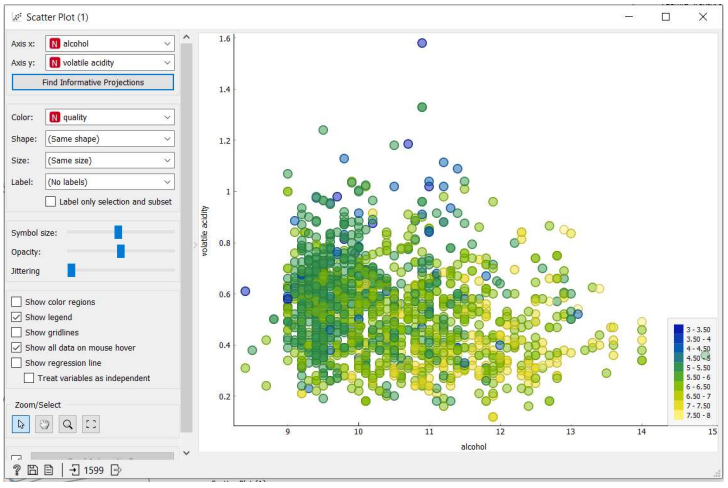


Figure 13:Scatter plot of Alcohol and Volatile Acidity Attributes of Non-Cleared Data

Figure 14:Mosaic Display of Alcohol and Volatile Acidity Attributes of Non-Cleared Data

We chose only one scatter plot otherwise we have to made 56 scatter plots for both non-cleared and cleared data. That's why we decide that Alcohol and Volatile Acidity because of these two attributes are the most

effective attributes for our data. This is also the reason why we only show Alcohol and Volatile Acidity's mosaic display.

As a second step, we removed outliers and made a new data table. After that, we checked the rank of the cleared data to see which attributes more intense after the pre-process.

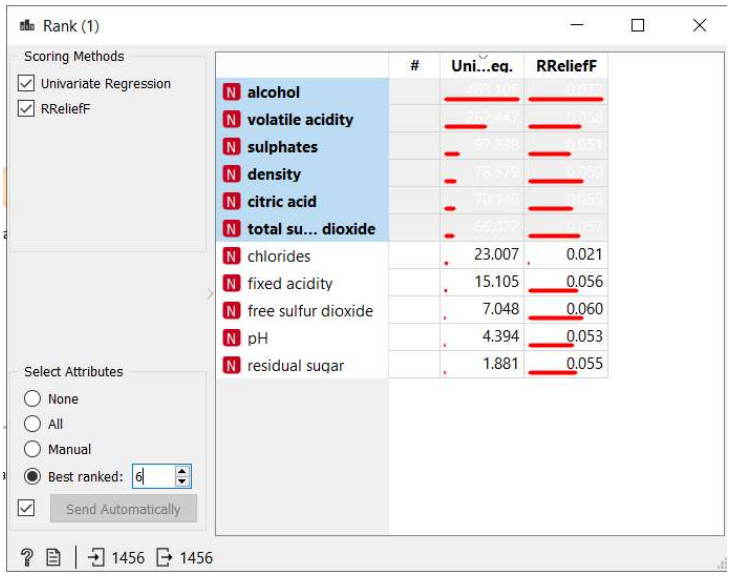


Figure 15:Rank of the Cleared Data

After detecting the more important attributes, we start to create feature statistics, box plot, mosaic display, and scatter plot. Also, we process our cleared data for kNN, Neural Network, and Decision Tree and put the results on the Test&Score. You can see the results on the below.

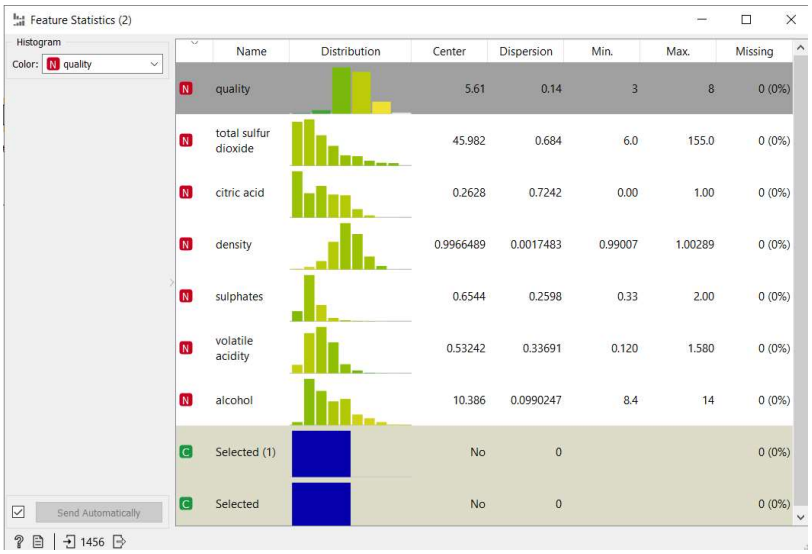


Figure 16:Histogram of cleared version

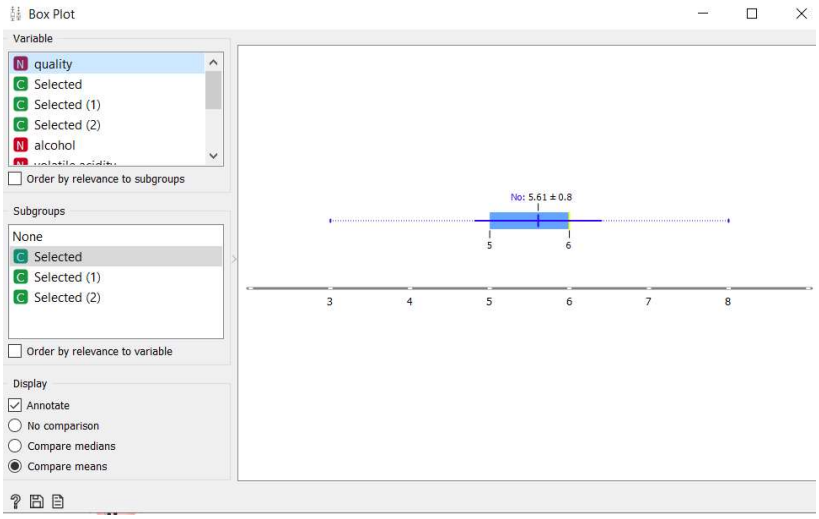


Figure 17:Quality's box plot of cleared version

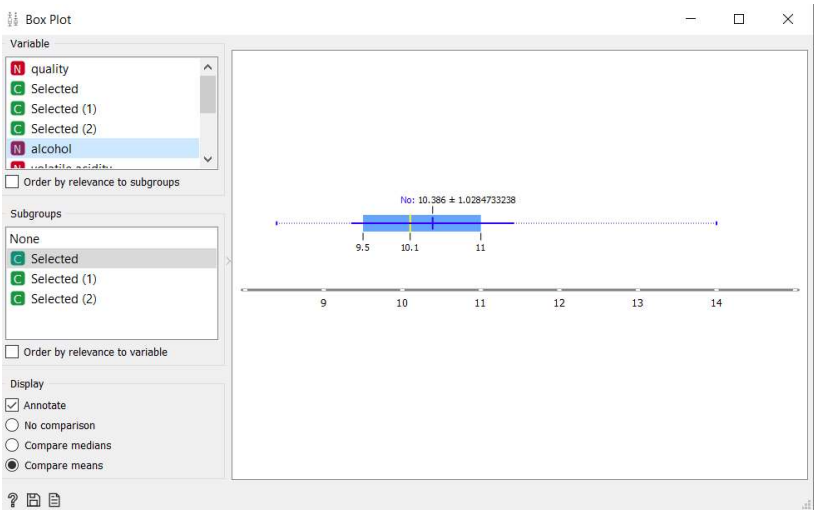


Figure 18:Alcohol's box plot of cleared version

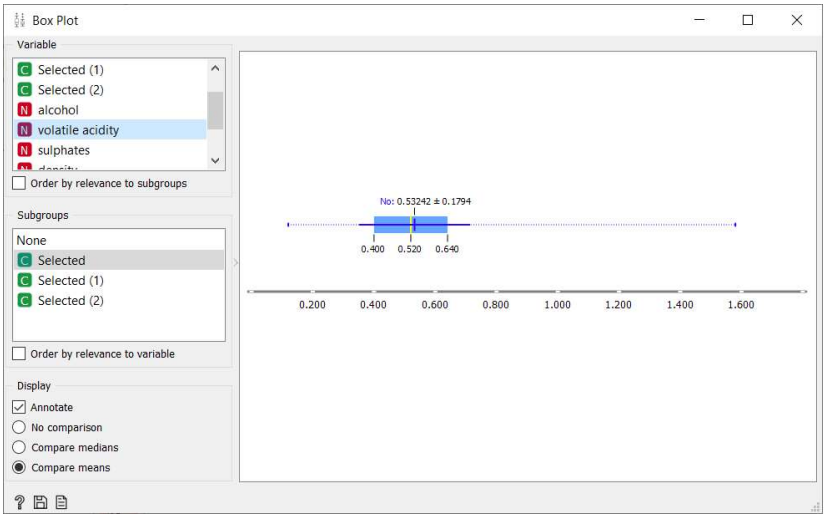


Figure 19:Volatile Acidity's box plot of cleared version

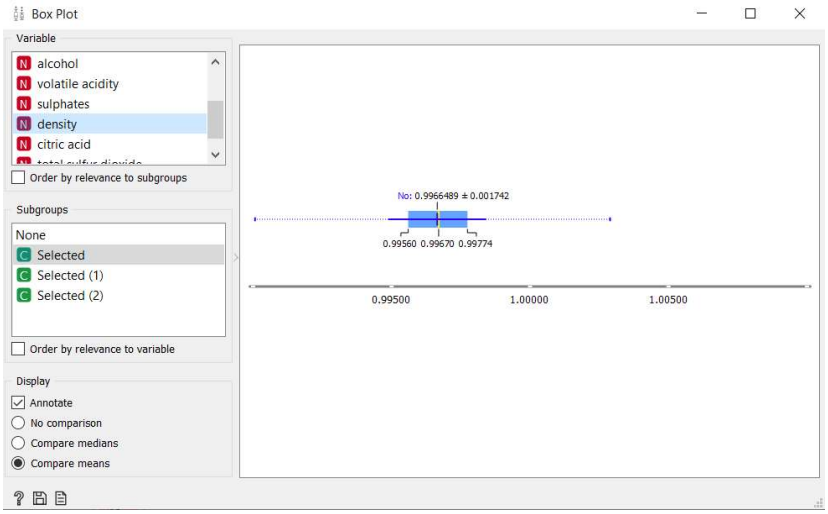


Figure 20:Density's box plot of cleared version

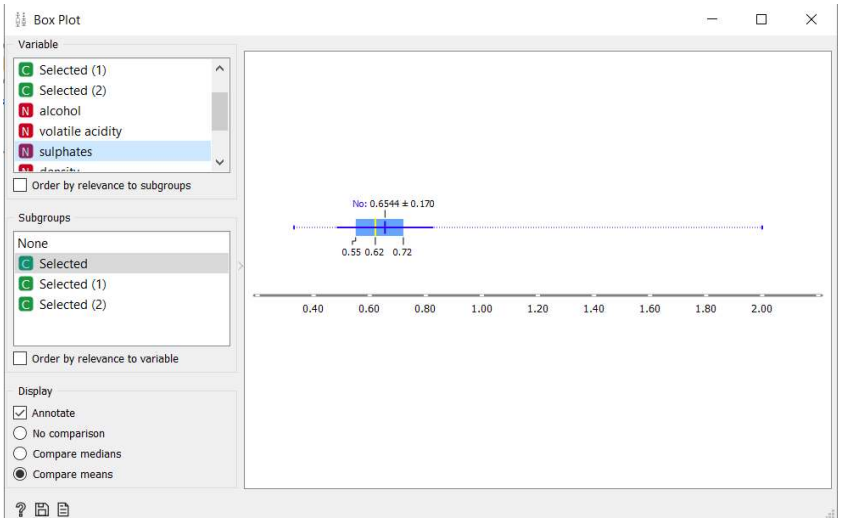


Figure 21:Sulphates' box plot of cleared version

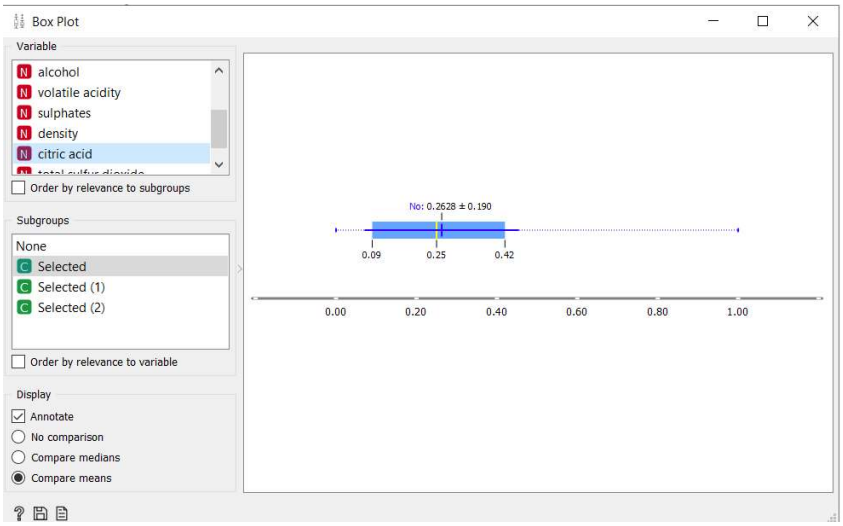


Figure 22:Citric Acid's box plot of cleared version

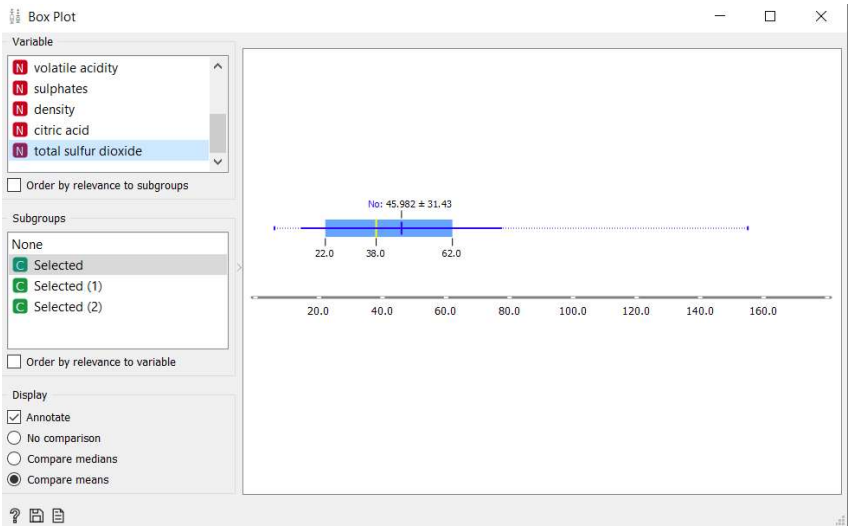


Figure 23:Total Sulfur Dioxide's box plot of cleared version

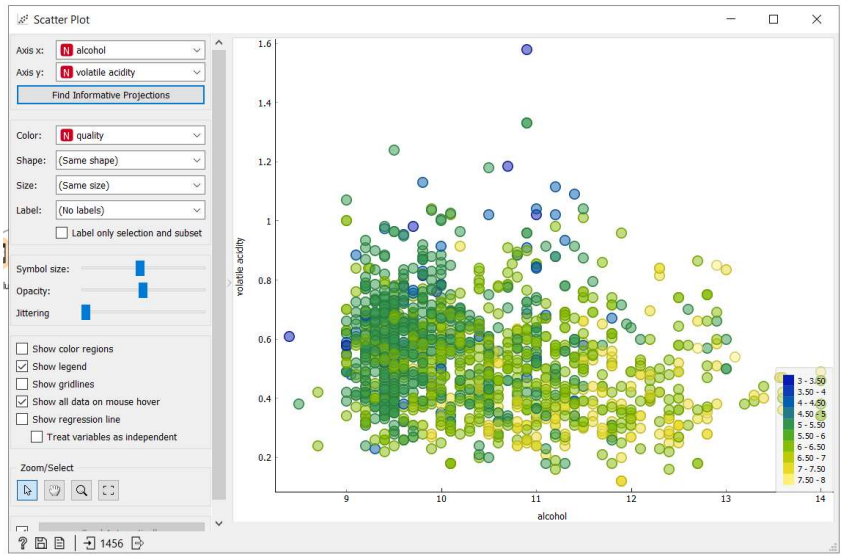


Figure 24:Scatter plot of Alcohol and Volatile Acidity Attributes of Cleared Data

Color range of the scatter plot is depends on Quality attribute.

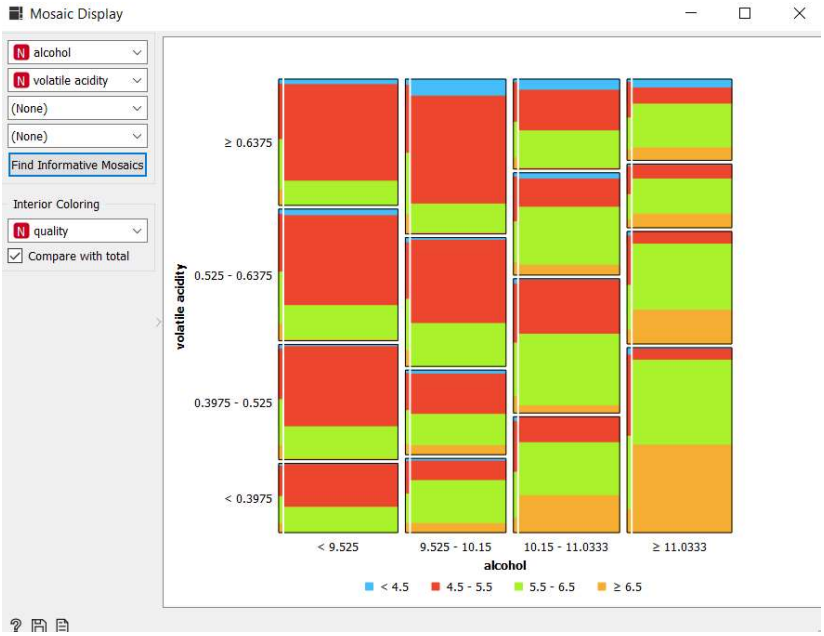


Figure 25:Mosaic Display of Alcohol and Volatile Acidity Attributes of Cleared Data

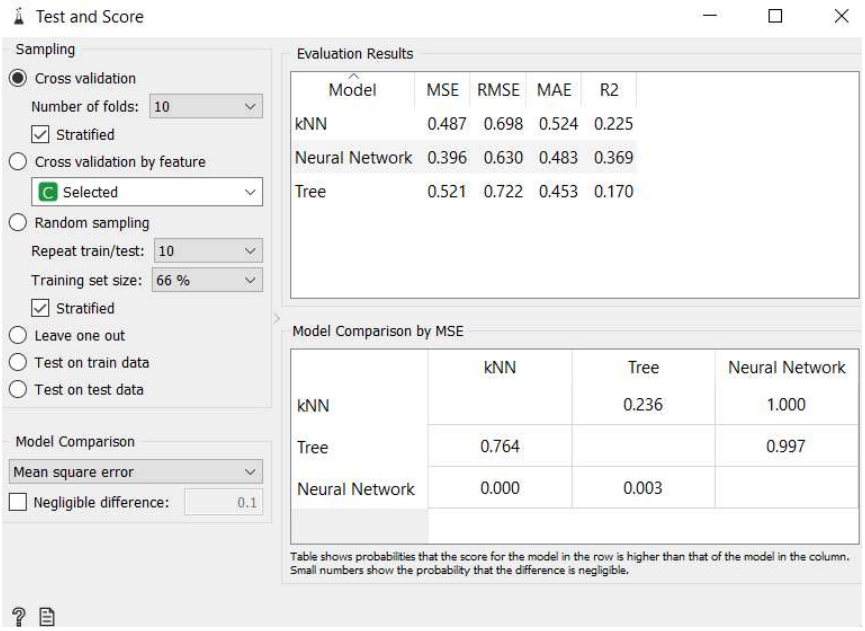


Figure 26:Test&Score of cleared version

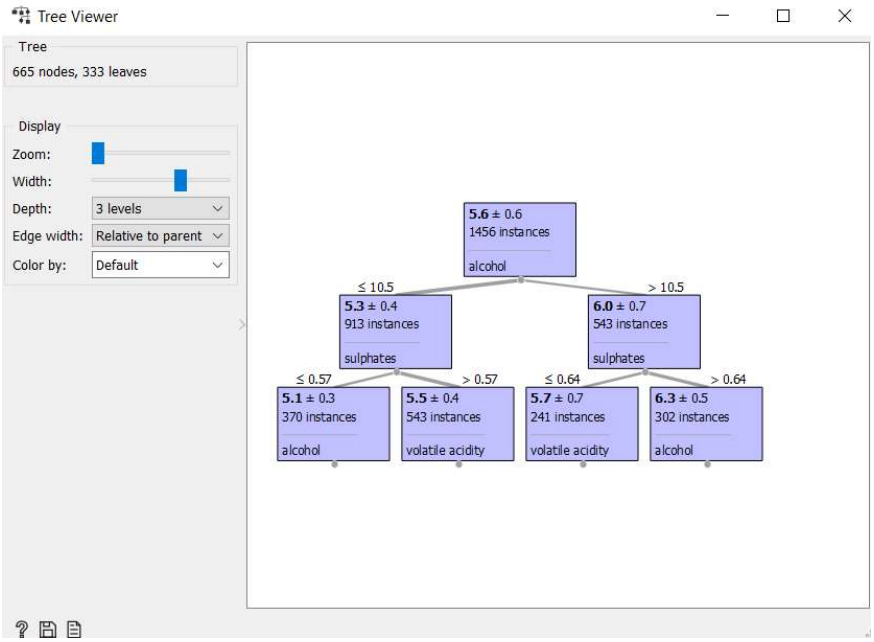


Figure 27:Decision Tree View of the cleared data

3.2.2 Classification Methods

To understand the importance of preprocessing, we used the classification methods to our cleaned dataset &noncleaned dataset that before removing the outliers, such as kNN, decision tree, and neural networks. You can see the results of uncleaned data in Figures 11 and 12. The results of the cleaned data in Figures 26 and 27.

3.2.3 Results

Overall, when you compare boxplots as clear and non-clear versions, you'll see that every boxplot's mean had been changed. Besides this, they almost the same because this data has already been cleaned by other data scientists. All figures are placed under the preprocessing section in terms of readability and comparison. You can see the results of uncleaned data in Figures 11 and 12. The results of the cleaned data in Figures 26 and 27. If you check the Test&Score of the cleaned data, kNN has the highest mean square error and Neural Network has the least MSE. Also, when we compared these methods depend on their mean absolute error (MAE) kNN has the highest value and Decision Tree has the lowest value. This is why we can say that the kNN method is not the best method for this dataset.

3.2 R STUDIO (Part II)

3.3.1 Preprocessing

First, we started our project by cleaning the useless data variables. We replaced all the “NULL” variables with NA strings. However, we saw that there were no missing values in this database. At first, we checked these functions, dim() and summary() to understand our data set a bit more. After all those things done, we turned some attributes into numeric and as well as removed the ones in character form, they were pretty much useless in our data set. Later, we checked if there were any duplicated records in our dataset and there were none, in the code, we even double-checked it for safety. But since this dataset had been used before by other data scientists, we saw that there is no duplicated data in this refined dataset.

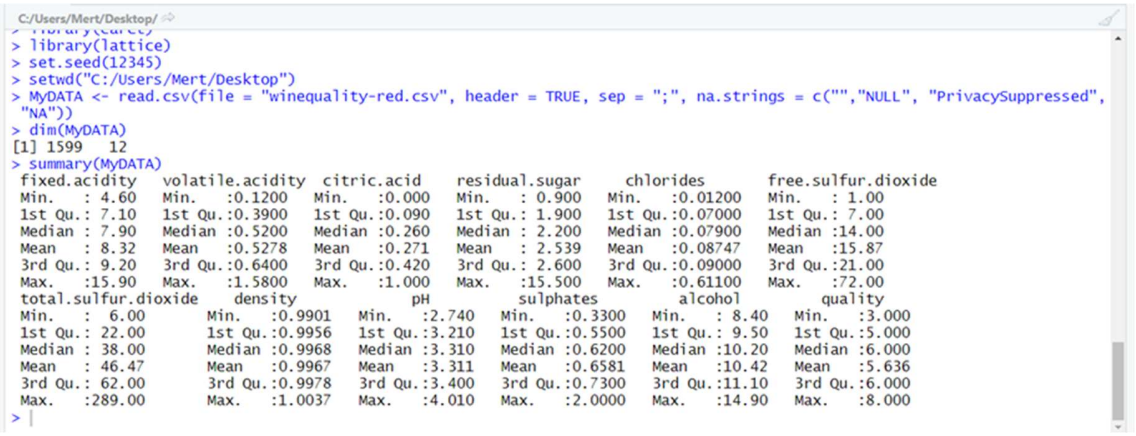
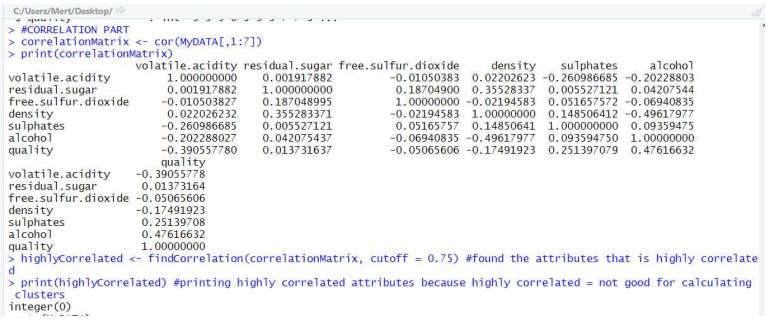
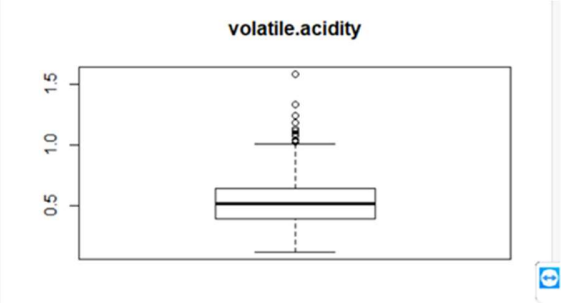


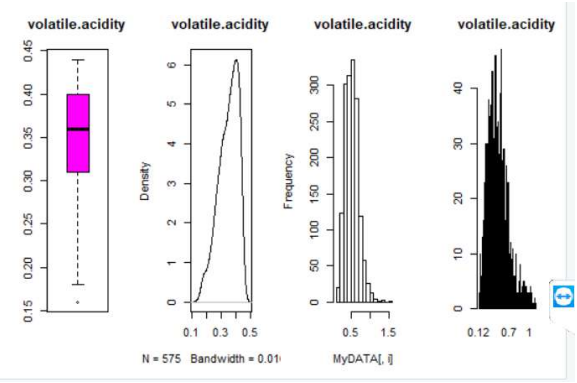
Figure 28:First checking dataset's information

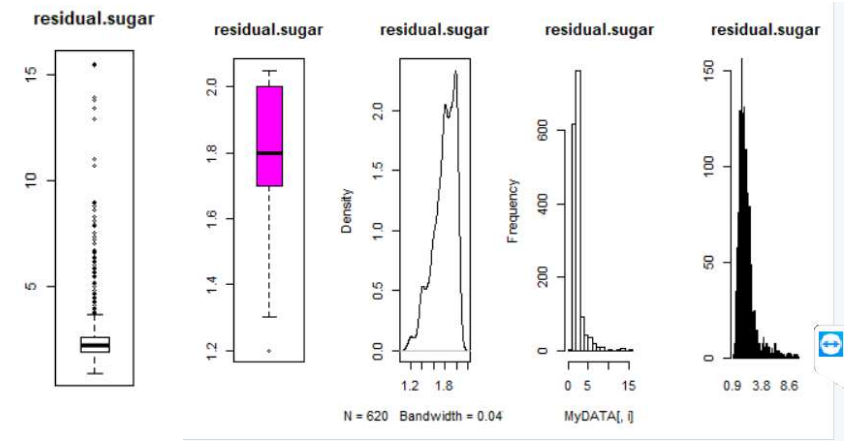


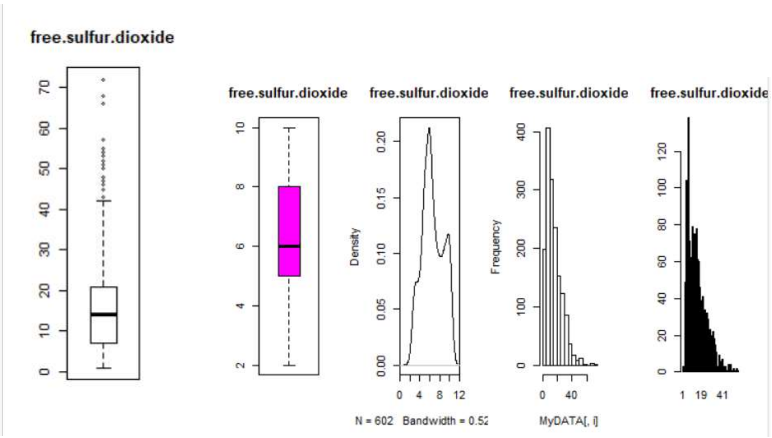
Later on, we checked if there are correlated attributes and outliers in our data, so we had to remove them to get a good view of the data set, but first, we determined highly correlated attributes and since there is no such thing of our refined data, we didn’t remove anything from our data set. After the correlation part, we quickly determined the outliers by checking each attributes boxplot with outliers and checking the values

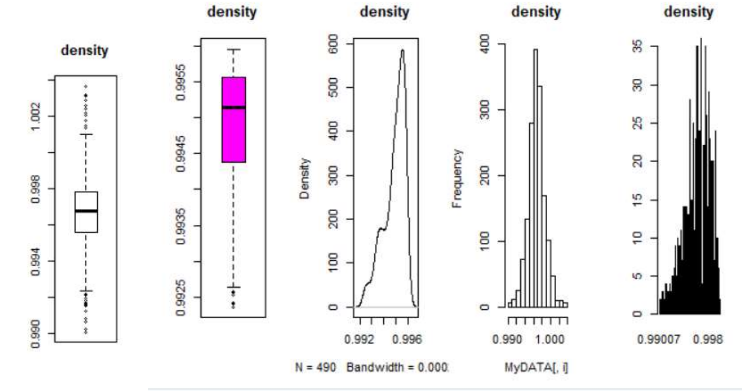
less than Q1 - 1.5 *IQR or greater than Q3 +1.5 * IQR. When we removed the extremes from our data set, we checked boxplot, plot, histogram, and barplot of the attributes, we colored the outlier-free boxplot with purple to see the difference. You can see each attributes boxplot, plot, histogram, and barplot in the below.

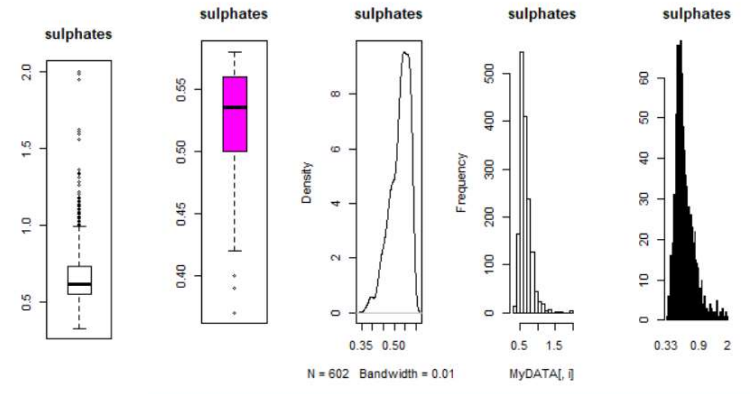


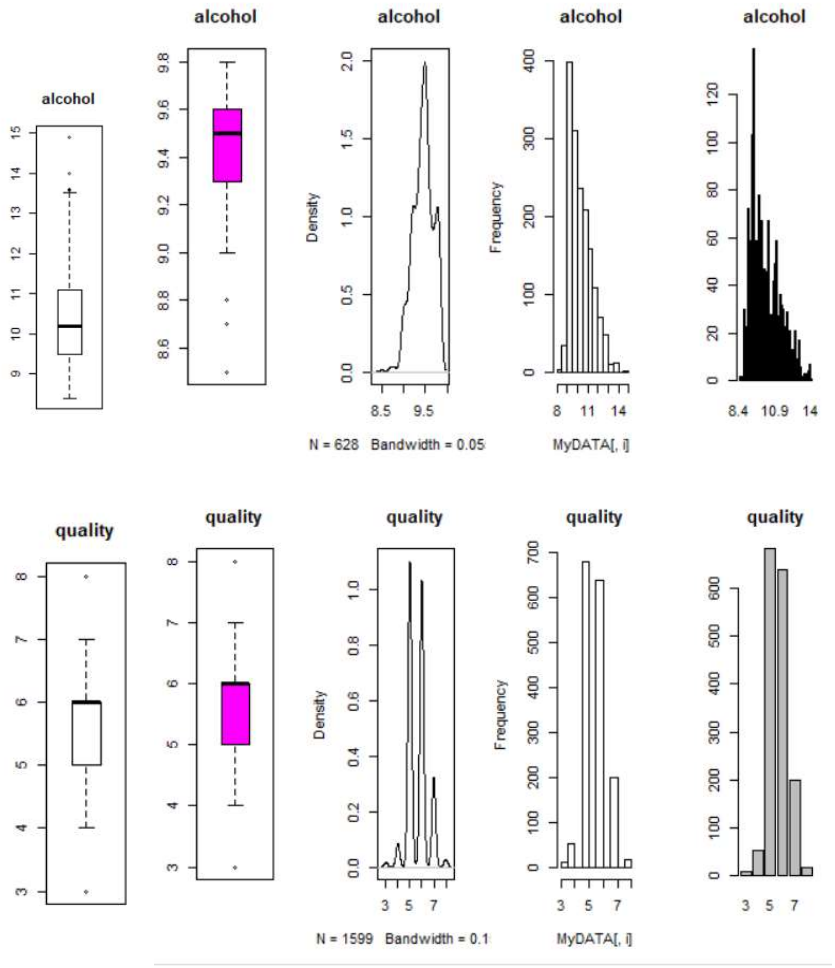




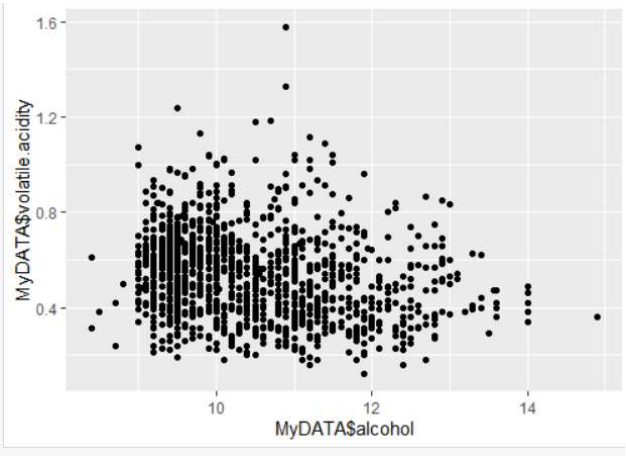

free.sulfur.dioxide
free.sulfur.dioxide
free.sulfur.dioxide
free.sulfur.dioxide
free.sulfur.dioxide
Density
Frequency
N = 602 Bandwidth = 0.52
MyDATA[, i]


density
density
density
density
density
Density
Frequency
N = 490 Bandwidth = 0.000
MyDATA[, i]


sulphates
sulphates
sulphates
sulphates
sulphates
Density
Frequency
N = 602 Bandwidth = 0.01
MyDATA[, i]


alcohol
alcohol
alcohol
alcohol
alcohol
Density
Frequency
N = 628 Bandwidth = 0.05
MyDATA[, i]
quality
quality
quality
quality
quality
Density
Frequency
N = 1599 Bandwidth = 0.1
MyDATA[, i]

Then we remove outliers and after removing the correlations and outliers from our data set, we checked the mosaic plot and the scatter plot, but because our data still has so many dimensions, the mosaic plot turned out to be unreadable, and with scatter plot, we had to check with chosen 2 variables from the data.



set.

Figure 29: Scatter plot's of top 2 attributes

3.3.2 Classification Methods

When we finished to preprocessing part, we got a more understandable data frame. This is the type of data frame that we want because, in that way, we could find some valuable pattern to visualize. As a second step, we started to search for classification methods. First, we divided our data set 75 percentage to 25 percent for training and test.

MyDATA	1599 obs. of 7 variables
testing	399 obs. of 7 variables
training	1200 obs. of 7 variables

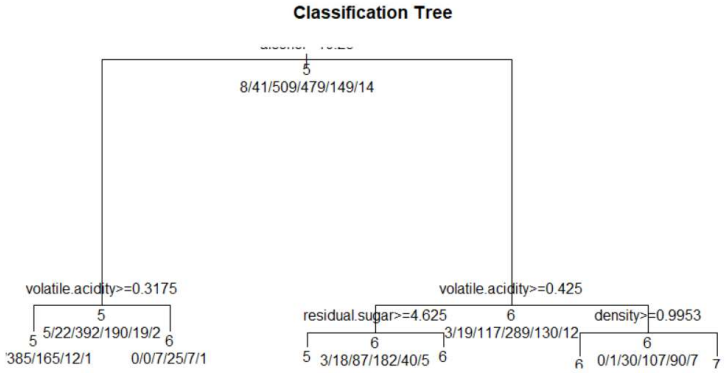
After that, we check the prediction and use confusionMatrix function. It gave us a clear understanding. You can check over the results and its classification tree on below (also first three nodes for understanding learning better);

```
Call:
rpart(formula = quality ~ ., data = training, method = "class")
  n= 1200

          CP nsplit rel error    xerror       xstd
1 0.24891462      0 1.0000000 1.0000000 0.02477589
2 0.02604920      1 0.7510854 0.7684515 0.02489956
3 0.01254221      2 0.7250362 0.7568741 0.02485859
4 0.01000000      5 0.6874096 0.7684515 0.02489956

Variable importance
            alcohol    volatile.acidity             density           sulphates      residual.sugar free.sulfur.dioxide
                 41                  22                  19                   8                   7                   3

Node number 1: 1200 observations,    complexity param=0.2489146
  predicted class=5  expected loss=0.5758333  P(node) =1
    class counts:     8    41   509   479   149    14
   probabilities: 0.007 0.034 0.424 0.399 0.124 0.012
  left son=2 (630 obs) right son=3 (570 obs)
  Primary splits:
      alcohol             < 10.25    to the left,  improve=76.474740, (0 missing)
      sulphates           < 0.585    to the left,  improve=31.823580, (0 missing)
      volatile.acidity    < 0.3625   to the right, improve=30.298630, (0 missing)
      density             < 0.995745 to the right, improve=24.563020, (0 missing)
      free.sulfur.dioxide < 27.5     to the right, improve= 3.754254, (0 missing)
  Surrogate splits:
      density             < 0.995745 to the right, agree=0.719, adj=0.409, (0 split)
      volatile.acidity    < 0.515    to the right, agree=0.618, adj=0.195, (0 split)
      sulphates           < 0.675    to the left,  agree=0.603, adj=0.165, (0 split)
      residual.sugar      < 2.375    to the left,  agree=0.558, adj=0.068, (0 split)
      free.sulfur.dioxide < 6.5      to the right, agree=0.553, adj=0.060, (0 split)

Node number 2: 630 observations,    complexity param=0.0260492
  predicted class=5  expected loss=0.3777778  P(node) =0.525
    class counts:     5    22   392   190    19     2
   probabilities: 0.008 0.035 0.622 0.302 0.030 0.003
  left son=4 (590 obs) right son=5 (40 obs)
  Primary splits:
      volatile.acidity    < 0.3175   to the right, improve=13.981360, (0 missing)
      sulphates           < 0.575    to the left,  improve=12.524740, (0 missing)
      free.sulfur.dioxide < 17.5     to the right, improve= 6.243788, (0 missing)
      density             < 0.996735 to the left,  improve= 4.647619, (0 missing)
      alcohol             < 9.85     to the left,  improve= 4.409737, (0 missing)

Node number 3: 570 observations,    complexity param=0.01254221
  predicted class=6  expected loss=0.4929825  P(node) =0.475
    class counts:     3    19   117   289   130    12
   probabilities: 0.005 0.033 0.205 0.507 0.228 0.021
  left son=6 (335 obs) right son=7 (235 obs)
  Primary splits:
      volatile.acidity    < 0.425    to the right, improve=13.451470, (0 missing)
      alcohol             < 11.45    to the left,  improve=13.001320, (0 missing)
      sulphates           < 0.645    to the left,  improve=11.375160, (0 missing)
      residual.sugar      < 4.625    to the right, improve= 6.690977, (0 missing)
      free.sulfur.dioxide < 13.5     to the right, improve= 5.355455, (0 missing)
  Surrogate splits:
      sulphates           < 0.725    to the left,  agree=0.640, adj=0.128, (0 split)
      alcohol             < 11.65    to the left,  agree=0.614, adj=0.064, (0 split)
      density             < 0.99161  to the right, agree=0.605, adj=0.043, (0 split)
      residual.sugar      < 6.35     to the left,  agree=0.600, adj=0.030, (0 split)
      free.sulfur.dioxide < 4.5      to the right, agree=0.600, adj=0.030, (0 split)
```

```
> pred<-predict(MyDATA_dtree, training[, 1:7], type="class")
> table(training$quality,pred)
   pred
      3   4   5   6   7   8
  3   0   0   5   3   0   0
  4   0   0  23  18   0   0
  5   0   0 399 104   6   0
  6   0   0 169 276  34   0
  7   0   0  17  82  50   0
  8   0   0   2   9   3   0
> |
```

```
Overall Statistics

               Accuracy : 0.6042
                 95% CI : (0.5759, 0.632)
    No Information Rate : 0.5125
    P-Value [Acc > NIR] : 1.07e-10

                  Kappa : 0.3504

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 3 Class: 4 Class: 5 Class: 6 Class: 7 Class: 8
Sensitivity                NA       NA   0.6488   0.5610  0.53763       NA
Specificity          0.993333  0.96583   0.8120   0.7133  0.91057  0.98833
Pos Pred Value             NA       NA   0.7839   0.5762  0.33557       NA
Neg Pred Value             NA       NA   0.6874   0.7004  0.95909       NA
Prevalence           0.000000  0.00000   0.5125   0.4100  0.07750  0.00000
Detection Rate       0.000000  0.00000   0.3325   0.2300  0.04167  0.00000
Detection Prevalence 0.006667  0.03417   0.4242   0.3992  0.12417  0.01167
Balanced Accuracy          NA       NA   0.7304   0.6371  0.72410       NA
> |
```

And after all the results and coding, we got 0.6042 accuracies in the decision tree.



After we got the decision tree's result, we wanted to see what can we see when we use the kNN classification algorithm so we defined the cross-validation 10 and k = 5 and we predict and put it in the value that is named knnPredict by us. We want to get the confusion matrix to see accuracy value and other parameter values, but data and reference are not factors with the same levels, so normal confusionMatrix method don't work this is why we tried another path for it.

```
k-Nearest Neighbors

1200 samples
   6 predictor

Pre-processing: centered (6), scaled (6)
Resampling: Cross-Validated (10 fold, repeated 5 times)
Summary of sample sizes: 1079, 1081, 1079, 1081, 1080, 1080, ...
Resampling results across tuning parameters:

  k   RMSE       Rsquared   MAE
   5  0.6863253  0.3039156  0.5216970
   7  0.6835089  0.3022892  0.5250859
   9  0.6794781  0.3047789  0.5248240
  11  0.6736233  0.3141782  0.5242860
  13  0.6690792  0.3224168  0.5230851
  15  0.6662359  0.3282105  0.5227450
  17  0.6650924  0.3304798  0.5236242
  19  0.6636157  0.3337196  0.5234839
  21  0.6618922  0.3375438  0.5225893
  23  0.6605365  0.3409354  0.5234753
  25  0.6606108  0.3411263  0.5247701
  27  0.6597797  0.3432594  0.5247560
  29  0.6592500  0.3446138  0.5240567
  31  0.6593254  0.3450436  0.5245549
  33  0.6597557  0.3444710  0.5257355
  35  0.6596646  0.3452961  0.5263543
  37  0.6599267  0.3451838  0.5272342
  39  0.6598906  0.3457429  0.5276485
  41  0.6596928  0.3467495  0.5276559
  43  0.6590251  0.3486904  0.5277059

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was k = 43.
> |
```

```
> kNNPredictions <-predict(knnFit, testing)
> cmkNN <-table(kNNPredictions, testing$quality)
> accuracykNN = (sum(diag(cmkNN)))/sum(cmkNN)
> accuracykNN
[1] 0.01002506
```

Again, after all the results and coding has been done, we got 0.01002506 accuracies. Definitely, Decision Tree's accuracy is much better.

Our last classification method is the neural network method. We defined a value called NNModel to train with our data set and learn it by using the neural network algorithm.

```
> NNModel
Neural Network

1200 samples
   6 predictor

Pre-processing: scaled (6), centered (6)
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 1200, 1200, 1200, 1200, 1200, 1200, ...
Resampling results across tuning parameters:

  size  decay  RMSE      Rsquared    MAE
  1     0e+00  4.703601         NaN  4.634341
  1     1e-04  4.703601         NaN  4.634341
  1     1e-01  4.703692  0.32504002  4.634434
  3     0e+00  4.703601         NaN  4.634341
  3     1e-04  4.703601  0.04437527  4.634341
  3     1e-01  4.703659  0.32633049  4.634401
  5     0e+00  4.703601         NaN  4.634341
  5     1e-04  4.703601  0.04336056  4.634341
  5     1e-01  4.703648  0.32815645  4.634389

RMSE was used to select the optimal model using the smallest value.
The final values used for the model were size = 1 and decay = 1e-04.
> |
```

```
RMSE was used to select the optimal model using the smallest value.
The final values used for the model were size = 1 and decay = 0.
> cmNN <-table(NNPredictions, testing$quality)
> accuracyNN = (sum(diag(cmNN)))/sum(cmNN)
> accuracyNN
[1] 0.005012531
>
```

When the program finished learning, training set we tested it and got 0.005012531 accuracies.

3.3.3 Results

Overall, Decision Tree's accuracy is much higher compared with the other accuracy results. To sum up, in this project, we preprocessed the data and made it less non-trivial data. After that, we used our clean data for getting valuable patterns. For this, we tested three different classification methods for it. Then we checked those methods' accuracy and see that the decision tree method gave us the best result for this data set.

4. Conclusion

Our problem was to visualize reasonable & understandable paths for learning about wine quality we aimed to find factors that affect the red wine quality. In this project, we had a chance to see how important preprocessing is. Besides this, we used different classification methods and each one has a different percentage of accuracy. After searching for other research papers we realized that every method will not give the same accuracy in different datasets. In addition to our project, we used R Studio as well as Orange to compare data and algorithms better. For this reason, we believe that this study can be adapted in all applications using machine learning algorithms. All figures are placed under the classification methods section in terms of readability and comparison.

5.References

- https://onlinelibrary.wiley.com/doi/epdf/10.1111/j.1755-0238.2008.00011.x
- https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1755-0238.2008.00019.x
- https://papers.ssrn.com/sol3/papers.cfm?abstract_id=231217

- https://papers.ssrn.com/sol3/papers.cfm?abstract_id=231217

- https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781785286544/1/ch01lvl1sec11/data-preprocessing-techniques
- https://en.proft.me/2017/01/22/classification-using-k-nearest-neighbors-r/
- https://rpubs.com/njvijay/16444
- https://www.youtube.com/watch?v=qLYYHWYr8xI