Supervised Learning on Bakary Data Using WEKA

Shariq Syed & Sudhin Domala

Outline

- Classification Tool: WEKA

  Waikato Environment for Knowledge Analysis
- Can be downloaded from http://www.cs.waikato.ac.nz/~ml/weka/index.html

Data Preparation: Removing Attributes

- Removing Attributes: We ended up removing attributes which had roughly 33% or more of its data missing but ideally
- On WEKA, we looked at the percentage of data which was missing and removed those from the dataset. These attributes were:
    - Mo: 88% missing
    - Co: 84% missing
    - As: 72% missing
    - Cd: 70% missing
    - Pb: 55% missing
    - Sc: 50% missing
    - Ni: 40% missing
    - Li: 39% missing

Data Preparation: Missing Data

- Many of the attributes contained missing data which can throw off the classifier and so it is imperative that these values be filled in with either 0's or averages given the other classes. To achieve this, we used WEKA's ReplaceMissingValues filter which replaced the missing values with the mean of the numeric values.

Data Preprocessing: Discretization

- In order to discretize the data, we applied another filter on WEKA which is called Discretize. The only modification I made was change the number of bins from 10 to 3.
- Data #1
  - Binning Method(Discrete Bin Count = 3)
  - useEqualFrequency = True
  - Equal Width Bins
- Data #2
  - Binning Method(Discrete Bin Count = 3,)
  - useEqualFrequency = False
  - Equal Depth Bins

Data Preprocessing: Discretization Set

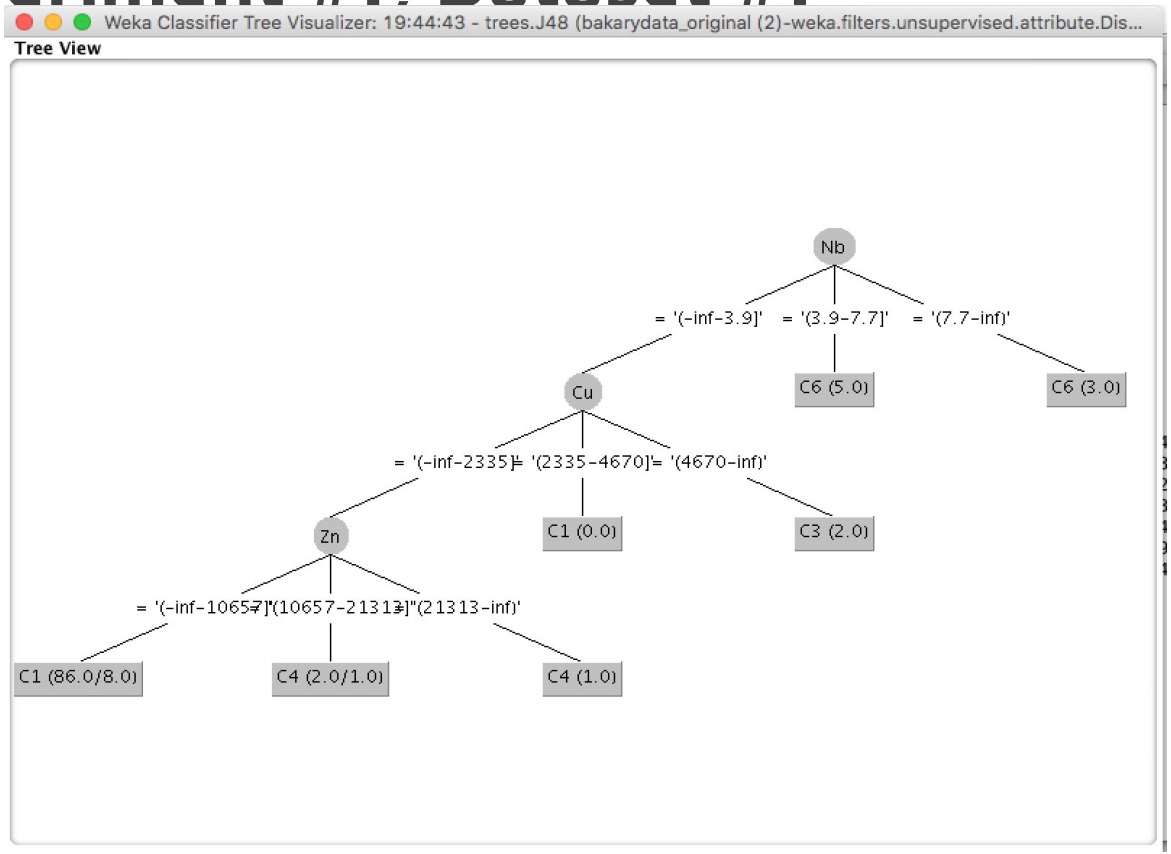
Data Preprocessing: Discretization Set

Experiments

- Experiment 1 -> Full Learning
  - Use all attributes to classify all classes (C1-C6)
  - Done by Discretization and usage of J48 Algorithm
- Experiment 2 -> Contrast Learning
  - Using all attributes to compare class C1 with the rest of the classes
  - Done by Discretization and Renaming Nominal Values on Type de roche, and J48
- Experiment 3 -> Limited Learning
  - Construction of decision tree using only the major attributes
  - Removal of non-major attributes in preprocessing and prior to experiment 1&2 implementations

Experiment #1, Dataset #1

- Decision tree using J48 algorithm
- Used k-fold(k=10)
- Predictive Accuracy: 84.8485%



Discriminant Rules

- IF Nb = “(-inf-3.9]” AND Cu = “(-inf-2335] – AND Zn = “(-inf-10657]”

  THEN class = “C1”

- IF Nb = “(-inf-3.9]” AND Cu = “(-inf-2335]” – AND Zn = “(10657-21313]”

  THEN class = “C4”

- IF Nb = “(-inf-3.9]” AND Cu = “(-inf-2335]” – AND Zn = “(21313-inf)”

  THEN class = “C4”

Discriminant Rules(cont.)

- IF Nb = “(-inf-3.9]” AND Cu = “(2335-4670)”

  THEN class = “C1”

- IF Nb = “(-inf-3.9]” AND Cu = “(4670-inf)”

  THEN class = “C3”

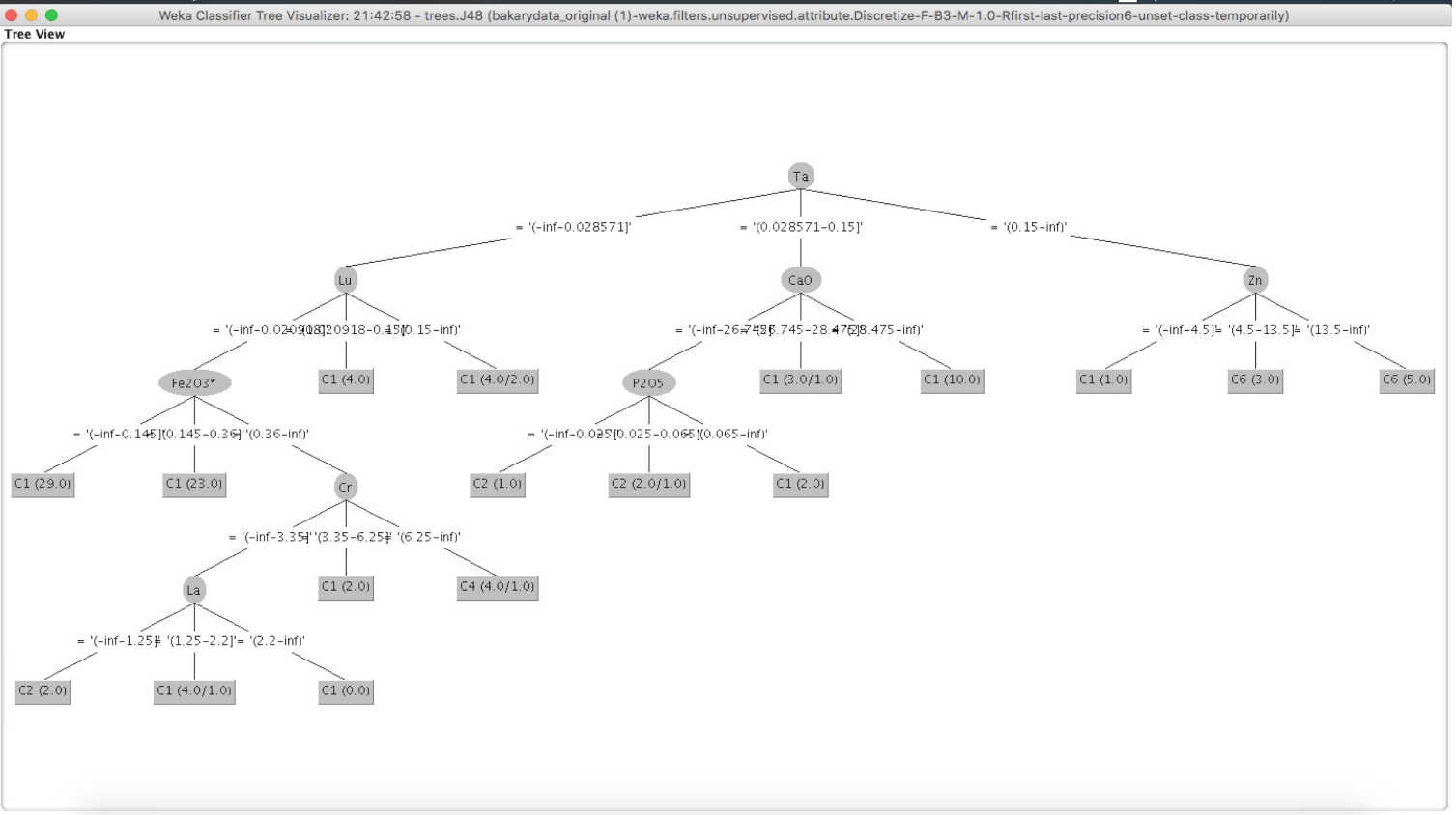
- IF Nb = “(3.9-7.7]”

  THEN class = “C6”

- IF Nb = “(7.7-inf)”

  THEN class = “C6”

Experiment #1, Dataset #2



- Used Training Set method for PA calculation
- Predictive Accuracy: 75.7576%

Discriminant Rules

- IF Ta="(-inf-0.028571]" AND Lu="(-inf-0.020918]" AND Fe2O3="(-inf-0.145]"

  THEN Class="C1"

- IF Ta="(-inf-0.028571]" AND Lu="(-inf-0.020918]" AND Fe2O3="(0.145-0.36]"

  THEN Class="C1"

- IF Ta="(-inf-0.028571]" AND Lu="(-inf-0.020918]" AND Fe2O3="(0.145-0.36]" AND Cr="(-inf-3.35]" AND La="(-inf-1.25]"

  THEN Class="C2"

Discriminant Rules(cont.)

- IF Ta="(-inf-0.028571]" AND Lu="(-inf-0.020918]" AND Fe2O3="(0.145-0.36]" AND Cr="(-inf-3.35]" AND La="(1.25-2.2]"

  THEN Class="C1"

- IF Ta="(-inf-0.028571]" AND Lu="(-inf-0.020918]" AND Fe2O3="(0.145-0.36]" AND Cr="(-inf-3.35]" AND La="(2.2-inf]"

  THEN Class="C1"

- IF Ta="(-inf-0.028571]" AND Lu="(-inf-0.020918]" AND Fe2O3="(0.145-0.36]" AND Cr="(3.35,6.25]"

  THEN Class="C1"

Discriminant Rules(cont.)

- IF Ta="(-inf-0.028571]" AND Lu="(-inf-0.020918]" AND Fe2O3="(0.145-0.36]" AND Cr="(6.25-inf]"

  THEN Class="C4"

- IF Ta="(-inf-0.028571]" AND Lu="(0.020918-0.15]"

  THEN Class="C1"

- IF Ta="(-inf-0.028571]" AND Lu="(0.15-inf)"

  THEN Class="C1"

Discriminant Rules(cont.)

- IF Ta=“(0.028571-0.15]” AND CaO=“(-inf-26.745]” AND P2O5=”(-inf-0.025]”

  THEN Class=“C2”

- IF Ta=“(0.028571-0.15]” AND CaO=“(-inf-26.745]” AND P2O5=”(0.025-0.065]”

  THEN Class=“C2”

- IF Ta=“(0.028571-0.15]” AND CaO=“(-inf-26.745]” AND P2O5=”(0.065-inf]”

  THEN Class=“C1”

- IF Ta=“(0.028571-0.15]” AND CaO=“(26.745-28.475]”

  THEN Class=“C3”

Discriminant Rules(cont.)

- IF Ta="(0.028571-0.15]" AND CaO="(28.475-inf]"

  THEN Class="C1"

- IF Ta="(0.15-inf]" AND Zn="(inf-4.5]"

  THEN Class="C1"

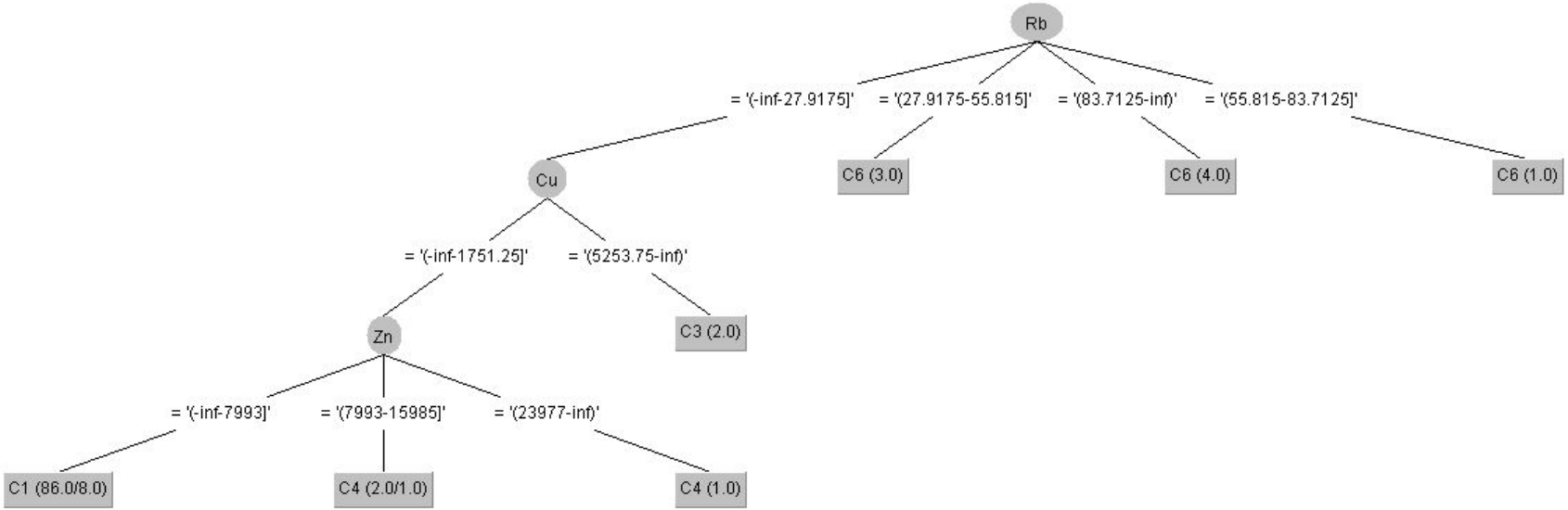
- IF Ta="(0.15-inf]" AND Zn="(4.5-13.5]"

  THEN Class="C6"

- IF Ta="(0.15-inf]" AND Zn="(13.5-inf]"

  THEN Class="C6"

Experiment #2, Data Set #1



- Predictive Accuracy: 83.8384%

Discriminant Rules

- IF Rb="(-inf-27.9175]" AND Cu="(-inf-1751.25]" AND Zn="(-inf-7993]

  THEN Class="C1"

- IF Rb="(-inf-27.9175]" AND Cu="(-inf-1751.25]" AND Zn="(7993-15985]

  THEN Class="C4"

- IF Rb="(-inf-27.9175]" AND Cu="(-inf-1751.25]" AND Zn="(23977-inf]

  THEN Class="C4"

- IF Rb="(-inf-27.9175]" AND Cu="(5253.75-inf]"

  THEN Class="C3"

Discriminant Rules(cont.)

- IF Rb=“(27.9175-55.815]”

  THEN Class=“C6”

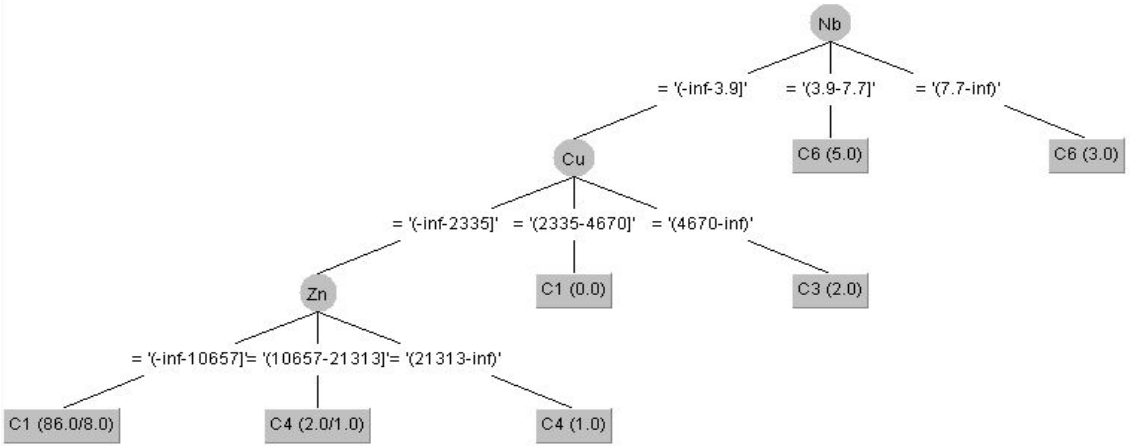
- IF Rb=“(55.815-83.7125]”

  THEN Class=“C6”

- IF Rb=“(83.7125-inf]”

  THEN Class=“C6”

Experiment #2, Data Set #2



- Predictive Accuracy: 84.8485%

Discriminant Rules

- IF Nb=“(-inf-3.9]” AND Cu=”(-inf-2335] AND Zn=”(-inf-10657]”

  THEN Class=“C1”

- IF Nb=“(-inf-3.9]” AND Cu=”(-inf-2335] AND Zn=”(10657-21313]”

  THEN Class=“C4”

- IF Nb=“(-inf-3.9]” AND Cu=”(-inf-2335] AND Zn=”(21313-inf]”

  THEN Class=“C4”

- IF Nb=“(-inf-3.9]” AND Cu=”(2335-4670]

  THEN Class=“C1”

Discriminant Rules(cont.)

- IF Nb=“(-inf-3.9]” AND Cu=”(4670-inf]

  THEN Class=“C3”

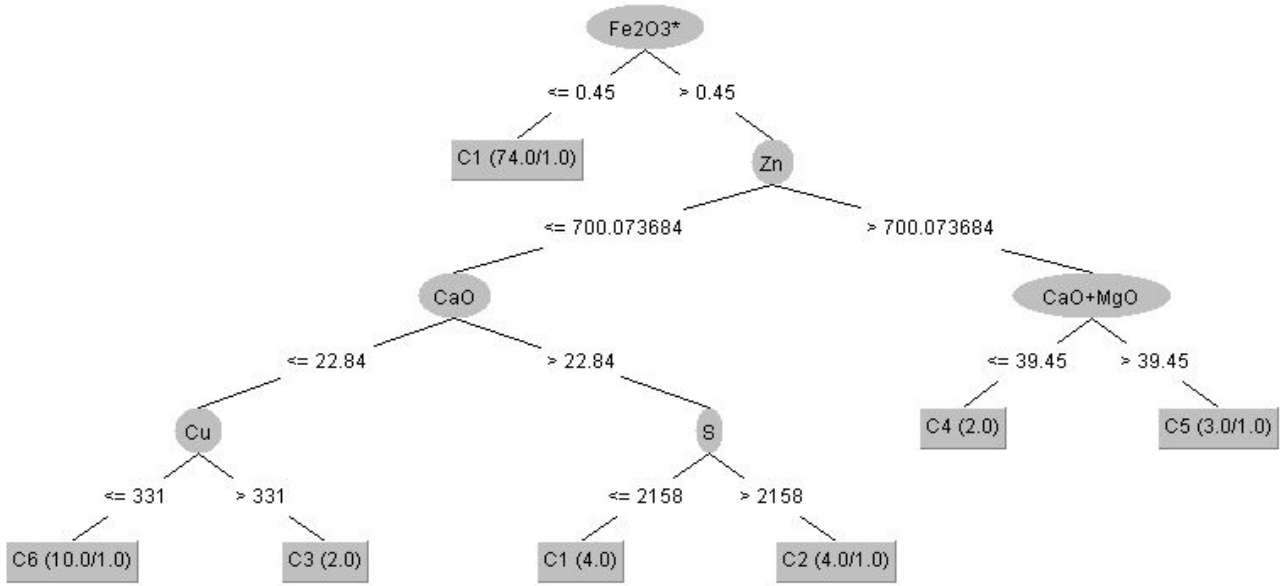
- IF Nb=“(3.9-7.7]”

  THEN Class=“C6”

- IF Nb=“(7.7-inf]”

  THEN Class=“C3”

Experiment #3, Data Set #1



- Predictive Accuracy: 81.8182%

Discriminant Rules

- IF Fe2O3=“(<=.45]”

  THEN Class=“C1”

- IF Fe2O3=“(>.45]” AND Zn=”<=700.073684” AND CaO=”<=22.84” AND Cu=”<=331”

  THEN Class=“C6”

- IF Fe2O3=“(>.45]” AND Zn=”<=700.073684” AND CaO=”<=22.84” AND Cu=”>331”

  THEN Class=“C3”

Discriminant Rules(cont.)

- IF Fe2O3=“(>.45]” AND Zn=”<=700.073684” AND CaO=”>22.84” AND S=”<=2158”

  THEN Class=“C1”

- IF Fe2O3=“(>.45]” AND Zn=”<=700.073684” AND CaO=”>22.84” AND S=”>2158”

  THEN Class=“C2”

- IF Fe2O3=“(>.45]” AND Zn=”>700.073684” AND CaO+MgO=”<=39.45”

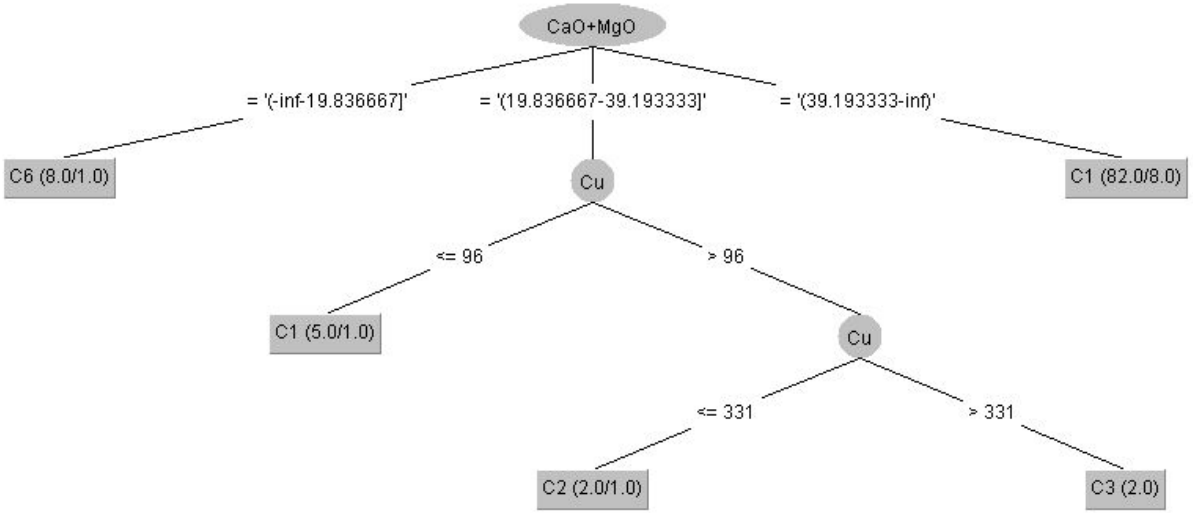
  THEN Class=“C4”

Discriminant Rules(cont.)

- IF Fe2O3=“(>.45]” AND Zn=”>700.073684” AND CaO+MgO=”>39.45”

  THEN Class=“C3”

Experiment #3, Data Set #2



- Predictive Accuracy: 84.8485%

Discriminant Rules

- IF CaO+MgO=“(-inf-19.836667]”

  THEN Class=“C6”

- IF CaO+MgO=“(19.836667-39.193333]” AND Cu=”<=96”

  THEN Class=“C1”

- IF CaO+MgO=“(19.836667-39.193333]” AND Cu=”>96” AND Cu=”<=331”

  THEN Class=“C2”

- IF CaO+MgO=“(19.836667-39.193333]” AND Cu=”>96” AND Cu=”>331”

  THEN Class=“C3”

Discriminant Rules(cont.)

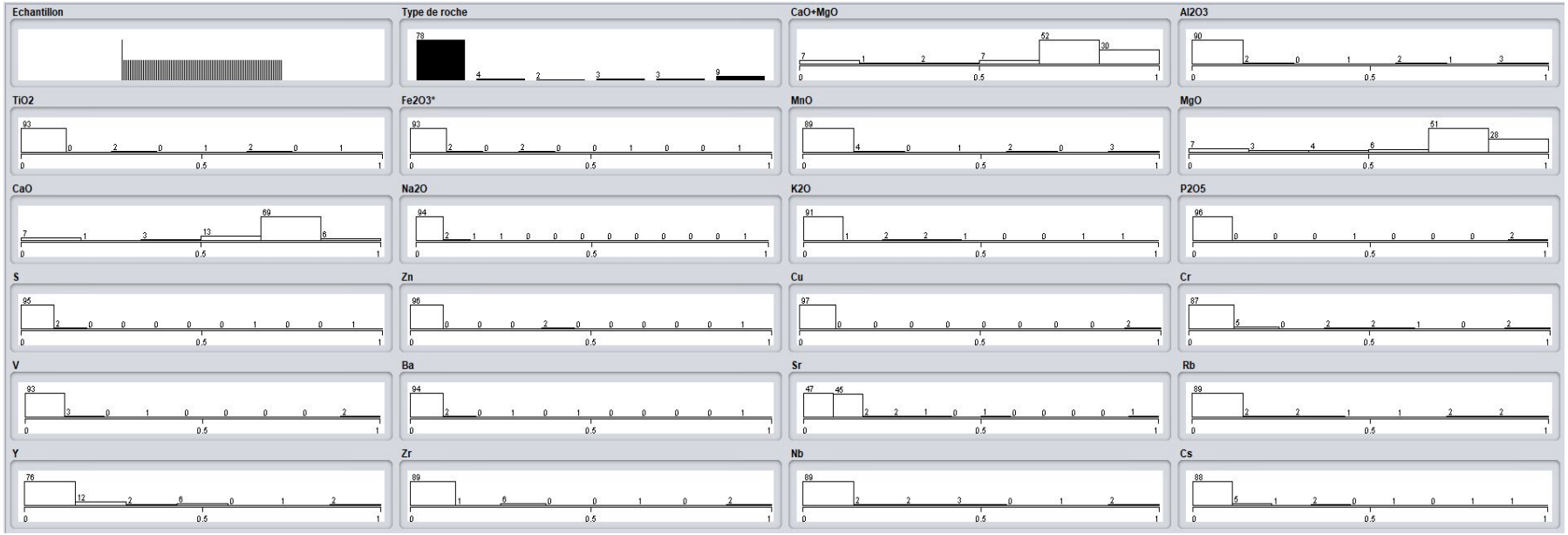
- IF CaO+MgO=“(39.193333-inf]”

  THEN Class=“C1”

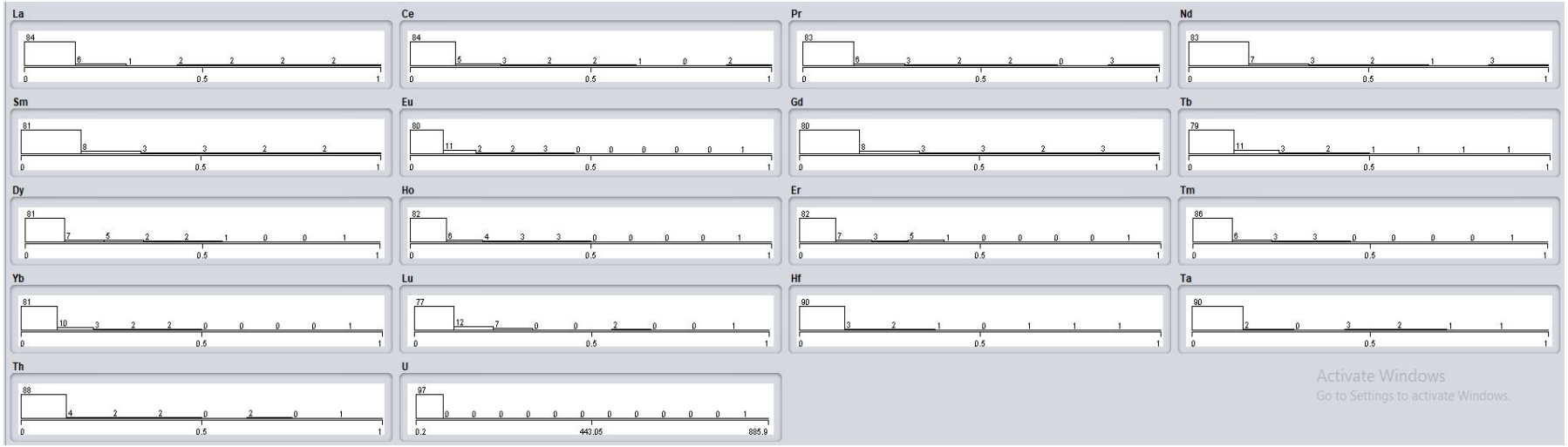
NN Data Preprocessing: Normalization

- In order to normalize the data, we applied another filter on WEKA which is called Normalize.
  - No modifications and no selections have been made.
  - Scale: 1.0
  - Translation: 0.0
- For experiment 2 (and thus experiment 3), we used “RenameNomial Values” to distinguish the classes on “Type de roche”
- Neuron networks display the input nodes in green, connections in grey lines, red nodes for hidden layers, and yellow boxes for classes.

NN Data Preprocessing: Normalization

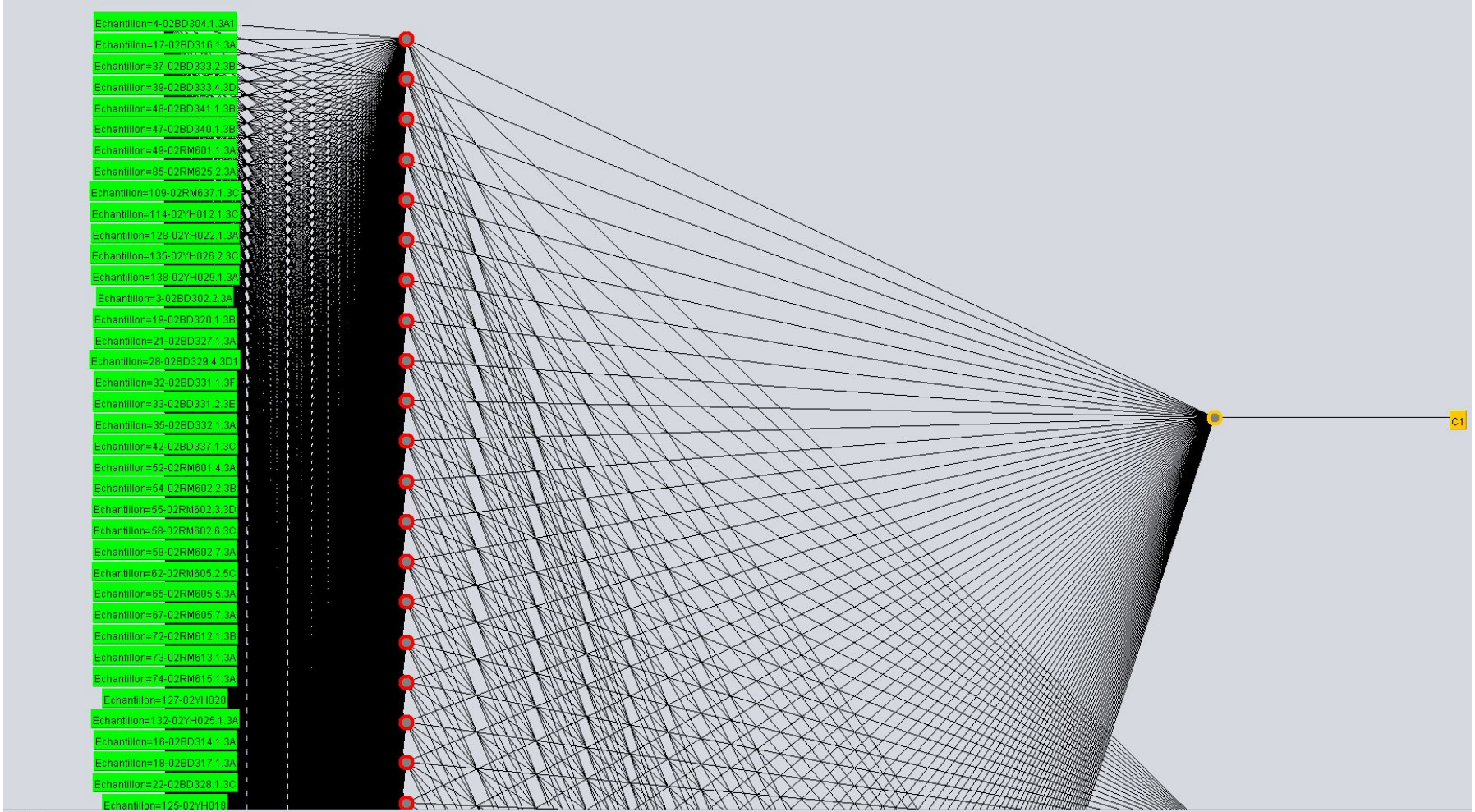


NN Data Preprocessing: Normalization



NN Set: Experiment #1

- Predictive Accuracy: 86.8687%
- To build neuron network we used: “MultilayerPerceptron” function in “Classify” section
- Had GUI set to true and Cross-Validation set of Folds of k=10



Controls

Start

Accept

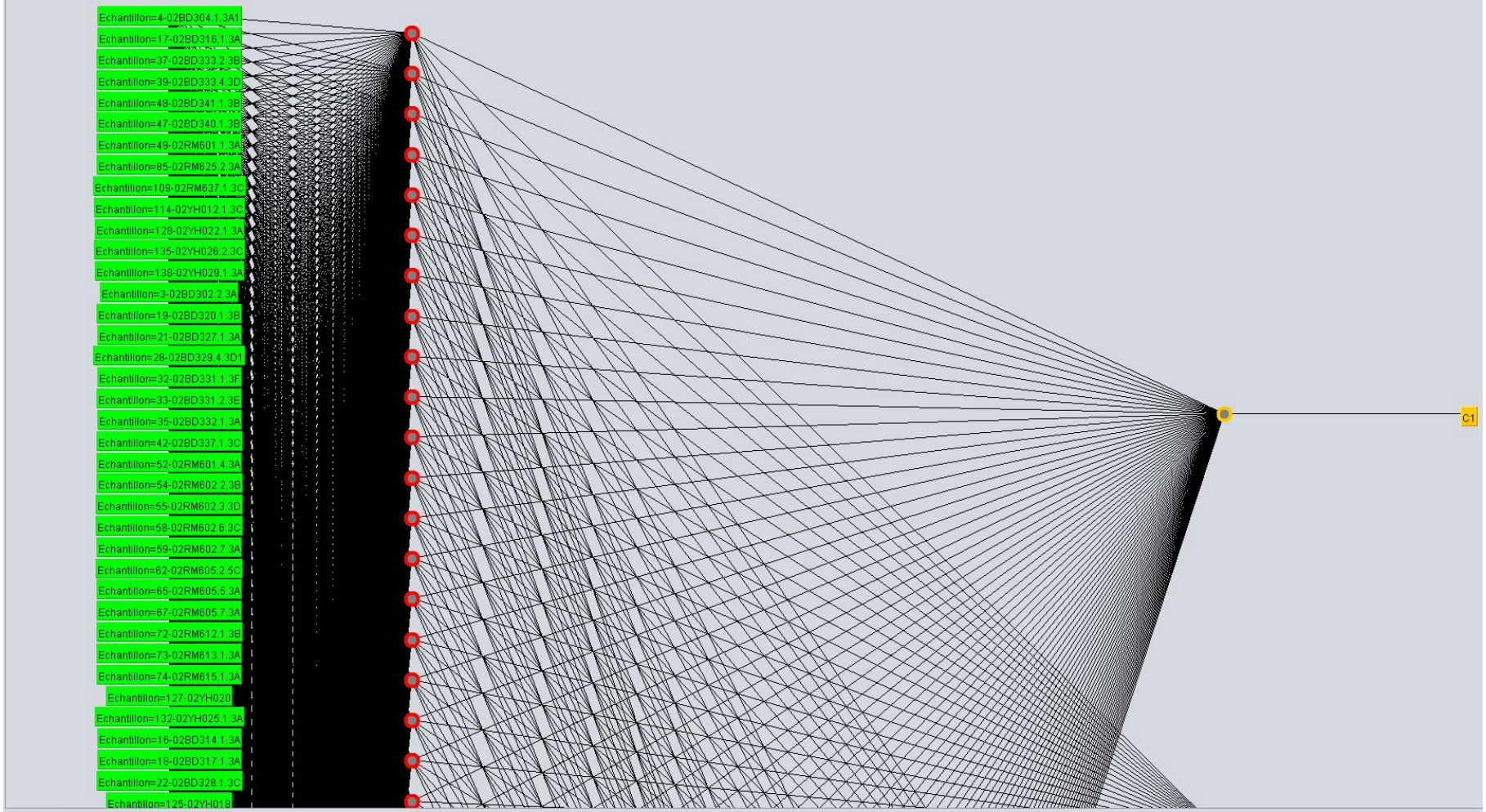
Epoch 0

Num Of Epochs 500

Error per Epoch = 0

NN Set: Experiment #2

- Predictive Accuracy: 86.8687%
- To build neuron network we used: “MultilayerPerceptron” with “Rename NominalValues” used prior to making NN.
- Had GUI set to true and Cross-Validation set of Folds of k=10
- Epoch set to 500 and implemented 10 fold (“start” X 10 and “accept” X 10)



Controls

Start

Accept

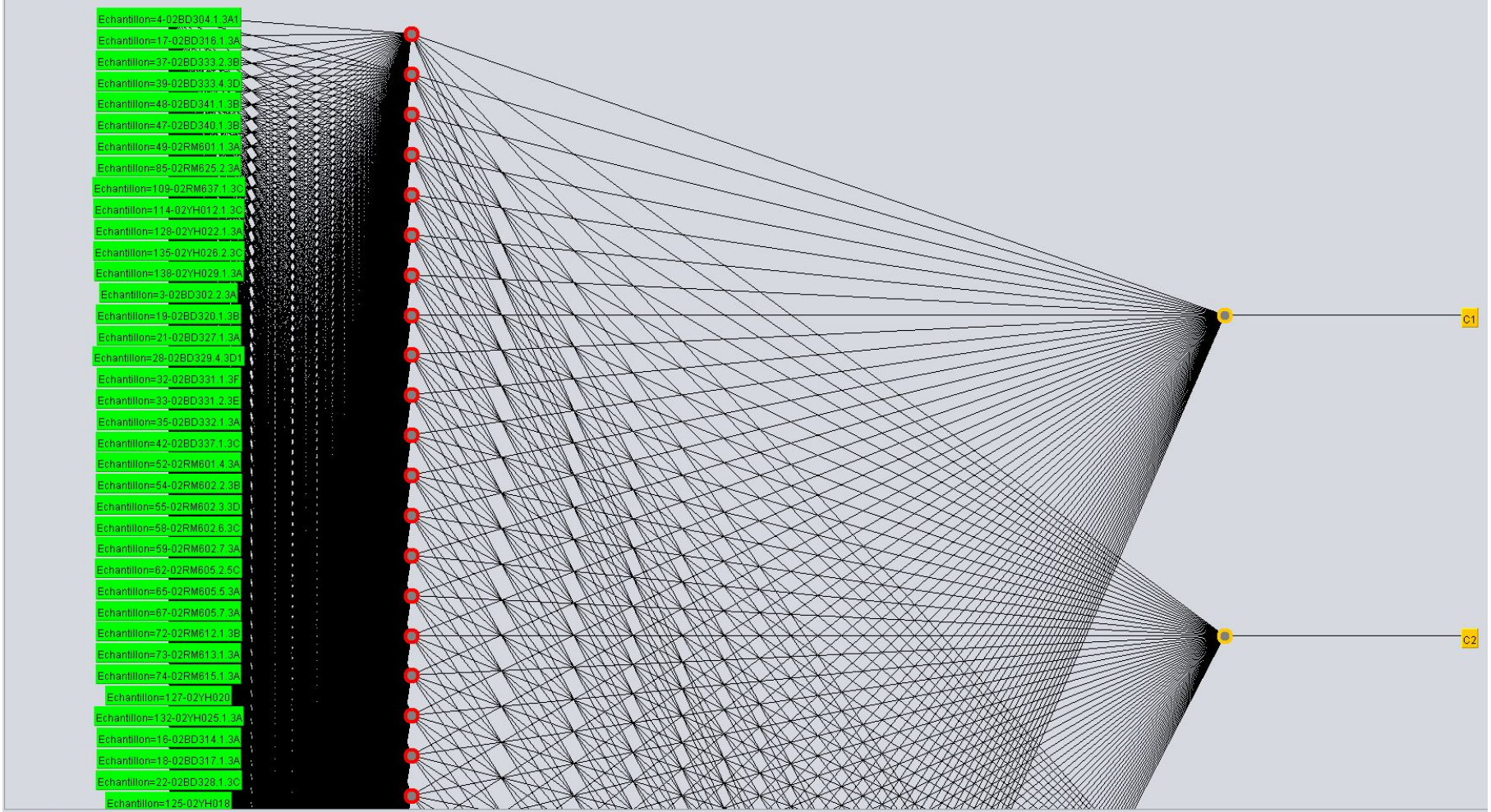
Epoch 500

Num Of Epochs 500

Error per Epoch = 0.0000287

NN Set: Experiment #3

- Predictive Accuracy: 87.8788%
- Same parameters used like experiment 1 and 2 with just important elements
- Epoch set to 500 and implemented 10 fold (“start” X 10 and “accept” X 10)



Controls

Start

Accept

Epoch 0

Num Of Epochs 500

Error per Epoch = 0

Summary of Predictive Accuracy

	Data #1	Data #2
Experiment #1	84.8485%	75.7576%
Experiment #2	83.8384%	84.8485%
Experiment #3	81.8182%	84.8485%

Summary of Predictive Accuracy

	Data Set
Experiment #1	86.8687%
Experiment #2	86.8687%
Experiment #3	87.8788%

Analysis

- Dataset #1 was carried out with equal frequency bins
- Dataset #2 was carried out with equal width bins
- The higher accuracy can be misleading due to the high volume of data however in the decision tree, dataset #2 overall had a higher accuracy
- WEKA produces different rules depending on the methods used for data preparation.
- Decision Trees: Depending on the sets used, different experiments yielded higher accuracies. (Experiment 1 for Data Set #1 and Experiment 2&3 for Data Set #2)
- Neural Network: Experiment 3 regarding a select set of attributes yielded the highest number for accuracy