Riga Technical University

Faculty of Computer Science, Information Technology and Energy

Report on the second practical assignment

Study course "Fundamentals of artificial intelligence"

Team number: 4

Students: /first name, last name, student ID/

- Grigoryan Anastasiya 213AEB029
- Danylo Karpov 220AIB025
- Kevin Randrianimanana 240AEB027
- Matthieu Porte 230ADB086
- Guy Leonard KAMAGA FOTSO WAFO 240ADM005

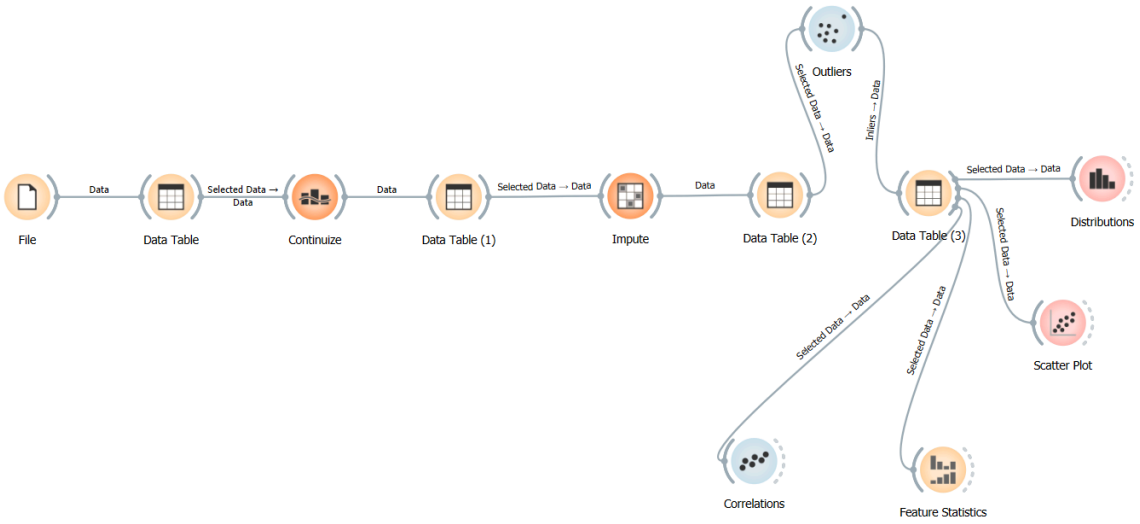
Teaching staff: Alla Anohina-Naumeca

Project link: https://github.com/matthieuporte/team4ai

Link to dataset: https://archive.ics.uci.edu/dataset/186/wine+quality

2023/2024 academic year

Orange tool workflow



< a screenshot of the workflow created in the Orange tool>

Part I

<this subsection should provide a general description of the dataset, accompanied by screenshots and references to the information sources used>

Description of the dataset

Dataset title: White wine quality

Dataset source: UC Irvine Machine Learning Repository
https://archive.ics.uci.edu/dataset/186/wine+quality

Creator and/or owner of the dataset: Paulo Cortez, A. Cerdeira, F. Almeida, T. Matos, J. Reis.

Description of the dataset problem domain: the data set under study describes the physicochemical properties of the Portuguese white wine "Vinho Verde". The purpose of the dataset is to model wine quality based on physico-chemical tests.

Dataset licensing conditions: this dataset is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) license. This allows for the sharing and adaptation of the dataset for any purpose, provided that the appropriate credit is given.

Information about the method or procedure for collecting the dataset: the description of the data does not indicate the exact method of data collection, but it is stated that the data relates to samples of Vinho Verde white wine from the north of Portugal. Physicochemical data were obtained through tests, and organoleptic data were collected through sensory evaluations.

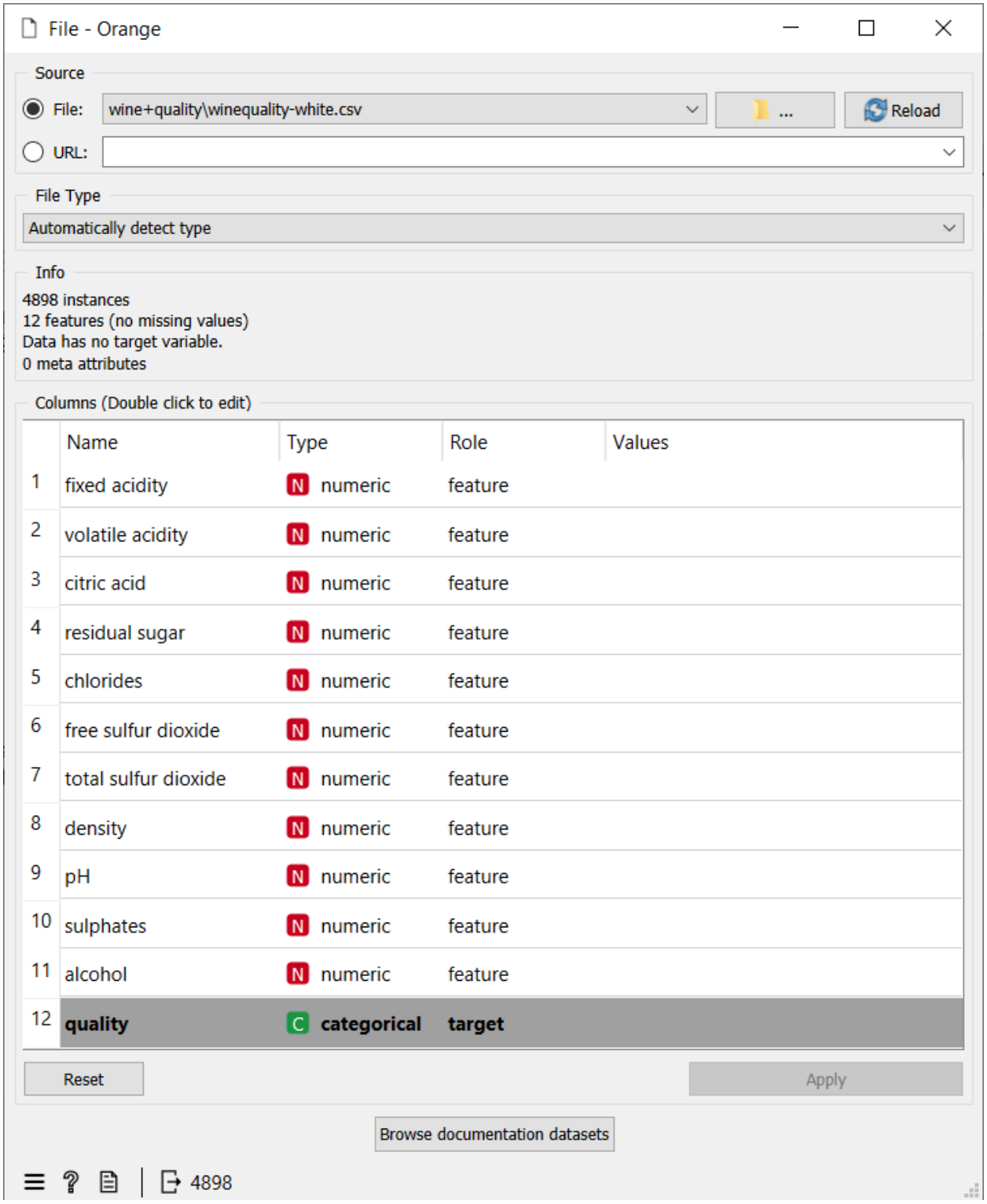
Description of the dataset content

Number of data objects in the dataset: 4898 instances.

Representation of features (attributes) of the dataset together with their roles in the Orange tool: in this dataset 12 features - fixed acidity, volatile

acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol are features, and quality is target.

<a screenshot of the Orange tool representing the features and their roles>


File - Orange
Source
File:
wine+quality\winequality-white.csv
...
Reload
URL:
File Type
Automatically detect type
Info
4898 instances
12 features (no missing values)
Data has no target variable.
0 meta attributes
Columns (Double click to edit)
Name
Type
Role
Values
1 fixed acidity N numeric feature
2 volatile acidity N numeric feature
3 citric acid N numeric feature
4 residual sugar N numeric feature
5 chlorides N numeric feature
6 free sulfur dioxide N numeric feature
7 total sulfur dioxide N numeric feature
8 density N numeric feature
9 pH N numeric feature
10 sulphates N numeric feature
11 alcohol N numeric feature
12 quality C categorical target
Reset
Apply
Browse documentation datasets
4898

Number of classes in the dataset: in the dataset represented 7 classes.

Description of classes:

<labels used to represent classes and the meaning of each class; if the dataset provides several possible data classifications, then the report should clearly identify which classification is being addressed in the assignment>

The “quality” has seven classes: 3, 4, 5, 6, 7, 8, 9. This indicates the quality of white wine on the scale from 0 to 10, where zero is the lowest value and 10 being the highest. In the current dataset the lowest value of the quality is 3 and higher - 10.

Number of data objects belonging to each class:

<add rows to table as needed>

Class label	Number of data objects
3	20
4	163
5	1457
6	2198

7	880
8	175
9	5

Description of features:

<add rows to table as needed>

Feature title	Explanation of the feature	Value type	Range of values
fixed_acidity	Describes the concentration of non-volatile substances in wine (malic acid, tartaric acid, citric acid). These acids do not evaporate quickly and play a decisive role in the taste of wine. Wine with a higher acidity level has a refreshing taste, while those with a lower	Continuous	From 3.80 to 14.20

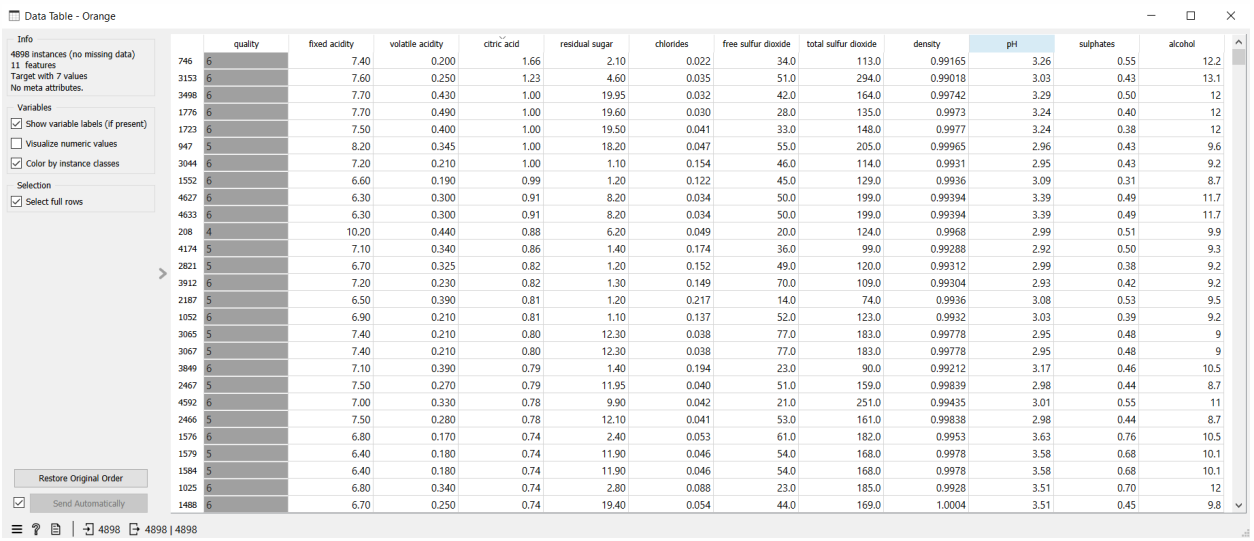
	acidity level have a flatter taste.		
volatile_acidity	Describes volatile acidity (for example the acetic acid content of wine). They evaporate easily and contribute to the aroma and taste. A higher rate can lead to unpleasant odors and tastes, which affects quality.	Continuous	From 0.080 to 1.100
citric_acid	Affect the overall acidity of the wine, gives a tart taste, and also serves as a preservative	Continuous	From 0.00 to 1.66
residual_sugar	Residual sugar balances the acidity of the wine and improves its taste. Higher levels of residual sugar result in sweeter wines, while a lower level favors dry wine and lowest levels result in drier wines.	Continuous	From 0.60 to 65.80

chlorides	Chlorides are a natural additive in wine that is essential for grape growth. The higher the concentration, the more salty the wine tastes, which is undesirable.	Continuous	From 0.009 to 0.346
free_sulfur_dioxide	Free sulfur dioxide is responsible for preserving wine and maintaining its stability and shelf life. High SO2 content can cause an unpleasant odor and also cause an allergic reaction.	Continuous	From 2.0 to 289.0
total_sulfur_dioxide	Indicate the total amount (free and bound forms) of sulfur dioxide, which is a preservative. High values may result in a sulfurous odor and astringency.	Continuous	From 9.0 to 440.0
density	The density of the wine affects the taste, as its value is influenced by the concentration of alcohol, sugar and acids. The higher the density value the more viscous and dense the taste and	Continuous	From 0.98711 to 1.03898

	lower, the opposite, watery and light.		
pH	Low pH values indicate high acidity and a more tart flavor, while higher values indicate low acidity (more alkaline) and flatter flavor.	Continuous	From 2.72 to 3.82
sulfates	Used as a preservative, but higher levels can lead to a bitter taste and unpleasant odor.	Continuous	From 0.22 to 1.08
alcohol	Wines with a higher alcohol content have a more intense taste, while wines with a lower alcohol content have a lighter taste. Also, wines with high alcohol content have better aging potential.	Continuous	From 8 to 14.2

Data file structure:

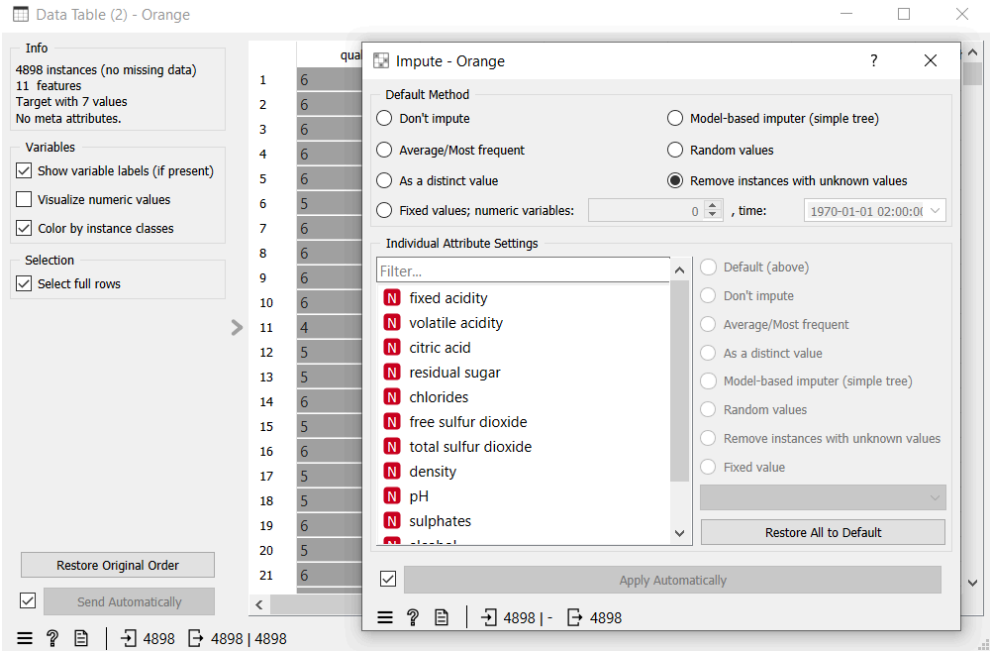
<a screenshot showing all the columns in the data file and their values for at least some data objects>



Information about missing values or outliers:

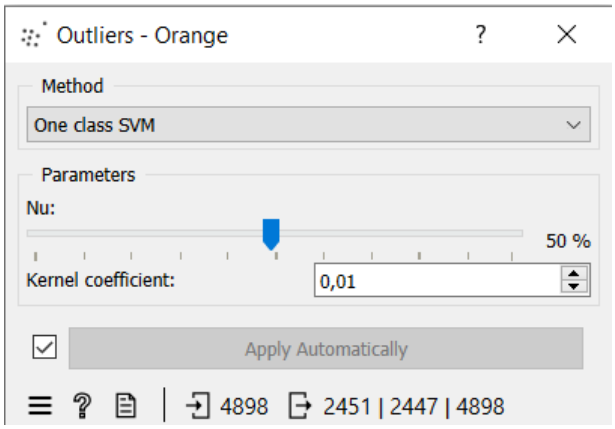
<a description of whether values of certain features (attributes) are missing in the dataset or whether there are outlier values and solutions that the students have used to solve the mentioned problems (demonstrated also by screenshots)>

There were no missing values identified in the dataset. This can be checked using the “Impute” tool, where we choose “Remove instance with unknown values”.

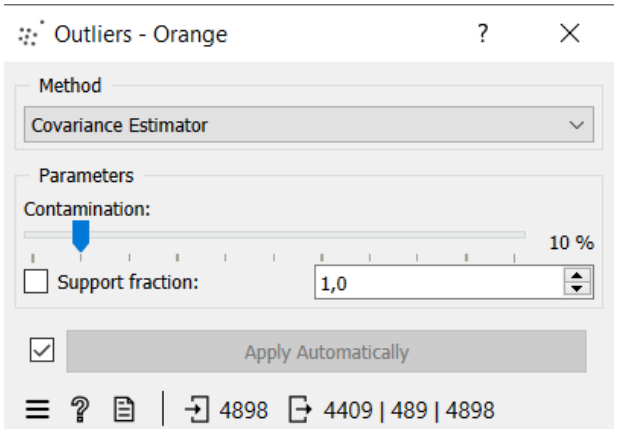


This screenshot represents “Impute” and “Data Table” tools where we can see that the quantity of instances does not change after operation: there were and remains 4898 instances. Also in Data Table Info we can see the title “no missing title” that means there is no lost data.

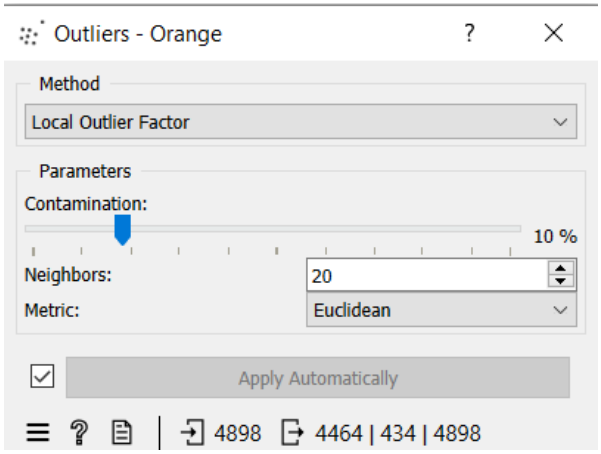
For outlier values, the “Outliers” tool was used. During the work, four methods for identifying outlier values were used to identify the best one.



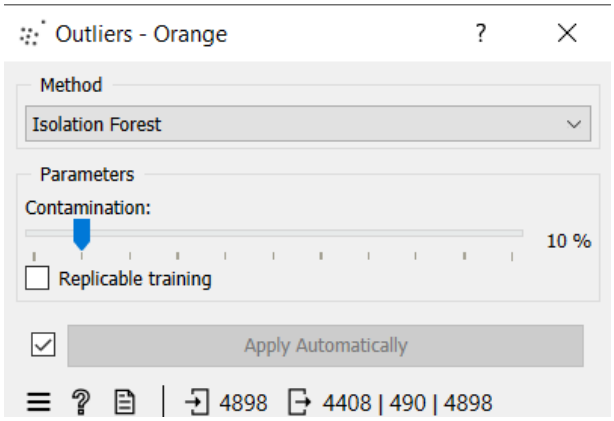
1. One class SVM shows 2447 outliers



2. Covariance Estimator shows 489 outliers



3. Local Outlier Factor shows 434 outlier

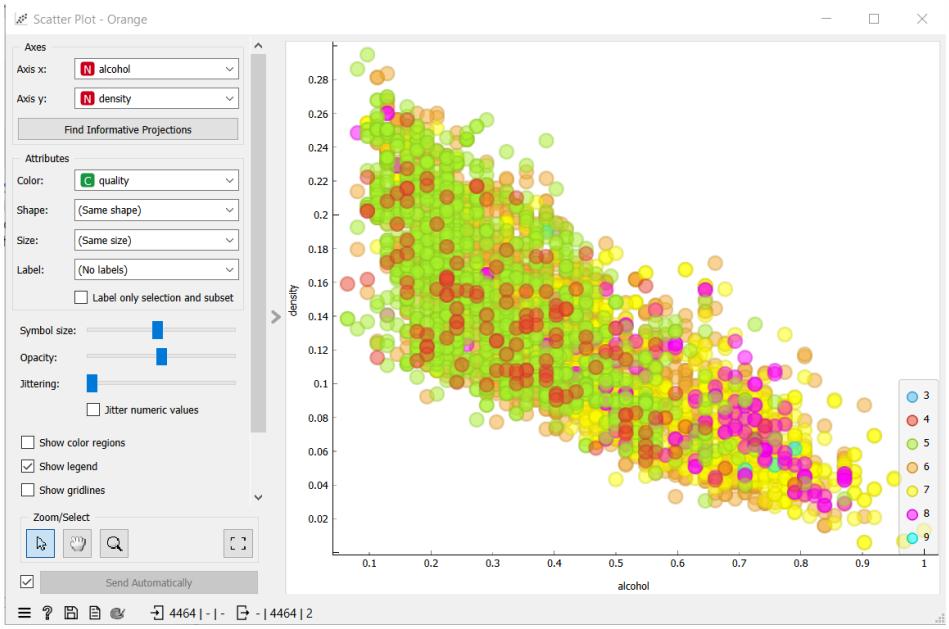


4. Isolation Forest shows 490 outlier

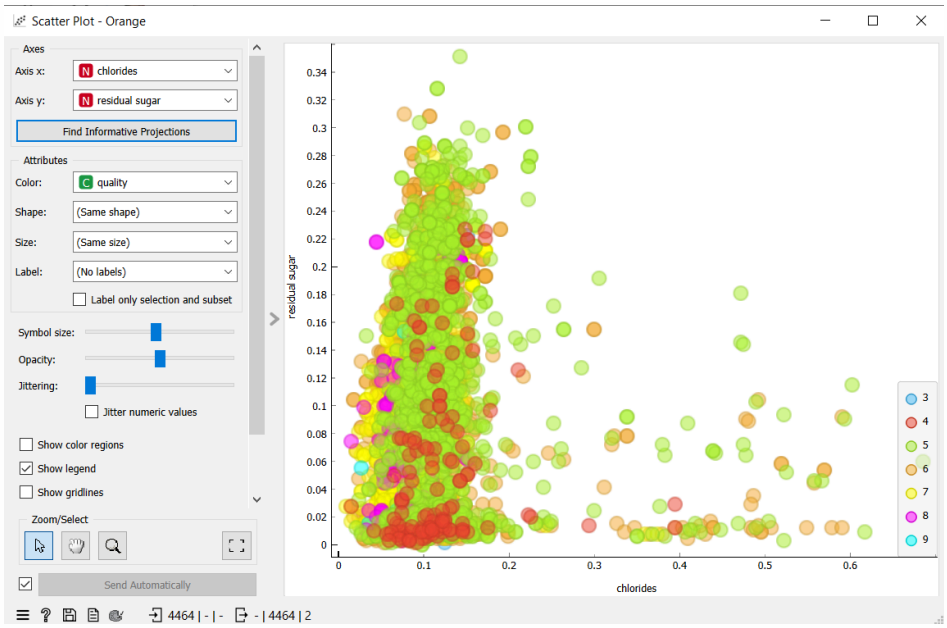
For further work, the Local Outlier Factor method was chosen since this dataset has a high data density, and this method evaluates the degree of density of each point and compares it with the surrounding density of its neighbors. If a given point has a lower density than its neighbors, then it can be considered an outlier.

Visual and statistical representation of the dataset

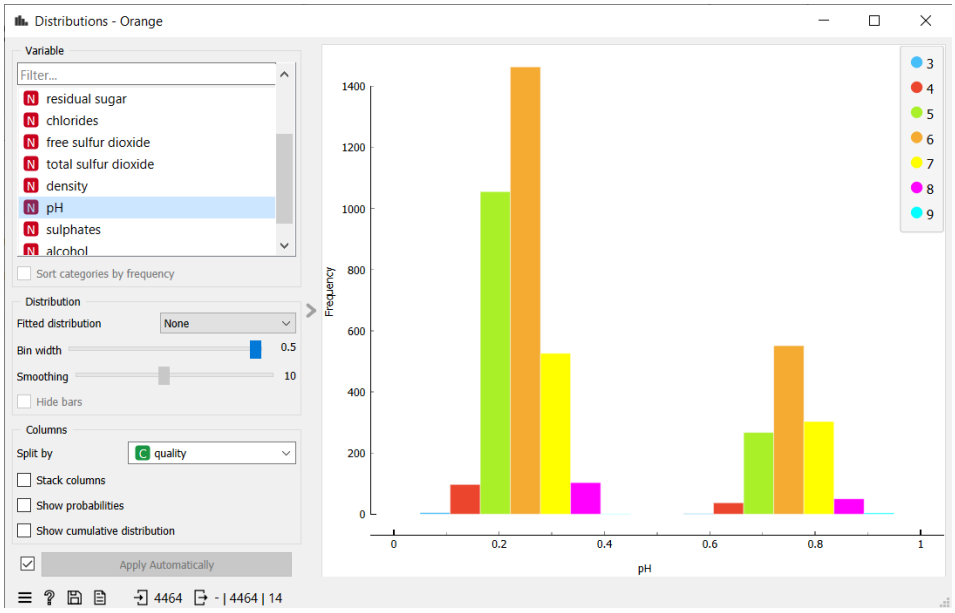
<a scatterplot screenshot>



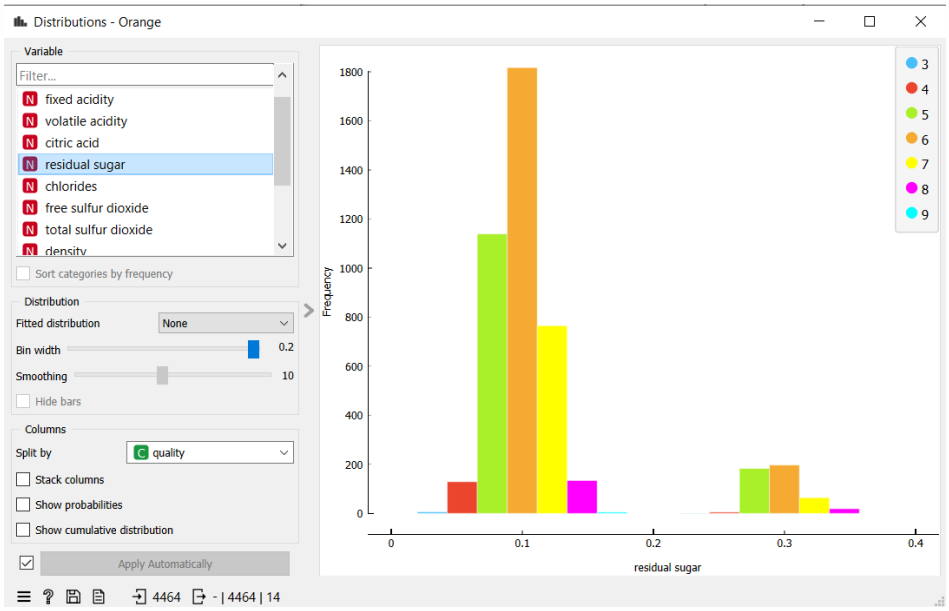
<a scatterplot screenshot>



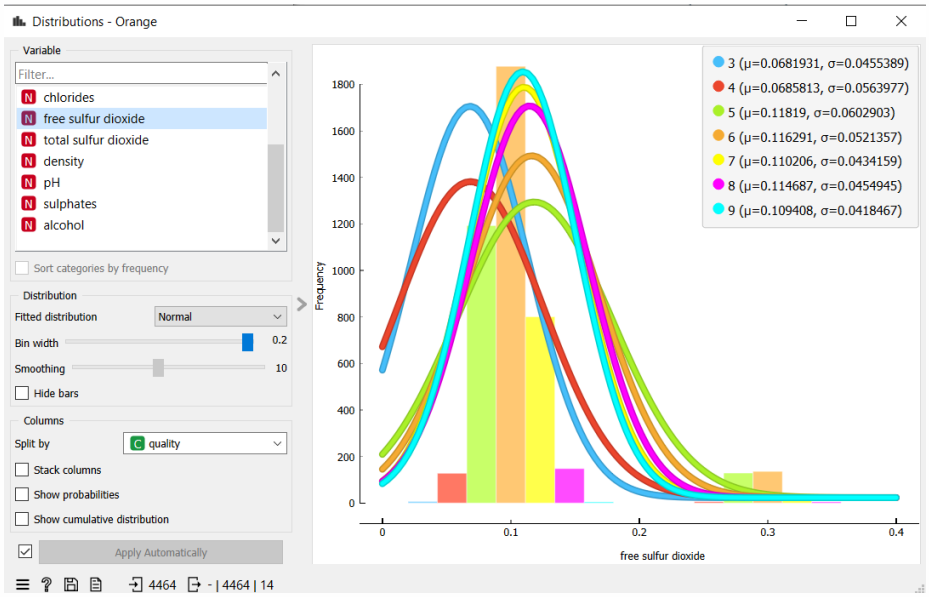
<a histogram screenshot>



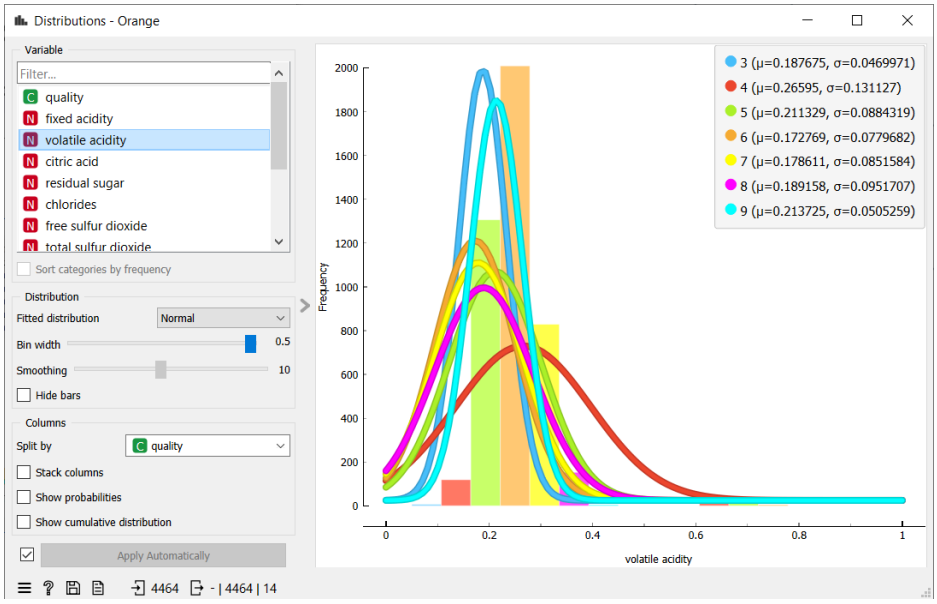
<a histogram screenshot>



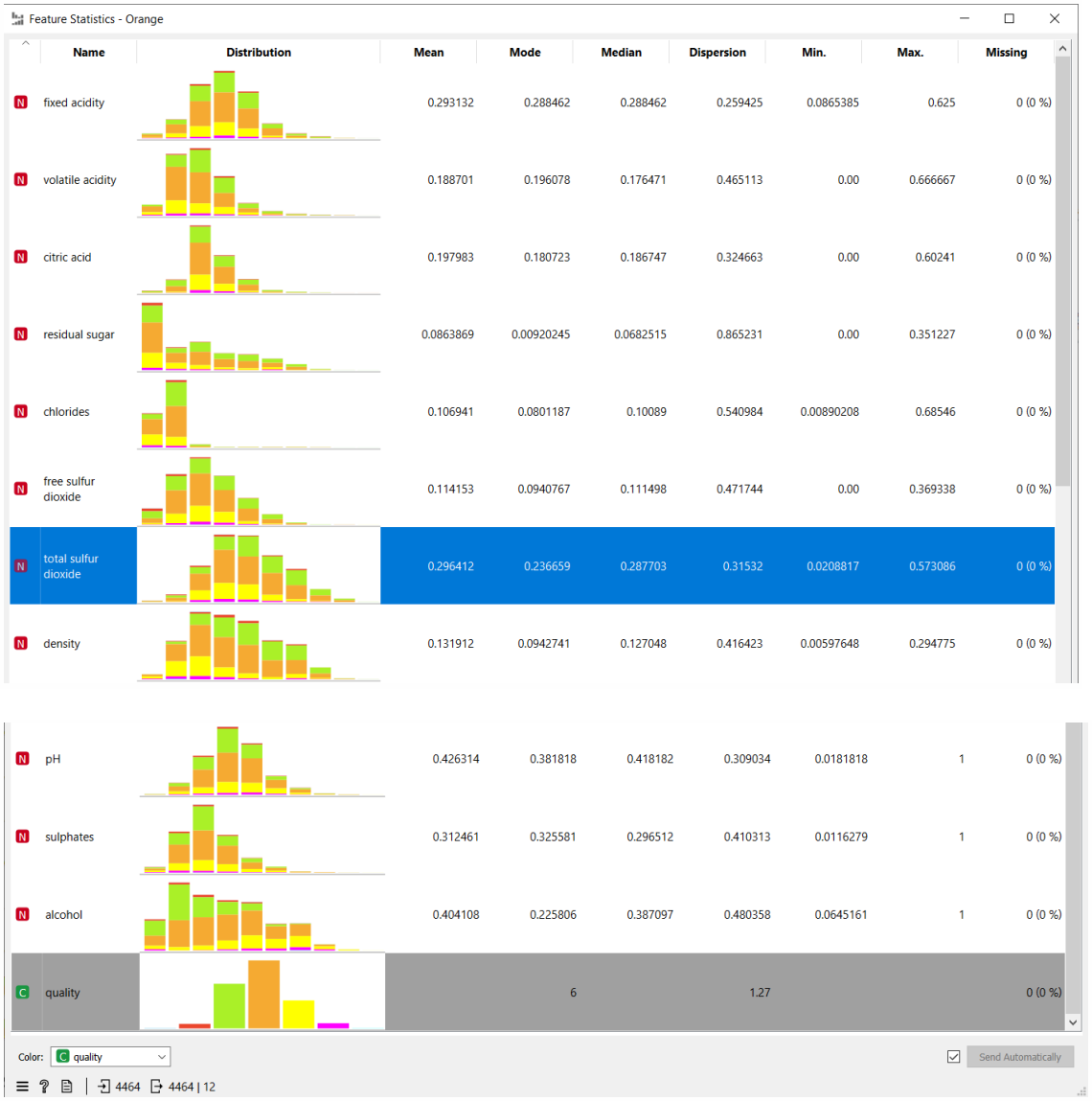
<a screenshot of feature distribution>



<a screenshot of feature distribution>



<a screenshot with statistics>

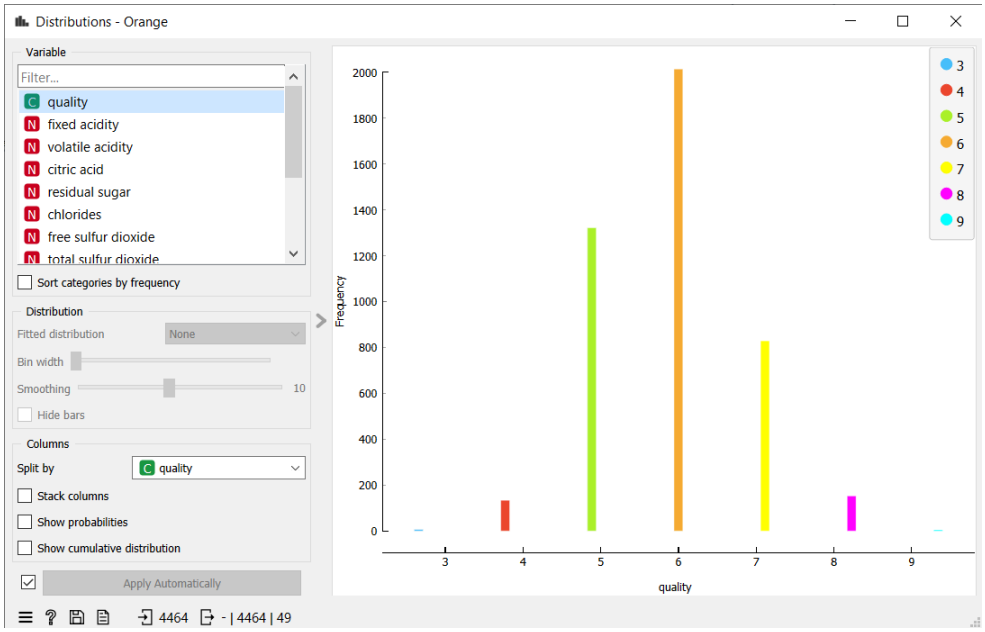


Answers to questions

<answers the questions below, referring to the screenshots above and providing an analysis of the results>

Are the classes in the dataset balanced, or does one class (or several classes) prevail?

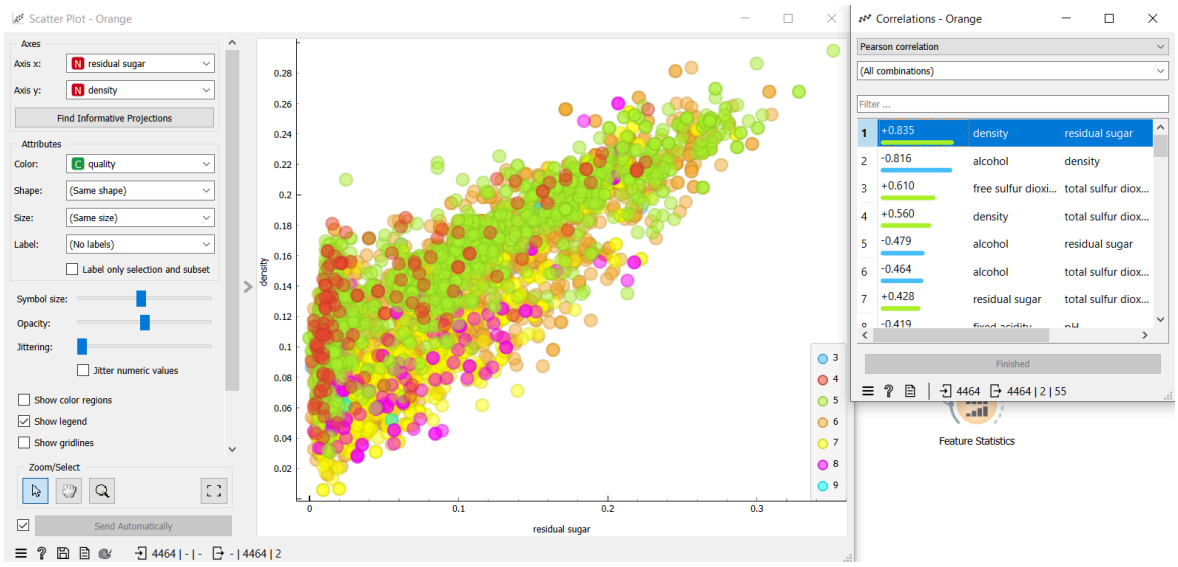
This dataset is not balanced, and the classes in it are not ordered. This can be understood by looking at the number of normal, good and bad wines. There are many more normal wines than bad or good ones. Processing such datasets is important for building reliable machine learning models.



We can see it on this histogram, wines with the normal quality such as 5, 6 and 7 have the largest numbers.

Does the visual representation of the data allow you to see the structure of the data?

Yes, the visual representation allows us to see the structure of the data, evaluate outliers, examine the density of objects, identify clusters of points, and examine linear and nonlinear correlation.



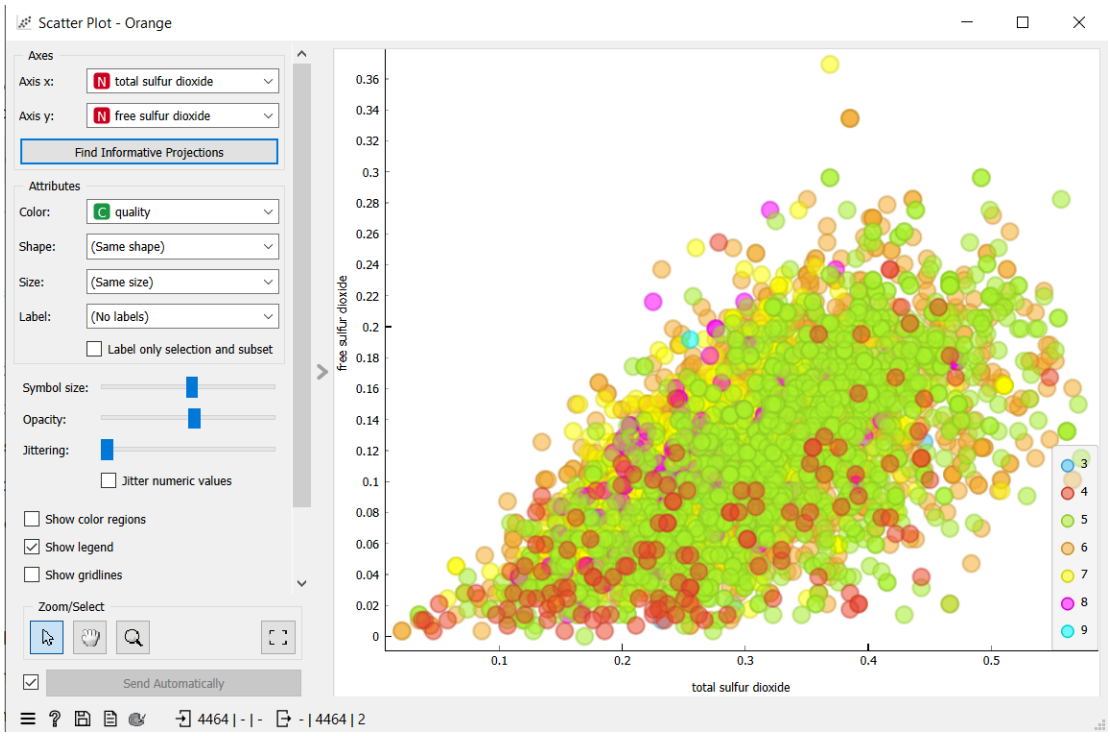
Here we can see that the largest correlation in the data is between density and residual sugar, this is clearly shown in the scatter plot, the correlation is strong and linear.

How many data groupings can be identified by studying the visual representation of the data?

In this dataset, we were unable to identify groups of data because the data is densely located with each other.

Are the identified data groupings close to each other or far from each other?

Almost all objects are located close to each other, but in the statistical data we can see the outliers.



Like on this scatter plot.

Conclusions arising from the analysis of statistical indicators

<analysis of statistical indicators by referencing specific values>

1. Fixed Acidity:

   Mean: 0.293132

   Mode: 0.288462

   Median: 0.259425

   The range from 0.0865385 to 0.625.

   The distribution seems positively skewed, the mean a little higher than median.

2. Volatile Acidity:
   Mean: 0.188701

   Mode: 0.196078

   Median: 0.176471

   The range 0.00 to 0.666667.

   The distribution seems negatively skewed, the mean lower than median.

3. Citric Acid:
   Mean: 0.197983

   Mode: 0.180723

   Median: 0.186747

   The range from 0.00 to 0.60241.

   The distribution seems positively skewed, the mean a little higher than median.

4. Residual Sugar:
   Mean: 0.0863869

   Mode: 0.00920245

   Median: 0.0682515

   The range from 0.00 to 0.351227.

   The distribution seems positively skewed, the mean a higher than median.

5. Chlorides:
   Mean: 0.106941

   Mode: 0.0801187

   Median: 0.10089

   The range from 0.00890208 to 0.68546.

   The distribution seems positively skewed, the mean a little higher than median.

6. Free Sulfur Dioxide:
   Mean: 0.114153

   Mode: 0.0940767

   Median: 0.111498

   The range from 0.00 to 0.369338.

   The distribution seems positively skewed, the mean a little higher than median.

7. Total Sulfur Dioxide:

   Mean: 0.296412
   Mode: 0.236659

   Median: 0.287703

   The range from 0.0208817 to 0.573086.

   The distribution seems positively skewed, the mean a little higher than median.

8. Density:
   Mean: 0.131912

   Mode: 0.0942741

   Median: 0.127048

   The range from 0.00597648 to 0.294775.

   The distribution seems positively skewed, the mean a little higher than median.

9. pH:
   Mean: 0.426314

Mode: 0.381818

Median: 0.418182

The range between 0.0181818 to 1.

The distribution seems positively skewed, the mean a little higher than median.

10. Sulphates:
    Mean: 0.312461

    Mode: 0.325581

    Median: 0.296512

    The range from 0.0116279 to 1.

    The distribution seems positively skewed, the mean a little higher than median.

11. Alcohol:
    Mean: 0.404108

    Mode: 0.225806

    Median: 0.387097
    The range from 0.0645161 to 1.

    The distribution seems positively skewed, the mean a little higher than median.

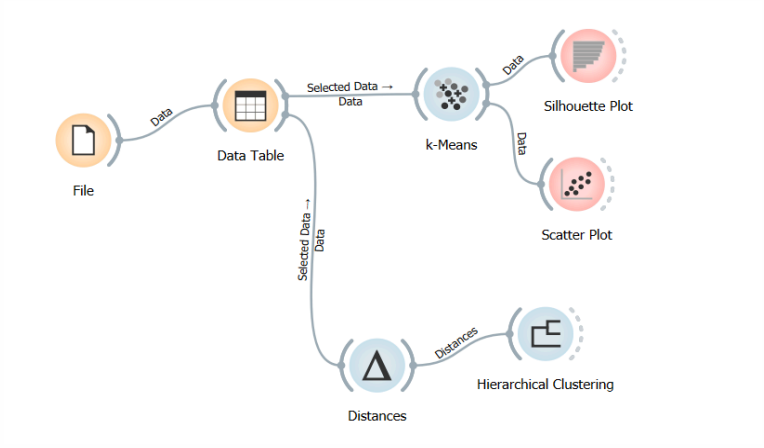
12. Quality:

    The average quality rating is 6, which means a fairly high level of quality of the wines provided.

Part II

<this subsection should describe the use of unsupervised machine learning algorithms, accompanied by screenshots and references to the information sources used>

Unsupervised machine learning algorithms, also called learning without a teacher, are used to find patterns or structure in data without the need for labeled output. Unlike supervised learning, where the algorithm is provided with labeled examples to learn from, unsupervised learning algorithms operate on unlabeled data and must discover patterns or relationships on their own.



Hierarchical clustering

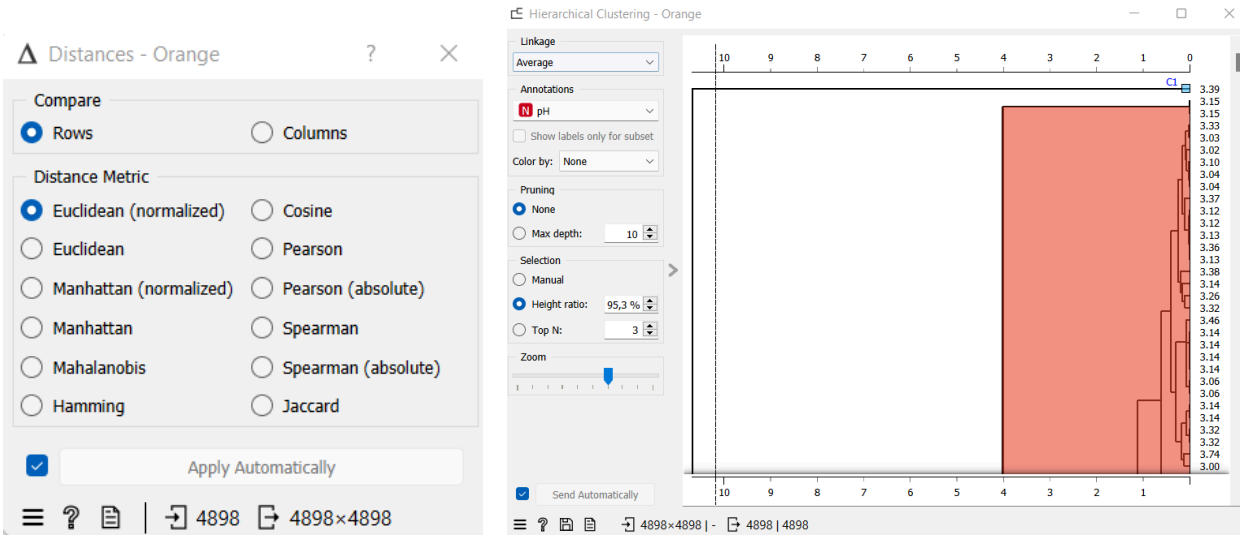
Hyperparameters available in the Orange tool:

<add rows to table as needed>

Hyperparameter	Description
Linkage method	Hierarchical clustering involves merging clusters based on a linkage criterion. The linkage method determines how the distance between clusters is measured during the merging process.

Distance metric	Hierarchical clustering requires a distance metric to measure the dissimilarity between data points or clusters.
Cutoff Threshold	Hierarchical clustering generates a dendrogram, which represents the merging process of clusters. The cutoff threshold specifies when to stop merging clusters.

<a screenshot with hyperparameter values set for the algorithm>



Description of experiments

3 experiments : citric acid, residual sugar, chlorides

Hyperparameters values :
- Linkage : single
- Distance metric : euclidean
- Cutoff Threshold : visual inspection of the dendrogram

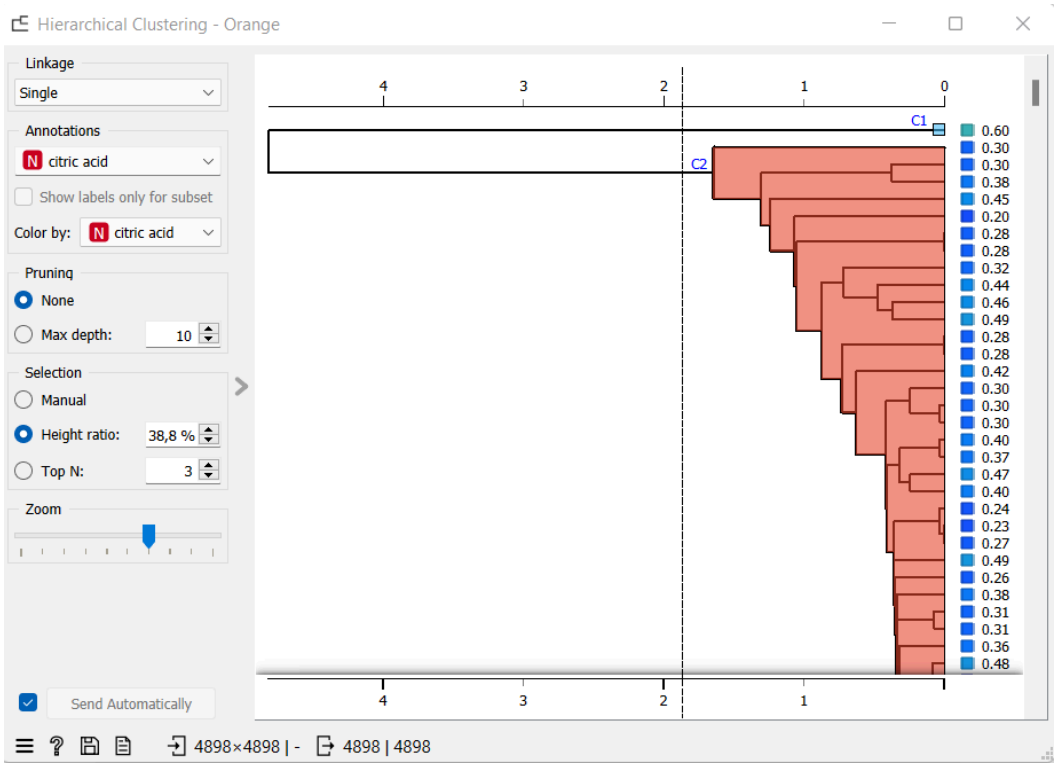
Experiment Description:

1. Apply hierarchical clustering to the "Wine Quality White" dataset using the linkage single and Euclidean distance as the distance metric.
2. Visualize the dendrogram to observe the hierarchical structure of the clustering process.
3. Use the cutoff line to observe how the number of clusters changes and assess the separability between them.
4. Compare the resulting clusters with the known quality ratings in the dataset.

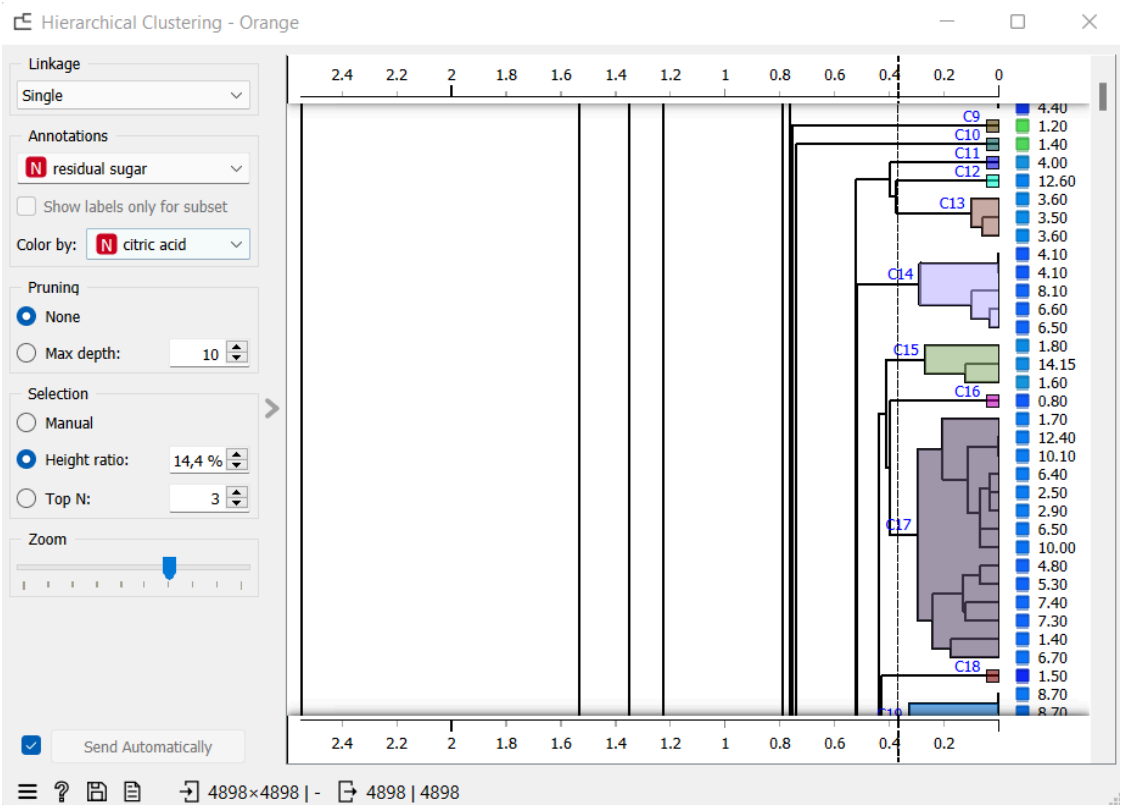
Conclusions:

- The dendrogram structure and clustering results indicate the presence of distinct clusters.
- Compare the clusters obtained from hierarchical clustering with the known quality ratings to evaluate how well they correspond. Higher correspondence indicates better performance in capturing the underlying structure of the data.

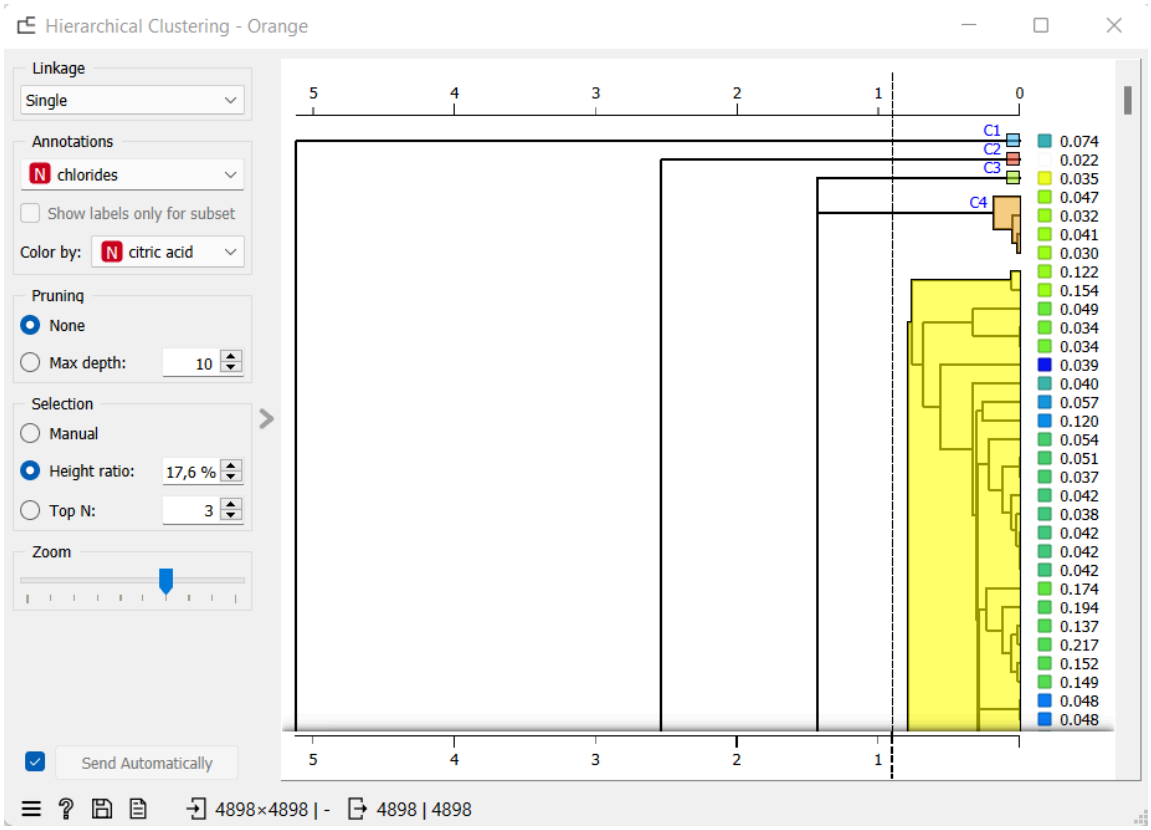
<a screenshot for Experiment 1 with a certain position of the horizontal cut off line>



<a screenshot for experiment 2 with a certain position of the horizontal cut off line >



<a screenshot for Experiment 3 with a certain position of the horizontal cut off line >



Conclusions from experiments:

<a description of how the number and content of clusters change according to the position of the horizontal cut off line, and conclusions about whether class separation is achieved referring to and analysing the screenshots above>

In the dendrogram, when we slide the cut off line from left to right we can see several numbers and contents of clusters which appear little by little. For example, we notice in the first screenshot the content of the cluster C2, and when we slide the cut off line toward the right the content of the cluster C2 is a merger of several clusters as we can see in the two others screenshots. Moreover, the content of a cluster has several data objects but in a certain position of the horizontal cut off line it has only one data object, for example in the third screenshot the cluster C2 has just one data object which is 0,022 but if we come back to the left with the cut off line we can see again his content and he has a lot of data objects. Finally, a good separation in hierarchical clustering results in clusters that are internally homogeneous and externally heterogeneous. Each cluster should contain data objects that are similar to each other based on the chosen distance metric. However, well-separated clusters should have distinct boundaries and contain data points that are dissimilar to those in other clusters. The content of each cluster should be distinct from the content of neighboring clusters. Distinctiveness between clusters indicates clear separation and meaningful clustering of data. For example, the cluster C13 in the residual sugar has similar data points so it's a good separability within this cluster but regarding the cluster C17 it has different data objects so it's not a good separability. However we can say

there is a good separation between these two clusters because their content is distinct in which the data objects of C13 are different from those of C17. And if we compare two dendrograms of two features such as citric acid and chlorides, the data objects or content of the clusters are distinct.

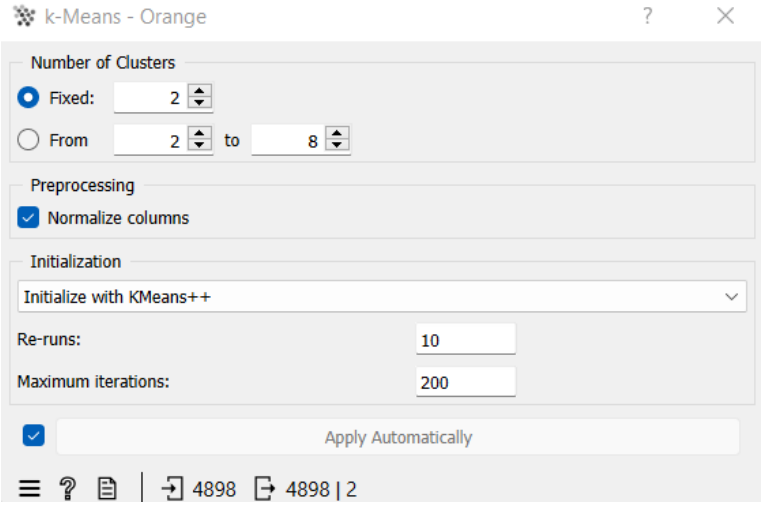
K-means algorithm

Hyperparameters available in the Orange tool:

<add rows to table as needed>

Hyperparameter	Description
Number of clusters (k)	This hyperparameter specifies the number of clusters that the algorithm should partition the data into.
Initialization Method	K-means requires initial centroids for each cluster. The initialization method determines how these initial centroids are selected.
Maximum Iterations	K-means is an iterative algorithm that converges to a solution. This hyperparameter specifies the maximum number of iterations allowed before the algorithm stops. Setting a maximum number of iterations prevents the algorithm from running indefinitely and ensures convergence.

<a screenshot with hyperparameter values set for the algorithm>



Description of experiments

5 experiments : density, ph, alcohol, quality, fixed acidity

Hyperparameters values :
- Number of clusters (k) : 2
- Initialization Method : Initialize with KMeans++
- Maximum Iterations : 200

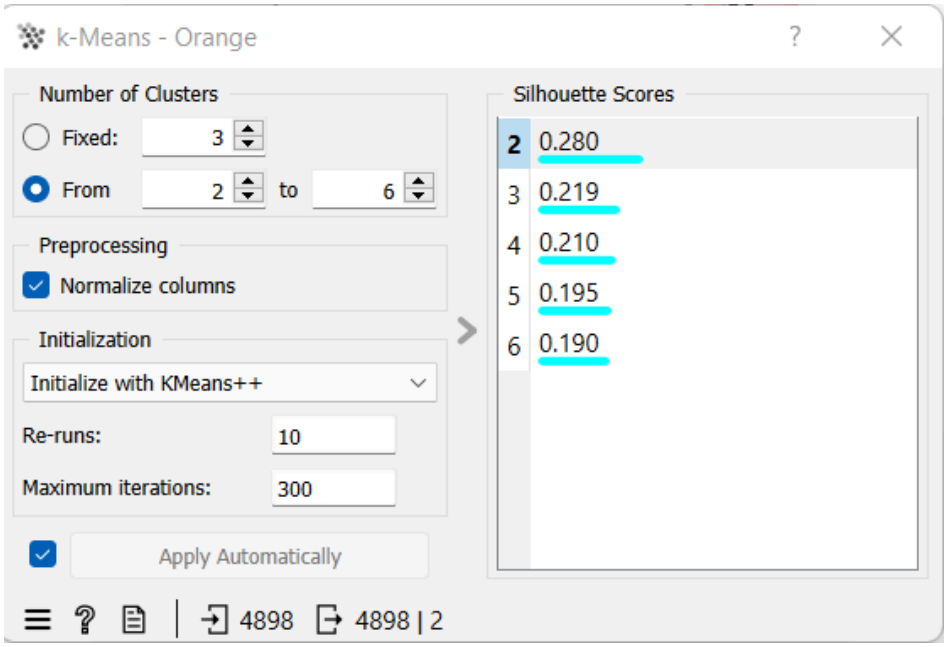
Experiment Description:

1. Apply K-means clustering to the "Wine Quality White" dataset with varying values of k.
2. Calculate the Silhouette coefficient for each clustering configuration to assess clustering quality.
3. Visualize the scatter plots to observe cluster separation.
4. Compare the resulting clusters with the known quality ratings in the dataset.

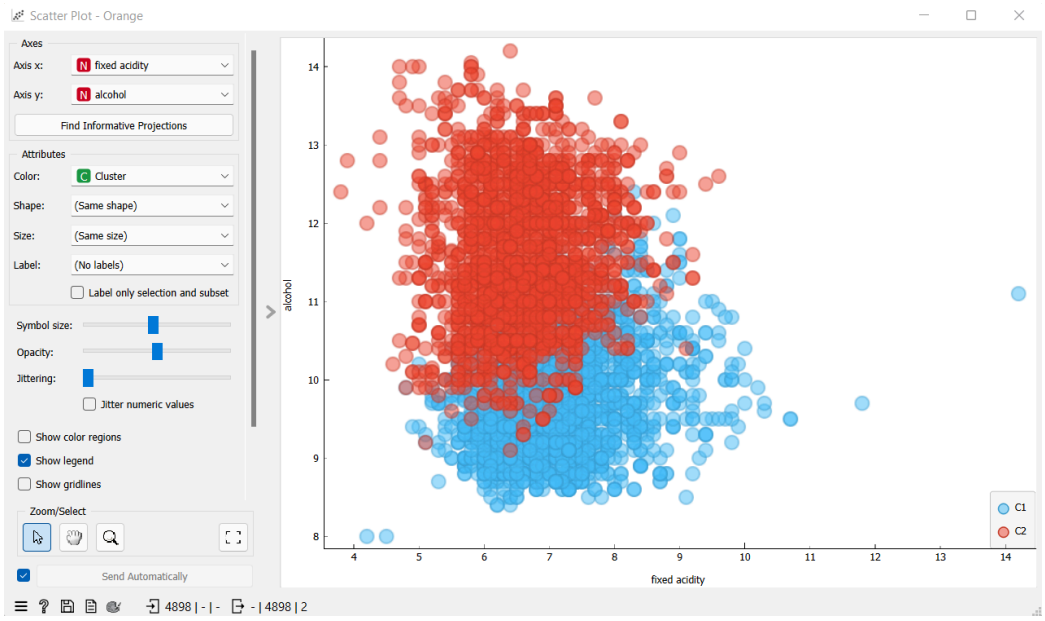
Conclusions:

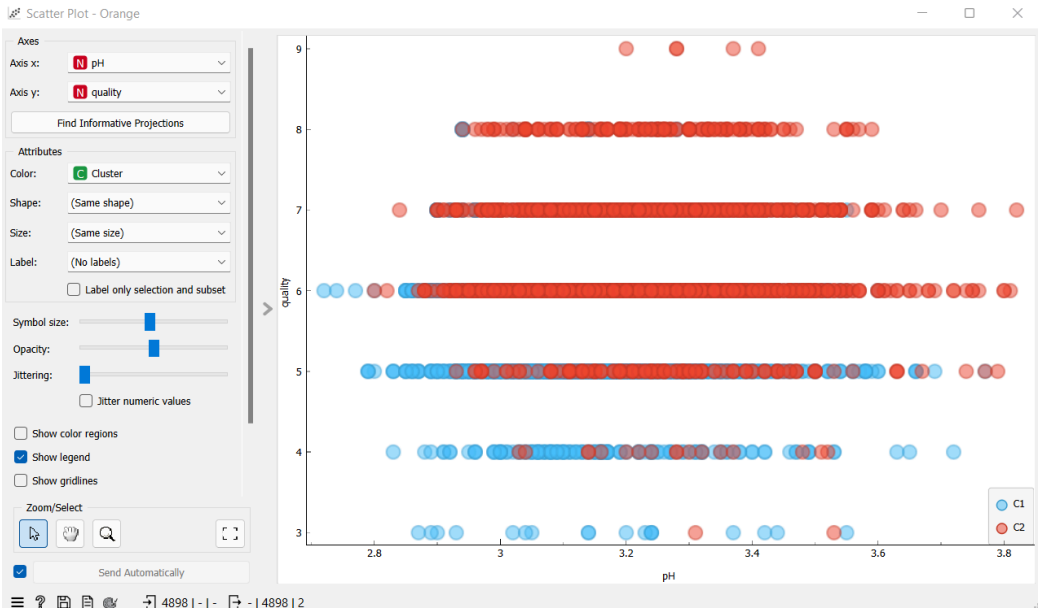
- Evaluate clustering quality using the Silhouette coefficient, aiming for higher values indicating better separation.
- Visual inspection of scatter plots helps assess the separability of clusters.
- Compare the obtained clusters with the known quality ratings to determine how well they capture the inherent structure of the data.

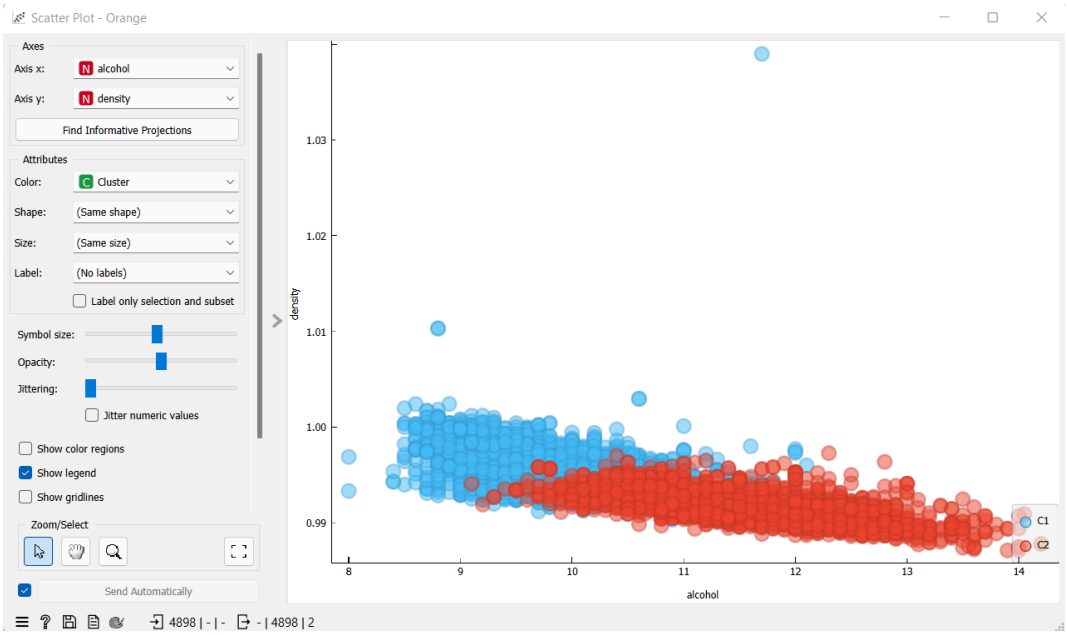
<a screenshot of Silhouette coefficient with at least 5 different k values>



<a screenshot of the scatterplot according to the best value of the Silhouette coefficient>







Conclusions from experiments:

<a description of whether the classes are separable referring to and analysing the screenshots above>

We can see in the screenshot of the Silhouette coefficient there are 5 values which are the Silhouette score. We notice these values aren't closer to 1, it means it's not a good separability. But the higher number is the second iteration with a Silhouette score of 0,280; it might be a good result in the scatterplot. For example, we can see in the first screenshot of the scatterplot

by analyzing fixed acidity and alcohol, the data points of the two clusters are too near each other and we can confuse them (they are similar); it shows there isn't a good separability.

Final conclusions

<conclusions whether the classes in the dataset are well or poorly separable based on the analysis of the performance of the two algorithms>

The results obtained from the two algorithms are different. Indeed, regarding the K-means the Silhouette score shows the values aren’t closer to 1, it means it’s poorly separable especially as the scatter plot confirms this result because the groups of data objects of each cluster are confused.
About the hierarchical clustering, the dendrogram shows by sliding the cutoff line toward the right the number of clusters multiplies and their content changes, so the result is in function of this analysis. There is a good separability if the data objects within a cluster are similar and the content of each clusters is distinct.

Part III

We need to choose at least two supervised machine learning algorithms (we choose artificial neural networks because it's in the task and also choose kNN and Random Forest. Also we split the dataset into training and test datasets, then perform at least 3 experiments with each algorithm using the training dataset. For each algorithm, we select the trained model that provides the best algorithm performance And in the end we are evaluating and comparing the performance.

Description of the selected algorithms

Our team selected two additional algorithms, namely: kNN and Random Forest. These 2 algorithms were chosen because they are among the easiest to understand. They are also very flexible and can be involved in a large number of tasks. The principle of their operation is intuitive, which makes working with them easier and more efficient, also the principle of their operation is different, one looks at the majority of neighboring elements, and the other at the average sample of all data. So with the help of these algorithms, we will be able to analyze our data sample in more detail, and make a more detailed analysis of the results.

Title of the first algorithm: **kNN**
Description of the first algorithm:

The kNNalgorithm (k-nearest neighbors algorithm) was first invented in 1951. The structure of the model in this algorithm is not defined by any clear lines, but entirely depends on the input data, that is, this means that this algorithm is non-parametric. Also, kNNis one of the simplest classification algorithms, is intuitively clear, and has a wide range of applications. In simple words, the essence of the algorithm is to classify the elements in the sample, according to the scheme, which neighbors are more around you, and you belong to them.

If the algorithm is aimed at classifying an object, the object will be assigned a class that is most common among its nearest k-neighbors.

When using an algorithm for regression, an object is assigned an average value over its nearest k-neighbors, the values of which are already known.

Title of the second algorithm: **Random Forest**
Description of the second algorithm:

The Random Forest algorithm is a very simple and understandable algorithm, which, thanks to its flexibility, can be used in a very large number of problems, which makes it very useful in data analysis. The very essence of the algorithm consists in making a forest of individual trees, and each tree in turn consists of a separate sub-sample of

data, thus we will receive a very large number of not very high-quality results, but their quality is compensated by their quantity. Therefore, after receiving data from all trees, we take only the most frequently occurring responses as an answer, thus we will get our analysis of the entire data sample.

If we describe the course of the algorithm in simple words, then: we load our entire sample; we make a random sample from it; we build the course of the decision based on this sample, and build a tree based on this decision; the tree will be built until there are n-elements in each leaf; then we will get the result from each tree; and at the end we look for the largest number of similar elements in the answers for the entire sample.

Description of hyperparameters

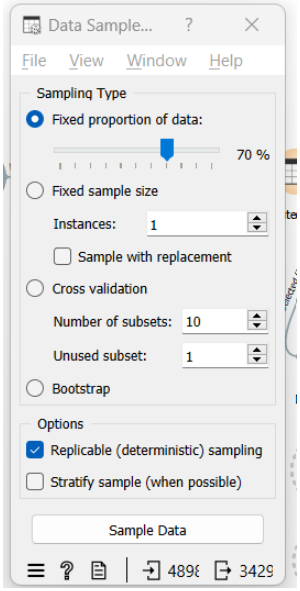
<a description of the hyperparameters available in the Orange tool should be given for each of the algorithms, adding rows to the table as necessary>

Hyperparameters	Description and values
Artificial neural network	
Neurons in hidden layer	It's an integer or a list of integers representing the number of neurons inside the neural network.
Activation	This is the mathematical function used to determine the activation of a neuron. It brings the computed value of a neuron between 0 and 1. Examples of those are the sigmoid function or even the ReLu function
Solver	This is the optimization algorithm used to train the network. It determines how weights are updated during training. The most basic one is the stochastic gradient descent.
Regularisation	The regularization parameter is a float that controls the strength of the regularization penalty. It is used to prevent overfitting, which is when the network becomes too specific.
Maximum number of iteration	This is the value that specifies how many times the training dataset will be presented to the network for training.
kNN	
Numbers of neighbors	The number of nearest neighbors that will be found will directly affect the result.
Metric	A method of measuring the distance between two points in space that plays a key role in kNN.

Weight	Evaluation of the quality and impact of neighbors on the result, which will be considered more important when evaluating a neighbor.
Random forest	
Number of trees	The number of trees that will be created during data analysis. A larger number of trees has a positive effect on the accuracy of the results.
Number of attributes considered at each split	The larger the value in this parameter, the more diverse trees we will have in the sampler, which in turn will give more accurate results. But an excessive increase will lead to an increase in the cost of computing capabilities of the computer.
Replicable training	Controls the random generation at each repeated training, helps to get an accurate and understandable result after each restart of the model.
Balance class distribution	Impact on the balance sheet and result. This balances the price of each given class and prevents the results from being skewed towards the most common class.
Limit depth of individual trees	The greater the depth of the trees, the more it will relearn, of course it will work very well on the same data, but it will be difficult for it on others.
Do not split subsets smaller than	The number of elements in the subset that will be issued to each tree.

Information about test and training datasets

<a screenshot of splitting dataset into test and training datasets>



Number of data objects in the training dataset: 3429

% proportion of data objects in the training dataset:

Class label	Number of data objects in the training dataset	% proportion of data objects in the training dataset
1	0	0
2	0	0
3	18	0.01
4	126	0.03
5	1063	0.31
6	1465	0.43
7	618	0.18
8	133	0.04
9	5	0.00
10	0	0

Number of data objects in the test dataset: 1469

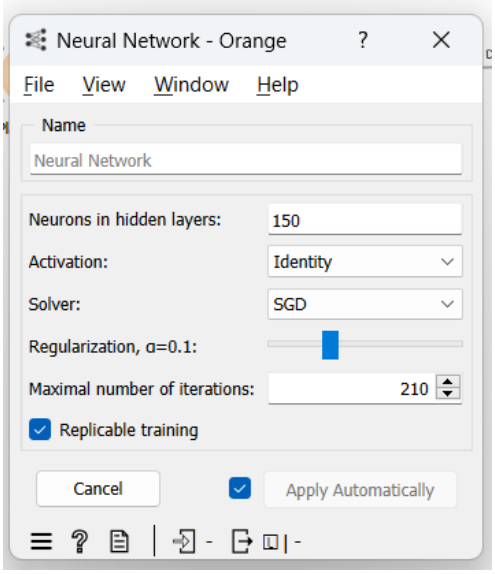
% proportion of data objects in the test dataset:

Class label	Number of data objects in the test dataset	% proportion of data objects in the test dataset
1	0	0
2	0	0
3	2	0.00
4	37	0.03
5	394	0.27
6	734	0.50
7	262	0.18
8	42	0.03
9	0	0
10	0	0

Experiments with artificial neural network

Experiment	Hyperparameter values
Experiment 1	100, Identity, SGD, 0.1, 210
Experiment 2	20,Logistic, SGD, 0.5, 210
Experiment 3	500,Identity, SGD, 0.005, 210

<a screenshot of hyperparameter values for Experiment 1>

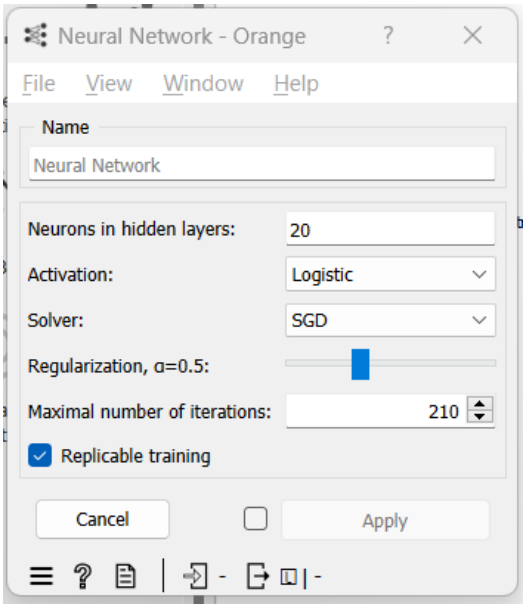


<a screenshot of performance metrics for Experiment 1>

Predicted

Actual	3	4	5	6	7	8	9	∑
3	0	0	2	4	0	0	0	6
4	0	4	30	16	1	0	0	51
5	0	0	225	234	4	0	0	463
6	0	1	113	505	38	0	0	657
7	0	0	9	175	56	0	0	240
8	0	0	2	32	13	0	0	47
9	0	0	0	1	4	0	0	5
∑	0	5	381	967	116	0	0	1469

<a screenshot of hyperparameter values for Experiment 2>

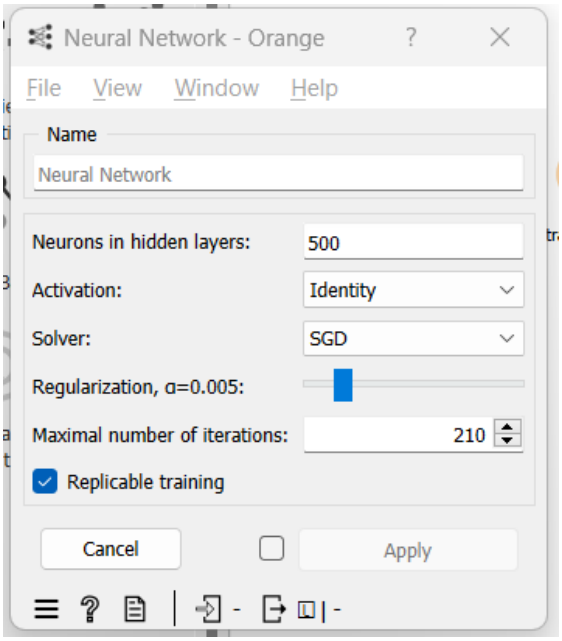


<a screenshot of performance metrics for Experiment 2>

Predicted

Actual	3	4	5	6	7	8	9	∑
3	0	0	2	4	0	0	0	6
4	0	0	29	21	1	0	0	51
5	0	0	199	263	1	0	0	463
6	0	0	93	562	2	0	0	657
7	0	0	11	221	8	0	0	240
8	0	0	4	42	1	0	0	47
9	0	0	0	5	0	0	0	5
∑	0	0	338	1118	13	0	0	1469

<a screenshot of hyperparameter values for Experiment 3>



<a screenshot of performance metrics for Experiment 3>

Predicted

Actual	3	4	5	6	7	8	9	Σ
3	0	0	2	4	0	0	0	6
4	0	4	29	17	1	0	0	51
5	0	0	237	222	4	0	0	463
6	0	1	124	493	39	0	0	657
7	0	0	9	176	55	0	0	240
8	0	0	5	28	14	0	0	47
9	0	0	0	1	4	0	0	5
Σ	0	5	406	941	117	0	0	1469

Conclusions from experiments:

- Increasing the number of neurons in the hidden layer doesn't mean an increase in the performance.
- Identity is a reliable activation function
- SGD (stochastic gradient descent) is a reliable solver function.

Model selected for testing:

The best experimental model we found is :

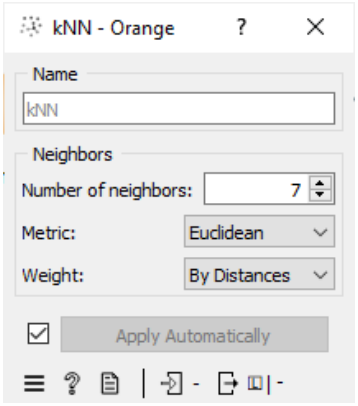
20,Logistic, SGD, 0.5, 210

Experiments with kNN

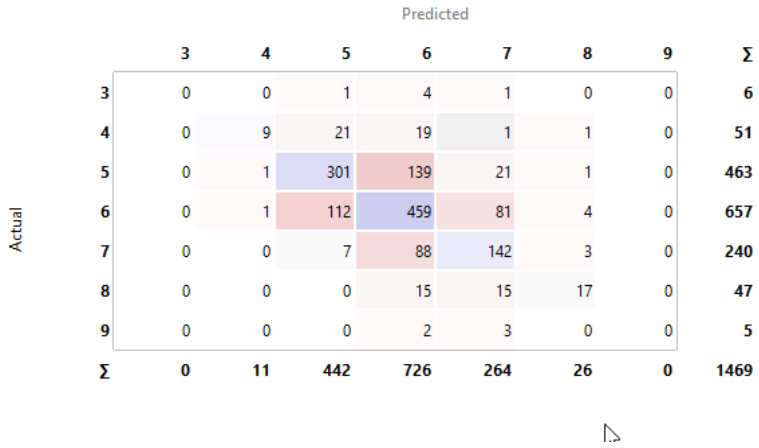
<add rows to table as needed>

Experiment	Hyperparameter values
Experiment 1	7, Euclidean, By Distances
Experiment 2	4, Euclidean , Uniform
Experiment 3	5, Chebyshev, Uniform

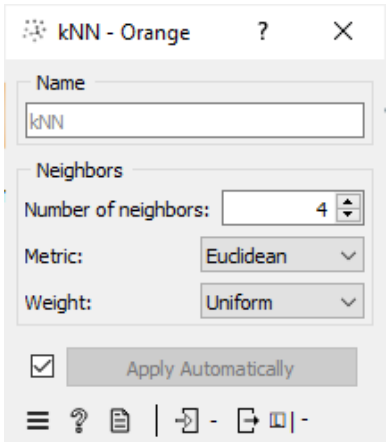
<a screenshot of hyperparameter values for Experiment 1>



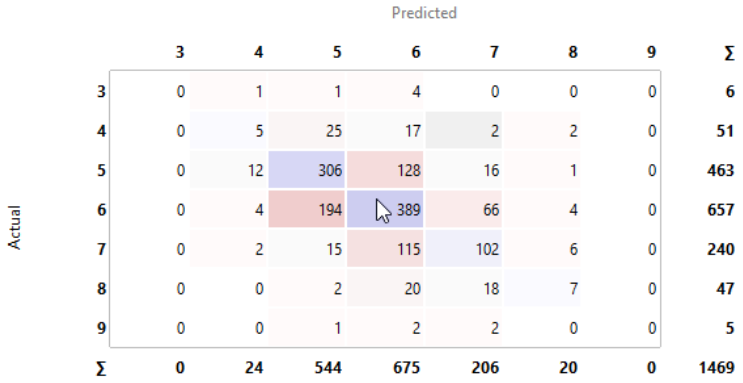
<a screenshot of performance metrics for Experiment 1>



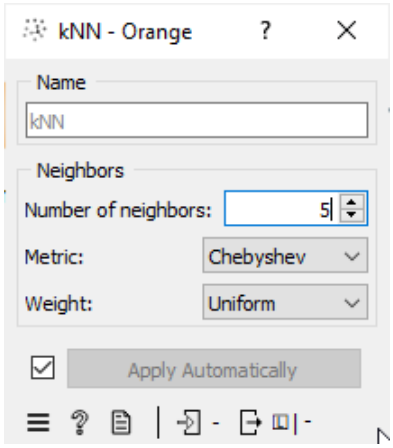
<a screenshot of hyperparameter values for Experiment 2>



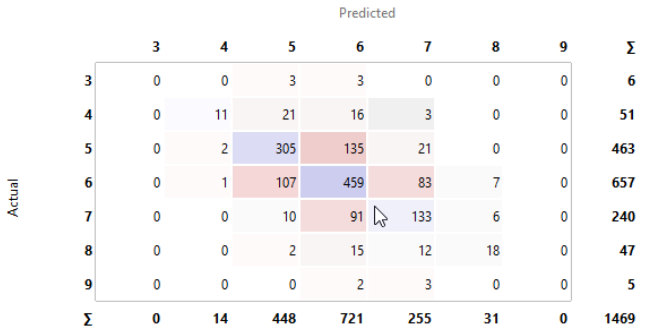
<a screenshot of performance metrics for Experiment 2>



<a screenshot of hyperparameter values for Experiment 3>



<a screenshot of performance metrics for Experiment 3>



Actual \ Predicted	3	4	5	6	7	8	9	Σ
3	0	0	3	3	0	0	0	6
4	0	11	21	16	3	0	0	51
5	0	2	305	135	21	0	0	463
6	0	1	107	459	83	7	0	657
7	0	0	10	91	133	6	0	240
8	0	0	2	15	12	18	0	47
9	0	0	0	2	3	0	0	5
Σ	0	14	448	721	255	31	0	1469

Conclusions from experiments:

- The number of neighbors affects the quality of the experiment
- The basis of the method is to determine a way to estimate the distance to neighbors
- Euclidean matrix, very sensitive to changes in the number of neighbors, the more it is specified in the parameters, the more accurate the value will be
- The best results were shown by the mix with the search for neighbors by distance and the Euclidean matrix

Model selected for testing:

We would like to choose next dataset :

7, Euclidean, By Distances

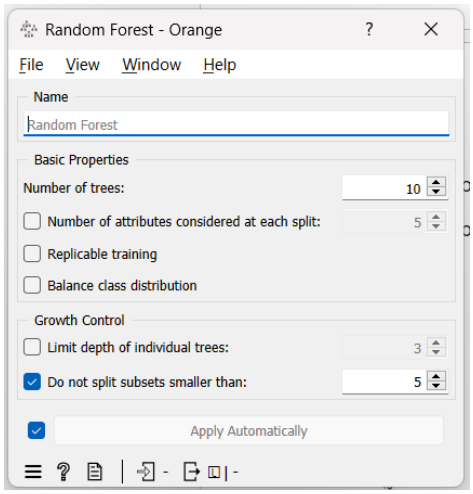
<indication of which experimental model is selected for the testing process>

Experiments with Random Forest

<add rows to table as needed>

Experiment	Hyperparameter values
Experiment 1	10,False,False,False,False,5
Experiment 2	30,False,False,False,False,20
Experiment 3	35,False,False,False,2,7

<a screenshot of hyperparameter values for Experiment 1>

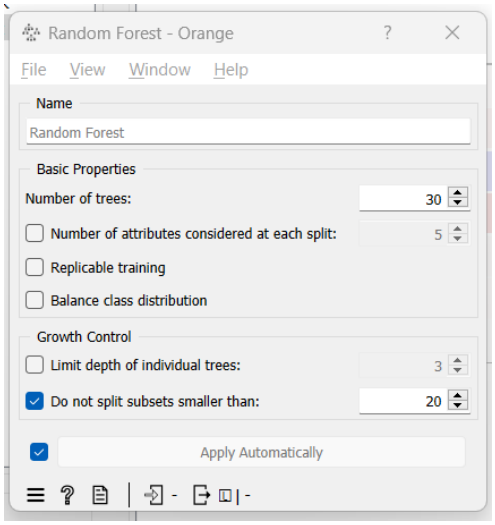


<a screenshot of performance metrics for Experiment 1>

Predicted

Actual	3	4	5	6	7	8	9	∑
3	0	1	2	2	1	0	0	**6**
4	0	6	28	16	1	0	0	**51**
5	0	5	310	136	12	0	0	**463**
6	0	0	108	482	67	0	0	**657**
7	0	0	7	109	122	2	0	**240**
8	0	0	1	24	8	14	0	**47**
9	0	0	1	1	3	0	0	**5**
∑	**0**	**12**	**457**	**770**	**214**	**16**	**0**	**1469**

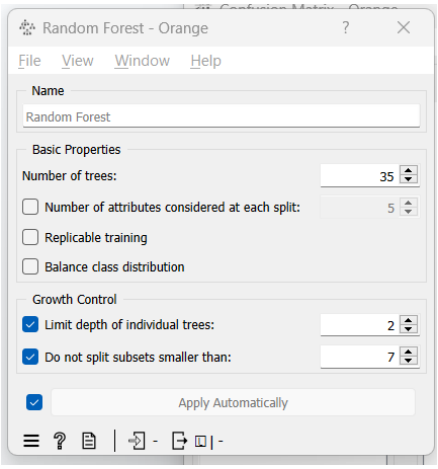
<a screenshot of hyperparameter values for Experiment 2>



<a screenshot of performance metrics for Experiment 2>

Actual \ Predicted	3	4	5	6	7	8	9	∑
3	0	0	2	3	1	0	0	6
4	0	1	33	16	1	0	0	51
5	0	0	283	172	8	0	0	463
6	0	0	102	521	33	1	0	657
7	0	0	6	126	107	1	0	240
8	0	0	1	26	16	4	0	47
9	0	0	0	2	3	0	0	5
∑	0	1	427	866	169	6	0	1469

<a screenshot of hyperparameter values for Experiment 3>



<a screenshot of performance metrics for Experiment 3>

Actual \ Predicted	3	4	5	6	7	8	9	∑
3	0	0	1	5	0	0	0	6
4	0	0	20	31	0	0	0	51
5	0	0	190	273	0	0	0	463
6	0	0	75	582	0	0	0	657
7	0	0	2	238	0	0	0	240
8	0	0	0	47	0	0	0	47
9	0	0	0	5	0	0	0	5
∑	0	0	288	1181	0	0	0	1469

Conclusions from experiments:

- Growth control is an important factor for performance.
- Like for the neural network, simply increasing the number of trees doesn't help

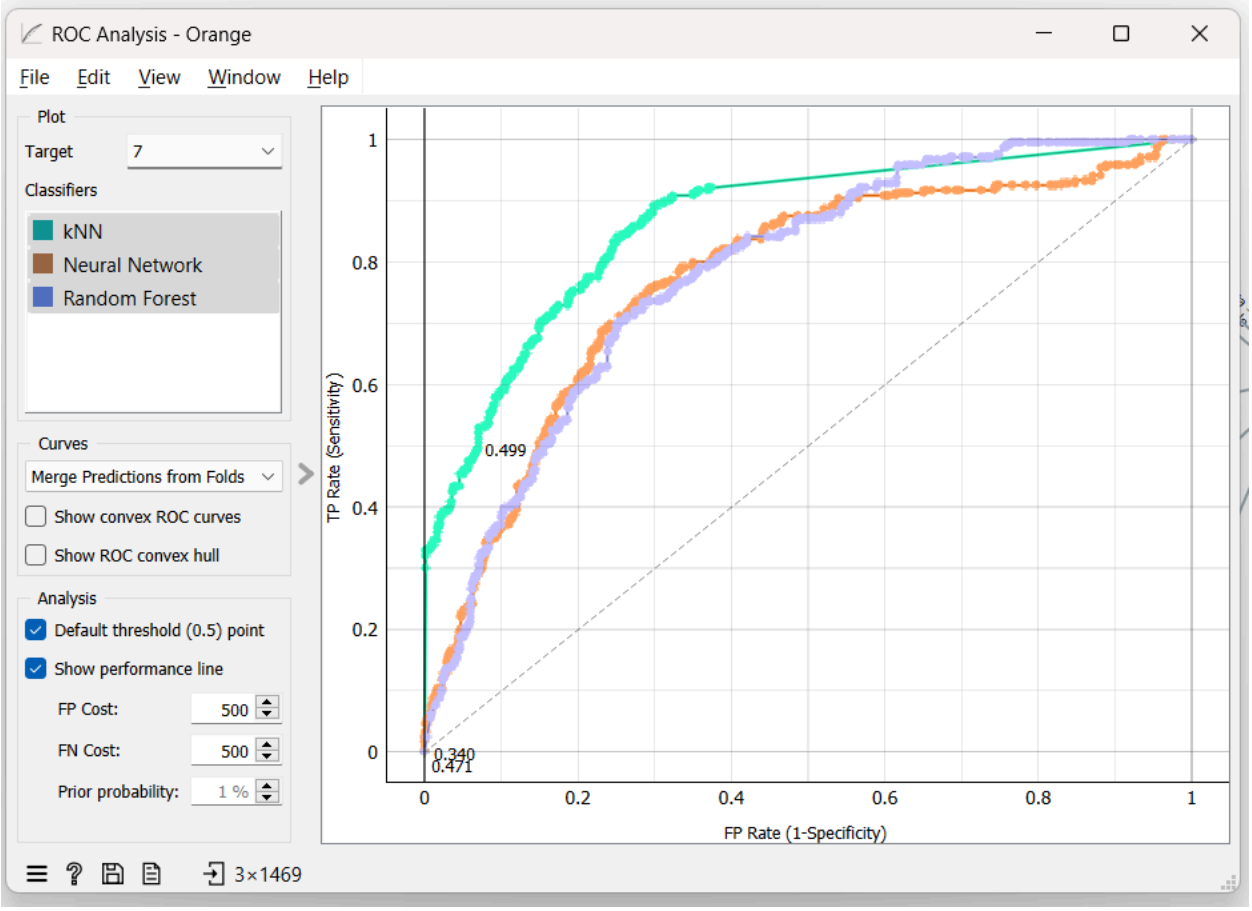
Model selected for testing:

We choose the following model for testing :

35,False,False,False,2,7

Testing results of the trained models

<a screenshot of performance metrics of models selected for testing>



Conclusions after testing:

The ROC analysis compares the number of true positive to the number of false positive. In other words the closer the curve is from the top left corner the more accurate it is. Hence we can see that after our few experiments, the kNN algorithm seems to be providing the best results.

Information sources

https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/data/outliers.html