Name - Pratik Bhujade

Part 1 – Classification

1) Description of dataset and findings

Title : Wine Quality Dataset

Wine quality dataset is collected from Portuguese "Vinho Verde" wine. It contains physicochemical analysis and sommelier evaluation of 6497 wine.

Objective : Intention to use the data [Cortez et al., 2009] was to show how different physicochemical parameters effects wine quality and sommelier evaluation.

Data Description : Dataset is very useful for ML applications and can be used to compare different ML and even shallow NNs. Dataset has 13 attributes: 1 Nominal and 12 numeric. I will use this dataset for both problems: classification and clustering.

Attributes:	**Type:**	**Range**:
type	Nominal	(red/white)
fixed acidity	Numeric	3.8 -15.9
volatile acidity	Numeric	0.08-1.58
citric acid	Numeric	0-1.66
residual sugar	Numeric	0.6-65.8
chlorides	Numeric	0.009-0.611
free sulfur dioxide	Numeric	1-289
total sulfur dioxide	Numeric	6-440
density	Numeric	0.987-1.039
pH	Numeric	2.72-4.01
sulfates	Numeric	0.22-2
alcohol	Numeric	9-14.9
quality	Numeric	3-9

Description of attributes:
0 – type: wine type, Red or White.

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

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

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

4 - 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

5 - chlorides: the amount of salt in the wine

6 - 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

7 - 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

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

9 - 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

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

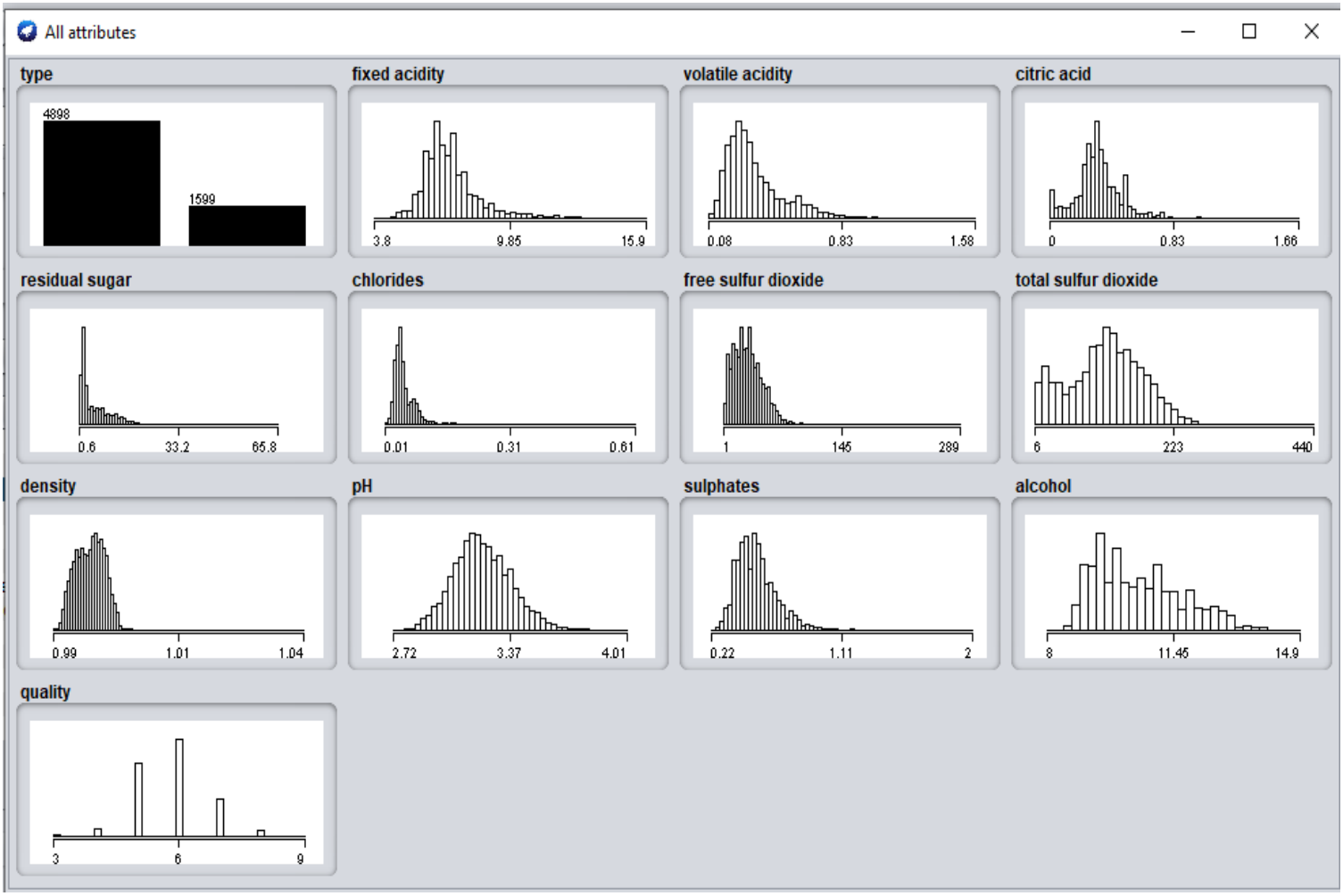
11 - alcohol: the percent alcohol content of the wine

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

Summary of findings : Dataset has 2 attributes, that can be used as label: type and quality. I will focus on wine type and in experiments. **My goal is to show that it is possible to distinguish wine color by**

phychochemical parameters. I will not use quality as a parameter as it is not phychochemical.

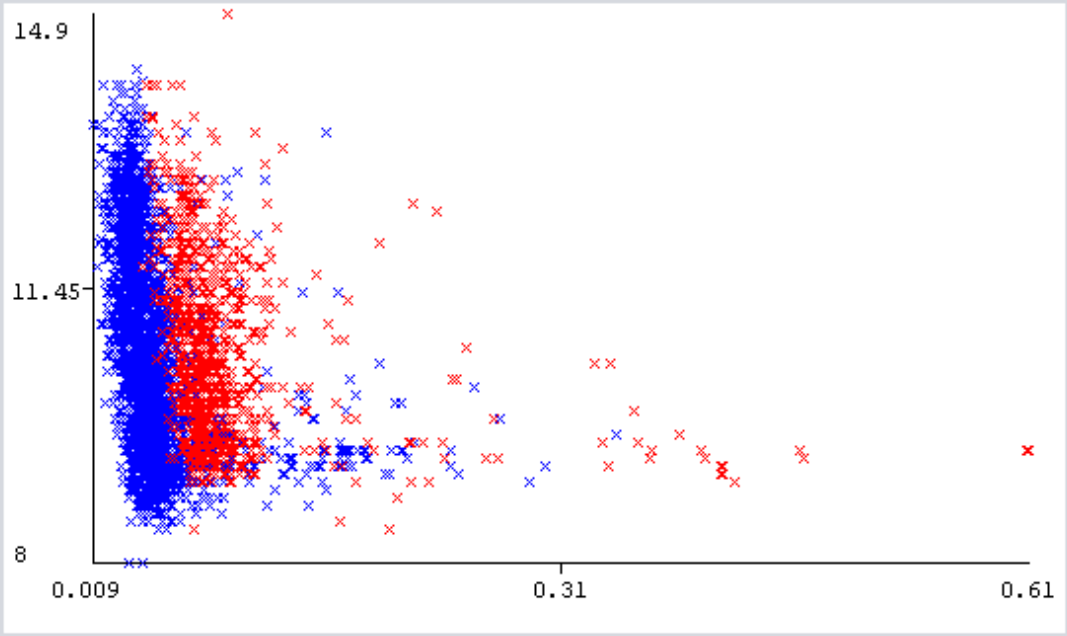
Fig 1:



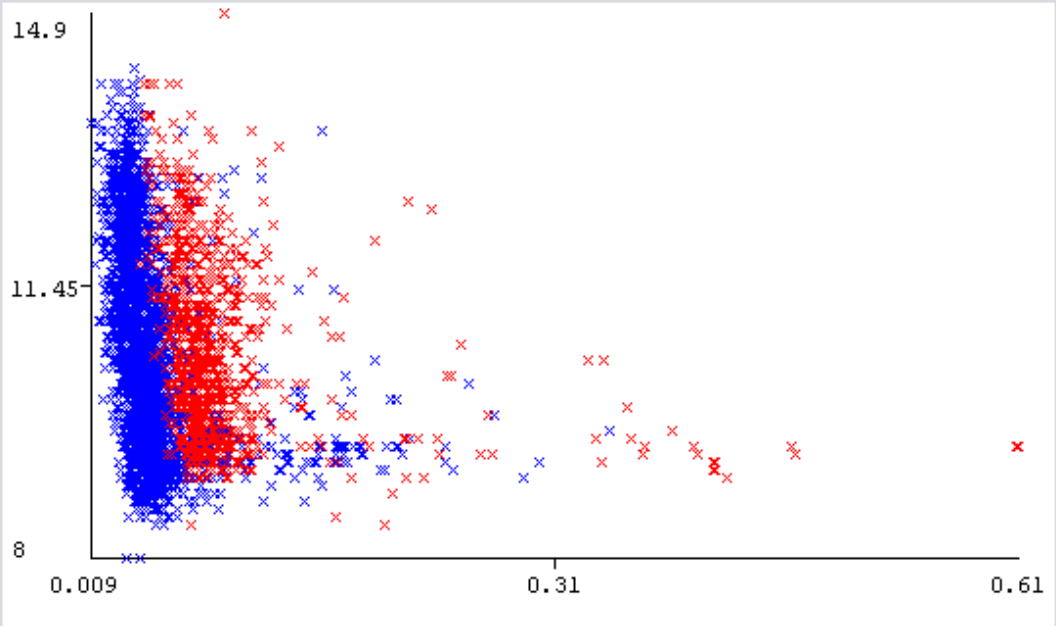
Most of dataset attributes have good distribution and do not have outliers. Dataset includes some repeated instances. I will deal with it in the preprocessing steps.

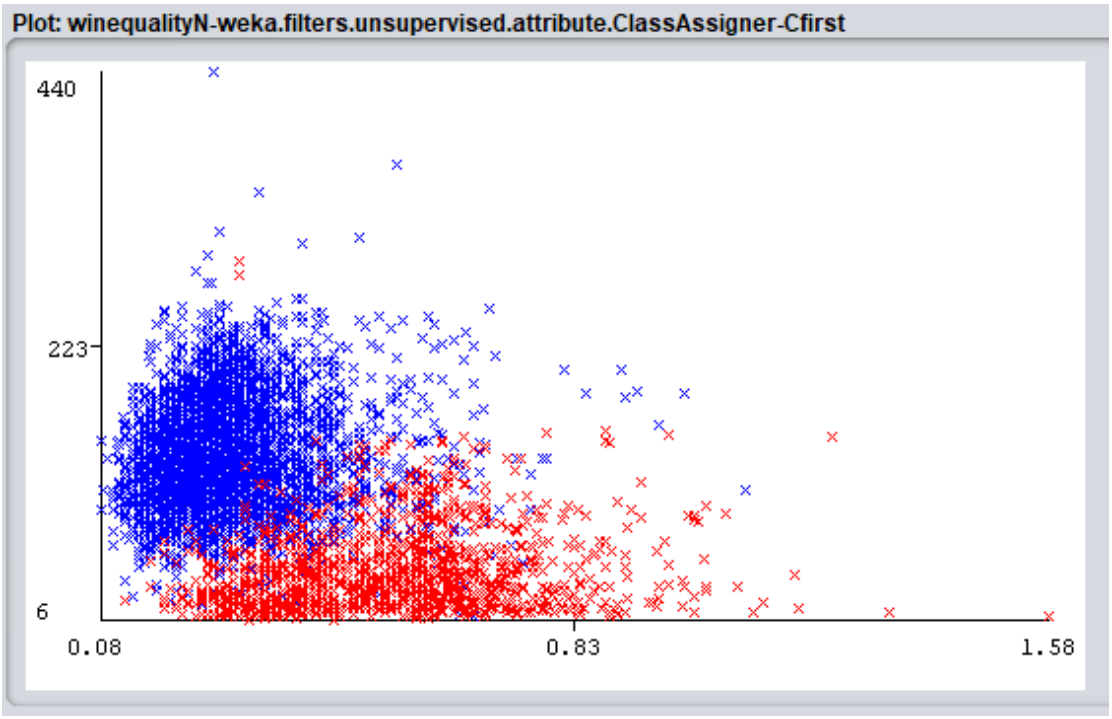
Fig 2:

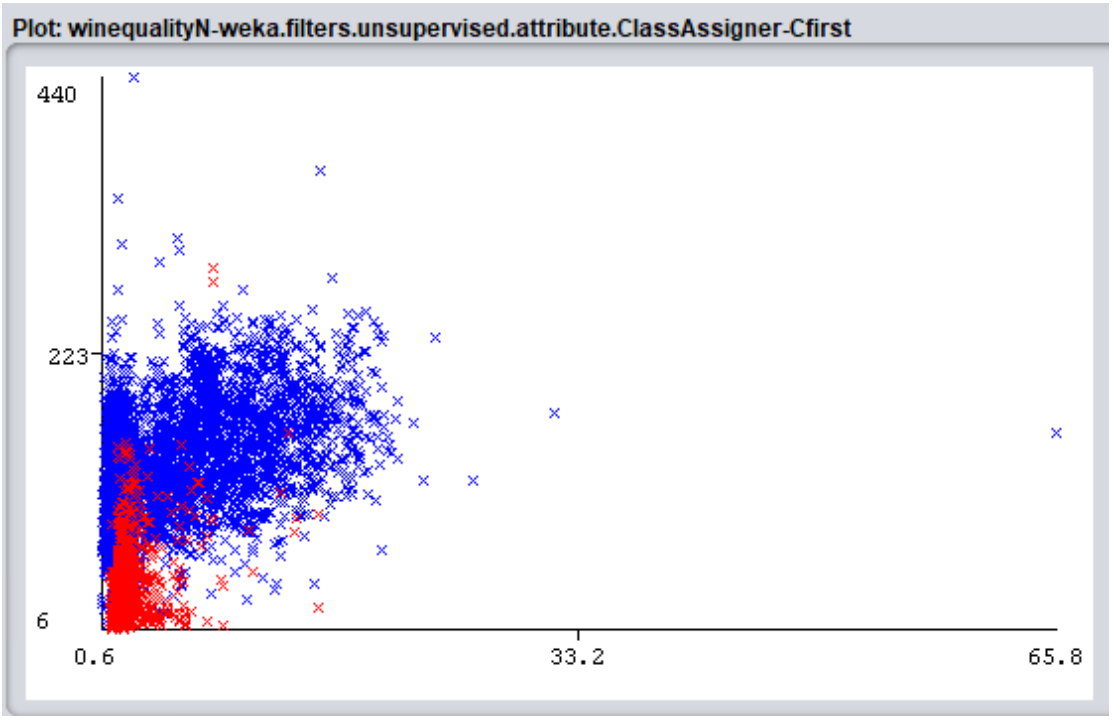
Plot: winequalityN-weka.filters.unsupervised.attribute.ClassAssigner-Cfirst



Plot: winequalityN-weka.filters.unsupervised.attribute.ClassAssigner-Cfirst







From visualizations its visible that wine type is not randomly distributed and clustering algorithms should be able to achieve good precision.

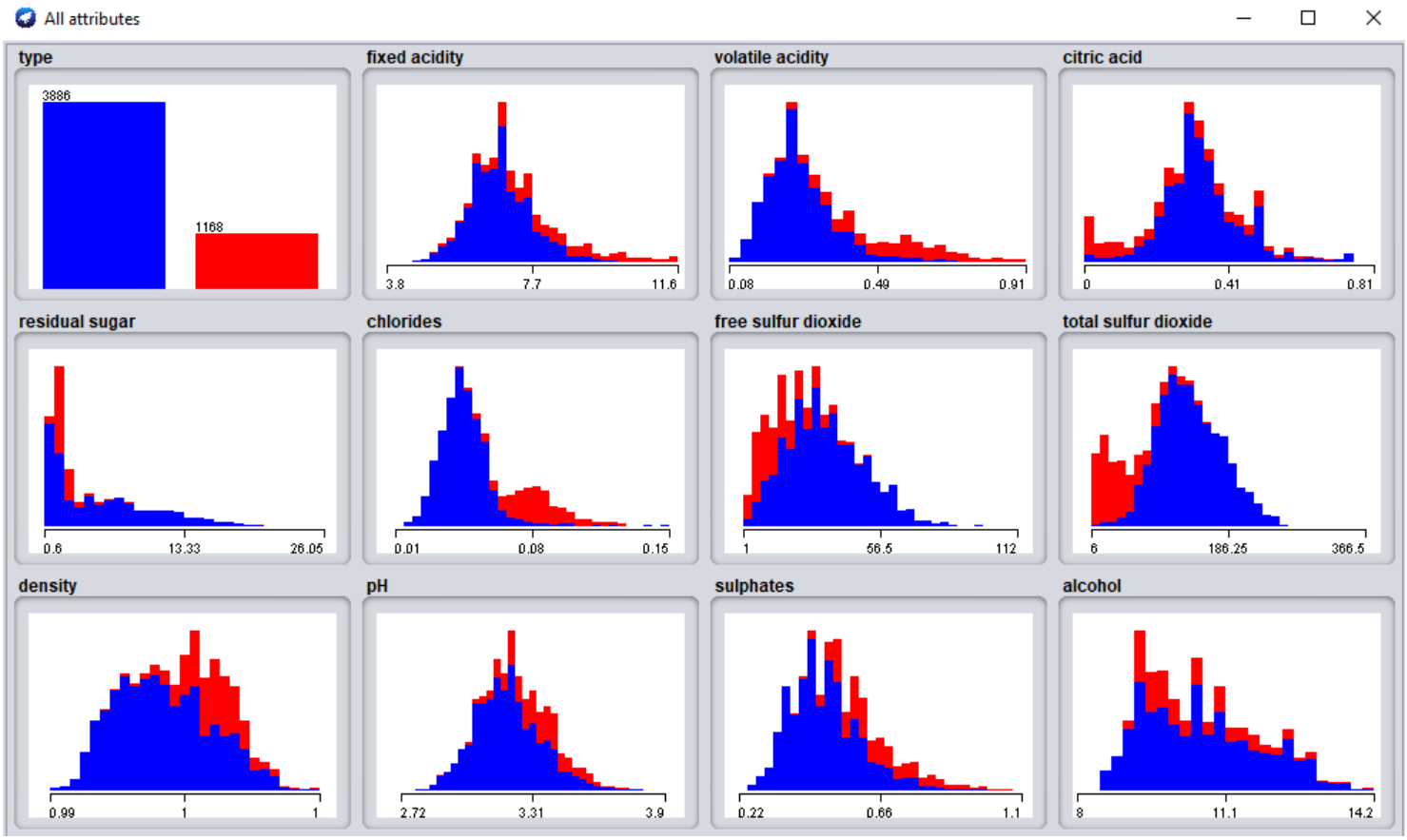
I used the j48 model for two differently preprocessed dataset (see step 8 in preprocessing). For numeric attributes, j48 gives 99.2% accuracy on test data. It turns out that only two parameters (total sulfur dioxide and chlorides) can give 95% accuracy. For second preprocessing I converted all numeric values to 3 nominal: High, Medium, Low. Despite losing a lot of information j48 give 95% accuracy. Converting numeric attributes to nominal was necessary for the Apriori algorithm and running on j48 allowed comparing the results of those two models. Most of the rules

derived from the Apriori association were already seen in the j48 tree. As white wine has unique low values for several attributes, association rules were mostly for white wine, but for j48 rules for red wine are included in the tree.

2) Preprocessing steps:

1) Remove **quality** from the attributes as I will not use it as a parameter
2) Assign class to the **type**(Red/White) as I intend to use it as label.
3) Remove duplicates: Removing reduces dataset size from 6497 to 5329
4) Replace missing values with mean of the attributes
5) Using Interquartile Range to detect outliers
6) Using Remove with values to remove outliers:

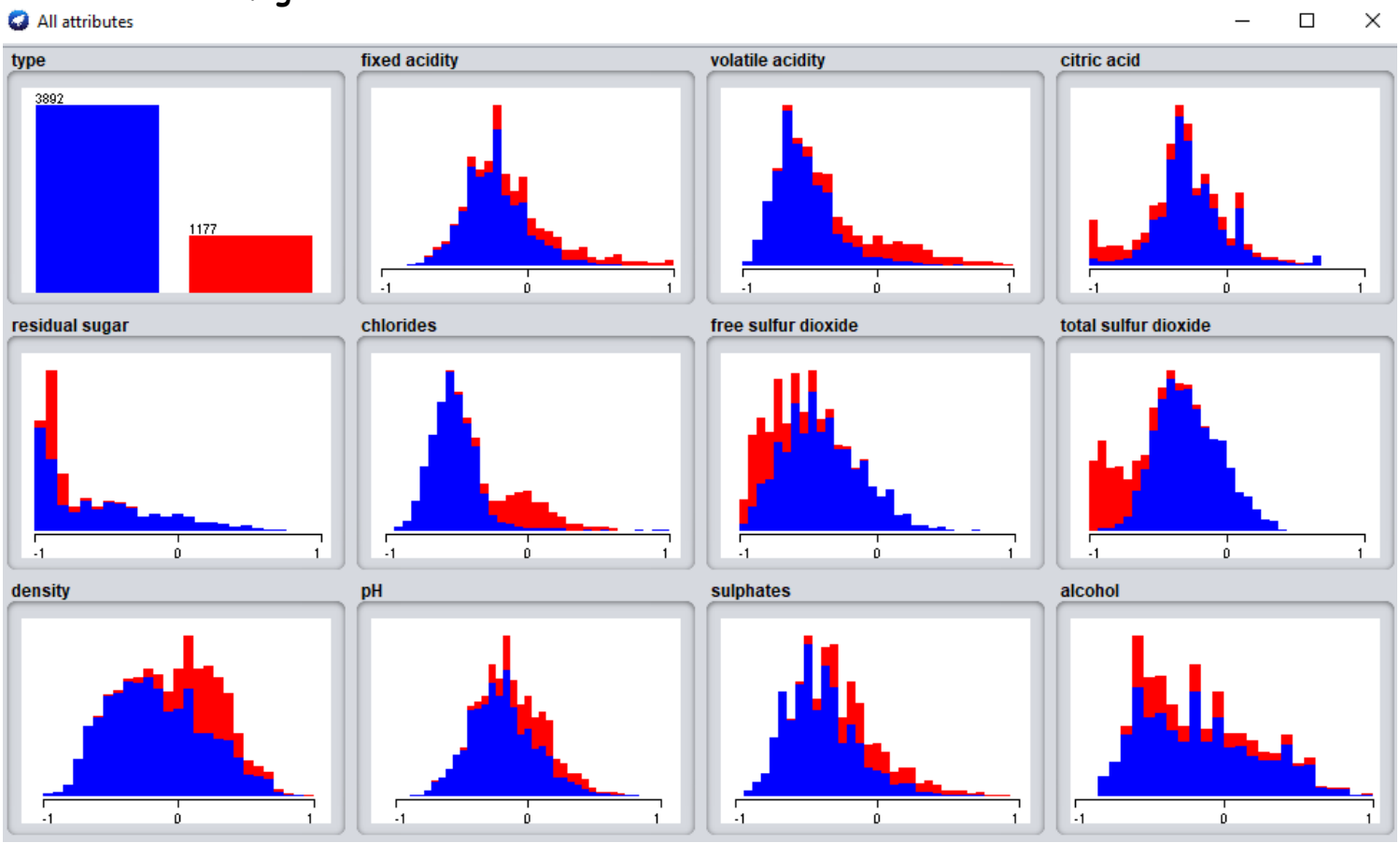
Dataset overview after removing outliers.
Fig 3:



Difference from **Fig 1** is clear as distributions does not have long tails.

7) Normalize in the range [-1,1]. For ML algorithms most of the cases normalization is important as attribute values can differ in order of magnitudes.
8) Discretize all numerical values to 3 nominals. This will allow us to use Association Rules. (I have used j48 without step 8 and other models with step 8)
9) Randomize before splitting to avoid only white wine in the test set

After those steps dataset includes 5069 instances. Some patterns are already visible in the data (fig 3). Dataset is ready for split. Final dataset is on **Fig 4:**



3) Divided the dataset into training and test set

Divided the dataset into training and testing data sets (9:1).

- **trainingSet.arff**
- **testingSet.arff**

Training set:

Fig 5 -6-7

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... | Open URL... | Open DB... | Generate... | Undo | Edit... | Save...

Filter

Choose: ClassAssigner -C first — Apply | Stop

Current relation

Relation: winequalityN-weka.filters.unsupervised.attribute.Remove-R13-weka.filters.un... Attributes: 12
Instances: 4562 Sum of weights: 4562

Attributes

All | None | Invert | Pattern

No.	Name
1	type
2	fixed acidity
3	volatile acidity
4	citric acid
5	residual sugar
6	chlorides
7	free sulfur dioxide
8	total sulfur dioxide
9	density
10	pH
11	sulphates
12	alcohol

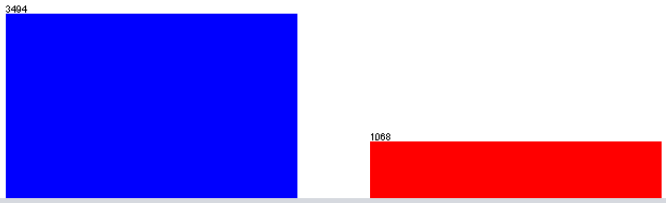
Remove

Selected attribute

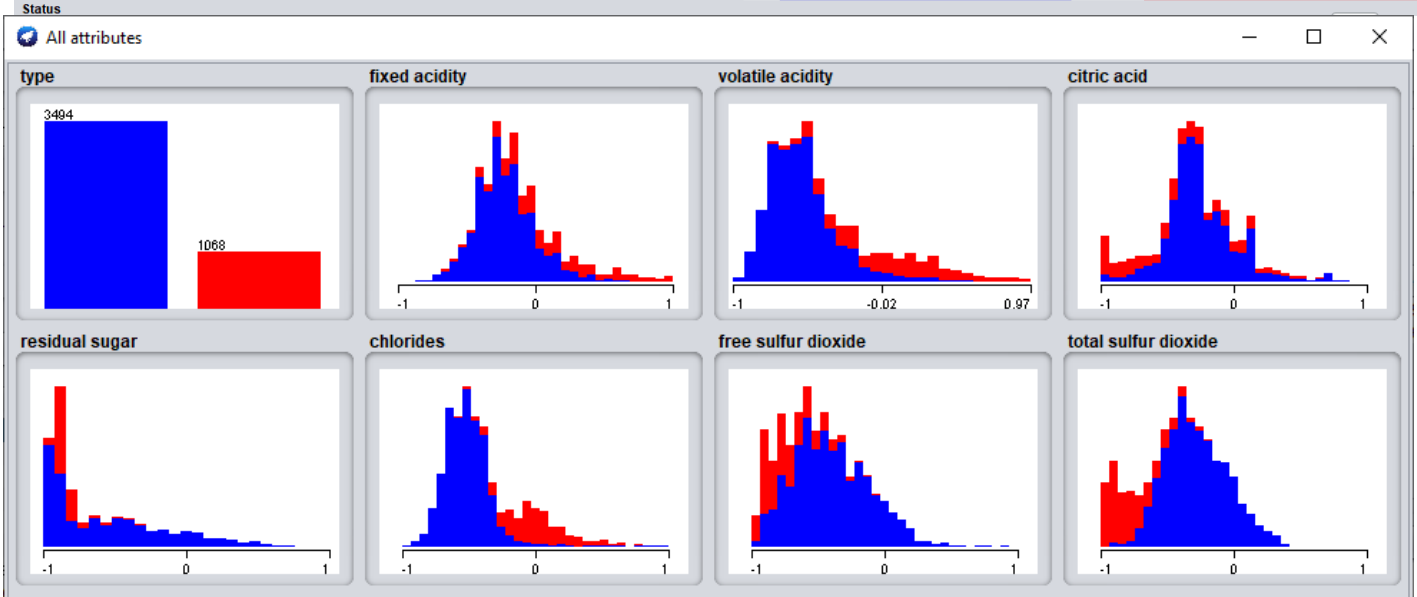
Name: type Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count	Weight
1	white	3494	3494.0
2	red	1068	1068.0

Class: type (Nom) — Visualize All



Status



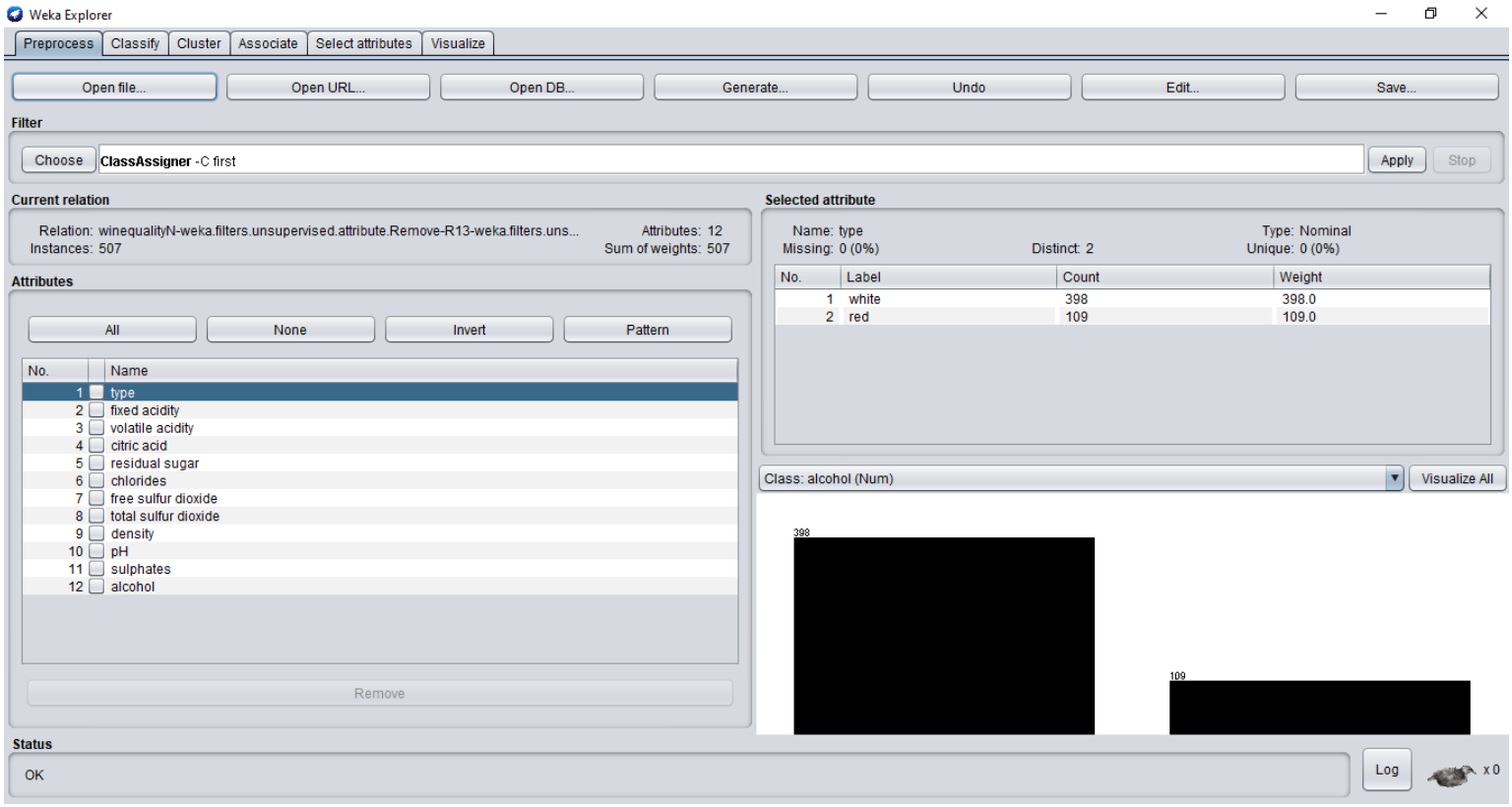
Viewer

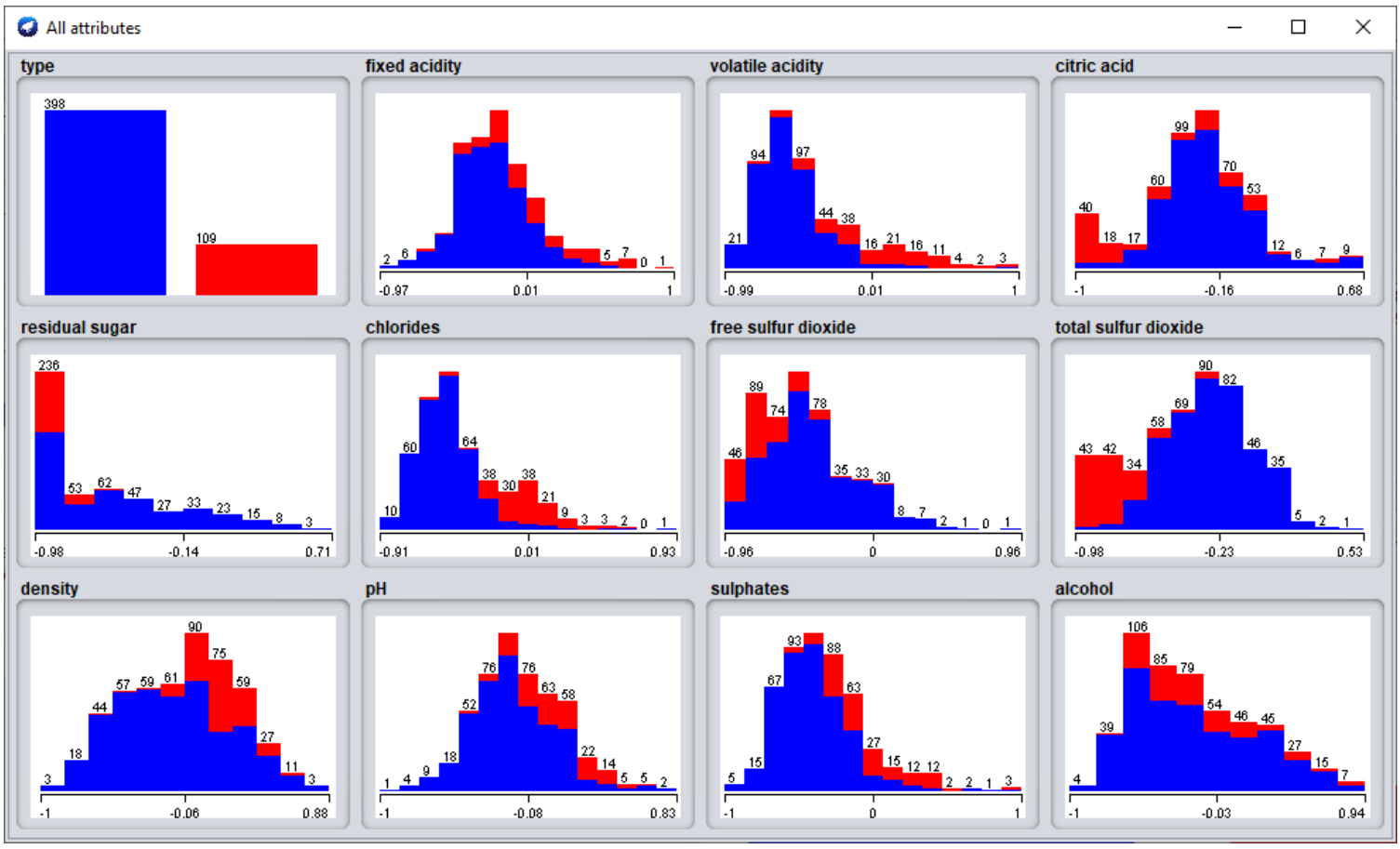
Relation: winequalityN-weka.filters.unsupervised.attribute.Remove-R13-weka.filters.unsupervised.attribute.ClassAssigner-Cfirst-weka.filters.unsupervised.instance.RemoveDupli...

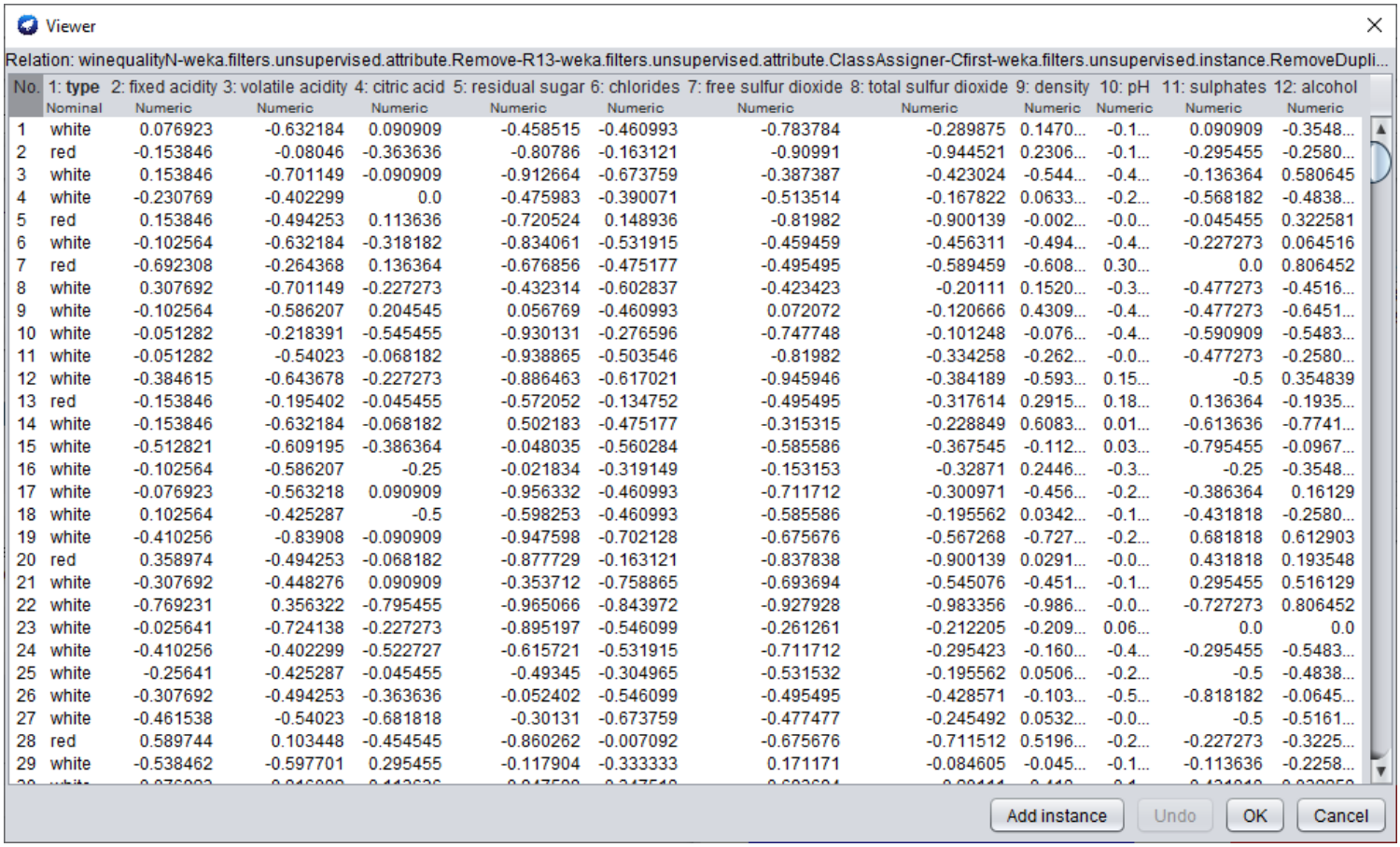
No.	1: type	2: fixed acidity	3: volatile acidity	4: citric acid	5: residual sugar	6: chlorides	7: free sulfur dioxide	8: total sulfur dioxide	9: density	10: pH	11: sulphates	12: alcohol
	Nominal	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric
1	red	0.897436	0.333333	-0.454545	-0.834061	0.078014	-0.72973	-0.739251	0.5449...	-0.4...	0.136364	-0.0322...
2	white	-0.435897	-0.563218	-0.363636	-0.633188	-0.234043	-0.459459	0.081831	0.17744	0.64...	0.113636	-0.3548...
3	white	-0.076923	-0.724138	-0.181818	-0.947598	-0.588652	-0.225225	-0.417476	-0.304...	0.08...	-0.727273	-0.3870...
4	white	-0.230769	-0.747126	-0.25	-0.938865	-0.687943	-0.621622	-0.550624	-0.650...	-0.3...	-0.090909	0.387097
5	white	-0.307692	-0.172414	0.113636	-0.379913	-0.489362	-0.72973	-0.0957	0.08872	-0.3...	-0.659091	-0.7741...
6	white	-0.410256	-0.356322	0.318182	0.257642	-0.503546	-0.261261	-0.395284	0.35488	-0.2...	-0.340909	-0.6774...
7	white	-0.025641	-0.563218	-0.454545	-0.720524	-0.304965	-0.675676	-0.395284	0.1013...	-0.2...	-0.477273	-0.7096...
8	white	-0.435897	-0.793103	0.113636	-0.965066	-0.64539	-0.54955	-0.445215	-0.380...	-0.1...	-0.545455	-0.4193...
9	white	-0.205128	-0.402299	0.681818	-0.074236	-0.148936	-0.225225	-0.20111	0.2281...	-0.5...	0.340909	-0.6129...
10	white	0.384615	-0.448276	-0.045455	-0.938865	-0.475177	-0.765766	-0.001387	-0.012...	-0.3...	-0.409091	-0.4838...
11	white	0.384615	-0.747126	-0.045455	-0.877729	-0.460993	-0.72973	-0.456311	0.0215...	-0.3...	0.0	-0.3548...
12	red	-0.153846	-0.195402	-0.045455	-0.572052	-0.120567	-0.513514	-0.323162	0.2915...	0.18...	0.113636	-0.1935...
13	white	-0.102564	-0.37931	0.522727	-0.327511	-0.375887	-0.837838	-0.478502	0.1140...	-0.2...	-0.363636	-0.0645...
14	white	-0.410256	-0.678161	-0.363636	0.393013	-0.29078	-0.045045	-0.284327	0.2091...	-0.1...	-0.636364	-0.1935...
15	white	-0.205128	0.011494	0.227273	-0.362445	-0.617021	-0.603604	-0.0957	-0.295...	-0.0...	-0.431818	0.354839
16	white	0.076923	-0.425287	-0.204545	-0.179039	-0.631206	-0.621622	-0.434119	-0.054...	-0.3...	-0.681818	0.16129
17	white	-0.410256	-0.195402	-0.204545	-0.257642	-0.29078	0.477477	0.348128	0.26616	0.10...	-0.363636	-0.83871
18	white	-0.358974	-0.241379	-0.25	-0.641921	-0.801418	-0.513514	-0.423024	-0.506...	-0.0...	-0.636364	0.451613
19	red	-0.25641	-0.218391	-0.386364	-0.30131	-0.163121	-0.585586	-0.212205	-0.025...	-0.2...	-0.204545	0.064516
20	white	-0.564103	-0.45977	-0.136364	-0.825328	-0.659574	-0.837838	-0.650485	-0.519...	-0.3...	-0.159091	-0.0967...
21	white	-0.307692	-0.632184	-0.113636	0.458515	-0.390071	-0.621622	-0.334258	0.4917...	-0.3...	-0.431818	-0.6129...
22	white	-0.205128	-0.494253	-0.181818	-0.659389	-0.361702	-0.459459	0.092926	0.0164...	0.15...	-0.204545	-0.2258...
23	red	-0.179487	0.321839	-0.636364	-0.868996	-0.078014	-0.873874	-0.894591	0.1343...	0.10...	-0.25	-0.4516...
24	white	-0.384615	-0.724138	-0.409091	-0.90393	0.191489	-0.297297	-0.140083	-0.329...	0.22...	0.0	-0.0322...
25	white	-0.230769	-0.126437	-0.409091	-0.502183	0.957447	-0.135135	-0.151179	-0.083...	-0.2...	-0.431818	-0.3548...
26	white	-0.205128	-0.632184	-0.477273	-0.432314	-0.546099	-0.657658	-0.495146	-0.321...	-0.3...	0.431818	0.096774
27	white	-0.461538	-0.448276	-0.25	-0.868996	-0.744681	-0.387387	-0.267684	-0.703...	0.10...	-0.545455	0.516129
28	white	-0.384615	-0.701149	-0.340909	-0.912664	-0.574468	-0.585586	-0.522885	-0.489...	-0.0...	-0.363636	0.032258
29	red	-0.205128	0.149425	-0.545455	-0.899563	-0.304965	-0.873874	-0.911234	-0.225...	0.11...	-0.386364	0.193548

Add instance | Undo | OK | Cancel

Test Set:
Fig 8-9-10







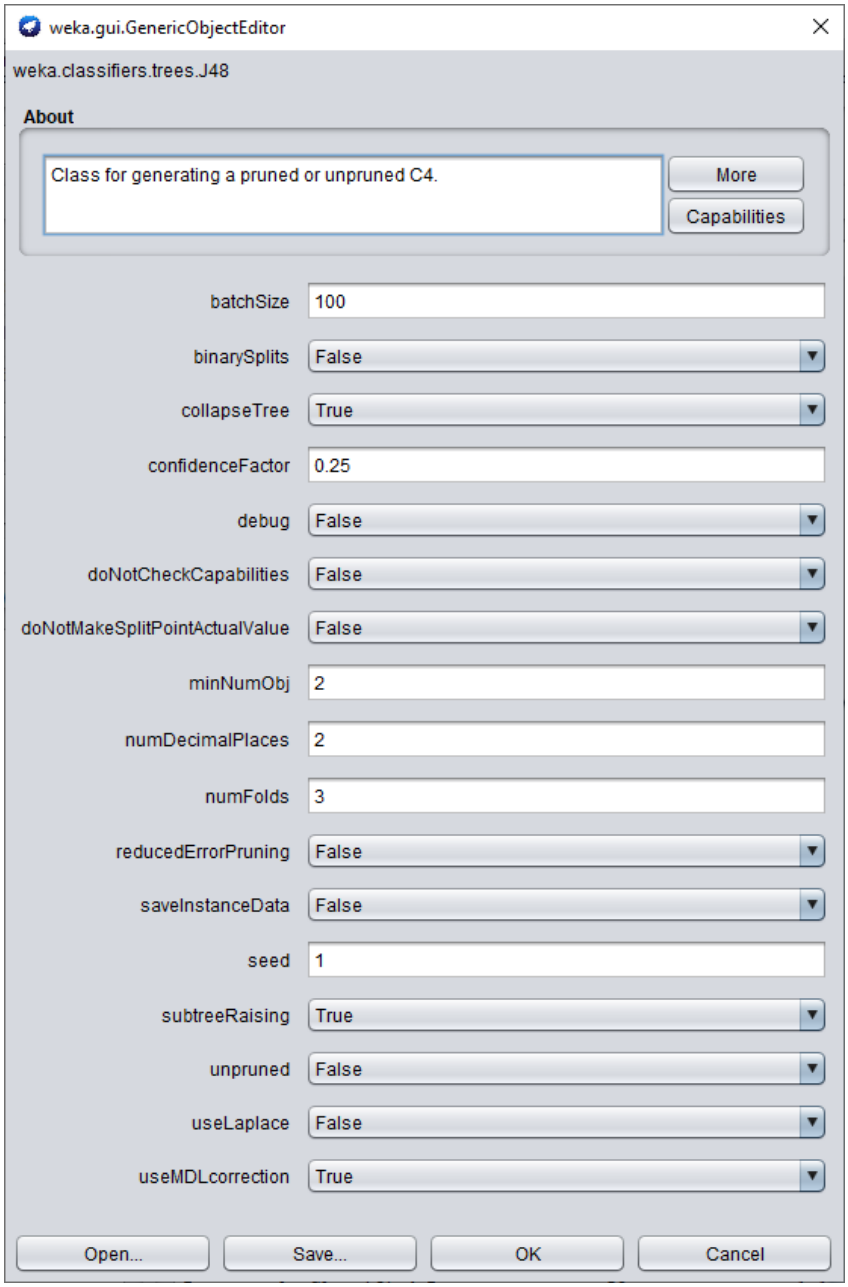
No.	1: type	2: fixed acidity	3: volatile acidity	4: citric acid	5: residual sugar	6: chlorides	7: free sulfur dioxide	8: total sulfur dioxide	9: density	10: pH	11: sulphates	12: alcohol
	Nominal	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric
1	white	0.076923	-0.632184	0.090909	-0.458515	-0.460993	-0.783784	-0.289875	0.1470...	-0.1...	0.090909	-0.3548...
2	red	-0.153846	-0.08046	-0.363636	-0.80786	-0.163121	-0.90991	-0.944521	0.2306...	-0.1...	-0.295455	-0.2580...
3	white	0.153846	-0.701149	-0.090909	-0.912664	-0.673759	-0.387387	-0.423024	-0.544...	-0.4...	-0.136364	0.580645
4	white	-0.230769	-0.402299	0.0	-0.475983	-0.390071	-0.513514	-0.167822	0.0633...	-0.2...	-0.568182	-0.4838...
5	red	0.153846	-0.494253	0.113636	-0.720524	0.148936	-0.81982	-0.900139	-0.002...	-0.0...	-0.045455	0.322581
6	white	-0.102564	-0.632184	-0.318182	-0.834061	-0.531915	-0.459459	-0.456311	-0.494...	-0.4...	-0.227273	0.064516
7	red	-0.692308	-0.264368	0.136364	-0.676856	-0.475177	-0.495495	-0.589459	-0.608...	0.30...	0.0	0.806452
8	white	0.307692	-0.701149	-0.227273	-0.432314	-0.602837	-0.423423	-0.20111	0.1520...	-0.3...	-0.477273	-0.4516...
9	white	-0.102564	-0.586207	0.204545	0.056769	-0.460993	0.072072	-0.120666	0.4309...	-0.4...	-0.477273	-0.6451...
10	white	-0.051282	-0.218391	-0.545455	-0.930131	-0.276596	-0.747748	-0.101248	-0.076...	-0.4...	-0.590909	-0.5483...
11	white	-0.051282	-0.54023	-0.068182	-0.938865	-0.503546	-0.81982	-0.334258	-0.262...	-0.0...	-0.477273	-0.2580...
12	white	-0.384615	-0.643678	-0.227273	-0.886463	-0.617021	-0.945946	-0.384189	-0.593...	0.15...	-0.5	0.354839
13	red	-0.153846	-0.195402	-0.045455	-0.572052	-0.134752	-0.495495	-0.317614	0.2915...	0.18...	0.136364	-0.1935...
14	white	-0.153846	-0.632184	-0.068182	0.502183	-0.475177	-0.315315	-0.228849	0.6083...	0.01...	-0.613636	-0.7741...
15	white	-0.512821	-0.609195	-0.386364	-0.048035	-0.560284	-0.585586	-0.367545	-0.112...	0.03...	-0.795455	-0.0967...
16	white	-0.102564	-0.586207	-0.25	-0.021834	-0.319149	-0.153153	-0.32871	0.2446...	-0.3...	-0.25	-0.3548...
17	white	-0.076923	-0.563218	0.090909	-0.956332	-0.460993	-0.711712	-0.300971	-0.456...	-0.2...	-0.386364	0.16129
18	white	0.102564	-0.425287	-0.5	-0.598253	-0.460993	-0.585586	-0.195562	0.0342...	-0.1...	-0.431818	-0.2580...
19	white	-0.410256	-0.83908	-0.090909	-0.947598	-0.702128	-0.675676	-0.567268	-0.727...	-0.2...	0.681818	0.612903
20	red	0.358974	-0.494253	-0.068182	-0.877729	-0.163121	-0.837838	-0.900139	0.0291...	-0.0...	0.431818	0.193548
21	white	-0.307692	-0.448276	0.090909	-0.353712	-0.758865	-0.693694	-0.545076	-0.451...	-0.1...	0.295455	0.516129
22	white	-0.769231	0.356322	-0.795455	-0.965066	-0.843972	-0.927928	-0.983356	-0.986...	-0.0...	-0.727273	0.806452
23	white	-0.025641	-0.724138	-0.227273	-0.895197	-0.546099	-0.261261	-0.212205	-0.209...	0.06...	0.0	0.0
24	white	-0.410256	-0.402299	-0.522727	-0.615721	-0.531915	-0.711712	-0.295423	-0.160...	-0.4...	-0.295455	-0.5483...
25	white	-0.25641	-0.425287	-0.045455	-0.49345	-0.304965	-0.531532	-0.195562	0.0506...	-0.2...	-0.5	-0.4838...
26	white	-0.307692	-0.494253	-0.363636	-0.052402	-0.546099	-0.495495	-0.428571	-0.103...	-0.5...	-0.818182	-0.0645...
27	white	-0.461538	-0.54023	-0.681818	-0.30131	-0.673759	-0.477477	-0.245492	0.0532...	-0.0...	-0.5	-0.5161...
28	red	0.589744	0.103448	-0.454545	-0.860262	-0.007092	-0.675676	-0.711512	0.5196...	-0.2...	-0.227273	-0.3225...
29	white	-0.538462	-0.597701	0.295455	-0.117904	-0.333333	0.171171	-0.084605	-0.045...	-0.1...	-0.113636	-0.2258...

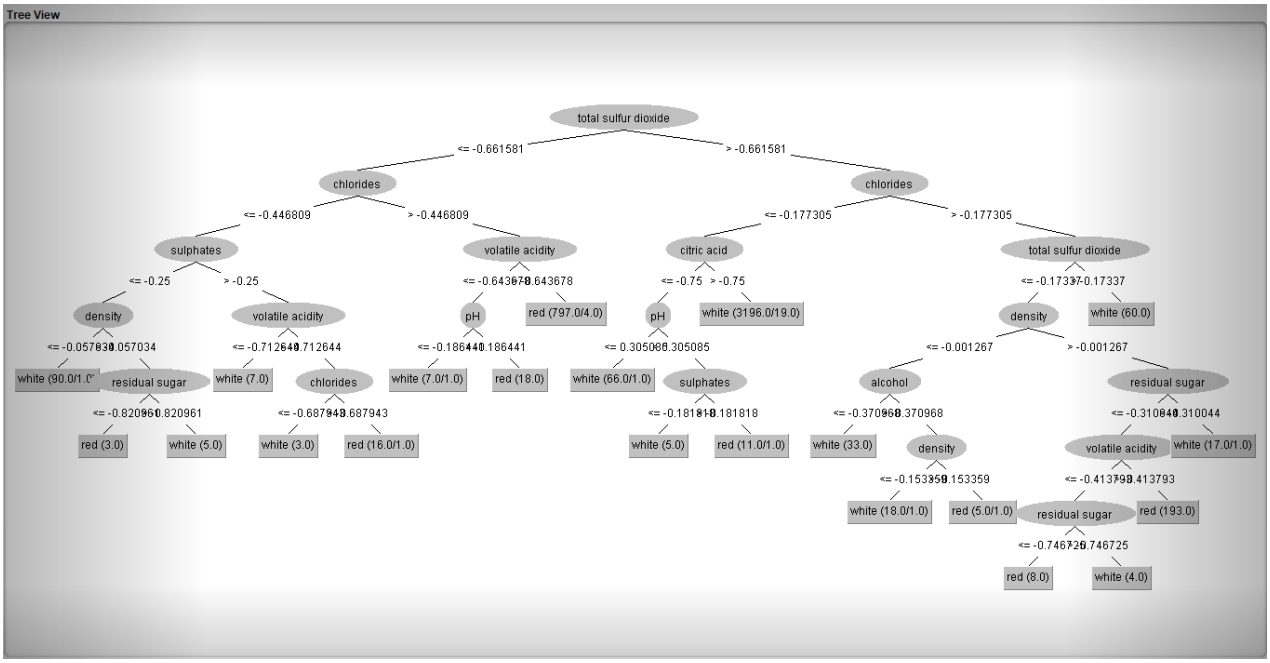
3.1) Classification: J48 Tree

First using default values of the model in Weka. I used the Test set for evaluation and model accuracy was more than 99%:

Correctly Classified Instances	502	99.0138 %
Incorrectly Classified Instances	5	0.9862 %

Model Parameters are shown in Fig 11





Tree on **Fig 12:**

Detailed Run output with confusion matrix and tree architecture:
=== Run information ===

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: winequalityN-weka.filters.unsupervised.attribute.Remove-R13-weka.filters.unsupervised.attribute.ClassAssigner-Cfirst-weka.filters.unsupervised.instance.RemoveDuplicates-weka.filters.unsupervised.attribute.ReplaceMissingValues-weka.filters.unsupervised.attribute.InterquartileRange-Rfirst-last-O3.0-E6.0-weka.filters.unsupervised.instance.RemoveWithValues-S0.0-C13-Llast-weka.filters.unsupervised.instance.RemoveWithValues-S0.0-C14-Llast-weka.filters.unsupervised.attribute.Remove-R13-14-weka.filters.unsupervised.attribute.Normalize-S2.0-T-1.0-weka.filters.unsupervised.instance.Randomize-S42-weka.filters.unsupervised.instance.RemovePercentage-P10.0-weka.filters.unsupervised.attribute.ClassAssigner-Cfirst
Instances: 4562
Attributes: 12
type
fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol
Test mode: user supplied test set: size unknown (reading incrementally)

=== Classifier model (full training set) ===

J48 pruned tree
----

total sulfur dioxide <= -0.661581
| chlorides <= -0.446809
| | sulphates <= -0.25
| | | density <= -0.057034: white (90.0/1.0)
| | | density > -0.057034
| | | | residual sugar <= -0.820961: red (3.0)
| | | | residual sugar > -0.820961: white (5.0)
| | sulphates > -0.25
| | | volatile acidity <= -0.712644: white (7.0)

```
|   |   |   volatile acidity > -0.712644
|   |   |   |   chlorides <= -0.687943: white (3.0)
|   |   |   |   chlorides > -0.687943: red (16.0/1.0)
|   chlorides > -0.446809
|   |   volatile acidity <= -0.643678
|   |   |   pH <= -0.186441: white (7.0/1.0)
|   |   |   pH > -0.186441: red (18.0)
|   |   volatile acidity > -0.643678: red (797.0/4.0)
total sulfur dioxide > -0.661581
|   chlorides <= -0.177305
|   |   citric acid <= -0.75
|   |   |   pH <= 0.305085: white (66.0/1.0)
|   |   |   pH > 0.305085
|   |   |   |   sulphates <= -0.181818: white (5.0)
|   |   |   |   sulphates > -0.181818: red (11.0/1.0)
|   |   citric acid > -0.75: white (3196.0/19.0)
|   chlorides > -0.177305
|   |   total sulfur dioxide <= -0.17337
|   |   |   density <= -0.001267
|   |   |   |   alcohol <= -0.370968: white (33.0)
|   |   |   |   alcohol > -0.370968
|   |   |   |   |   density <= -0.153359: white (18.0/1.0)
|   |   |   |   |   density > -0.153359: red (5.0/1.0)
|   |   |   density > -0.001267
|   |   |   |   residual sugar <= -0.310044
|   |   |   |   |   volatile acidity <= -0.413793
|   |   |   |   |   |   residual sugar <= -0.746725: red (8.0)
|   |   |   |   |   |   residual sugar > -0.746725: white (4.0)
|   |   |   |   |   volatile acidity > -0.413793: red (193.0)
|   |   |   |   residual sugar > -0.310044: white (17.0/1.0)
|   |   total sulfur dioxide > -0.17337: white (60.0)

Number of Leaves  :         21

Size of the tree :   41


Time taken to build model: 0.05 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances         502               99.0138 %
Incorrectly Classified Instances         5                0.9862 %
Kappa statistic                          0.9707
```

Mean absolute error 0.0158
Root mean squared error 0.0938
Relative absolute error 4.5284 %
Root relative squared error 22.8039 %
Total Number of Instances 507

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.995	0.028	0.992	0.995	0.994	0.971	0.991	0.995	white
	0.972	0.005	0.981	0.972	0.977	0.971	0.991	0.985	red
Weighted Avg.	0.990	0.023	0.990	0.990	0.990	0.971	0.991	0.993	

=== Confusion Matrix ===

a	b	<-- classified as
396	2	a = white
3	106	b = red

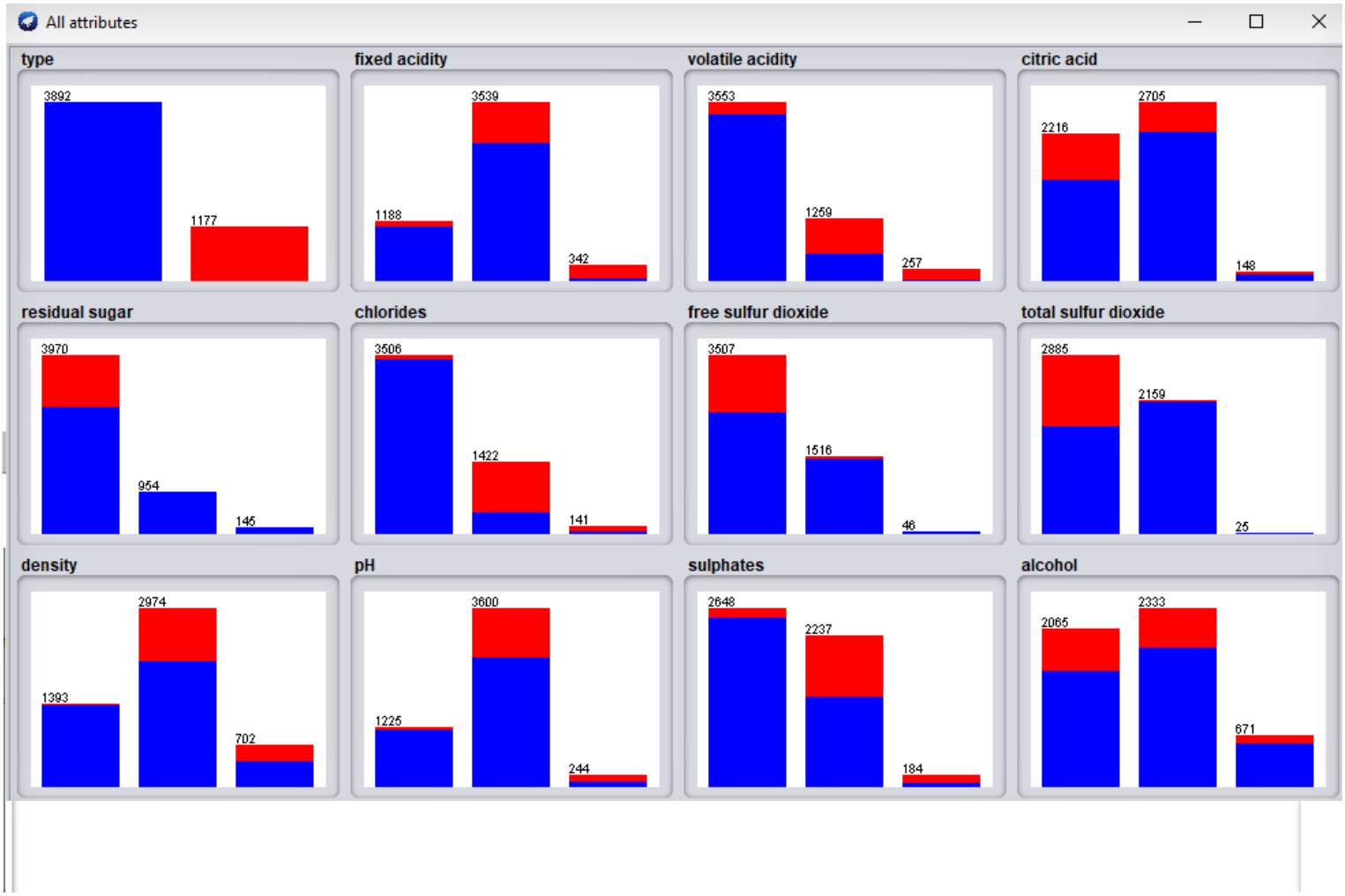
To find if accuracy could be improved by making leaf size bigger, I set it to 10 and on the test set incorrectly classified 5 instances reduced to 4 and accuracy increased to 99.2%.

The resulted tree is complex and to decrease complexity I increased min leaf size to 100, 500. Accuracy reduced to 97% and 92%. Resulted trees are shown on **Fig 13-14:**

As it turns out Total sulfur dioxide and chlorides are the most important factors. Together they produce 95% accuracy and individually 90% and 92% that are amazingly high.

3.2) Classification: Association Rules

For association, rules I have discretized (step 8) dataset numerical values into 3 bins. Overview of dataset attributes are shown on **Fig 15:**



First try model with default parameters:

- Minimum support: 0.45 (2281 instances)
- Minimum metric <confidence>: 0.9
- Number of cycles performed: 11

=== Run information ===

Scheme: weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Relation: winequalityN-weka.filters.unsupervised.attribute.Remove-R13-weka.filters.unsupervised.attribute.ClassAssigner-Cfirst-weka.filters.unsupervised.instance.RemoveDuplicates-weka.filters.unsupervised.attribute.ReplaceMissingValues-weka.filters.unsupervised.attribute.InterquartileRange-Rfirst-last-O3.0-E6.0-weka.filters.unsupervised.instance.RemoveWithValues-S0.0-C13-Llast-weka.filters.unsupervised.instance.RemoveWithValues-S0.0-C14-Llast-weka.filters.unsupervised.attribute.Remove-R13-14-weka.filters.unsupervised.attribute.Normalize-S2.0-T-1.0-weka.filters.unsupervised.instance.Randomize-S42-weka.filters.unsupervised.attribute.ClassAssigner-Cfirst-weka.filters.unsupervised.attribute.Discretize-B3-M-1.0-Rfirst-last-precision6-weka.filters.unsupervised.attribute.Reorder-R2-last,1
Instances: 5069

Attributes: 12
fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol
type
=== Associator model (full training set) ===

Apriori
=======

Minimum support: 0.45 (2281 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 11

Generated sets of large itemsets:

Size of set of large itemsets L(1): 12

Size of set of large itemsets L(2): 23

Size of set of large itemsets L(3): 8

Best rules found:

1. volatile acidity='(-inf--0.333333]' chlorides='(-inf--0.333333]' 2980 ==> type=white 2954 <conf:(0.99)> lift:(1.29) lev:(0.13) [665] conv:(25.63)
2. fixed acidity='(-0.333333-0.333333]' chlorides='(-inf--0.333333]' 2418 ==> type=white 2378 <conf:(0.98)> lift:(1.28) lev:(0.1) [521] conv:(13.69)
3. chlorides='(-inf--0.333333]' 3506 ==> type=white 3426 <conf:(0.98)> lift:(1.27) lev:(0.14) [734] conv:(10.05)
4. chlorides='(-inf--0.333333]' pH='(-0.333333-0.333333]' 2388 ==> type=white 2332 <conf:(0.98)> lift:(1.27) lev:(0.1) [498] conv:(9.73)
5. residual sugar='(-inf--0.333333]' chlorides='(-inf--0.333333]' 2598 ==> type=white 2522 <conf:(0.97)> lift:(1.26) lev:(0.1) [527] conv:(7.83)
6. fixed acidity='(-0.333333-0.333333]' volatile acidity='(-inf--0.333333]' 2474 ==> type=white 2347 <conf:(0.95)> lift:(1.24) lev:(0.09) [447] conv:(4.49)

7. sulphates='(-inf--0.333333]' 2648 ==> type=white 2502 <conf:(0.94)> lift:(1.23) lev:(0.09) [468] conv:(4.18)
8. volatile acidity='(-inf--0.333333]' 3553 ==> type=white 3317 <conf:(0.93)> lift:(1.22) lev:(0.12) [588] conv:(3.48)
9. free sulfur dioxide='(-inf--0.333333]' total sulfur dioxide='(-inf--0.333333]' 2606 ==> residual sugar='(-inf--0.333333]' 2398 <conf:(0.92)> lift:(1.17) lev:(0.07) [357] conv:(2.7)
10. total sulfur dioxide='(-inf--0.333333]' 2885 ==> residual sugar='(-inf--0.333333]' 2636 <conf:(0.91)> lift:(1.17) lev:(0.07) [376] conv:(2.5)

To get better results it is important to decrease minimum support. It will allow us to get rules that apply to the smaller group but with higher confidence. For that purpose, I set delta (Iteratively decrease support by this factor) to 0.1, rules were the following:

1. chlorides='(-inf--0.333333]' sulphates='(-inf--0.333333]' 2200 ==> type=white 2184 <conf:(0.99)> lift:(1.29) lev:(0.1) [494] conv:(30.05)
2. volatile acidity='(-inf--0.333333]' chlorides='(-inf--0.333333]' 2980 ==> type=white 2954 <conf:(0.99)> lift:(1.29) lev:(0.13) [665] conv:(25.63)
3. fixed acidity='(-0.333333-0.333333]' volatile acidity='(-inf--0.333333]' chlorides='(-inf--0.333333]' 2084 ==> type=white 2065 <conf:(0.99)> lift:(1.29) lev:(0.09) [464] conv:(24.19)
4. volatile acidity='(-inf--0.333333]' sulphates='(-inf--0.333333]' 2157 ==> type=white 2137 <conf:(0.99)> lift:(1.29) lev:(0.09) [480] conv:(23.85)
5. volatile acidity='(-inf--0.333333]' residual sugar='(-inf--0.333333]' chlorides='(-inf--0.333333]' 2206 ==> type=white 2183 <conf:(0.99)> lift:(1.29) lev:(0.1) [489] conv:(21.34)
6. total sulfur dioxide='(-0.333333-0.333333]' 2159 ==> type=white 2127 <conf:(0.99)> lift:(1.28) lev:(0.09) [469] conv:(15.19)
7. fixed acidity='(-0.333333-0.333333]' chlorides='(-inf--0.333333]' 2418 ==> type=white 2378 <conf:(0.98)> lift:(1.28) lev:(0.1) [521] conv:(13.69)
8. chlorides='(-inf--0.333333]' 3506 ==> type=white 3426 <conf:(0.98)> lift:(1.27) lev:(0.14) [734] conv:(10.05)
9. chlorides='(-inf--0.333333]' pH='(-0.333333-0.333333]' 2388 ==> type=white 2332 <conf:(0.98)> lift:(1.27) lev:(0.1) [498] conv:(9.73)
10. residual sugar='(-inf--0.333333]' chlorides='(-inf--0.333333]' 2598 ==> type=white 2522 <conf:(0.97)> lift:(1.26) lev:(0.1) [527] conv:(7.83)

Because some rules do not output wine type I changed parameter car to True. With that, all rules shall be about class. Also, changed a number of rules to 20

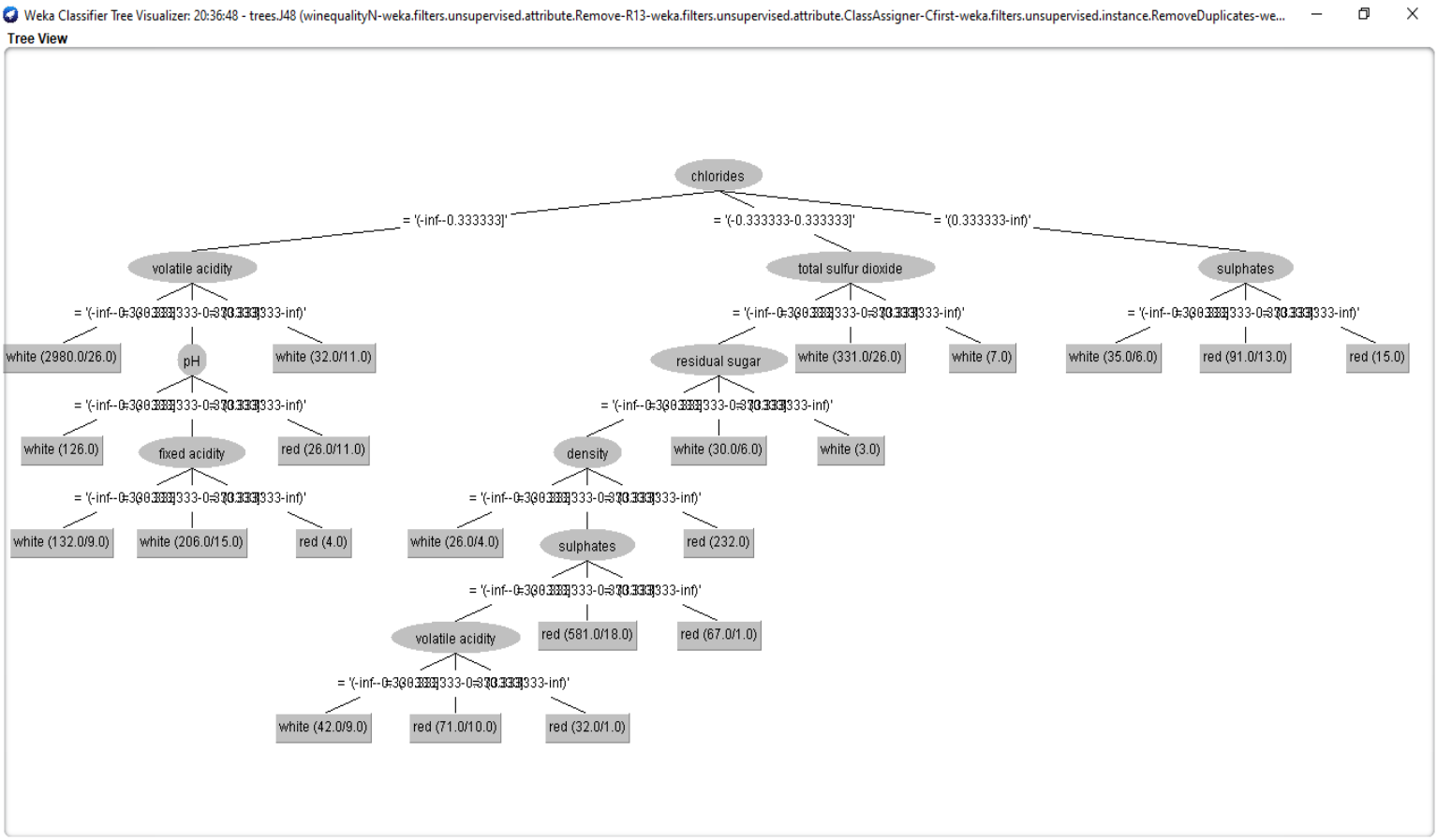
1. chlorides='(-inf--0.333333]' total sulfur dioxide='(-0.333333-0.333333]' 1799 ==> type=white 1798 conf:(1)

2. volatile acidity='(-inf--0.333333]' total sulfur dioxide='(-0.333333-0.333333]' 1824 ==> type=white 1821 conf:(1)
3. volatile acidity='(-inf--0.333333]' chlorides='(-inf--0.333333]' sulphates='(-inf--0.333333]' 1886 ==> type=white 1880 conf:(1)
4. chlorides='(-inf--0.333333]' sulphates='(-inf--0.333333]' 2200 ==> type=white 2184 conf:(0.99)
5. volatile acidity='(-inf--0.333333]' chlorides='(-inf--0.333333]' 2980 ==> type=white 2954 conf:(0.99)
6. fixed acidity='(-0.333333-0.333333]' volatile acidity='(-inf--0.333333]' chlorides='(-inf--0.333333]' 2084 ==> type=white 2065 conf:(0.99)
7. volatile acidity='(-inf--0.333333]' sulphates='(-inf--0.333333]' 2157 ==> type=white 2137 conf:(0.99)
8. volatile acidity='(-inf--0.333333]' citric acid='(-0.333333-0.333333]' chlorides='(-inf--0.333333]' 1839 ==> type=white 1821 conf:(0.99)
9. volatile acidity='(-inf--0.333333]' chlorides='(-inf--0.333333]' pH='(-0.333333-0.333333]' 2023 ==> type=white 2002 conf:(0.99)
10. volatile acidity='(-inf--0.333333]' residual sugar='(-inf--0.333333]' chlorides='(-inf--0.333333]' 2206 ==> type=white 2183 conf:(0.99)
11. citric acid='(-0.333333-0.333333]' chlorides='(-inf--0.333333]' 2043 ==> type=white 2014 conf:(0.99)
12. total sulfur dioxide='(-0.333333-0.333333]' 2159 ==> type=white 2127 conf:(0.99)
13. fixed acidity='(-0.333333-0.333333]' chlorides='(-inf--0.333333]' 2418 ==> type=white 2378 conf:(0.98)
14. chlorides='(-inf--0.333333]' 3506 ==> type=white 3426 conf:(0.98)
15. chlorides='(-inf--0.333333]' pH='(-0.333333-0.333333]' 2388 ==> type=white 2332 conf:(0.98)
16. residual sugar='(-inf--0.333333]' chlorides='(-inf--0.333333]' pH='(-0.333333-0.333333]' 1859 ==> type=white 1805 conf:(0.97)
17. residual sugar='(-inf--0.333333]' chlorides='(-inf--0.333333]' 2598 ==> type=white 2522 conf:(0.97)
18. chlorides='(-inf--0.333333]' free sulfur dioxide='(-inf--0.333333]' 2177 ==> type=white 2100 conf:(0.96)
19. fixed acidity='(-0.333333-0.333333]' volatile acidity='(-inf--0.333333]' 2474 ==> type=white 2347 conf:(0.95)
20. sulphates='(-inf--0.333333]' 2648 ==> type=white 2502 conf:(0.94)

To compare it classification output I run j48 to this dataset, with confidence 0.9 as in clustering and min leaf size 30 not to get too complicated tree:

Fig 16:

To summarize most of the rules for the type are also the same for j48 output in fig 16. Additionally, j48 outputs predictions of red wine that is



not the case in Apriori, as rules (all 20) are only for white wine. White wine has more range for parameters and in most cases, has values shared with red, also unique that are only characteristics of white as seen from fig 15. Those include low PH, acidity, chlorides, sulfates, density.

Part 2 - Clustering

1) Description of dataset and findings

I used the same Wine quality dataset for clustering.

The first objective is to find out if clustering into 2 clusters will give us Red and White wine clusters. Also, find out what happens when I increase the number of clusters.

I created k-means clustering, DBSCAN with k-means and EM. After tuning parameters, all models were able to give clusters that corresponded Red and White wine classes. The incorrect classification was less than 2% for each model. The best result gave DBSCAN with EM, then followed DBSCAN with k-means and lastly k-means. As it was expected DBSCAN outperformed simple k-means despite the fact that we have removed outliers in the dataset in the preprocessing step.

2)Preprocessing Steps

Dataset preprocessing is the same for clustering algorithms as in the previous (j48) task. I have tested other preprocessing but I have chosen best for all tasks, that is why I use same preprocessing, Additionally, I have tested the dataset without any preprocessing to see the difference in clustering. It gave same result of white and Red wine clusters even for the random initialization of class centroids.

2.1) Clustering: K-Means

I tried several configurations for k-means clustering. Changing the distance function between euclidean and Manhattan did not change clusters very much. The initialization method had a very big effect on cluster formation. When choosing random, resulting 2 clusters did not correspond red and white clusters, but choosing farthest first, k-means++ or canopy did give the same clusters corresponding class with very high accuracy.

Farthest first k-means ++, canopy results for different distance metrics.

Euclidean Distance

```
Class attribute: type
Classes to Clusters:

    0    1  <-- assigned to cluster
 2440 1452 | white
 1165   12 | red

Cluster 0 <-- red
Cluster 1 <-- white

Incorrectly clustered instances :       2452.0   48.3725 %
```

```
Class attribute: type
Classes to Clusters:

    0    1  <-- assigned to cluster
 3823   69 | white
   32 1145 | red

Cluster 0 <-- white
Cluster 1 <-- red

Incorrectly clustered instances :       101.0     1.9925 %
```

Manhattan Distance

Random initialization:

```
Class attribute: type
Classes to Clusters:

    0    1  <-- assigned to cluster
 3819   73 | white
   27 1150 | red

Cluster 0 <-- white
Cluster 1 <-- red

Incorrectly clustered instances :       100.0     1.9728 %
```

Screenshots of clusters:

As it is seen from clustering screenshots, k-means clustering gave corresponding clusters to the class.

Changing cluster number to 5 and as a result, the red wine cluster was split into two clusters and white wine to 3 clusters. Clusters 0 and 1 are Red and clusters 2-3-4 are white wine clusters.

```
Class attribute: type
Classes to Clusters:

    0    1    2    3    4  <-- assigned to cluster
   31   55 1251 1135 1420 | white
  433  695   37    4    8 | red
```

Visualization of clusters:

2.2) Clustering: DBSCAN

For the density-based clustering, I chose the k-means algorithm configuration with the best results from the previous task. The model gave slightly worse results than k-means:

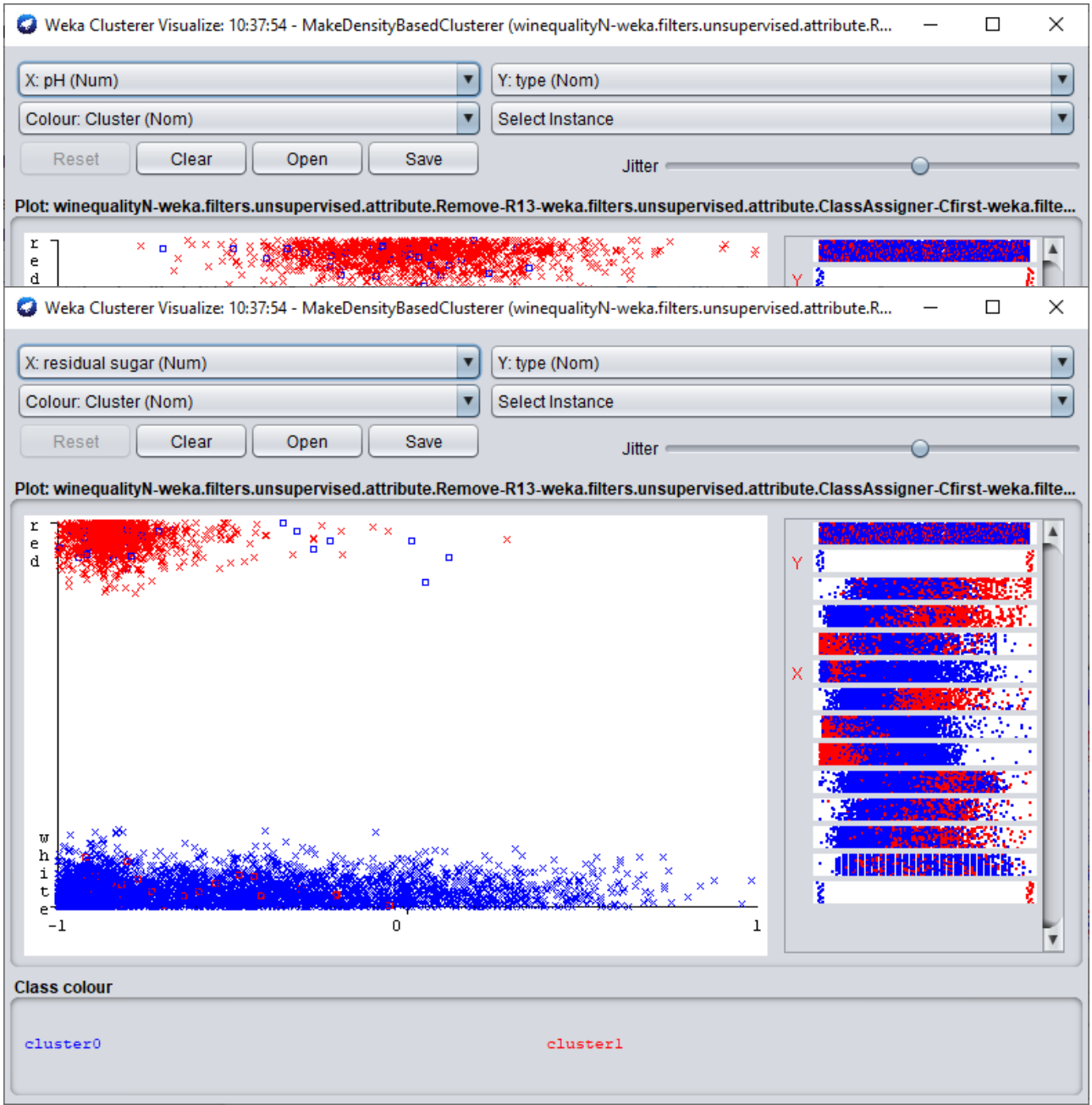
```
Class attribute: type
Classes to Clusters:

    0    1  <-- assigned to cluster
 3801   91 | white
   26 1151 | red

Cluster 0 <-- white
Cluster 1 <-- red

Incorrectly clustered instances :       117.0     2.3081 %
```

I changed parameters several times and find that when minStdDev is 0.25 (minimum allowable standard deviation for DBSCAN) it can outperform k-means:

```
Class attribute: type
Classes to Clusters:

    0    1  <-- assigned to cluster
 3847   45 | white
   39 1138 | red

Cluster 0 <-- white
Cluster 1 <-- red

Incorrectly clustered instances :       84.0      1.6571 %
```

But clusters are very similar to each other in the case of DBSCAN and k-means because both gave red and white wine clusters with very high accuracy.

Clusters screenshots:



Choosing EM in the density-based algorithm farther improved the accuracy of the model.

```
Class attribute: type
Classes to Clusters:

    0    1  <-- assigned to cluster
   54 3838 | white
 1148   29 | red

Cluster 0 <-- red
Cluster 1 <-- white

Incorrectly clustered instances :       83.0      1.6374 %
```

Part 3 - Overall Evaluation

1) Report Quality and presentation of knowledge

In my work, I used two methods, classification, and clustering, for predicting wine type from the physicochemical analysis. Dataset included 13 attributes: 1 Nominal and 12 numeric. Before testing models, some preprocessing steps were done to fill missing values, remove outliers, standardize and prepare the dataset for ML models. The j48 model achieved the best results (99.2%) but even clustering algorithms had less than 2% error. Several metrics were computed. Also, I plot decision trees and evaluate the importance of the input variables. Apriori association results were very interesting and unexpected, as it turned out that 95% accuracy can be achieved by using only two parameters. Clustering algorithms were very successful clustering Red and White wine separately. Even increasing clustering numbers just divided clusters to subclusters and did not mix two types.

To conclude, determining wine type from the physicochemical analysis is a very easy task for ML algorithms as for supervised as well for unsupervised if we use preprocessing and proper cleaning of the dataset.

2) References

I used Kaggle's dataset of wine quality, which is taken from published paper and preprocessed in a way to be useful for ML and DL.

- https://www.kaggle.com/rajyellow46/wine-quality
- https://www.scitepress.org/Papers/2015/55519/55519.pdf