Overview of Machine Learning Tools and Libraries

Daniel Pop, Gabriel Iuhasz
Institute e-Austria Timişoara
Bd. Vasile Pârvan No. 4, 300223 Timişoara, România
E-mail: {danielpop, iuhasz.gabriel}@info.uvt.ro

Abstract

Over the last three decades many general-purpose machine learning frameworks and libraries emerged from both academia and industry. The aim of this overview is to survey the market of ML tools and libraries and to compare them in terms of features and supported algorithms. As there is a large number of solutions available offering a large spectrum of features, we will firstly introduce a set of criteria, grouped in four categories, for both pruning and comparing the candidates. Based on these criteria, we will synthetically present the results in tables and we shortly discuss the findings in each category.

1 Introduction

Given the enormous growth of collected and available data in companies, industry and science, techniques for analyzing such data are becoming ever more important. Research in machine learning (ML) combines classical questions of computer science (efficient algorithms, software systems, databases) with elements from artificial intelligence and statistics up to user oriented issues (visualization, interactive mining, user assistance and smart recommendations). Over the last three decades, many general purpose machine learning frameworks, as well as special purpose machine learning libraries, such as for phishing detection [1] or speech processing [2], as emerged from both academia and industry. In this survey, we will only consider the general purpose frameworks.

The **objectives** of this work are driven by the scope and objectives of an ongoing initiative: design a distributed, open source system for scientific problem solving. In this context, we are particularly interested in aspects such as usability, ability to handle large data sets from various sources, interoperability with other libraries, distributed computing support. A second motivation behind this work is represented by the fact that there are no recent, similar surveys available, the most recent one found by the authors being a data mining survey more than 4 years old [5].

Some words on **methodology** are necessary since ML domain, rebranded some decade ago in Data Mining, Knowledge Discovery or alike, produced a lot of projects, libraries, tools and frameworks. We are not aiming to review all available frameworks for ML ever created, rather to reach the most used and active ones. For example, Machine Learning Open Source Software repository (mloss.org) lists over 400 entries at the date of this paper (summer 2011). Some of these entries are Weka / R packages and addons, or refer to specific problem domains (biology, mathematics etc). Even ignoring these entries, we are left with hundreds of packages. And these are only the open source ones. Therefore, we need a systematic approach to make a narrower selection.

In the first phase, we needed to identify what specifc domain repositories, public dissemination channels, previous surveys and similar works are available to start with. For example, searching for surveys about machine learning on popular Web search engines (Google, Bing, Yahoo! Search) returned no valuable results. Refining the search to 'data mining survey', few useful results were returned [3, 4, 5], most recent one being over 4 years old (details about these related papers are given in section 2). Other sources of candidates for initial list are the results from polls and surveys conducted by popular, independent online bodies, such as Rexer Analytics [6] or KD Nuggets [8]. At the end of phase one, the initial list was including more than 80 candidates – standalone tools, plugings and libraries – originating from different domains and providers, such as relational database systems providers (Oracle Data Miner, Microsoft SQL Server, IBM Intelligent Miner, IBM SPSS Modeler), mathematics and statistics software (MATLAB, Mathematica, MathSoft S-Plus, Statistica, R), data min-

ing software providers (RuleQuest C5.0/See5/Cubist, Salford Systems CART/SPM) or academia (KNIME, Weka, Orange).

In the second phase, we pruned the initial list of candidates by removing outdated candidates. Out of 63 distinct products covered by the 3 previous surveys [3, 4, 5], 44 (70%) were flaged as outdated[1], out of which 27 (61%) were coming from industry and the rest from research organisations. The final list was completed by adding software employing best class neural network implementations because of large applicability of these methods in modern systems. In the end, we selected a final list of 30 libraries and tools for review, out of over 100 candidates considered.

The paper is organised as follows. Next section shortly reviews latest available similar surveys and presents the findings of the most recent online polls conducted by renown independent organisations. Section 3 details the criteria used in this survey for tools and libraries evaluation and comparison, as well as the rationale behind their selection. Section 4 discusses the main findings of this survey, while the last one presents our conclusions and future work.

2 Previous work

In the paper [3], from DataMining Lab and presented at 4th International Conference on KDD (KDD98), the authors present a comparison of 17 leading DM tools at that time. Most of those tools (13, i.e. 77%) disappeared from the market (e.g. Unica Technologies, DataMindCorp), were acquired by other companies and then abandoned (e.g. Thniking Machine and Integral Solutions acquired by IBM) or simply users lost interest in them and no recent versions were issued (e.g. WizWhy and WizRule from WizSoft). One year after, in 1999, Goebel et al. [4] pull together a very interesting survey of DM tools presenting 43 products, but we can observe exactly the same outdaing ratio as in previous study (77%), leaving only 10 survivors. In their paper, the authors even identified similar online survey projects, which unfortunatelly are not maintained anylonger, except for KD Nuggets [8]. A more recent survey [5] (2007) focuses only on open-source software systems for data mining. Being a more recent study, the outdating ratio is better – only (50%) – and thus 6 out of 12 projects are still alive.

Starting with 2007, Rexer Analytics [6] is conducting yearly, on-line surveys on Data Mining tools usage.

[1]In this context, we condisered a product as *outdated* if no new versions of the product were released after 2010, or we couldn't find it on the web at all.

Table 1. ML tools usage

	KD Nuggets [9]		Rexer Analytics [7]
1	R	31%	R
2	Excel	30%	SAS
3	RapidMiner	27%	IBM SPSS Statistics
4	KNIME	22%	Weka
5	Weka	15%	StatSoft Statistica
6	StatSoft Statistica	14%	RapidMiner
7	SAS	13%	MATLAB
8	Rapid Analytics	10%	IBM SPSS Modeler
9	MATLAB	10%	MS SQL Server
10	IBM SPSS Statistics	8%	SAS Enterprise Miner
11	IBM SPSS Modeler	7%	KNIME
12	SAS Enterprise Miner	6%	C4.5/C5/See5
13	Orange	5%	Mathematica
14	MS SQL Server	5%	Minitab
15	Other free DM software	5%	Salford Systems

The latest one available [7] (2011) shows that R system continues to dominate the market (47%), while StatSoft Statistica, which has been climbing in the rankings, is selected as the primary data mining tool by the most data miners. Data miners report using an average of 4 software tools overall. Statistica, Knime, RapidMiner and Salford Systems received the strongest satisfaction ratings in 2011. Another important online survey source is the latest KD Nuggets report [9] (2011). Table 1 shows the results on ML tools usage from both Rexer Analytics and KD Nuggets.[2]

3 Properties considered in this survey

Back in 1998, Kurt Threaling [10] identified several challenges ahead of data mining software tools: database integration, automated model scoring, exporting models to other applications, business templates, effort knob, incorporate financial information, computed target columns, time-series data, use vs. view and wizards. Inline with the objectives and moti-

[2]Rexer Analytics and KD Nuggets surveys are open, on-line surveys so that big players may use their channels to include more votes or positive feedback for one tool or another. Although they don't mirror with 100% accuracy the market, it is very unlikely that important players were missed by these reports.

vation of our survey, we will consider two of these challenges and we will evaluate how are they implemented in the reviewed products.

The *effort knob* refers in general to the feedback the system is giving to the end-user upon changing or tuning various parameters of the algorithm in order to obtain a more accurate prediction model. This kind of tweaking may increase the processing time by order of magnitude. The relationship between parameters and processing time is something a user should not care about, instead the system shall provide constant feedback regarding effort estimates so that users can easily see how costly (in terms of resources such as memory and processor time) the operation is. Alternatively, users shall be able to control the global behavior and resource consumption and the system shall adjust the parameters accordingly. For example, setting the effort level to a low value, the system should produce a model quickly, doing the best it can given the limited amount of time. On the other hand, if the end-user sets the level to a higher value, the system might run overnight to produce the best model possible. In our study, this feature is named Effort Feedback.

The second challenge to which we devoted some attention is *Wizards*, that can also significantly improve the end-user's experience. Besides simplifying the process, they can prevent human error by keeping the user on track. Therefore, we will look for the availability of any wizard-like helpers in evaluated products.

An interesting finding of Rexer Analytics Yearly survey on Data Mining edition 2011 is the fact that "Data Mining most often occurs on a desktop or laptop computer, and frequently the data is stored locally." There is a tremendous data explosion today in all domains, so that sooner or later the big data sets will be the commonality, not the marginal cases. Thus, the ability of a product to handle big data sets is considered in our study as well.

Another important issue in comparing machine learning products is the type of methods, algorithms, models they support. A recent paper [11] shows which are the 10 most used ML algorithms. These are presented in Table 2, along with a similar list inferred from Rexer Analytics survey [7]. Based on these models and algorithms, we will use in our evaluation the following classes of machine learning problems: classification, clustering, association analysis, factor analysis, regression analysis, time series and pre-processing capabilities. Under each class, different tools implement different algorithms, or variants of the same algorithm. More details are given in Section 4.

Another property considered in this survey relates to programming language supported by a framework. According to [9], in terms of languages for ML, the most popular is R (30.7%) followed by SQL (23.3%) and Java (17.3%). Phyton, C/C++, Perl, Awk/Gawk/Shell and F# completes the list.

Table 2. ML methods

	Zheng (2010) [11]	Rexer Analytics (2011) [7]
1	C4.6 / Classification	Regression
2	K-means / Clustering	Decision trees
3	SVM / Statistical learning	Cluster analysis
4	Apriori / Association analysis	Time series
5	EM / Statistical learning	Neural networks
6	Page rank / Link mining	Factor analysis
7	Adaboost / Ensemble learning	Text mining
8	KNN / Classification	Association rules
9	Naive Bayes / Classification	SVM
10	CART / Classification	Bayesian

To conclude this section, here is the list of properties considered considered in this study for the evaluation of ML products:

- General: licensing model (commercial, open-source), operating system, ability to handle big data sets, language support
- Classes of machine learning problems supported: classification, clustering, association analysis, factor analysis, regression analysis, time series, pre-processing
- End-user support: effort feedback support, wizards, visual programming (i.e. the ability to create data flows = sequences of operations applied to data), model visualization
- Others: support for parallel processing (multi-core / GPU), ability to access various data sources (although some libraries rely on the capabilities of the environment to provide the data, some others have their own load functions), ability to export the model to various formats, and activity on the project.

4 Discussions

Table 3 presents all the tools and libraries we considered in this survey. Before discussing in detail about all the properties identified in the previous section, we would like to share some general remarks inferred from analysed packages:

- some frameworks does not have native ML support, however they are extended using popular addons, such as Mathematica Learning Framework, Matlab etc.
- some frameworks are integrated with other ML frameworks; for example RapidMiner with R and Weka
- some frameworks use 3rd party libraries to support specific ML methods, such as LibSVM[3] for Support Vector Machine, or LibLinear[4] for linear classification, as well as popular libraries for mathematics support, like NumPy[5], SciPy[6] or LAPACK[7].

Note: In this paper, by R we refer to R system plus machine learning packages, such as caret.

4.1 Generalities

Considering the *licensing model*, 10 out 30 products considered in this survey (those *italicized* in Table 3) are commercial, close-source products, while the others are licensed under various open-source licenses (GNU (L)GPL, Apache or MIT) with a strong preference towards GPL and LGPL.

In terms of *operating system* (column OS), as most of them rely on virtual machines (Java, Python), they are running cross-platform (Windows, Unixes, Mac OS X). The few exceptions are large commerical applications developed for Windows operating system.

The ability to *handle large data sets* (column HLD) is largely impacted by two factors: the programming language and environment used to develop the tool and the supported machine learning methods. One can observe that most of the products originating in Python world, such as Mlpy, Pyml and YAPLF, have problems in handling large data sets, maybe due to the lack of mature Python libraries for large data processing at the time tool development was started. Machine learning methods also impact this criteria, some of them, such as neural networks, being not well suited candidates for large data sets handling.

Programming language support and interfacing (column Language) is an important criteria when comes to integrate a library in your own application. As we see, all of the surveyed products are supporting at least one external interface, which is usually its native language / platform. Many of them offer support for additional programming lanuages as well. The most popular languages are Java (11), C/C++ (10) and Python (9), followed by .NET, Fortran, R etc.

4.2 Supported classes of ML problems

Before discussing about methods and algorithms supported by each of the investigated products, we need to say few words about their categorization. For example, we defined a category called Generalized Linear Methods for Classification (GLMC) that includes different algorithms and variants based on linear models. We introduced these categories (methods) to have a more fair comparison between different tools by reducing the impact of those implementing many variants or algorithms for the same method. On the other hand, this taxonomy will eventually help us to build an intelligent recommendation system for machine learning problems solving. The following methods were introduced:

- GLMC = Generalized Linear Models for Classification: LR=Logistic Regression, (D)LDAC = (Diagonal) Linear Discriminant Analysis Classifier, Basic perceptron, Elastic Net Classifier, Golub classifier, Stochastic Gradient descent;
- GLMR = Generalized Linear Models for Regression: (P)LS=(Partial) Least Squares, RR=Ridge Regression, LARS=Least Angle Regression, Elastic Net, Stochastic Gradient descent;
- Non-linear models for classification: PC=Parzen-based, FDC=Kernel Fisher Discriminant Classifier, k-NN=k-Nearest-Neighbor, CART=Classification and Regression Trees, Randomized trees + Gradient tree boosting, MLC=Maximum Likelihood Classifier
- EM=Expectation maximization
- NB=Naïve Bayes
- NN=Multi-layer Neural Networks
- LDA=(Diagonal) Linear Discriminant Analysis
- PCA=Principal Component Analysis

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[4] http://www.csie.ntu.edu.tw/~cjlin/liblinear/
[5] http://www.numpy.org/
[6] http://www.scipy.org/
[7] http://www.netlib.org/lapack/

Table 3. ML tools and libraries

Name	HLD	OS	Language
Aleph	No	Win/Unix	Yap Prolog
C4.5/C5/See5	Yes	Win/Unix	C/C++
Encog	Yes	Win/Unix	Java/.NET
FuzzyML	Yes	Win/Unix	ADA
IBM Cognos	Yes	Web?	PowerHouse?
IBM SPSS Modeler	Yes	Win/Unix/OSX	Java
JavaML	Yes	Win/Unix	Java
JHepWork	Yes	Win/Unix	Java/Jython/Jruby/BeanShell
Joone	No	Win/Unix	Java
KNIME	Yes	Win/Unix	Java/Python/Perl
LIONsolver	Yes	Win/Unix	C/C++
Mathematica Learning Framework	Yes	Win/Unix	C++
MATLAB	Yes	Win/Unix	C/C++/Java/Fortran/Python
MLC++	No	Win/Unix	C++
Mlpy	No	Win/Unix	Python
MS SQL Server	Yes	Win	.NET
Neuroph	No	Win/Unix	Java
Oracle Data Miner	Yes	Win/Unix	Java
Orange	No	Win/Unix	C++/Python
PCP	Yes	Win/Unix	C/C++/Fortran
Pyml	No	Win/Unix	Python
R	Yes	Win/Unix	C/Fortran/R
RapidMiner	Yes	Win/Unix/OSX	Java/Groovy
Salford Systems	Yes	Win	C/C++/.NET?
SAS Enterprise Miner	Yes	Win/Unix	C
scikit-learn	Yes	Win/Unix/OSX	C/C++/Python/Cython
Shogun	Yes	Win/Unix	C/C++/Python/R/Matlab
Statistica	Yes	Win	.NET/R
Weka	Yes	Win/Unix/OSX	Java
YAPLF	No	Win/Unix	Python

- SRDA=Spectral Regression Discriminant Analysis
- FDA=Fischer Discriminant
- k-M=k-Means
- HAC=Hierarchical Aglomerative Clustering
- SVM=Support Vector Machine
- DTW=Dynamic Time Warping
- GMM=Gaussian Mixture models
- Wavelet=Wavelet transform
- FRS=Feature ranking / selection (including RFE)
- SC=Spectral Clustering
- MS=Mean shift
- AP=Affinity Propagation
- Manifold: Isomap, LLE, LTSA, MDS
- NMF=Non-negative matrix factorization
- DL=Dictionary Learning
- LP=Label preprocessing
- FE=Feature extraction: Text feature extraction, Image feature extraction
- KA=Kernel approximation
- FZC=fuzzy classification
- O-Cluster=Orthogonal Partitioning Clustering

- MCC=Markov Chain Clustering
- BN=Bayesian Network
- NORM=Normalization, Binarization, Standardizatoin
- RAN=randomization
- SEG=segregation
- BAL=balance
- DISC=discretization
- MI=Missing values
- IM=Instance Manipulation
- RESAM=Re-sampling
- CFSCT=Convert, Filter, Split, Combine, Transform

Tables 4 – 6 present ML problems supported by each product. Enclosed in paranthesis, after the product name, is the number of methods offered by each specific product. We observe that the spectrum ranges from specialised tools based on one method (e.g. Aleph, C4.5/C5/See5, Joone) to versatile software packages supporting tens of methods (e.g. scikit-learn, Weka, R, RapidMiner). Among the methods, CART, k-NN and various architectures of multi-layer neural networks seem to be the most popular ones.

Note: in table 4, by (P) we denoted a proprietary method or algorithm, i.e. details were not unveiled for it by the provider.

4.3 End-user support

Out of the four criteria used in the evaluation of end-user support and guidance, a form of *model visualization* exists in most of the evaluated products. The least interest was deveoted to *effort feedback support*, only four products (IBM Cognos, IBM SPSS Modeler, MLC++ and Orange) offering this feature. Surprisingly, consolidated and popular tools such as RapidMiner, Knime, SAS Enterprise Miner or Weka does not support end-users in making their decisions by quality (of the model) vs. speed (of building the model) trade-off tweaking. But, we can observe in popular tools (e.g. IBM SPSS Modeler, Oracle Data Miner or Weka) a preference towards visual programming that enables end-users to create workflows using sophisticated graphical user interfaces (GUI). Each workflow is composed of a sequence of operations (e.g. preprocessing, building a model, cross validation, visualize the model, export etc.) that are applied on initial data set(s). Some other tools prefer to offer wizards to create complex data analysis processes (e.g. Encog, JHepWork, Statistica). Only two products – Orange (free) and IBM SPSS Modeler (commercial) – include all the four features considered in this survey. Consult Table 7 for exhaustive matrix of supported features by product.

4.4 Miscellaneous

The ability to read data from various *input formats* is an important criteria one shall consider in selecting the tool / library to work with. We noticed that CSV and TXT are the most popular formats, more than 50% of products supporting them, with Weka's ARFF and relational database connectivity on second and third places (around 20%), respectively. Library-specific formats, such as libSVM or Encog, are in use by products relying on these specific packages.

Things are more complicated in respect to the *Export format* used for constructed models. Interoperability between tools is achieved only with the help of Predictive Model Markup Language (PMML) [8], 20% of surveyed products supporting this XML format. Most of the tools use proprietary formats (either XML or binary/serialization). Some of the tools does not export in any way the constructed models (e.g. JavaML, LIONsolver, etc.).

Analyzing big data sets is a time consuming activity, that can be parallelized for most of the ML algorithms. All processors on the market nowdays are based on multi-core architectures, so that multi-core/GPU parallelisation is the most frequent technique used by aprox. 60% of surveyed products to speed-up computation. There also products, such as Knime, RapidMiner, Salford Systems, or SAS Enterprise Miner, that can be setup to execute data analysis tasks on distributed environments (such as computing grids), but specialized technical assistance from their providers is required most cases. None of them is a native, distributed solution.

More details on input formats, export formats and parallel computing are given in Table 8 .

5 Conclusions and future work

Our main findings are summarised below:

(1) Shortly after we started this survey, we have been overwhelmed by the large number of libraries, tools, projects addressing machine learning, showing huge interest in this topic among research teams in academia

[8] PMML is an XML-based markup language developed by the Data Mining Group, http://www.dmg.org/

and industry, equally. We had to churn the candidates carefully and the activity in the project in last period was a well performing criteria.

(2) Looking at selected products mainly from the perspective of end-user experience and support, looking for intelligent agents to guide users during the process, we found that the offer is quite limited and there is room for new (and smart) players.

(3) Applying popular machine learning algorithms to large amounts of data raised new challenges for ML practitioners. Traditional ML libraries does not support well processing of huge data sets, so that new approaches are needed based on parallelization of time-consuming tasks using modern parallel computing frameworks, such as MPI, MapReduce, or CUDA. A sequel survey will investigate machine learning solutions designed for distributed computing environments, such as grids or cloud computing.

Our future plans aim at building a smart platform for problem solving applied in the field of Machine Learning, which will be able to smartly support end-users in their activities by, for example, selecting the most appropiate method for a given data set, or tweaking algorithms' parameters.

Acknowledgments

This work was supported by EC-FP7 project HOST, FP7-REGPOT-2011-1 284595 and by the strategic grant POSDRU/CPP107/ DMI1.5/S/78421, Project ID 78421 (2010), co-financed by the European Social Fund - Investing in People, within the Sectoral Operational Programme Human Resources Development 2007 - 2013.

References

[1] http://www.ml.cmu.edu/research/dap-papers/dap-guang-xiang.pdf

[2] http://www.cs.nyu.edu/~mohri/pub/hbkb.pdf

[3] John F. Elder IV and Dean W. Abbott, A Comparison of Leading Data Mining Tools, KDD 98

[4] Michael Goebel, Le Gruenwald , in SIGKDD Explorations. ACM SIGKDD, June 1999, Vol I, Issue I, page 20 -33, "A SURVEY OF DATA MINING AND KNOWLEDGE DISCOVERY SOFTWARE TOOLS"

[5] X. Chen, G. Williams, and X. Xu , "A Survey of Open Source Data Mining Systems", 2007

[6] Rexer Analytics, http://www.rexeranalytics.com

[7] Rexer Analytics Survey 2011, http://www.rexeranalytics.com/Data-Miner-Survey-Results-2011.html

[8] KD Nuggets, http://www.kdnuggets.com

[9] KD Nuggets Survey 2012, http://www.kdnuggets.com/software/suites.html

[10] Threaling K, Some Thoughts on the Current State of Data Mining Software Applications http://www.thearling.com/text/dsstar/top10.htm, 1998

[11] Zheng Zhu, Data Mining Survey, ver 1.1009, 2010, http://www.dcs.bbk.ac.uk/~zheng/doc/datamining.pdf

Table 4. ML frameworks x methods

Name	Classification	Clustering	Association analysis	Factor analysis	Regression analysis	Time-series	Pre-processing
Aleph (1)	CART	CART	NONE	NONE	NONE	NONE	NONE
C4.5 / C5 / See5 (1)	CART	CART	NONE	NONE	NONE	NONE	NONE
Encog (8)	NN, BN, SVM	NN	NONE	NONE	NN	NN	NORM, RAN, SEG, BAL, MI
FuzzyML (1)	FZC						
IBM Cognos (0)							
IBM SPSS Modeler (3)	NN, k-NN, SVM	NN			NN	NN	
JavaML (10)	CART, k-NN, SVM	k-M, SOM, MCC	NONE	NONE	NONE	NONE	NORM, DISC, MI, IM
JHepWork (10)	NN, BN	K-M, Fuzzy c-M	NONE	NONE	NN, GLMR,	NONE	NORM, RAN, SEG, BAL, MI
Joone (1)	NN	NN	NONE	NONE	NN	NONE	NORM
KNIME (13)	k-NN, CART, NN, NB, SVM	k-M, HAC, Fuzzy c-M	NONE	PCA	SVM, GLMR, NN	NN	NORM, BINNING, CFSCT
LIONsolver (8)	NN, k-NN, SVM	k-M	NONE	NONE	SVM, GLMR, NN	NONE	NORM, DISC, SEG
Mathematica Learning Framework (12)	CART	k-M, SOM	(P)	(P)	(P)	(P)	NORM, RAN, SEG, BAL, MI
MATLAB (16)	NN, k-NN, SVM, NB, CART,	k-M, HAC, MCC	(P)	(P)	(P)	NN	NORM, RAN, SEG, BAL, MI
MLC++ (6)	CART, NN, k-NN, NB, WIN-NOW,CN2	NN	NONE	NONE	NN	NONE	NORM
Mlpy (17)	GLMC, SVM, PC, FDC, k-NN, CART, MLC	HAC, k-M	NONE	PCA, FDA, SRDA, LDA	GLMR, SVM	DTW	FRS, RESAM

Table 5. ML frameworks x methods (cont.)

Name	Classification	Clustering	Association analysis	Factor analysis	Regression analysis	Time-series	Pre-processing
MS SQL Server (16)	NB, CART, NN, GLMC	(P), MCC	(P)	NONE	CART, GLMR, NN	(P)	NORM, DISC, SEG, Aggr., Outliers removal, MI, BINNING
Neuroph (4)	NN	NN	NONE	NONE	NN	NONE	NORM, DISC, SEG
Oracle Data Miner (11)	NB, GLMC, SVM, CART	k-M, O-Cluster	(P)	NONE	SVM, GLMR	NONE	NORM, DISC, SEG
Orange (13)	LR, k-NN, CART, SVM, CN2,	k-M, NN	Apriori	NONE	NN, GLMR, SVM, CART	NONE	NORM, DISC, SEG
PCP (6)	BN, SVM, k-NN, NN, GLMC	NONE	NONE	PCA	NONE	NONE	NONE
Pyml (5)	SVM, k-NN	NONE	NONE	NONE	GLMR, SVM	NONE	FRS, NORM
R (15)	SVM, k-NN, CART, NB, NN	k-M, HAC, EM, Fuzzy c-M, SOM, CLARA	NONE	NONE	GLMR, GLS (?), MARS (?)	ARMA, NN	NORM
RapidMiner (15)	k-NN, NB, SVM, NN, GMM, CART	k-M, EM, k-Medoids, DBSCAN	NONE	NONE	GLMR, NN, SVM	NN	NORM, BINNING, CFSCT
Salford Systems							
SAS Enterprise Miner (5)	CART, NN, SVM	SOM			GLMR, NN,		
scikit-learn (23)	SVM, k-NN, NB, CART, GLMC	GMM, k-M, HAC, DBSCAN, SC, MS, AP	NONE	PCA, Manifold, ICA, NMF, DL	GLMR, SVM	NONE	NORM, LP, FE, KA, FRS
Shogun (21)	SVM, kNN, NN	k-M, GMM, HAC, DBSCAN, Sc, MS, AP	NONE	PCA, Manifold, ICA, NMF, DL	SVM, GLMR	NONE	NORM, LP, FR, KA, FRS

Table 6. ML frameworks x methods (cont.)

Name	Classification	Clustering	Association analysis	Factor analysis	Regression analysis	Time-series	Pre-processing
Statistica (12)	SVM, CART, k-NN, NB, NN, GLMC	k-M, EM, NN	NONE	PCA, FDA	GLMR, NN, SVM	NN	NORM
Weka (16)	NB, SVM, NN, CART, WINNOW, LVQ, SOM, ARIS	HAC, k-M, x-M	Apriori	NONE	NN, GLMR, SVM	NN	NORM, BINNING, CFSCT
YAPLF (3)	NN, SVM	NONE	NONE	NONE	NONE	NONE	NORM

Table 7. ML end-user support

Name	Model visualization	Visual programming	Wizards (templates)	Effort feedback support
Aleph	No	No	No	No
C4.5/C5/See5	No	No	No	No
Encog	Yes	No	Yes	No
FuzzyML	Yes	No	No	No
IBM Cognos	Yes	No	Yes	Yes
IBM SPSS Modeler	Yes	Yes	Yes	Yes
JavaML	No	No	No	No
JHepWork	Yes	No	Yes	No
Joone	Yes	No	No	No
KNIME	Yes	Yes	No	No
LIONsolver	Yes	Yes	No	No
Mathematica Learning Framework	Yes	No	No	No
MATLAB	Yes	No	No	No
MLC++	No	No	Yes	Yes
Mlpy	Yes	No	No	No
MS SQL Server	Yes	No	Yes	No
Neuroph	Yes	No	Yes	No
Oracle Data Miner	Yes	Yes	Yes	No
Orange	Yes	Yes	Yes	Yes
PCP	No	No	No	No
Pyml	Yes	No	No	No
R	Yes	No	No	No
RapidMiner	Yes	Yes	Yes	No
Salford Systems	Yes	No	Yes	No
SAS EnterpriseMiner	Yes	Yes	Yes	No
scikit-learn	No	No	No	No
Shogun	No	No	No	No
Statistica	Yes	No	Yes	No
Weka	Yes	Yes	Yes	No
YAPLF	No	No	No	No

Table 8. ML miscellaneous

Name	Support for parallel-processing	Input formats supported	Model export	Activity on the project
Aleph	Multi-core/GPU	CSV, ARFF, RDBMS	CWM,PMML	Low
C4.5/C5/See5	Multi-core/Hyper-Threading	C4.5	NONE	High
Encog	Multi-core/GPU	CSV, TXT, EGB	Proprietary (XML-based or Java-serialization)	High
FuzzyML	No			
IBM Cognos				
IBM SPSS Modeler		SPSS		
JavaML	No	ARFF, CSV	NONE	High
JHepWork	Multi-core/Hyper-Threading	TXT,CSV,XML,Pfile,pbu xls,database		High
Joone	Multi-core	CSV, TXT	serialisation	Low
KNIME	Multi-core / Grid*	CSV, TXT, ARFF, Data Table	zip, PMML	High
LIONsolver	Multi-core	CSV, Excel, JDBC source	NONE	High
Mathematica Learning Framework	Multi-core	CSV, TXT	NONE	High
MATLAB	Multi-core	CSV, TXT	NONE	High
MLC++	No	TXT, C4.5	custom	Low
Mlpy	No	csv, txt, libSVM	LibSVM, LibLinear	High
MS SQL Server	Multi-core/Hyper-Threading	MSSQL DMX	DMX	High
Neuroph	Multi-core/GPU	CSV, TXT, EGB	Proprietary (XML-based or Java-serialization)	Medium
Oracle Data Miner	Yes	OracleDB	PMML	High
Orange	Soon	CSV, TXT, ARFF, XLS	NONE	High
PCP	No	STS,CSV,TXT	NONE	Low
Pyml	No	CSV,LibSVM, TXT	Python Serialization	Moderate
R	Multi-core/GPU	CSV, TXT, Octvace, ARFF	plugins	High
RapidMiner	Multi-core / Grid*	CSV, Excel, XML, Access, Data Table, Binary File, C4.5, BibTex, SPSS, Stat	XML, PMML	High
Salford Systems	Yes / Grid*		PMML	High
SAS EnterpriseMiner	Yes / Grid*	SAS data sets	plugins	High
scikit-learn	Multi-core	libsvm, csv, txt	LibSVM, LibLinear	High
Shogun	Multi-core	HDF5, libSVM	NONE	High
Statistica	Multi-core/GPU	sta, xls, txt	PMML	High
Weka	Multi-core,	ARFF	model	High
YAPLF	Multi-core	TXT, CSV	NONE	Low