Rock Classification using Descriptive and Non-Descriptive Classifiers

Final Project Report. CSE 634 - Spring '18.
Abhinav Adarsh, Deepak Gupta, Jatin Garg, Kiranmayi Kasarapu, Vipul Gandhi
Computer Science Department, Stony Brook University

1. ABSTRACT

Data mining is a process of inferring knowledge from huge data set. Data Mining has three major components Clustering or Classification, Association Rules and Sequence Analysis. By simple definition, in classification/clustering we analyze a set of data and generate a set of grouping rules which can be used to classify future data. Data mining is the process to extract information from a data set and transform it into an understandable structure. It is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. The actual data mining task is the automatic or semi-automatic analysis of large quantities of data to extract previously unknown interesting patterns. Data mining involves six common classes of tasks. Anomaly detection, Association rule learning, Clustering, Classification, Regression, Summarization. Classification is a major technique in data mining and widely used in various fields. Classification is a data mining technique used to predict group membership for data instances. Several major kinds of classification method including decision tree induction, Bayesian networks, K-nearest neighbor classifier, etc. In this project, we built descriptive and non-descriptive classifiers types. For the descriptive classifiers, we used Decision tree to generate sets of discriminant rules which describes the content of data. For non-descriptive classifiers, we used Neural Networks.

2. INTRODUCTION

Classification is used to classify each item in a set of data into one of predefined set of classes or groups. The task of classification is where a model or classifier is constructed to predict categorical labels (the class label attributes). Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. For example, a classification model could be used to identify loan applicants as low, medium, or high credit risks. A classification task begins with a data set in which the class assignments are known. For example, a classification model that predicts credit risk could be developed based on observed data for many loan applicants over a period of time. In addition to the historical credit rating, the data might track employment history, home ownership or rental, years of residence, number and type of investments, and so on. Credit rating would be the target, the other attributes would be the predictors, and the data for each customer would constitute a case. Classifications are discrete and do not imply order. Continuous, floating-point values would indicate a numerical, rather than a categorical, target. A predictive model with a numerical target uses a regression algorithm, not a classification algorithm. The simplest type of classification problem is binary classification. In binary classification, the target attribute has only two possible values: for example, high credit rating or low credit rating. Multiclass targets have more than two values: for example, low, medium, high, or unknown credit rating. In the model build (training) process, a classification algorithm finds relationships between the values of the predictors and the values of the target. Different classification algorithms use different techniques for finding relationships. These relationships are summarized in a model, which can then be applied to a different data set in which the class assignments are unknown. Classification has many applications in customer segmentation, business modeling, marketing, credit analysis, and bio medical and drug response modeling.

3. BACKGROUND

3.1 Decision Tree

Decision tree learning is a method commonly used in data mining. The goal is to create a model that predicts the value of a target variable based on several input variables. Each interior node corresponds to one of the input variables. There are edges to children for each of the possible values of that input variable. Each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf.

Decision tree induction algorithms function recur-

sively. First, an attribute must be selected as the root node. In order to create the most efficient i.e. smallest tree, the root node must effectively split the data. Each split attempts to pare down a set of instances (the actual data) until they all have the same classification. The best split is achieved with the attribute having the highest information gain.

The basic algorithm for decision tree induction is a greedy algorithm that constructs decision trees in a top-down recursive, divide-and-conquer manner. The Tree induction algorithm, summarized as follows.

1. The algorithm operates over a set of training instances, C.
2. If all instances in C are in class P, create a node P and stop, otherwise select a feature or attribute F and create a decision node.
3. Partition the training instances in C into subsets according to the values of V.
4. Apply the algorithm recursively to each of the subsets C.

These algorithms usually employ a greedy strategy that grows a decision tree by making a series of locally optimum decisions about which attribute to use for partitioning the data. For example, Hunt's algorithm, id3, c4.5, cart, sprint are greedy decision tree induction algorithms.

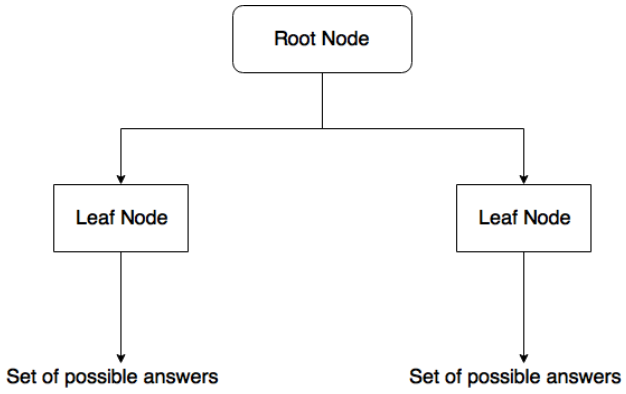

Figure 1: Decision Tree

3.1.1 J48 Algorithm

Decision tree J48 is the implementation of algorithm ID3 (Iterative Dichotomiser 3) developed by the WEKA project team. The J48 Decision tree classifier follows a simple algorithm. In order to classify a new item, it first creates a decision tree based on the attribute values of the available training data. Whenever it encounters a set of items (training set), it identifies the attribute that discriminates the various instances most distinctively. The feature with the highest information gain classifies the training set in the best way. Now, among the possible values of this feature, if there is any value for which there is no ambiguity, that is, for which the data instances falling within its category have the same value for the target variable, then we terminate that branch and assign to it the target value that we have obtained.

For the other cases, we then look for another attribute that gives us the highest information gain. Hence we continue in this manner until we either get a clear decision of what combination of attributes gives us a particular target value, or we run out of attributes. In the event that we run out of attributes, or if we cannot get an unambiguous result from the available information, we assign this branch a target value that the majority of the items under this branch possess.

Now that we have the decision tree, we follow the order of attribute selection as we have obtained for the tree. By checking all the respective attributes and their values with those seen in the decision tree model, we can assign or predict the target value of this new instance.

3.2 Neural Networks

Neural networks are nonlinear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data. Using neural networks as a tool, data warehousing firms harvest information from datasets in the process known as data mining. The difference between these data warehouses and an ordinary database is that there is actual manipulation and cross-fertilization of the data helping users makes more informed decisions.

3.2.1 Multi-layer Perceptron

Multi-layer Perceptrons (MLPs) are arguably the most popular neural network model used for pattern recognition. It is a class of feedforward artificial neural network. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

3.2.2 Activation Function

If a multilayer perceptron has a linear activation function in all neurons, that is, a linear function that maps the weighted inputs to the output of each neuron, then linear algebra shows that any number of layers can be reduced to a two-layer input-output model. In MLPs

some neurons use a nonlinear activation function that was developed to model the frequency of action potentials, or firing, of biological neurons.

The two common activation functions are both sigmoids, and are described by

$$y(v_i) = tanh(v_i)\ and \tag{1}$$

$$y(v_i) = (1 + e^{-v_i})^{-1} \tag{2}$$

The first equation is a hyperbolic tangent that ranges from -1 to 1, while the other is the logistic function, which is similar in shape but ranges from 0 to 1. Here y_i is the output of the i^{th} node (neuron) and v_i is the weighted sum of the input connections. Alternative activation functions have been proposed, including the rectifier and softplus functions. More specialized activation functions include radial basis functions which are used in radial basis networks, which are another class of supervised neural network models.

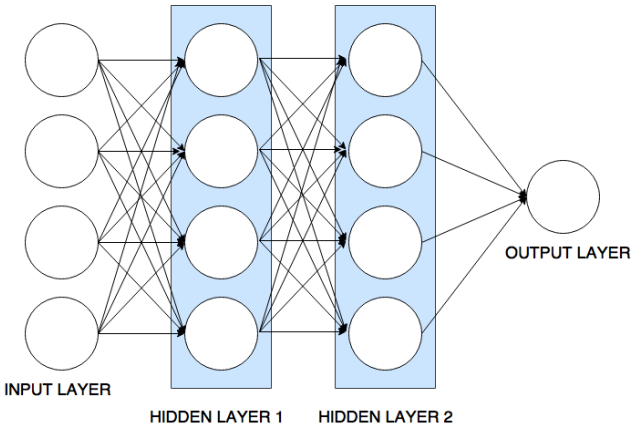

Figure 2: Neural Network

3.2.3 *Layers*

The MLP consists of three or more layers (an input and an output layer with one or more hidden layers) of nonlinearly-activating nodes making it a deep neural network. Since MLPs are fully connected, each node in one layer connects with a certain weight w_{ij} to every node in the following layer.

3.2.4 *Learning*

Learning occurs in the perceptron by changing connection weights after each piece of data is processed, based on the amount of error in the output compared to the expected result. This is an example of supervised learning, and is carried out through backpropagation, a generalization of the least mean squares algorithm in the linear perceptron.

4. DATASET

We are provided with a real life classification data set which has *Type De Roche* (Rock Type) as a *Class* attribute. There are 98 records with 48 attributes and 6 classes. Classes included are:

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

Most important attributes as determined by the expert for the provided data set are: S, Zn, Pb, Cu, CaO+MgO, CaO, MgO and Fe2O3. This is a real life experimental data and it contains some missing values(NaN).

5. DATA PREPARATION AND PREPROCESSING

We identified attributes containing a lot of NaN values and performed cleaning by removing the attributes having more than 60 percent missing values. For the rest of the attributes, in order to amputee missing values, we replaced them with mean of each column. After applying mean, we found out that few attributes with large values was affecting the mean and thus, presented a bad choice. We, therefore used mode of each column (attribute) for handling missing values. Moreover, we changed the data type of each attribute from string to numeric for performing arithmetic computations. After the above process, we got the data in a usable form, we will call it *Project Data*− **PD**.

Now, for the preprocessing part, to be used for decision trees descriptive classifier, we used PD and applied two methods of discretization, resulting in two data sets, namely **PD1** and **PD2**. We used supervised binning with max 5 bins and equal width for PD1 and equal frequency binning using unsupervised binning for PD2.

6. METHOD

For building the classifiers, we used the data sets PD, PD1 and PD2 that we got after performing cleaning and preprocessing. For each sets of data PD1, PD2 (for Decision Trees), and PD (for Neural Networks), we performed the following Experiments 1, 2 and 3.

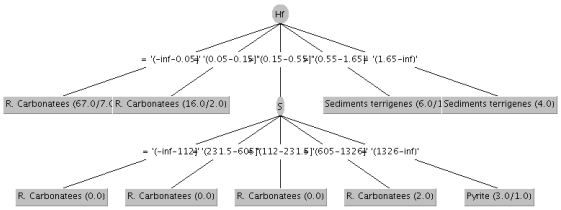

Figure 3: Decision Tree for data set PD1 used for Experiment 1

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          91               92.8571 %
Incorrectly Classified Instances         7                7.1429 %
Kappa statistic                          0.8025
Mean absolute error                      0.0289
Root mean squared error                  0.1423
Relative absolute error                 21.5921 %
Root relative squared error             56.9003 %
Total Number of Instances               98

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.987    0.095    0.974      0.987    0.981      0.908    0.909     0.947     R. Carbonatees
               0.750    0.032    0.500      0.750    0.600      0.593    0.871     0.760     Pyrite
               0.000    0.010    0.000      0.000    0.000      -0.015   0.979     0.500     Chalcopyrites
               1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Galene
               0.000    0.000    ?          0.000    ?          ?        0.223     0.027     Spahlerite
               1.000    0.011    0.900      1.000    0.947      0.943    0.994     0.900     Sediments terrigenes
Weighted Avg.  0.929    0.077    ?          0.929    ?          ?        0.898     0.900

=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
 76  0  0  0  0  1 |  a = R. Carbonatees
  0  3  1  0  0  0 |  b = Pyrite
  0  2  0  0  0  0 |  c = Chalcopyrites
  0  0  0  3  0  0 |  d = Galene
  2  1  0  0  0  0 |  e = Spahlerite
  0  0  0  0  0  9 |  f = Sediments terrigenes
```

Figure 4: Summary for Experiment1 (PD1)

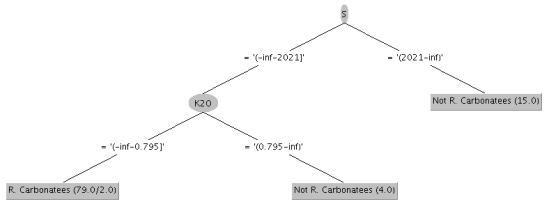

Figure 5: Decision Tree for data set PD1 used for Experiment 2

- Experiment 1: Use all records to perform the full classification (learning), i.e. build a classifier for all classes C1-C6 simultaneously.
- Experiment 2: Use all records to perform the contrast classification (contrast learning), i.e. contrasting class **C1** with a class **notC1** that contains other classes.
- Experiment 3: repeat Experiments 1, 2 for all records with the most important attributes as defined by the expert only.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          95               96.9388 %
Incorrectly Classified Instances         3                3.0612 %
Kappa statistic                          0.9075
Mean absolute error                      0.0514
Root mean squared error                  0.1774
Relative absolute error                 15.0778 %
Root relative squared error             43.1931 %
Total Number of Instances               98

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.987    0.095    0.974      0.987    0.981      0.908    0.908     0.946     R. Carbonatees
               0.905    0.013    0.950      0.905    0.927      0.908    0.908     0.880     Not R. Carbonatees
Weighted Avg.  0.969    0.078    0.969      0.969    0.969      0.908    0.908     0.932

=== Confusion Matrix ===

  a  b   <-- classified as
 76  1 |  a = R. Carbonatees
  2 19 |  b = Not R. Carbonatees
```

Figure 6: Summary for Experiment2 (PD1)

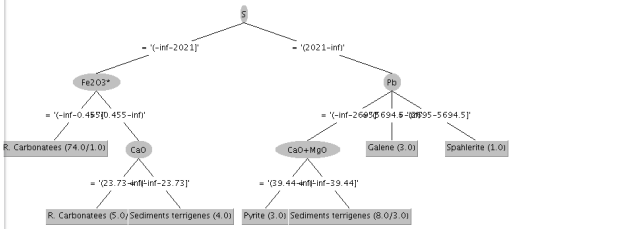

Figure 7: Decision Tree for data set PD1 used for Experiment 3 (Most Important Attributes, Exp1)

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          90               91.8367 %
Incorrectly Classified Instances         8                8.1633 %
Kappa statistic                          0.7772
Mean absolute error                      0.0381
Root mean squared error                  0.1596
Relative absolute error                 28.482  %
Root relative squared error             63.8712 %
Total Number of Instances               98

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.974    0.095    0.974      0.974    0.974      0.879    0.944     0.972     R. Carbonatees
               0.750    0.000    1.000      0.750    0.857      0.861    0.866     0.760     Pyrite
               0.000    0.000    ?          0.000    ?          ?        0.938     0.183     Chalcopyrites
               1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Galene
               0.000    0.000    ?          0.000    ?          ?        0.505     0.041     Spahlerite
               1.000    0.067    0.600      1.000    0.750      0.748    0.964     0.595     Sediments terrigenes
Weighted Avg.  0.918    0.081    ?          0.918    ?          ?        0.931     0.885

=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
 75  0  0  0  0  2 |  a = R. Carbonatees
  0  3  0  0  0  1 |  b = Pyrite
  0  0  0  0  0  2 |  c = Chalcopyrites
  0  0  0  3  0  0 |  d = Galene
  2  0  0  0  0  1 |  e = Spahlerite
  0  0  0  0  0  9 |  f = Sediments terrigenes
```

Figure 8: Summary for Experiment3 using PD1 (Most Important Attributes, Exp1)

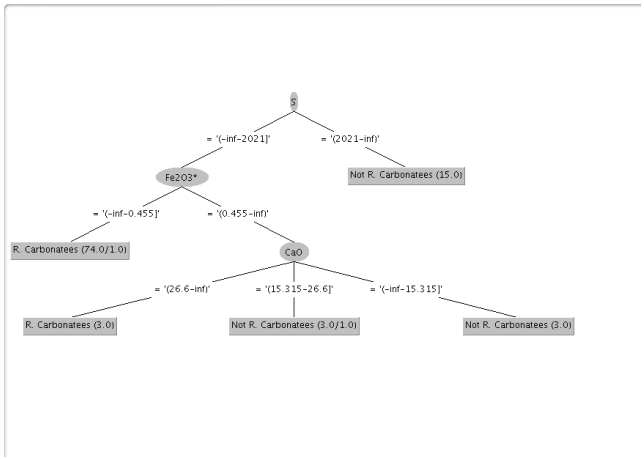

Figure 9: Decision Tree for data set PD1 used for Experiment 3 (Most Important Attributes, Exp2)

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          87               88.7755 %
Incorrectly Classified Instances        11               11.2245 %
Kappa statistic                          0.6369
Mean absolute error                      0.0429
Root mean squared error                  0.1883
Relative absolute error                 32.1031 %
Root relative squared error             75.3708 %
Total Number of Instances               98

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.987    0.429    0.894      0.987    0.938      0.676    0.874     0.944     R. Carbonatees
               0.000    0.000    ?          0.000    ?          ?        0.332     0.051     Pyrite
               1.000    0.010    0.667      1.000    0.800      0.812    0.995     0.833     Chalcopyrites
               0.333    0.000    1.000      0.333    0.500      0.571    0.744     0.680     Galene
               0.000    0.000    ?          0.000    ?          ?        0.867     0.151     Spahlerite
               0.889    0.011    0.889      0.889    0.889      0.878    0.994     0.960     Sediments terrigenes
Weighted Avg.  0.888    0.338    ?          0.888    ?          ?        0.861     0.875

=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
 76  0  1  0  0  0 |  a = R. Carbonatees
  4  0  0  0  0  0 |  b = Pyrite
  0  0  2  0  0  0 |  c = Chalcopyrites
  1  0  0  1  0  1 |  d = Galene
  3  0  0  0  0  0 |  e = Spahlerite
  1  0  0  0  0  8 |  f = Sediments terrigenes
```

Figure 10: Neural Network summary for Experiment 1

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          87               88.7755 %
Incorrectly Classified Instances        11               11.2245 %
Kappa statistic                          0.6051
Mean absolute error                      0.1192
Root mean squared error                  0.306
Relative absolute error                 34.9865 %
Root relative squared error             74.5135 %
Total Number of Instances               98

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.987    0.476    0.884      0.987    0.933      0.639    0.944     0.984     R. Carbonatees
               0.524    0.013    0.917      0.524    0.667      0.639    0.944     0.844     Not R. Carbonatees
Weighted Avg.  0.888    0.377    0.891      0.888    0.876      0.639    0.944     0.954

=== Confusion Matrix ===

  a  b   <-- classified as
 76  1 |  a = R. Carbonatees
 10 11 |  b = Not R. Carbonatees
```

Figure 11: Neural Network summary for Experiment 2

For each experiment, we compared the resulting descriptive classifiers with each other and compare each descriptive classifier with the resulting non-descriptive classifier.

7. RESULTS

7.1 Decision Tree model:

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          87               88.7755 %
Incorrectly Classified Instances        11               11.2245 %
Kappa statistic                          0.6185
Mean absolute error                      0.0442
Root mean squared error                  0.1889
Relative absolute error                 33.0667 %
Root relative squared error             75.5983 %
Total Number of Instances               98

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               1.000    0.476    0.885      1.000    0.939      0.681    0.915     0.970     R. Carbonatees
               0.000    0.000    ?          0.000    ?          ?        0.606     0.067     Pyrite
               1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Chalcopyrites
               0.333    0.000    1.000      0.333    0.500      0.571    0.695     0.678     Galene
               0.000    0.000    ?          0.000    ?          ?        0.874     0.148     Spahlerite
               0.778    0.011    0.875      0.778    0.824      0.809    0.981     0.920     Sediments terrigenes
Weighted Avg.  0.888    0.375    ?          0.888    ?          ?        0.902     0.895

=== Confusion Matrix ===

  a  b  c  d  e  f   <-- classified as
 77  0  0  0  0  0 |  a = R. Carbonatees
  4  0  0  0  0  0 |  b = Pyrite
  0  0  2  0  0  0 |  c = Chalcopyrites
  1  0  0  1  0  1 |  d = Galene
  3  0  0  0  0  0 |  e = Spahlerite
  2  0  0  0  0  7 |  f = Sediments terrigenes
```

Figure 12: Neural Network summary for Experiment 3-1(Most important attributes)

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          86               87.7551 %
Incorrectly Classified Instances        12               12.2449 %
Kappa statistic                          0.5942
Mean absolute error                      0.1167
Root mean squared error                  0.3029
Relative absolute error                 34.2782 %
Root relative squared error             73.7721 %
Total Number of Instances               98

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.961    0.429    0.892      0.961    0.925      0.607    0.945     0.985     R. Carbonatees
               0.571    0.039    0.800      0.571    0.667      0.607    0.945     0.861     Not R. Carbonatees
Weighted Avg.  0.878    0.345    0.872      0.878    0.870      0.607    0.945     0.958

=== Confusion Matrix ===

  a  b   <-- classified as
 74  3 |  a = R. Carbonatees
  9 12 |  b = Not R. Carbonatees
```

Figure 13: Neural Network summary for Experiment 3-2(Most important attributes)

For dataset PD1, we got accuracy of 92%. 7 data points were wrongly classified which included classes C3 and C5 for Experiment 1. For Experiment 2, we got accuracy of 96.9%. All the C1 class points were correctly classified except 1 and from class "notC1", two points were wrongly classified. For Experiment 3, we repeated the experiments 1 and 2 with most important attributes as defined by the experts. We got accuracy of 91.8% for Experiment3-1 and class C3, C5 were the only wrongly classified ones. For Experiment3-2, model gave 93.8% accuracy. Class C6 was always classified correctly in Experiment1 and Experiment3-1.

For dataset PD2, we got accuracy of 81%. Classes C1 and C6 are classified correctly most of the time whereas the remaining classes are incorrectly classified. For Experiment 2, we got accuracy of 82.6%. Both of the classes had some incorrect instances. For Experiment 3, we repeated the experiments 1 and 2 with most important attributes as defined by the experts. We got accuracy of 79.5% for Experiment3-1 and almost all the classes got at least one data point incorrectly classified. For Experiment3-2 model gave 88.7% accuracy and "notC1" class had half of the values incorrectly classified.

7.2 Neural network model:

	PD1	PD2	Neural Net
Experiment1	TP rate: 92.9 % FP rate: 7.7 %	TP rate: 81.6 % FP rate: 41.7 %	TP rate: 88.8 % FP rate: 33.8 %
Experiment2	TP rate: 96.9 % FP rate: 7.8 %	TP rate: 82.7 % FP rate: 46.3 %	TP rate: 88.8 % FP rate: 37.7 %
Experiment3 (Exp. 1)	TP rate: 91.8 % FP rate: 8.1 %	TP rate: 79.6 % FP rate: 38.3 %	TP rate: 88.8 % FP rate: 37.5 %
Experiment3 (Exp. 2)	TP rate: 93.9 % FP rate: 8.6 %	TP rate: 88.8 % FP rate: 30.8 %	TP rate: 87.8 % FP rate: 34.5 %

Table 1: Results for Experiment 1,2 and 3. [TP : True Positive, FP : False Positive]

For dataset PD, we got accuracy of 88.70%. Classes C2 and C5 are classified incorrectly all the time. For Experiment 2, we got accuracy of 88.77%. 50% of "notC1" was classified incorrectly. For Experiment 3, we repeated the experiments 1 and 2 with most important attributes as defined by the experts. We got accuracy of 88.77% for Experiment3-1 and class C2 and C5 are incorrectly classified whereas class C1 is correctly classified for all data points. For Experiment3-2 model gave 87.75% accuracy and half of the "notC1" class's data points were incorrectly classified.

8. CONCLUSION

We built descriptive and non-descriptive type classifiers on the Bakary dataset. For the descriptive classifiers, we used Decision tree to generate sets of discriminant rules which describes the content of data. When descriptive classification model is applied, we got the best accuracy of in Experiment 2 which gave 96.9%. For non-descriptive classifiers, we used Neural Networks and the best accuracy was received for Neural Networks with 88.7%.

9. BIBLIOGRAPHY

[1] G. Kesavaraj, S. Sukumaran*A study on classification techniques in data mining*

[2] D.R. Hush, *Classification with Neural Networks: A Performance Analysis*

[3] Multilayer perceptron.
https://en.wikipedia.org/wiki/Multilayer perceptron