CSE 521 - DATA MINING
PROJECT REPORT

TEAM - 5

VENKATA RAVI TEJA TAKKELLA - 113219890

NAGIREDDY SUSMITHA REDDY - 113271915

KRISHNA SHASHANK GORANTA - 113221080

INTRODUCTION

The main goal of this project is to use two Internet based Classification tools to build two types of classifiers: **descriptive** and **non-descriptive** using different tools.

PROJECT TOOLS

- WEKA
- Orange
- RapidMiner

S1 : DATA PREPARATION

Cleaning The Data :

The initial dataset given is in .xls format, so we converted that to csv first. While converting, we also removed the blank rows and footers as they are not required for our process. These blank rows will raise errors when we load them into the classifiers.

After converting the file to csv format, we removed **Echantillon (not required for classification)** and one **Type de roche (Redundant column).** Without removing this redundant column, classifiers will think that it's another nominal attribute.

There is a field under the Li attribute which has a value mentioned as **'<0.3'.** To make the data consistent and numerical, we changed it to **0.29**.

We corrected the misspellings and the class names provided in the csv to the format mentioned in the project description. So the final set of class names look like:

C1 : R. Carbonatees AND R. Carbonatees impures
C2 : Pyrate
C3 : Charcopyrite
C4 : Galene
C5 : Spahlerite
C6 : Sediments terrigenes

Attribute Selection :

We observed that some attributes have more than **30%** of missing values. Replacing them with their means will cause a lot of biasness and so we removed them from our dataset. Out of them, we even found **Pb** (55%), but since it's mentioned that it's a critical value by experts; we have not discarded it. So the final list of attributes which are discarded are :

As (72%), **Cd** (70%), **Ni** (40%), **Sc** (50%), **Co** (85%), **Li** (39%), **Mo** (89%)

After the above changes, we end up with **41** attributes and **one** class field.

Creating PD and PED :

Once we cleaned the data and selected the attributes, we then created our datasets PD and PED (using the most important attributes determined by the expert).

Filling Missing Values :

After removing the attributes having more than 30% missing values, the rest of the missing values have been removed based on the classifier's tool. For Ex: for **WEKA** "**ReplaceMissingValues**" filter under unsupervised section. We used the mean of the attribute's values to replace all the missing values.

We didn't remove any outliers as most of the attribute values are the main reason behind their classification and any extreme value makes it easier for it to be classified.

These missing values have been replaced using the tools we selected (WEKA, RapidMiner and Orange). So this step is repeated across all the tools before we start building the classifiers.

Motivation :

The entire agenda for this preparation is to generate noiseless and consistent data, which is important for our classification process.

Result :

After this preparation step, we get cleaner data which is ready to be loaded into the classifiers for preprocessing.

So the process till here is the same for all the classifiers except replacing missing values and the rest (**S2** and **S3**) is dependent on the classifier selected.

WEKA

Introduction about the tool :

WEKA is an Open source containing a collection of visualization tools and algorithms for data analysis and predictive modeling, together with graphical user interfaces for easy access to these functions. Weka supports several standard data mining tasks, more specifically, data preprocessing, clustering, classification, regression, visualization, and feature selection.

For building descriptive and non descriptive classifiers, we have generated two cleaned data sets **PD** and **PED** in the form of csv files in the Data Preparation step for Weka. We have used these same csv files for the Orange tool.

Initially, we replaced the missing values using the **"ReplaceMissingValues"** under the Unsupervised filter. This method uses the modes and means of the attributes to replace the missing values of the respective attribute. This is part of Data Preparation but included here.

S2 : DATA PREPROCESSING

1) Descriptive Classifiers :

Discretization methods try to convert the numerical attributes to nominal. This is done by converting the data into intervals i.e bins. Since we are trying to generate a decision tree, converting the data to bins is much required as this decreases the number of branches at each node to a value which is understandable and able to be visualized. This also helps the classifier to find the trends and patterns among the attributes to generate the result faster and more accurately.

So we have used different techniques for different attributes for generating the datasets **PD1** and **PED1**.

PD1 :

We were asked to create not more than 4 bins for each attribute using different discretization techniques. The following table provides the discretization methods and corresponding attributes.

Discretization Method	Attributes
1 bin	P2O5, Sr, Y, Tb, Dy, Ho, Tm, U
2 bins Equi-Width useEqualFrequency : False	Fe2O3*, MgO, S, Cu, Al2O3, TiO2, MnO, MgO, Na2O, K2O, Cu, Cr, V, Ba, Rb, Zr, Nb, Cs, La , Ce, Pr , Nd, Sm, Eu , Gd, Er, Yb, Lu, Hf, Ta, Th
2 bins Equi-Depth useEqualFrequency : True	Zn
3 bins	CaO+MgO , CaO

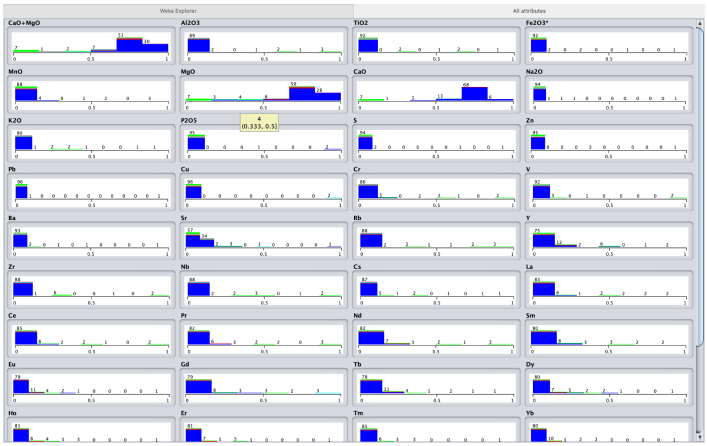
PED1 :

We have just secluded the important attributes from the PD1 dataset as specified by the expert. And PED1 dataset is generated.

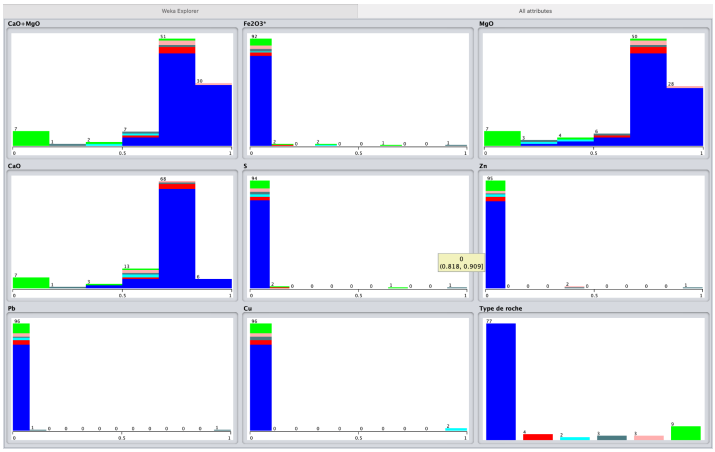
2) Non Descriptive Classifiers :

We were asked to normalize the data for classification. For this, we have selected the "**Normalize**" option under the unsupervised filter. This uses the **"Min-Max Normalization" (0-1 Normalization)** technique.

PD dataset visualised :



PED dataset visualised :



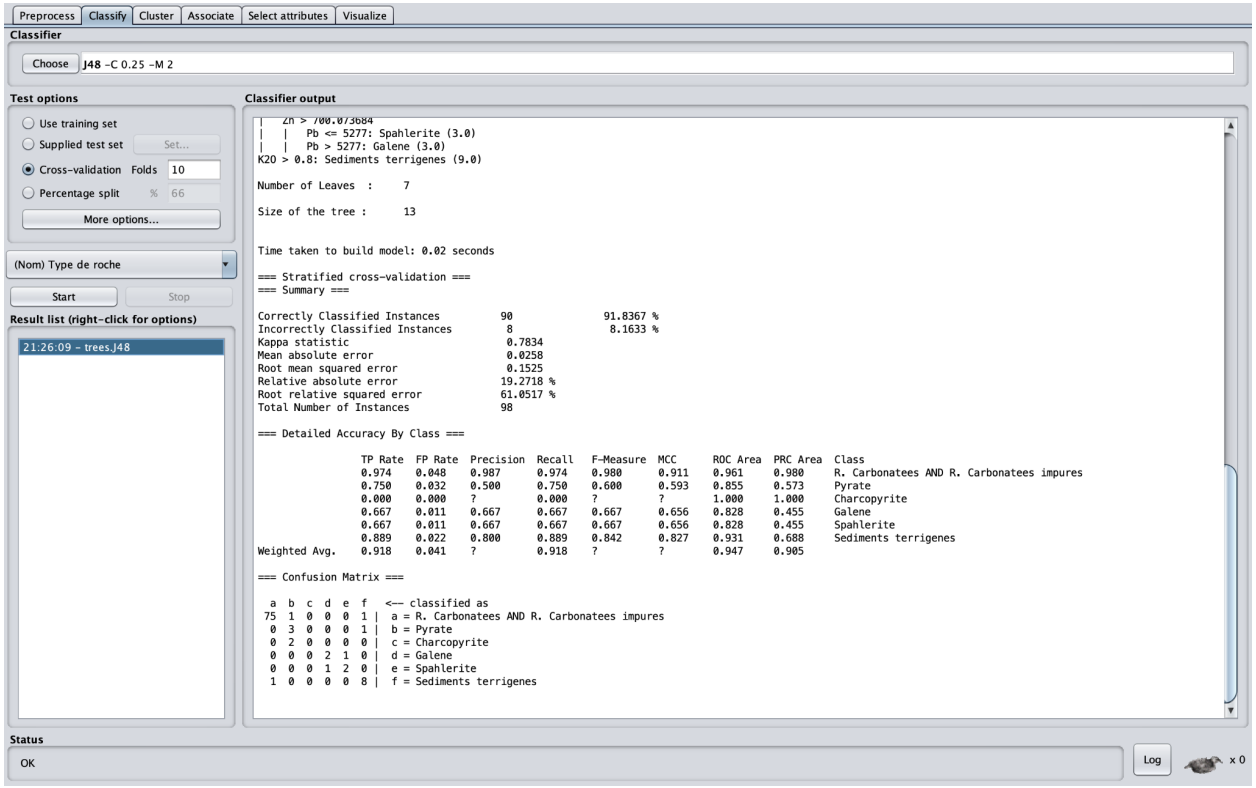
S3 : BUILDING CLASSIFIERS

Experiment 1 (Full Classification) :

We were asked to build a Decision Tree classifier and a Neural Network classifier for all classes C1-C6 simultaneously using different topologies and testing methods. This also needs to be done for both PD and PED.

Descriptive Classifier for PD :

We used the J48 decision tree algorithm for classification and used **K-Fold** of 10 folds cross validation technique to train the classifier on training and validation data.



The Predictive accuracy obtained using this algorithm is : **91.8367 %**

And the decision tree obtained is :

We can also see the Rules Accuracy beside each leaf of the denomination :

True Positive / False Positive

```
K2O <= 0.8
|   Zn <= 700.073684
|   |   Tb <= 0.8
|   |   |   Fe2O3* <= 0.76: R. Carbonatees AND R. Carbonatees impures (74.0)
|   |   |   Fe2O3* > 0.76
|   |   |   |   CaO <= 26.86: Pyrate (4.0)
|   |   |   |   CaO > 26.86: R. Carbonatees AND R. Carbonatees impures (3.0)
|   |   Tb > 0.8: Charcopyrite (2.0)
|   Zn > 700.073684
|   |   Pb <= 5277: Spahlerite (3.0)
|   |   Pb > 5277: Galene (3.0)
K2O > 0.8: Sediments terrigenes (9.0)
```

Number of Leaves : 7

Size of the tree : 13

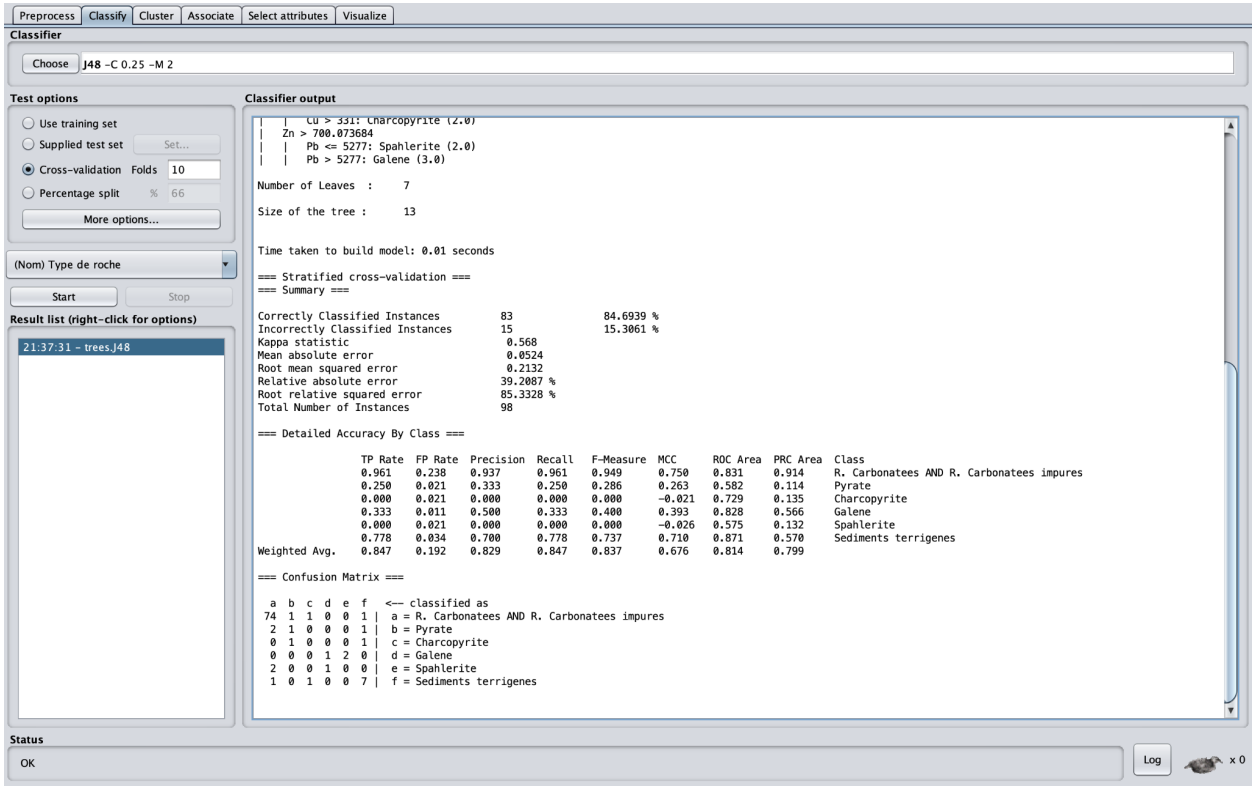
Discriminant Rules (Predicate Form) :

Rules Accuracy Mentioned on the side

- IF K2O(x, <= 0.8) AND Zn(x, <= 700.073684) AND Tb(x, <= 0.8) AND Fe2O3*(x, <= 0.76) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures) **(100%)**
- IF K2O(x, <= 0.8) AND Zn(x, <= 700.073684) AND Tb(x, <= 0.8) AND Fe2O3*(x, > 0.76) AND CaO(x, <= 26.86) THEN TYPE DE ROCHE(x, Pyrate)**(100%)**
- IF K2O(x, <= 0.8) AND Zn(x, <= 700.073684) AND Tb(x, <= 0.8) AND Fe2O3*(x, > 0.76) AND CaO(x, >26.86) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)**(100%)**
- IF K2O(x, <= 0.8) AND Zn(x, <= 700.073684) AND Tb(x, > 0.8) THEN TYPE DE ROCHE(x, Charcopyrite)**(100%)**
- IF K2O(x, <= 0.8) AND Zn(x, > 700.073684) AND Pb(x, <= 5277) THEN TYPE DE ROCHE(x, Spahlerite)**(100%)**
- IF K2O(x, <= 0.8) AND Zn(x, > 700.073684) AND Pb(x, > 5277) THEN TYPE DE ROCHE(x, Galene)**(100%)**
- IF K2O(x, > 0.8) THEN TYPE DE ROCHE(x, Sediments terrigenes)**(100%)**

Descriptive Classifier for PED :

We used the J48 decision tree algorithm for classification and used **K-Fold** of 10 folds cross validation technique to train the classifier on training and validation data.



The Predictive Accuracy obtained using this algorithm is : **84.6939 %**

And the decision tree obtained is :

We can also see the Rules Accuracy beside each leaf of the denomination :

True Positive / False Positive

```
Fe2O3* <= 0.45: R. Carbonatees AND R. Carbonatees impures (74.0/1.0)
Fe2O3* > 0.45
|   Zn <= 700.073684
|   |   Cu <= 331
|   |   |   CaO <= 22.84: Sediments terrigenes (10.0/1.0)
|   |   |   CaO > 22.84
|   |   |   |   S <= 2158: R. Carbonatees AND R. Carbonatees impures (4.0)
|   |   |   |   S > 2158: Pyrate (3.0)
|   |   Cu > 331: Charcopyrite (2.0)
|   Zn > 700.073684
|   |   Pb <= 5277: Spahlerite (2.0)
|   |   Pb > 5277: Galene (3.0)
```

Number of Leaves : 7

Size of the tree : 13

Discriminant Rules (Predicate Form) :

Rules Accuracy Mentioned on the side

- IF Fe2O3*(x, <= 0.45) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)**(98.66%)**
- IF Fe2O3*(x, > 0.45) AND Zn(x, <= 700.073684) AND Cu(x, <= 331) AND CaO(x, <= 22.84) THEN TYPE DE ROCHE(x, Sediments terrigenes)**(90.9%)**
- IF Fe2O3*(x, > 0.45) AND Zn(x, <= 700.073684) AND Cu(x, <= 331) AND CaO(x, > 22.84) AND S(x, <= 2158) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)**(100%)**
- IF Fe2O3*(x, > 0.45) AND Zn(x, <= 700.073684) AND Cu(x, <= 331) AND CaO(x, > 22.84) AND S(x, > 2158) THEN TYPE DE ROCHE(x, Pyrate)**(100%)**
- IF Fe2O3*(x, > 0.45) AND Zn(x, <= 700.073684) AND Cu(x, > 331) THEN TYPE DE ROCHE(x, Charcopyrite)**(100%)**
- IF Fe2O3*(x, > 0.45) AND Zn(x, > 700.073684) AND Pb(x, <= 5277) THEN TYPE DE ROCHE(x, Spahlerite)**(100%)**
- IF Fe2O3*(x, > 0.45) AND Zn(x, > 700.073684) AND Pb(x, > 5277) THEN TYPE DE ROCHE(x, Galene)**(100%)**

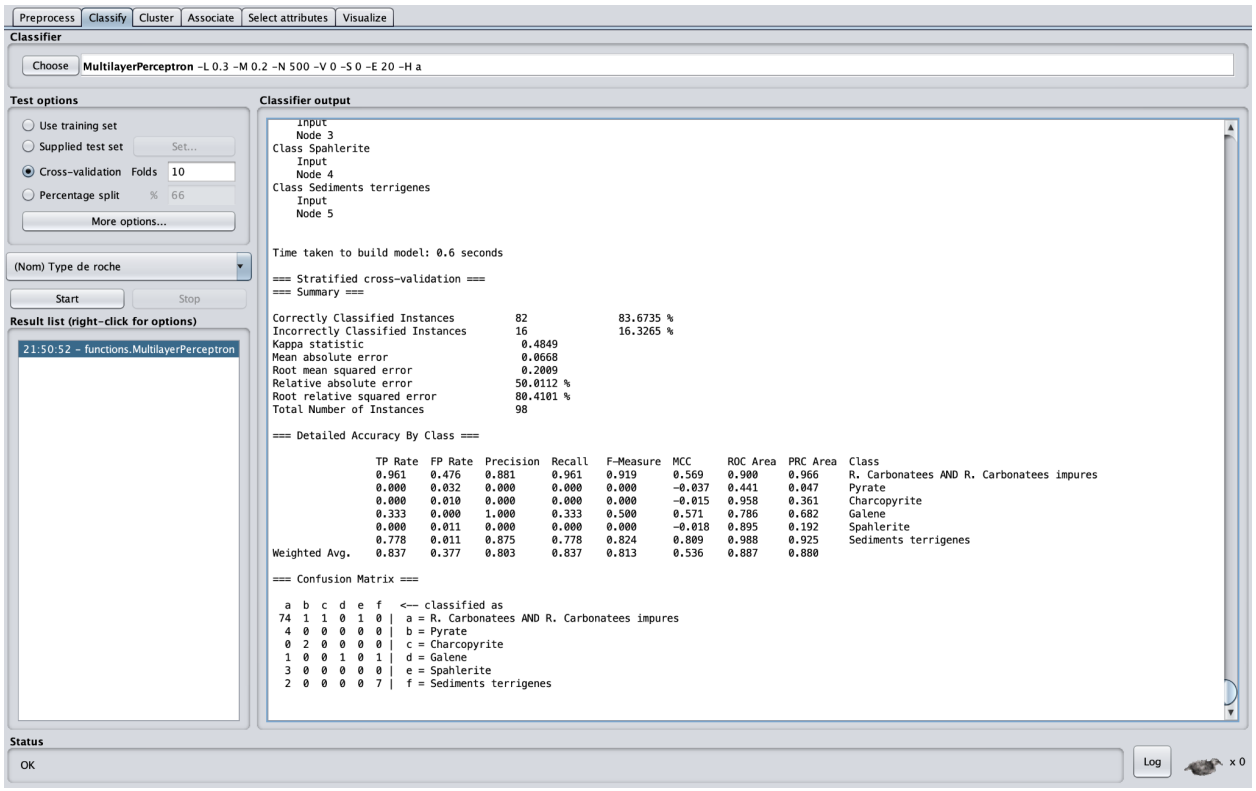
Non-Descriptive Classifier for PD :

We used the MultiLayerPerceptron for classification and used the **K-Fold** of 10 folds cross validation technique to train the classifier on training and validation data.

We have used different hyperparameters and the accuracies are as below.

Case 1 :

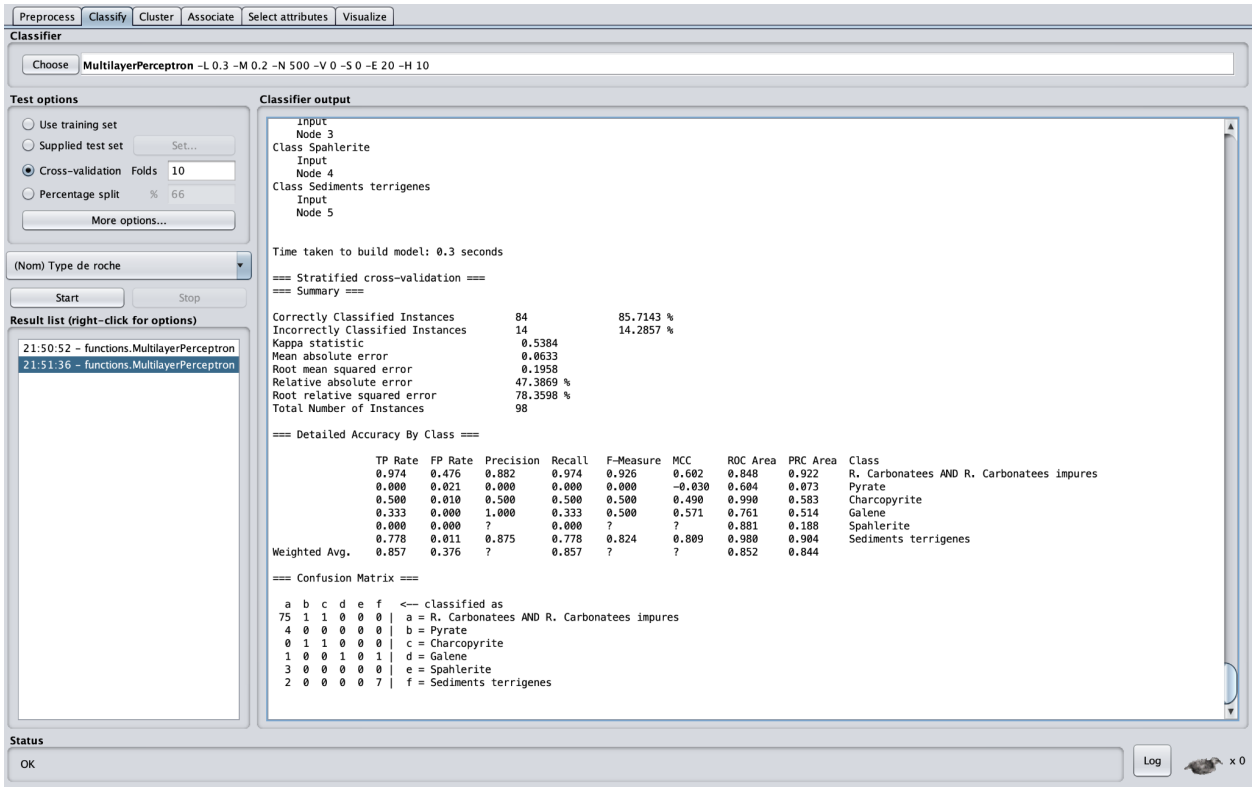
Learning rate : 0.3
Momentum : 0.2
Epochs : 500
Number of Hidden layers : a (average of input and output)



The Predictive Accuracy of this algorithm is : **83.6735 %**

Case 2 :

Learning rate : 0.3
Momentum : 0.2
Epochs : 500
Number of Hidden layers : 10



The Predictive Accuracy of this algorithm is : **85.7143 %**

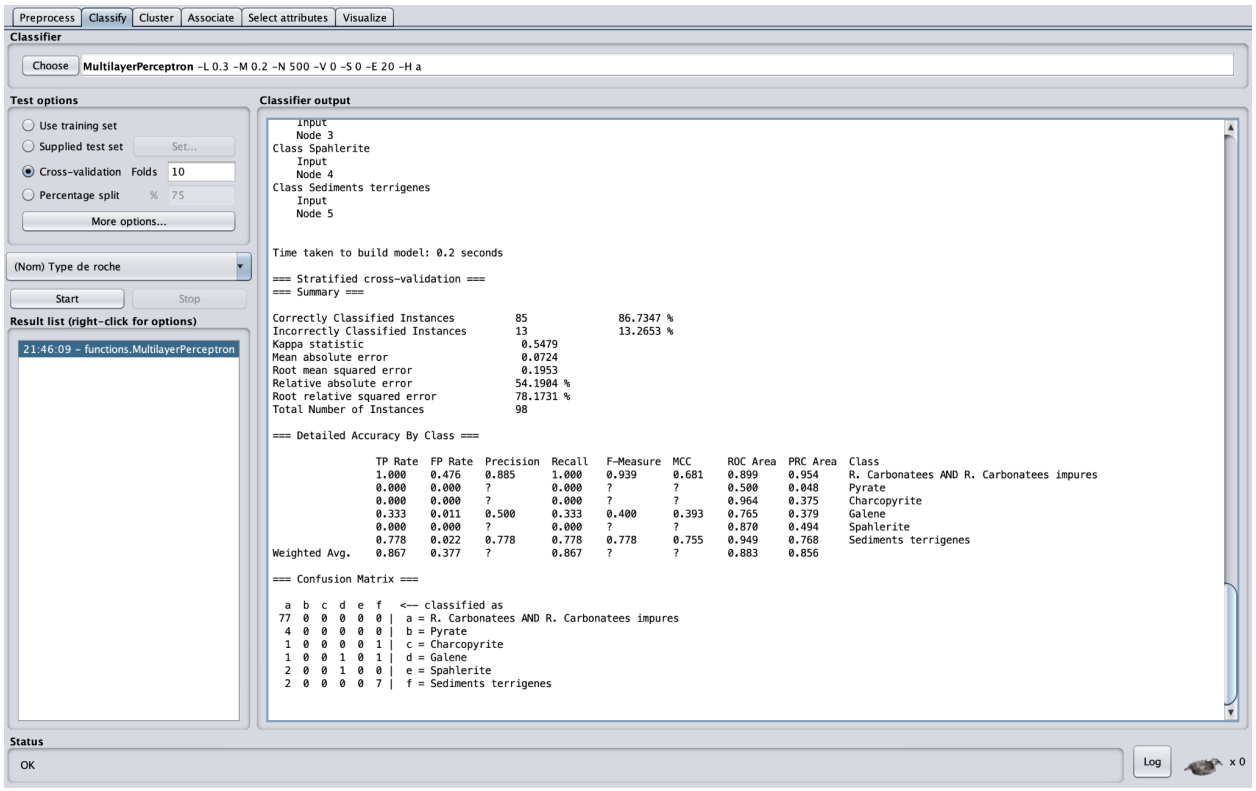
Non-Descriptive Classifier for PED :

We used the MultiLayerPerceptron for classification and used the **K-Fold** of 10 folds cross validation technique to train the classifier on training and validation data.

We have used different hyperparameters and the accuracies are as below.

Case 1 :

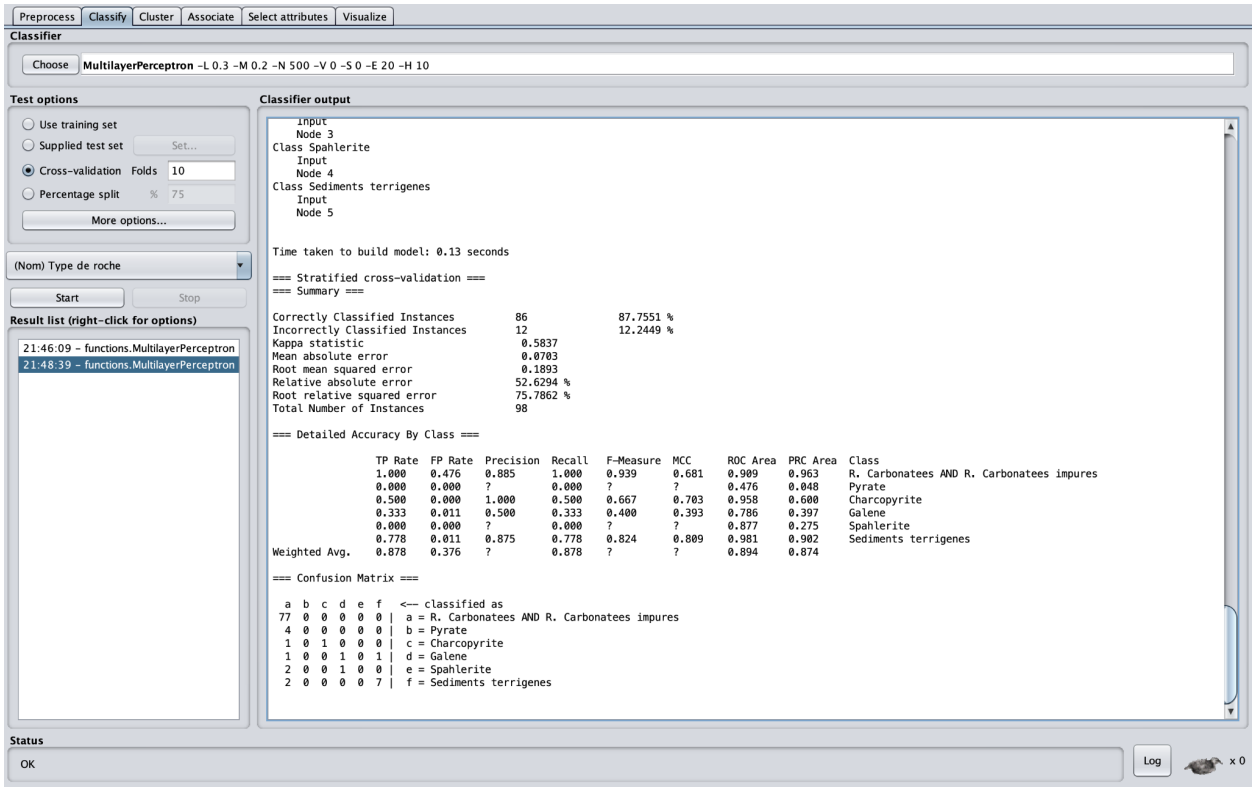
Learning rate : 0.3 Momentum : 0.2 Epochs : 500 Number of Hidden layers : a (average of input and output)



The Predictive Accuracy of this algorithm is : **86.7347 %**

Case 2 :

Learning rate : 0.3
Momentum : 0.2
Epochs : 500
Number of Hidden layers : 10



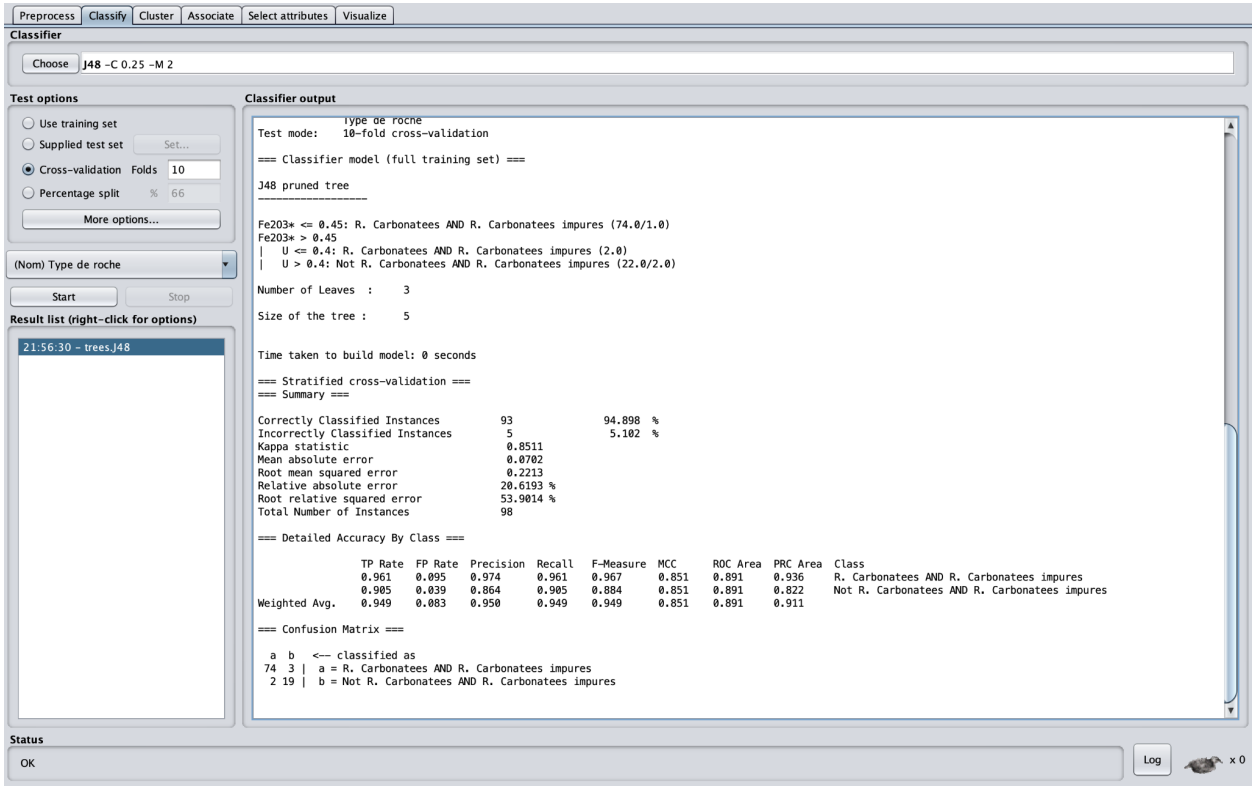
The Predictive Accuracy of this algorithm is : **87.7551 %**

Experiment 2 (Contrast Classification) :

We were asked to build a Decision Tree classifier and a Neural Network classifier to perform the contrast classification for the class C1 i.e. **R. Carbonatees AND R. Carbonatees impures** using different topologies and testing methods. This also needs to be done for both PD and PED.

Descriptive Classifier for PD :

We used the J48 decision tree algorithm for classification and used **K-Fold** of 10 folds cross validation technique to train the classifier on training and validation data.



The Predictive Accuracy obtained using this algorithm is : **94.898 %**

And the decision tree obtained is :

We can also see the Rules Accuracy beside each leaf of the denomination :

True Positive / False Positive

Fe2O3* <= 0.45: R. Carbonatees AND R. Carbonatees impures (74.0/1.0)
Fe2O3* > 0.45
| U <= 0.4: R. Carbonatees AND R. Carbonatees impures (2.0)
| U > 0.4: Not R. Carbonatees AND R. Carbonatees impures (22.0/2.0)

Number of Leaves : 3

Size of the tree : 5

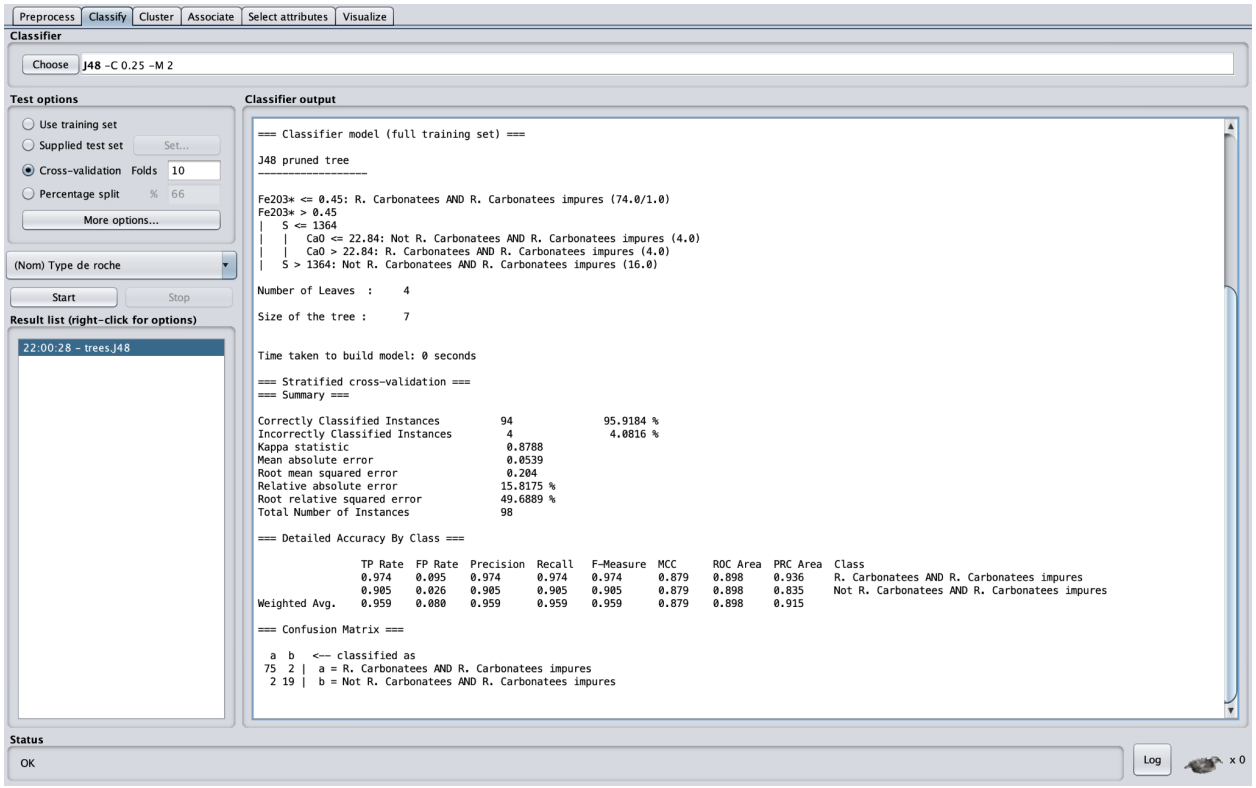
Discriminant Rules (Predicate Form) :

Rules Accuracy Mentioned on the side

- IF Fe2O3*(x, <= 0.45) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)**(98.66%)**
- IF Fe2O3*(x, > 0.45) AND U(x, <= 0.4) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)**(100%)**
- IF Fe2O3*(x, > 0.45) AND U(x, > 0.4) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures)**(91.66%)**

Descriptive Classifier for PED :

We used the J48 decision tree algorithm for classification and used **K-Fold** of 10 folds cross validation technique to train the classifier on training and validation data.



The Predictive Accuracy obtained using this algorithm is : **95.9184 %**

And the decision tree obtained is :

We can also see the Rules Accuracy beside each leaf of the denomination :

True Positive / False Positive

```
Fe2O3* <= 0.45: R. Carbonatees AND R. Carbonatees impures (74.0/1.0)
Fe2O3* > 0.45
|   S <= 1364
|   |   CaO <= 22.84: Not R. Carbonatees AND R. Carbonatees impures (4.0)
|   |   CaO > 22.84: R. Carbonatees AND R. Carbonatees impures (4.0)
|   S > 1364: Not R. Carbonatees AND R. Carbonatees impures (16.0)
```

```
Number of Leaves  : 4

Size of the tree : 7
```

<u>Discriminant Rules (Predicate Form) :</u>

Rules Accuracy Mentioned on the side

- IF Fe2O3*(x, <= 0.45) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)**(98.66%)**
- IF Fe2O3*(x, > 0.45) AND S(x, <= 1364) AND CaO(x, <= 22.84) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures)**(100%)**
- IF Fe2O3*(x, > 0.45) AND S(x, <= 1364) AND CaO(x, > 22.84) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)**(100%)**
- IF Fe2O3*(x, > 0.45) AND S(x, <= 1364) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures)**(100%)**

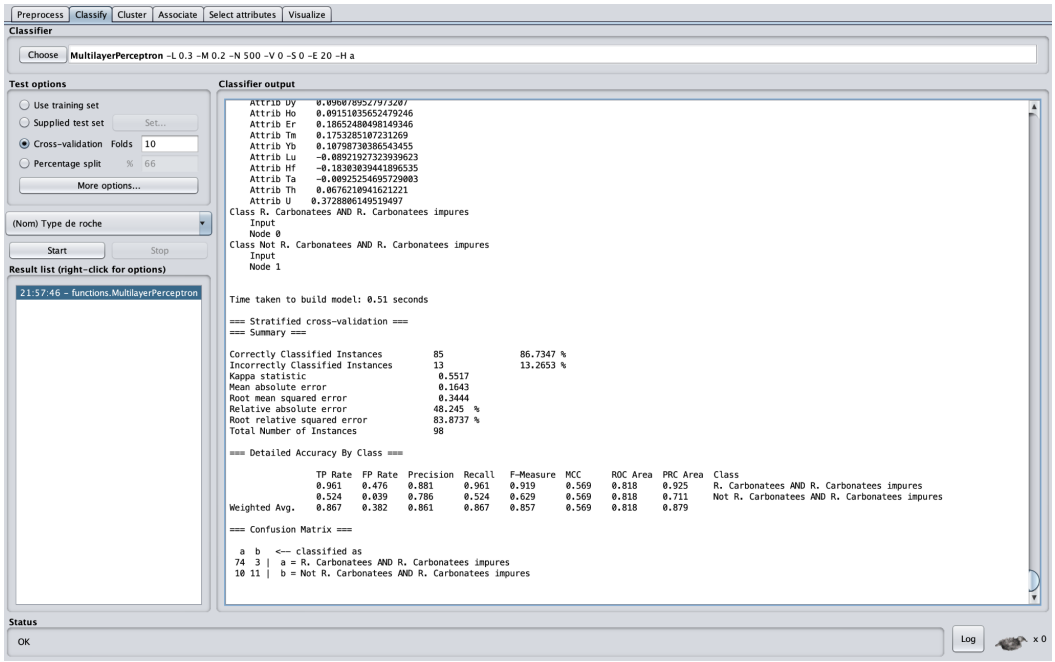
Non-Descriptive Classifier for PD :

We used the MultiLayerPerceptron for classification and used the **K-Fold** of 10 folds cross validation technique to train the classifier on training and validation data.

We have used different hyperparameters and the accuracies are as below.

Case 1 :

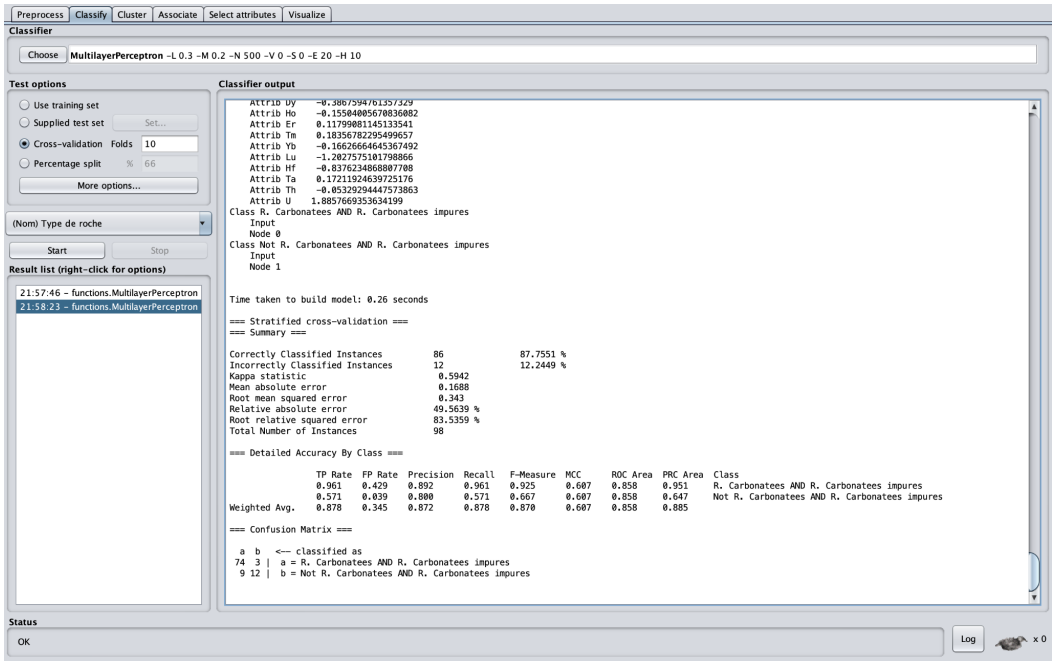
Learning rate : 0.3
Momentum : 0.2
Epochs : 500
Number of Hidden layers : a (average of input and output)



The Predictive Accuracy of this algorithm is : **86.7347 %**

Case 2 :

Learning rate : 0.3
Momentum : 0.2
Epochs : 500
Number of Hidden layers : 10



The Predictive Accuracy of this algorithm is : **87.7551 %**

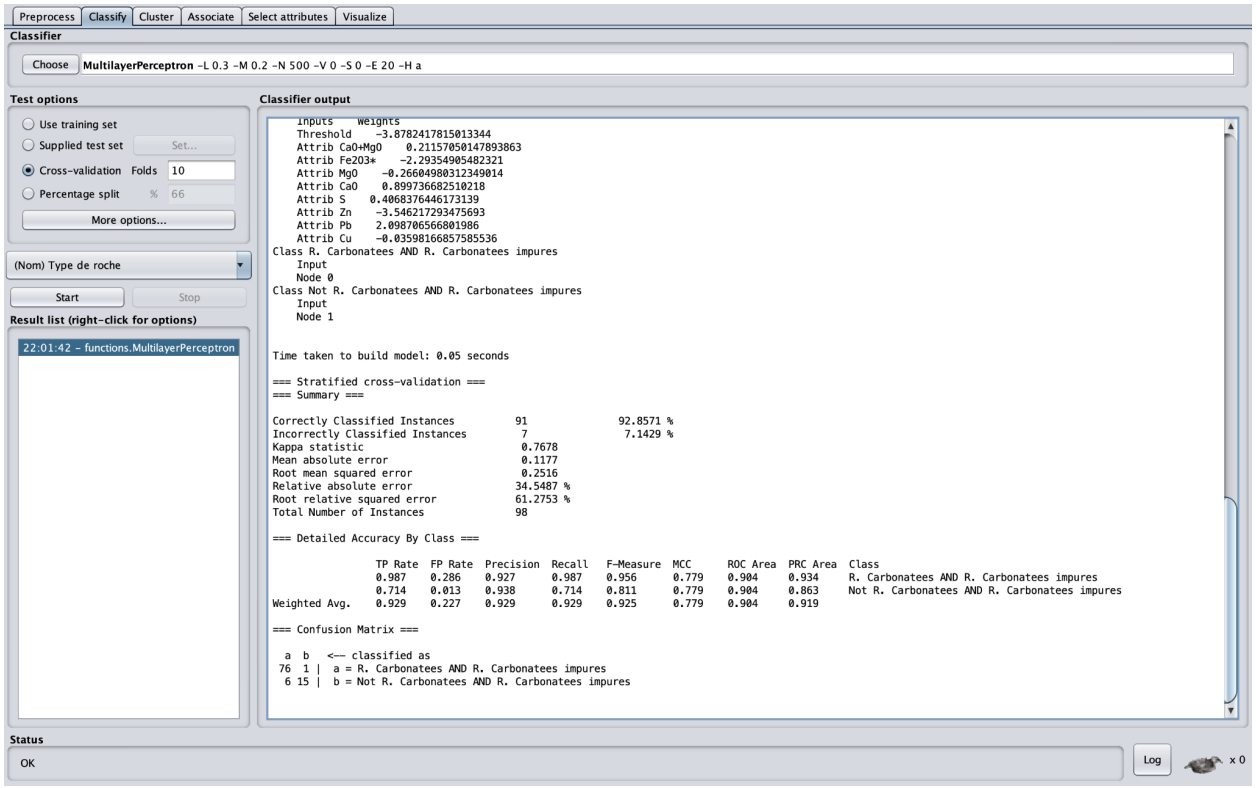
Non-Descriptive Classifier for PED :

We used the MultiLayerPerceptron for classification and used the **K-Fold** of 10 folds cross validation technique to train the classifier on training and validation data.

We have used different hyperparameters and the accuracies are as below.

Case 1 :

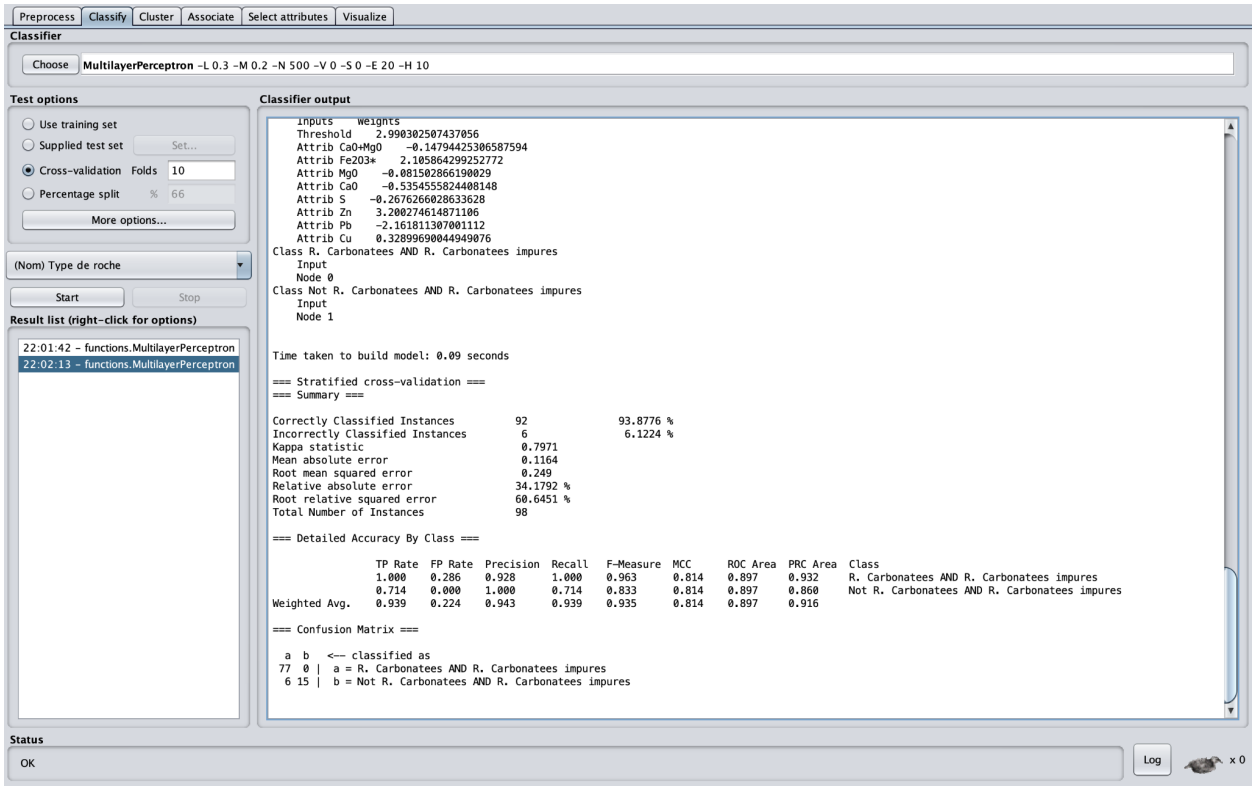
Learning rate : 0.3
Momentum : 0.2
Epochs : 500
Number of Hidden layers : a (average of input and output)



The Predictive Accuracy of this algorithm is : **92.8571 %**

<u>Case 2</u> :

Learning rate : 0.3
Momentum : 0.2
Epochs : 500
Number of Hidden layers : 10



The Predictive Accuracy of this algorithm is : **93.8776 %**

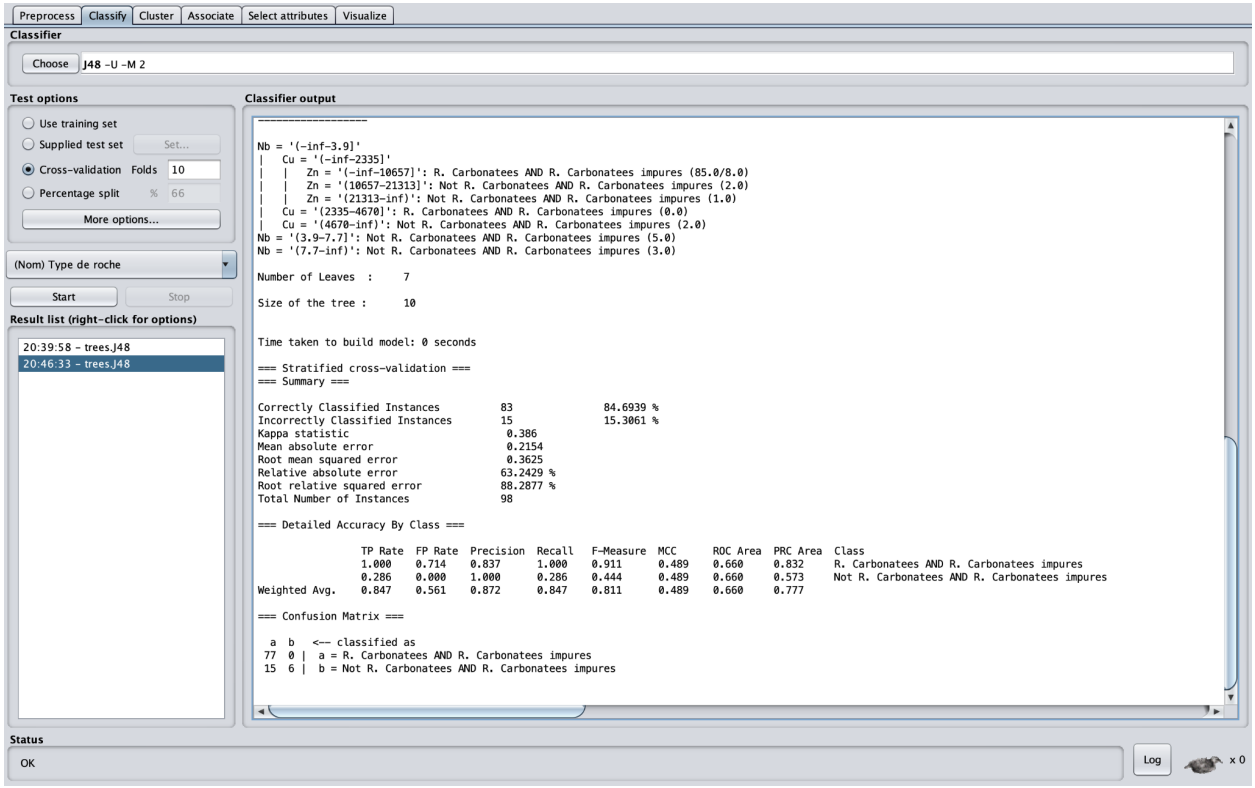
Experiment 3 :

PD1 :

Since the generation of the PD1 dataset is already done as part of Data Preprocessing. We continue to generate the classifiers.

1) M1 :

We used the J48 decision tree algorithm for classification and used **K-Fold** of 10 folds cross validation technique to train the classifier on training and validation data. The tree obtained does have all the possible values of each attribute(after discretization) which are part of the decision tree. It also contains a separate branch coming out of each attribute mentioning the possible values and possible classification paths.



The Predictive Accuracy obtained using this algorithm is : **84.6939 %**

And the decision tree obtained is :

We can also see the Rules Accuracy beside each leaf of the denomination :

True Positive / False Positive

```
Nb = '(-inf-3.9]'
|   Cu = '(-inf-2335]'
|   |   Zn = '(-inf-10657]': R. Carbonatees AND R. Carbonatees impures (85.0/8.0)
|   |   Zn = '(10657-21313]': Not R. Carbonatees AND R. Carbonatees impures (2.0)
|   |   Zn = '(21313-inf)': Not R. Carbonatees AND R. Carbonatees impures (1.0)
|   Cu = '(2335-4670]': R. Carbonatees AND R. Carbonatees impures (0.0)
|   Cu = '(4670-inf)': Not R. Carbonatees AND R. Carbonatees impures (2.0)
Nb = '(3.9-7.7]': Not R. Carbonatees AND R. Carbonatees impures (5.0)
Nb = '(7.7-inf)': Not R. Carbonatees AND R. Carbonatees impures (3.0)
```

```
Number of Leaves  : 7

Size of the tree : 10
```

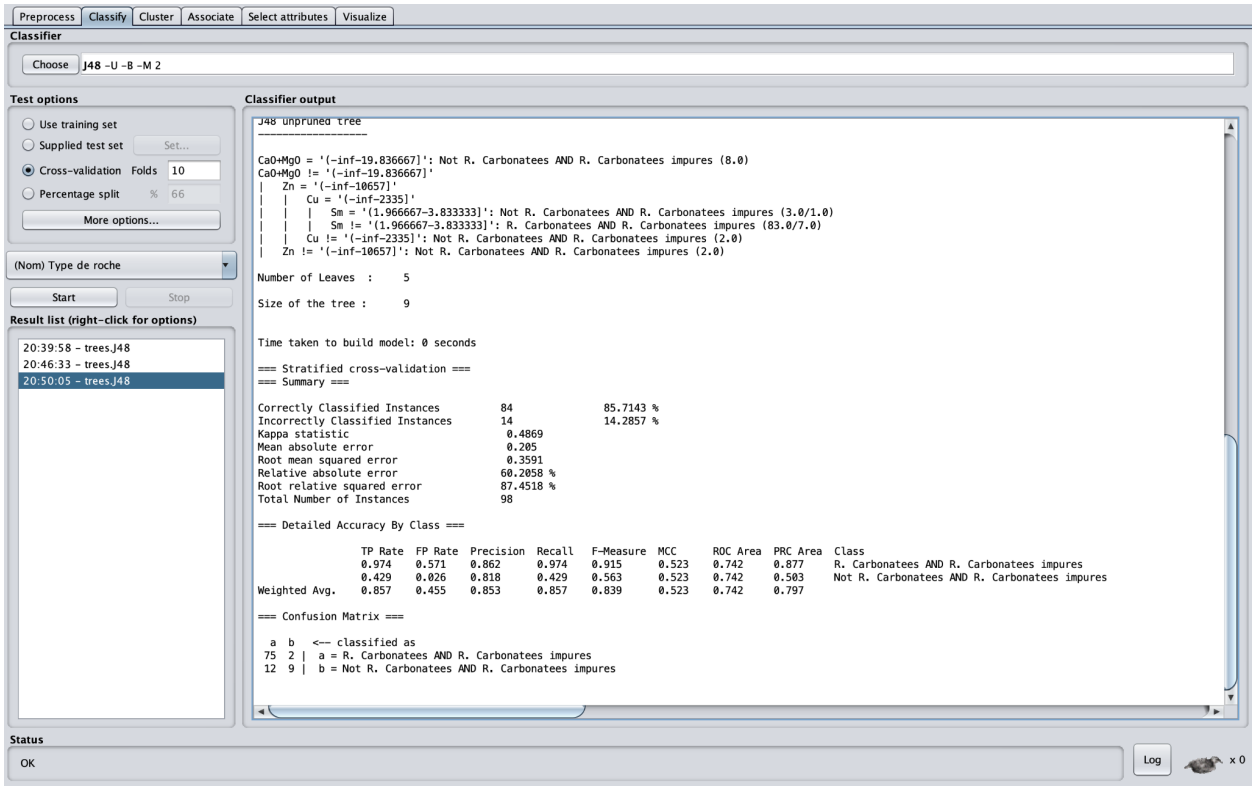
Discriminant Rules (Predicate Form) :

Rules Accuracy Mentioned on the side

- IF Nb(x, <= 3.9) AND Cu(x, <= 2335) AND Zn(x, <= 10657) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)**(91.4%)**
- IF Nb(x, <= 3.9) AND Cu(x, <= 2335) AND Zn(x, > 10657) AND Zn(x, <= 21313) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures)**(100%)**
- IF Nb(x, <= 3.9) AND Cu(x, <= 2335) AND Zn(x, > 21313) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures)**(100%)**
- IF Nb(x, <= 3.9) AND Cu(x, > 2335) AND Cu(x, <= 4670) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)**(100%)**
- IF Nb(x, <= 3.9) AND Cu(x, > 4670) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures)**(100%)**
- IF Nb(x, > 3.9) AND Nb(x, <= 7.7) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures)**(100%)**
- IF Nb(x, > 7.7) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures)**(100%)**

2) M2 :

We used the J48 decision tree algorithm for classification and used **K-Fold** of 10 folds cross validation technique to train the classifier on training and validation data. This decision tree contains only binary branches for each attribute which is part of the final tree. These values of splitting points are not similar to what we've seen for M1. These aren't some actual true values present in the dataset but some splitting points which are generated from WEKA.



The Predictive Accuracy obtained using this algorithm is : **85.7143 %**

And the decision tree obtained is :

We can also see the Rules Accuracy beside each leaf of the denomination :

True Positive / False Positive

```
CaO+MgO = '(-inf-19.836667]': Not R. Carbonatees AND R. Carbonatees impures (8.0)
CaO+MgO != '(-inf-19.836667]'
|   Zn = '(-inf-10657]'
|   |   Cu = '(-inf-2335]'
|   |   |   Sm = '(1.966667-3.833333]': Not R. Carbonatees AND R. Carbonatees impures (3.0/1.0)
|   |   |   Sm != '(1.966667-3.833333]': R. Carbonatees AND R. Carbonatees impures (83.0/7.0)
|   |   Cu != '(-inf-2335]': Not R. Carbonatees AND R. Carbonatees impures (2.0)
|   Zn != '(-inf-10657]': Not R. Carbonatees AND R. Carbonatees impures (2.0)
```

Number of Leaves : 5

Size of the tree : 9

Discriminant Rules (Predicate Form) :

Rules Accuracy Mentioned on the side

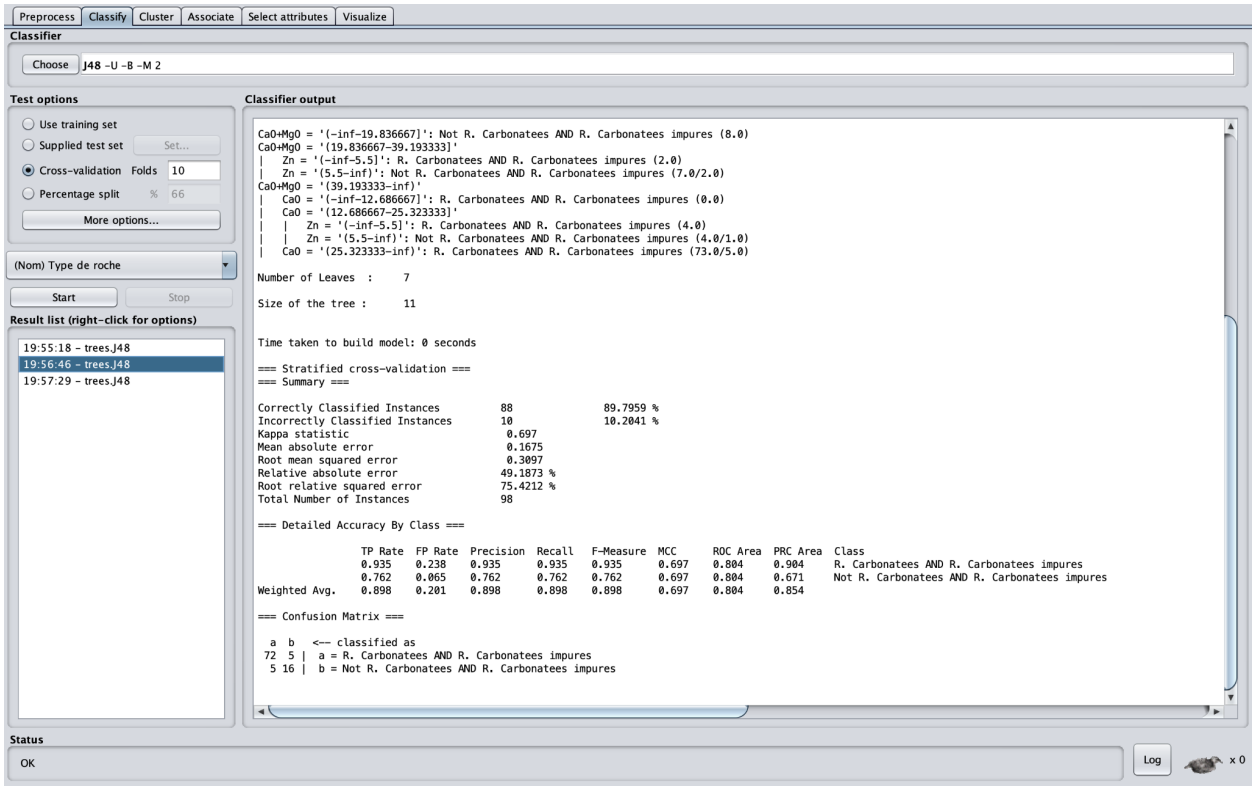
- IF CaO+MgO (x, <= 19.836667) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures)**(100%)**
- IF CaO+MgO (x, > 19.836667) AND Zn(x, <= 10657) AND Cu(x, <= 2335) AND Sm(x, 1.966667 .. 3.833333) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures)**(75%)**
- IF CaO+MgO (x, > 19.836667) AND Zn(x, <= 10657) AND Cu(x, <= 2335) AND Sm(x, !(1.966667 .. 3.833333)) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)**(92.22%)**
- IF CaO+MgO(x, > 19.836667) AND Zn(x, <= 10657) AND Cu(x, > 2335) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures)**(100%)**
- IF CaO+MgO (x, > 19.836667) AND Zn(x, > 10657) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures)**(100%)**

PED 1 :

Since the generation of the PED1 dataset is already done as part of Data Preprocessing. We continue to generate the classifiers.

1) **M1 :**

We used the J48 decision tree algorithm for classification and used **K-Fold** of 10 folds cross validation technique to train the classifier on training and validation data. The tree obtained does have all the possible values of each attribute(after discretization) which are part of the decision tree. It also contains a separate branch coming out of each attribute mentioning the possible values and possible classification paths.



The accuracy obtained using this algorithm is : **89.7959 %**

And the decision tree obtained is :

We can also see the Rules Accuracy beside each leaf of the denomination :

True Positive / False Positive

```
CaO+MgO = '(-inf-19.836667]': Not R. Carbonatees AND R. Carbonatees impures (8.0)
CaO+MgO = '(19.836667-39.193333]'
|   Zn = '(-inf-5.5]': R. Carbonatees AND R. Carbonatees impures (2.0)
|   Zn = '(5.5-inf)': Not R. Carbonatees AND R. Carbonatees impures (7.0/2.0)
CaO+MgO = '(39.193333-inf)'
|   CaO = '(-inf-12.686667]': R. Carbonatees AND R. Carbonatees impures (0.0)
|   CaO = '(12.686667-25.323333]'
|   |   Zn = '(-inf-5.5]': R. Carbonatees AND R. Carbonatees impures (4.0)
|   |   Zn = '(5.5-inf)': Not R. Carbonatees AND R. Carbonatees impures (4.0/1.0)
|   CaO = '(25.323333-inf)': R. Carbonatees AND R. Carbonatees impures (73.0/5.0)
```

```
Number of Leaves  : 7

Size of the tree : 11
```

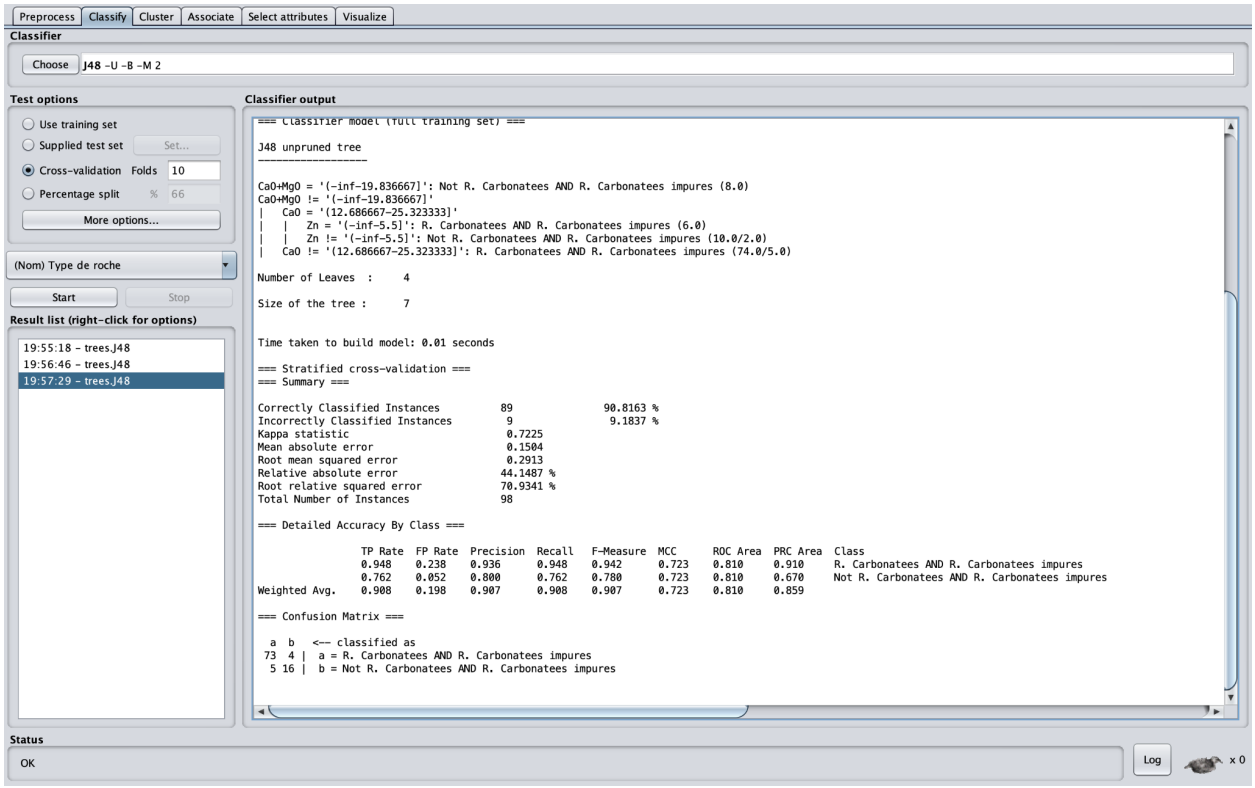
Discriminant Rules (Predicate Form) :

Rules Accuracy Mentioned on the side

- IF CaO+MgO(x, <= 19.836667) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures)**(100%)**
- IF CaO+MgO(x, > 19.836667) AND CaO+MgO(x, <= 39.193333) AND Zn(x, <= 5.5) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)**(100%)**
- IF CaO+MgO(x, > 19.836667) AND CaO+MgO(x, <= 39.193333) AND Zn(x, > 5.5) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures)**(77.77%)**
- IF CaO+MgO(x, > 39.193333) AND CaO(x, <= 12.686667) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)**(100%)**
- IF CaO+MgO(x, > 39.193333) AND CaO(x, > 12.686667) AND CaO(x, <= 25.323333) AND Zn(x, <= 5.5) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)**(100%)**
- IF CaO+MgO(x, > 39.193333) AND CaO(x, > 12.686667) AND CaO(x, <= 25.323333) AND Zn(x, > 5.5) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures)**(80%)**
- IF CaO+MgO(x, > 39.193333) AND CaO(x, > 25.323333) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)**(93.5%)**

2) M2 :

We used the J48 decision tree algorithm for classification and used **K-Fold** of 10 folds cross validation technique to train the classifier on training and validation data. This decision tree contains only binary branches for each attribute which is part of the final tree. These values of splitting points are not similar to what we've seen for M1. These aren't some actual true values present in the dataset but some splitting points which are generated from WEKA.



The accuracy obtained using this algorithm is : **90.8163 %**

And the decision tree obtained is :

We can also see the Rules Accuracy beside each leaf of the denomination :

True Positive / False Positive

```
CaO+MgO = '(-inf-19.836667]': Not R. Carbonatees AND R. Carbonatees impures (8.0)
CaO+MgO != '(-inf-19.836667]'
|   CaO = '(12.686667-25.323333]'
|   |   Zn = '(-inf-5.5]': R. Carbonatees AND R. Carbonatees impures (6.0)
|   |   Zn != '(-inf-5.5]': Not R. Carbonatees AND R. Carbonatees impures (10.0/2.0)
|   CaO != '(12.686667-25.323333]': R. Carbonatees AND R. Carbonatees impures (74.0/5.0)
```

```
Number of Leaves  : 4

Size of the tree : 7
```

<u>Discriminant Rules (Predicate Form) :</u>

Rules Accuracy Mentioned on the side

- IF CaO+MgO(x, <= 19.836667) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures)**(100%)**
- IF CaO+MgO(x, > 19.836667) AND CaO(x, 12.686667 .. 25.323333) AND Zn(x, <= 5.5) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)**(100%)**
- IF CaO+MgO(x, > 19.836667) AND CaO(x, 12.686667 .. 25.323333) AND Zn(x, > 5.5) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures)**(83.33%)**
- IF CaO+MgO(x, > 19.836667) AND CaO(x, !(12.686667 .. 25.323333)) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)**(93.6%)**

SUMMARY

Predictive Accuracies for Descriptive Classifiers :

PD - Complete	91.8367
PED - Complete	84.6939
PD - Contrast	94.898
PED - Contrast	95.9184

Here we have used the J48 unpruned classifier along with 10 Fold Validation for both PD and PED (both Complete and Contrast classifiers). If we see, the classifier has worked much better for a complete data set than just with the important attributes when we tried to classify all the classes at once. On the other hand, contrast classification has worked much better when we worked on important attributes. The Rules Accuracies have been mentioned as part of the classifiers description above. Where as the attributes **Zn**, **Pb**, **CaO** and **Fe2O3*** seem to be common across both the complete classification decision trees. Where as **Fe2O3*** seem to be common for contrast classification.

Apart from these, the decision trees are much more compact for contrast classification compared to complete classifiers.

Predictive Accuracies for Non - Descriptive Classifiers :

Complete classification:

PD - Case 1	83.6735
PD - Case 2	85.7143
PED - Case 1	86.7347
PED - Case 2	87.7551

Contrast classification:

PD - Case 1	86.7347
PD - Case 2	87.7551
PED - Case 1	92.8571
PED - Case 2	93.8776

Here we used a Neural Network algorithm along with 10 Fold cross validation. If we can see the accuracies are higher for the classifiers where we have selected 10 hidden layers compared to the default settings. Along with that the accuracies are higher for contrast classifiers compared to the complete classification.

Highest Predictive Accuracies across all Classifiers :

Complete Classification	91.8367	Descriptive Classifier when ran on PD data
Contrast Classification	95.9184	Descriptive Classifier when ran on PED data

Predictive Accuracies for Experiment 3 :

PD1 - M1	84.6939
PD1 - M2	85.7143
PED1 - M1	89.7959
PED1- M2	90.8163

Here we have used the J48 unpruned classifier along with 10 Fold Validation for both PD1 and PED1. If observed the decision trees are different and their corresponding rules accuracy have been mentioned along with the tree. Coming to the predictive accuracy, we can clearly see that PED1 data has worked much better compared to the PD1 data with the discretization methods used (mentioned as part of Data Preprocessing).

ORANGE

Introduction about the tool :

Orange is an Open source machine learning and data visualization tool that helps to uncover hidden data patterns and provide intuition behind data analysis procedures. It also helps to Build data analysis workflows visually, with a large, diverse toolbox. It performs data analysis with clever data visualizations. It aids the user to build statistical distributions, box plots and scatter plots, decision trees, hierarchical clustering, heatmaps, MDS and linear projections.

For building descriptive and non descriptive classifiers, we have generated two cleaned data sets **PD** and **PED** in the form of csv files in the Data Preparation step for Weka. We have used these same csv files for the Orange tool.

S2 : DATA PREPROCESSING

1) Descriptive Classifiers :

Firstly we have uploaded the cleaned csv file pertaining to data set **PD** to Orange by dragging and dropping the File-widget to the workspace and then uploading it. In the File widget we have mapped all the attributes to **Features** and then mapped the **Type de roche** column to **Target**.

By creating a **Data Table** widget in the workspace and connecting it with a File widget, we can identify and verify the data which we are supposed to use to build a decision tree. In the '**Data Table** widget' we can see that there are missing values for some attributes.

To populate the missing values in Orange we can use a **Preprocess** widget. By connecting the File widget output to Preprocess widget, we can perform preprocess operations in Orange. We have populated missing values by using the '**Impute Missing Values**' option in the preprocess window. Here we have selected the '**Average/Most frequent**' method to fill the missing values.

We have verified the preprocessed data by creating another Data Table widget and connecting it from the Preprocess widget.

2) Non Descriptive Classifiers :

We were asked to normalize the data for classification. For this, we have selected the "**Normalize Features**" option in the Preprocess window and then selected **Normalize to interval[0.1]**. This uses the **"Min-Max Normalization" (0-1 Normalization)** technique.

S3 : BUILDING CLASSIFIERS

Experiment 1 (Full Classification) :

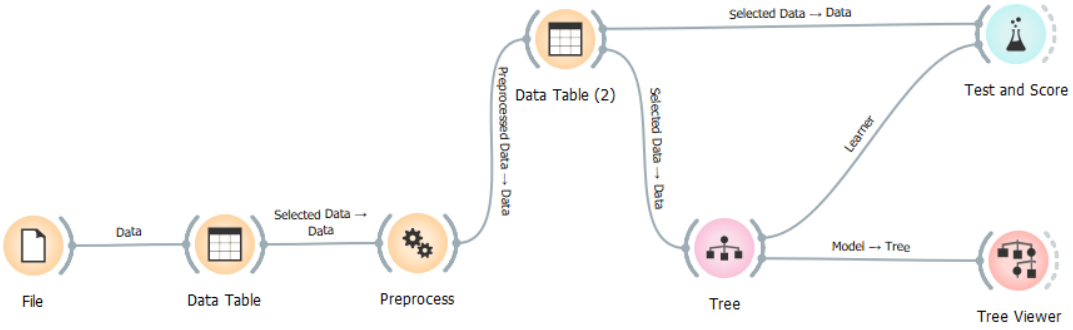
We were asked to build a Decision Tree classifier and a Neural Network classifier for all classes C1-C6 simultaneously using different topologies and testing methods. This needs to be done for both PD and PED.

Descriptive Classifier for PD :

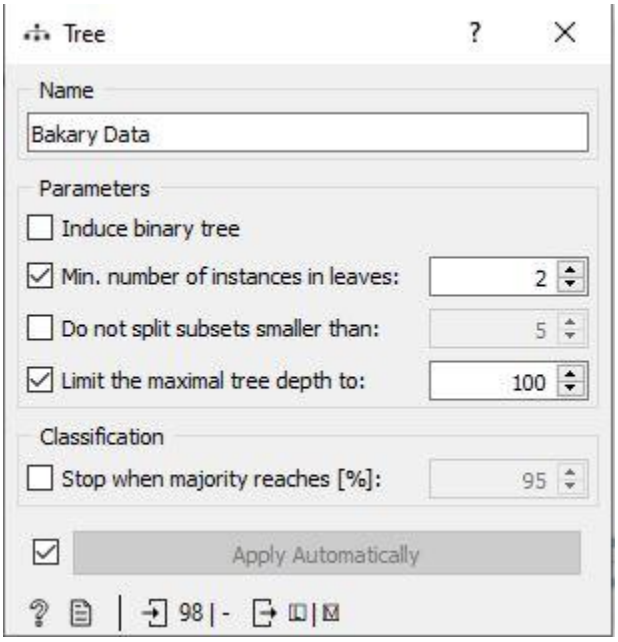
For the purpose of building a descriptive classifier for PD, we have used a Tree widget in Orange.

Tree widget of Oranges takes preprocessed data and builds the decision tree. The tree can be visualised using a **Tree Viewer**. We use **Test and Score** to get the **Classification Accuracy**(Predictive accuracy) of the decision tree.

The discriminant rules can be written from the tree viewer. The architecture for building the decision tree for this experiment is as below.



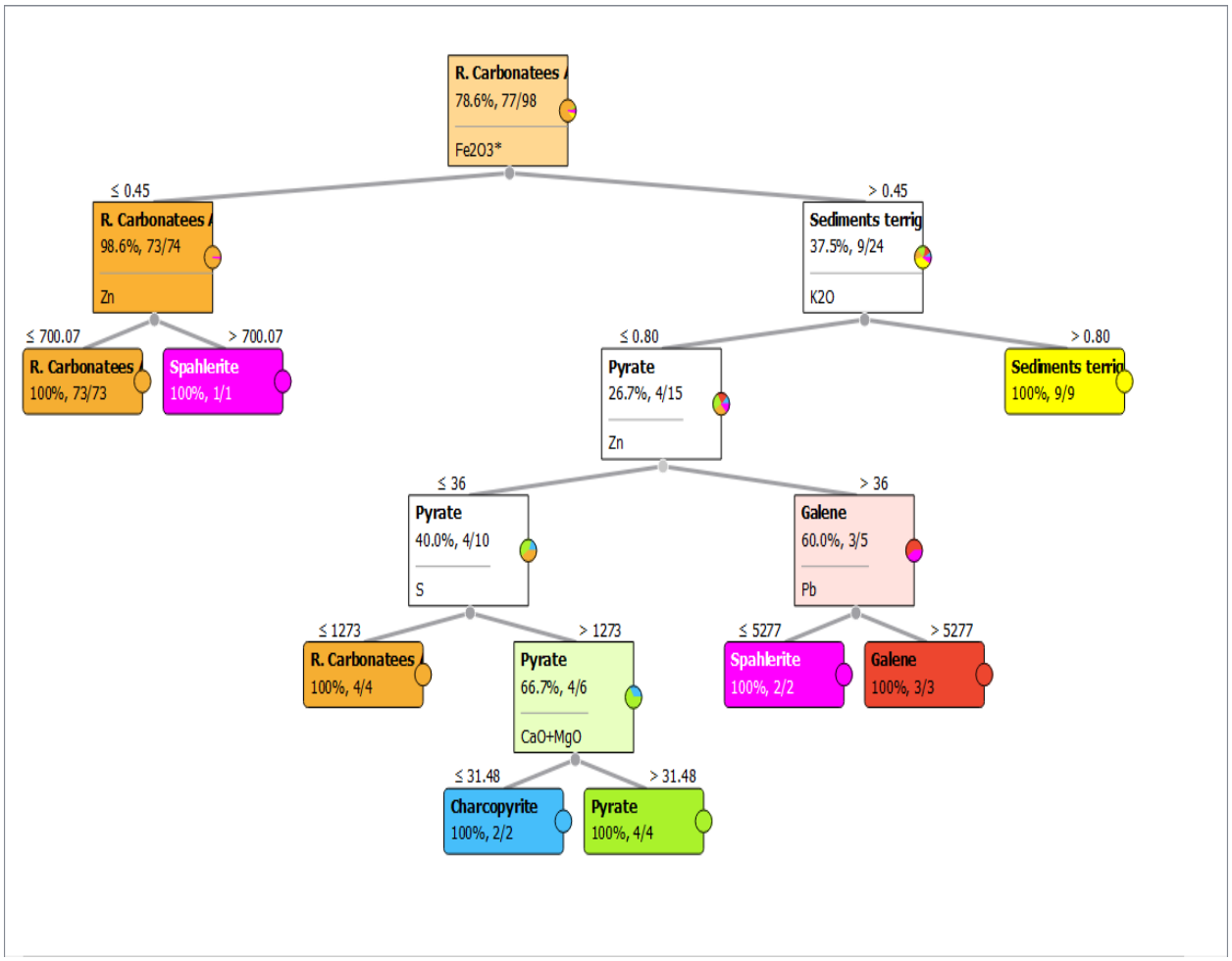
The tree configuration for the experiment:

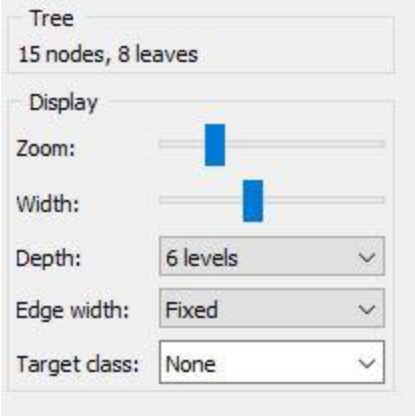


The tree viewer of Orange gives us an option to select the number of levels that a tree can contain. We saw that selecting 6 levels for the tree built from **PD** dataset gave the best results by classifying all the 6 classes. Lower number levels were only able to classify a few classes.

We can also see the Rules Accuracy in each leaf node of the Tree :

True Positive / Total





<u>Discriminant rules(Predicate Form)</u>:

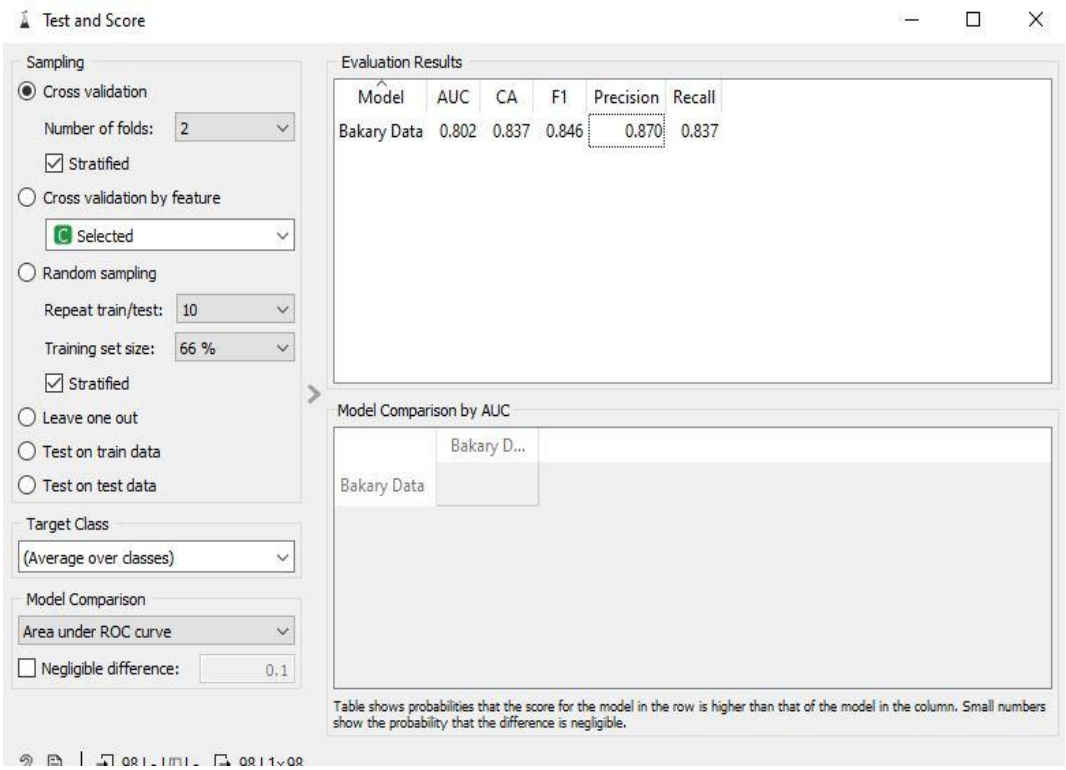
Rules Accuracy Mentioned on the side

- IF Fe2O3(x, <= 0.45) AND Zn(x, <= 700.07) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures).**(100%)**
- IF Fe2O3(x, <= 0.45) AND Zn(x, > 700.07) THEN TYPE DE ROCHE(x, Spahlerite). **(100%)**
- IF Fe2O3(x, > 0.45) AND K2O(x, > 0.8) THEN TYPE DE ROCHE(x, Sediments terrigenes). **(100%)**
- IF Fe2O3(x, > 0.45) AND K2O(x, <= 0.8) AND Zn(x, >36) AND Pb(x, >5277) THEN TYPE DE ROCHE(x, Galene). **(100%)**
- IF Fe2O3(x, > 0.45) AND K2O(x, <= 0.8) AND Zn(x, >36) AND Pb(x, <= 5277) THEN TYPE DE ROCHE(x, Spahlerite). **(100%)**
- IF Fe2O3(x, > 0.45) AND K2O(x, <= 0.8) AND Zn(x, <= 36) AND S(x, <= 1273) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures). **(100%)**
- IF Fe2O3(x, > 0.45) AND K2O(x, <= 0.8) AND Zn(x, <= 36) AND S(x, > 1273) AND CaO+MgO(x, >31.48) THEN TYPE DE ROCHE(x, Pyrate). **(100%)**
- IF Fe2O3(x, > 0.45) AND K2O(x, <= 0.8) AND Zn(x, <= 36) AND S(x, > 1273) AND CaO+MgO(x, <= 31.48) THEN TYPE DE ROCHE(x, Charcopyrite). **(100%)**

We have used ‘stratified k fold cross validation’ for testing the decision tree. Here we have taken k as 2 because the Orange tool will not allow k to be greater than 2 in some cases. In the **PD** dataset, the least common class has only 2 instances. So, when we try to run ‘k fold cross validation’ with k greater than 2, Orange will throw us the error “Can't run stratified k-fold cross validation; the least common class has only 2 instances”.

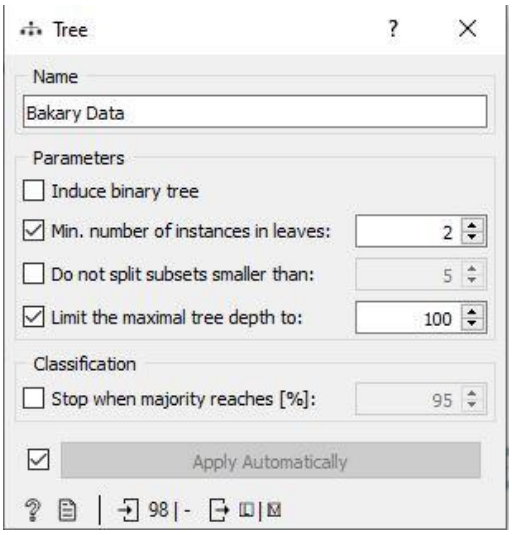
The Predictive Accuracy(Classification Accuracy) is **83.7%.**
The precision and recall for the model:

Precision - 87.0%
Recall - 83.7%



Descriptive Classifier for PED :

After performing data preprocessing on PED in step 2, it is ready to be used for building a decision tree. For the purpose of building a descriptive classifier for PED, we have used a Tree widget again in Orange.
We have used the same tree topology of the decision tree that we have used for the dataset PD.

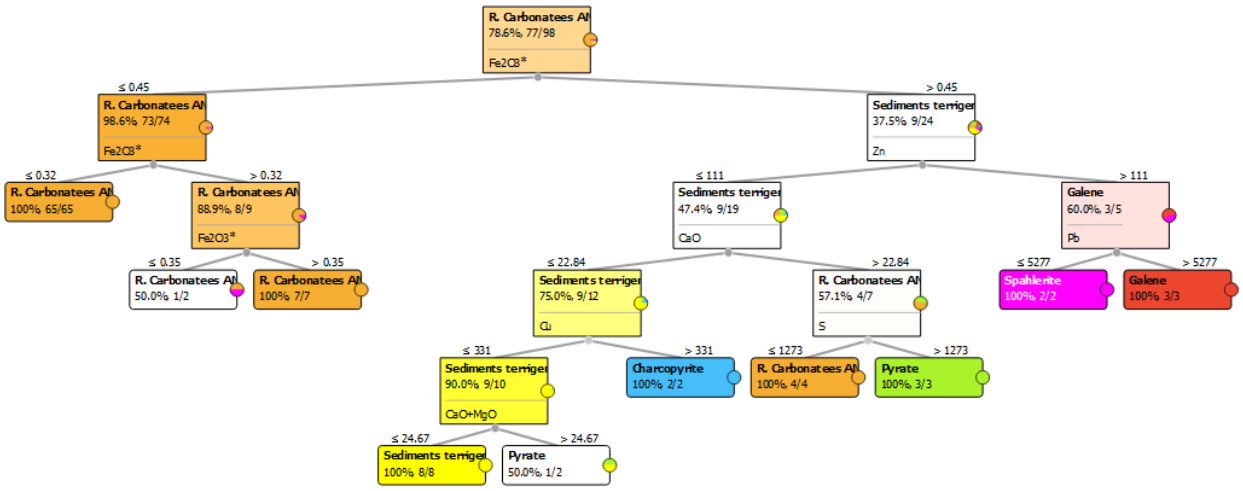


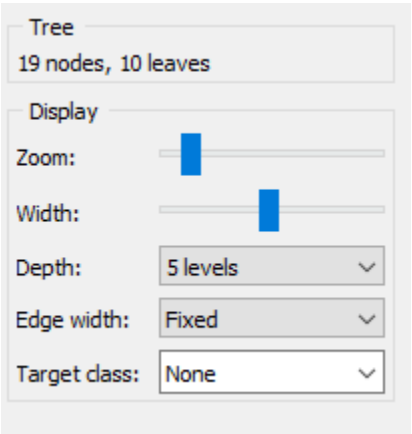
Selecting 5 levels for the tree built from **PD** dataset gave the best results by classifying all the 6 classes. Lower number levels were only able to classify few classes.

The tree structure for dataset PED:

We can also see the Rules Accuracy in each leaf node of the Tree :

True Positive / Total





<u>Discriminant rules(Predicate Form)</u>:

Rules Accuracy Mentioned on the side

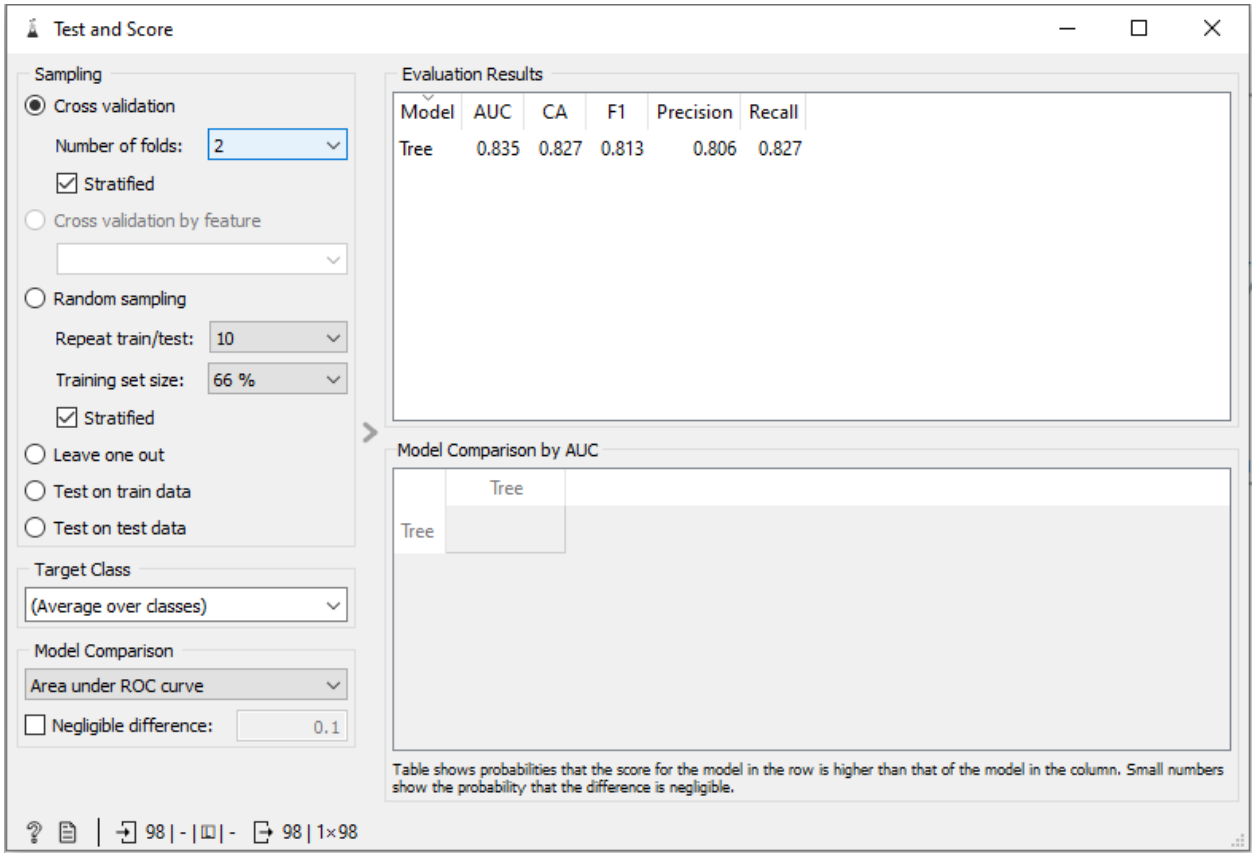
- IF Fe2O3(x, <= 0.45) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures).**(100%)**
- IF Fe2O3(x, > 0.45) AND Zn(x, > 111) AND Pb(x, >5277) THEN TYPE DE ROCHE(x, Galene). **(100%)**
- IF Fe2O3(x, > 0.45) AND Zn(x, > 111) AND Pb(x, <= 5277) THEN TYPE DE ROCHE(x, Spahlerite). **(100%)**
- IF Fe2O3(x, > 0.45) AND Zn(x, <= 111) AND CaO(x, <= 22.84) AND Cu(x, <=331) AND CaO+MgO(x, <= 24.67)THEN TYPE DE ROCHE(x, Sediments terrigenes). **(100%)**
- IF Fe2O3(x, > 0.45) AND Zn(x, <= 111) AND CaO(x, <= 22.84) AND Cu(x, <=331) AND CaO+MgO(x, > 24.67)THEN TYPE DE ROCHE(x, Pyrate). **(50%)**

- IF Fe2O3(x, > 0.45) AND Zn(x, <= 111) AND CaO(x, <= 22.84) AND Cu(x, >331) THEN TYPE DE ROCHE(x, Charcopyrite). **(100%)**
- IF Fe2O3(x, > 0.45) AND Zn(x, <= 111) AND CaO(x, > 22.84) AND S(x, <=1273) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures). **(100%)**
- IF Fe2O3(x, > 0.45) AND Zn(x, <= 111) AND CaO(x, > 22.84) AND S(x, > 1273) THEN TYPE DE ROCHE(x, Pyrate). **(100%)**

We have used 'stratified k fold cross validation' for testing the decision tree. Here we have taken k as 2 because Orange will not allow k to be greater than 2 for the PED dataset, because the least common class has only 2 instances.

The Predictive Accuracy(Classification Accuracy) is **82.7%.**
The Precision and Recall for the model :

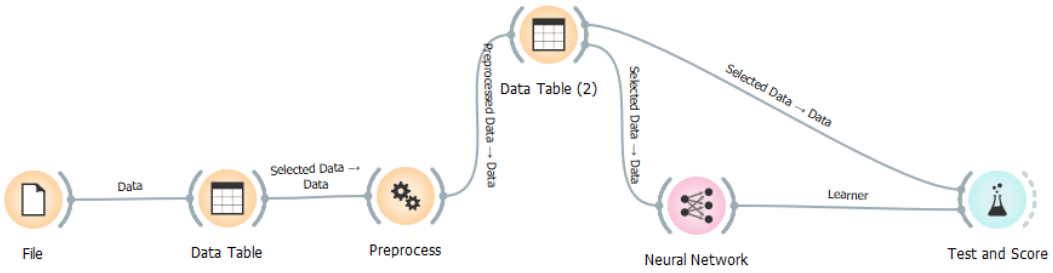
Precision - 80.6%
Recall - 82.7%



Non-Descriptive Classifier for PD :

We used the Neural Network widget for classification and used the **K-Fold** of 2 folds cross validation technique to train the classifier on training and validation data.

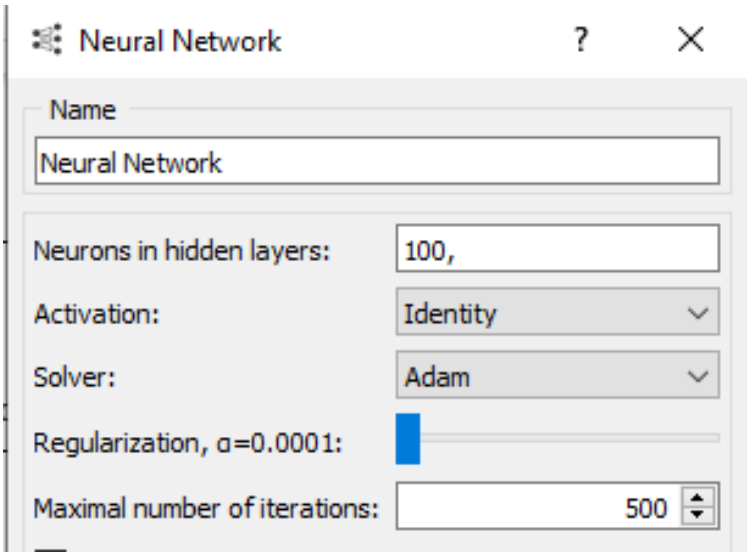
The architecture for this experiment is:



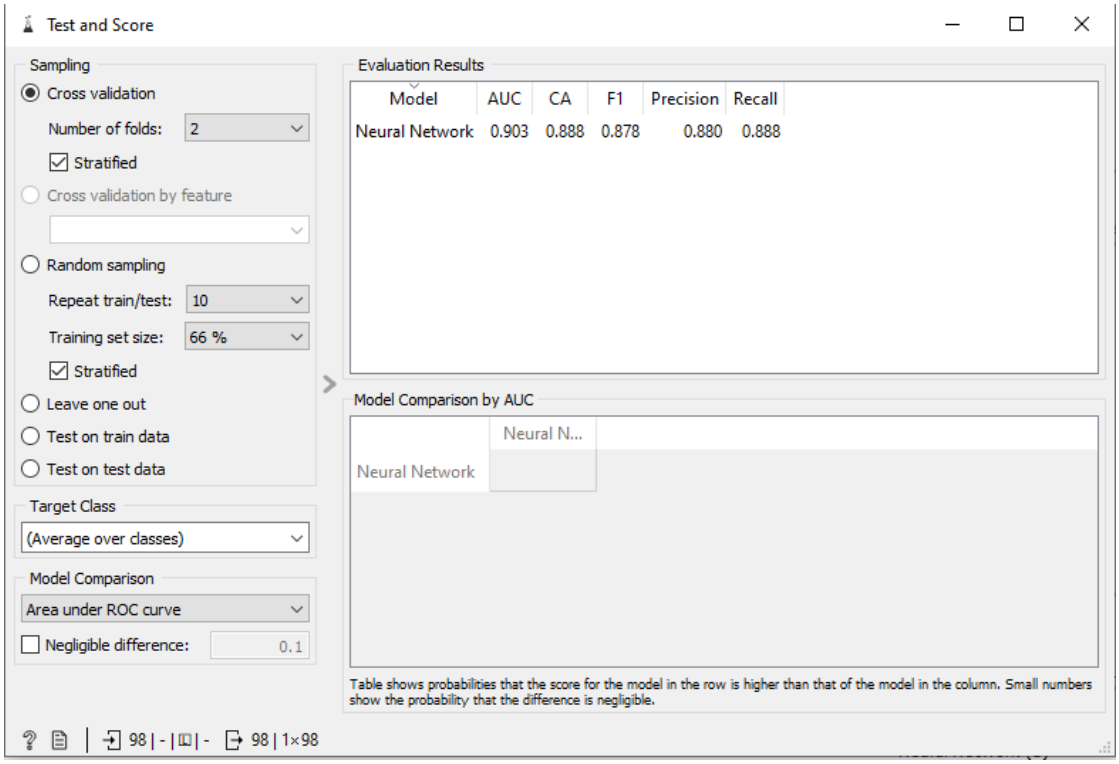
We have used different hyperparameters and the accuracies are as below.
The learning rate and momentum for a neural network in Orange tool are constant and cannot be modified.

Case 1 :

Learning rate : 0.001(constant for the tool) Momentum : 0.9(constant for the tool) Epochs : 500 Neurons in Hidden layers : 100

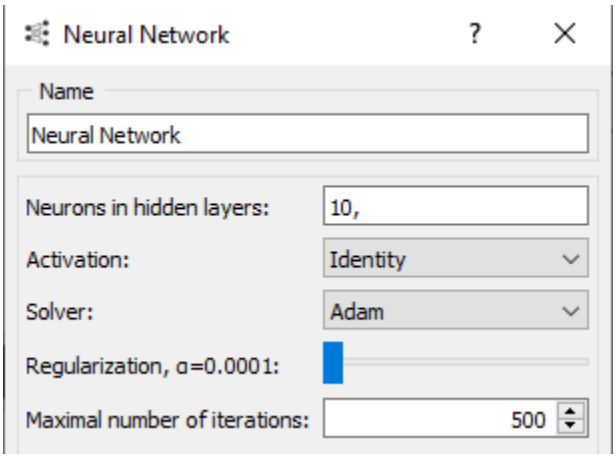


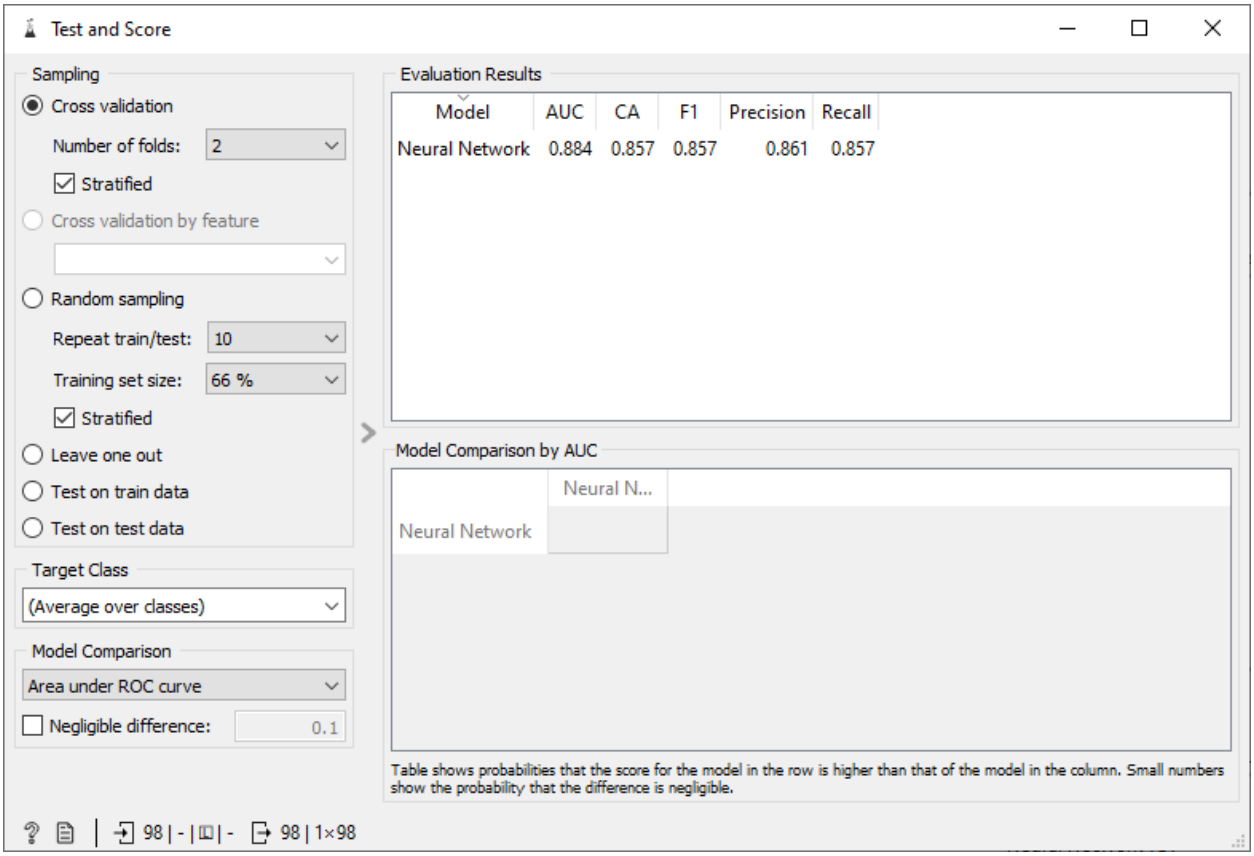
The Predictive Accuracy of this model is : **88.8 %**
Here we have taken k as 2 in 'Stratified k-fold' because Orange will not allow k to be greater than 2 for the PD dataset, because the least common class has only 2 instances.



Case 2 :

Learning rate : 0.001(constant for the tool)
Momentum : 0.9(constant for the tool)
Epochs : 500
Neurons in Hidden layers : 10



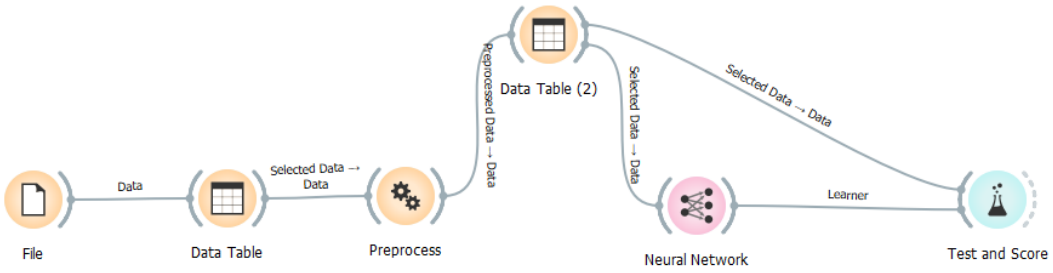


The Predictive Accuracy of this algorithm is : **85.7 %**

Non-Descriptive Classifier for PED :

We used the Neural Network widget for classification and used the **K-Fold** of 2 folds cross validation technique to train the classifier on training and validation data.

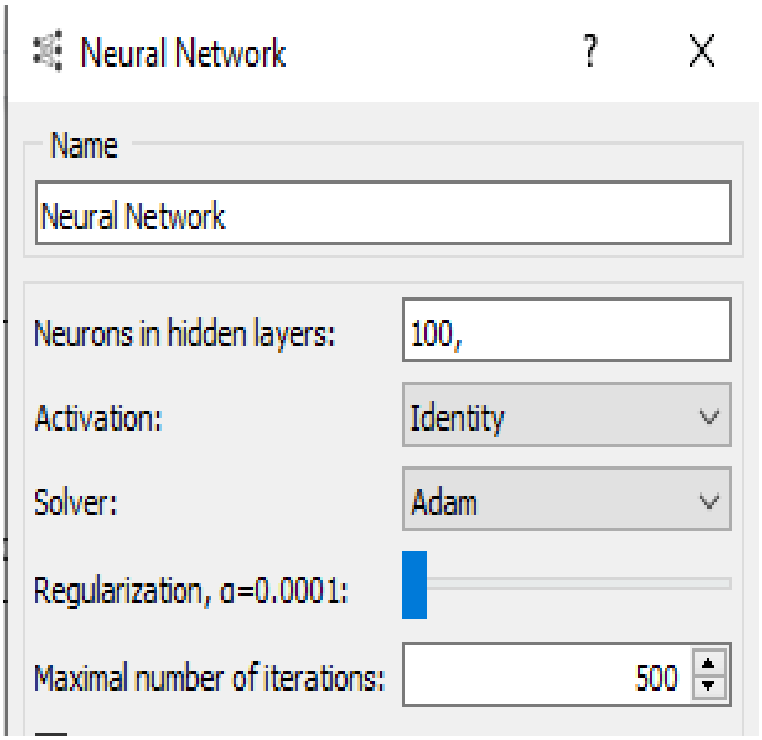
The architecture for this experiment is:



We have used different hyperparameters and the accuracies are as below.
The learning rate and momentum for a neural network in Orange tool are constant and cannot be modified.

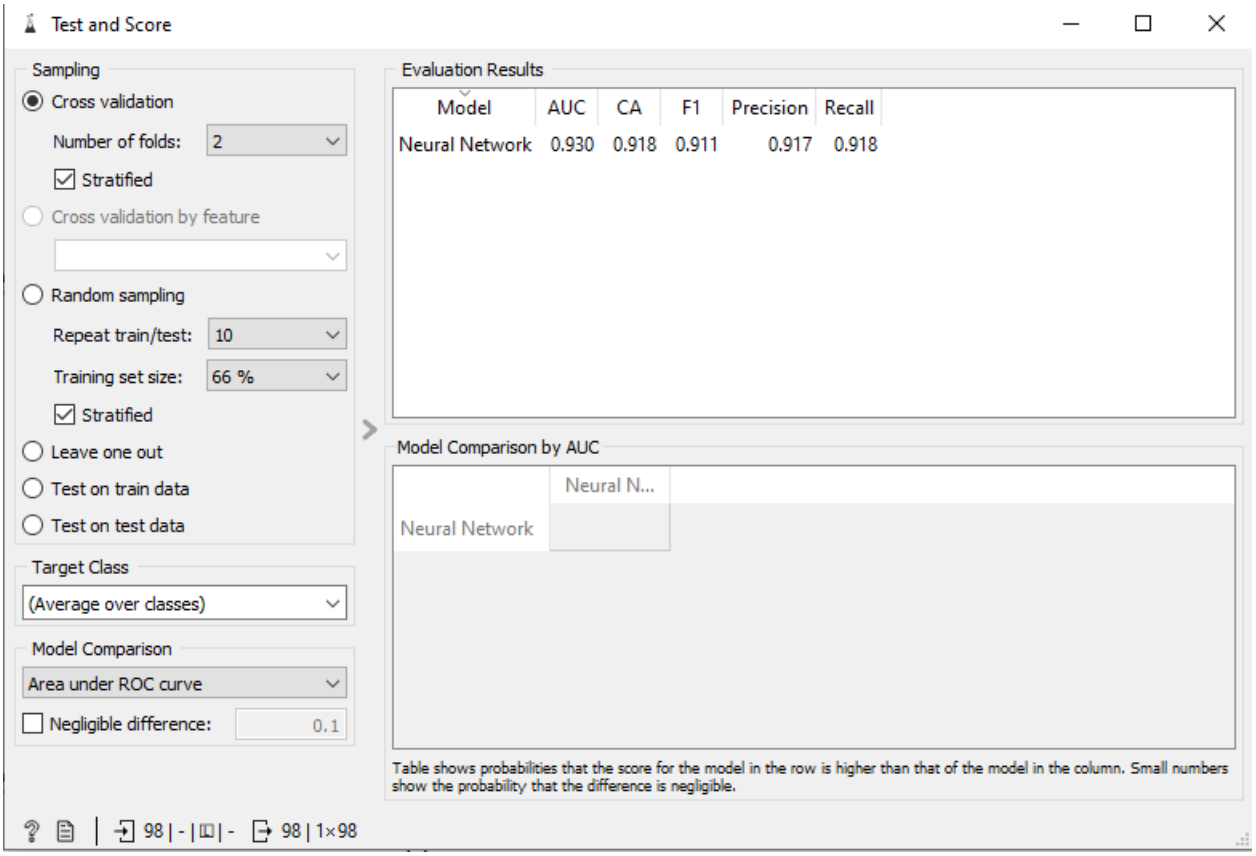
Case 1 :

Learning rate : 0.001(constant for the tool)
Momentum : 0.9(constant for the tool)
Epochs : 500
Neurons in Hidden layers : 100



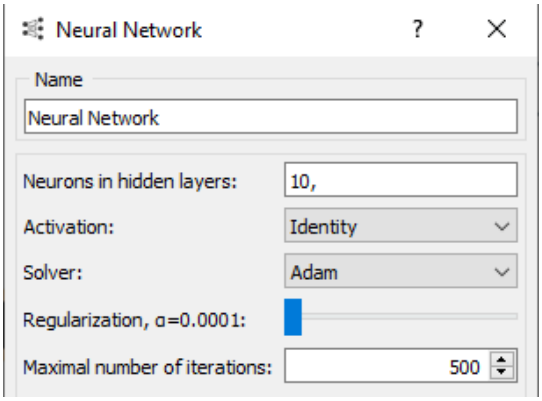
The Predictive Accuracy of this algorithm is : **91.8 %**

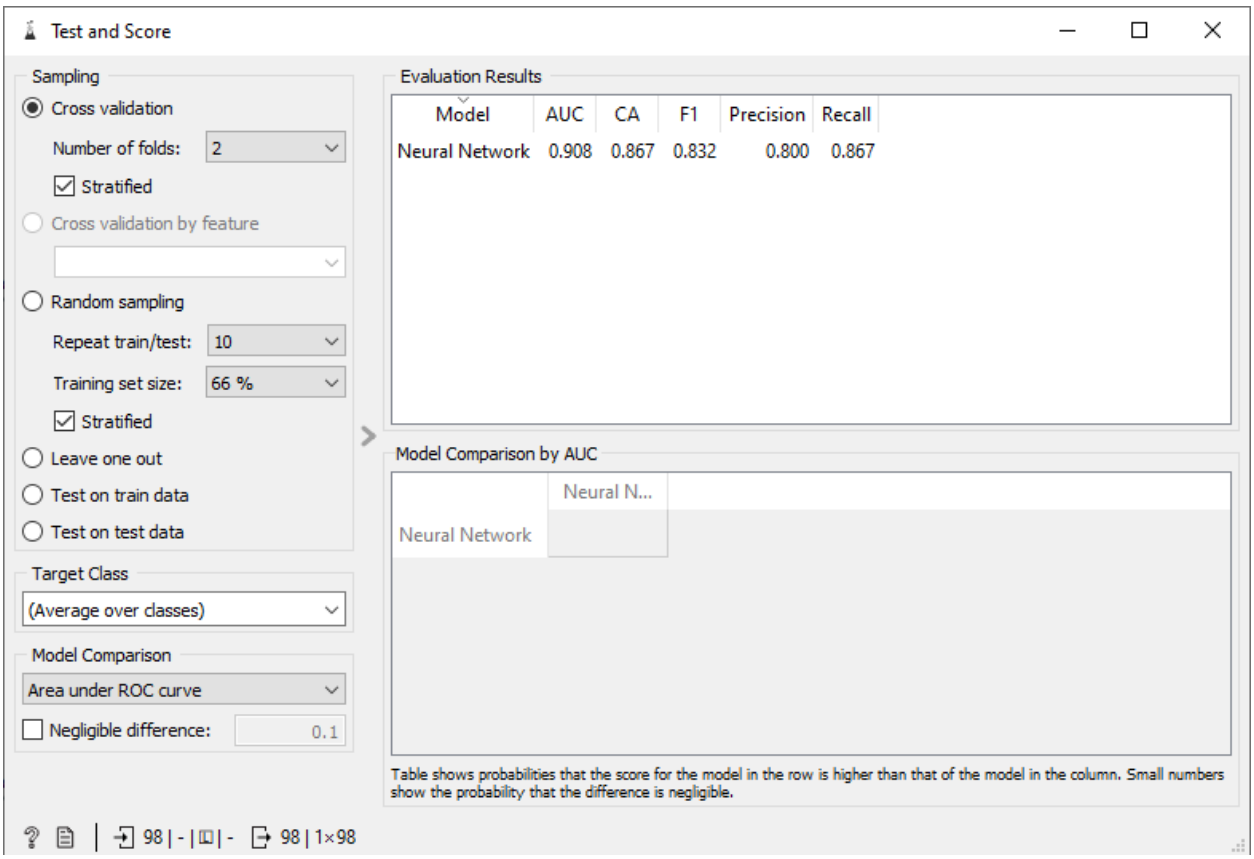
Here we have taken k as 2 in 'Stratified k-fold' because Orange will not allow k to be greater than 2 for the PED dataset, because the least common class has only 2 instances.



Case 2 :

Learning rate : 0.001(constant for the tool)
Momentum : 0.9(constant for the tool)
Epochs : 500
Neurons in Hidden layers : 10





The Predictive Accuracy of this algorithm is : **86.7 %**

Experiment 2 (Contrast Classification) :

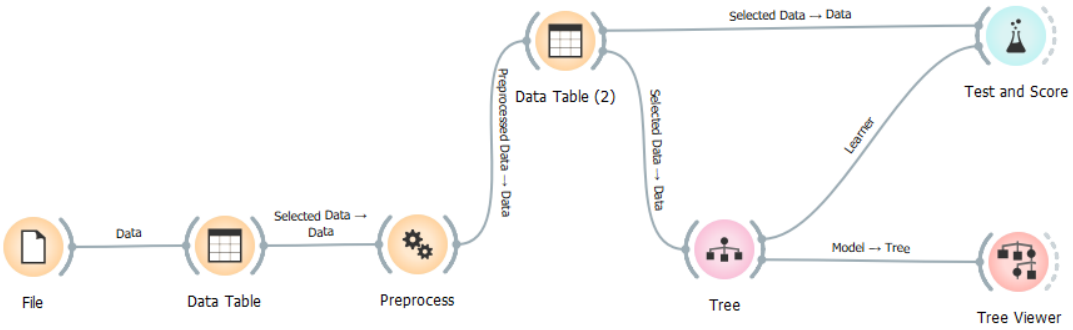
We were asked to build a Decision Tree classifier and a Neural Network classifier to perform the contrast classification for the class C1 i.e. **R. Carbonatees AND R. Carbonatees impures** using different topologies and testing methods. This also needs to be done for both PD and PED.

Descriptive Classifier for PD :

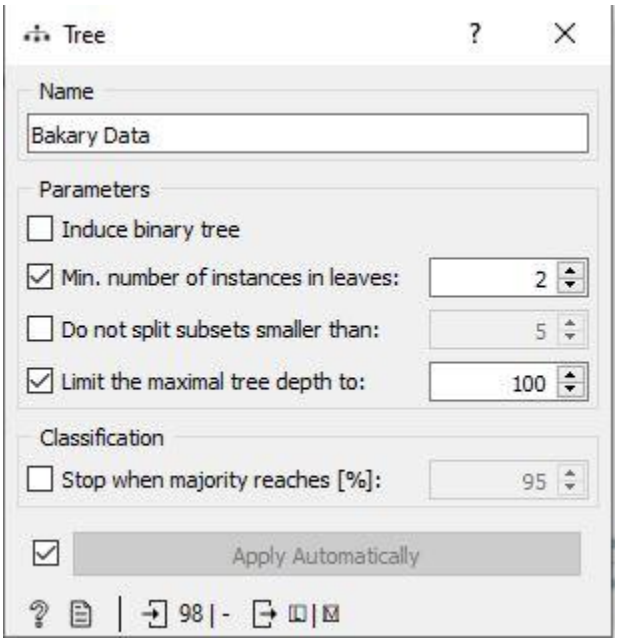
For the purpose of building a descriptive classifier for PD, we have used a Tree widget in Orange.

Tree widget of Oranges takes preprocessed data and builds the decision tree. The tree can be visualised using a **Tree Viewer**. We use **Test and Score** to get the **Classification Accuracy**(Predictive accuracy) of the decision tree.

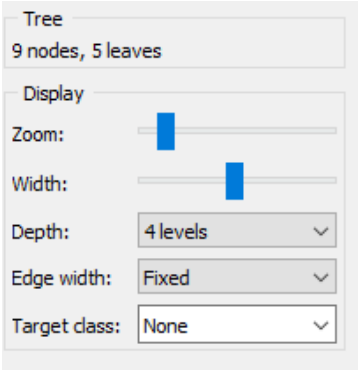
The discriminant rules can be written from the tree viewer.
The architecture for building the decision tree for this experiment is as below.



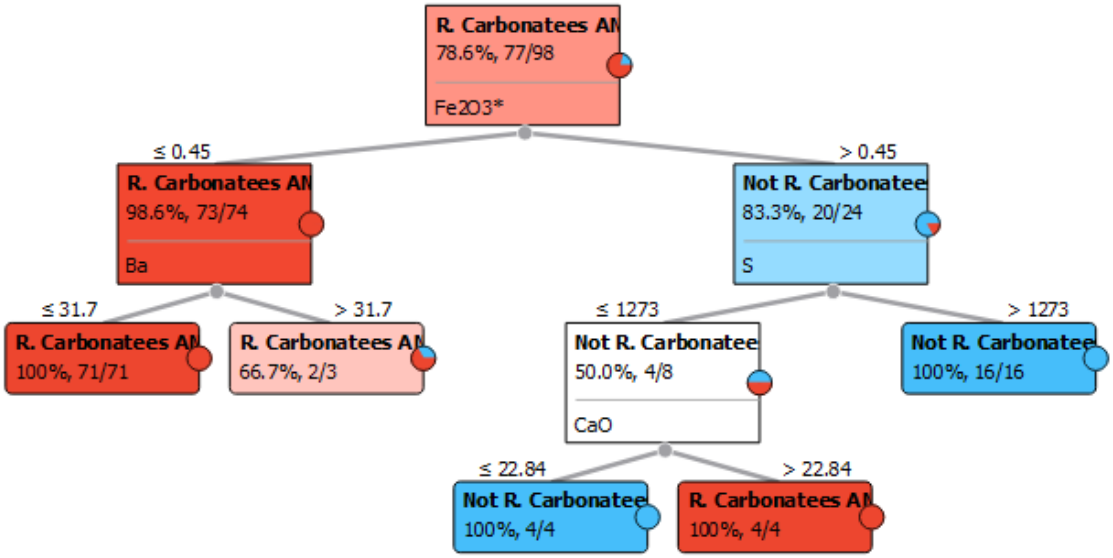
The tree topology for the experiment:



Selecting 4 levels for the tree built from **PD** dataset gave the best results by classifying 2 classes with good rules accuracy.



We can also see the Rules Accuracy in each leaf node of the Tree :
True Positive / Total

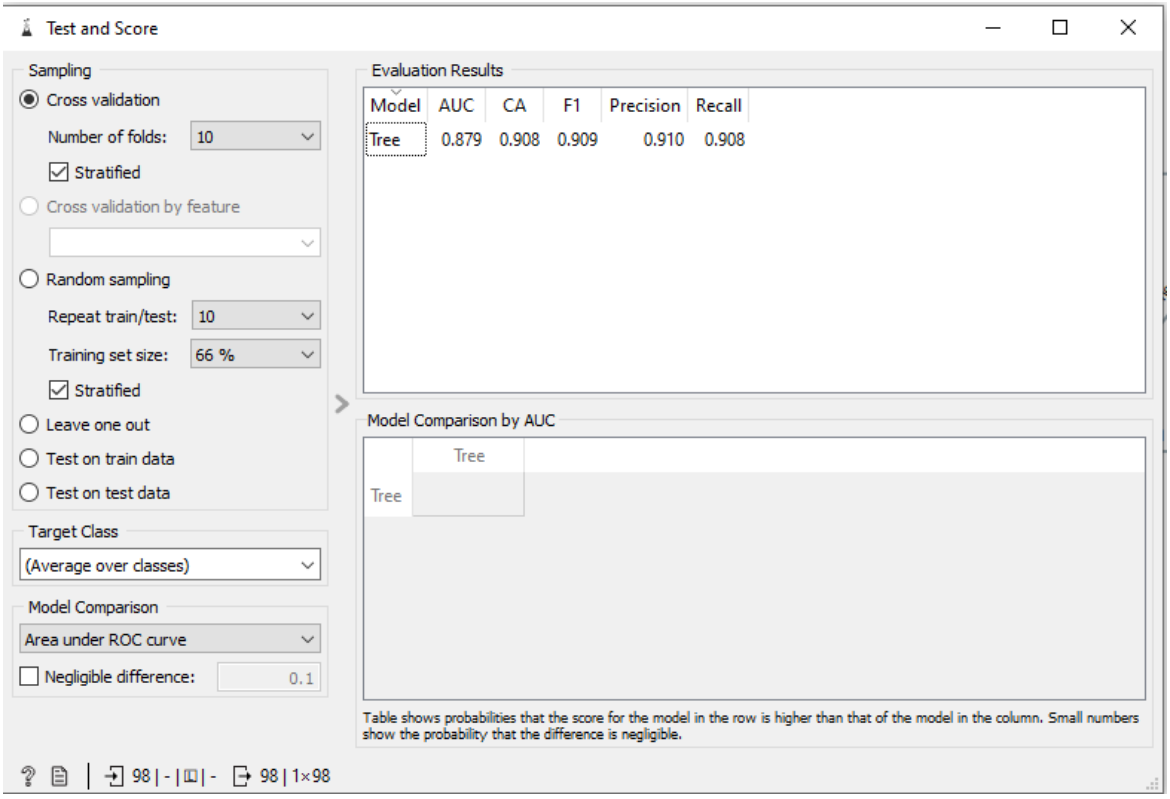


<u>Discriminant rules(Predicate Form)</u>:

Rules Accuracy Mentioned on the side

- IF Fe2O3(x, <= 0.45) AND Ba(x, <= 31.7) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures). **(100%)**
- IF Fe2O3(x, <= 0.45) AND Ba(x, > 31.7) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures). **(66%)**
- IF Fe2O3(x, > 0.45) AND S(x, <= 1273) AND CaO(x, <= 22.84) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures). **(100%)**
- IF Fe2O3(x, > 0.45) AND S(x, <= 1273) AND CaO(x, > 22.84) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures). **(100%)**
- IF Fe2O3(x, > 0.45) AND S(x, > 1273) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures). **(100%)**

We have used ‘stratified k fold cross validation’ for testing the decision tree. Here we have taken k as 10.

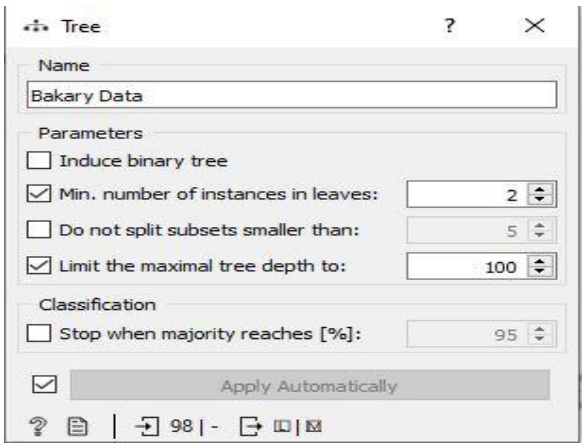


The Classification Accuracy(Predictive Accuracy) is **90.8%.**
The precision and recall for the model :

Precision - 91.0%
Recall - 90.8%

Descriptive Classifier for PED :

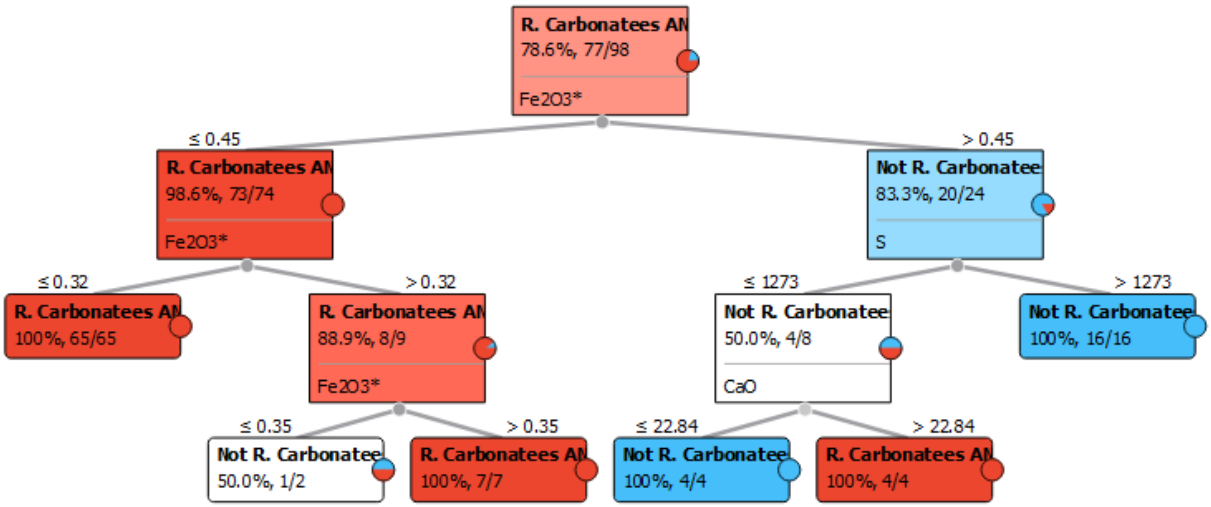
After performing data preprocessing on PED in step 2, it is ready to be used for building a decision tree. For the purpose of building a descriptive classifier for PED, we have used a Tree widget again in Orange.
We have used the same tree topology of the decision tree that we have used for the dataset PD.

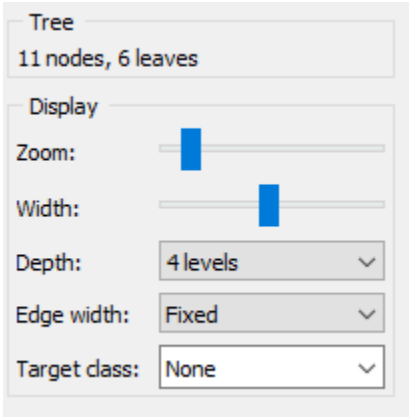


Selecting 4 levels for the tree built from **PED** dataset gave the best results by classifying 2 classes with good rules accuracy.

The tree for dataset PED:

We can also see the Rules Accuracy in each leaf node of the Tree :
True Positive / False Positive



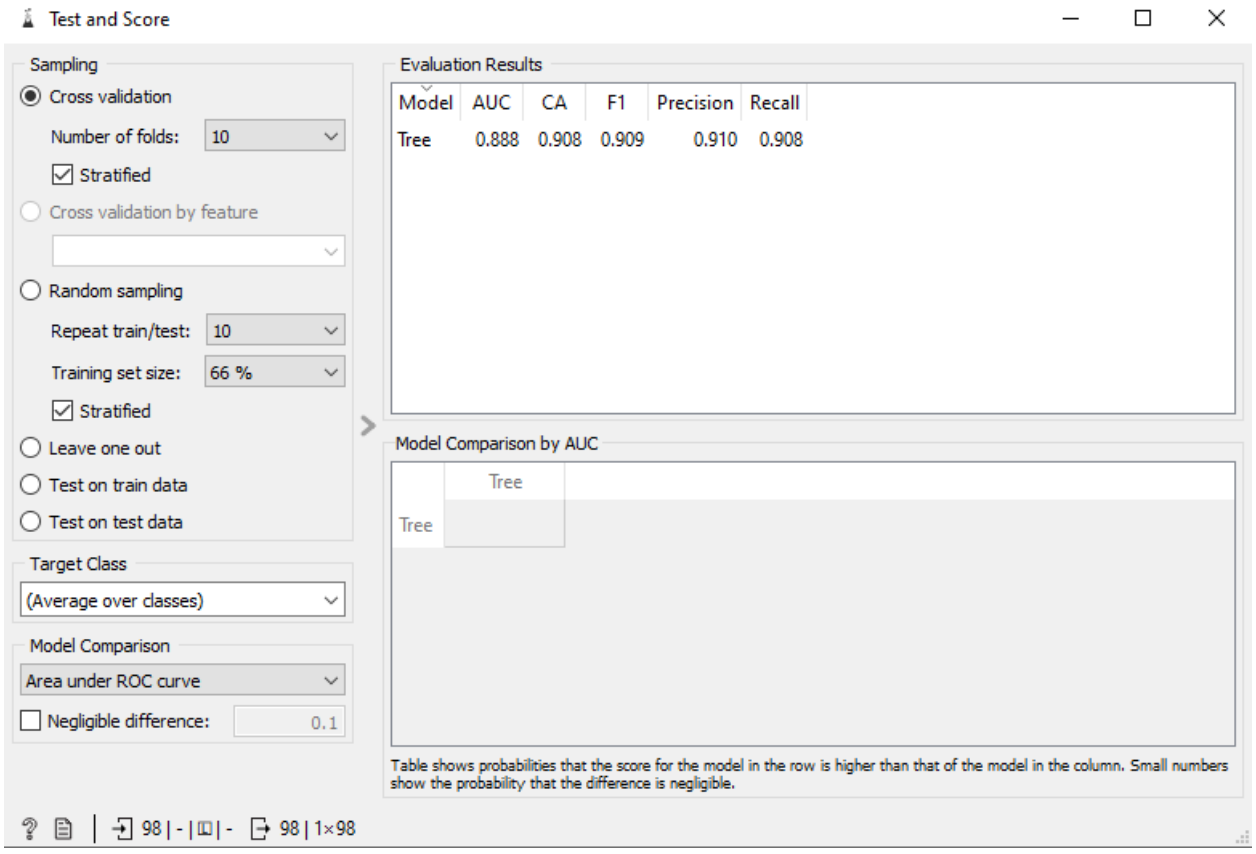


Discriminant rules(Predicate Form):

Rules Accuracy Mentioned on the side

- IF Fe2O3(x, <= 0.32) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures).**(100%)**
- IF Fe2O3(x, > 0.32) AND Fe2O3(x, <= 0.35) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures). **(50%)**
- IF Fe2O3(x, > 0.35) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures).**(100%)**
- IF Fe2O3(x, > 0.45) AND S(x, <= 1273) AND CaO(x, <=22.84) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures). **(100%)**
- IF Fe2O3(x, > 0.45) AND S(x, <= 1273) AND CaO(x, >22.84) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures). **(100%)**
- IF Fe2O3(x, > 0.45) AND S(x, > 1273) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures). **(100%)**

We have used 'stratified k fold cross validation' for testing the decision tree. Here we have taken k as 10.



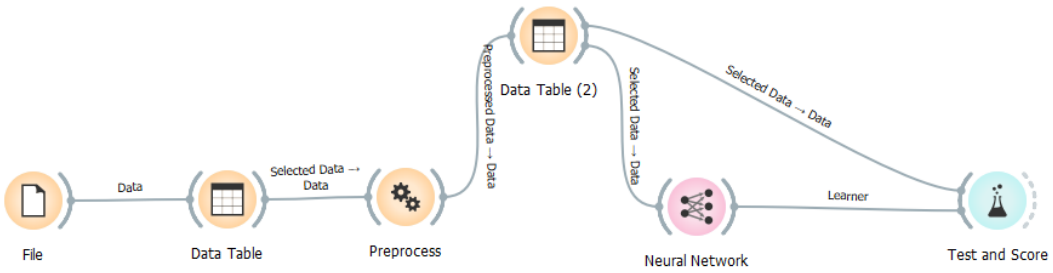
We have received the Classification Accuracy(Predictive Accuracy) as **90.8%.**
The precision and recall for the model :

Precision - 91%
Recall - 90.8%

Non-Descriptive Classifier for PD :

We used the Neural Network widget for classification and used the **K-Fold** of 10 folds cross validation technique to train the classifier on training and validation data.

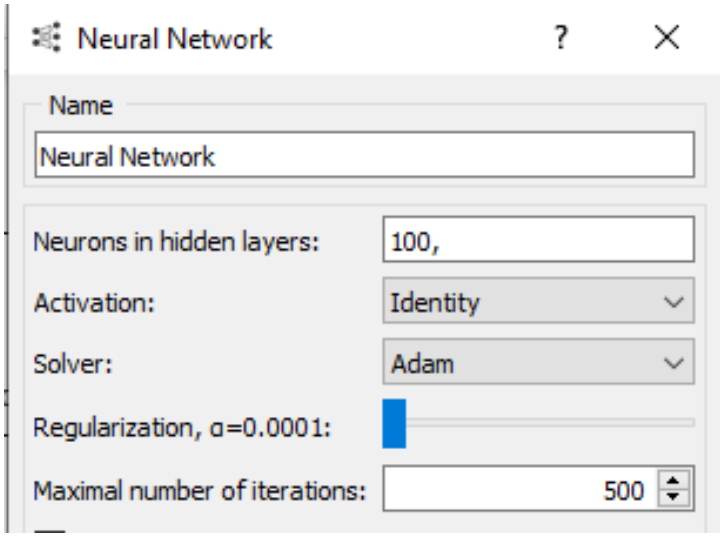
The architecture for this experiment is:

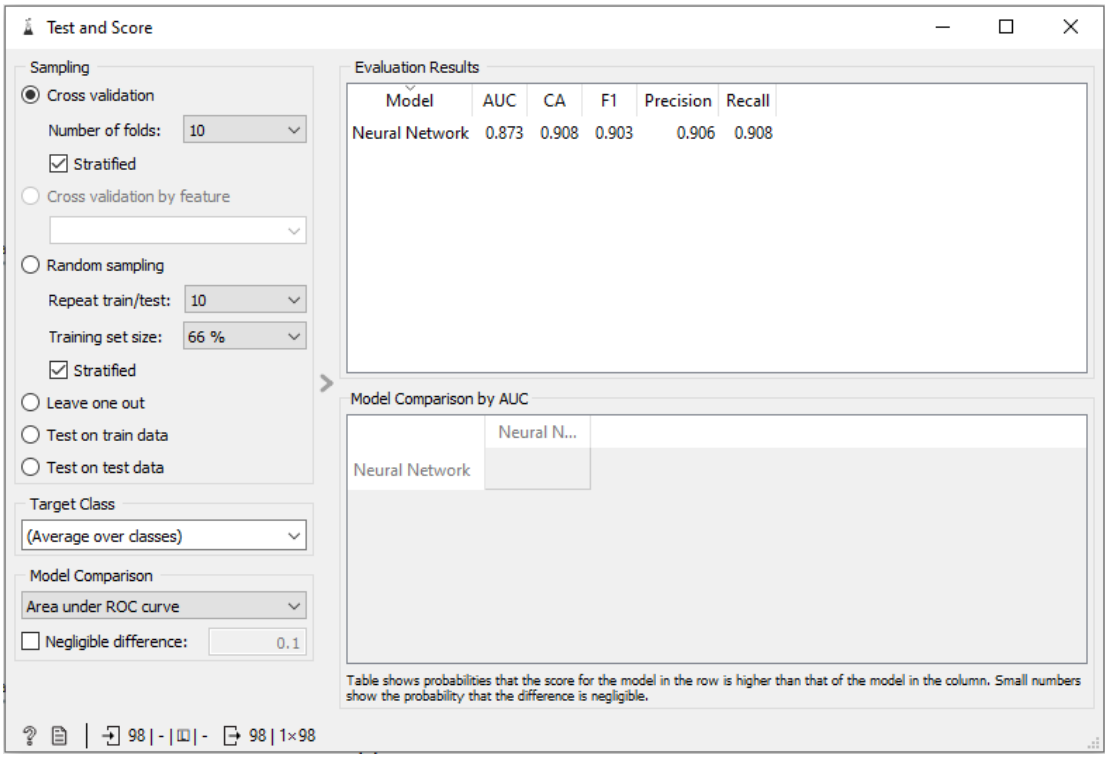


We have used different hyperparameters and the accuracies are as below.
The learning rate and momentum for a neural network in Orange tool are constant and cannot be modified.

Case 1 :

Learning rate : 0.001(constant for the tool) Momentum : 0.9(constant for the tool) Epochs : 500 Neurons in Hidden layers : 100

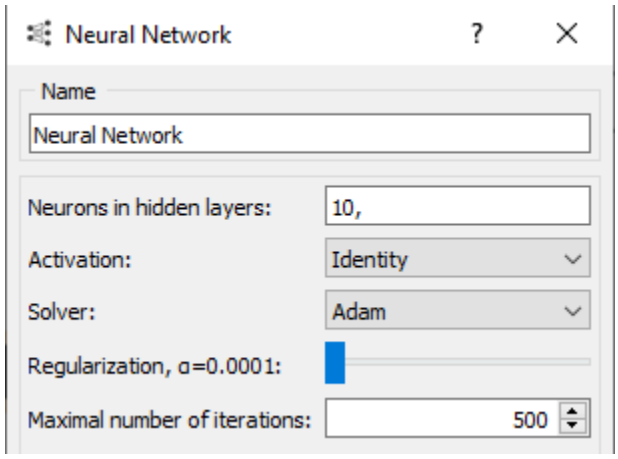


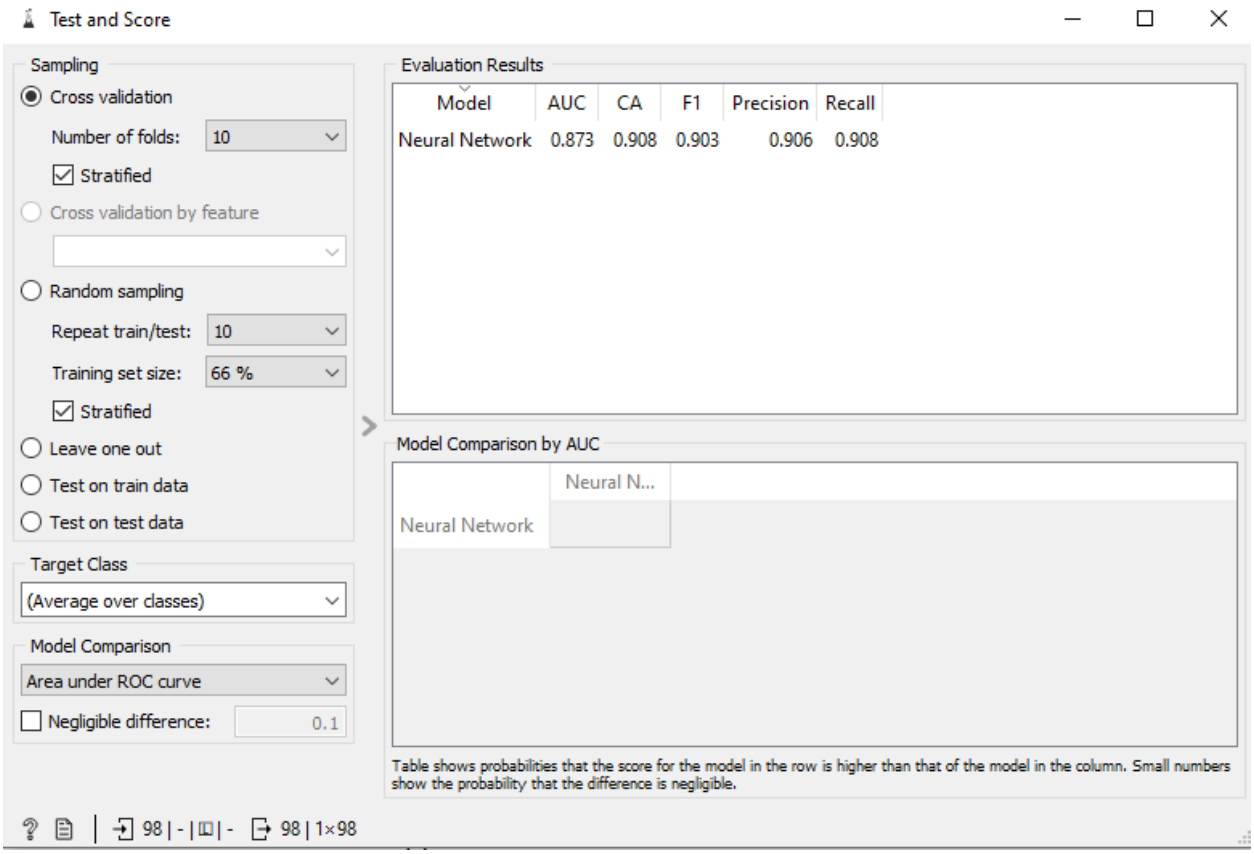


The Predictive Accuracy of the model is : **90.8 %**
Here we have used the **K-Fold** of 20 folds cross validation technique to train the classifier on training and validation data.

Case 2 :

Learning rate : 0.001(constant for the tool)
Momentum : 0.9(constant for the tool)
Epochs : 500
Neurons in Hidden layers : 10



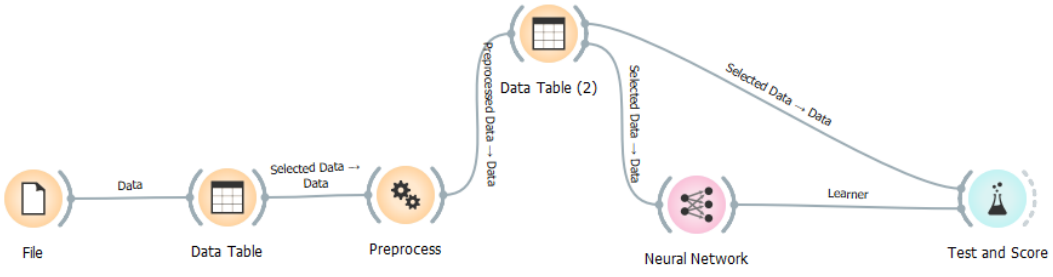


The Predictive Accuracy of this algorithm is : **90.8 %**

Non-Descriptive Classifier for PED :

We used the Neural Network widget for classification and used the **K-Fold** of 10 folds cross validation technique to train the classifier on training and validation data.

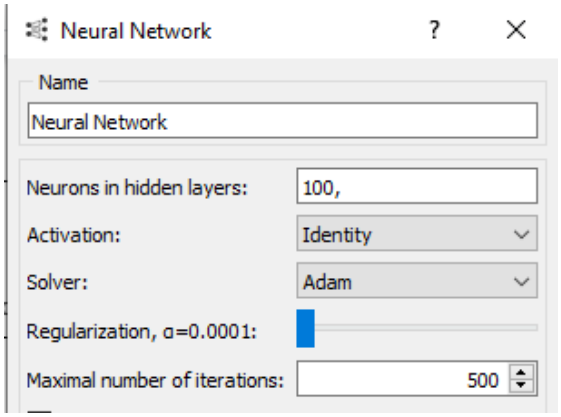
The architecture for this experiment is:



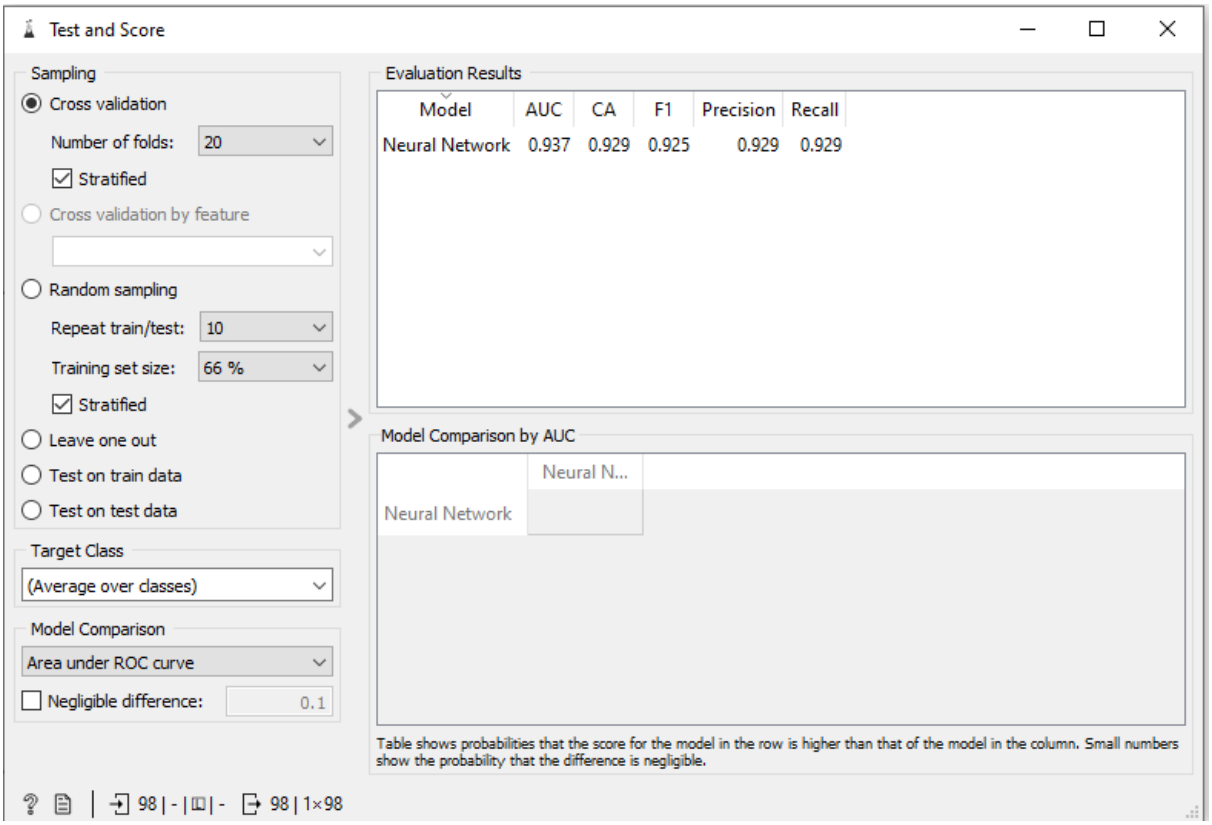
We have used different hyperparameters and the accuracies are as below.
The learning rate and momentum for a neural network in Orange tool are constant and cannot be modified.

Case 1 :

Learning rate : 0.001(constant for the tool)
Momentum : 0.9(constant for the tool)
Epochs : 500
Neurons in Hidden layers : 100



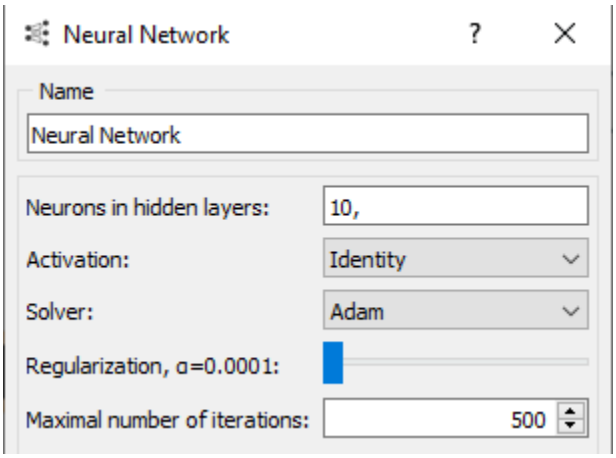
Here we have used the **K-Fold** of 20 folds cross validation technique to train the classifier on training and validation data.

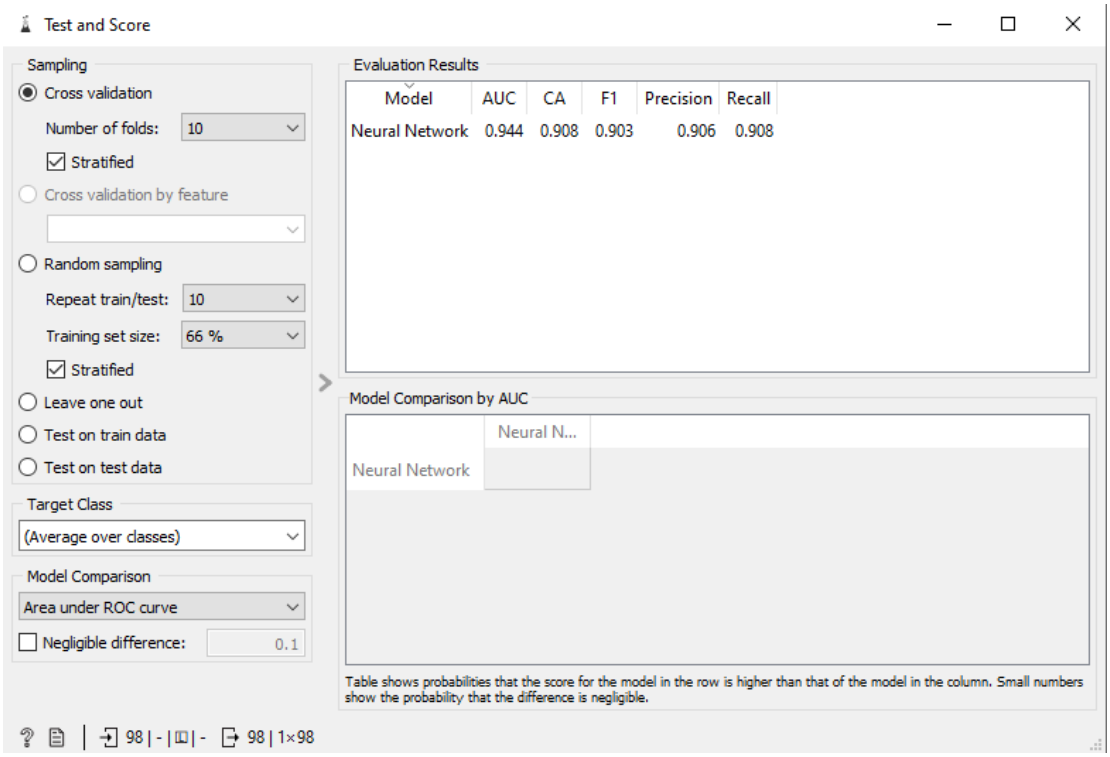


The Predictive Accuracy of this algorithm is : **92.9 %**

Case 2 :

Learning rate : 0.001(constant for the tool)
Momentum : 0.9(constant for the tool)
Epochs : 500
Neurons in Hidden layers : 10





The Predictive Accuracy of the model is : **90.8 %**

SUMMARY

Predictive Accuracies for Descriptive Classifiers :

PD - Complete	83.7
PED - Complete	82.7
PD - Contrast	90.8
PED - Contrast	90.8

Here we have used the built in tree model in the Orange tool along with either 10 Fold Validation or 2 fold cross validation for both PD and PED (both Complete and Contrast classifiers). We see, the classifier has worked better for a complete data set than just with the important attributes when we tried to do Full classification. On the other hand, In case of contrast classification the accuracy for PD and PED remained the same.

The Rules Accuracies have been mentioned as part of the classifiers description above. Where as the attributes **Zn**, **S**, **Cu**, **Pb**, **CaO, CaO+MgO** and **Fe2O3*** seem to be common across both the complete classification decision trees. Where as **Fe2O3*** seem to be common for contrast classification.

Apart from these, the decision trees are much more compact for contrast classification compared to complete classifiers.

Predictive Accuracies for Non - Descriptive Classifiers :

Complete classification:

PD - Case 1	88.8
PD - Case 2	85.7
PED - Case 1	91.8
PED - Case 2	86.7

Contrast classification:

PD - Case 1	90.8

PD - Case 2	90.8
PED - Case 1	92.9
PED - Case 2	90.8

Here we used a Neural Network algorithm along with either 10 Fold cross validation or 2 Fold cross validation or 20 Fold cross validation. We can see the accuracies are higher for the classifiers where we have selected 100 hidden layers and used 20 fold cross validation for testing, when compared to the default settings. Along with that the accuracies are higher for contrast classifiers compared to the complete classification.

Highest Predictive Accuracies across all Classifiers :

Complete Classification	91.8	Non Descriptive Classifier when ran on PED data
Contrast Classification	92.9	Non Descriptive Classifier when ran on PED data

RAPIDMINER

Introduction about the tool :

RapidMiner is an open source data science software platform that provides an integrated environment for data preprocessing and visualization, predictive analytics and statistical modeling, evaluation, and deployment. RapidMiner is written in the Java programming language. This tool provides a GUI to design and execute analytical workflows, which are called processes. These processes consist of multiple “operators”. Each operator performs a single task within the process, and the output of each operator forms the input of the next one. RapidMiner is developed on an open core model and has cross-platform support.

We have chosen RapidMiner as our third tool for this project. As per the given requirements, We have performed Experiments 1 and 2 using **RapidMiner** Tool on both the datasets **PD and PED.** We have used the same cleaned csv files of both data sets **PD** and **PED** , that were earlier generated in the Data Preparation step for Weka.

S2 : DATA PREPROCESSING

1) **Descriptive Classifiers :**

Firstly we have uploaded the cleaned csv files into the data folder of RapidMiner tool. We then opened the csv files inside the “Turbo Prep” Model Widget. Under this feature, we can perform pre-processing of data like missing values, binning, merging etc. For our data, we have observed a lot of missing values inside the csv files. So we have populated missing values by using the ‘**Replace Missing Values**’ option in the preprocess window. Here we have selected the ‘**Average**’ method to fill the missing values. We then exported the newly generated csvs into the data folder, which can later be used for building classifiers.

2) **Non Descriptive Classifiers :**

We were asked to normalize the data for classification. For this, we have selected the “**Normalise**” Operator in the Preprocess window . This uses the **“Z-Transform”** technique to normalise the given data.

S3 : BUILDING CLASSIFIERS

Experiment 1 (Full Classification) :

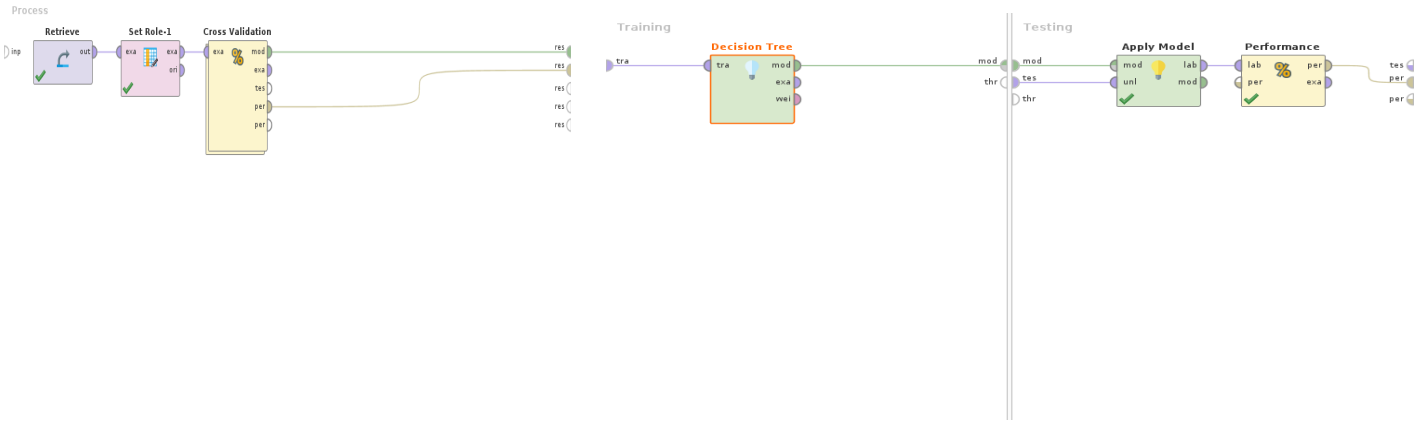
We were asked to build a Decision Tree classifier and a Neural Network classifier for all classes C1-C6 simultaneously using different topologies and testing methods. This needs to be done for both PD and PED.

Descriptive Classifier for PD :

We used “Cross Validation” and “Decision Tree” Operators to build this classifier. The Cross Validation Operator is a nested Operator. It has two subprocesses: a Training subprocess and a Testing subprocess. The Training subprocess is used for training a model.The trained model is then applied in the Testing subprocess. The Training subprocess is connected to Decision Tree Operator and the Testing subprocess is connected to “Performance Operator” to measure the predictive accuracy and also to obtain the performance vector .

Parameters :

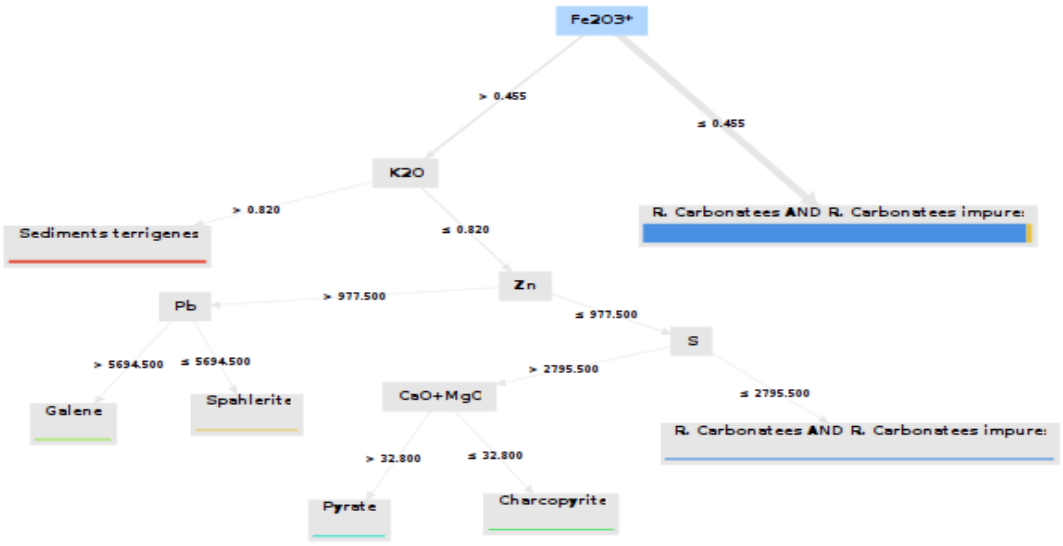
Cross Validation Operator :
number of folds = 10
Decision Tree Operator :
criterion = information_gain
maximum depth = 10
Apply_pruning = true



The decision tree obtained is :
(Note : Rule Accuracy is mentioned in the brackets)

Fe2O3* > 0.455
| K2O > 0.820: Sediments terrigenes **(9%)**
| K2O ≤ 0.820
| | Zn > 977.500
| | | Pb > 5694.500: Galene **(3%)**
| | | Pb ≤ 5694.500: Spahlerite **(2%)**
| | Zn ≤ 977.500
| | | S > 2795.500
| | | | CaO+MgO > 32.800: Pyrate **(4%)**
| | | | CaO+MgO ≤ 32.800: Charcopyrite **(2%)**
| | | S ≤ 2795.500: R. Carbonatees AND R. Carbonatees impures **(4%)**
Fe2O3* ≤ 0.455: R. Carbonatees AND R. Carbonatees impures **(73%)**

Graphical Representation of Decision Tree :

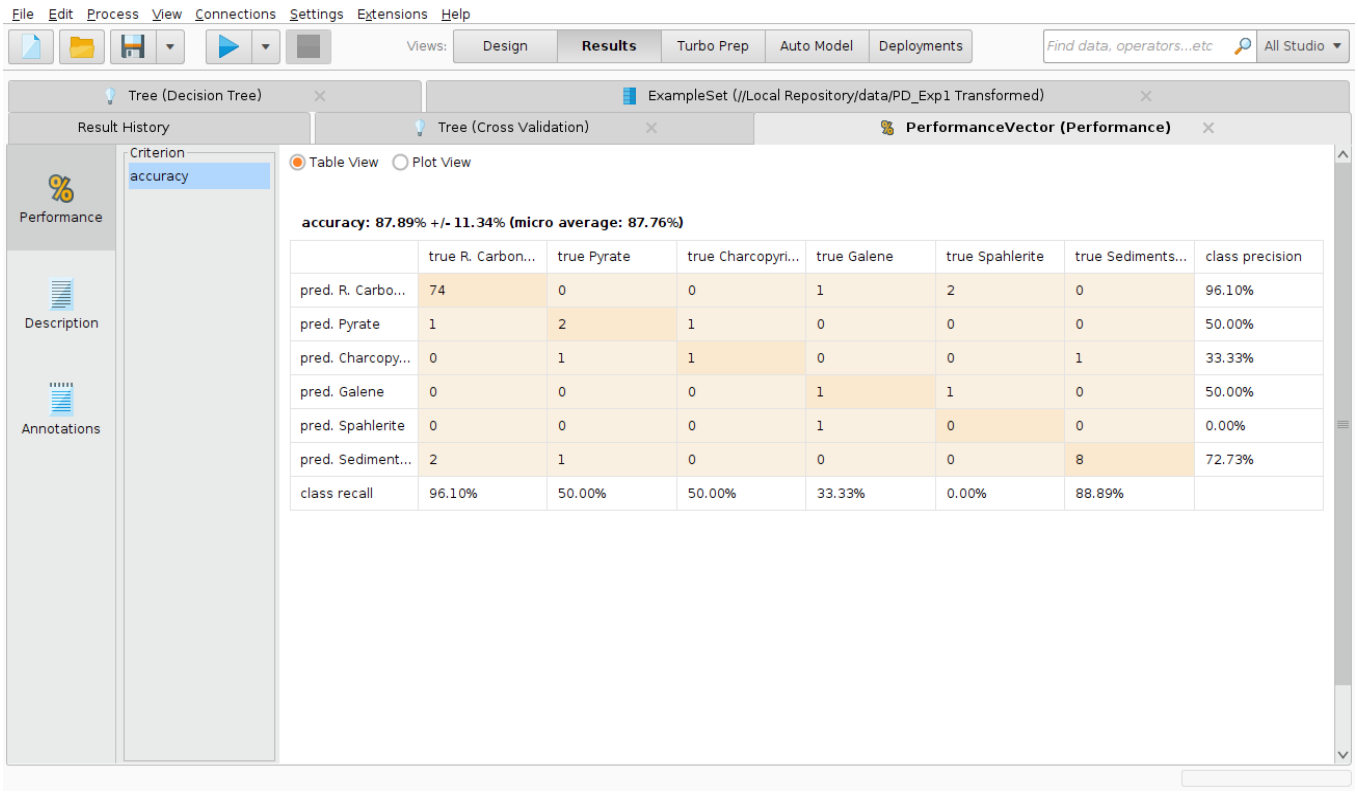


<u>Discriminant Rules (Predicate Form) :</u>

- IF Fe2O3*(x, > 0.455) AND K2O(x, > 0.820) THEN TYPE DE ROCHE(x, Sediments terrigenes)
- IF Fe2O3*(x, > 0.455) AND K2O(x, <= 0.820) AND Zn(x, > 977.500) AND Pb(x, > 5694.500) THEN TYPE DE ROCHE(x, Galene)
- IF Fe2O3*(x, > 0.455) AND K2O(x, <= 0.820) AND Zn(x, > 977.500) AND Pb(x, <= 5694.500) THEN TYPE DE ROCHE(x, Spahlerite)
- IF Fe2O3*(x, > 0.455) AND K2O(x, <= 0.820) AND Zn(x, <= 977.500) AND S(x, > 2795.500) AND CaO+MgO(x, > 32.800) THEN TYPE DE ROCHE(x, Pyrate)
- IF Fe2O3*(x, > 0.455) AND K2O(x, <= 0.820) AND Zn(x, <= 977.500) AND S(x, > 2795.500) AND CaO+MgO(x, <= 32.800) THEN TYPE DE ROCHE(x, Charcopyrite)
- IF Fe2O3*(x, > 0.455) AND K2O(x, <= 0.820) AND Zn(x, <= 977.500) AND S(x, <= 2795.500) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)
- IF Fe2O3*(x, <= 0.455) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)

The accuracy obtained for the above is **87.89% +/- 11.34% (micro average: 87.76%).**

Performance Vector :



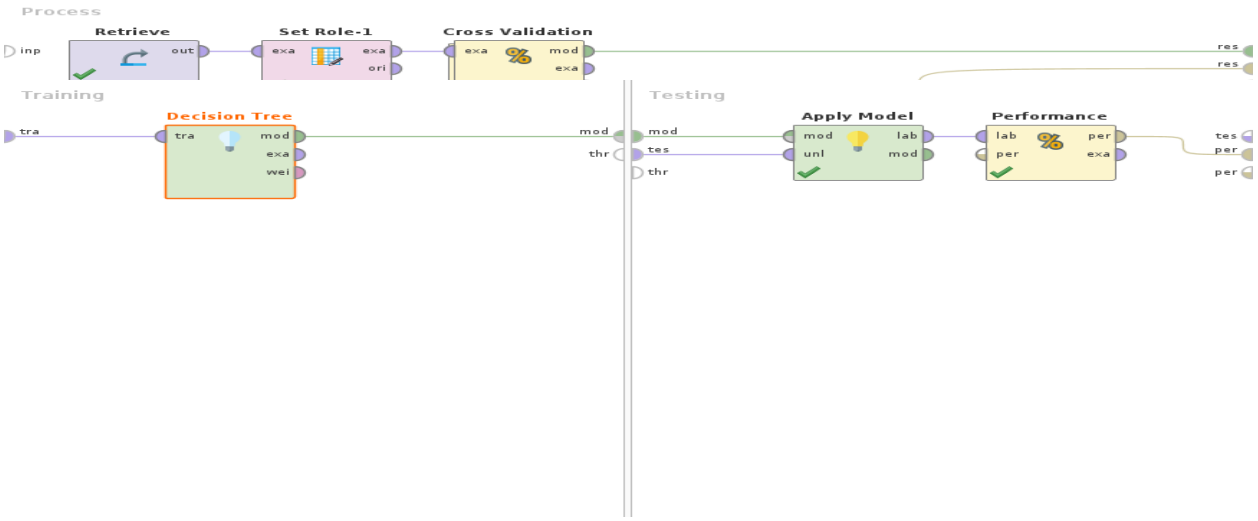
	true R. Carbon...	true Pyrate	true Charcopyri...	true Galene	true Spahlerite	true Sediments...	class precision
pred. R. Carbo...	74	0	0	1	2	0	96.10%
pred. Pyrate	1	2	1	0	0	0	50.00%
pred. Charcopy...	0	1	1	0	0	1	33.33%
pred. Galene	0	0	0	1	1	0	50.00%
pred. Spahlerite	0	0	0	1	0	0	0.00%
pred. Sediment...	2	1	0	0	0	8	72.73%
class recall	96.10%	50.00%	50.00%	33.33%	0.00%	88.89%	

Descriptive Classifier for PED :

We used “Cross Validation” and “Decision Tree” Operators to build this classifier. The Cross Validation Operator is a nested Operator. It has two subprocesses: a Training subprocess and a Testing subprocess. The Training subprocess is used for training a model.The trained model is then applied in the Testing subprocess. The Training subprocess is connected to Decision Tree Operator and the Testing subprocess is connected to “Performance Operator” to measure the predictive accuracy and also to obtain the performance vector .

Parameters :

Cross Validation Operator :
number of folds = 10
Decision Tree Operator :
criterion = information_gain
maximum depth = 10
Apply_pruning = true

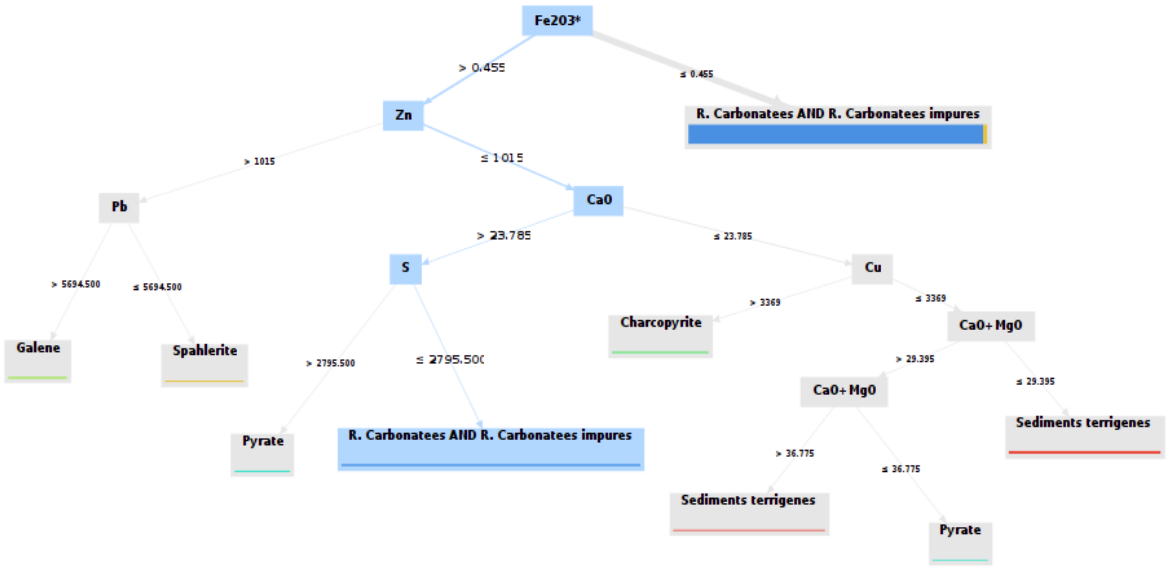


The decision tree obtained is :

(Note : Rule Accuracy is mentioned in the brackets)

Fe2O3* > 0.455
| Zn > 1015
| | Pb > 5694.500: Galene **(3%)**
| | Pb ≤ 5694.500: Spahlerite **(2%)**
| Zn ≤ 1015
| | CaO > 23.785
| | | S > 2795.500: Pyrate **(3%)**
| | | S ≤ 2795.500: R. Carbonatees AND R. Carbonatees impures **(4%)**
| | CaO ≤ 23.785
| | | Cu > 3369: Charcopyrite **(2%)**
| | | Cu ≤ 3369
| | | | CaO+MgO > 29.395
| | | | | CaO+MgO > 36.775: Sediments terrigenes **(1%)**
| | | | | CaO+MgO ≤ 36.775: Pyrate **(1%)**
| | | | CaO+MgO ≤ 29.395: Sediments terrigenes **(8%)**
Fe2O3* ≤ 0.455: R. Carbonatees AND R. Carbonatees impures **(73%)**

Graphical Representation of Decision Tree :

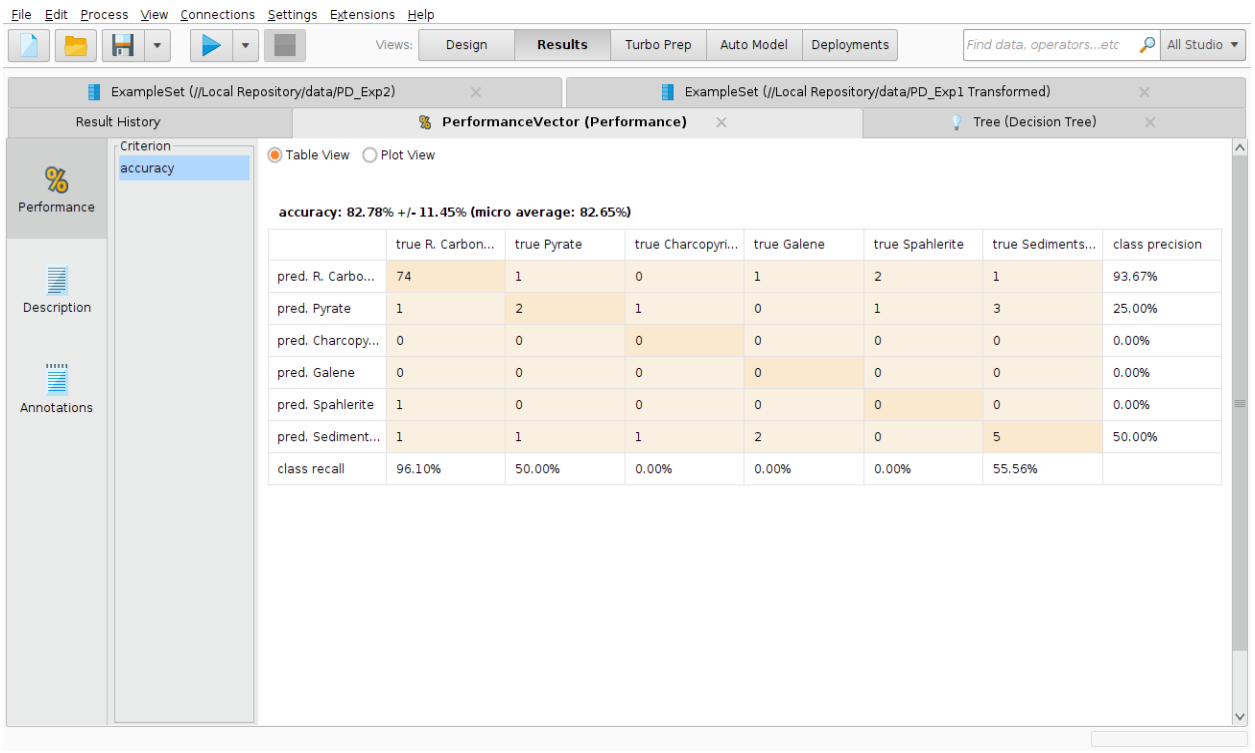


<u>Discriminant Rules (Predicate Form) :</u>

- IF Fe2O3*(x, > 0.455) AND Zn(x, > 1015) AND Pb(x, > 5694.500) THEN TYPE DE ROCHE(x, Galene)
- IF Fe2O3*(x, > 0.455) AND Zn(x, > 1015) AND Pb(x, >= 5694.500) THEN TYPE DE ROCHE(x, Spahlerite)
- IF Fe2O3*(x, > 0.455) AND Zn(x, <= 1015) AND CaO(x, > 23.785) AND S(x,> 2795.500) THEN TYPE DE ROCHE(x, Pyrate)
- IF Fe2O3*(x, > 0.455) AND Zn(x, <= 1015) AND CaO(x, > 23.785) AND S(x,<= 2795.500) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)
- IF Fe2O3*(x, > 0.455) AND Zn(x, <= 1015) AND CaO(x, <= 23.785) AND Cu(x, > 3369) THEN TYPE DE ROCHE(x, Charcopyrite)
- IF Fe2O3*(x, > 0.455) AND Zn(x, <= 1015) AND CaO(x, <= 23.785) AND Cu(x, <= 3369) AND CaO+MgO(x,> 29.395) AND CaO+MgO(x,> 36.775) THEN TYPE DE ROCHE(x, Sediments terrigenes)
- IF Fe2O3*(x, > 0.455) AND Zn(x, <= 1015) AND CaO(x, <= 23.785) AND Cu(x, <= 3369) AND CaO+MgO(x,> 29.395) AND CaO+MgO(x,<= 36.775) THEN TYPE DE ROCHE(x, Pyrate)
- IF Fe2O3*(x, > 0.455) AND Zn(x, <= 1015) AND CaO(x, <= 23.785) AND Cu(x, <= 3369) AND CaO+MgO(x,<= 29.395) THEN TYPE DE ROCHE(x, Sediments terrigenes)
- IF Fe2O3*(x, <= 0.455) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)

The accuracy obtained for the above is **82.78% +/- 11.45% (micro average: 82.65%)**

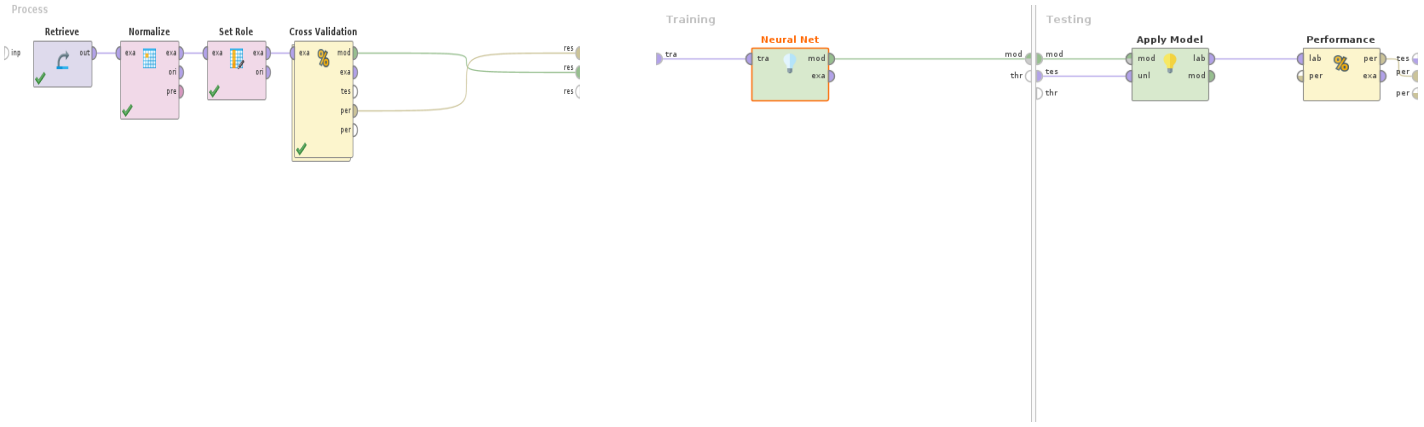
Performance Vector :



	true R. Carbon...	true Pyrate	true Charcopyri...	true Galene	true Spahlerite	true Sediments...	class precision
pred. R. Carbo...	74	1	0	1	2	1	93.67%
pred. Pyrate	1	2	1	0	1	3	25.00%
pred. Charcopy...	0	0	0	0	0	0	0.00%
pred. Galene	0	0	0	0	0	0	0.00%
pred. Spahlerite	1	0	0	0	0	0	0.00%
pred. Sediment...	1	1	1	2	0	5	50.00%
class recall	96.10%	50.00%	0.00%	0.00%	0.00%	55.56%	

Non-Descriptive Classifier for PD :

Firstly, we normalized the data using the “Normalise” Operator. We then used “Cross Validation” and “Neural Network” Operators to build this classifier.The Cross Validation Operator is a nested Operator. It has two subprocesses: a Training subprocess and a Testing subprocess. The Training subprocess is used for training a model.The trained model is then applied in the Testing subprocess. The Training subprocess is connected to Neural Network Operator and the Testing subprocess is connected to “Performance Operator” to measure the predictive accuracy and also to obtain the performance vector

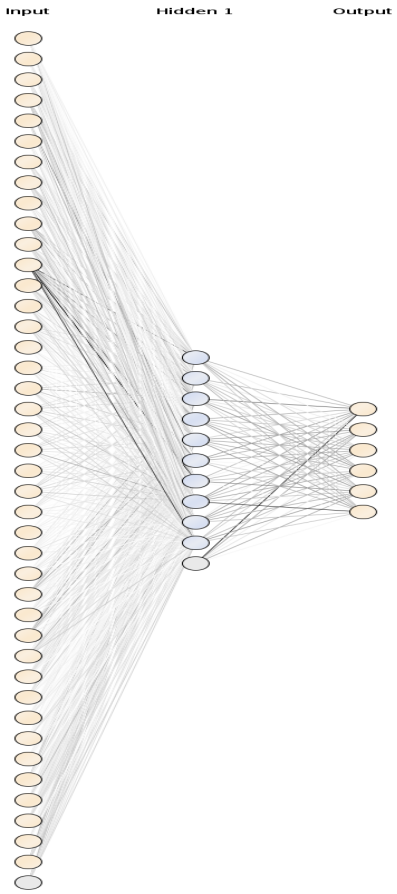


We have used different hyperparameters and the accuracies are as below.

Case 1 :

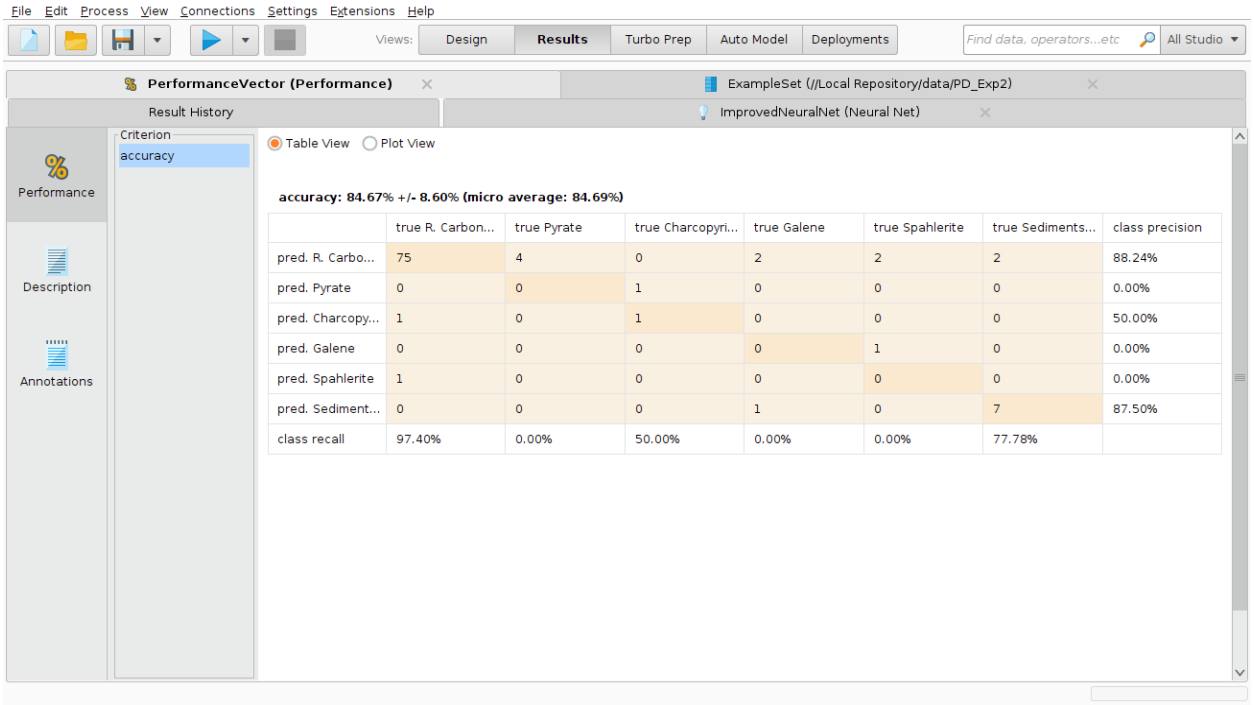
Cross Validation Operator :
number of folds = 10
Neural Network Operator :
Learning rate = 0.3
Momentum = 0.2
Epochs = 500
Number of Hidden layers = 1 (with size = 10)

Graphical representation of Nueral Network (Case 1):



The accuracy obtained for the above is **84.67% +/- 8.60% (micro average: 84.69%)**

Performance Vector :

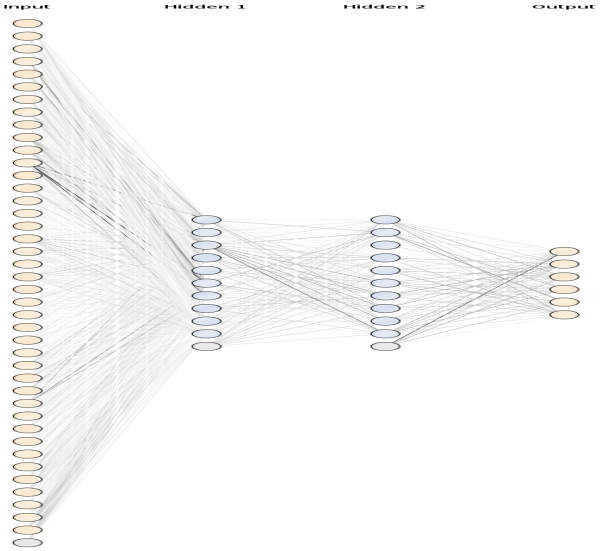


	true R. Carbon...	true Pyrate	true Charcopyri...	true Galene	true Spahlerite	true Sediments...	class precision
pred. R. Carbo...	75	4	0	2	2	2	88.24%
pred. Pyrate	0	0	1	0	0	0	0.00%
pred. Charcopy...	1	0	1	0	0	0	50.00%
pred. Galene	0	0	0	0	1	0	0.00%
pred. Spahlerite	1	0	0	0	0	0	0.00%
pred. Sediment...	0	0	0	1	0	7	87.50%
class recall	97.40%	0.00%	50.00%	0.00%	0.00%	77.78%	

Case 2 :

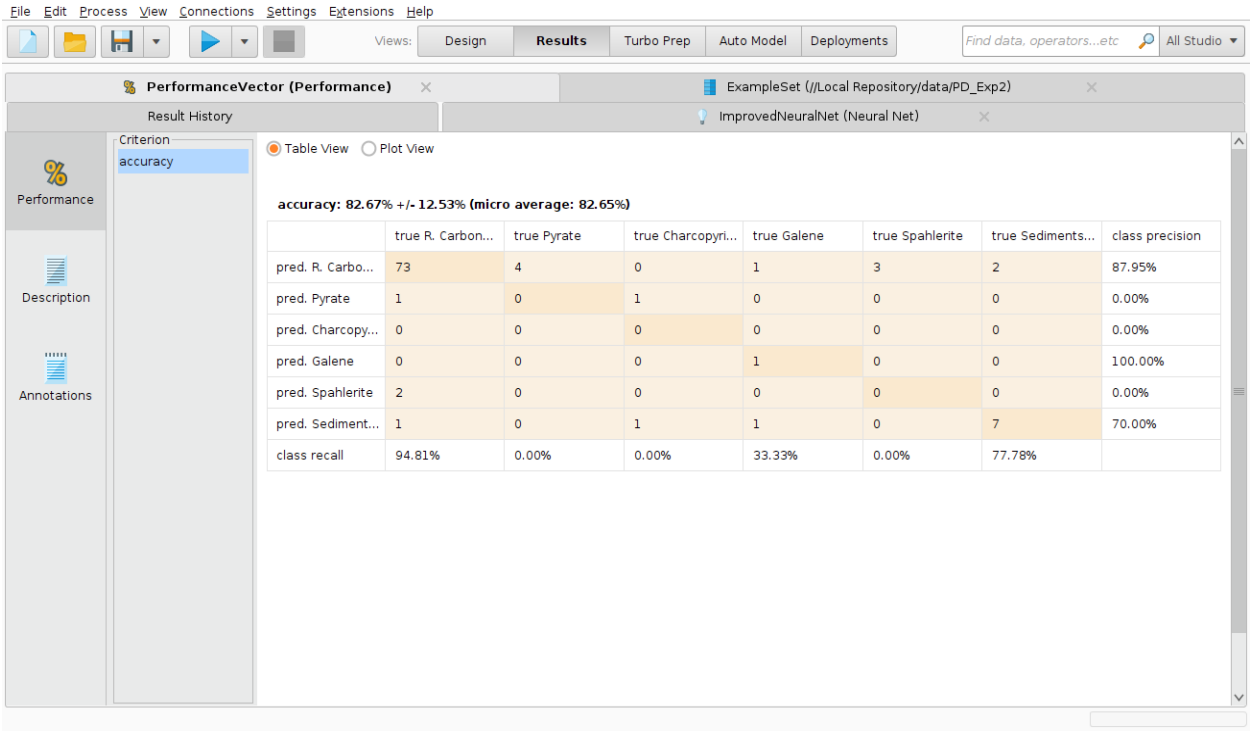
Cross Validation Operator :
number of folds = 10
Neural Network Operator :
Learning rate = 0.5
Momentum = 0.2
Epochs = 1000
Number of Hidden layers = 2 (with size = 10)

Graphical representation of Neural Network (Case 2):



The accuracy obtained for the above is **82.67% +/- 12.53% (micro average: 82.65%)**

Performance Vector :

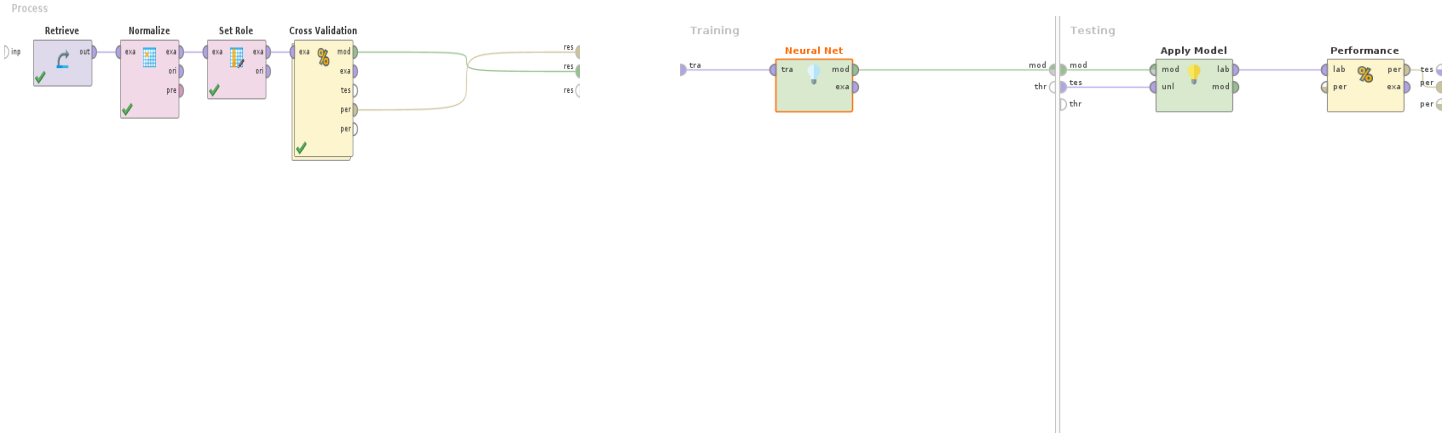


	true R. Carbon...	true Pyrate	true Charcopyri...	true Galene	true Spahlerite	true Sediments...	class precision
pred. R. Carbo...	73	4	0	1	3	2	87.95%
pred. Pyrate	1	0	1	0	0	0	0.00%
pred. Charcopy...	0	0	0	0	0	0	0.00%
pred. Galene	0	0	0	1	0	0	100.00%
pred. Spahlerite	2	0	0	0	0	0	0.00%
pred. Sediment...	1	0	1	1	0	7	70.00%
class recall	94.81%	0.00%	0.00%	33.33%	0.00%	77.78%	

Non-Descriptive Classifier for PED :

Firstly, we normalized the data using the “Normalise” Operator. We then used “Cross Validation” and “Neural Network” Operators to build this classifier.The Cross Validation Operator is a nested Operator. It has two subprocesses: a Training subprocess and a Testing subprocess. The Training subprocess is used for training a model.The trained model is then applied in the Testing subprocess. The Training subprocess is connected to Neural Network Operator and the Testing subprocess is connected to “Performance Operator” to measure the predictive accuracy and also to obtain the performance vector

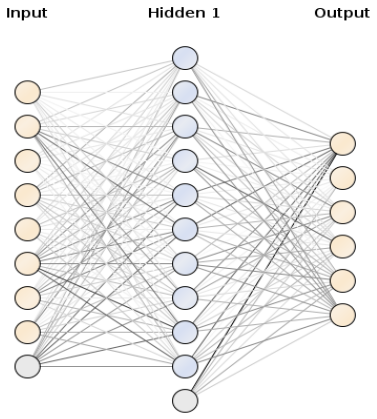
We have used different hyperparameters and the accuracies are as below.



Case 1 :

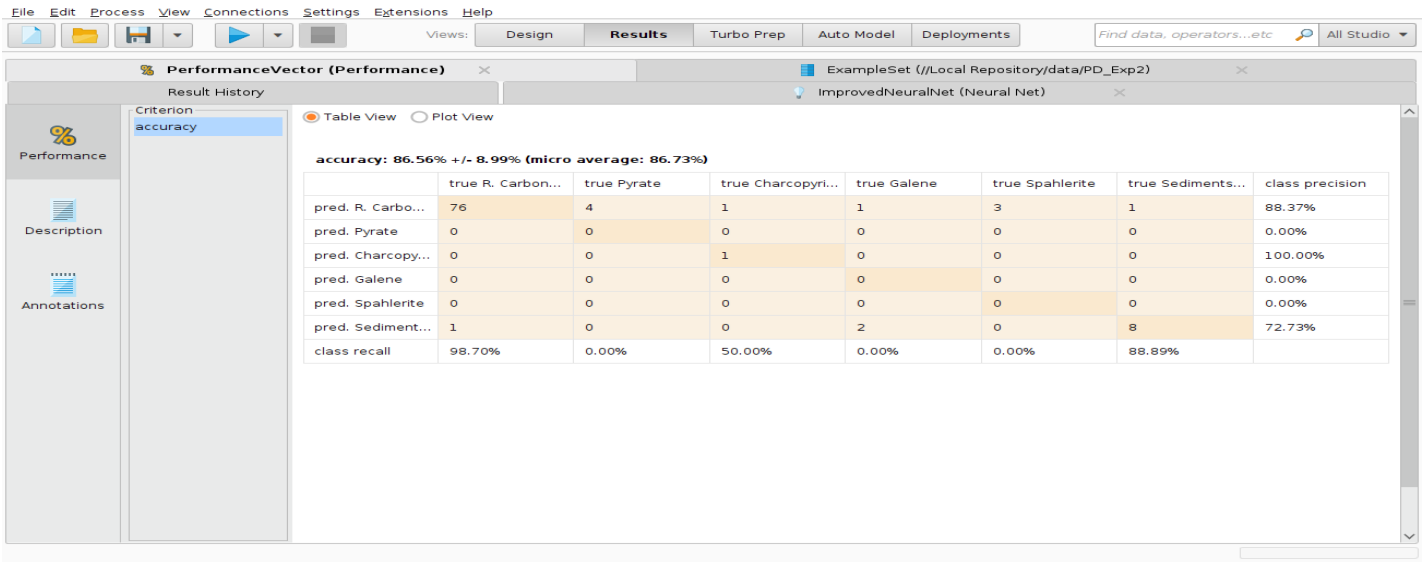
Cross Validation Operator :
number of folds = 10
Neural Network Operator :
Learning rate = 0.3
Momentum = 0.2
Epochs = 500
Number of Hidden layers = 1 (with size = 10)

Graphical representation of Neural Network (Case 1):



The accuracy obtained for the above is **86.56% +/- 0.99% (micro average: 86.73%)**

Performance Vector :

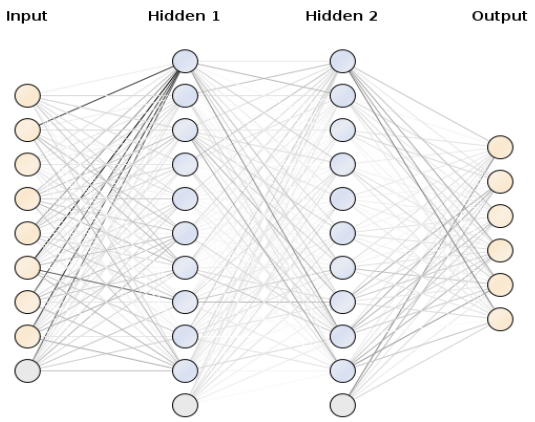


	true R. Carbon...	true Pyrate	true Charcopyri...	true Galene	true Spahlerite	true Sediments...	class precision
pred. R. Carbo...	76	4	1	1	3	1	88.37%
pred. Pyrate	0	0	0	0	0	0	0.00%
pred. Charcopy...	0	0	1	0	0	0	100.00%
pred. Galene	0	0	0	0	0	0	0.00%
pred. Spahlerite	0	0	0	0	0	0	0.00%
pred. Sediment...	1	0	0	2	0	8	72.73%
class recall	98.70%	0.00%	50.00%	0.00%	0.00%	88.89%	

Case 2 :

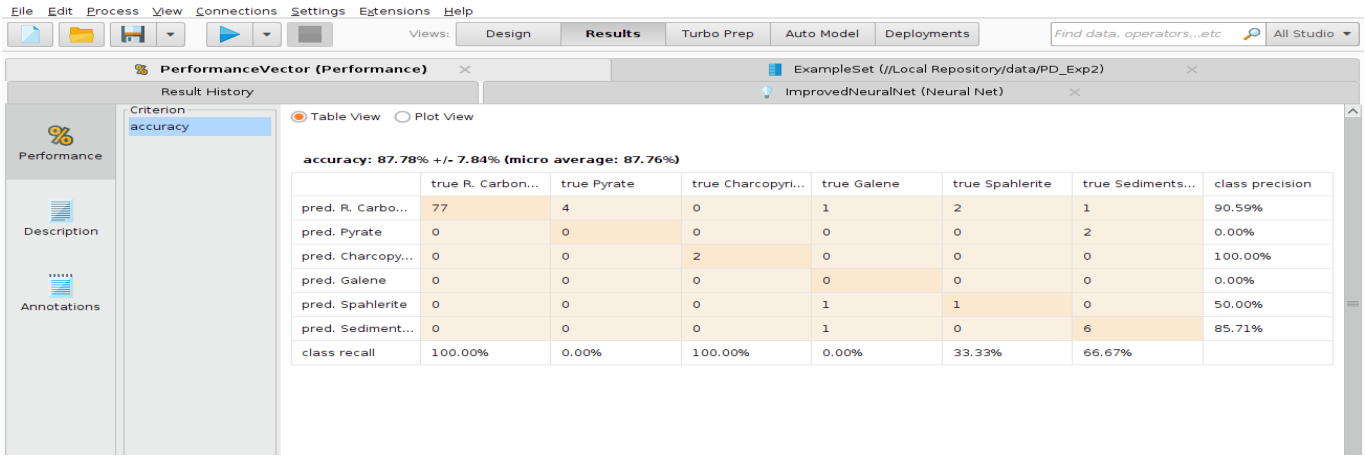
Cross Validation Operator :
number of folds = 10
Neural Network Operator :
Learning rate = 0.5
Momentum = 0.2
Epochs = 1000
Number of Hidden layers = 2 (with size = 10)

Graphical representation of Neural Network (Case 2):



The accuracy obtained for the above is **87.78% +/- 7.84% (micro average: 87.76%)**

Performance Vector :



	true R. Carbon...	true Pyrate	true Charcopyri...	true Galene	true Spahlerite	true Sediments...	class precision
pred. R. Carbo...	77	4	0	1	2	1	90.59%
pred. Pyrate	0	0	0	0	0	2	0.00%
pred. Charcopy...	0	0	2	0	0	0	100.00%
pred. Galene	0	0	0	0	0	0	0.00%
pred. Spahlerite	0	0	0	1	1	0	50.00%
pred. Sediment...	0	0	0	1	0	6	85.71%
class recall	100.00%	0.00%	100.00%	0.00%	33.33%	66.67%	

Experiment 2 (Contrast Classification) :

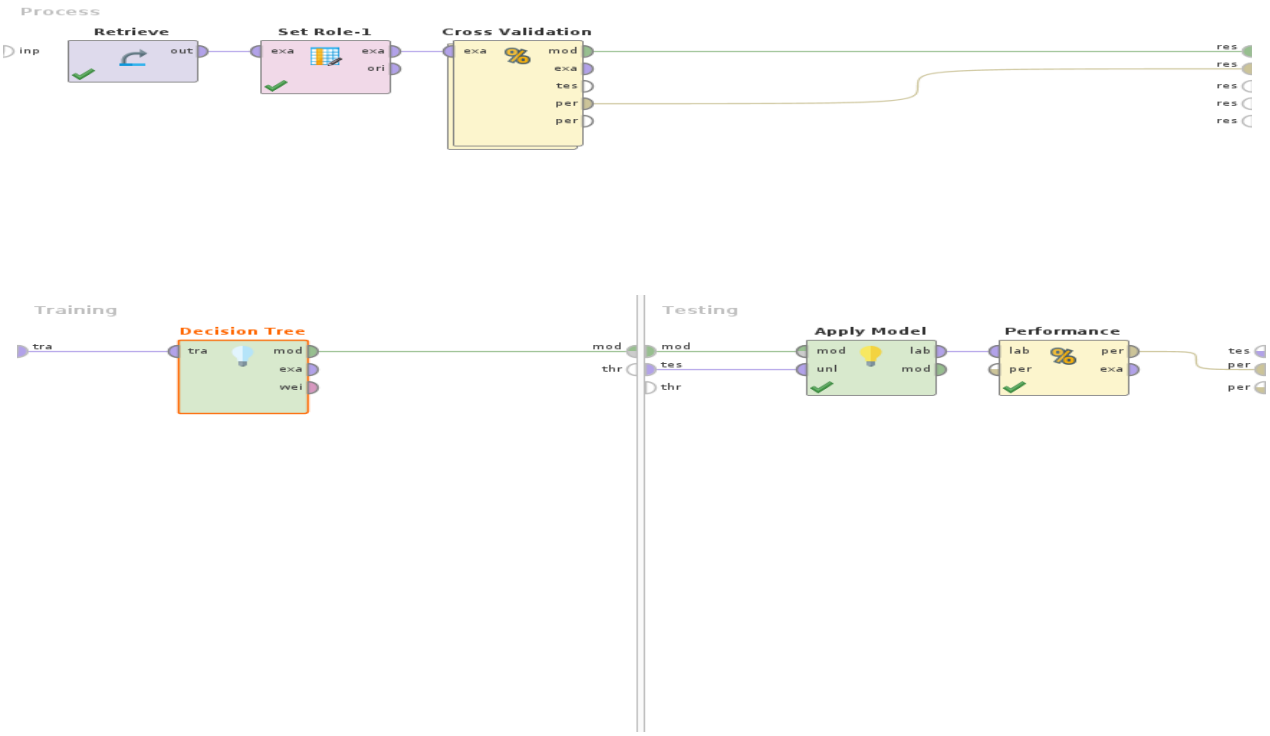
We were asked to build a Decision Tree classifier and a Neural Network classifier to perform the contrast classification for the class C1 i.e. **R. Carbonatees AND R. Carbonatees impures** using different topologies and testing methods. This also needs to be done for both PD and PED.

Descriptive Classifier for PD :

We used “Cross Validation” and “Decision Tree” Operators to build this classifier. The Cross Validation Operator is a nested Operator. It has two subprocesses: a Training subprocess and a Testing subprocess. The Training subprocess is used for training a model.The trained model is then applied in the Testing subprocess. The Training subprocess is connected to Decision Tree Operator and the Testing subprocess is connected to “Performance Operator” to measure the predictive accuracy and also to obtain the performance vector .

Parameters :

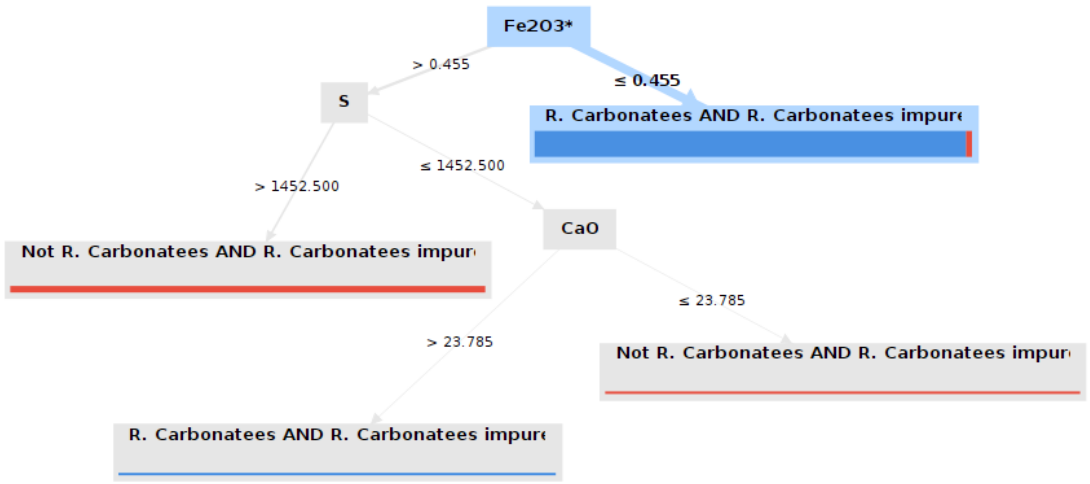
Cross Validation Operator :
number of folds = 10
Decision Tree Operator :
criterion = information_gain
maximum depth = 10
Apply_pruning = true



The decision tree obtained is :
(Note : Rule Accuracy is mentioned in the brackets)

Fe2O3* > 0.455
| S > 1452.500: Not R. Carbonatees AND R. Carbonatees impures **(16%)**
| S ≤ 1452.500
| | CaO > 23.785: R. Carbonatees AND R. Carbonatees impures **(4%)**
| | CaO ≤ 23.785: Not R. Carbonatees AND R. Carbonatees impures **(4%)**
Fe2O3* ≤ 0.455: R. Carbonatees AND R. Carbonatees impures **(73%)**

Graphical Representation of Decision Tree :

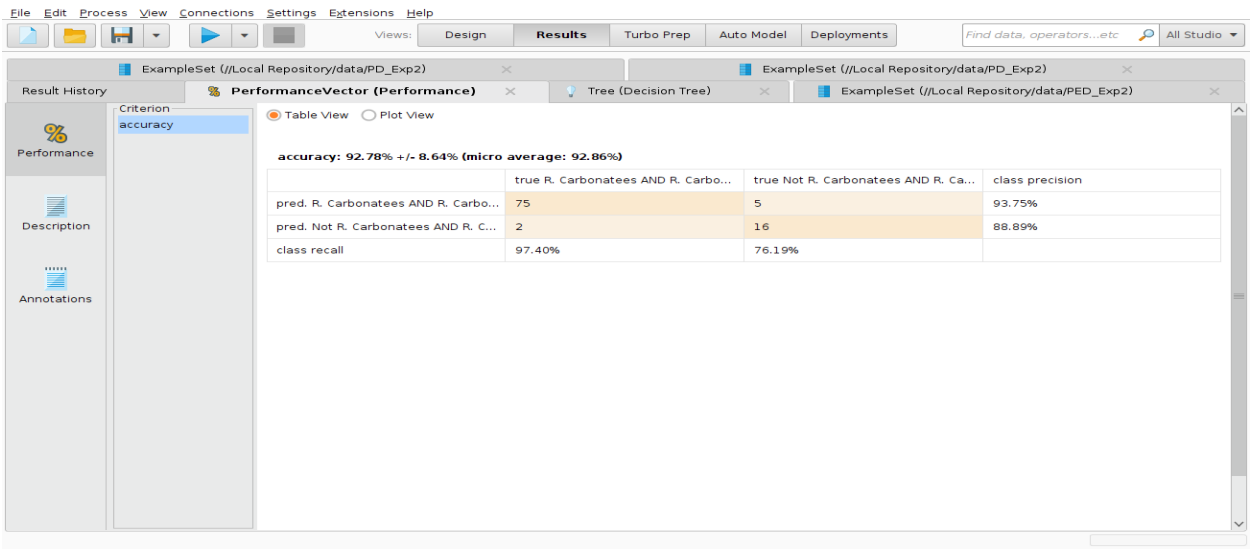


<u>Discriminant Rules (Predicate Form) :</u>

- IF Fe2O3*(x, > 0.455) AND S(x, > 1452.500) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures)
- IF Fe2O3*(x, > 0.455) AND S(x, <= 1452.500) AND CaO(x, > 23.785) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)
- IF Fe2O3*(x, > 0.455) AND S(x, <= 1452.500) AND CaO(x, <= 23.785) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures)
- IF Fe2O3*(x, <= 0.455) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)

The accuracy obtained for the above is **92.78% +/- 8.64% (micro average: 92.86%)**

Performance Vector :

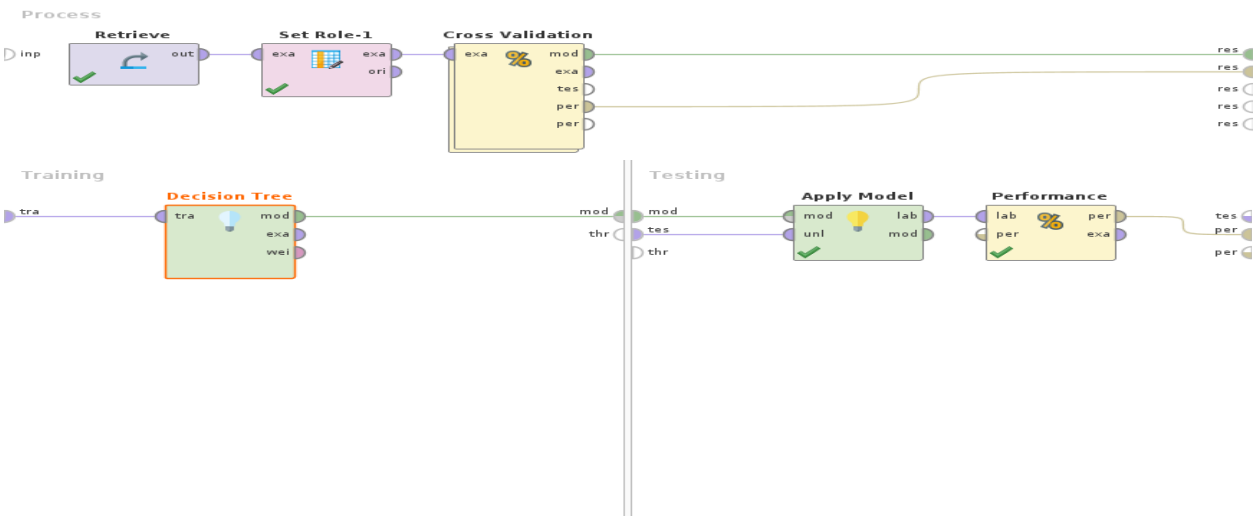


Descriptive Classifier for PED :

We used “Cross Validation” and “Decision Tree” Operators to build this classifier. The Cross Validation Operator is a nested Operator. It has two subprocesses: a Training subprocess and a Testing subprocess. The Training subprocess is used for training a model.The trained model is then applied in the Testing subprocess. The Training subprocess is connected to Decision Tree Operator and the Testing subprocess is connected to “Performance Operator” to measure the predictive accuracy and also to obtain the performance vector .

Parameters :

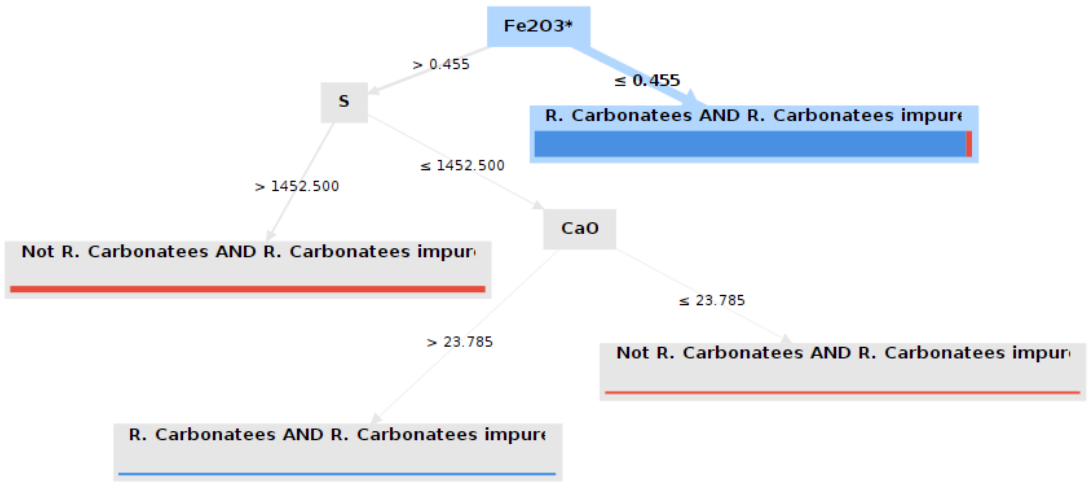
Cross Validation Operator :
number of folds = 10
Decision Tree Operator :
criterion = information_gain
maximum depth = 10
Apply_pruning = true



The decision tree obtained is :
(Note : Rule Accuracy is mentioned in the brackets)

Fe2O3* > 0.455
| S > 1452.500: Not R. Carbonatees AND R. Carbonatees impures **(16%)**
| S ≤ 1452.500
| | CaO > 23.785: R. Carbonatees AND R. Carbonatees impures **(4%)**
| | CaO ≤ 23.785: Not R. Carbonatees AND R. Carbonatees impures **(4%)**
Fe2O3* ≤ 0.455: R. Carbonatees AND R. Carbonatees impures **(73%)**

Graphical Representation of Decision Tree :

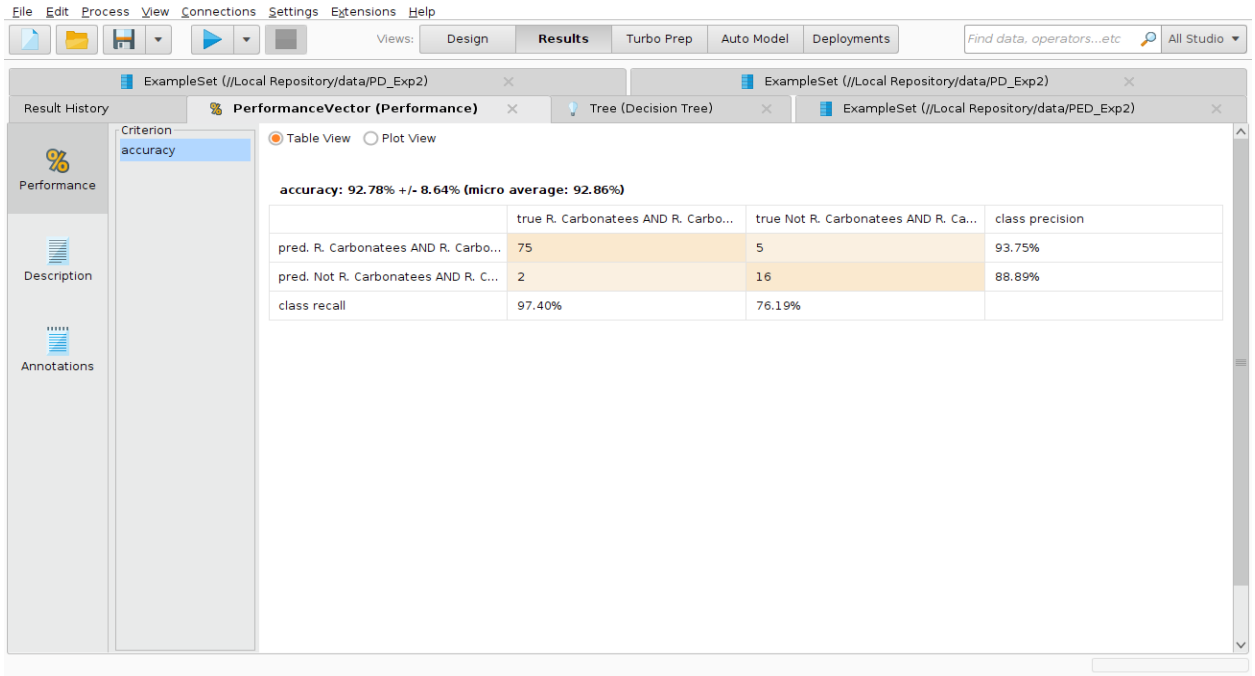


Discriminant Rules (Predicate Form) :

- IF Fe2O3*(x, > 0.455) AND S(x, > 1452.500) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures)
- IF Fe2O3*(x, > 0.455) AND S(x, <= 1452.500) AND CaO(x, > 23.785) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)
- IF Fe2O3*(x, > 0.455) AND S(x, <= 1452.500) AND CaO(x, <= 23.785) THEN TYPE DE ROCHE(x, Not R. Carbonatees AND R. Carbonatees impures)
- IF Fe2O3*(x, <= 0.455) THEN TYPE DE ROCHE(x, R. Carbonatees AND R. Carbonatees impures)

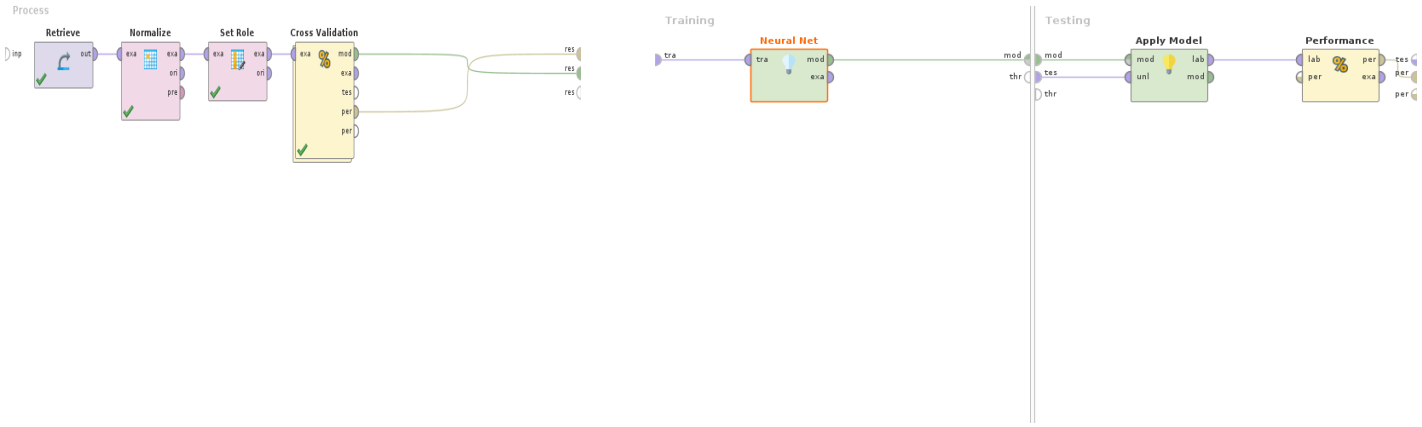
The accuracy obtained for the above is **93.78% +/- 8.86% (micro average: 93.88%)**

Performance Vector :



Non-Descriptive Classifier for PD :

Firstly, we normalized the data using the “Normalise” Operator. We then used “Cross Validation” and “Neural Network” Operators to build this classifier.The Cross Validation Operator is a nested Operator. It has two subprocesses: a Training subprocess and a Testing subprocess. The Training subprocess is used for training a model.The trained model is then applied in the Testing subprocess. The Training subprocess is connected to Neural Network Operator and the Testing subprocess is connected to “Performance Operator” to measure the predictive accuracy and also to obtain the performance vector

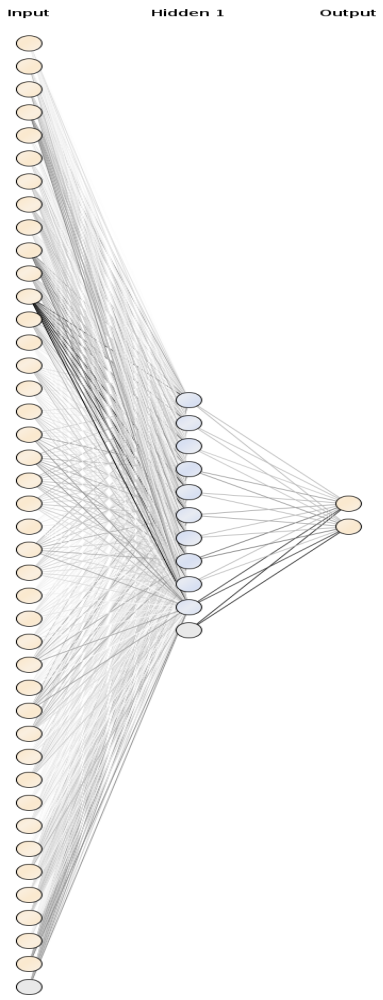


We have used different hyperparameters and the accuracies are as below.

Case 1 :

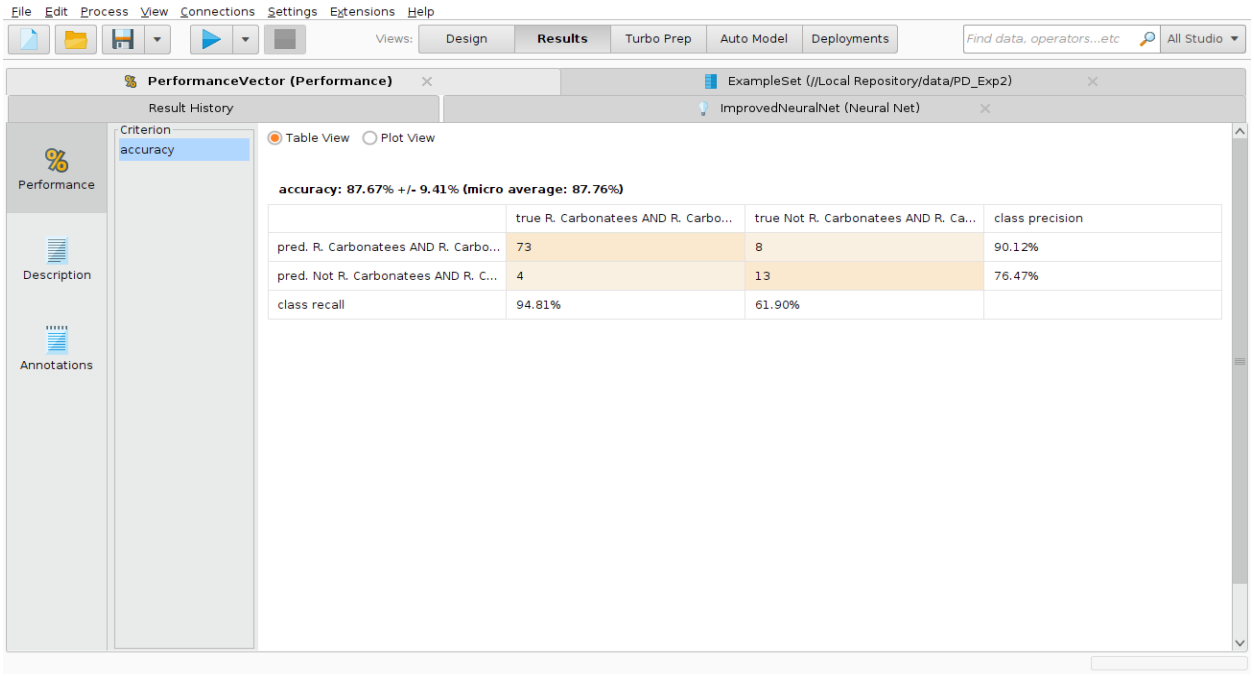
Cross Validation Operator :
number of folds = 10
Neural Network Operator :
Learning rate = 0.3
Momentum = 0.2
Epochs = 500
Number of Hidden layers = 1 (with size = 10)

Graphical representation of Nueral Network (Case 1):



The accuracy obtained for the above is **87.67% +/- 9.41% (micro average: 87.76%)**

Performance Vector :

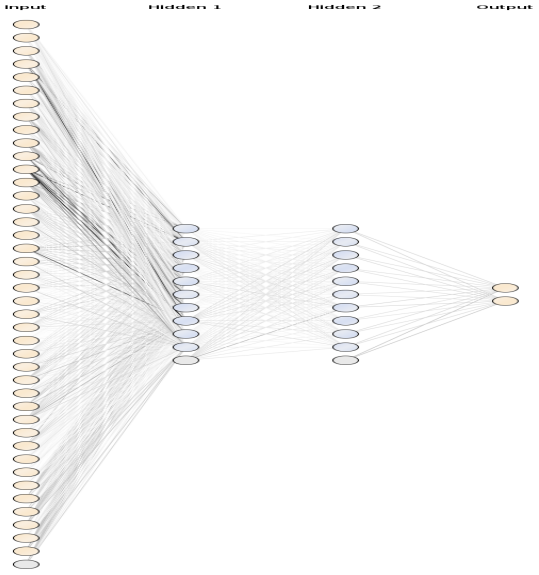


	true R. Carbonatees AND R. Carbo...	true Not R. Carbonatees AND R. Ca...	class precision
pred. R. Carbonatees AND R. Carbo...	73	8	90.12%
pred. Not R. Carbonatees AND R. C...	4	13	76.47%
class recall	94.81%	61.90%	

Case 2 :

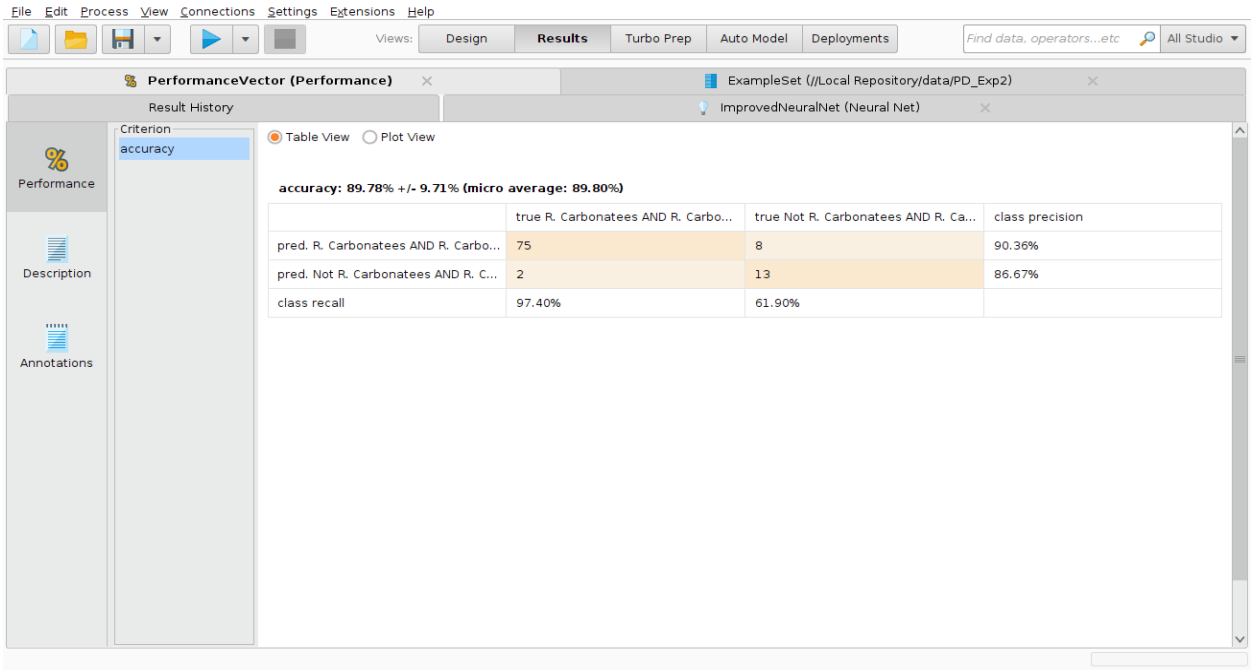
Cross Validation Operator :
number of folds = 10
Neural Network Operator :
Learning rate = 0.5
Momentum = 0.2
Epochs = 1000
Number of Hidden layers = 2 (with size = 10)

Graphical representation of Neural Network (Case 2):



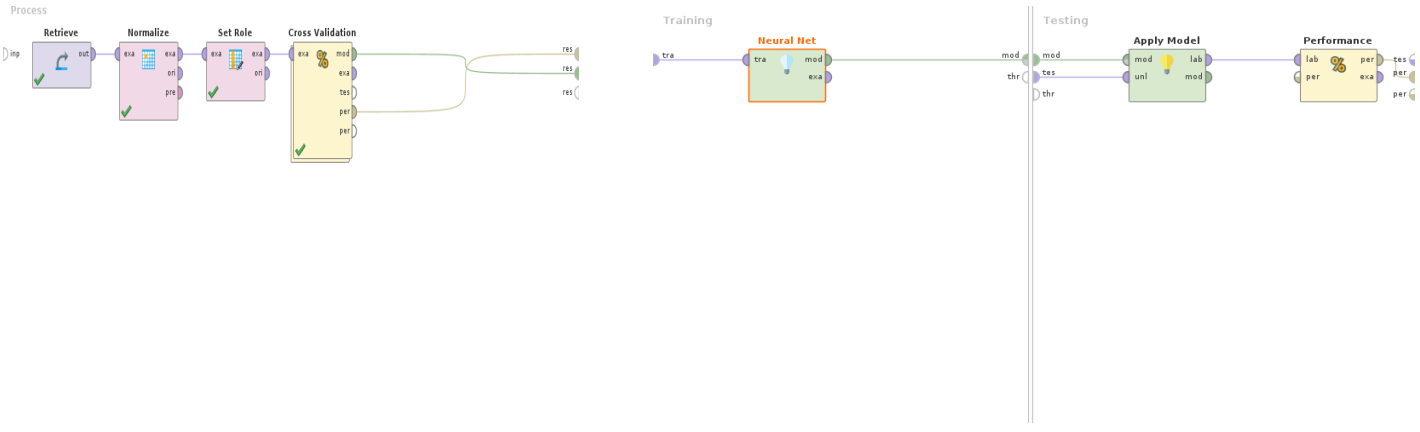
The accuracy obtained for the above is **9.78% +/- 9.71% (micro average: 89.80%)**

Performance Vector :



Non-Descriptive Classifier for PED :

Firstly, we normalized the data using the “Normalise” Operator. We then used “Cross Validation” and “Neural Network” Operators to build this classifier.The Cross Validation Operator is a nested Operator. It has two subprocesses: a Training subprocess and a Testing subprocess. The Training subprocess is used for training a model.The trained model is then applied in the Testing subprocess. The Training subprocess is connected to Neural Network Operator and the Testing subprocess is connected to “Performance Operator” to measure the predictive accuracy and also to obtain the performance vector

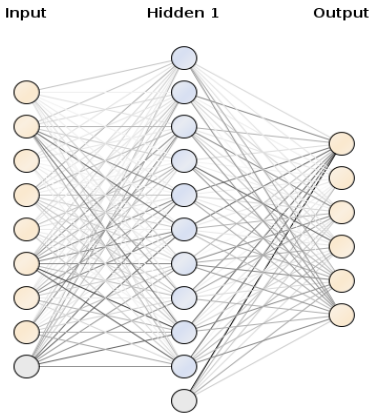


We have used different hyperparameters and the accuracies are as below.

Case 1 :

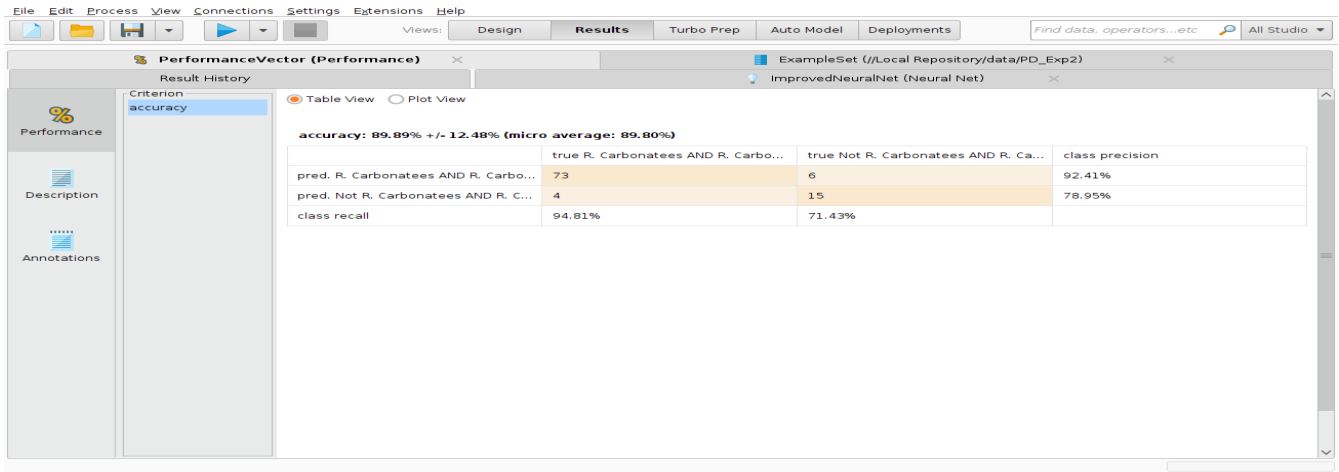
Cross Validation Operator :
number of folds = 10
Neural Network Operator :
Learning rate = 0.3
Momentum = 0.2
Epochs = 500
Number of Hidden layers = 1 (with size = 10)

Graphical representation of Nueral Network (Case 1):



The accuracy obtained for the above is **89.89% +/- 12.48% (micro average: 89.80%)**

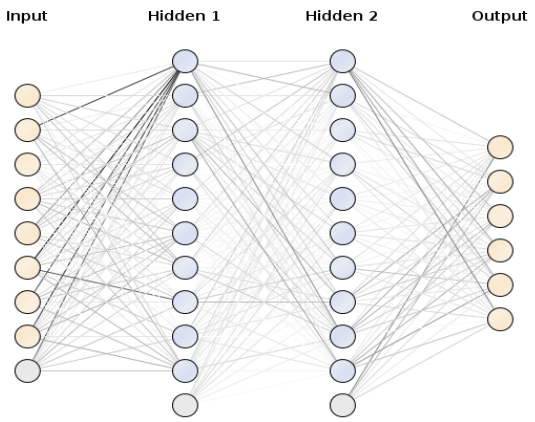
Performance Vector :



Case 2 :

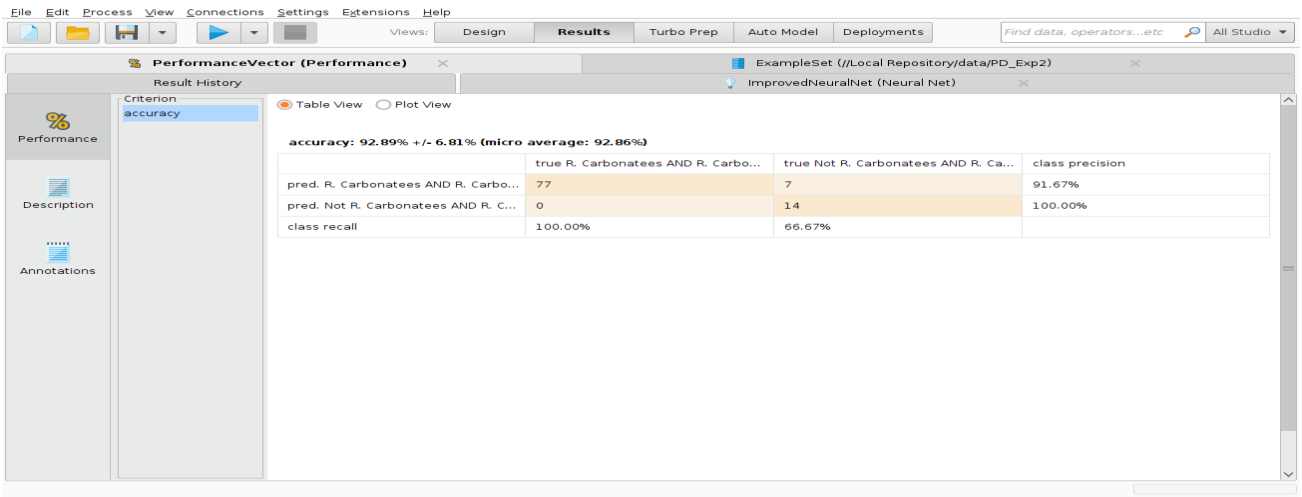
Cross Validation Operator :
number of folds = 10
Neural Network Operator :
Learning rate = 0.5
Momentum = 0.2
Epochs = 1000
Number of Hidden layers = 2 (with size = 10)

Graphical representation of Neural Network (Case 2):



The accuracy obtained for the above is **92.89% +/- 6.81% (micro average: 92.86%)**

Performance Vector :



SUMMARY

Predictive Accuracies for Descriptive Classifiers :

PD - Complete	87.76
PED - Complete	82.65
PD - Contrast	92.86
PED - Contrast	93.88

Here we have used the in-built Decision Tree model in the RapidMiner tool with 10 fold cross validation for both PD and PED (both Complete and Contrast classifiers). If we see, the classifier has worked better for a complete data set than just with the important attributes when we tried to do Full classification. On the other hand, In case of contrast classification the accuracy is more PED when compared to PD, which means that classifier has worked better with dataset of important attributes in contrast classification

The Rules Accuracies have been mentioned as part of the classifiers description above. Where as the attributes **Zn**, **S**, **Pb**, **CaO+MgO** and **Fe2O3*** seem to be common across both the complete classification decision trees. Where as **Fe2O3*, S, CaO** seem to be common for contrast classification.

Apart from these, the decision trees are much more compact for contrast classification compared to complete classifiers.

Predictive Accuracies for Non - Descriptive Classifiers :

Complete classification:

PD - Case 1	84.69
PD - Case 2	82.65
PED - Case 1	86.73
PED - Case 2	87.76

Contrast classification:

PD - Case 1	87.76
PD - Case 2	89.80

PED - Case 1	89.80
PED - Case 2	92.86

Here we used a Neural Network algorithm along with 10 Fold cross validation. We have normalised the data using the in-built “Z transformation” technique present in the Rapid Miner tool. We have used two topologies for analysing these classifiers. We can see that the accuracies are higher for the classifiers of Topology 2 (refers to higher learning rate , more hidden layers) when compared to the default settings in most of the cases. Along with that the accuracies are higher for contrast classifiers compared to the complete classification.

Highest Predictive Accuracies across all Classifiers :

Complete Classification	87.76	Descriptive Classifier when ran on PD data & Non Descriptive Classifier when ran on PED data
Contrast Classification	93.88	Descriptive Classifier when ran on PED data

CONCLUSION:

- The data provided is so biased towards a single class and developing classifiers to classify all the classes simultaneously is not a right option as we have so less data for other classes.
- If we observe, the same is shown with higher accuracies for **contrast** classifiers compared to **complete** classifiers.
- The need to compare rules accuracy is not a valid point, since the rules are different across all the decision trees and comparing them without similar paths from root to leaves is not required.
- We used **K-Fold Cross validation** across all the classifiers and tools. Of all the options (Percentage split, training set and K-Fold), K-Fold seems to be a better solution due to inclusion of all data points for training and testing at least once.
- Since the testing method is the same across all the classifiers and tools, comparison on predictive accuracy is valid and provides conclusive evidence.
- The highest predictive accuracies across complete and contrast classifiers are mentioned below based on each tool.

	WEKA	**Orange**	**RapidMiner**
Complete	91.8367 (PD - Descriptive)	91.8 (PED - Non Descriptive)	87.76 (PD -Descriptive & PED - Non Descriptive)
Contrast	95.9184 (PED - Descriptive)	92.9 (PED - Non Descriptive)	93.88 (PED - Descriptive)

- If we observe, the overall predictive accuracy is higher for contrast classification compared to complete classification.
- WEKA has given the higher predictive accuracy for both the classifiers followed by RapidMiner and Orange.
- While performing complete classification, WEKA didn't classify all the classes for most of the configuration settings whereas Orange and RapidMiner didn't compromise on that for bit less predictive accuracy.

- **Experience with Tools :**
  - WEKA takes a huge consumption of RAM and crashes immediately if the memory limit has reached. On the other hand, RapidMiner and Orange have taken less disk space.
  - RapidMiner and Orange are more user friendly compared to WEKA and also developer friendly for logging, debugging and breakpoints.
  - Orange is better in visualisations of workflow creation and classification algorithms.
  - WEKA has better configuration settings for classification providing different options for each classifier.
  - We would prefer WEKA for preprocessing, Orange or Rapid Miner for Descriptive classification and Orange for Non-Descriptive classification.