Volume 4, Issue 12, December 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

RaajHans: A Data Mining Tool for Integer Database

Mr. Avinash R. Pingale, Ms. Aparna A. Junnarkar
Department of Comp. Engg., Savitribai Phule UoP,
Pune, Maharashtra, India

Abstract— RaajHans is a smart data mining tool which is able to mine structured integer data. It has Classification, Association and Clustering as its data mining operations. For clustering operation, it uses Fuzzy C-means algorithm which guarantees a more accurate solution than traditional algorithms. For classification purpose, it uses Naïve-Bayes classifier algorithm and Association uses Apriori algorithm. RaajHans can be open source software application. Users can customize its algorithms as per their needs. Initially it is designed for structured integer data, but user can customize it to operate on unstructured integer data. Raajhans is not limited for certain database. It can work on any kind of purely integer database.

Keywords— Genetic Algorithms, Data Mining, FCM, Database Converter, ARFF

I. INTRODUCTION

The idea of RaajHans immerged from the need of Data mining on the data whose nature in today's era changes wisely. RaajHans basically provides three data mining operations as Clustering, Classification and Association. It accepts purely integer database with some attributes. Pre-processor of RaajHans is a comprehensive tool for pre-processing the data set. RaajHans takes ARFF files or integer Database as input. If ARFF file is provided as input to RaajHans, it directly recognizes attributes and instances out of the database. ARFF files are Attribute Relation File Format. As ARFF file contains attributes and its relations with comma separated format, it is quite convenient to perform data mining operation on such ARFF files. Hence RaajHans converts integer database into ARFF. Weka treats values separated by comma from first line of the ARFF as attributes. Example is shown in figure 1. Whereas for performing Data mining operations

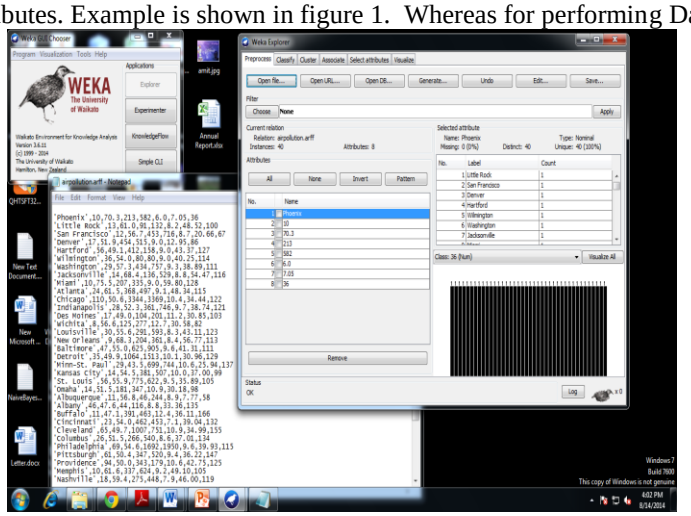

Fig 1.Attribute selection in Weka

RaajHans detects '@attribute' string and space. Whatever comes after that can be treated as attribute. After selecting attributes, pre-processor provides it to further data mining operations. Attribute selection is provided in form of GUI that makes it bit easier for user to select attribute and view instances of the data set. If text database is given as input it is been converted to its equivalent ARFF file. Database must be supposed to be a table which should have some column attributes. The table given as input must have compatible values in it. Table records must not remain empty. Whenever table is supplied as an input it determines number of column attributes and values of the column attributes. After fetching attributes it can easily recognizes data related to respective attributes. Data is then arranged in manner under header '@data'. Attributes fetched are arranged under '@attribute' header of ARFF file. Data under '@data' header is arranged one after next separated using comma. Same attributes also placed under header '@attribute' one by one. If attributes possesses any relation among them, then the relations are also mentioned under '@ relation' header of ARFF. This pre-processed data then can be fed to perform Data mining operations.

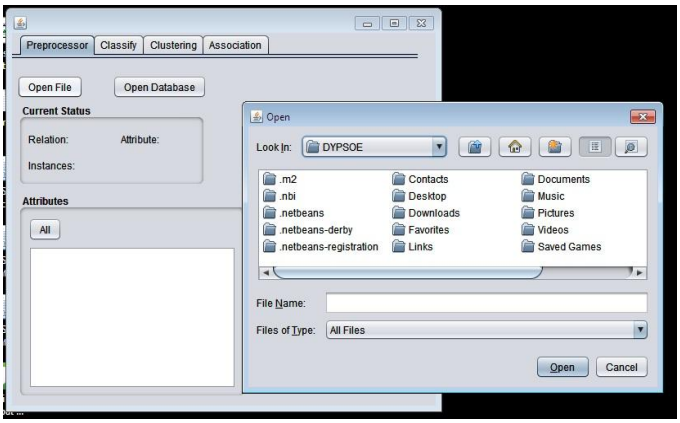

Fig.2 Pre-processor GUI

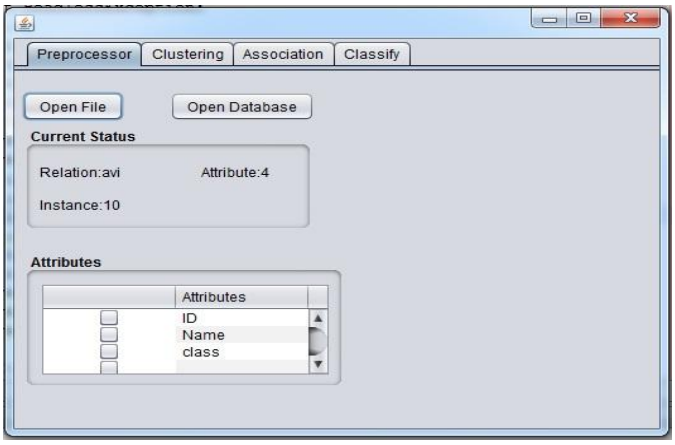

Fig.3 Status of opened file

When you will select database as an input, then RaajHans will ask you to save database as ARFF file. Once database is saved as ARFF, we are ready to perform data mining operations on it.

II. CLASSIFICATION

Classification is one of the most applied data mining technique which employs a set of pre-classified examples to develop a model that can classify the population of records at large[. Its applications mainly imply Fraud Detection, Risk Analysis and weather forecasting. Classification process involves learning and classification. For Classification, RaajHans is having a Naïve-Bayes Classification algorithm which is again based on supervised learning process. The operation of classification starts on ARFF file. The RaajHans recognizes the last attribute from the file and identifies total number of distinct items in that particular attribute. On basis of distinct values found, the classes are to be formed. As much distinct values are there, that many classes would be formed. This is the advantage of RaajHans that it identifies the attribute which can be divided into classes. The only restriction is, user must specify classes into the last attribute of the file. It only selects the last attribute and identifies possible classes out of it. This restriction comes because data in today's era is not having a unique structure or format. Anyone can place data according to their need. Hence RaajHans is also having its own format of data in which last attribute is considered as a collection of class. After determining classes it takes an element from test set and then finds the probability of belongingness of that particular item with the first class from the class set. Iteratively it will check belongingness with each class and then will create heap of the probabilities. The cluster with higher priority is selected and then the element is assigned to that particular cluster. RaajHans uses Naïve-Bayes Classification. Naïve-Bayes is a probabilistic model & the probability model for a conditional model. i.e P(C | F1- - - - Fn)

Over a dependant class variable C with a small number of outcomes or classes, conditional on several feature variables F1 through Fn. The problem is that if the number of features of n is large or if a feature can take on a large number of values, then putting such a model on a probability tables is infeasible. Hence we can use Baysian Model which uses independent classes.

$$i.e. P(C|F1---Fn) = \frac{P(C)P(F1---Fn|C)}{P(F1---Fn)}$$

Let us see how the data is being traversed thought the algorithm.

For Classification operation, RaajHans uses Naïve-Bayes Classifier. Naïve-Bayes Classifier can be trained very efficiently in a supervised learning process. Naïve-Bayes is a probabilistic model & the probability model for the classifier is a conditional model.

i.e. P(C|F1-------Fn)
As per discussion given above in section 2 we can use Naïve-Bayes instead to solve the problem.
i.e. P(C|F1---------Fn) =
(P(C)P(F1-----Fn | C))/P(F1----Fn)

In terms of Bayesian probability we can put above equation
Posterior=(Prior*Likelihood)/Evidence

Let's discuss the mathematical model for the same for the data being classified in RaajHans.
Let S be a system which represents a classifier.
S={X, Y, Fme, DD/NDD, Φ, Completeness}
Where,
X=set of input
Y=set of output
Fme= Function which describes a system
DD/NDD= will be the conclusion that whether problem is deterministic or non-deterministic in polynomial time.
Φ= Set of constants.
Completeness= will state the completeness of the problem.
Let us define above shown terms.
X={x1, x2, x3- - - - -xn }
Where xi is the attribute selected from the data set X.

For classification we are supposed to filter data to divide into some particular classes. Hence we are supposed to determine the classes first. Hence we should choose attribute that is further divided into classes depending upon number of distinct values from that particular attribute.

{C}= {xi | xi contains classes c1, c2, c3- - - - - ci}
Once we get classes, we can start classification.
Y= {c1, c2, c3, - - - - - ci}
Where ci is a class at instance i.

We have defined input & output. Let us discuss about external input to be provided. We have to provide a random observation. We call it as train data. Train data can be formed by choosing any number of attributes from the test data. We are interested in finding P(Ci | d) where d is observation.

Let us apply Naïve-Bayes theorem.

$$P(C_i|\, d) = \frac{P(d\,|C_i)\,P(C_i)}{P(d)}$$

Where
P(d | Ci)= Probability of generating instances 'd' given class Ci.
P(Ci)= Probability of occurrence of class Ci.
P(d)= Probability of occurrence of instance d.
Once we calculated above values we can calculate probability of belongingness of d to class Ci.

We should iteratively calculate the probability with each class. And then choose the probability which is higher amongst the array of probability. This can be done using bubble sort technique. Hence the series for bubble sort will become like
P(d | Ci)=(n(n+1))/2

If likeliness of d with class Ci is greater than others then it must be associated with Ci .
Hence we can say that Naïve-Bayes is a NP-Complete problem.

III. CLUSTERING

Clustering can be viewed as a technique of classifying objects which belongs to similar class [9]. That means those object shows properties matching to some particular class can be grouped into one cluster. For example a bank wants to choose customers who has loan in some particular financial year. In this example criteria of clustering can be defined by the bank itself. This criterion is nothing but the property of the object 'customer'. Clustering technique determines such objects and enlists them. The idea behind clustering is use of Fuzzy C-means algorithm rather than traditional one. Traditional algorithms provide results in terms of 0 and 1 [8]. They identify objects and check them for relevancy. Relevancy can be determined in terms of 0 and 1. If objects belongs to some cluster then its value is set as 1 else 0. Problem with traditional clustering algorithms is their result can change if the size of database changes [9]. For example k-means algorithm can alter its performance when dataset size gets changed [3]. It also can give redundant result when density of the data set gets changed. These problems can be overcome by using Fuzzy C-means algorithm who is successor of k-means algorithm. K-means algorithm is updated with some parameters and it is then converted to Fuzzy C-means. Fuzzy C-means is a method of allocating a data point to clusters not hard and fast but fuzzy. In hard clustering data might be divided into distinct clusters and data element belongs to exactly one cluster., whereas fuzzy clustering states that data elements can belong to one or more clusters and each element is associated with a membership function. The FCM algorithm tries to partition a finite set of elements X={x1, x2, x3...., xn} into set of Fuzzy clusters on basis of

some given criterion. The algorithm returns a list of c cluster centres. One thing FCM does in advance is to minimize an objective function. We can select and objective function which found minimum and then can assign that particular item to the corresponding cluster in the cluster group. By using FCM we can determine maximum reliability of the item set that it can belong to the most suitable cluster.

RaajHans uses FCM for clustering hence we will consider data and its flow according to the flow of operation of FCM.

Let

S be the system (Fuzzy C-means algorithm) and cane be defined as:

S={X, Y, Fme, DD/NDD, Completeness, Φ}

Where

X= wet of input

Y=set of output

Fme= A function which states the system

In case of FCM, input is a dataset which contains text records.

X={x1, x2, x3...xn} be a record of dataset X

{x1.....xn} may bind with each with some relation and that relation can vary according to dataset. This set of records collectively form a dataset, hence we can say that

X={x1Ux2U.....Uxn}

Now let c is an integer $2 \leq c \leq n$, a conventional c-partition, hence we can define output Y as:

Y= {Y1, Y2.....Yc}

Where Y must satisfy following conditions for each cluster.

$Yi \neq \Phi \quad 1 \leq i \leq c$ -------- (a)

$Yi \cap Yj = \Phi \quad i \neq j$ --------- (b) $\bigcup_{i=1}^{c} Yi = Y$

Yi is a cluster in Y.

We are interested in finding out membership function for each sample chosen from the sample space (In our case, from a dataset X)

Hence

$J_m = \sum_{i=1}^{N} \sum_{j=1}^{c} u_{ij}^{m} \|x_i - c_j\|^2$ ---------- (A)

Where,

Jm =A objective function which we are supposed to minimize.

c = number of clusters in Y $2 \leq c \leq n$

m = weighing component.

uij= degree of membership of xi in the cluster j.

Next state of the algorithm will be finding out a membership function for each item in the dataset.

Let

$$u_{ij} = \frac{1}{\sum_{k-1}^{c} \left[\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right]^{2/(m-1)}}$$

Let's call above equation as (B)

In above equation cj is next cluster centre and xi is the item at ith position which we are supposed to find.

Next cluster centre cj can be calculated as

$$c_j = \frac{\sum_{i-1}^{N} u_{ij}^{m} x_i}{\sum_{i-1}^{N} u_{ij}^{m}}$$

This iteration of finding cj will stop when

maxij {|u_ij^((k+1))-u_ij^((k)) |}<∈

Where ∈ is a termination criterion between 0 and 1 and k is iteration steps.

Above function (B) will generate different uij for each iteration (i.e. for each item i selected from dataset X)

Hence,

$$Y = \left\{ \bigcup_{i=j=1}^{N} u_{ij} \right\}$$

Above equation gives set of output that is set of membership function.

In cluster module pre-processed files is been taken as input and FCM will be applied on that file. Output generated from FCM is been displayed on the Cluster Output window. Start and stop buttons can be used to start and stop data mining operation which is very useful when data contains missing values and can go into infinite state. In future we can use pool of algorithms to perform the clustering operation.

IV. ASSOCIATION

Association is the next model in the hierarchy of the RaajHans. Association is one of the data mining operations. For association operation RaajHans uses Apriori algorithm. Apriori algorithm is a technique built for learning association rules. In RaajHans it is used for generating frequent item sets. Frequent item sets are calculated on basis of 'minsupport'. User should provide a minimum support for items which are supposed to occur frequently. We will then select such items

which satisfies minimum support condition. Then Algorithm will move further and checks that items can be grouped or not. If items can be grouped then again we will check for the minimum support condition. We will traverse through algorithm until we finish all the possible combinations of the items in a transaction.

Apriori algorithm can be applied to the pre-processed data. Then it will ask user for minimum support. On basis of minimum support provided by the user, Association module will display appropriate output in the 'Association Output' text area. Association data mining is greatly applicable in the field of marketing where companies are supposed to figure out the product minimum support and confidence. Analysis of sell and purchase can be made easy and effective using Association technique. Companies can figure out the status of the product in the market and support it is getting from the customer in each transaction.

For example, Flipakart an online shopping store wants to search out products which are frequently searched and bought by the customer. Example of flipkart is somehow same as association, but it doesn't use association in actual sense. Association generates the list of products which are searched by the user frequently and as well provides confidence of purchasing some particular product.

Association in RaajHans uses Apriori algorithm for the purpose of determining Association. Apriori algorithm states that "If an itemset is frequent then all of its subsets must also be frequent". Let us start by applying database and minsup to Apriori.

Let S be the system (Apriori Association) and cane be defined as:

S={X, Y, Fme, DD/NDD, Completeness, Φ}

Where

X= wet of input

Y=set of output

Fme= A function which states the system

In case of Apriori, input is a dataset which contains text records.

X={[x1, x2, x3...xn], minsup} be a record of dataset X

{x1.....xn} may bind with each with some relation and that relation can vary according to dataset. This set of records collectively form a dataset, hence we can say that

X={x1Ux2U.....Uxn}

DD/NDD= will be the conclusion that whether problem is deterministic or non-deterministic in polynomial time.

Φ= Set of constants.

Completeness= will state the completeness of the problem.

Minsup= minimum support provided for given dataset.

Y = {L1, L2, L3- - - - - Lk}

Where Lk = frequent item set.

Now we know that support for a given dataset can be calculated as

Support= N_x/N

Where,

Nx= No. of transactions of item x

N= Total no. of transactions.

Now, Apriori algorithm can have two distinct operations for finding minsup of the item x and confidence of item x simultaneously. Hence there will be two models of the algorithm exist and can together be seen as an Association operation.

$$M1 = \frac{N_x}{N} \quad ------(A)$$

&

$$M2 = \frac{supp(x_i \cup x_j)}{supp(x_i)} \quad ------(B)$$

Where x_i & x_j possesses a relation xi xj (i.e. if xi then xj)

Equation (A) gives minsupp of item x and Equation (B) gives confidence of item x at level k.

Equation (A) can execute for all items in the dataset D & form a candidate set Ck. Then all M1 should be compared with misuppwhich is given. If minsupp of anu item from candidate set Ck found less, then that item should be dropped from the Ck. We will keep finding Ck until no frequent item set found.

This searching will take k(k+1)/2 iterations.

Hence

$$Fme = \{\frac{N_x}{N}, \frac{supp(x_i \cup x_j}{supp(x_i)}\}$$

As the function M1 takes input from user, we can say that its nature remains constant, but output varies accordingly. Hence we can say that the Apriori is a non-deterministic problem. Both functions possess a complete solution, as we can determine the output to the system in polynomial time and it can be verified. Hence we can say that M1 and M2 possess NP-Complete solution.

V. HOW TO USE RAAJHANS

RaajHans has very user-friendly GUI which helps user to easily perform destined Data Mining operations. As RaajHans takes ARFF or database as input, user can directly feed ARFF file or a database to RaajHans. RaajHans only works on active ARFF file hence database is converted into ARFF. To do so user is only supposed to provide .frm file that is database table in SQL. This file is then automatically converted to ARFF and will be saved to destination folder of user's choice.

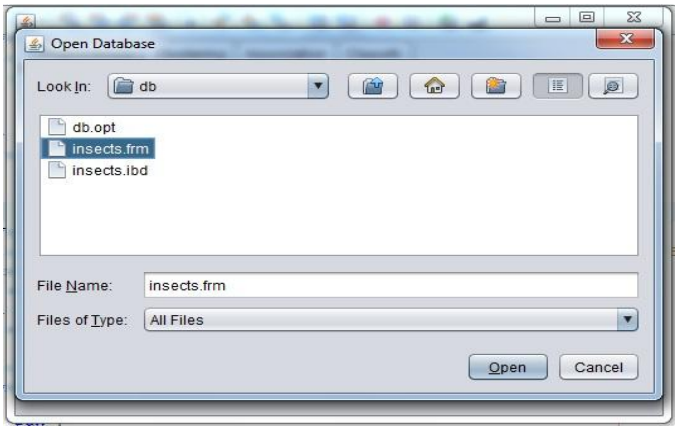

fig. 5 Opening .frm file to convert into ARFF

After opening a .frm file, it will ask user to save it to ARFF file. User is supposed to provide name for the ARFF file and click the save button. This will save ARFF file on the hard disk in the destination folder. After saving the ARFF, user should open it into RaajHans using button 'Open ARFF' shown in fig 2. Status of opened file can be seen under the 'Current Status' section. It will display name of the opened file, number of attributes present in the file and number of instances present. Now you are ready to apply any of the data mining operations on the file. To perform Clustering, user should navigate to the tab named 'Clustering'. Under it there are Start and Stop buttons which will help to start clustering operation. If a dataset is huge and needs some more time to execute then stop button can help to stop the operation.

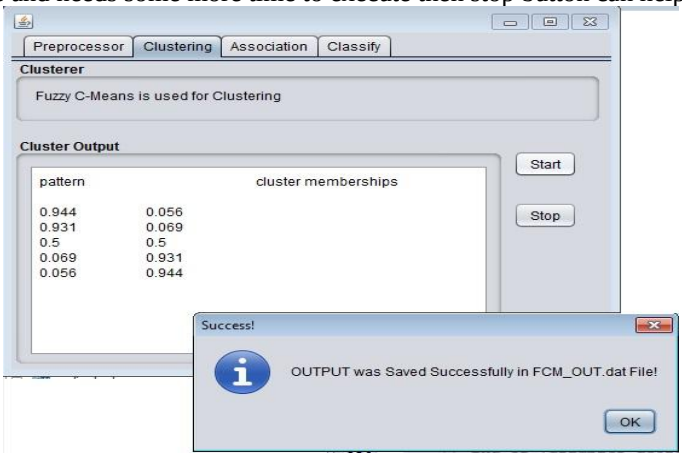

Fig. 6 Clustering Output

As shown in above figure, clustering gives pattern of clusters and its membership value with the cluster. For further use backup of output may be saved in FCM_OUT.dat file.

To use Association user need to just click Start button.

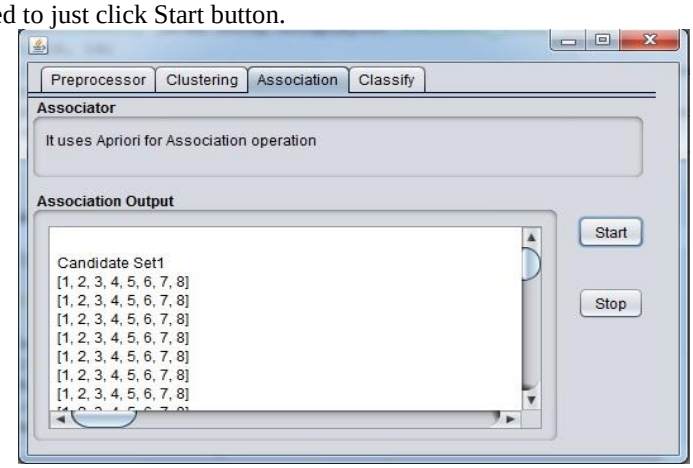

Fig. 7 Association Output

Above shown figure describes output of the Association. It displays possible candidate sets and finally shows possible frequent patterns that may be generated from the data set.

For the perspective of use of classification module, user should provide a train data set. On basis of train dataset, RaajHans will recognize possible classes out from dataset. After that user should provide data which he wants to test. We call such data as Test data. He should enter parameters of such Test data and RaajHans will immediately display to which class it may belong.

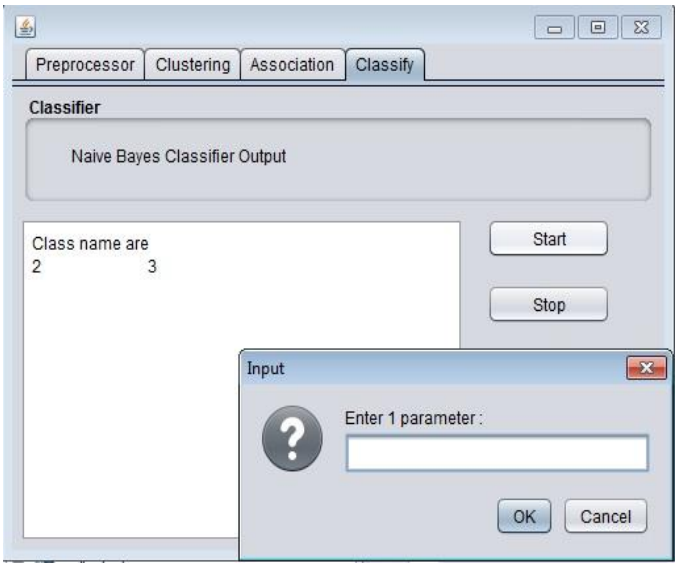

Fig. 8 Classification Output

VI. SYSTEM REQUIREMENTS

System requirements for RaajHans are quite basic but its performance can vary according to available buffer memory. The clustering algorithm needs a buffer space to store the temporary results. The memory needed can change according to supplied data and provided number of clusters. As much as dynamic memory available, that much speed of execution can be achieved.

Dual core CPU with 1.33 GHz speed is quite adequate for RaajHans to execute smoothly. Large RAM can make its speed of execution relatively faster. If other heavy applications are running in background and if they are using RAM then the performance of Raajhans can slow down for some amount of time. Though it slows down its performance cannot be hampered. Its output will remain inconsistent. Hard disk needed is depends on the user. If the requirement of the user is large that is, if he wants to manage large database, he can use hard disk as large as he wants.

VII. SUMMARY AND CONCLUSION

No doubt RaajHans is an effective too for mining the data. It has three data mining techniques and three algorithms for the same purpose. RaajHans pre-processor can pre-process database and ARFF files as well. Pre-processor is quite effective to generate ARFF from different database such as SQL, Oracle, xls. It has a user friendly GUI so that user can easily understand the nature of data and flow of data. RaajHans can be updated with efficient use of data mining algorithms which are based on self-learning techniques. It can also be updated for MongoDB and BIGDATA so that it can convert these databases into its equivalent ARFF file. The data in today's era is not generalized. Every organization maintains their data structures. Hence we cannot predict exact tuples and attributes out of it. Hence its pre-processor can be made flexible to read all data structures and convert them to RaajHans readable format.

REFERENCES

[1] Weka User Manual

[2] Ian H. Witten, Eieb Frank, Len Trigg 'Practical Machine Learning tools and Techniques with Java implementation', Mark Hall, Aug 1999

[3] Kanhaiya Lal, N.C. Mahanti 'Role of soft computing as a tool in data mining'. Department of Computer Sc. Engg., Birla Institute of Technology Patna, Bihar, India, Department of Applied Mathematics, Birla Institute of Technology Mesra, Ranchi, India.

[4] Sankar K. Pal, Soft data mining, computational theory of perceptions, and rough-fuzzy approach, Machine Intelligence Unit, Indian Statistical Institute Kolkata,India.17 March 2003.

[5] R. S. Michalski, M. Kubat, I. Bratko, Machine Learning and Data Mining: Methods and Applications, 1998.

[6] Akira Imada, 'An Introduction to Soft Computing, December 17, 2003

[7] Sushmita Mitra, Sankar K. Pal and Pabitra Mitra, Data Mining in Soft Computing Frame- work: A Survey IEEE transactions on neural networks, vol. 13, no. 1, january 2002.

[8] Mrs. Bharati M. Ramageri, Data mining techniques and applications Indian Journal of Computer Science and Engineering, Vol. 1 No. 4 301-305

[9] Data Mining: Concepts and Techniques, 2nd Edition, Jiawei Han and Micheline Kamber, Morgan Kauffman, 2006

[10] http://www-sal.cs.uiuc.edu/~hanj/bk2

[11] Denver Dash, Gregory F. Cooper , 'Exact model averaging with naive Bayesian classifiers', Decision Systems Laboratory, Intelligent Systems Program, University of Pittsburgh, Pittsburgh, PA 15213 USA

[12] http://www.cs.uml.edu/~jlu1/TA/DM_Fall2013/p1.html

[13] http://www2.cs.uregina.ca/~dbd/cs831/notes/ itemsets/itemset_apriori.html

[14] http://home.deib.polimi.it/matteucc/Clustering /tutorial_html/cmeans.html

