Performance Evaluation of Open Source Data Mining Tools

Syeda Saba Siddiqua[1] Mohd Sameer[2] Ashfaq Ahmed Khan[3]
[1,2,3]Computer Engineering

Abstract— This is an attempt at evaluation of Open Source Data mining tools. Initially the paper deliberates on what can be and what cannot be the focus of inquiry, for the evaluation. Then it outlines the framework under which the evaluation is to be done. Next it defines the performance criteria to be measured. The tool selection strategy for the study is framed using various online resources and tools selected based on it. A table lists the different set of criteria and the findings of each tool against it. After capturing the findings of the study in a tabular fashion, a framework implementation strategy is made. This details the relative scaling for the evaluation. Based on the scorings, a conclusion remark with some suggestions summarizes the findings of the study. Lastly some assumptions/Limitations are discussed.

Key words: Evaluation Framework, Rapid Miner, Knime

I. INTRODUCTION

There has been an explosion of Data Mining tools in the market. On the Technical forums, there are all the claims, arguments and counters that create a lot of noise about the "best tool" in the market. Therefore, there is need that there has to be some clarity about this. Hence, this study tries to clear the air from academic perspective.

There are different ways in which different tools read, handle and tweak the data before displaying the results for the requested task. To try to measure execution times for benchmarking the tool on a standard Data Set was not the correct approach to analyze performance. The tool is not to be seen as a data cruncher. It is rather a platform that implements the mining process. Hence, the job execution times would reflect just not implementation of algorithm but also account for input/output times, intermediate results calculation times and garbage collection and cleansing.

The focus of our inquiry should be the implementation of process of "data mining" by tool rather than abstract trial to measure execution time for jobs running in these environment. And hence, new Frame work is proposed to select the best tool for Academic Use. The tools were analyzed from a beginner's perspective. To come up with benchmarking parameters, the literature study of many journal articles was done. These all resources are listed in the citations at the end of this report.

II. AN EVALUATION FRAMEWORK

At the core of this tool evaluation methodology is a scoring framework. The literature review was done to understand the context in which the tools are evaluated. After a detailed review of these frameworks, the mixed approach was preceded that encompasses both frameworks and carefully draws into considerations only the striking elements that broadly define "open source". On each of the features from the above frame work, a decision was made either to include it or to narrow its scope before inclusion or to drop it all together.

III. TOOL SELECTION STRATEGY & FAMILIARITY

Some initial online review of the most popular Open Source Tools available was done. The decision was based on easy installation and access to documentation and popularity of the tools. After installation of the elected tools, A quick scan through video tutorials was done to become familiar with the tools. Basic hands-on exercise with each one of the tools was also done. So as to get the feel of the User Interfaces, This step greatly helped in the assessments.

IV. DIFFERENT CRITERIA: ASSESSMENT OF TOOLS

The below table consists of different set of criteria and findings on each tool. These include the four groups represented by the framework (performance, functionality, usability, and ancillary task support) plus an additional group of Open Source System specific criteria.

A. Performance

This category focuses on the qualitative aspects of a tool's ability to easily handle data under a variety of circumstances rather than on performance variables that are driven by hardware configurations and/or inherent algorithmic characteristics.

B. Functionality

Software functionality helps assess how well the tool will adapt to different data mining problem domains.

C. Usability

One problem with easy-to-use mining tools is their potential misuse. Not only should a tool be easily learned, it should help guide the user toward proper data mining rather than "data dredging". KDD is a highly iterative process. Practitioners typically adjust modeling variables to generate more valid models. A good tool will provide meaningful diagnostics to help debug problems and improve the output.

D. Ancillary Task Support

These tasks include data selection, cleansing, enrichment, value substitution, data filtering, binning of continuous data, generating derived variables, randomizing, deleting records, etc. Since it is rare that a data set is truly clean and ready for mining, the practitioner must be able to easily fine-tune the data for the model building phase of the KDD process.

Criteria	Weka	Rapid Miner	Knime

Performance Criteria			
Platform Variety	Weka was initially written in C, later converted to Java, which makes it run on almost every Platform.	Rapid Miner is completely written in Java, which makes it run on almost every Platform.	Knime is offered for Linux and windows platforms. Experimental KNIME version for Mac OSX with Java 1.6
Software Architecture	Weka's modular, extensible architecture allows sophisticated data mining processes to be built up from the wide collection of algorithms and tools. Extending the toolkit is easy with simple API, plugin mechanisms and facilities in automating the integration of new learning algorithms. Graphical user interfaces.	Rapid Miner can be used as stand-alone program on the desktop with its graphical user interface, on a server via its command line version, or as data mining engine for your own products and Java library for developers.	KNIME is a bundle containing Eclipse Software licensed under the Eclipse Public License and separate KNIME plug-ins licensed under the General Public License.
Robustness	Results are logged only after completion of job. For Crash during job runs alternative suggestion is to use databases as intermediate/backup storage.	Users report of Crashes due to large data structures, too long file names and temporary directories getting filled. However, there are workarounds available to resolve these crashes.	User reports of crashes due to Flow layouts, work space settings after upgrades. However, these are usually fixed by the module owners/ developers community.
Performance aspects of Data			
Heterogeneous Data Access	Data can be imported from a file in various formats: ARFF, CSV, C4.5, binary. Data can also be read from a URL or from an SQL database (using JDBC)	Support of several file formats including arff, C4.5, csv, bibtex, dBase, and reading directly from databases.	KNIME can only read data from three sources: ARFF files (which are Weka files), text-delimited files (which include csv files), and from a database
Data Size	It can handle millions of data sets	Can handle several 100 million data sets	KNIME allows analysis of 300 million customer addresses, 20 million cell images and 10 million molecular structures
Performance aspects of Integration			
Interoperability	Weka frame work acts like a Plug-in where utilities can be integrated using APIs	It's fully integrated with weka through APIs	It's fully integrated with weka, the statistical toolkit R, and JFree Chart through APIs

TABLE 1

Functionality

Algorithmic Variety	Large number of algorithms covering Classification, regression. Predictive modeling, association rules and clustering	Large Number of Built-in Data Mining Operators for Preprocessing, selecting algorithms, optimization schemes	It incorporates over 100 processing nodes for data I/O, preprocessing and cleansing, modeling, analysis and data mining as well as various interactive views, such as scatter plots, parallel coordinates and others
Algorithm Modifiability	In explorer, it is easy to optimize (the Workflow) by tweaking parameters of algorithms	Rapid Miner provides flexible operators for data input and data output	This is provided by a number of parameters within each operator
Model Validation	Explorer interface provides many model evaluation means like ROC curves, Cross-validation schemes	Model validation is available through GUI Operators like Cross-validation	Many model evaluation schemes available
Functional aspects of Data			
Data Type Flexibility	Explorer has rich features that can select Attributes based on results from search criteria's and attribute evaluations	A combination of Attribute selection operator and attribute type will allow for data type flexibility during reads by a particular operator	Data Read Operators are Less flexible and will not tolerate missing values
Data Sampling	Weka-filters-unsupervised instance packages allow for random sampling of data sets	Richness in sampling provided by Absolute-Sampling, Absolute-Stratified-Sampling	It provides binning operators from Data manipulation class
Functional aspects of Reporting			
Reporting	Data can be inspected visually by plotting attribute values against the class, or against other attribute values	The richness of reporting is captured in a Results Perspective. Also, the results can be stored physically on repositories	Offers data visualization components to present results
Model Exporting	Model objects can be serialized and stored as binary files, which can be exported	Model Writer, Model Loader and Model Applier. Deal with model files(.mod) to achieve import and export of models	Models, processes can be wrapped into a node and then can be exported to other users

TABLE 2

Usability			
User Interface	Rich User Interfaces tailor made for user friendly experience	GUI is like any traditional development platforms	GUI is like any traditional development platforms
Learning Curve	It took 25 person hours for a quick study and case study exercise on Decision Trees, Clustering and Association Rules for a new learner	It took 25 person hours for a quick study and case study exercise on Decision Trees, Clustering and Association Rules for a new learner	It took 25 person hours for a quick study and case study exercise on Decision Trees, Clustering and Association Rules for a new learner
User Types	Well Categorized user types exists such as explorer, experimenter, knowledge flow, Command Line Interface	No tailor made modes available. Though, broadly categorized into Command line interface and GUI	Standard and Command line.

Action History	With Log button, As we perform actions in WEKA, the log keeps a record of what has happened	Though Associated with a Process chain in GUIs. It is not available for sessions	Provides a listing of Most recently used operator nodes
Usability aspects of Data			
Data Visualization	Visualization aides for data analysis exist for generic plots to specialized entities such as dendo-grams	Visualization operators take care of simple charts, tabular views to complex visualizations such as SOMs	Views can range from simple table views to more complex views on the underlying data or the generated model.
Error Reporting	With WEKA 3.6 error logging had improved, to include all error information while executing in GUI	Messages are precisely available for attribute level or row level or process level issues	Console takes care of warnings and error messages. This can be logged in by configuring Level settings at /KNIME/KNIME GUI

TABLE 3

Ancillary Tasks Support			
Task Support for Data Handling			
Binning	Yes it allows binning and the decision to use binning lies with the user	Yes it allows binning and the decision to use binning lies with the user	Yes it allows binning and the decision to use binning lies with the user
Deriving Attributes	Integration with the pentagon bi suite, allows for etl operations	Together with the hundreds of operators for data pre-processing, Rapid-Miner can therefore also be used, apart from data analysis, for data integration and transformation with outstanding results	ETL operations on data achieved using Data transformation Components
Randomization	WEKA offers two possibilities for randomizing a dataset: randomize and Randomize filter	Offered through Data Sampling Components	Offered through Data Manipulation Components
Record Deletion	Possible through Data Read Operators	Possible through Data Read Operators	The addition of data sets and the deletion of unused data sets is Possible through Report Operator
Handling Blanks	Filter Operators can do global substitutions for missing values	But substitution is handled separately by data processing operators. There are no direct substitution symbols for replacement of undesired data. This tool is not synched up with latest formats for office excel sheets	No options to tolerate blanks from input data
Task Support for Data Manipulation			
Metadata Manipulation	Though Metadata can be referenced by the tool, it cannot be manipulated.	Though, Rapid Miner has a flexible interactive design which lets user to additional meta data on the available data sets to enable automated search and optimized preprocessing with are both needed for an effective data mining processes.	Though Metadata can be referenced by the tool, it cannot be manipulated.

Data Cleansing	WEKA provides a filter that substitutes the mean (for numeric attributes) or the mode (for nominal attributes) for each missing value	Has a host of preprocessing operators to choose from. Attribute selection is done through attribute selection operator. No Missing value substitutions. Such rows are omitted altogether	Very primitive. Not room for flexibility in component behavior
Result Feedback	The Explorer save the built classifier in the model file, but also the header information of the dataset the classifier was built with.	The results can be exported and reused	The results can be exported and reused

TABLE 4

Open Source Features			
Scenario: Use OSS for becoming independent of proprietary solutions and providers. (The OSS helps to reduce dependence on software product companies with big market power and low customer orientation.) AND Use OSS for research purposes (The OSS is used to support research activities.)			
Feature 1.1 Functional: Required functionality covered, clear direction of product evolution recognizable	Yes	Yes	Yes
Feature 1.2 Technical: Reliability	Stabilized development over a period of 2 decades	Stabilized development	New features are into beta stages
Feature1.3 Organizational: Sufficient support available	Professional support available & its backed by university of Weka to	Professional support available	Professional support available
Feature1.4 Economical: Flexible maintenance according to individual needs	Yes	Yes	Yes
Feature 1.5 Political: Possibility for influencing further development with respect to individual needs	Large User Community	User Community	Small User community

TABLE 5

V. Methodical Application of Framework for Scoring

The criteria within each category are assigned weights so that the total weight within each category equals 1.00 or 100%. This weighting must be conducted with respect to the intended use of the software. Consider an organization whose data warehouse is centrally located on a Windows NT server, and whose local area network consists exclusively of Windows NT workstations. Such an organization will probably assign a low weight to platform support since any other platforms on which the tool is supported do not matter.

Relative Performance	**Rating**
Much worse than the reference tool	1
Worse than the reference tool	2
Same as the reference tool	3
Better than the reference tool	4
Excellent than the reference tool	5

VI. Scoring Guidelines

Once the criteria have been weighted with respect to a set of targeted needs, the tools can now be scored for comparison. Scoring is done relative to a "reference tool". Generally the evaluator is predisposed toward a favorite tool based on his experience. This "favorite" should be selected as the

reference tool. Any tool may be selected in the absence of a favorite. The reference tool receives a rating of 3 for each criterion. Other tools are then rated against the reference tool for each criterion.

Using this scheme a score is calculated for every criterion for each tool. These scores are then totaled to produce a score for each category. Finally, the categorical scores are combined in a weighted-average to calculate an overall tool score. By default each criteria category receives a weight of .20. However, some adjustment was done to these weights to allow emphasizing or de -emphasize particular categories of criteria.

Criteria	**Weight**	**Tool A**		**Tool B**		**Tool C**	
Performance (.25)		Rating	Score	Rating	Score	Rating	Score
Platform Variety	0.1	3	0.3	3	0.3	3	0.3
Software Architecture	0.1	3	0.3	3	0.3	4	0.4
Heterogeneous Data Access	0.15	4	0.6	3	0.45	2	0.3
Data Size	0.3	3	0.9	3	0.9	3	0.9
Interoperability	0.15	2	0.3	3	0.45	3	0.45
Robustness	0.2	3	0.6	3	0.6	3	0.6
Performance Score			3.00		3.00		**2.95**
Functionality (.30)		Rating	Score	Rating	Score	Rating	Score
Algorithmic Variety	0.1	4	0.4	3	0.3	2	0.2
Model Validation	0.15	4	0.6	3	0.45	3	0.45
Data Type Flexibility	0.15	5	0.75	3	0.45	2	0.3
Algorithmic Modifiability	0.15	5	0.75	3	0.45	3	0.45
Data Sampling	0.15	3	0.45	3	0.45	3	0.45
Reporting	0.15	3	0.45	3	0.45	3	0.45
Model Exporting	0.15	3	0.45	3	0.45	3	0.45
Functionality Score			**3.85**		**3.00**		**2.75**
Usability (.20)		Rating	Score	Rating	Score	Rating	Score
User Interface	0.15	5	0.75	3	0.45	3	0.45
Learning Curve	0.2	3	0.6	3	0.6	3	0.6
User Types	0.2	5	1	3	0.6	2	0.4
Data Visualization	0.05	3	0.15	3	0.15	3	0.15
Error Reporting	0.15	4	0.6	3	0.45	2	0.3
Action History	0.05	5	0.25	3	0.15	3	0.15
Domain Variety	0.2	3	0.6	3	0.6	2	0.4
Usability Score			**3.95**		**3.00**		**2.45**

Ancillary Support Task (.05)		Rating	Score	Rating	Score	Rating	Score
Data Cleansing	0.05	4	0.2	3	0.15	2	0.1
Binning	0.25	3	0.75	3	0.75	3	0.75
Deriving Attributes	0.05	3	0.15	3	0.15	4	0.2
Randomization	0.2	3	0.6	3	0.6	3	0.6
Record Deletion	0.05	3	0.15	3	0.15	3	0.15
Handling Blanks	0.1	4	0.4	3	0.3	2	0.2
Meta Data Manipulation	0.1	2	0.2	3	0.3	2	0.2
Result Feedback	0.2	3	0.6	3	0.6	3	0.6
Ancillary Support Task Score			**3.05**		**3.00**		**2.80**
OSS Feature 1 (.20)		Rating	Score	Rating	Score	Rating	Score
Feature 1.1	0.25	3	0.75	3	0.75	2	0.5
Feature 1.2	0.25	3	0.75	3	0.75	2	0.5
Feature 1.3	0.2	4	0.8	3	0.6	3	0.6
Feature 1.4	0.2	4	0.8	3	0.6	2	0.4
Feature 1.5	0.1	4	0.4	3	0.3	1	0.1
OSS Feature 1 Score			**3.50**		**3.00**		**2.10**
Weighted Average			**3.47**		**3.00**		**2.61**

The Scores Table: Tool A – Weka; Tool B – Rapid Miner; Tool C – Knime

VII. CONCLUSION

In the light of the findings of this evaluation a clear picture starts to emerge. The tool "Weka" is the clear winner mainly because of the tool's functionality and usability superiority. The excellent use of User-Centric Interface design and great flexibility in application of algorithms are the bench marks for any good Open Source data mining tool in an academic setting. A steady and dedicated development team backed by a corporate sponsor makes Weka a promising OSS model.

But Weka has to be on its toes as a relatively new comer Rapid-Miner is challenging it. Rapid Miner has well developed and stabilized system software and its user base is catching up fast. It's functionality is as good as Weka's and in fact it can neatly be integrated with Weka to take advantage of both the software. The business model built around Rapid-Miner and its geographical location is attracting finances and sponsorships through its Partners Program. This tool certainly has a promising feature.

Knime is in its primitive stages. Many of the interoperability and exporting modules are in beta stages of development. It has a Long way to go!

VIII. ASSUMPTIONS

The scoring system is based on a reference tool. And my familiarity with the reference tool is something that cannot be measured and is abstracted as a rating of "3". Now, It is assumed that my understanding and familiarity with the reference tool i.e.; Rapid- Miner broadly represents with that of an average Data Mining professional. And hence it should be taken with caution. If a data mining expert is to replace me as an evaluator, that person could rate the Rapid-Miner according to his experience, which could in turn make the total evaluation entirely different.

BIBLIOGRAPHY

[1] Retrieved from WEKA: http://weka.wikispaces.com/Related+Projects

[2] Adriaans, P. a. (1996). *Data Mining*. Addison-Wesley Longman.
[3] Berthold, M. R., Cebron, N., Dill, F., Fatta, G. D., Gabriel, T. R., Georg, F., et al. *Knime: The Konstanz Information Miner.* Konstanz: Atlanta Chair of Bioinformatics and Information Mining.
[4] Bouckaert, R. R., Frank, E., Hall, M. A., Holmes, G., Pfahringer, B., Reutemann, P., et al. (2010). WEKA—Experiences with a Java Open-Source Project. *Journal of Machine Learning Research*, 2533-2541.
[5] Collier, K., Sautter, D., Medidi, M., Morgan, J., Ratliff, M., Marjaniemi, C., et al. (2010). *A methodology for evaluating and selecting DATA MINING software.* Point of Reference, Center for Data Insight; KPMG Peat Marwick LLP, FlagStaff.
[6] Cruz, D., Wieland, T., & Ziegler, A. (2006). Evaluation Criteria for Free/Open Source Software Products Based on Project Analysis. *Software Process Improvement and Practice*, 107-122.
[7] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (VOL 11, ISSUE 1, 2009). The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations*, 01-18.
[8] *http://rapid-i.com/content/view/8/56/lang,en/*. (n.d.). Retrieved from Rapid I - References: http://rapid-i.com/content/view/8/56/lang,en/
[9] KNIME. (n.d.). Retrieved from http://www.knime.org/downloads/license
[10] KNIME. (n.d.). *Features*. Retrieved December 12, 2010, from http://www.knime.org/features
[11] *Knime Forums*. (n.d.). Retrieved december 15, 2010, from Knime: http://tech.knime.org/node/20737
[12] KNIME. (2010). *KNIME Report Designer Quick Start Guide.*
[13] *Machine Learning Group*. (n.d.). Retrieved december 12, 2010, from Machine Learning Group at University of Waikato.: http://www.cs.waikato.ac.nz/~ml/
[14] *MOA Massive Online Analysis Details*. (n.d.). Retrieved December 14, 2010, from MOA Massive Online Analysis: http://moa.cs.waikato.ac.nz/details/classification/
[15] Rapid-I. (2010). *Rapid Miner 5.0 Manual.* Rapid-I.
[16] *Rapid Miner*. (n.d.). Retrieved december 14, 2010, from it. toolbox: http://it.toolbox.com/wiki/index.php/RapidMiner Performance Evaluation of Open Source Data Mining Tools 24 *Rapid Miner*. (2010). Retrieved December 14, 2010, from Enterprise Edition: Comparison: http://rapid-i.com/content/view/181/190/lang,en/#enterprise
[17] *Rapid Miner Forums*. (n.d.). Retrieved December 15, 2010, from Rapid Miner Forums: http://rapid-i.com/rapidforum/index.php/topic,17.0.html
[18] *Salient features of Rapid Miner*. (n.d.). Retrieved December 14, 2010, from http://it.toolbox.com: http://it.toolbox.com/wiki/index.php/RapidMiner#Salient_features_of_RapidMiner
[19] *The Wekalist Archives*. (n.d.). Retrieved December 14, 2010, from The Wekalist Archives: https://list.scms.waikato.ac.nz/pipermail/wekalist/2006-May/007026.html
[20] University of Wakaito. (2010). *Weka Manual 3.7.2.*
[21] Weka. (n.d.). *Weka---Machine Learning Software in Java*. Retrieved December 12, 2010, from Weka---Machine Learning Software in Java: http://sourceforge.net/projects/weka/files/
[22] wiki. (n.d.). Retrieved from http://en.wikipedia.org/wiki/KNIME

