IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-03, Issue-2 E-ISSN: 2347-2693

Automated Assistance for Data Mining Implementation

Shinde Pratik Rajendra[1*], Bari Kalpesh Ramkrushna[2], Shinde Chandrakant Giridhar[3], Vishwasrao Gokul Jayvant[4] and Prof. S. N. Bhadane[5]

[1*,2,3,4,5]*Information Technology, Savitribai Phule Pune University, India*

*jacob.pratik@gmail.com, kalpesh.bari@hotmail.com, cs2366@gmail.com, viswasgokul27@gmail.com, satish.bhadane@gmail.com

www.ijcaonline.org

Received: Jan /09/2014 Revised: Jan/08/2014 Accepted: Jan/20/2014 Published: Jan/31/ 2014

Abstract— These days there are number of tools available for assistance in implementation of data mining. Some of them are RapidMiner, WEKA, Orange, Rattle, and KNIME. WEKA tool includes a number of techniques like classification, clustering, regression, etc. For solving classification problems this tool provide variety of strategies such as decision tree, neural networks, lazy classifiers etc. For each strategy, the tool allows the user to select specific values for large number parameters for e.g. in case of a neural network classifier, parameters are to be provided by user such as epochs, learning rate, momentum etc. With WEKA an expert user could study which strategy could be the best compatible for the any particular dataset. For that test run can be performed on the test dataset of the user using the various strategies and the resultant outputs are to be evaluated by the expert user its own. This is tedious task to be performed. This paper aims at developing a system that could work on its own even on the evaluation part and thus make it possible for effective implementation of data mining by developing a database to record the nature of data such as number and type of attributes, presence or absence of missing values etc. along with different values for developing classifier models and the accuracy of the classifier. Such a database can then be made available to the novice users to build a model based on past experience.

Keywords— Classification, data mining, WEKA, ARFF etc.

I. INTRODUCTION

In the real world the amount of data is overwhelming. The amount of data in our lives seems to go on and on increasing and there is no end in sight. The majority of the information is in unprocessed form. There are enormous amount of information that is hidden in the raw data. Data mining is the extraction of implicit, previously unknown, and potentially useful information from data. The idea is to build a computer program that can sift through databases automatically, seeking regularities or patterns. Several tools are available for solving data mining problems, both in open source and commercial category [1] [2]. One such tool is WEKA. Because WEKA is open source it is widely used by many organizations. WEKA provides variety of strategies such as decision tree, neural networks, lazy classifiers etc. [3]. For each strategy, the tools allow the user to select specific values for large number parameters for e.g. in case of a neural network classifier, parameters are to be provided by the user such as epochs, learning rate, momentum, etc. [4] Although default setting for such parameter is provided by tools, it is often found that the classifier performance (accuracy) can be enhanced by making series of experiments with different values for these parameters. Thus, for a novice user it is difficult to guess proper values for these parameters and the only option is to try with series of experiments which is time consuming. WEKA tool assists in the process but the evaluation part has to be carried out manually by the user itself. For that an expert user is required who could handle the tool and understand all the essential part of the system.

This project aims at developing a database that records performance of the classifiers and nature of data along with different values used for building classifier. Thus even a novice user could easily be part of the process for implementation of data mining and also an expert user will be relieved from the tedious task. This information is to be provided in a form that will guide even the novice users based on past experiments.

II. WEKA(WAIKATO ENVIRONMENT FOR KNOWLEDGE ANALYSIS

WEKA was developed at the University of Waikato in New Zealand [4] [3]. "WEKA" stands for the Waikato Environment for Knowledge Analysis [4]. The system is written in Java, an object oriented programming language

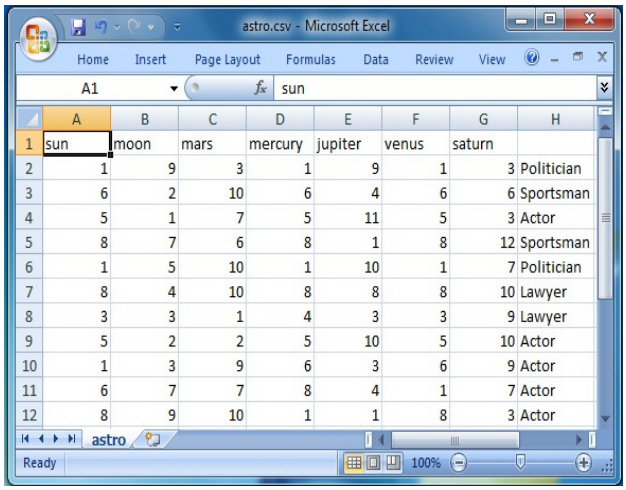
that is widely available for all major computer platforms, and WEKA has been tested under Linux, Windows, and Macintosh operating systems. Java allows us to provide a uniform interface to many different learning algorithms, along with methods for pre and post processing and for evaluating the result of learning schemes on any given dataset [4].

WEKA provides implementations of learning algorithms that you can easily apply to your dataset [4]. It also includes a variety of tools for transforming datasets, such as the algorithms for discretization [5].You can preprocess a dataset, feed it into a learning schema, and analyze the resulting classifier and its performance-all without writing any program code at all [4]. The workbench includes methods for all the standard data mining problems: regression, classification, clustering, association rule mining, and attribute selection. All algorithms take their input in the form of a single relational table in the ARFF format, which can read from a file or generated by a database query [3] [4].

One way of using WEKA is to apply a learning method to a dataset and analyze its output to extract information about the data. Another is to apply several learners and compare their performance in order to choose one for prediction. The learning methods are called classifiers [4].

Suppose you have some data and you want to build a decision tree from it. A common situation is for the data to be stored in a spreadsheet or database. However, WEKA expects it to be in ARFF format, because it is necessary to have type information about each attribute which cannot be automatically deduced from the attribute values. Before you can apply any algorithm to your data, it must be converted to ARFF form [4]. This can be done very easily. Most spreadsheet and database programs allow you to export your data into a file in comma separated format—as a list of records where the items are separated by commas. Once this has been done, you need only load the file into a text editor or a word processor; add the dataset's name using the @relation is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations. tag, the attribute information using @attribute, and a @data line; save the file as raw text—and that's it [4].

In the following example we assume that your data is stored in a Microsoft Excel spread sheet, and you're using Notepad for text processing. Of course, the process of converting data into ARFF format is very similar for other software packages. Figure 1 shows an Excel spread sheet containing the Astrology data.



	A	B	C	D	E	F	G	H
1	sun	moon	mars	mercury	jupiter	venus	saturn	
2	1	9	3	1	9	1	3	Politician
3	6	2	10	6	4	6	6	Sportsman
4	5	1	7	5	11	5	3	Actor
5	8	7	6	8	1	8	12	Sportsman
6	1	5	10	1	10	1	7	Politician
7	8	4	10	8	8	8	10	Lawyer
8	3	3	1	4	3	3	9	Lawyer
9	5	2	2	5	10	5	10	Actor
10	1	3	9	6	3	6	9	Actor
11	6	7	7	8	4	1	7	Actor
12	8	9	10	1	1	8	3	Actor

Fig.1. Database in excel spread sheet

It is easy to save this data in comma-separated format. First, select the Save As… item from the File pull-down menu. Then, in the ensuing dialog box, select CSV (Comma Delimited) from the file type popup menu, enter a name for the file, and click the Save button. (A message will warn you that this will only save the active sheet: just click OK.) Now load this file into Notepad [3] [4]. The screen will look like figure 2.

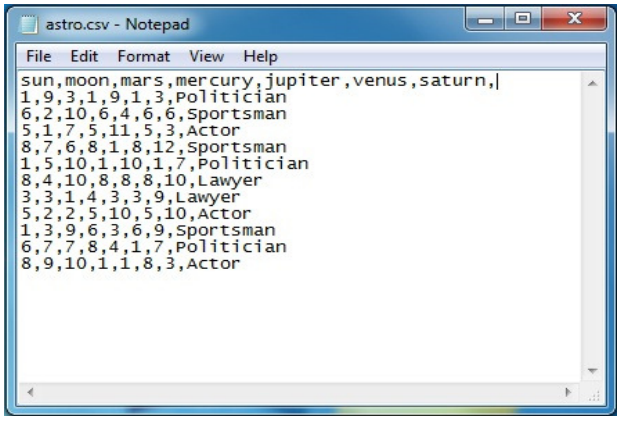


Fig: Data in csv format

The rows of the original spreadsheet have been converted into lines of text, and the elements are separated from each other by commas. All you have to do is convert the first line, which holds the attribute names, into the header structure that makes up the beginning of an ARFF file. Figure 3 shows the result. The dataset's name is introduced by a @relation tag, and the names, types, and values of each attribute are defined by @attribute tags. The data section of the ARFF file begins with a @data tag. Once the structure of your dataset matches Figure3, you should save it as a text file. Choose Save as… from the File menu, and specify Text Only with Line Breaks as the file type by using the corresponding popup menu. Enter a file name, and

press the Save button. Rename the file to weather.arff to indicate that it is in ARFF format. Note that the classification schemes in Weka assume by default that the class is the last attribute in the ARFF file, which fortunately it is in this case [3] [4].

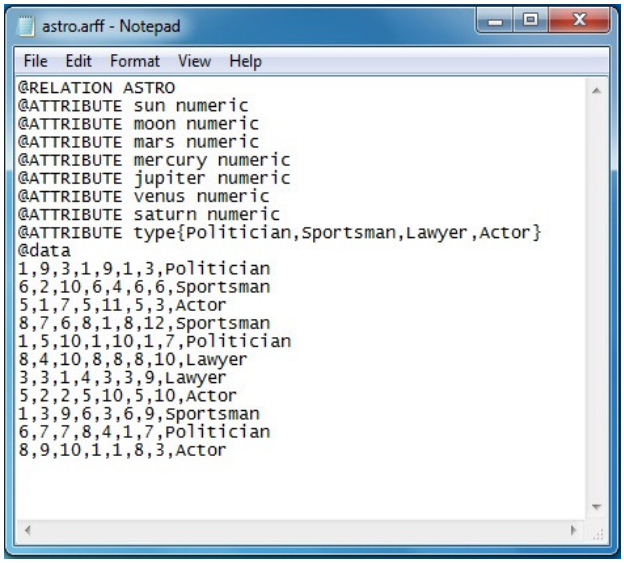


Fig.2. Data in arff format

III. LOADING DATA INTO THE EXPLORER

Let's load this data into the Explorer and start analyzing it. Start WEKA to get the panel shown in figure 4 Select Explorer from the four graphical user interface choices.

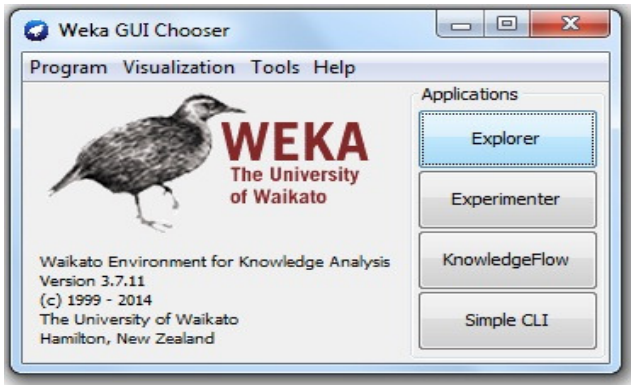


Fig.3. WEKA GUI

What you see next is the main Explorer screen, shown in figure 5. Actually, the figure shows what it will look like after you have loaded in the weather data. The six tabs along the top are basic operations that the explorer supports:

1) Preprocess: choose the dataset and modify it in various way.

2) Classify: train learning schemes that perform classification or regression and evaluate them.

3) Cluster: learn clusters for the dataset.

4) Associate: learn association rules for the data and evaluate them.

5) Select attributes: select the most relevant aspects in the dataset.

6) Visualize: view different two-dimensional plots of the data and interact with them.

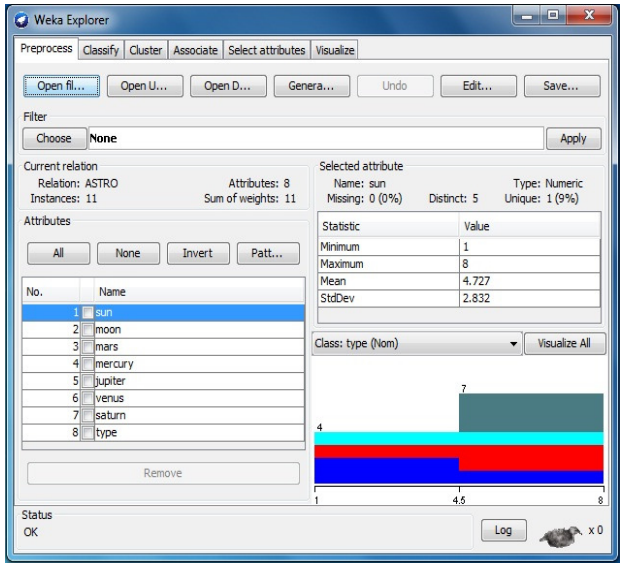


Fig.4. WEKA Explorer

IV. USING THE TEMPLATE

From the working explained in the previous section we could understand the role of human in the execution of the tool. The user has to provide a test dataset as an input to WEKA tool. This should be in arff format. Then select the classifier for test run, perform tests on as many classifiers as per the user's wish. After all the test runs are completed the user has to go through all the outputs, which are the statistics and parameters generated as an output after the test runs. This is the evaluation process that is carried out in order to determine the classifier with the best performance for the dataset. This classifier can then be used to build model for the database.

A. Our Approach

The evaluation process after WEKA is finished with its execution is time consuming and tedious job to be carried out manually. In this paper we propose a system that would work to reduce human efforts on the evaluation part. This system will carry out the process of evaluation on its own after the user is finished with the test runs on WEKA. The outputs from test will be evaluated that is all the parameters and statistics will be compared and calculated to determine the best classifier for the user's dataset.

Also a database will be maintained in the system. This database will keep record of the user's dataset and the test results for them. This will make sure that user's don't have to provide the input dataset again to perform the tests and even will be able to visualize previous results.
Following figure 6 will provide you with a view of the proposed system.

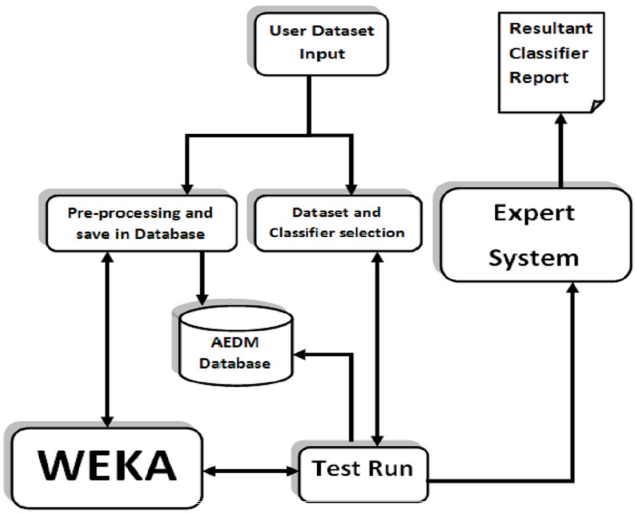


Fig.5. System Architecture for proposed system

V. UML REPRESENTATION

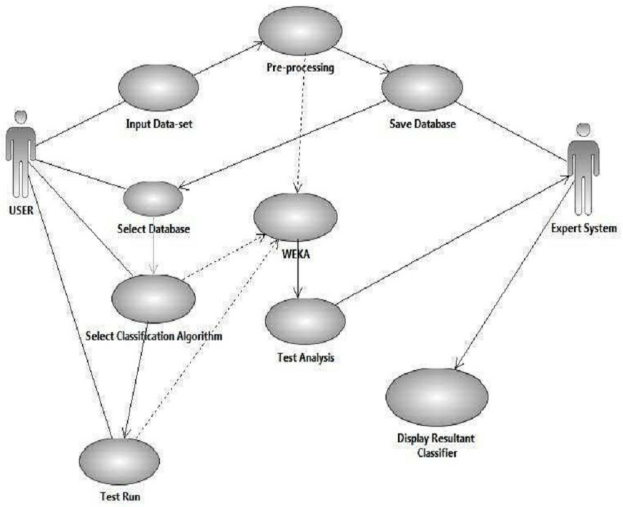


Fig.6. Use case diagram

VI. DATABASE DETAILS

This project has two database tables Dataset_Info and Classifier_Result. Dataset_Info table has 6 fields that store details about dataset. Classifier_Result table has 9 fields that store analysis result. Database is not a part of WEKA originally. This is an additional feature in this project where user could save the test dataset and its output into the system itself for future use. The two datasets are as follows.

TABLE 1
Database_Info

Column Name	Constraint	Datatype
Id	Primary key	Number
Dataset Name	Not null	Varchar
No_Attributes	Not null	Number
No_Insances	Not null	Number
Missing_Status	Not null	Number

TABLE 2
Classifier_Result

Column Name	Constraint	Datatype
Id	Foreign key	Number
Classifier	Not null	Varchar
Para1	Not null	Varchar
Para2	Not null	Varchar
:	:	:
ParaN	Not null	Varchar
Accuracy	Not null	Number
Time_to_Build	Not null	Time

REFERENCES

[1] Data Mining: A Knowledge Discovery Approach, K. Cios, W. Pedrycz, R. Swiniarski, L. Kurgan, Springer, ISBN: **978-0-387-33333-5**, **2007**.
[2] Data mining: concepts, models, methods, and algorithms, mehmed kantardzic, ISBN: **0471228524**, Wiley-IEEE Press, **2002**.
[3] Ian Witten and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd Edition, Morgan Kaufmann, ISBN **0120884070**, **2005**.
[4] WEKA manual.
[5] Zdravko Markov, Ingrid Russell, An Introduction to the Weka DataMining System.