CSE 521: DATA MINING PROJECT

USING WEKA AND ORANGE FOR DESCRIPTIVE AND NON-DESCRIPTIVE CLASSIFIERS ON BAKARY DATA

SUMMER 2021

ZEESHAN SHAIKH - 113221938
HONG WANG - 113263916

OUTLINE

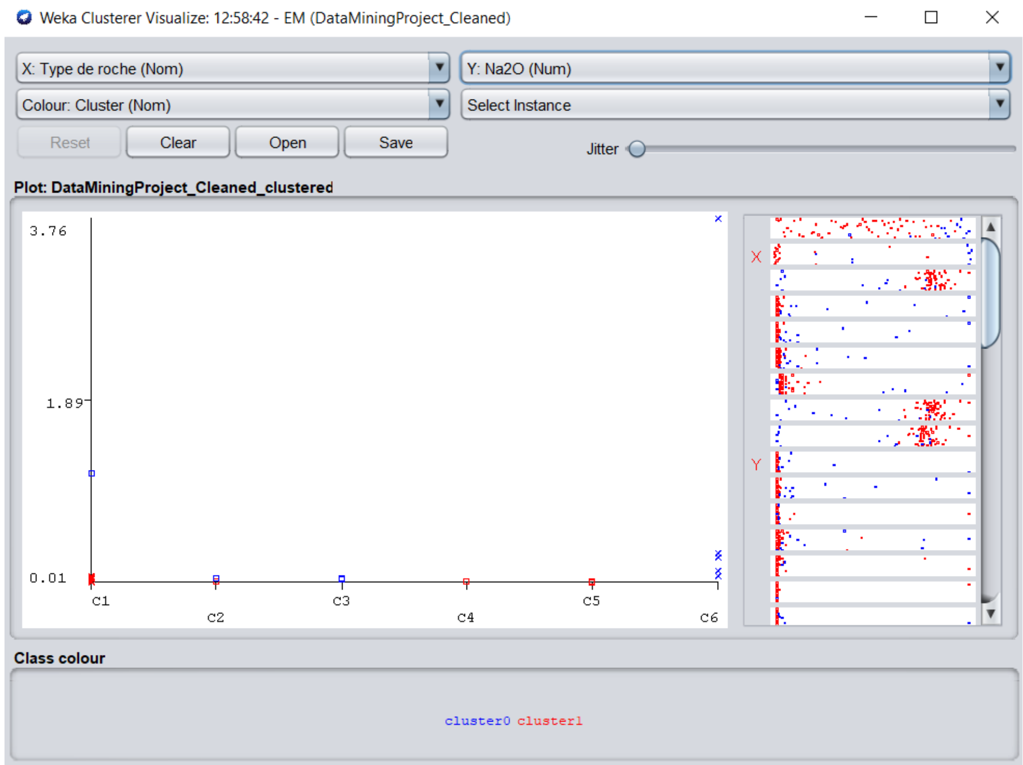
TOOL 1: WEKA

TOOL 2: ORANGE

Data Preparation

1. File Preparation
    - The file contains two extra spreadsheets which were empty, they were removed.
    - The initial dataset contains a duplicate of the class “type de roche”. This has been removed.
    - The attribute “Li” has an instance that contains “<”. The value was set as “< 3”. This has been changed to 2.99.
    - The attribute “U” had a special missing value with “?” entry. This was changed to empty and later dealt with in the “Missing Data” section.
    - An empty attribute called “ppms” was found and deleted.
    - The first attribute named “Enchillion” was removed as it was found to have no significance with the class prediction. The class “Type de roche” was moved to the end of the attributes.
    - It was noticed that since the names of the prediction class “type de roche” were very big, it made things unclear. Hence, the names of the class are changed to C1, C2, C3, C4, C5 and C6 respectively.
    - There were also some spelling errors in the name of the “type de roche” attributes which were solved by changing the names with the notations mentioned in the previous point.
    - Removed the first row as weka considered it as an empty value.

2. Outlier Detection:
    - For outlier detection, we have used Clustering as a method of identification. This can be done using the “Cluster” feature in Weka.
    - NOTE: Important point to be noted here is that while clustering, the data needs to be evaluated per class and not over it’s range since we are dealing with under-represented classes as well.
    - Example:



Here, you can notice that we spotted a clear outlier for class C6 and one outlier for class C1.
Outliers for all the attributes have been cleaned using the same method.

- The values have been removed using the “RemoveWithValues” filter which can be found under “weka.filters.unsupervised.instance”.
- The clusters can be visualised using the “Visualize Cluster” feature.
- Given the small number of data, only outliers in “Expert advised important” attributes were strictly evaluated (>3*sd). For all other attributes, outliers have been removed if they are > 5*sd. This is because if all the instances appearing to be outliers are removed, then the data is significantly reduced.

3. Missing Data:
    - In the given data, there are a total of 48 attributes (excluding the prediction class). Many of these attributes have missing values.
    - Case Scenario 1:
      For the attributes named “Co” and “Mo”, the missing values consist of 85% and 89% of the total data. As such, these are too high to have any meaningful contribution to the learning process and are thus dropped.

- Case Scenario 2:
  For the rest of the attributes, all of them have missing values. For these attributes, the missing values are replaced with the mean values.
  In Weka, you can do this by ReplaceMissingValues filter present under filters.unsupervised.attributes.

For Dataset : PD

1. Attribute Selection:
   Weka provides the ability to select the best possible subset based on user defined criterias. This can be found in filters under "supervised.attribute".
   The tool can be divided into two subparts:
   - Attribute Evaluator
   - Search Method

For our experiment, we used the "CfsSubsetEval" as it evaluates the worth of a subset of attributes by considering the individual predictive ability along with the degree of redundancy between them and the "BestFirst" search method as it uses a greedy hill climbing approach with a backtracking facility.

The motivation behind this is
The final set of attributes selected are: Al2O3, MgO, S, Zn, Cu, Cd, Sr and Rb.

For Dataset : PED

1. Attribute Selection:
   For this particular dataset, we will use the attributes which are determined as most important by the expert, which are: S, Zn, Pb, Cu, CaO+MgO, CaO, MgO and Fe2O3.

Data Preprocessing

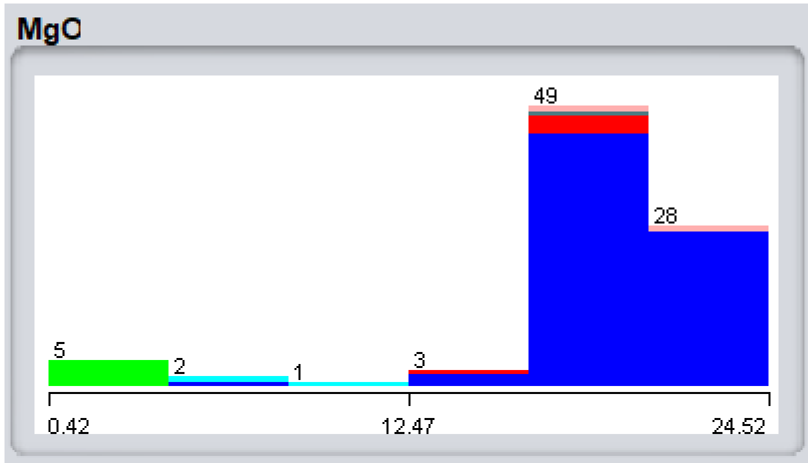
For Dataset: PD1 and PED1

1. Data Discretization:
   We implemented two methods of Discretization here, namely Equal Width and Equal Frequency Binning.

   Equal Width Binning:
   In this method, we decided to apply Equal Width Binning to the attributes which had a comparatively equitable distribution over the range of min to max values.
   The idea here is to bin the element with respect to width to create more balanced bins.

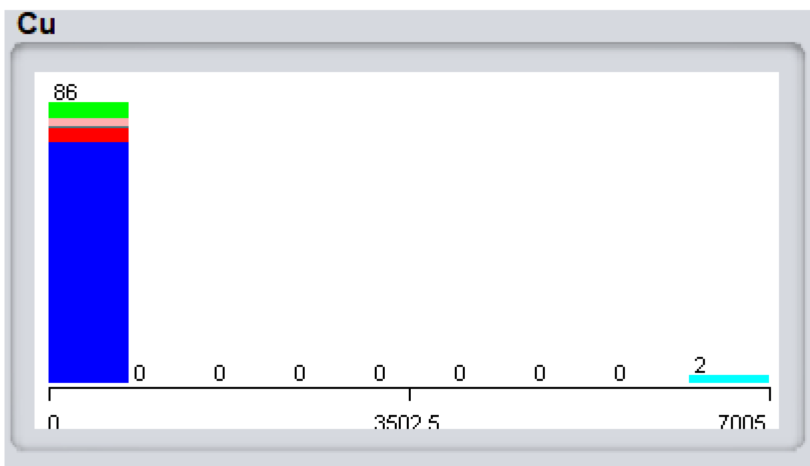
   This can be done in weka using the Discretize feature under weka.filters.unsupervised.attributes.
   NOTE: Weka provides a feature that allows for finding the optimized number of bins for a particular attribute. We set the maximum bins to 3!

Example of an attribute selected for Equal Width Binning:



Equal Frequency Binning:
For Equal Frequency Binning, we decided to choose attributes in which the data is highly skewed. In such cases, equal frequency binning should theoretically help with our splits for the tree.

Example of an attribute selected for Equal Frequency Binning:



2. Principal Component Analysis:
   For Experiment 1, we added Principal Component Analysis to check for its impact on performance. Although it improved the performance by a factor of 1-3%, it made understanding the rules of the tree quite difficult.
   Thus, we decided to not use this method in our final approach but instead, decided to highlight the use of PCA and its impact on the performance so that if the readability of

the decision trees isn't a requirement, this method can be used to gain a fraction of performance.

PCA can be applied by using the Filter “PrincipalComponents” under weka.filters.unsupervised.attributes.

3. Other methods used and considered:
   As a means of Normalization of data for our Neural Networks, we decided to experiment with both “Normalize” and “Standardize” filters available in weka. “Normalize”, as the name suggests, is used to Normalize the numeric data whereas “Standardize” uses a z-score metric.
   We found that for our dataset, both do not have any impact on the performance.
   These can be found under weka.filters.unsupervised.attributes

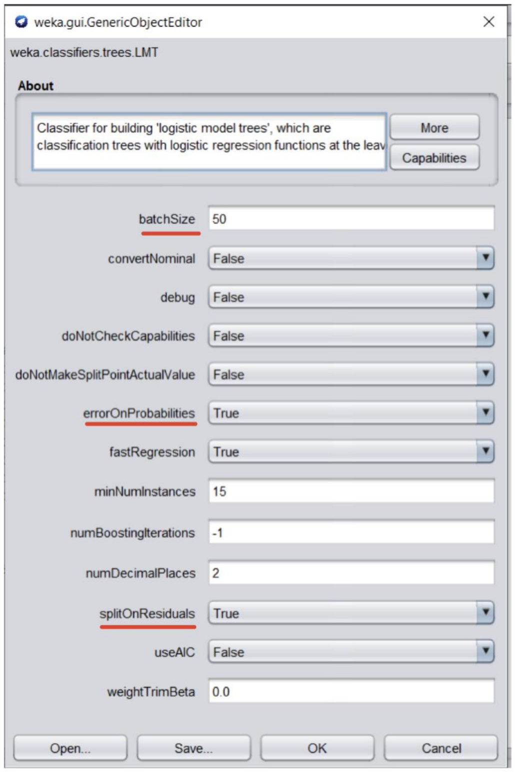
Experiment 1

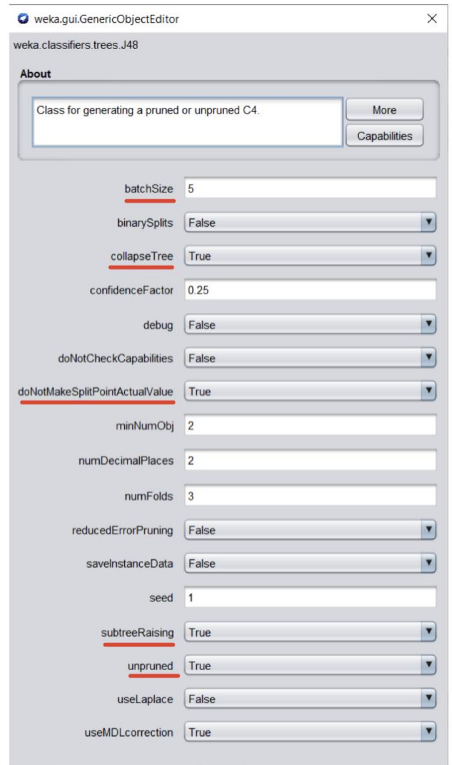
1. Decision Tree Classifier Using PD and PED

- Tool 1: WEKA
  The two trees used as Descriptive Classifiers are LMT and J48.

  Tree 1: LMT
  Training method: Cross-Validation with 25 folds.
  Tree 2: J48
  Training Method: 70-30 dataset split for training and testing.

Tree Configurations:





The red markers indicate the changes made from default configurations.

In-depth Analysis of both the trees for Dataset PD:

LMT.

```
Correctly Classified Instances          83               94.3182 %
Incorrectly Classified Instances         5                5.6818 %
Kappa statistic                          0.8016
Mean absolute error                      0.0206
Root mean squared error                  0.1394
Relative absolute error                 19.0318 %
Root relative squared error             63.087  %
Total Number of Instances               88

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.986    0.071    0.986      0.986    0.986      0.915    0.994     0.999     C1
                 0.750    0.000    1.000      0.750    0.857      0.861    0.991     0.893     C2
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     C3
                 0.000    0.023    0.000      0.000    0.000      -0.016   0.207     0.014     C4
                 0.000    0.012    0.000      0.000    0.000      -0.016   0.930     0.216     C5
                 1.000    0.012    0.833      1.000    0.909      0.907    0.998     0.967     C6
Weighted Avg.    0.943    0.061    0.945      0.943    0.943      0.882    0.984     0.963

=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
 73  0  0  0  0  1 |  a = C1
  1  3  0  0  0  0 |  b = C2
  0  0  2  0  0  0 |  c = C3
  0  0  0  0  1  0 |  d = C4
  0  0  0  2  0  0 |  e = C5
  0  0  0  0  0  5 |  f = C6
```

J48.

```
Correctly Classified Instances          80               90.9091 %
Incorrectly Classified Instances         8                9.0909 %
Kappa statistic                          0.6614
Mean absolute error                      0.0252
Root mean squared error                  0.1476
Relative absolute error                 23.2736 %
Root relative squared error             66.9218 %
Total Number of Instances               88

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.143    0.974      1.000    0.987      0.914    0.959     0.985     C1
                 0.750    0.024    0.600      0.750    0.667      0.653    0.872     0.761     C2
                 0.000    0.012    0.000      0.000    0.000      -0.016   0.988     0.667     C3
                 0.000    0.023    0.000      0.000    0.000      -0.016   0.489     0.011     C4
                 0.000    0.012    0.000      0.000    0.000      -0.016   0.988     0.667     C5
                 0.600    0.000    1.000      0.600    0.750      0.765    0.888     0.700     C6
Weighted Avg.    0.909    0.122    0.903      0.909    0.903      0.840    0.947     0.933

=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
 74  0  0  0  0  0 |  a = C1
  0  3  1  0  0  0 |  b = C2
  0  2  0  0  0  0 |  c = C3
  0  0  0  0  1  0 |  d = C4
  0  0  0  2  0  0 |  e = C5
  2  0  0  0  0  3 |  f = C6
```

In-depth Analysis of both the trees for Dataset PED:

LMT.

```
=== Summary ===

Correctly Classified Instances          25               96.1538 %
Incorrectly Classified Instances         1                3.8462 %
Kappa statistic                          0.8354
Mean absolute error                      0.0729
Root mean squared error                  0.1545
Relative absolute error                 26.7186 %
Root relative squared error             42.7578 %
Total Number of Instances               26

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.250    0.957      1.000    0.978      0.847    1.000     1.000     C1
                 0.750    0.000    1.000      0.750    0.857      0.847    1.000     1.000     notC1
Weighted Avg.    0.962    0.212    0.963      0.962    0.959      0.847    1.000     1.000

=== Confusion Matrix ===

  a  b   <-- classified as
 22  0 |  a = C1
  1  3 |  b = notC1
```

J48.

```
Correctly Classified Instances          81               92.0455 %
Incorrectly Classified Instances         7                7.9545 %
Kappa statistic                          0.7308
Mean absolute error                      0.0216
Root mean squared error                  0.1316
Relative absolute error                 21.5468 %
Root relative squared error             62.3069 %
Total Number of Instances               88

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.067    0.986      1.000    0.993      0.960    0.967     0.986     C1
                 0.750    0.024    0.600      0.750    0.667      0.653    0.868     0.574     C2
                 0.000    0.000    ?          0.000    ?          ?        0.750     0.511     C3
                 0.000    0.023    0.000      0.000    0.000      -0.016   0.489     0.011     C4
                 0.000    0.012    0.000      0.000    0.000      -0.016   0.988     0.667     C5
                 1.000    0.012    0.833      1.000    0.909      0.907    0.992     0.783     C6
                 0.000    0.000    ?          0.000    ?          ?        0.483     0.011     C7
Weighted Avg.    0.920    0.058    ?          0.920    ?          ?        0.948     0.916

=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 73  0  0  0  0  0  0 |  a = C1
  1  3  0  0  0  0  0 |  b = C2
  0  2  0  0  0  0  0 |  c = C3
  0  0  0  0  1  0  0 |  d = C4
  0  0  0  2  0  0  0 |  e = C5
  0  0  0  0  0  5  0 |  f = C6
  0  0  0  0  0  1  0 |  g = C7
```

2. *Neural Network Classifier Using PD and PED*

Network 1: MultiLayer Perceptron
Network Details:
Batch Size: 100
Hidden Layers: 100
learningRate: 0.4
Decay: True
Seed: 10
Epoch: 500
Cross Validation Folds: 10

Network 2: MultiClassClassifier
Network Details:
Batch Size: 10
Underlying Classifier: MultiLayer Perceptron
logLossDecoding: True
Method: Random correction code
Seed: 10

Activation function: Sigmoid

In depth Analysis of the MultiLayer Perceptron on PD:

```
Correctly Classified Instances          82               93.1818 %
Incorrectly Classified Instances         6                6.8182 %
Kappa statistic                          0.7459
Mean absolute error                      0.0243
Root mean squared error                  0.1269
Relative absolute error                 22.3938 %
Root relative squared error             57.5336 %
Total Number of Instances               88

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.143    0.974      1.000    0.987      0.914    0.997     0.999     C1
                 0.750    0.012    0.750      0.750    0.750      0.738    0.982     0.850     C2
                 0.500    0.000    1.000      0.500    0.667      0.703    1.000     1.000     C3
                 0.000    0.023    0.000      0.000    0.000      -0.016   0.345     0.017     C4
                 0.000    0.012    0.000      0.000    0.000      -0.016   0.895     0.141     C5
                 0.800    0.000    1.000      0.800    0.889      0.889    1.000     1.000     C6
Weighted Avg.    0.932    0.121    0.932      0.932    0.929      0.868    0.987     0.962

=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
 74  0  0  0  0  0 |  a = C1
  1  3  0  0  0  0 |  b = C2
  1  0  1  0  0  0 |  c = C3
  0  0  0  0  1  0 |  d = C4
  0  0  0  2  0  0 |  e = C5
  0  1  0  0  0  4 |  f = C6
```

In depth Analysis of the MultiClass Classifier on PD:

```
Correctly Classified Instances          84               95.4545 %
Incorrectly Classified Instances         4                4.5455 %
Kappa statistic                          0.8362
Mean absolute error                      0.0159
Root mean squared error                  0.1207
Relative absolute error                 14.5992 %
Root relative squared error             54.687  %
Total Number of Instances               88

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.071    0.987      1.000    0.993      0.957    0.992     0.998     C1
                 0.750    0.000    1.000      0.750    0.857      0.861    0.958     0.806     C2
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     C3
                 0.000    0.023    0.000      0.000    0.000      -0.016   0.908     0.111     C4
                 0.000    0.012    0.000      0.000    0.000      -0.016   0.733     0.069     C5
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     C6
Weighted Avg.    0.955    0.061    0.955      0.955    0.954      0.923    0.984     0.959

=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
 74  0  0  0  0  0 |  a = C1
  1  3  0  0  0  0 |  b = C2
  0  0  2  0  0  0 |  c = C3
  0  0  0  0  1  0 |  d = C4
  0  0  0  2  0  0 |  e = C5
  0  0  0  0  0  5 |  f = C6
```

In depth Analysis of the MultiLayer Perceptron on PED:

Correctly Classified Instances	83	94.3182 %
Incorrectly Classified Instances	5	5.6818 %
Kappa statistic	0.7878	
Mean absolute error	0.0189	
Root mean squared error	0.1074	
Relative absolute error	17.3803 %	
Root relative squared error	48.6513 %	
Total Number of Instances	88	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	1.000	0.143	0.974	1.000	0.987	0.914	0.993	0.999	C1
	0.750	0.024	0.600	0.750	0.667	0.653	0.976	0.833	C2
	0.000	0.000	?	0.000	?	?	1.000	1.000	C3
	0.000	0.000	?	0.000	?	?	0.632	0.030	C4
	1.000	0.012	0.667	1.000	0.800	0.812	1.000	1.000	C5
	0.800	0.000	1.000	0.800	0.889	0.889	1.000	1.000	C6
Weighted Avg.	0.943	0.121	?	0.943	?	?	0.989	0.980	

=== Confusion Matrix ===

a	b	c	d	e	f	<-- classified as
74	0	0	0	0	0	a = C1
1	3	0	0	0	0	b = C2
1	1	0	0	0	0	c = C3
0	0	0	0	1	0	d = C4
0	0	0	0	2	0	e = C5
0	1	0	0	0	4	f = C6

In depth Analysis of the MultiClass Classifier on PED:

```
Correctly Classified Instances          84               95.4545 %
Incorrectly Classified Instances         4                4.5455 %
Kappa statistic                          0.83
Mean absolute error                      0.0158
Root mean squared error                  0.1236
Relative absolute error                 14.6238 %
Root relative squared error             56.015  %
Total Number of Instances               88

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.143    0.974      1.000    0.987      0.914    0.965     0.993     C1
                 0.750    0.024    0.600      0.750    0.667      0.653    0.872     0.709     C2
                 0.000    0.000    ?          0.000    ?          ?        0.983     0.700     C3
                 0.000    0.000    ?          0.000    ?          ?        0.920     0.125     C4
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     C5
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     C6
Weighted Avg.    0.955    0.121    ?          0.955    ?          ?        0.964     0.964

=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
 74  0  0  0  0  0 |  a = C1
  1  3  0  0  0  0 |  b = C2
  1  1  0  0  0  0 |  c = C3
  0  1  0  0  0  0 |  d = C4
  0  0  0  0  2  0 |  e = C5
  0  0  0  0  0  5 |  f = C6
```

Summary and Analysis of Experiment 1:

Here is the summary of our results for Experiment 1.

Dataset/Network Accuracy	LMT	J48	NN1	NN2
PD	94.32%	90.9%	93.18%	95.45%
PED	96.15%	92.04%	93.32%	95.45%

Initially, we approached the problem by working on all the attributes provided in the dataset. But we quickly realized that there was a big gap between the performance of PD and PED dataset.

Thus, we decided to modify the PD dataset by selecting a better set of attributes using methods discussed in the Data Preprocessing subpart and found our results to be much closer to the dataset with expert provided attributes.

In our experiment, we found that LMT does a much better job than J48. LMT uses a classification model with logistic regression functions at the leaves. We found that the rules of LMT were better suited.
The reason for such a conclusion is that after checking the models over various datasplititng methods, we noticed that both the trees do a very good job at classifying the class “C1”, with no misclassifications noticed. However, J48 made more errors when it comes to underrepresented classes. This is where LMT outperformed J48.

In the case of our Neural Networks, we noticed that the results were surprisingly very similar even though the datasets in question here had quite different attributes. This shows that although the expert data is still better, using machine learning techniques to define a better subset of attributes for classification can provide for an accurate result in the absence of an expert.

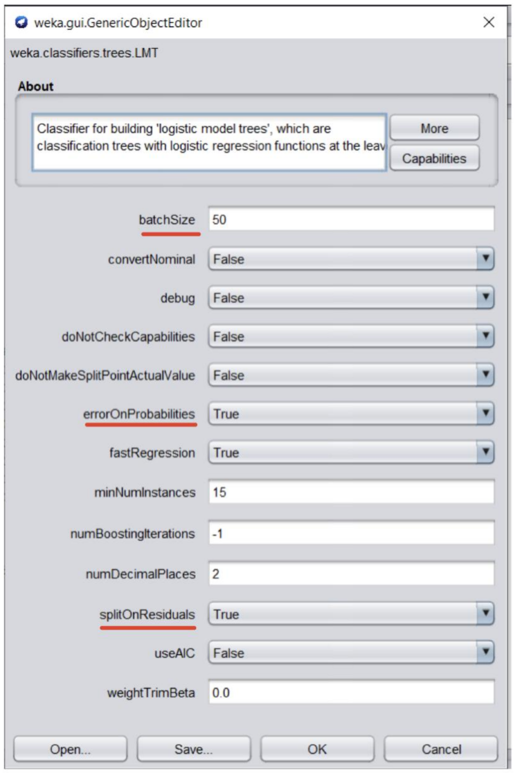
Experiment 2

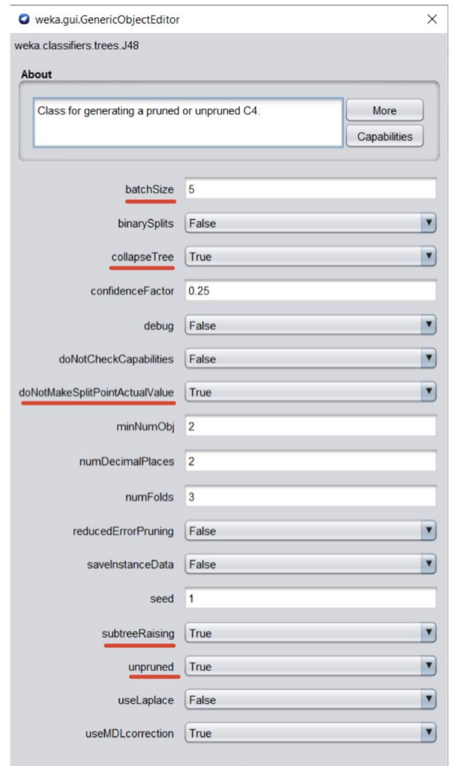
3. Decision Tree Contrast Classifier Using PD and PED

- The two trees used as Descriptive Classifiers are LMT and J48.

  Tree 1: LMT
  Training method: 70-30 dataset split for training and testing.
  Tree 2: J48
  Training Method: Cross-Validation with 25 Folds.

Tree Configurations:





The red markers indicate the changes made from default configurations.

In-depth Analysis of both the trees for Dataset PD:

LMT

```
Correctly Classified Instances          26              100      %
Incorrectly Classified Instances         0                0      %
Kappa statistic                          1
Mean absolute error                      0.042
Root mean squared error                  0.1076
Relative absolute error                 15.4039 %
Root relative squared error             29.7804 %
Total Number of Instances               26

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     C1
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     notC1
Weighted Avg.    1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000

=== Confusion Matrix ===

  a  b   <-- classified as
 22  0 |  a = C1
  0  4 |  b = notC1
```

J48

```
Correctly Classified Instances          85               96.5909 %
Incorrectly Classified Instances         3                3.4091 %
Kappa statistic                          0.8688
Mean absolute error                      0.0376
Root mean squared error                  0.1849
Relative absolute error                 13.7289 %
Root relative squared error             50.3119 %
Total Number of Instances               88

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.986    0.143    0.973      0.986    0.980      0.870    0.986     0.996     C1
                 0.857    0.014    0.923      0.857    0.889      0.870    0.986     0.881     notC1
Weighted Avg.    0.966    0.122    0.965      0.966    0.965      0.870    0.986     0.978

=== Confusion Matrix ===

  a  b   <-- classified as
 73  1 |  a = C1
  2 12 |  b = notC1
```

In-depth Analysis of both the trees for Dataset PED:

LMT.

```
=== Summary ===

Correctly Classified Instances          25               96.1538 %
Incorrectly Classified Instances         1                3.8462 %
Kappa statistic                          0.8354
Mean absolute error                      0.0729
Root mean squared error                  0.1545
Relative absolute error                 26.7186 %
Root relative squared error             42.7578 %
Total Number of Instances               26

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.250    0.957      1.000    0.978      0.847    1.000     1.000     C1
                 0.750    0.000    1.000      0.750    0.857      0.847    1.000     1.000     notC1
Weighted Avg.    0.962    0.212    0.963      0.962    0.959      0.847    1.000     1.000

=== Confusion Matrix ===

  a  b   <-- classified as
 22  0 |  a = C1
  1  3 |  b = notC1
```

J48.

```
Correctly Classified Instances          84               95.4545 %
Incorrectly Classified Instances         4                4.5455 %
Kappa statistic                          0.8079
Mean absolute error                      0.0512
Root mean squared error                  0.2145
Relative absolute error                 18.7116 %
Root relative squared error             58.3655 %
Total Number of Instances               88

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.286    0.949      1.000    0.974      0.823    0.914     0.969     C1
                 0.714    0.000    1.000      0.714    0.833      0.823    0.914     0.827     notC1
Weighted Avg.    0.955    0.240    0.957      0.955    0.951      0.823    0.914     0.946

=== Confusion Matrix ===

  a  b   <-- classified as
 74  0 |  a = C1
  4 10 |  b = notC1
```

4. Neural Network Contrast Classifier Using PD and PED

Network 1: MultiLayer Perceptron
Network Details:
Batch Size: 100
Hidden Layers: 100
learningRate: 0.4
Decay: True
Seed: 10
Epoch: 500
Cross Validation Folds: 10

Network 2: MultiClassClassifier
Network Details:
Batch Size: 10
Underlying Classifier: MultiLayer Perceptron
logLossDecoding: True
Method: Random correction code
Seed: 10

In depth Analysis of the MultiLayer Perceptron for contrast classification on PD:

```
Correctly Classified Instances          87               98.8636 %
Incorrectly Classified Instances         1                1.1364 %
Kappa statistic                          0.9563
Mean absolute error                      0.0191
Root mean squared error                  0.1086
Relative absolute error                  6.982  %
Root relative squared error             29.5611 %
Total Number of Instances               88

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.071    0.987      1.000    0.993      0.957    1.000     1.000     C1
                 0.929    0.000    1.000      0.929    0.963      0.957    1.000     1.000     notC1
Weighted Avg.    0.989    0.060    0.989      0.989    0.988      0.957    1.000     1.000

=== Confusion Matrix ===

  a  b   <-- classified as
 74  0 |  a = C1
  1 13 |  b = notC1
```

In depth Analysis of the MultiClass Classifier for contrast classification on PD:

```
Correctly Classified Instances          83               94.3182 %
Incorrectly Classified Instances         5                5.6818 %
Kappa statistic                          0.7517
Mean absolute error                      0.063
Root mean squared error                  0.2317
Relative absolute error                 23.0104 %
Root relative squared error             63.1918 %
Total Number of Instances               88

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.357    0.937      1.000    0.967      0.776    0.784     0.874     C1
                 0.643    0.000    1.000      0.643    0.783      0.776    0.784     0.805     notC1
Weighted Avg.    0.943    0.300    0.947      0.943    0.938      0.776    0.784     0.863

=== Confusion Matrix ===

  a  b   <-- classified as
 74  0 |  a = C1
  5  9 |  b = notC1
```

In depth Analysis of the MultiLayer Perceptron for contrast classification on PED:

```
Correctly Classified Instances          86               97.7273 %
Incorrectly Classified Instances         2                2.2727 %
Kappa statistic                          0.9098
Mean absolute error                      0.0357
Root mean squared error                  0.1586
Relative absolute error                 13.0561 %
Root relative squared error             43.1547 %
Total Number of Instances               88

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.143    0.974      1.000    0.987      0.914    0.994     0.999     C1
                 0.857    0.000    1.000      0.857    0.923      0.914    0.994     0.974     notC1
Weighted Avg.    0.977    0.120    0.978      0.977    0.977      0.914    0.994     0.995

=== Confusion Matrix ===

  a  b   <-- classified as
 74  0 |  a = C1
  2 12 |  b = notC1
```

In depth Analysis of the MultiClass Classifier for contrast classification on PED:

```
Correctly Classified Instances          85               96.5909 %
Incorrectly Classified Instances         3                3.4091 %
Kappa statistic                          0.8605
Mean absolute error                      0.0427
Root mean squared error                  0.1792
Relative absolute error                 15.5024 %
Root relative squared error             48.9133 %
Total Number of Instances               88

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.214    0.961      1.000    0.980      0.869    0.926     0.953     C1
                 0.786    0.000    1.000      0.786    0.880      0.869    0.926     0.920     notC1
Weighted Avg.    0.966    0.180    0.967      0.966    0.964      0.869    0.926     0.948

=== Confusion Matrix ===

  a  b   <-- classified as
 74  0 |  a = C1
  3 11 |  b = notC1
```

Summary and Analysis of Experiment 2:

Here is the summary of our results for Experiment 2.

Dataset/Network Accuracy	LMT	J48	NN1	NN2
Contrast PD	100%	96.6%	98.86%	94.31%
Contrast PED	96.17%	95.45%	97.72%	96.59%

To our surprise, LMT provided a perfect accuracy score for the contrast PD classification. As we noticed in the previous segment, LMT was working perfectly with class C1 but had a few errors when it came to under-represented classes. Now that those classes have been grouped together, it shows perfect results.
This is what the tree looks like:

```
LM_1:
Class C1 :
1.25 +
[MgO] * 0.11 +
[S] * -0 +
[Zn] * -0 +
[ Rb ] * -0.03

Class notC1 :
-1.25 +
[MgO] * -0.11 +
[S] * 0     +
[Zn] * 0     +
[ Rb ] * 0.03
```

Now, J48, although a few errors in class “notC1” were expected, it makes an error for class “C1” as well.

Again, for Neural Networks, we notice that the MultiClass Classifier performs worse. This was a surprise as it is built on MultiLayer Perceptron with additional functionalities like “1 against 1” or “1 against all” functionalities. However, for multilayer perceptron, we again notice a close accuracy between the PD and PED dataset indicating the effectiveness of the chosen attributes for PD.

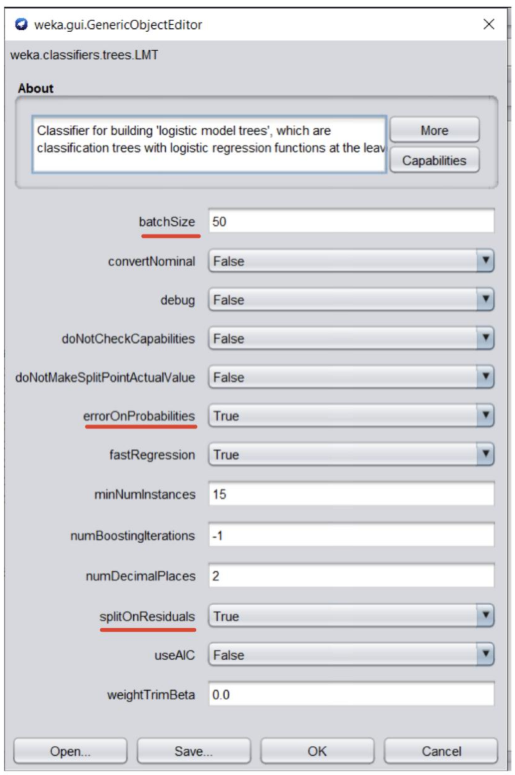
Experiment 3

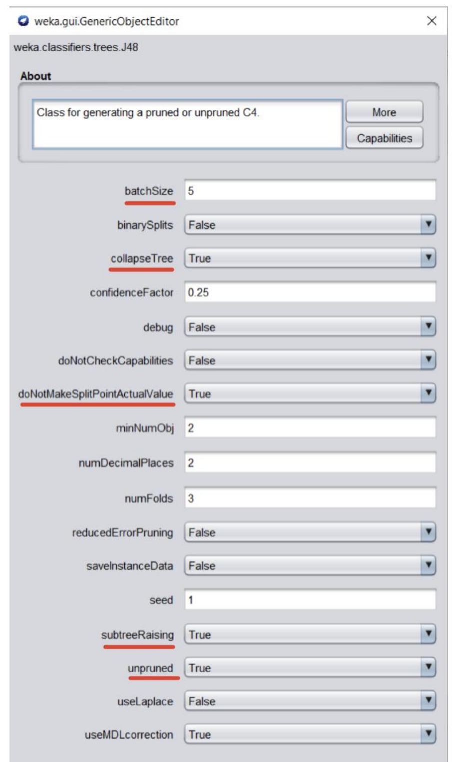
5. Decision Tree Classifier Using PD1 and PED1:

- The two trees used as Descriptive Classifiers are LMT and J48.

  Tree 1: LMT
  Training method: Cross-Validation with 25 folds.
  Tree 2: J48
  Training Method: 70-30 dataset split for training and testing.

Tree Configurations:





The red markers indicate the changes made from default configurations.

In-depth Analysis of both the trees for Dataset PD1:

M1 Classifier: J48 with Cross-Validation.

```
=== Summary ===

Correctly Classified Instances          85               96.5909 %
Incorrectly Classified Instances         3                3.4091 %
Kappa statistic                          0.8688
Mean absolute error                      0.0553
Root mean squared error                  0.1845
Relative absolute error                 20.1976 %
Root relative squared error             50.2063 %
Total Number of Instances               88

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.986    0.143    0.973      0.986    0.980      0.870    0.857     0.945     C1
                 0.857    0.014    0.923      0.857    0.889      0.870    0.857     0.814     notC1
Weighted Avg.    0.966    0.122    0.965      0.966    0.965      0.870    0.857     0.924

=== Confusion Matrix ===

  a  b   <-- classified as
 73  1 |  a = C1
  2 12 |  b = notC1
```

M2 Classifier: J48 with Binary Splits.

```
=== Summary ===

Correctly Classified Instances          85               96.5909 %
Incorrectly Classified Instances         3                3.4091 %
Kappa statistic                          0.8688
Mean absolute error                      0.0624
Root mean squared error                  0.19
Relative absolute error                 22.811  %
Root relative squared error             51.7134 %
Total Number of Instances               88

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.986    0.143    0.973      0.986    0.980      0.870    0.858     0.941     C1
                 0.857    0.014    0.923      0.857    0.889      0.870    0.858     0.799     notC1
Weighted Avg.    0.966    0.122    0.965      0.966    0.965      0.870    0.858     0.919

=== Confusion Matrix ===

  a  b   <-- classified as
 73  1 |  a = C1
  2 12 |  b = notC1
```

Discriminant Rules of both the trees for Dataset PD1:

M1 Classifier: J48 with Cross-Validation.

```
Al2o3 = '(-inf-4.52]'
|   Zn = '(-inf-826.333333]'
|   |   S = '(-inf-12955.666667]': C1 (76.0/2.0)
|   |   S = '(12955.666667-25873.333333]': notC1 (2.0)
|   |   S = '(25873.333333-inf)': notC1 (2.0)
|   Zn = '(826.333333-inf)': notC1 (3.0)
Al2o3 = '(4.52-inf)': notC1 (5.0)
```

An easier way to interpret this is:
IF AL2O3 = (-inf, 4.52] and Zn = (-inf, 826.3):
    IF S = (-inf, 12955.67] : Class C1
    IF S = (12955.67, inf) : Class notC1
FOR ALL ELSE Class notC1

Note that the rules of the tree should be represented as in the image, the second interpretation is for ease of understanding.

M2 Classifier: J48 with Binary Splits.

```
S = '(-inf-12955.666667]'
|   Al2o3 = '(-inf-4.52]'
|   |   Zn = '(-inf-826.333333]': C1 (76.0/2.0)
|   |   Zn != '(-inf-826.333333]': notC1 (3.0)
|   Al2o3 != '(-inf-4.52]': notC1 (3.0)
S != '(-inf-12955.666667]': notC1 (6.0)
```

An easier way to interpret this is:

IF S = (-inf, 12955.67]:
    IF Al2O3 = (-inf, 4.52] :
        IF Zn = (-inf, 826.34]: Class C1
FOR ALL ELSE Class notC1

Note that the rules of the tree should be represented as in the image, the second interpretation is for ease of understanding.

In-depth Analysis of both the trees for Dataset PED1:

M1 Classifier: J48 with Cross-Validation.

```
Correctly Classified Instances          84               95.4545 %
Incorrectly Classified Instances         4                4.5455 %
Kappa statistic                          0.8197
Mean absolute error                      0.0565
Root mean squared error                  0.1998
Relative absolute error                 20.6378 %
Root relative squared error             54.3802 %
Total Number of Instances               88

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.986    0.214    0.961      0.986    0.973      0.823    0.854     0.943     C1
                 0.786    0.014    0.917      0.786    0.846      0.823    0.854     0.804     notC1
Weighted Avg.    0.955    0.182    0.954      0.955    0.953      0.823    0.854     0.921

=== Confusion Matrix ===

  a  b   <-- classified as
 73  1 |  a = C1
  3 11 |  b = notC1
```

M2 Classifier: J48 with Binary Splits.

```
=== Summary ===

Correctly Classified Instances          85               96.5909 %
Incorrectly Classified Instances         3                3.4091 %
Kappa statistic                          0.8688
Mean absolute error                      0.0567
Root mean squared error                  0.1902
Relative absolute error                 20.7056 %
Root relative squared error             51.7713 %
Total Number of Instances               88

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.986    0.143    0.973      0.986    0.980      0.870    0.857     0.945     C1
                 0.857    0.014    0.923      0.857    0.889      0.870    0.857     0.799     notC1
Weighted Avg.    0.966    0.122    0.965      0.966    0.965      0.870    0.857     0.921

=== Confusion Matrix ===

  a  b   <-- classified as
 73  1 |  a = C1
  2 12 |  b = notC1
```

Discriminant Rules of both the trees for Dataset PED1:

M1 Classifier: J48 with Cross-Validation.

```
S = '(-inf-12955.666667]'
|   CaO+MgO = '(-inf-19.836667]': notC1 (3.0)
|   CaO+MgO = '(19.836667-39.193333]': C1 (3.0/1.0)
|   CaO+MgO = '(39.193333-inf)'
|   |   Zn = '(-inf-826.333333]': C1 (73.0/1.0)
|   |   Zn = '(826.333333-inf)': notC1 (3.0)
S = '(12955.666667-25873.333333]': notC1 (3.0)
S = '(25873.333333-inf)': notC1 (3.0)
```

An easier way to interpret this is:
IF S = (-inf, 12955.67]:
        IF CaO+MgO = (19.837, 39.193] : Class C1
        IF CaO+MgO = (39.193, inf) :
                IF Zn = (-inf, 826.33): Class C1
FOR ALL ELSE Class notC1

Note that the rules of the tree should be represented as in the image, the second interpretation is for ease of understanding.

M2 Classifier: J48 with Binary Splits.

```
S = '(-inf-12955.666667]'
|   MgO = '(-inf-8.453333]': notC1 (5.0/1.0)
|   MgO != '(-inf-8.453333]'
|   |   Zn = '(-inf-826.333333]': C1 (74.0/1.0)
|   |   Zn != '(-inf-826.333333]': notC1 (3.0)
S != '(-inf-12955.666667]': notC1 (6.0)
```

An easier way to interpret this is:

IF S = (-inf, 12955.67]:
    IF MgO != (-inf, 8.46] :
        IF Zn = (-inf, 826.34]: Class C1
FOR ALL ELSE Class notC1

Note that the rules of the tree should be represented as in the image, the second interpretation is for ease of understanding.

Summary and Analysis of Experiment 3:

Here is the summary of our results for Experiment 3.

Dataset/Network Accuracy	M1	M2
PD1	96.6%	96.4%
PED1	95.45%	96.6%

The interesting conclusion of this was to see the use of different attributes be used as a result of using binary splits or not. The broader conclusion based on discriminant rules is that since the notC1 classes, despite being pooled together, are still underrepresented, the tree makes error in differentiating between them.
It is also interesting to note the changes in the attributes still lead to a misclassification in the same class subgroup.

TOOL 2: ORANGE

Data Preprocessing

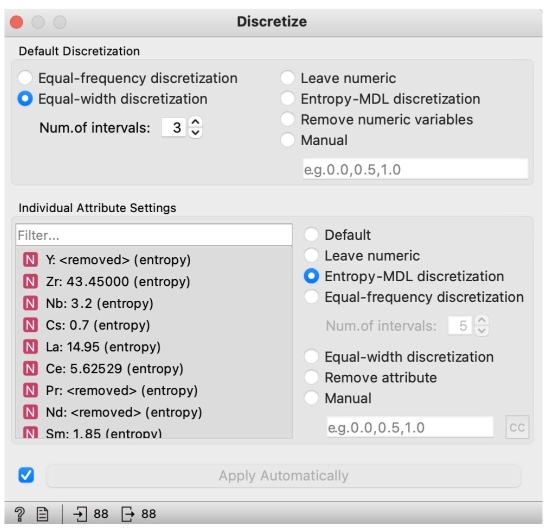
Descriptive Classifier

For Dataset: PD1 and PED1

1. **PD1:** Data Discretization

   Method 1: Equal-width discretization
   Evenly splits the range between the smallest and the largest observed value.

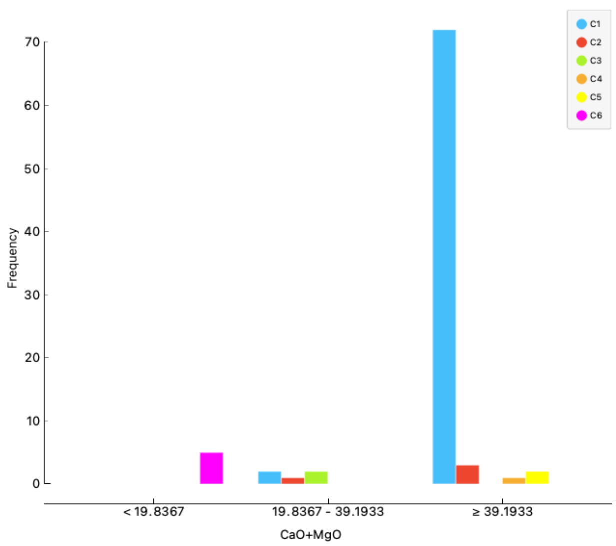
   Method 2: Entropy-MDL discretization (top-down discretization)
   Recursively splits the attribute at a cut maximizing information gain, until the gain is lower than the minimal description length of the cut.

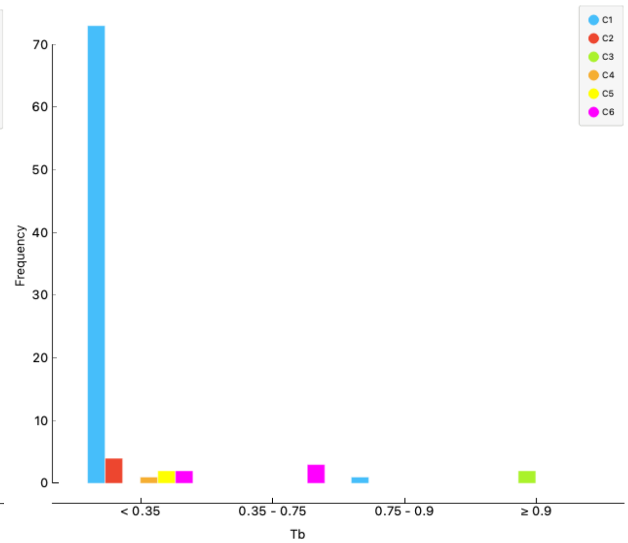


We used the above two methods to create a set PD1 with no more than 4 bins. For the first half of these total 46 attributes(from CaO+MgO to Li), we applied method 1, the Equal-width discretization, and set the intervals as 3. For the rest half of these attributes, we applied method 2, the Entropy-MDL discretization.

In the end, we got our new set PD1, and each attribute has at most 3 bins. Following are 2 examples from method1 and 2.

Method 1: Method 2:





2. **PED1:** By using the "Select Columns" tool in ORANGE, we simply picked out the most important 8 attributes and created the new subset PED1.

Non - Descriptive Classifier

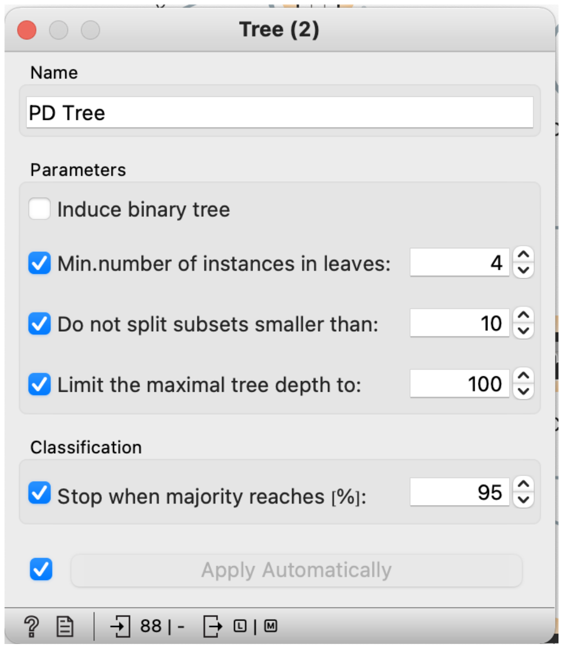
For Dataset: PD and PED

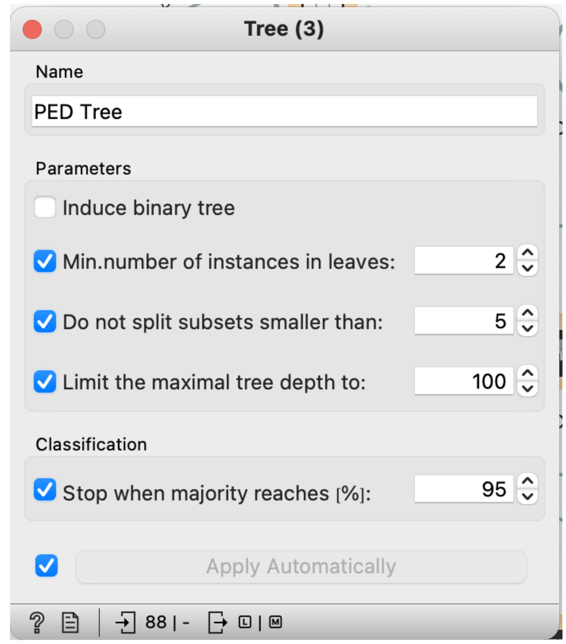
ORANGE uses the default method of normalization, it normalizes the data by centering to mean and scaling to standard deviation of 1.
Therefore, we can simply create datasets PD and PED.

Experiment 1

1. Decision Tree Classifier Using PD and PED

- Tool: ORANGE
- Testing Method: 10-folds Cross-Validation
- Parameters setting:

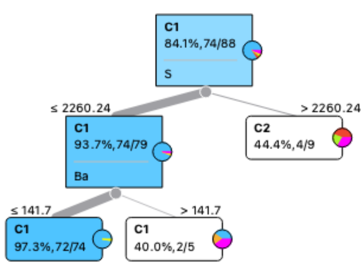




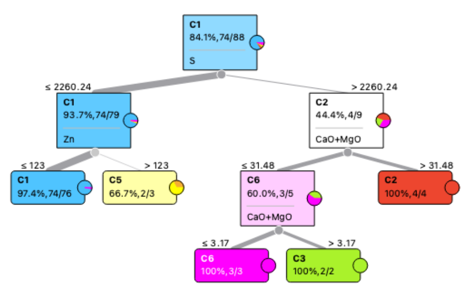
Simply changed the minimum number of instances in leaves and the split point, to see how these factors will impact the accuracy of classifiers.

- Tree Viewer:

PD Tree



PED Tree



- Accuracy Compare:

Evaluation Results

Model	AUC	CA	F1	Precision	Recall
PED Tree	0.727	0.852	0.823	0.809	0.852
PD Tree	0.770	0.841	0.814	0.820	0.841

AUC: Area under ROC curve
CA: Classification Accuracy

From the table we can see that PED Tree with lower number of instances in leaves and split point has better accuracy than PD Tree.

- Confusion Matrix:

PD Tree

Actual \ Predicted	C1	C2	C3	C4	C5	C6	Σ
C1	97.3 %	2.7 %	0.0 %	0.0 %	0.0 %	0.0 %	74
C2	75.0 %	25.0 %	0.0 %	0.0 %	0.0 %	0.0 %	4
C3	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	2
C4	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	1
C5	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	2
C6	40.0 %	40.0 %	0.0 %	0.0 %	0.0 %	20.0 %	5
Σ	80	7	0	0	0	1	88

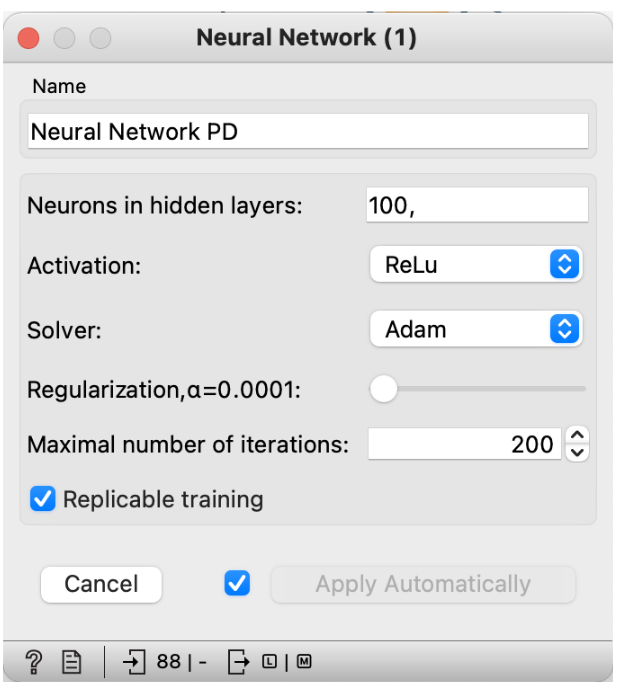
PED Tree

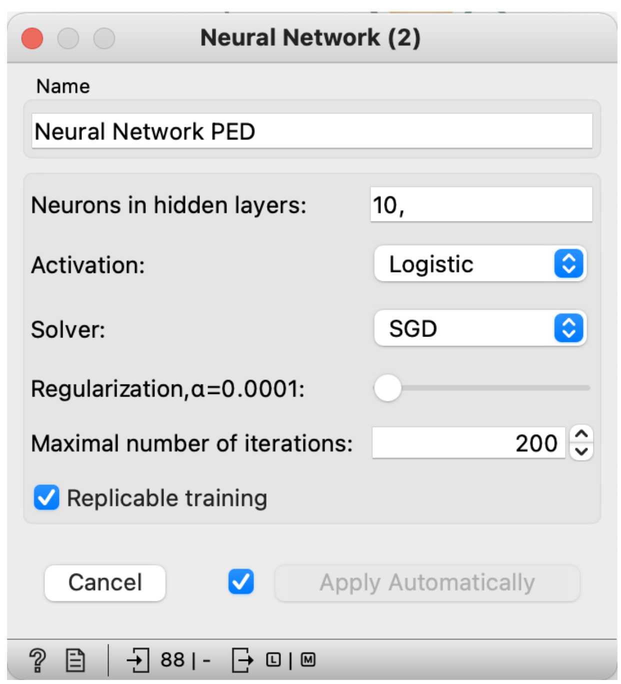
Actual \ Predicted	C1	C2	C3	C4	C5	C6	Σ
C1	98.6 %	0.0 %	0.0 %	0.0 %	1.4 %	0.0 %	74
C2	75.0 %	25.0 %	0.0 %	0.0 %	0.0 %	0.0 %	4
C3	0.0 %	50.0 %	0.0 %	0.0 %	0.0 %	50.0 %	2
C4	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	1
C5	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	2
C6	40.0 %	0.0 %	40.0 %	0.0 %	0.0 %	20.0 %	5
Σ	81	2	2	0	1	2	88

2. *Neural Network Classifier Using PD and PED*

- Network Topology 1:
  Neurons in hidden layers :100
  Activation: ReLu
  Solver: Adam
  (default)

- Network Topology 2:
  Neurons in hidden layers :10,
  Activation: Logistic
  Solver: SGD





- Testing Method: 10-folds Cross-Validation
  Random Sampling(repeat train: 10; training set size: 90%)

- Accuracy Compare:
  10-folds Cross-Validation:

Evaluation Results

Model	AUC	CA	F1	Precision	Recall
Neural Network PED	0.755	0.841	0.768	0.707	0.841
Neural Network PD	0.835	0.909	0.889	0.880	0.909

Random Sampling(repeat train: 10; training set size: 90%):

Evaluation Results

Model	AUC	CA	F1	Precision	Recall
Neural Network PED	0.606	0.844	0.773	0.713	0.844
Neural Network PD	0.887	0.911	0.892	0.883	0.911

- Confusion Matrix:

PD

Predicted

Actual	C1	C2	C3	C4	C5	C6	Σ
C1	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	76
C2	66.7 %	33.3 %	0.0 %	0.0 %	0.0 %	0.0 %	6
C3	0.0 %	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	1
C4	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	1
C5	0.0 %	0.0 %	0.0 %	0.0 %	100.0 %	0.0 %	1
C6	20.0 %	20.0 %	0.0 %	0.0 %	0.0 %	60.0 %	5
Σ	82	4	0	0	1	3	90

PED

Predicted

Actual	C1	C2	C3	C4	C5	C6	Σ
C1	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	76
C2	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	6
C3	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	1
C4	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	1
C5	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	1
C6	100.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	5
Σ	90	0	0	0	0	0	90

3. *Compare Descriptive with Non-Descriptive*

- Accuracy Compare:

Evaluation Results

Model	AUC	CA	F1	Precision	Recall
Neural Network PD	0.835	0.909	0.889	0.880	0.909
PED Tree	0.727	0.852	0.823	0.809	0.852
PD Tree	0.770	0.841	0.814	0.820	0.841
Neural Network PED	0.755	0.841	0.768	0.707	0.841

Experiment 2

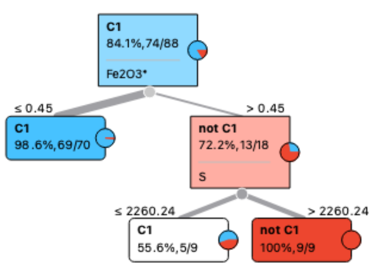
In order to perform the contrast classification for class **C1** with a class **notC1** that contains other classes, we first changed all the names of classes C2 to C6 into a new class named **notC1**, and this made things much easier for our next move.

Thanks to the design of Orange, we only need to change the dataset in the file widget and don't have to do anything else. All the remaining steps were automatically done in Orange since they were already done in the previous Experiment 1. Following are the results:

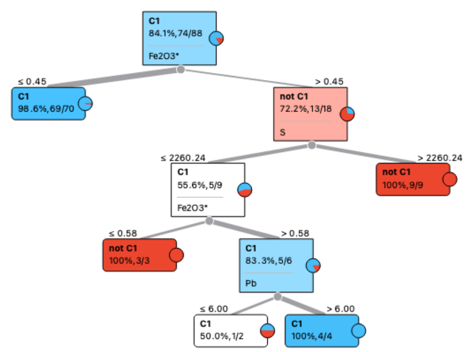
1. Decision Tree Classifier Using PD and PED

- Testing Method: 10-folds Cross-Validation
- Parameters setting: Same as Experiment 1
- Tree Viewer:

PD



PED



- Accuracy Compare:

Evaluation Results

Model	AUC	CA	F1	Precision	Recall
PD Tree	0.875	0.864	0.854	0.850	0.864
PED Tree	0.749	0.875	0.873	0.872	0.875

- Confusion Matrix:

PD

PED

		Predicted		
		C1	not C1	Σ
Actual	C1	94.6 %	5.4 %	74
	not C1	57.1 %	42.9 %	14
	Σ	78	10	88

		Predicted		
		C1	not C1	Σ
Actual	C1	93.2 %	6.8 %	74
	not C1	42.9 %	57.1 %	14
	Σ	75	13	88

2. *Neural Network Classifier Using PD and PED*

- Network Topology: Same as Experiment 1
- Testing Method: Same as Experiment 1
- Accuracy Compare:

10-folds Cross-Validation:

Evaluation Results

Model	AUC	CA	F1	Precision	Recall
Neural Network PD	0.851	0.943	0.938	0.947	0.943
Neural Network PED	0.776	0.841	0.768	0.707	0.841

Random Sampling(repeat train: 10; training set size: 90%):

Evaluation Results

Model	AUC	CA	F1	Precision	Recall
Neural Network PD	0.923	0.922	0.915	0.920	0.922
Neural Network PED	0.849	0.844	0.773	0.713	0.844

- Confusion Matrix:

PD

		Predicted		
		C1	not C1	Σ
Actual	C1	100.0 %	0.0 %	74
	not C1	35.7 %	64.3 %	14
	Σ	79	9	88

PED

		Predicted		
		C1	not C1	Σ
Actual	C1	100.0 %	0.0 %	74
	not C1	100.0 %	0.0 %	14
	Σ	88	0	88

3. Compare Descriptive with Non-Descriptive

- Accuracy Compare:

Evaluation Results

Model	AUC	CA	F1	Precision	Recall
Neural Network PD	0.851	0.943	0.938	0.947	0.943
PED Tree	0.749	0.875	0.873	0.872	0.875
PD Tree	0.875	0.864	0.854	0.850	0.864
Neural Network PED	0.776	0.841	0.768	0.707	0.841

Analysis

Of the data:

- Attribute selection plays an important role.
- Out of 49 attributes, only 12 attributes in total were deemed to be useful.
- PCA has a strong impact on performance but makes it difficult to interpret.
- The data is skewed i.e Class C1 has higher representation which leads to accurate prediction for that class but poor performance when it comes to other classes.
- Even after pooling the other classes as “notC1”, it's still underrepresented leading to more errors in the “notC1” class.

Of tools used:

1. The tools we used are WEKA and Orange. Both of them are good toolkits for learners to study data visualization and machine learning.

2. Compare the accuracy of WEKA and Orange in Experiment 1 and 2, we got the following results:

		WEKA		ORANGE	
		PD	PED	PD	PED
Ex1	Decistion Tree	94.30%	96.20%	84.10%	85.20%
		90.90%	92.10%	86.70%	84.40%
	Neural Network	93.20%	94.32%	90.90%	84.10%
		95.50%	95.50%	91.90%	84.40%
Ex2	Decistion Tree	100.00%	96.20%	86.40%	87.50%
		96.60%	95.50%	87.80%	87.80%
	Neural Network	98.90%	97.70%	94.30%	84.10%
		94.30%	96.60%	92.20%	84.40%

From the above table it is clear that WEKA tool estimates higher accuracy for both Decision Tree and Neural Network than Orange.
For Orange, the accuracy of Neural Network is better than Decision Tree when we use the PD dataset, and it shows the opposite situation when we use the PED dataset.
For WEKA, although different algorithms and topologies lead to some accuracy changes, the overall accuracy rate is quite high, basically reaching over 95%. This can be attributed to the multitude of in-built features that provide for optimization techniques at various points in training.

3. In general, the experience with Orange is good, because the interface of Orange is nicely designed and the analytics workflow is easy to create with the use of drag and drop of its widgets. But compared with WEKA, Orange has fewer classifiers and adjustable parameters, which makes it impossible to perform more in-depth and specific analysis.
   For instance, you cannot see how the neural network classifier looks like in Orange, but WEKA provides a clear picture to show you how it looks, therefore when you change the parameters you can notice the difference in the network. Also, the random forest widget does not allow the user to see which variables have the highest information gain.

4. As for preference, when dealing with complicated algorithms and huge datasets in real life, WEKA seems to be a better choice than Orange, since it has larger capabilities, more complicated algorithms and adjustable parameters which can meet users' requirements.