RED WINE QUALITY

Machine Learning

Name	ID
Rubaiyad Noor Shahriar	19-39541-1
Muhammad, Akib	19-39372-1
Md. Adnan Shakib	18-38015-2
Md Amir Habib	19-41429-3

JUNE 14, 2022

AMERICAN INTERNATION UNIVERSITY BNAGLADESH
408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh

Contents

List of Figures

List of Tables

Notations

- $P(c|x)$ = the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
- $P(c)$ = the prior probability of *class*.
- $P(x|c)$ = the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ = the prior probability of *predictor*.

Introduction

Background study

Wine is an alcoholic drink typically made from fermented grapes. Yeast consumes the sugar in the grapes and converts it to ethanol and carbon dioxide, releasing heat in the process. Different varieties of grapes and strains of yeasts are major factors in different styles of wine. These differences result from the complex interactions between the biochemical development of the grape, the reactions involved in fermentation, the grape's growing environment (terroir), and the wine production process. Many countries enact legal appellations intended to define styles and qualities of wine. These typically restrict the geographical origin and permitted varieties of grapes, as well as other aspects of wine production. Wines not made from grapes involve fermentation of other crops including rice wine and other fruit wines such as plum, cherry, pomegranate, currant and elderberry.

Wine has been produced for thousands of years. The earliest evidence of wine is from ancient China (c. 7000 BC) Georgia (6000 BC), Persia (5000 BC), and Italy (4000 BC). New World wine has some connection to alcoholic beverages made by the indigenous peoples of the Americas, but is mainly connected to later Spanish traditions in New Spain. Later, as Old-World wine further developed viticulture techniques, Europe would encompass three of the largest wine-producing regions. Today, the five countries with the largest wine-producing regions are in Italy, Spain, France, the United States, and China.

Wine has long played an important role in religion. Red wine was associated with blood by the ancient Egyptians and was used by both the Greek cult of Dionysus and the Romans in their Bacchanalia; Judaism also incorporates it in the Kiddush, and Christianity in the Eucharist. Egyptian, Greek, Roman, and Israeli wine cultures are still connected to these ancient roots. Similarly, the largest wine regions in Italy, Spain, and France have heritages in connection to sacramental wine, likewise, viticulture traditions in the Southwestern United States started within New Spain as Catholic friars and monks first produced wines in New Mexico and California.

Problem Statement

The two datasets are related to red and white variants of the Portuguese "Vinho Verde" wine. For more details, consult the reference [Cortez et al., 2009]. 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 much more normal wines than excellent or poor ones).

We are going to classify this dataset according to the value of the quality

Objective

Our objective of this project is to classify the dataset for red wine quality. We will be using Naïve bayes algorithm, K-Nearest Neighbor algorithm, Decision tree algorithm to classify the dataset. The class mentioned as "*quality*". There are 5 classes which are numerical but determines the quality of the wine. By using the three-classification algorithm we will provide a comparative discussion on which classification algorithm provides the most efficient result.

Dataset

Data Information

fixed acidity: most acids involved with wine or fixed or nonvolatile (do not evaporate readily).

volatile acidity: the amount of acetic acid in wine, which at too high of levels can lead to an unpleasant, vinegar taste.

citric acid: found in small quantities, citric acid can add 'freshness' and flavor to wines.

residual sugar: 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.

chlorides: the amount of salt in the wine.

free sulfur dioxide: 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.

total sulfur dioxide: 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.

density: the density of wine is close to that of water depending on the percent alcohol and sugar content.

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.

sulphates: a wine additive which can contribute to sulfur dioxide gas (S02) levels, which acts as an antimicrobial and antioxidant.

alcohol: the percent alcohol content of the wine.

quality: output variable (based on sensory data, score between 0 and 10)

Model Development

Naïve Bayes

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:

$$\mathrm{P(c|x)} = \frac{P(X|C)\,P(c)}{P(x)}$$

Above,

- $P(c|x)$ is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

K-Nearest Neighbor (KNN) Algorithm

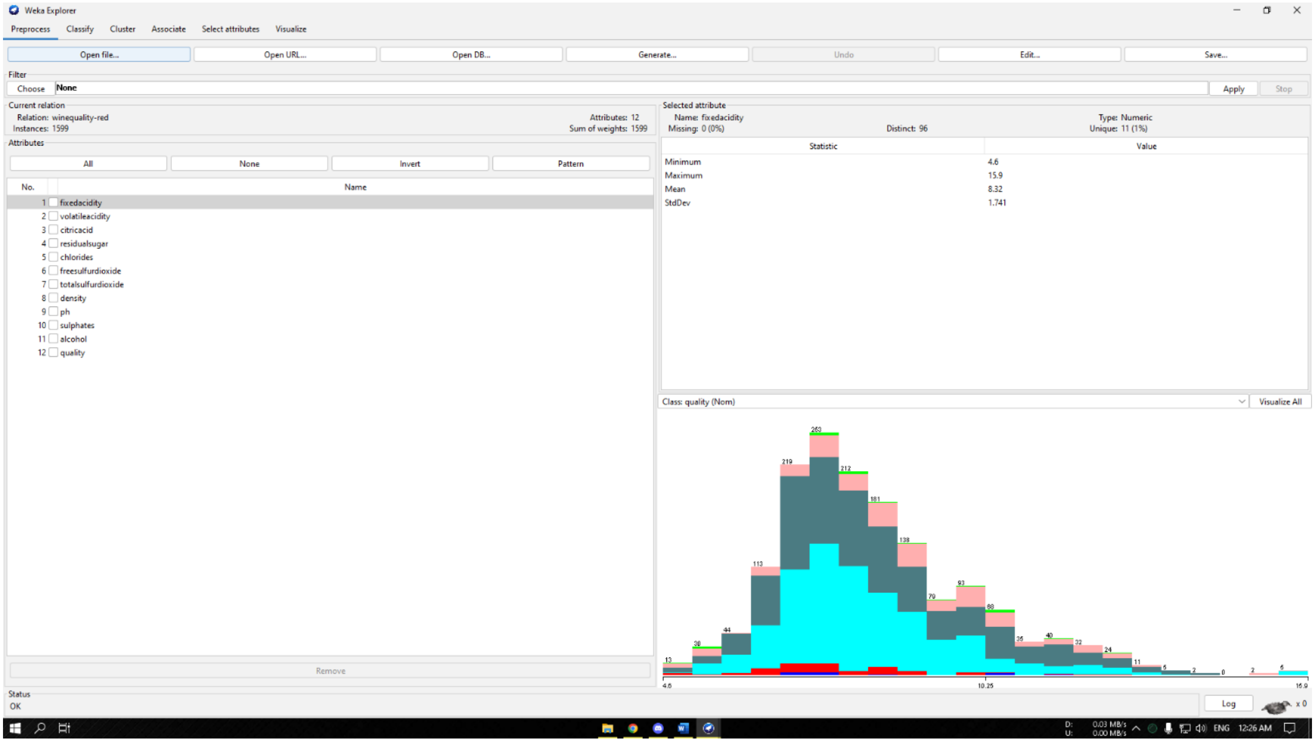
K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

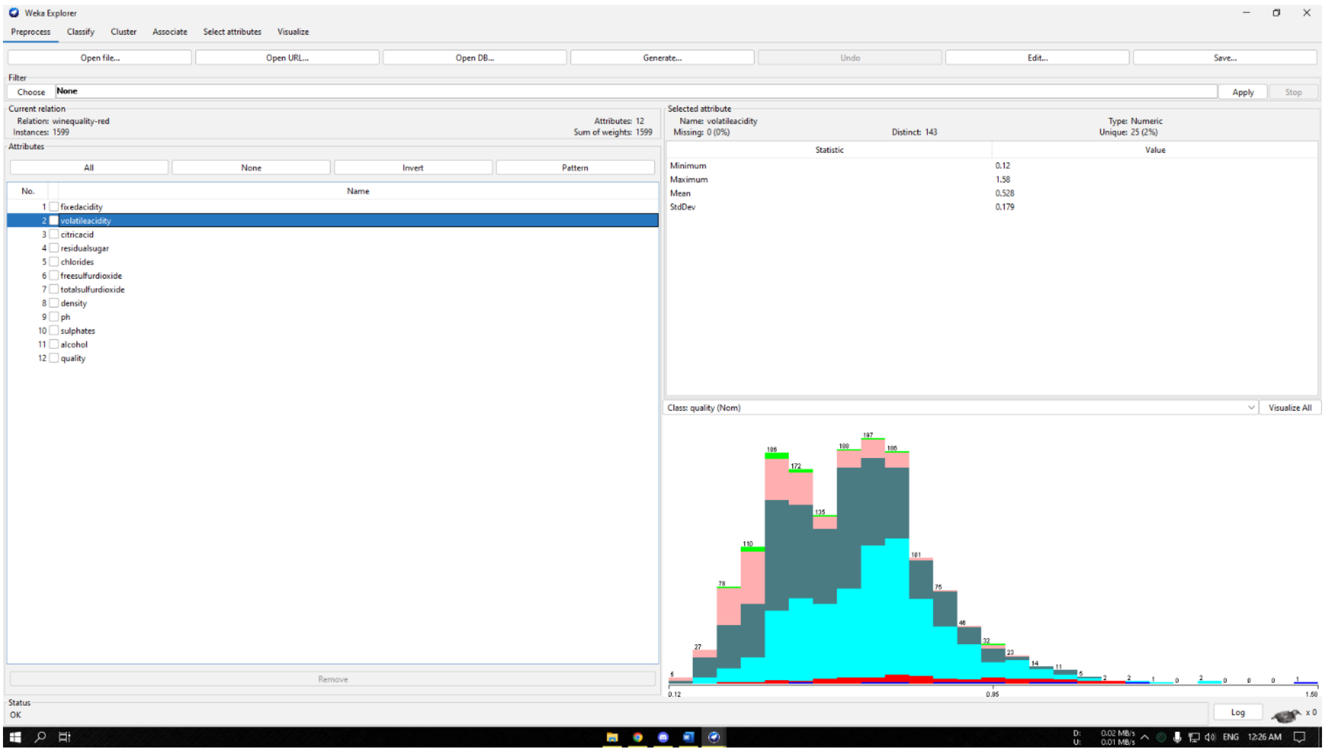
Decision Trees

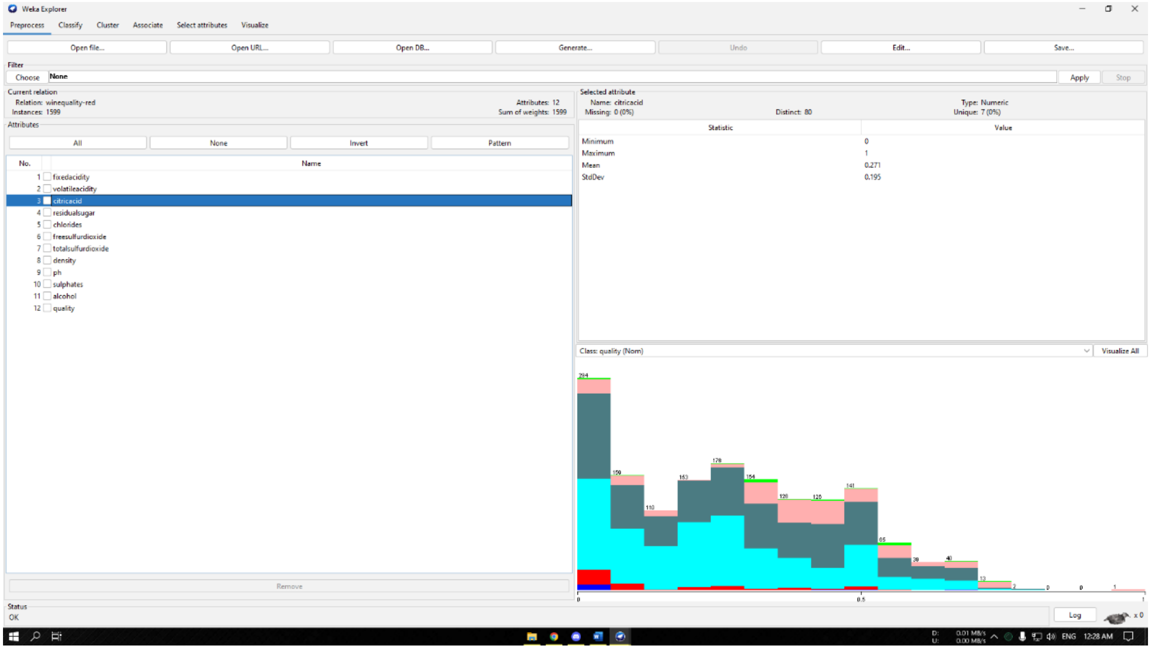
Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

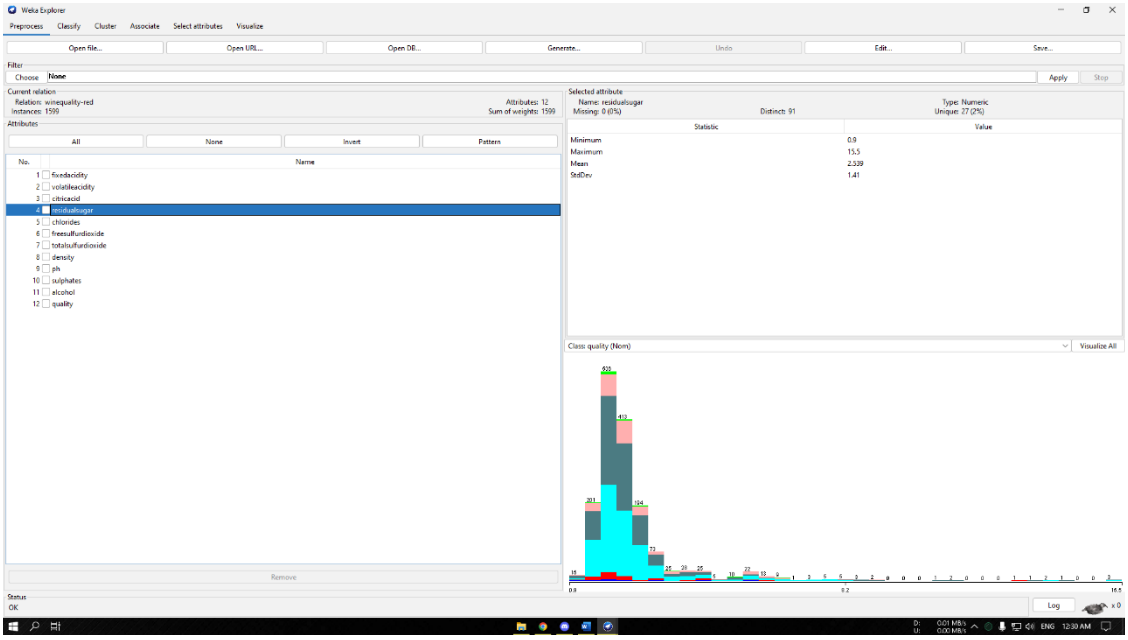
Graphical Representations

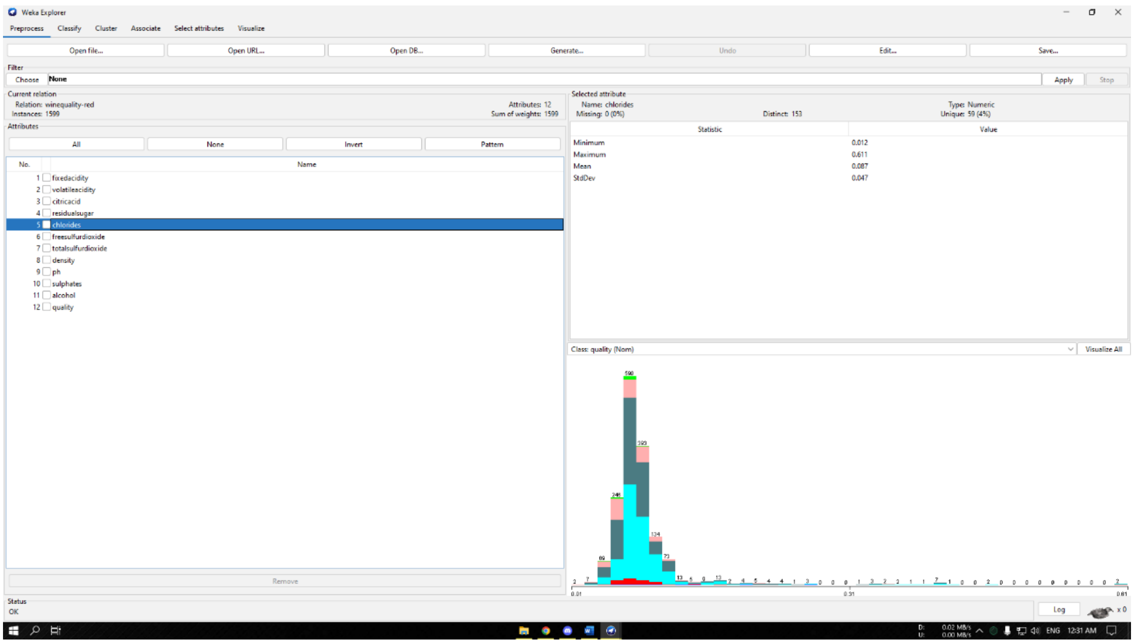
Variable value

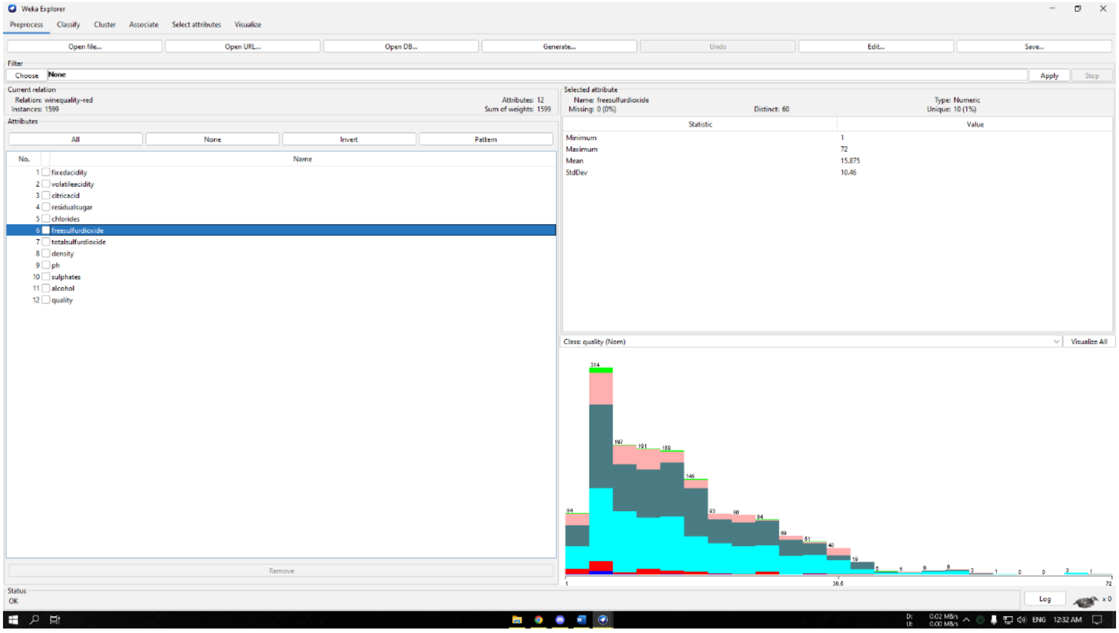


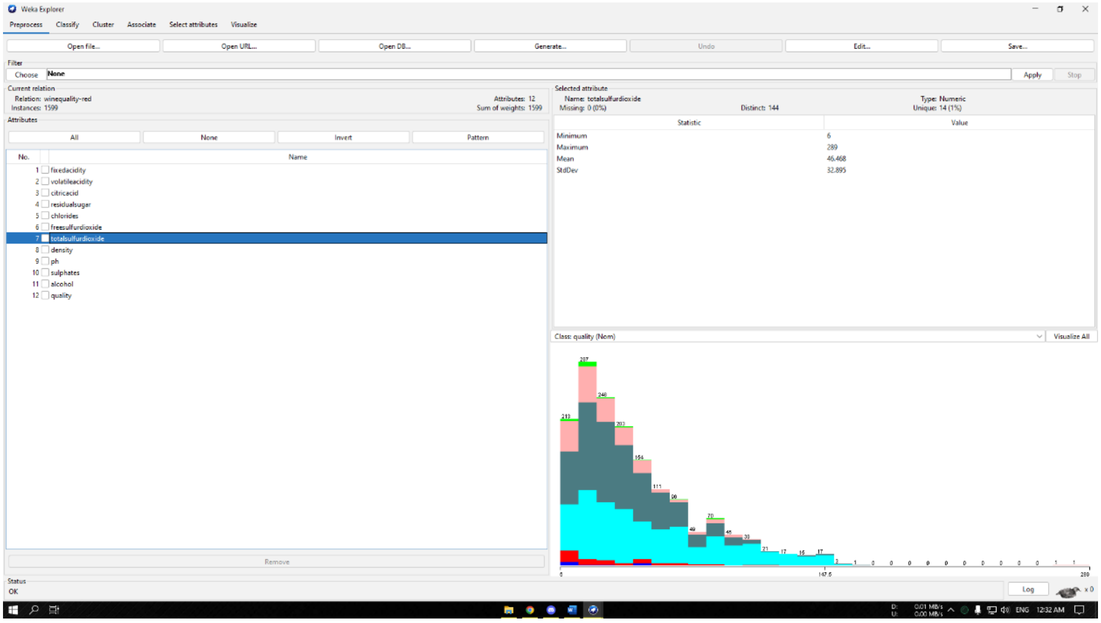


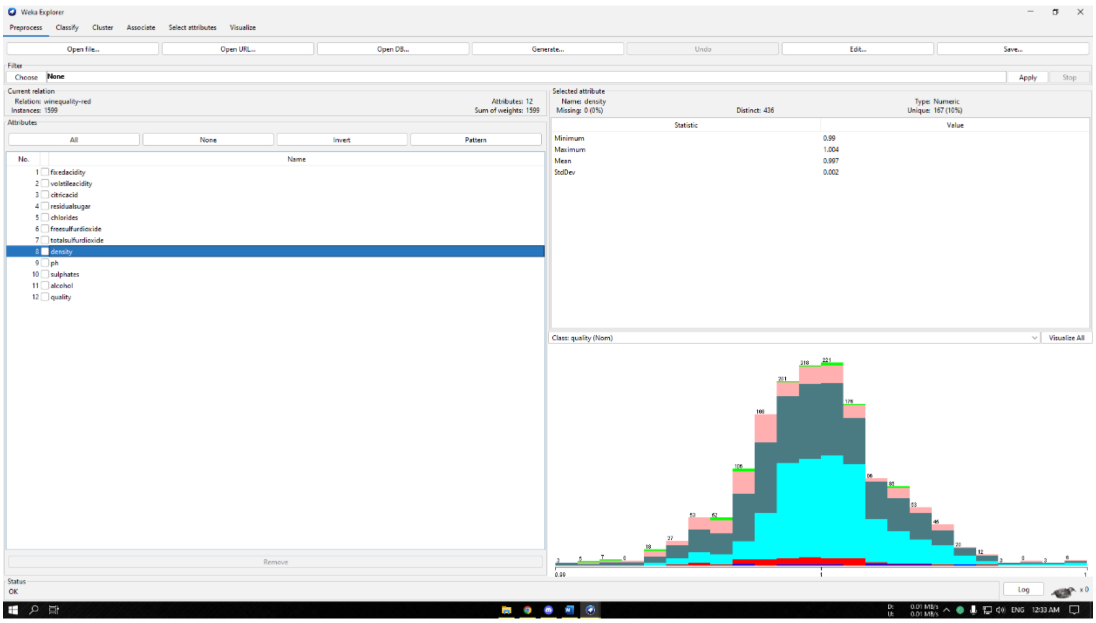


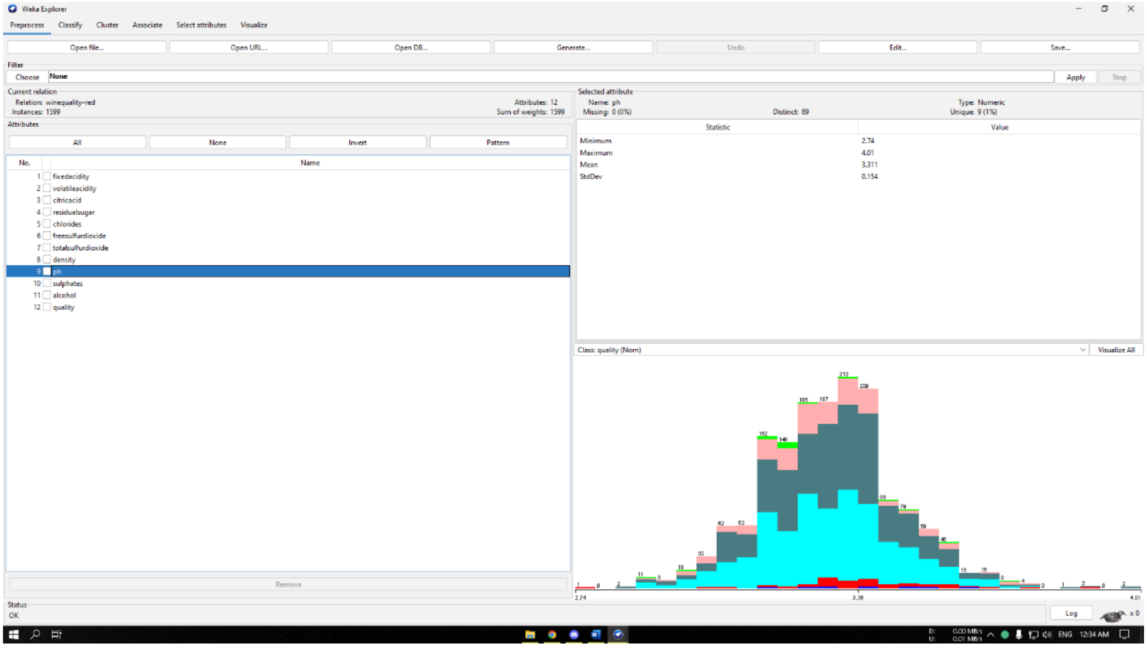


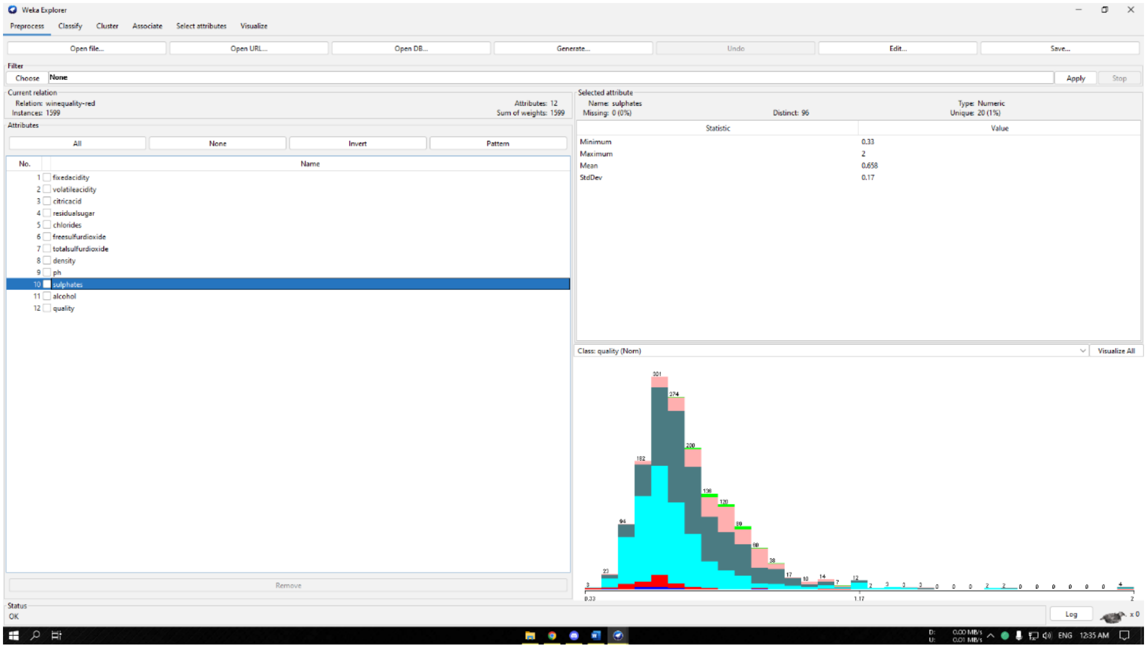


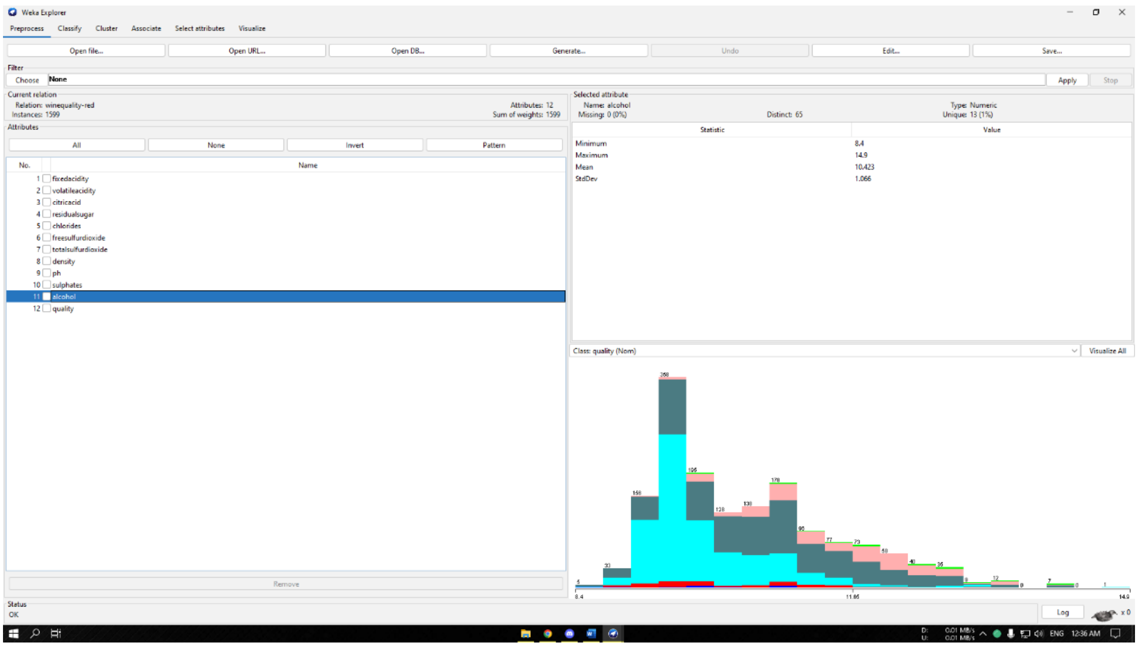

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Stop
Current relation
Relation: winequality-red
Instances: 1599
Attributes: 12
Sum of weights: 1599
Attributes
All
None
Invert
Pattern
No.
Name
1 fixedacidity
2 volatileacidity
3 citricacid
4 residualsugar
5 chlorides
6 freesulfurdioxide
7 totalsulfurdioxide
8 density
9 ph
10 sulphates
11 alcohol
12 quality
Remove
Selected attribute
Name: freesulfurdioxide
Missing: 0 (0%)
Distinct: 60
Type: Numeric
Unique: 10 (1%)
Statistic
Value
Minimum 1
Maximum 72
Mean 15.875
StdDev 10.46
Class: quality (Nom)
Visualize All
Status
OK
Log

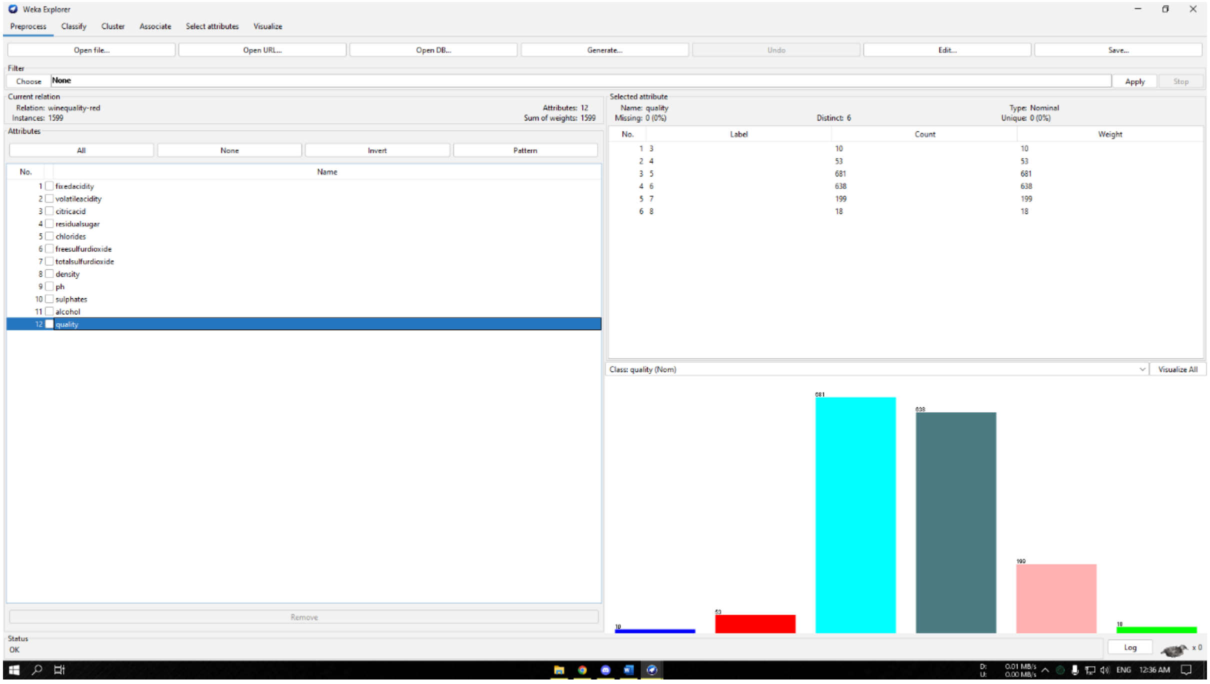

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Stop
Current relation
Relation: winequality-red
Instances: 1599
Attributes: 12
Sum of weights: 1599
Attributes
All
None
Invert
Pattern
No.
Name
1 fixedacidity
2 volatileacidity
3 citricacid
4 residualsugar
5 chlorides
6 freesulfurdioxide
7 totalsulfurdioxide
8 density
9 ph
10 sulphates
11 alcohol
12 quality
Remove
Selected attribute
Name: totalsulfurdioxide
Missing: 0 (0%)
Distinct: 144
Type: Numeric
Unique: 14 (1%)
Statistic
Value
Minimum 6
Maximum 289
Mean 46.468
StdDev 32.895
Class: quality (Nom)
Visualize All
Status
OK
Log


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Stop
Current relation
Relation: winequality-red
Instances: 1599
Attributes: 12
Sum of weights: 1599
Attributes
All
None
Invert
Pattern
No.
Name
1 fixedacidity
2 volatileacidity
3 citricacid
4 residualsugar
5 chlorides
6 freesulfurdioxide
7 totalsulfurdioxide
8 density
9 ph
10 sulphates
11 alcohol
12 quality
Remove
Selected attribute
Name: density
Missing: 0 (0%)
Distinct: 436
Type: Numeric
Unique: 167 (10%)
Statistic
Value
Minimum 0.99
Maximum 1.004
Mean 0.997
StdDev 0.002
Class: quality (Nom)
Visualize All
Status
OK
Log

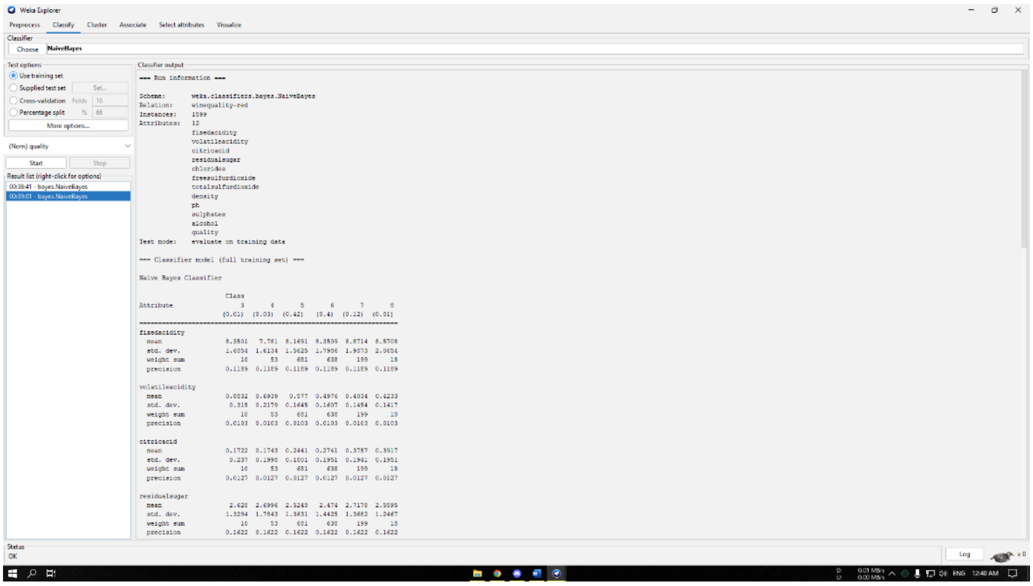

Weka Explorer
Relation: winequality-red
Instances: 1599
Attributes: 12
Name: ph
Missing: 0 (0%)
Distinct: 89
Type: Numeric
Unique: 9 (1%)
Minimum 2.74
Maximum 4.01
Mean 3.311
StdDev 0.154
Class: quality (Nom)

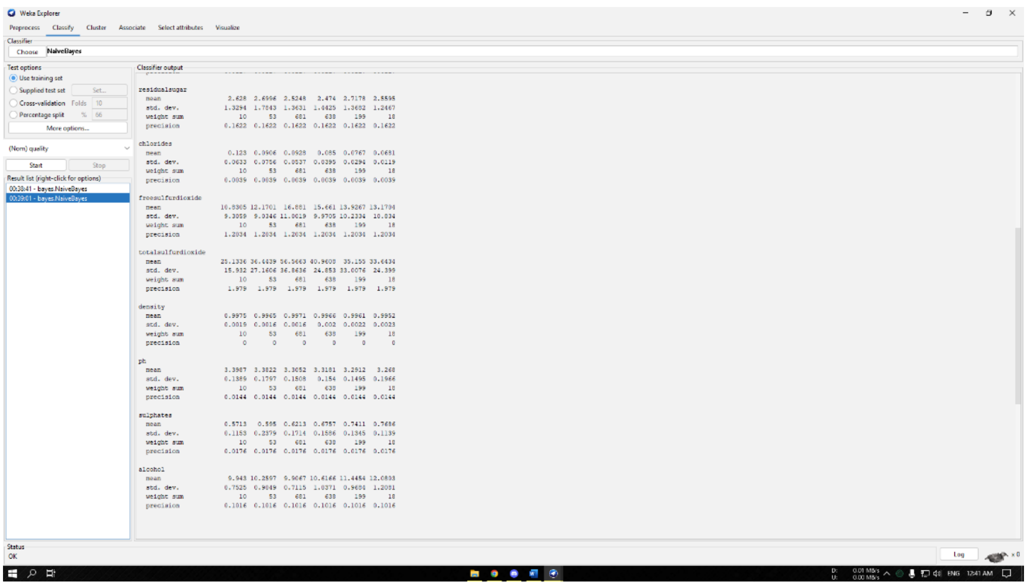

Weka Explorer
Relation: winequality-red
Instances: 1599
Attributes: 12
Name: sulphates
Missing: 0 (0%)
Distinct: 96
Type: Numeric
Unique: 20 (1%)
Minimum 0.33
Maximum 2
Mean 0.658
StdDev 0.17
Class: quality (Nom)

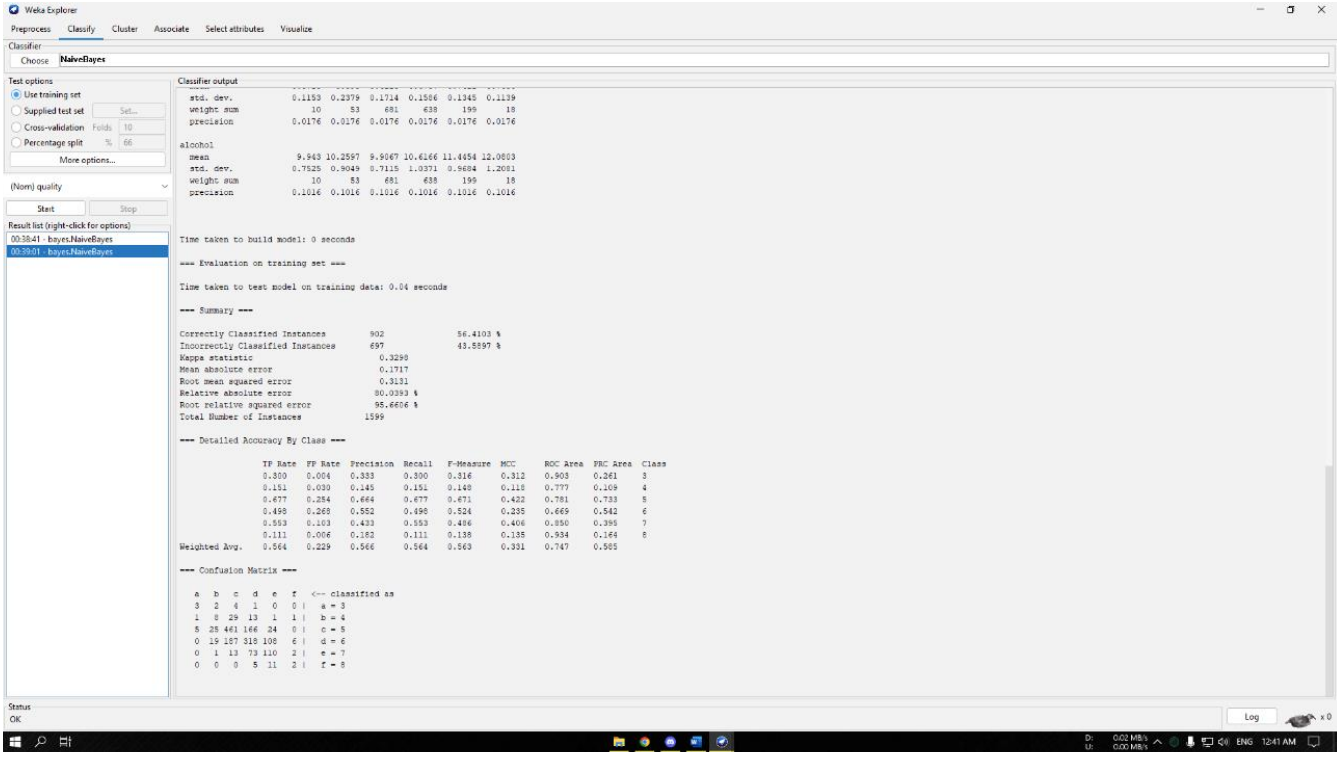

Weka Explorer
Relation: winequality-red
Instances: 1599
Attributes: 12
Name: alcohol
Missing: 0 (0%)
Distinct: 65
Type: Numeric
Unique: 13 (1%)
Minimum 8.4
Maximum 14.9
Mean 10.423
StdDev 1.066
Class: quality (Nom)



Evaluating Training set

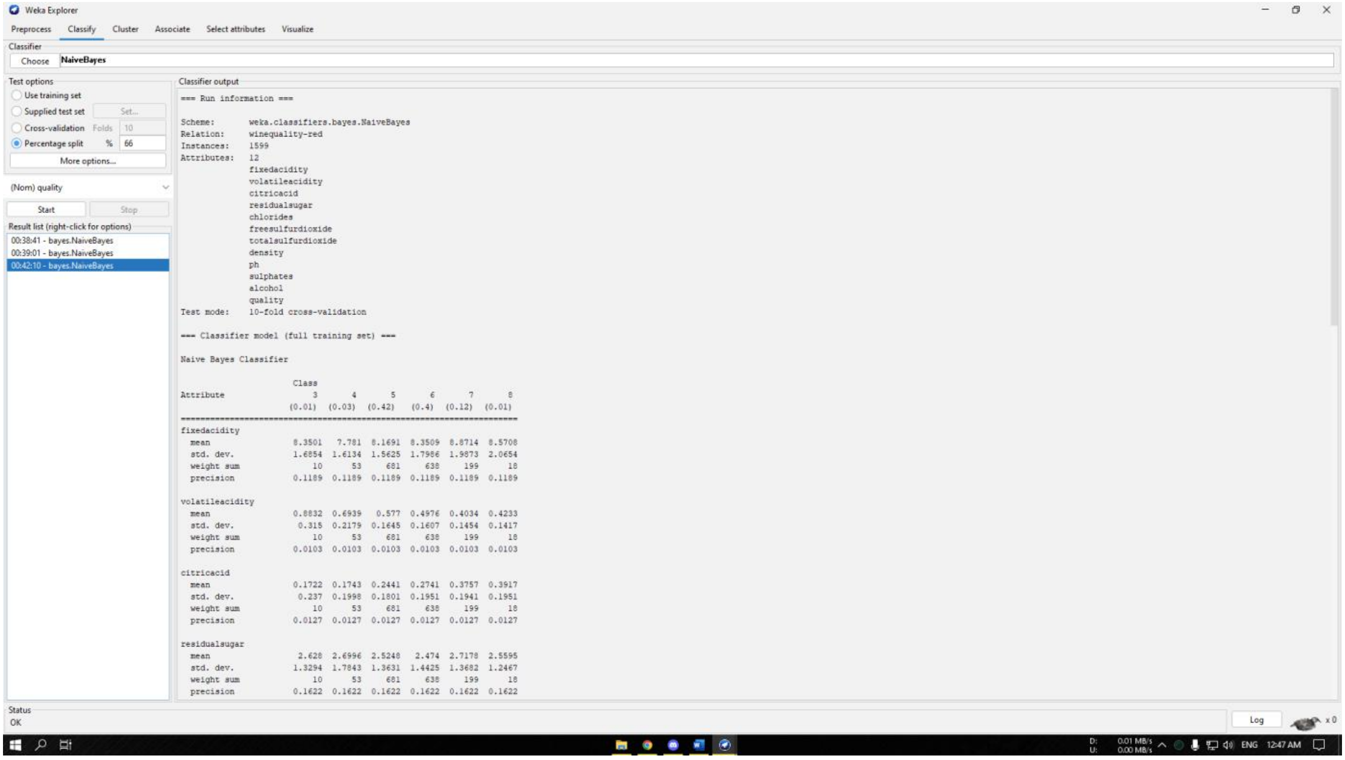


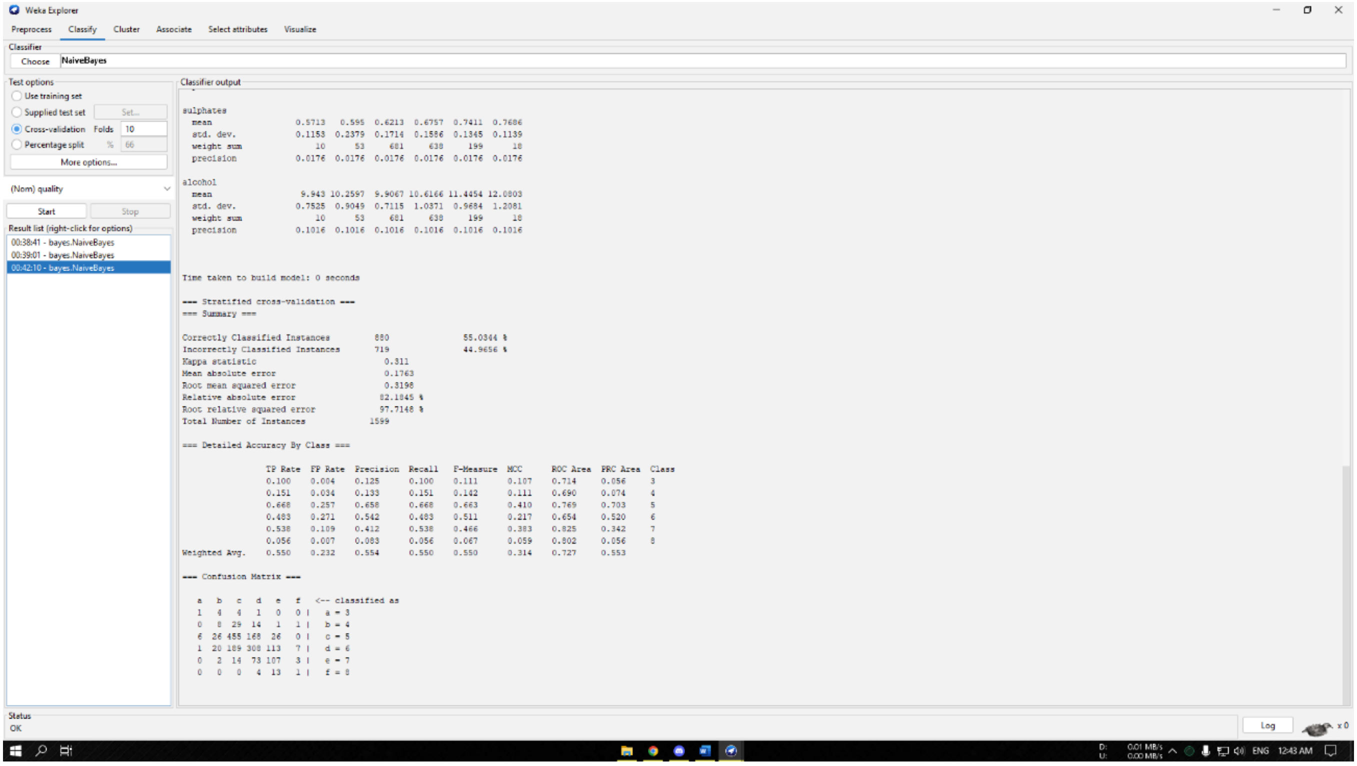




Naïve Bayes Algorithm Implementation

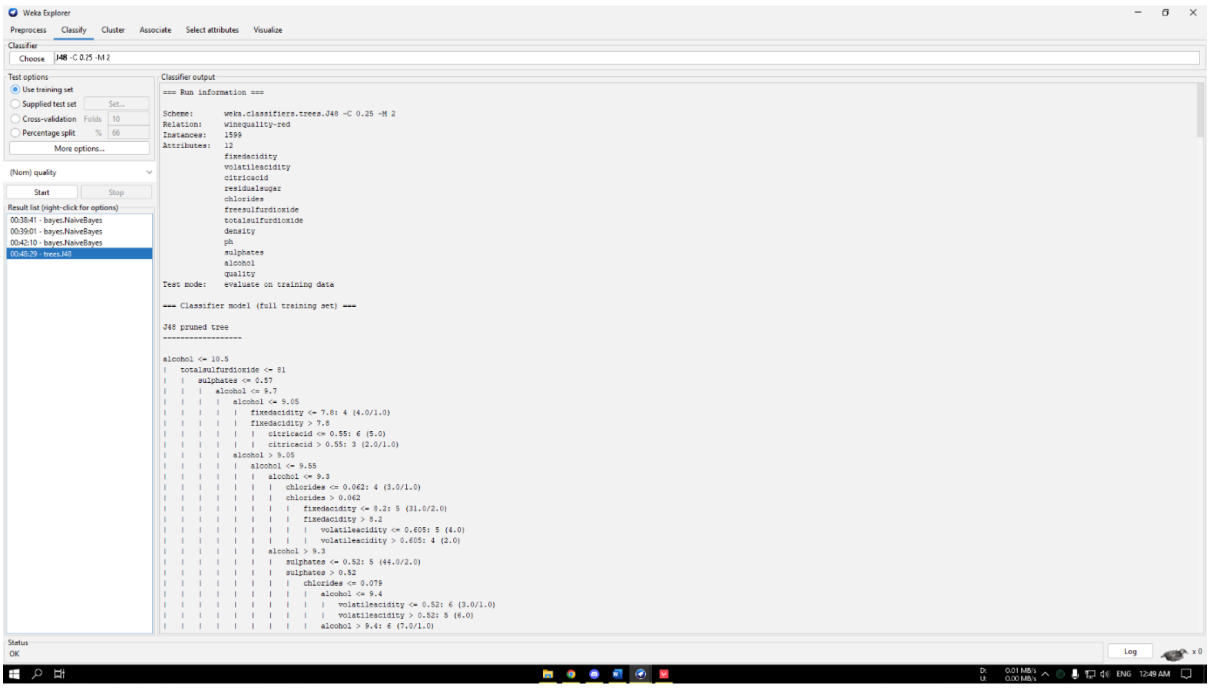
Naïve bayes is a classification algorithm which uses probability to classify the provided dataset. The result of this classification algorithm depends on categorical attributes. It combines the prior and posterior probability.

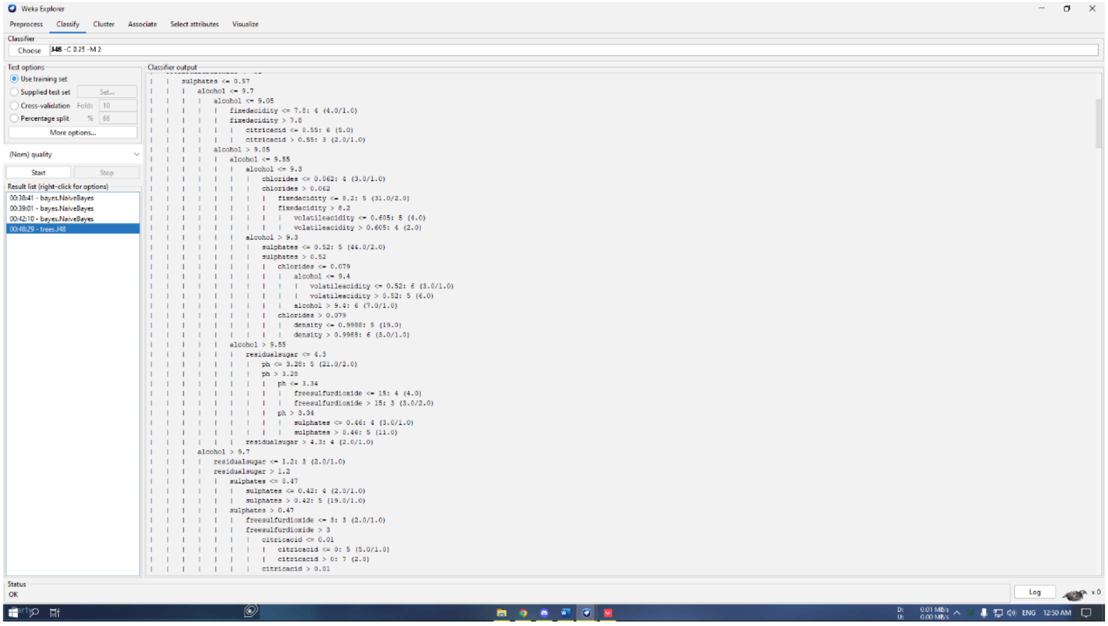


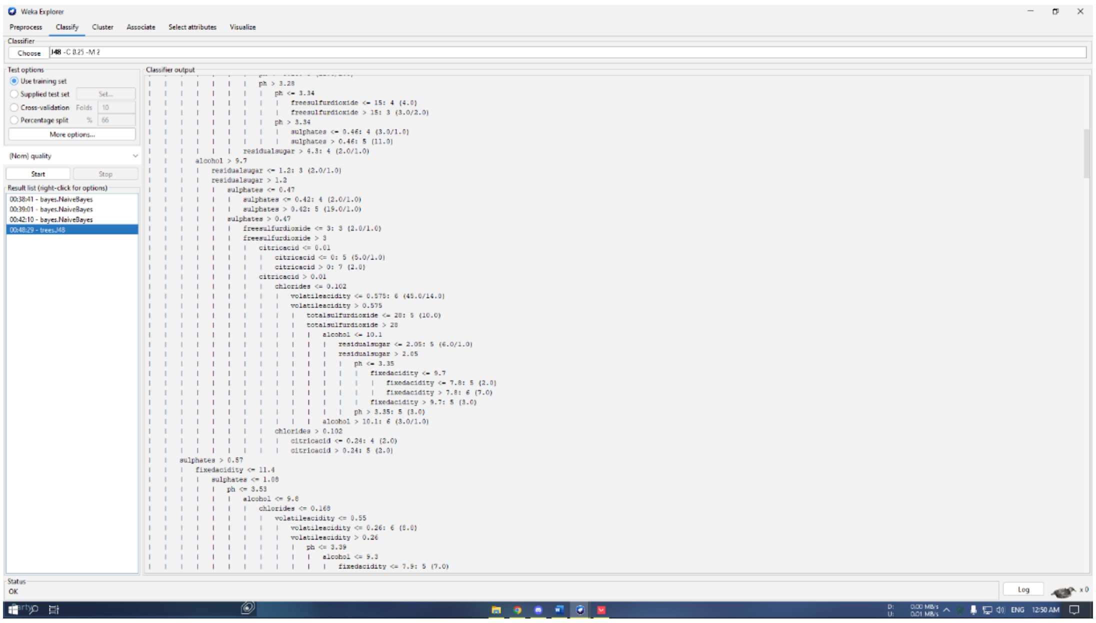


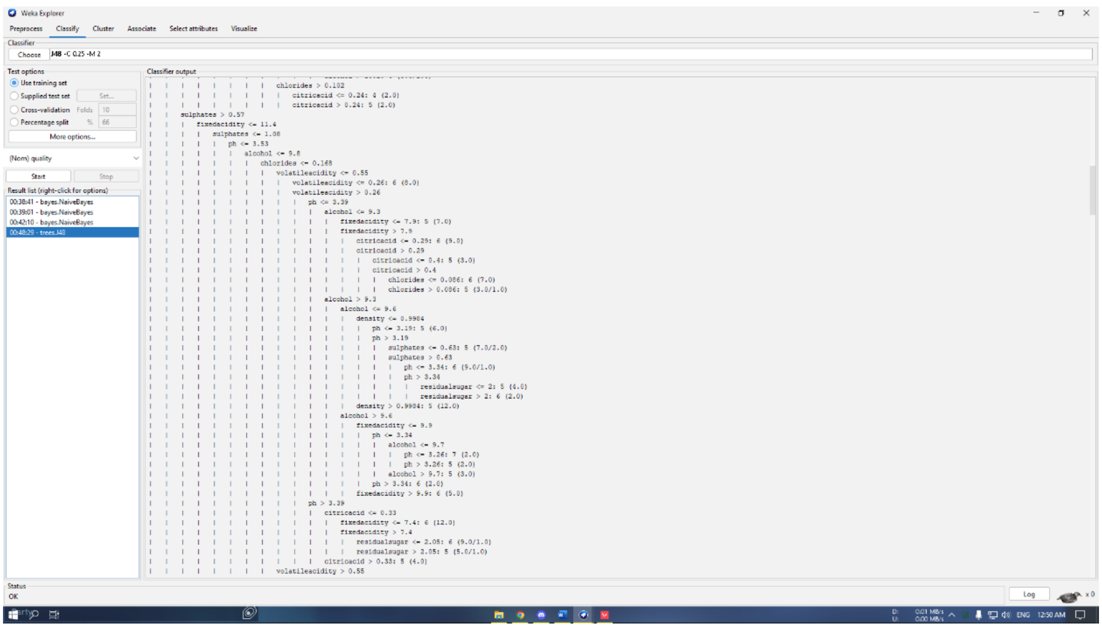
Decision Tree Implementation

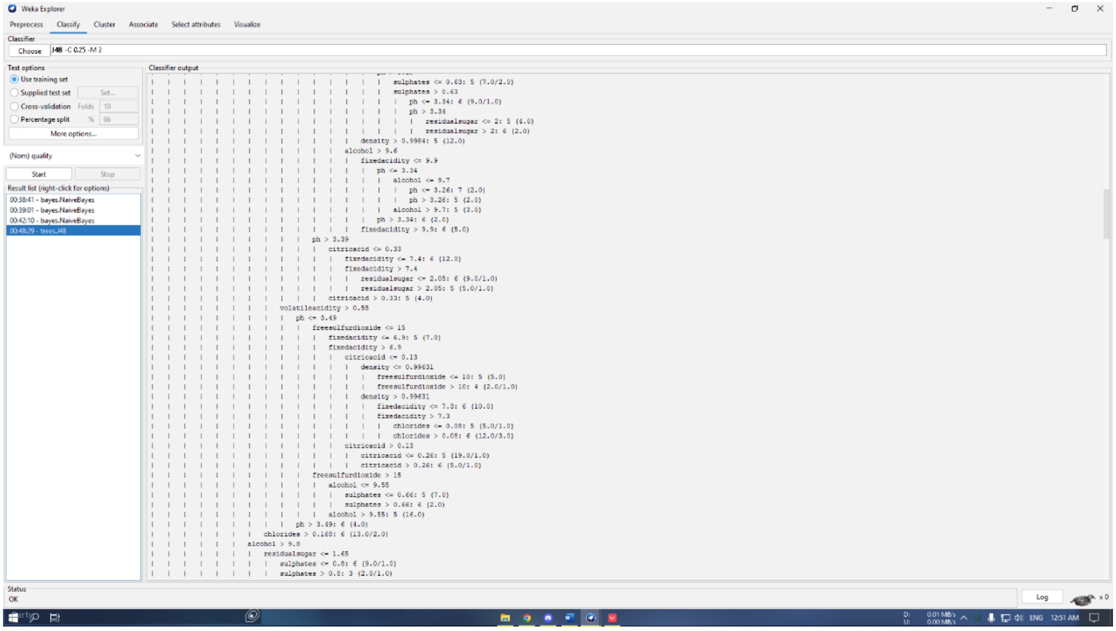
A decision tree is a class discriminator that recursively partitions the training set until each partition consists entirely or dominantly of examples from one class. Each non-leaf node of the tree contains a split point that is a test on one or more attributes and determines how the data is partitioned.

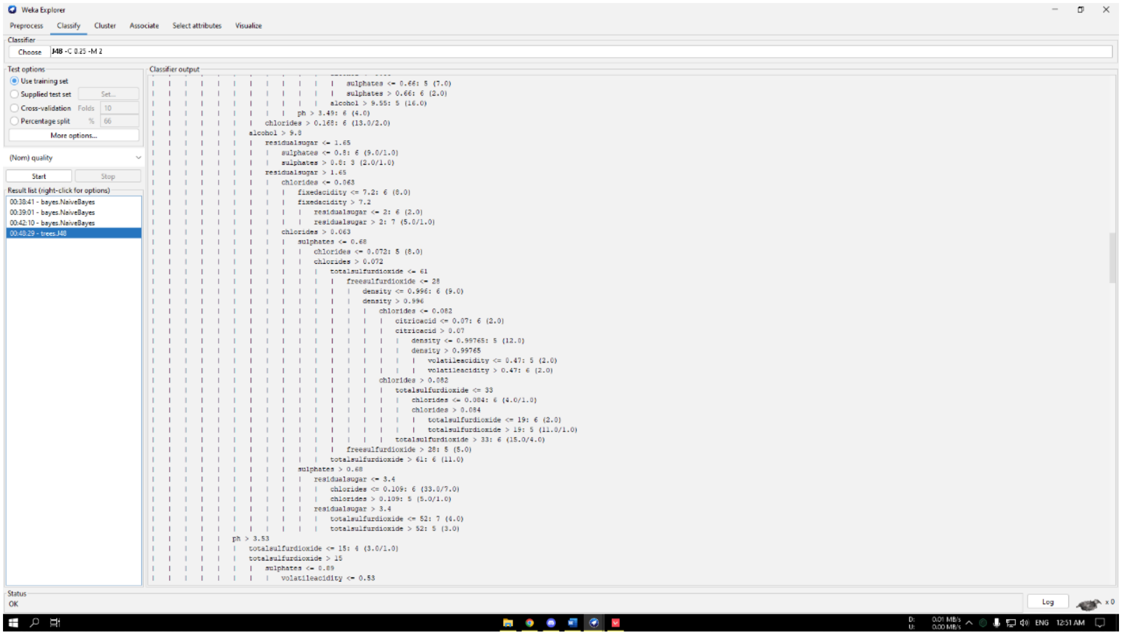


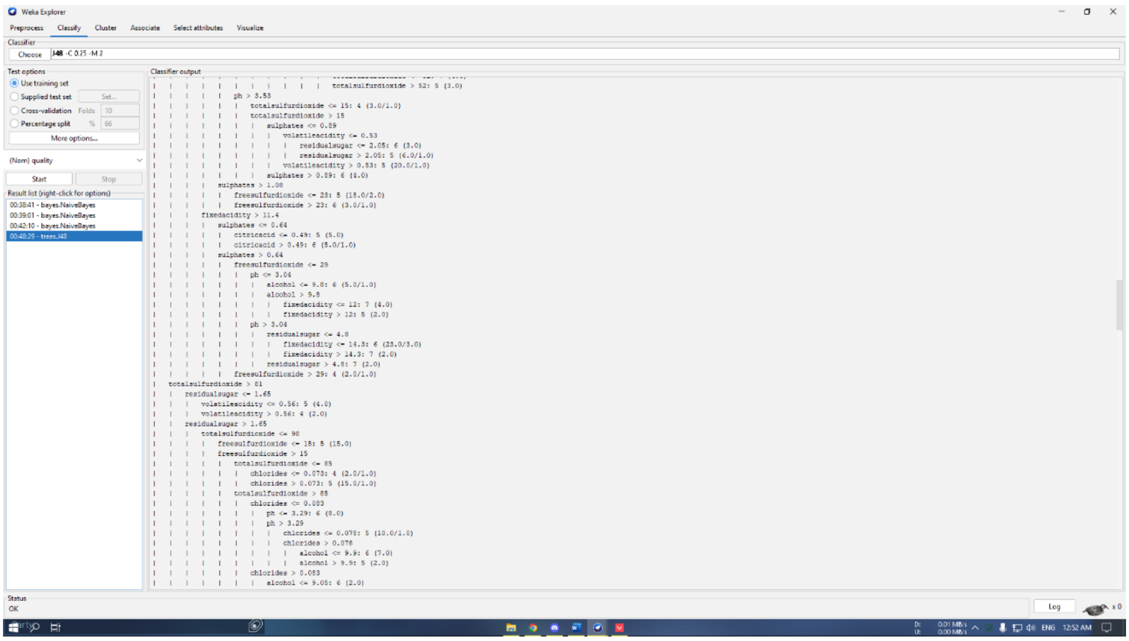

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Cross-validation
Percentage split
More options...
(Nom) quality
Start
Stop
Result list (right-click for options)
00:38:41 - bayes.NaiveBayes
00:39:01 - bayes.NaiveBayes
00:42:10 - bayes.NaiveBayes
00:48:29 - trees.J48
Classifier output
Status
OK
Log

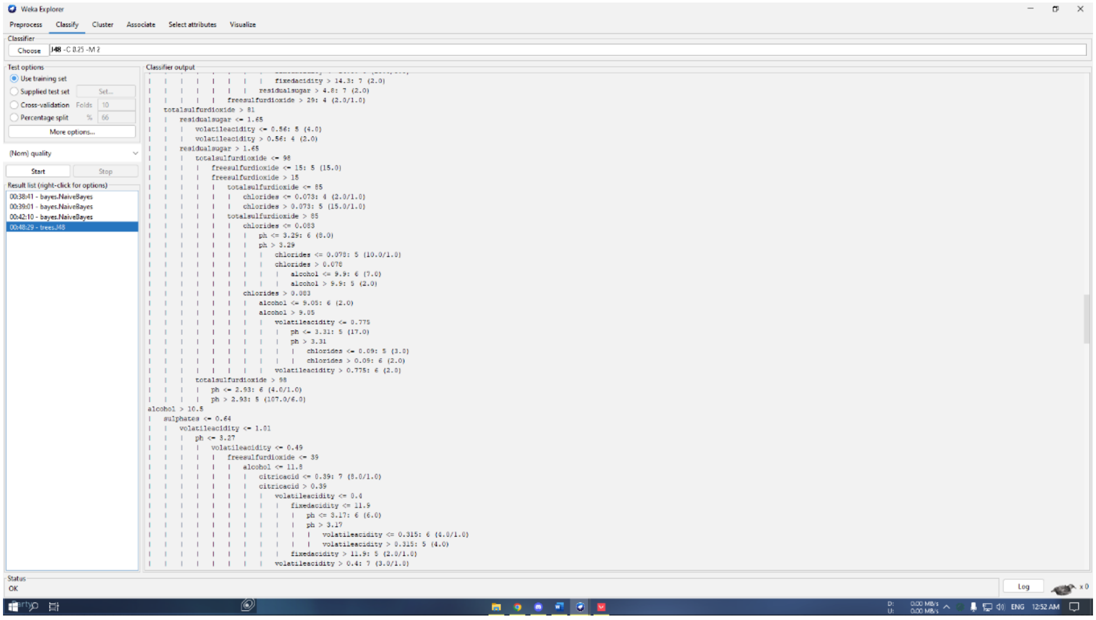

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Cross-validation
Percentage split
More options...
(Nom) quality
Start
Stop
Result list (right-click for options)
00:38:41 - bayes.NaiveBayes
00:39:01 - bayes.NaiveBayes
00:42:10 - bayes.NaiveBayes
00:48:29 - trees.J48
Classifier output
Status
OK
Log

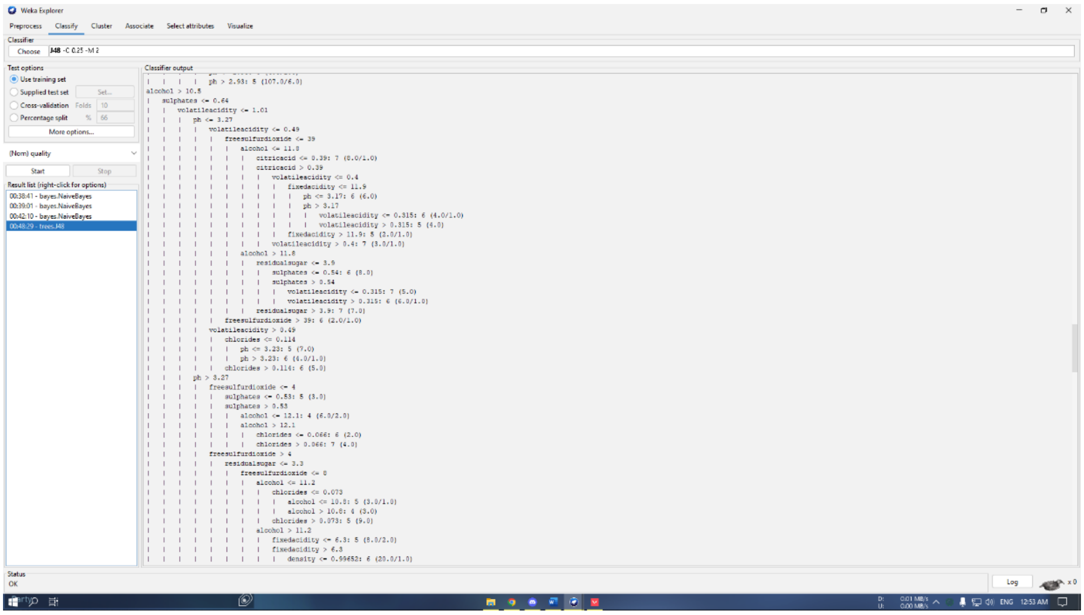

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Cross-validation
Percentage split
More options...
(Nom) quality
Start
Stop
Result list (right-click for options)
00:38:41 - bayes.NaiveBayes
00:39:01 - bayes.NaiveBayes
00:42:10 - bayes.NaiveBayes
00:48:29 - trees.J48
Classifier output
Status
OK
Log

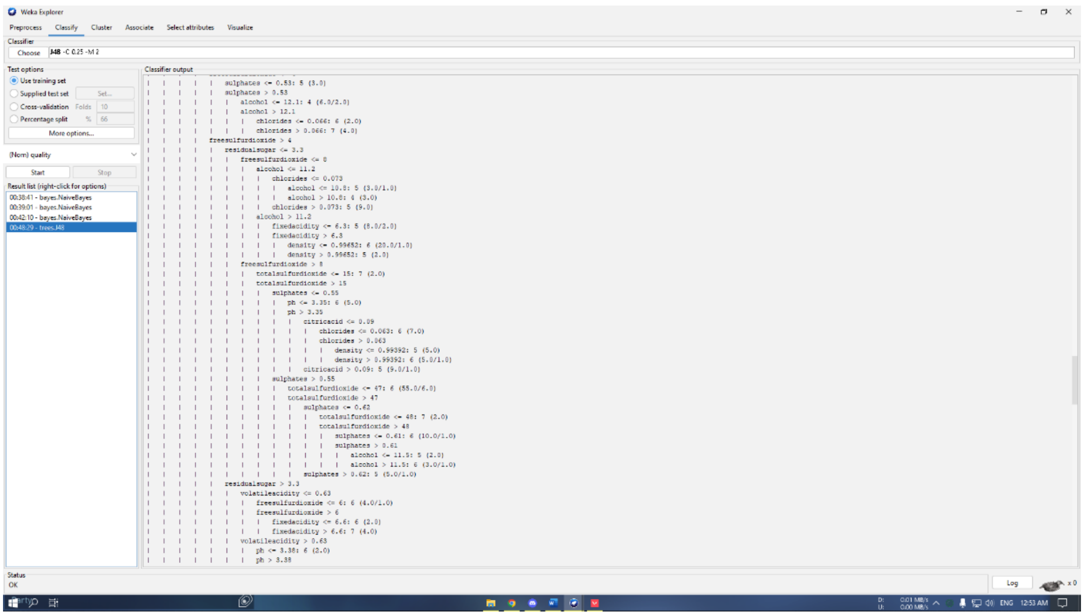

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Cross-validation
Percentage split
More options...
(Nom) quality
Start
Stop
Result list (right-click for options)
00:38:41 - bayes.NaiveBayes
00:39:01 - bayes.NaiveBayes
00:42:10 - bayes.NaiveBayes
00:48:29 - trees.J48
Classifier output
Status
OK
Log

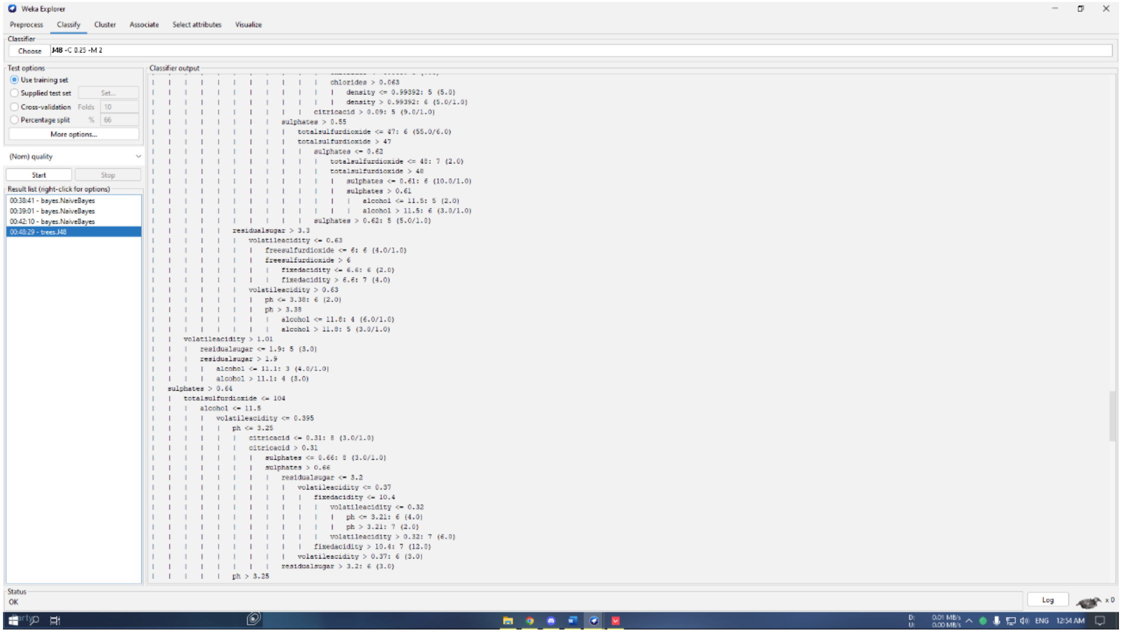

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Cross-validation
Percentage split
More options...
(Nom) quality
Start
Stop
Result list (right-click for options)
00:38:41 - bayes.NaiveBayes
00:39:01 - bayes.NaiveBayes
00:42:10 - bayes.NaiveBayes
00:48:29 - trees.J48
Classifier output
Status
OK
Log

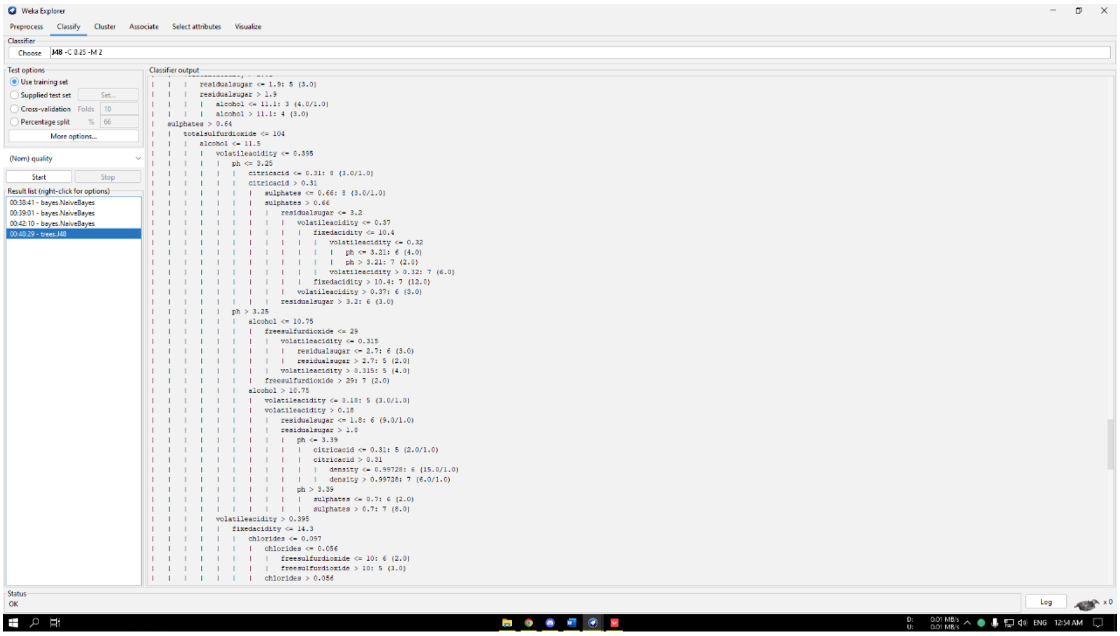

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Cross-validation
Percentage split
More options...
(Nom) quality
Start
Stop
Result list (right-click for options)
00:38:41 - bayes.NaiveBayes
00:39:01 - bayes.NaiveBayes
00:42:10 - bayes.NaiveBayes
00:48:29 - trees.J48
Classifier output
Status
OK
Log

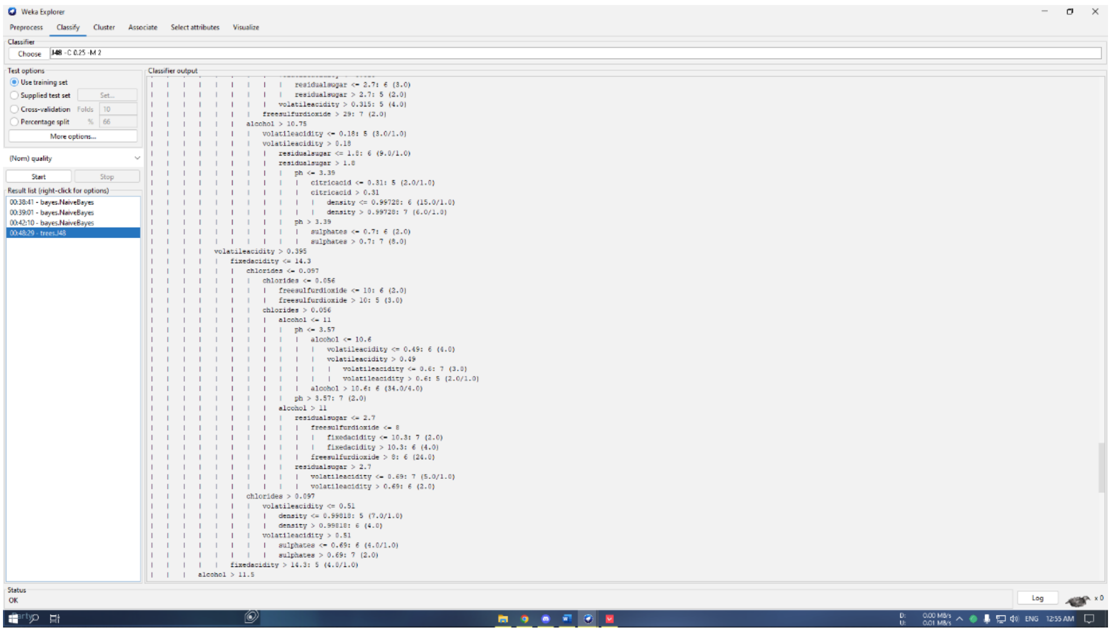


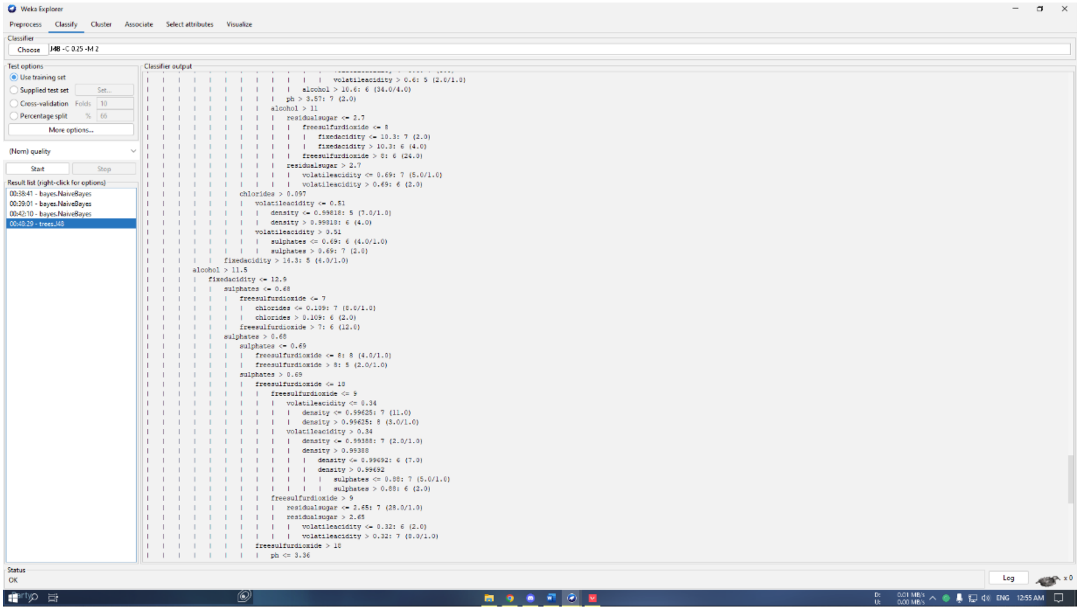


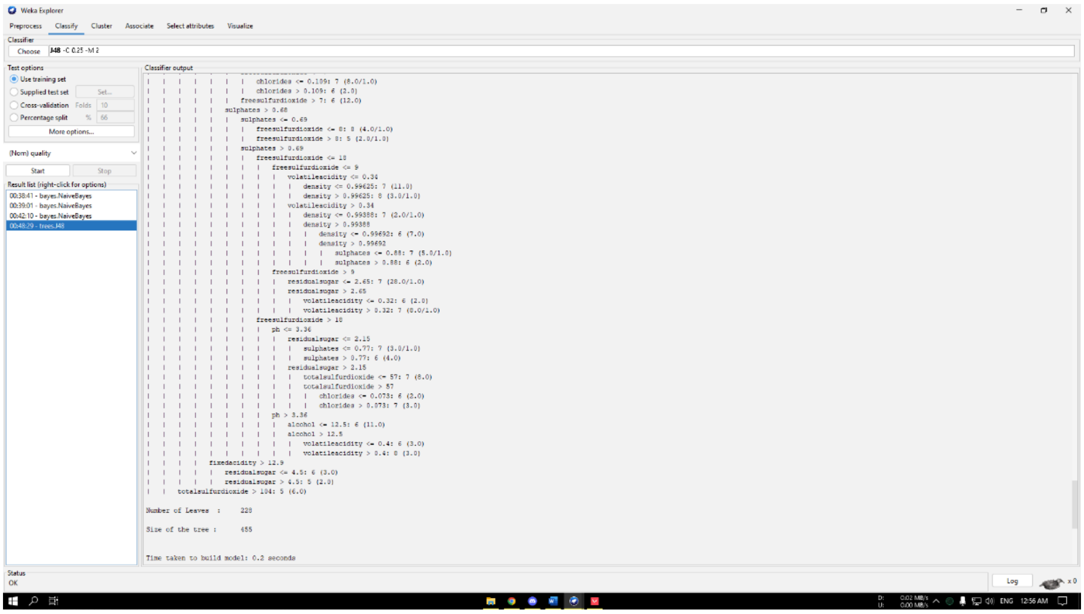


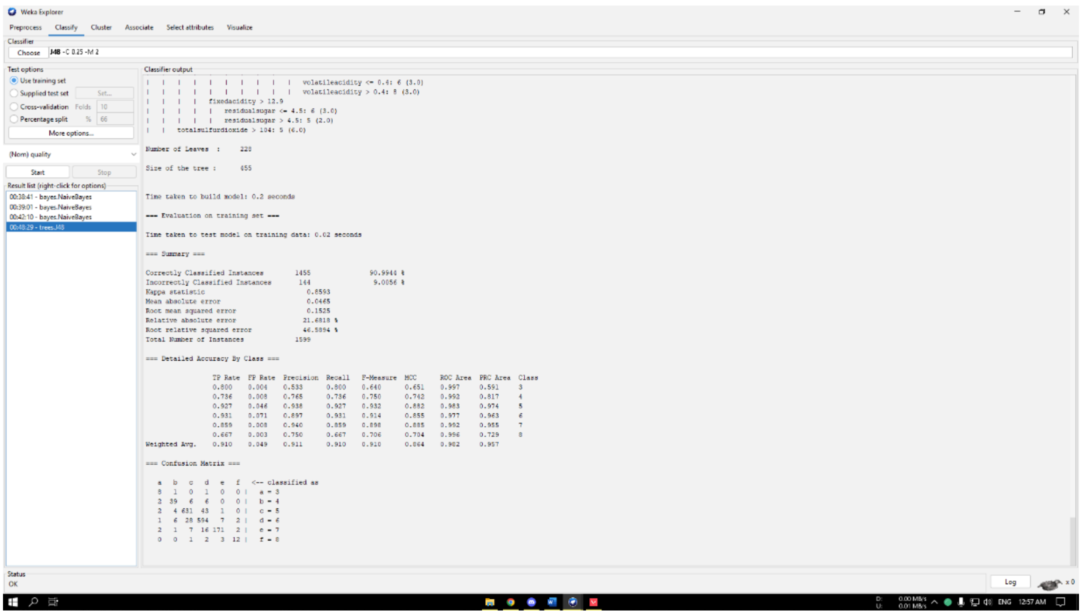

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Cross-validation
Percentage split
More options...
(Nom) quality
Start
Stop
Result list (right-click for options)
Classifier output
Status
OK
Log


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Cross-validation
Percentage split
More options...
(Nom) quality
Start
Stop
Result list (right-click for options)
Classifier output
Status
OK
Log


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Cross-validation
Percentage split
More options...
(Nom) quality
Start
Stop
Result list (right-click for options)
Classifier output
Status
OK
Log

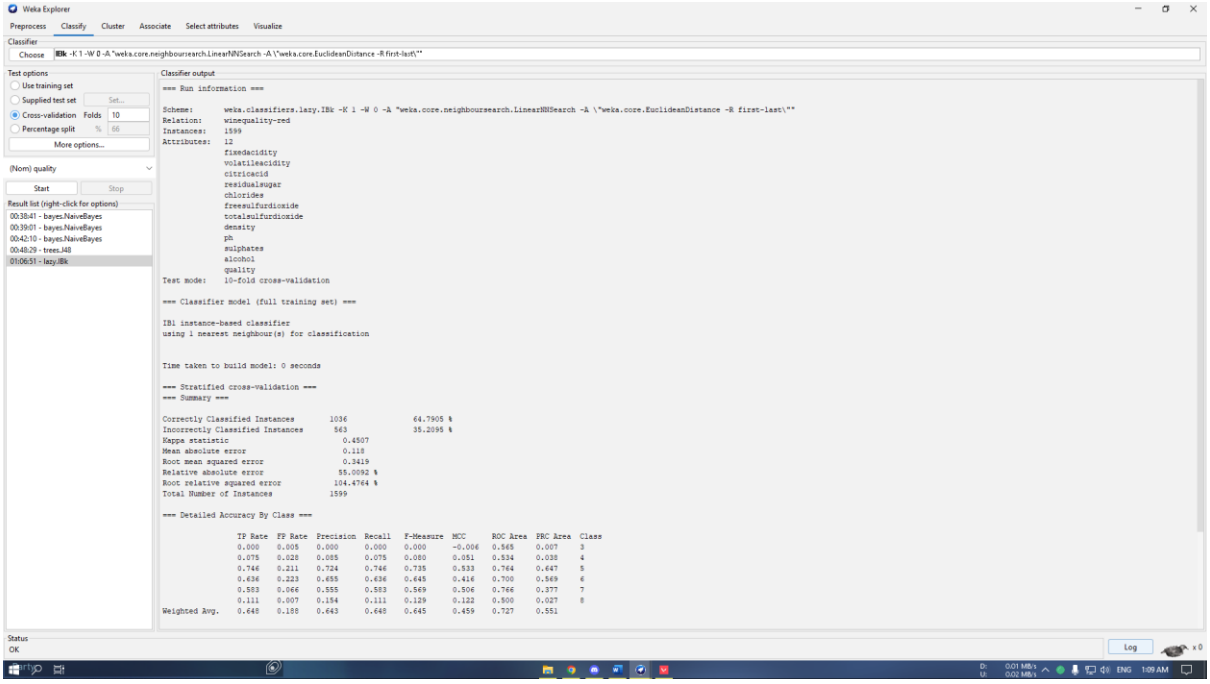


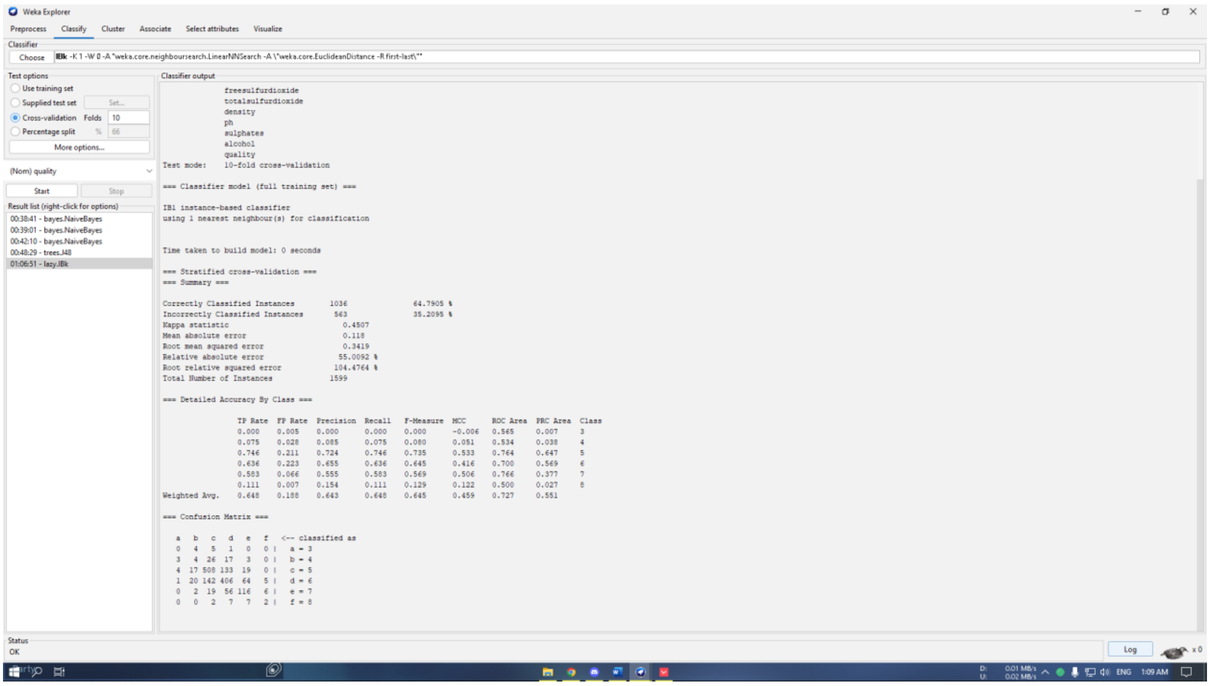



Weka Explorer
Number of Leaves : 228
Size of the tree : 455
Time taken to build model: 0.2 seconds
=== Evaluation on training set ===
Time taken to test model on training data: 0.02 seconds
=== Summary ===
Correctly Classified Instances 1455 90.9944 %
Incorrectly Classified Instances 144 9.0056 %
Kappa statistic 0.8593
Mean absolute error 0.0465
Root mean squared error 0.1525
Relative absolute error 21.6818 %
Root relative squared error 46.5894 %
Total Number of Instances 1599
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.800 0.004 0.533 0.800 0.640 0.651 0.997 0.591 3
0.736 0.008 0.765 0.736 0.750 0.742 0.992 0.817 4
0.927 0.046 0.938 0.927 0.932 0.882 0.983 0.974 5
0.931 0.071 0.897 0.931 0.914 0.855 0.977 0.963 6
0.859 0.008 0.940 0.859 0.898 0.885 0.992 0.955 7
0.667 0.003 0.750 0.667 0.706 0.704 0.996 0.729 8
Weighted Avg. 0.910 0.049 0.911 0.910 0.910 0.864 0.982 0.957
=== Confusion Matrix ===
a b c d e f <-- classified as
8 1 0 1 0 0 | a = 3
2 39 6 6 0 0 | b = 4
2 4 631 43 1 0 | c = 5
1 6 28 594 7 2 | d = 6
2 1 7 16 171 2 | e = 7
0 0 1 2 3 12 | f = 8

K-Nearest Neighbor (KNN) Algorithm Implementation

KNN is conceptually simple and has the advantage of being nonparametric. That is, the method can be used even when the variables are categorical—though if we are using numeric variables in the mix, it is best to standardize them to eliminate differences in scale. The challenge is that when the number of data points is very large special methods must be employed to rapidly search the space and find the "most similar" items.





Discussion & Conclusion

The analysis of the dataset using the WEKA tool comparison among data mining classification algorithms (Decision tree, KNN, Naive Bayas), shows that all KNN algorithms are more accurate and they have less error rate and they are easier algorithms as compared to the Decision tree and Naive Bayas. The result of implementation in WEKA on the same dataset showed that the Decision Tree outperforms and Bayesian classification are less than the accuracy of KNN. The comparative study has shown that each algorithm has its own set of advantages and disadvantages as well as its own area of implementation. None of the algorithms can satisfy all constraints and criteria. Depending on the application and requirements, a specific algorithm can be chosen. We think KNN will be the right choice for this dataset according to the WEKA result.

Naïve Bayes

```
=== Confusion Matrix ===

   a   b   c   d   e   f   <-- classified as
   1   4   4   1   0   0 |   a = 3
   0   8  29  14   1   1 |   b = 4
   6  26 455 168  26   0 |   c = 5
   1  20 189 308 113   7 |   d = 6
   0   2  14  73 107   3 |   e = 7
   0   0   0   4  13   1 |   f = 8
```

K-Nearest neighbor Algorithm

```
=== Confusion Matrix ===

   a   b   c   d   e   f   <-- classified as
   0   4   5   1   0   0 |   a = 3
   3   4  26  17   3   0 |   b = 4
   4  17 508 133  19   0 |   c = 5
   1  20 142 406  64   5 |   d = 6
   0   2  19  56 116   6 |   e = 7
   0   0   2   7   7   2 |   f = 8
```

Decision Tree Algorithm

```
=== Confusion Matrix ===

   a   b   c   d   e   f   <-- classified as
   1   3   2   3   1   0 |   a = 3
   4   7  23  16   3   0 |   b = 4
   2  24 478 159  18   0 |   c = 5
   3  19 168 393  50   5 |   d = 6
   1   4  22  65 100   7 |   e = 7
   0   0   0  11   7   0 |   f = 8
```

Comparative analysis

Total number of instances **1599**

Perspective \ Algorithm	Naïve Bayes	K-Nearest Neighbor	Decision Tree
Correctly Classified Instances	880	1036	979
Incorrectly Classified Instances	719	563	620
Relative absolute Error	82.1845%	55.0092%	63.6062%
Root relative squired Error	97.7148%	104.4764%	101.8383%

Table: Accuracy comparison table

References

1. https://www.kaggle.com/datasets/uciml/red-wine-quality-cortez-et-al-2009
2. https://datauab.github.io/red_wine_quality/
3. https://drive.google.com/file/d/1CiRl6IqOM77SdjKvNGAJCdAzUCDBaXWG/view?usp=sharing