



Evaluation of Knowledge Extraction Using Various Classification Data Mining Techniques

Shabia Shabir Khan*, Mushtaq Ahmed Peer

Department of Computer Science

Kashmir University, India

Abstract— *The advantageous tool of data mining used to extract the nuggets of knowledge from a database / data warehouse [1] has gained a lot of attention from the research and commercial field. This paper presents evaluation of various classification data mining algorithms using WEKA (Waikato Environment for Knowledge Analysis) and concentrates on the values of certain important evaluation measures. In Section III and IV, we have performed evaluation experiment on the basic data mining algorithms using WEKA. In addition to this, comparative study has been recorded in section V as detailed in conclusion.*

Keywords— *Data Mining, Data warehouse, Classification Technique, Evaluation Measures.*

I. INTRODUCTION

Data mining is considered to have been originated from three branches of artificial intelligence --neural networks, machine-learning and genetic algorithms that has lead us to great analytical advancement [2],[3],[4],[9]. It is actually the process of uncovering the hidden patterns or trends (Knowledge extraction) in the data that are not immediately apparent by just summarizing the data. It is used to predict the future (predictive analytics) in addition to explain the current or past situation (descriptive analytics) by providing the answer to “How” or “why”. One of the important techniques of data mining is the Classification technique that groups the data into a certain number classes on the basis of their differences. Various Classification based algorithms provided by open source tool WEKA have been used for evaluation on real data set in this paper and the comparison made between them, as recorded in table, taking certain important evaluation measures into consideration.

II. DATA MINING – CLASSIFICATION TECHNIQUE

The classification process groups the data into the classes on the basis of their differences. Some of the classification techniques or classifiers are the Decision Tree Classifier, Neural Network Classifier, Naïve Bayes Classifier and so on. Each of these techniques use the learning algorithm that generates the model that best fits the relationship between the predictors (attributes for prediction) and the prediction (class). The main aim of each of these techniques is to provide a model that accurately predicts the class of the unknown tuples or records. Given steps below represent basic principle of working for each of these classifiers which is same:

- i. Provide the training set that consists of the training records along with their associated class label.
- ii. The Classification model is built by applying the learning algorithm used in respective technique.
- iii. The model built is applied on the test set that consists of the tuples that do not have the associated class label.

As far as training data is concerned, we can go for the cross-validation that involves the partitioning of the training data into mutually exclusive and same-sized subsets. Analysis is performed on one subset which is termed as the training set and the validation of the analysis is done using the other subset termed as the validation set or testing set. This is the case of simple one round cross-validation; however we can go for multiple rounds or fold cross-validation that can be performed using different partitions in an attempt to reduce the variability. In the software tool ‘WEKA’, we can specify the number of folds we need. For n-fold cross-validation, the data is randomly divided into ‘n’ subsets or folds of equal size. Further, the model is then trained using n-1 folds and is tested using one-nth fold. Repeating the process n times so that all folds are used for testing provides the n test sets that are used for the performance computations. Such a process uses all the data for both training and testing in an effective way [5].

The performance of the data can be checked using 2 important measures/ metrics:

Accuracy= (No. of correct predictions)/ (Total No of predictions)

Error rate= (No. of wrong predictions)/ (Total No of predictions)

Various data mining algorithms have been provided for extracting the useful information from the data that are categorised under Rule based Techniques (ZeroR & OneR), Bayes Theorem based Technique (Naïve Bayes), Neural Network based Technique (Multi-Layer Perceptron) and Decision tree based Technique(J48 and Random Forest).

III. DATA MINING USING WEKA OPEN SOURCE SOFTWARE:

WEKA open source software consists of a collection of data mining learning algorithms and data preprocessing (transforming of dataset using filters) tools. It has been developed at the University of Waikato in New Zealand. In the evaluation, we have used graphical user interface of WEKA 3.7.7 downloaded from website <http://www.cs.waikato.ac.nz>. It accepts the data in specific formats such as ARFF (Attribute-Relation File Format (ARFF)), CSV and C4.5's format. We shall start with the evaluation of a simple arff file "weather.arff". The sample dataset was directly loaded in the WEKA software using the 'open URL' option (<http://www.hakank.org/weka/weather.arff>).

Applying the basic technique of classification (ZeroR) in the next tab of WEKA user Interface named 'classify', we started the process and following results were obtained:

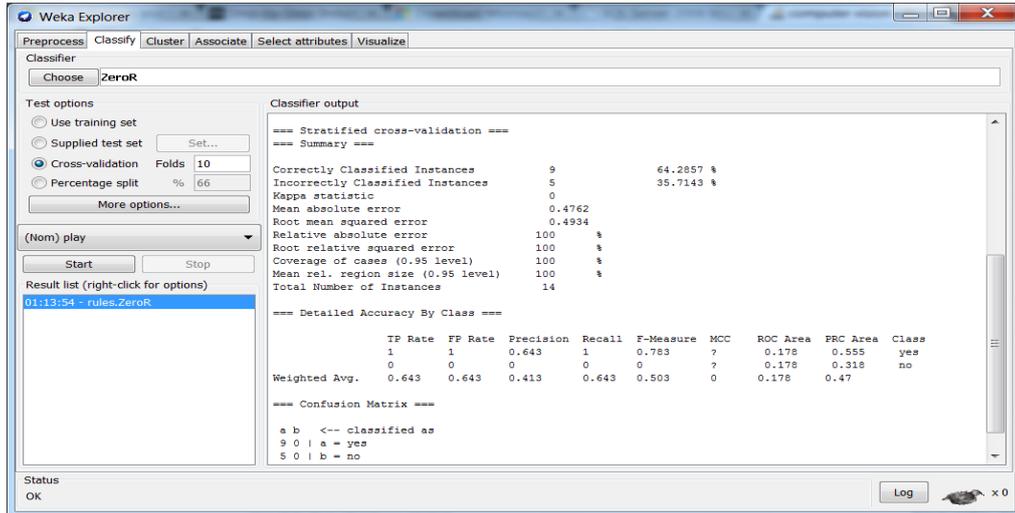


Fig 1. Evaluating the classification method on "Weather.arff" using ZeroR Classifier.

A. Evaluation measures:

The description of each measure here is shown below:

- The correctly classified instances show %age of test instances that were correctly classified (Accuracy),
- The incorrectly classified instances show %age of test instances that were incorrectly classified (Error Rate).
- No. of Instances = 14
Correctly classified: $aa + bb = 9 + 0 = 9$
Incorrectly classified: $ab + ba = 5 + 0 = 5$
- Accuracy = $(9/14) * 100 = 64.2857\%$
Error Rate = $(5/14) * 100 = 35.7143\%$
- The Contingency table or confusion matrix, with class 'a' and class 'b' is the representation of the number of instances, correctly or incorrectly classified.
- Kappa is the normalized statistic measure of agreement that is calculated by taking the agreement expected by chance away from the observed agreement between the classifier and actual truth and dividing by the maximum possible agreement. The possible value for Kappa lies in the range [-1, 1] although this statistics usually falls between 0 and 1. The value of '1' perfect agreement i.e. Full on agreement upon the classification process by the rater, however, the value of '0' indicate the agreement no better than expected by chance. So, when the value of k is greater than 0, it means that the classifier is doing better compared to chance thus indicating perfect agreement at $K=1$ else if the value of k is 0, then it denotes the chance agreement. A kappa with the negative rating indicates worse agreement than that expected by chance [6].

- Kappa Statistic $K = (P_A - P_E) / (1 - P_E)$ Where,
 P_A is the Observed %age Agreement, and P_E is the Chance (hypothetical) %age Agreement
In the above case, for calculating k, we have:

$$\text{Total Instances} = TP + FP + TN + FN$$

$$P_A = (aa + bb) / \text{Total Instances} = 9 / 14 = 0.64$$

$$P_E = (\text{Pr (Predicted A)} * \text{Pr (Actual A)}) + (\text{Pr (Predicted B)} * \text{Pr (Actual B)})$$

$$= (14/14 * 9/14) + (0/14 * 5/14) = 0.64$$

$$\text{Then, } K = (0.64 - 0.64) / (1 - 0.64) = 0 \text{ (Chance/ Random / Hypothetical Agreement)}$$

- ROC (Receiver Operating Characteristics) Analysis is used for analyzing and illustrating the performance of various systems by using the four basic types / groups of classification :
True Positive (TP) – Correct Positive Prediction
False Positive (FP) – Incorrect Positive Prediction

True Negative (TN) – Correct Negative Prediction
 False Negative (FN) – Incorrect Negative Prediction

- We would always like to have FP and FN as zero. The two important measures that ROC Analysis takes into consideration are:
 True Positive Rate (TPR): Ability or Recall (R) of a Classifier to correctly classify Positive Instances.

$$TPR = \frac{TP}{TP+FN}$$
 False Positive Rate (FPR): Inability of a Classifier to correctly classify negative instances.

$$FPR = \frac{FP}{FP+TN}$$
- We have another metric that is used in many applications that determines the actual fraction of records that are declared to be under Positive Class. Such a metric is Precision (P)
 Precision, $P = \frac{TP}{TP+FP}$
 Precision is inversely proportional to the Classifier's False Positive Errors

$$F\text{-measure} = \frac{2 * R * P}{R+P}$$
- ROC Space / Area: It shows the performance of a Classifier as a single point in the 2-D space which is a representation of the two important measures of ROC Analysis (TPR and FPR) wherein the Y axis represents the TPR and X axis represents the FPR [7].

B. Training sets and Tests sets in WEKA:

In order to test the efficiency of our learning models we use training and test sets. In the supervised learning we provide the training set to build a learning model and further, we provide the test set so as to check the performance. For this we divide our dataset into two, usually disjoint, subsets. One subset will act as a training set and the other as test set. The training set, which is used to build a predictive model, consists of the predictor attributes as well as the prediction (class label) attribute. On the other hand we have the unseen test set, which is without any class label and is used to check the performance of the model trained. Besides this we have the Validation set that is used to validate the trained model i.e. estimate how good the trained model is. In WEKA, the data set can be divided into training and test set. Also, we can go for cross-validation process where same sized disjoint sets are created so as to train the model fold wise. K-fold cross-validation, (usually k=10) is used to divide the data into equally sized K subsets/ folds. In such case the model is trained using (k-1) folds and the Kth fold is used as test set. The whole process is repeated k times in an attempt to use all the folds for testing thus allowing the whole of the data to be used for both training and testing. The real dataset "credit.g.arff" (German credit dataset) has been uploaded from the URL (<http://repository.seasr.org/Datasets/UCI/arff/>).

The next step is to split the "credit.g.arff" dataset into 40% testing set and 60% training set. For this we use the WEKA filter – "Randomize" Filter so as to create a random permutation. Further, another filter "Remove Percentage" is applied two times. First by keeping option "invert Selection" as 'false' and then 'true' so as to keep the 40% of the dataset saved as a test set and rest as the training set, respectively. Further, we applied the "Remove Percentage" filter following by which, we get two datasets:

The "creditTraining.arff" with 80% of the instances in the original datasets

The "creditTest.arff" with 20% of the instances in the original data

As the number of instances in original dataset were 1000, the training subset will contain 800 instances while as the test set will contain 200 instances.

We start by using the training set in the preprocess panel. This is followed by the selection of the particular algorithm we are concerned with. The 10 fold cross validation option is being selected.

Next to use our sets in the experiments we choose the training set and move to the "Classify" panel and choose the procedure that we have to use and start the experiment. After that we apply the same procedure on our testing set to check what it predicts on the unseen data. For that, we select "supplied test set" and choose the testing dataset that we created. We run the algorithm again and we notice the differences in accuracy.

IV. CLASSIFICATION ALGORITHMS USED FOR EVALUATION:

A. Rule-Based Techniques:

We have two important rule based techniques that have been used in software tools like WEKA.

1) ZeroR Classifier::

The rule behind this algorithm is the consideration of the majority or common class of training data set to be taken as the real Zero R prediction. So, it relies on the target prediction and ignores all predictors. There is no predictability power of Zero R algorithm; however it is used to determine a baseline performance that acts as a benchmark for the other classification methods. The classification technique used above, as shown in figure 5.3, is the ZeroR technique. Applying the same classifier on "creditTraining.arff" training dataset. Following results have been provided:

TABLE I
 EVALUATION OF ZEROR CLASSIFIER ON TEST DATASET

Experiment using ZeroR Classifier	
Time taken to test model on supplied test set:	0.02 seconds
Correctly Classified Instances	69 %
Incorrectly Classified Instances	31 %

Kappa statistic	0
Root mean squared error	0.4626
Precision	0.476
Recall	0.69
F-measure	0.563

2) *OneR Classifier:*

The rule behind the algorithm is to find the single attribute that best predicts the class of the data. It generates a one-level decision tree and infers accurate rules that are easy to interpret. It works by creating one rule for each attribute in the training data and selects among them the best /one rule with the smallest/ lowest error rate. The error rate of a rule is the number of training data instances in which the class of an attribute value does not match for that attribute value in the rule. In case the error rate for the rules is same, then the One- rule is chosen at random.

Another classifier is the OneR Classifier, which after applying on the training dataset “creditTraining.arff” produces following values for the evaluation measures.

TABLE II
EVALUATION OF ONER CLASSIFIER ON TEST DATASET

Experiment using OneR Classifier	
Time taken to test model on supplied test set:	0.01 seconds
Correctly Classified Instances	64 %
Incorrectly Classified Instances	36 %
Kappa statistic	-0.0017
Root mean squared error	0.6
Precision	0.571
Recall	0.64
F-measure	0.587

B. *Bayes Theorem-Based Technique:*

Bayes' Theorem states that, if the probability of any event A conditional on event B is to be obtained, then calculate the probability of both A and B together and divide it by the probability of B. This is stated as follows:

$$\Pr (B | A) = \Pr (A \text{ and } B) / \Pr (A)$$

Where P (B|A) is the Conditional probability (posterior probability) denoting the probability of B given that A has already occurred. This kind of classifier, along with the Class labels, provides relative probabilities, thus expressing the decision confidence.

It helps in predicting the class of evidence or a data tuple X by the Bayesian formula, shown below:

$$\Pr (C_i | X) = (\Pr (X | C_i) * \Pr (C_i)) / \Pr (X)$$

Naïve Bayes:

The Naive Bayes Classifier is a simple probabilistic type of classifier that is based on the concept of Bayes theorem with naïve or strong independence assumptions [8]. Applying the Naive Bayes Classifier on “creditTraining.arff ” training dataset , following measures have been recorded

TABLE III
EVALUATION OF NAIVEBAYES CLASSIFIER ON TEST DATASET

Experiment using NaiveBayes Classifier	
Time taken to test model on supplied test set	0.02 seconds
Correctly Classified Instances	75.5 %
Incorrectly Classified Instances	24.5 %
Kappa statistic	0.4298
Root mean squared error	0.4058
Precision	0.756
Recall	0.755
F-measure	0.756

C. *Neural Network Based Technique:*

ANN is the simulation of the Biological Neural System. Its structure has been inspired by the biological nerve cells called neurons that are linked to other neuron by the connective fiber strands called axons. An ANN comprises of interconnected collection of nodes and the directed links. The simplest of all ANNs is the Perceptron that consists of two

kinds of nodes or neurons or units. They are the input nodes (representing input attributes) and the output nodes. Each input node is being connected to the output node by the weighted link. The training of such an ANN amounts to adapting the respective weights for the links until they accurately predict the class.

Multi-Layer Perceptron:

A more complex structure than that of the Perceptron model is the multilayer ANN, where, in addition to the input and output layers, we have the intermediary hidden layer in between that consist consisting of hidden units/ nodes.

Applying Multi-Layer Perceptron Classifier on training data set “creditTraining.arff” following values have been seen, as shown in Table IV.

TABLE IV
EVALUATION OF MULTI PERCEPTRON CLASSIFIER ON TEST DATASET

Experiment using Multilayer Perceptron Classifier	
Time taken to test model on supplied test set:	0.03 seconds
Correctly Classified Instances	68.5 %
Incorrectly Classified Instances	31.5 %
Kappa statistic	0.2797
Root mean squared error	0.5153
Precision	0.692
Recall	0.685
F-measure	0.688

D. Decision Tree-Based Technique:

Decision Tree Classifier is the hierarchical structure, consisting of nodes and the directed edges, that organizes a series of questions about the predictors (attributes) and their possible answers in an effective way.

1) *J48 (C4.5 DT)*

The J48 Decision tree classifies a new item by creating a decision tree on the basis of the predictor value. It selects that attribute / predictor value that has the highest information gain and best classifies the data instances the highest information gain Applying J48 Decision Tree based Classifier on training data set “creditTraining.arff”, as shown in Table V.

TABLE V
EVALUATION OF J48 CLASSIFIER ON TEST DATASET

Experiment using J48 Classifier	
Time taken to test model on supplied test set:	0.03 seconds
Correctly Classified Instances	69.5 %
Incorrectly Classified Instances	30.5 %
Kappa statistic	0.2434
Root mean squared error	0.4818
Precision	0.679
Recall	0.695
F-measure	0.684

2) *Random Forest:*

Random Forest Classifier works on the kind of technique wherein the predictions are taken from multiple decision trees (base classifiers). Table VI shows the values of evaluation measures after applying Random Forest Classifier on dataset.

TABLE VI
EVALUATION OF RANDOM FOREST CLASSIFIER ON TEST DATASET

Experiment using Random Forest Classifier	
Time taken to test model on supplied test set:	0.01 seconds
Correctly Classified Instances	74.5 %
Incorrectly Classified Instances	25.5 %
Kappa statistic	0.3791
Root mean squared error	0.4254
Precision	0.736
Recall	0.745
F-measure	0.739

V. COMPARISON FROM WEKA FOR VARIOUS ALGORITHMS:

Following table shows the comparison of various evaluation measures between the classifiers used:

TABLE VII
COMPARISON BETWEEN THE CLASSIFICATION ALGORITHMS.

Evaluation Measures	ZeroR	OneR	Naïve Bayes	Multilayer Perceptron	J48	Random Forest
Time taken to test model	0.02 sec	0.01 sec	0.02 sec	0.03 sec	0.03 sec	0.01 sec
Correctly Classified Instances	69 %	64%	75.5%	68.5%	69.5%	74.5%
Incorrectly Classified Instances	31 %	36%	24.5%	31.5%	30.5%	25.5%
Kappa statistic	0	-0.0017	0.4298	0.2797	0.2434	0.3791
Root mean squared error	0.4626	0.6	0.4058	0.5153	0.4818	0.4254
Precision	0.476	0.57 1	0.756	0.692	0.679	0.736
Recall	0.69	0.64	0.755	0.685	0.695	0.745
F-measure	0.563	0.58 7	0.756	0.688	0.684	0.739

VI. CONCLUSIONS

The values for the evaluation measures might be different for the different data sets used and accordingly the algorithms may perform in the different way for the different datasets. In the Table VII of Section V, we could clearly see that for the same real dataset (*German Credit dataset*), different algorithms worked in a different ways. Naïve Bayes Classifier has shown the highest percent of correctly classified Instances (75%) which is followed by Random Forest (74.5%). As far as F-measure is concerned, Naïve bayes Classifier gave the highest value of all i.e. 0.756 which is followed by value 0.739 in case of Random Forest. Taking these evaluation measures into consideration, we could easily recommend naïve Bayes Classifier as the best Classifier for Credit dataset. However this may not be same for all the datasets. A general Classifier needs to be built that should be adaptable to the different types of the datasets

REFERENCES

- [1] Fayyad, U.M.,Shapiro,G.P.,Smyth,P., Uthurusamy,R., “Advances in Knowledge Discovery and Data Mining” ,*The MIT Press*, (1996)
- [2] Chen, M.S., Han, J., and Yu, P.S., “Data mining: An overview from a database perspective” ,*IEEETransactions on Knowledge and Data Engineering*, 8(6):866-883,1996
- [3] Agrawal R, Srikant R , “Mining sequential patterns”, In: Yu P, Chen A (Eds) *Proceedings of the eleventh international conference on data engineering, Taipei, Taiwan* , pp 3–14, March 1995
- [4] Schumaker, R.P. et al., “Sports Data Mining Methodology, Sports Data Mining, Integrated”, In:*Information Systems 26, Springer Science+Business Media, LLC* 2010
- [5] Han, J., Kamber, M., and Pei, J., “Data Mining: Concepts and Techniques”, 3rd edition, Morgan Kaufmann, (2011); (1st ed., 2000-2001) (2nd ed., 2006)
- [6] Bartko,J.J, Carpenter, W.T, “On the methods and theory of reliability”, *J NervMent Dis.*1976 ;163:307–317
- [7] Tan, P.N. , Steinbach, S. , Kumar, V. , “Introduction to Data Mining” , *Pearson Education* , Edition -2007
- [8] K. P. Murphy, "Naive Bayes classifiers Generative classifiers," Bernoulli, pp. 1-8, 2006.
- [9] Shabia Shabir, M.A.Peer, “Expedition for the exploration of Apposite Knowledge”, *IJCSIT-2012(IJCSIT) International Journal of Computer Science and Information Technologies*, Vol. 3 (5) , pages 5164 – 5168, (2012)