RAFAL RAK
WOJCIECH STACH

05/11/2004

Report on

Evaluation of XLMiner Demo Version 2

CMPUT 695 course assignment no. 2

TABLE OF CONTENTS

1 Introduction

The purpose of this report is to present and evaluate a data mining tool: XLMiner - Demo Version 2. This version of the program is a trial and has several limitations, especially in terms of data size and parameters of several DM techniques.
The remaining of this document is organized as follows. Section 2 contains a short description of the program. The purpose of section 3 is to briefly remind the reader the methods/algorithms that were used in the tool (it is assumed that the reader is familiar with the basic methods of data mining). Section 4 discuses functionalities provided by the tool. In section 5 we attempt to evaluate the program whereas section 6 summarizes the report.

2 Program Overview

XLMiner [11] is an add-in for Microsoft Excel. This data mining tool has been developed by Cytel Software Corp [2], and is distributed by Resampling Stats, Inc. [7] According to the producer the minimum hardware requirements are Pentium 133 MHz processor, 1 GB hard drive with a minimum of 15 MB free disk space, and 32 MB RAM (recommended Pentium 200 MHz processor (or higher), 1 GB hard drive with minimum 60 MB free disk space, and 64 MB RAM (or more)). The program runs on Microsoft Windows 98 / NT 4.0 / 2000 / XP operating system and requires Microsoft Excel 2000/XP to be installed.
Installation process consists of several simple steps using InstallShield Wizard component. Apart from the program files the installation process creates also exemplary datasets to be used while evaluating the tool.

3 Data Mining Methods and Algorithms

3.1 Data Preprocessing

Sampling
Sampling is the method of reducing data to satisfy the time and cost limitations of mining the data. There are various strategies for sampling such as *systematic sampling*, *simple random sampling*, or *stratified random sampling*. Systematic sampling extracts every *n*th record from a database, whereas simple random sampling randomly chooses records that have an equal probability of being chosen. Stratified random sampling first divides data into groups of similar records, called strata, and then randomly chooses records from each stratum.

Missing Values
Missing Values is one of the issues and tasks of the Data Cleaning process. The goal is to solve the problem of empty attributes in tuples. There are several techniques of accomplishing it [3]: we can either ignore the tuple (effective if a tuple contains several missing values), manually fill in the missing value (time-consuming), use a global constant, use the attribute mean/mode/median, or use the most probable value (determined by using regression, inference-based tools, or decision tree induction).

Binning
Binning is a method usually used to solve the problem of *noisy data* i.e. the data that needs to be "smoothed". The method is based on, first of all, partition the data into bins, and, secondly, perform one of smoothing techniques. Generally, there are two types of partitioning the data: *equidepth* (i.e. each bin contains the same number of values) and *equiwidth* (i.e. each bin contains

values of a constant range). The second phase can be achieved using either smoothing by bin means or medians (values in bins are replaced by bin means or medians), or smoothing by boundaries (bin values are replaced by the closest boundary, i.e. minimum or maximum, value in a bin). Another technique, ranking bins by simple labelling, can be useful in categorizing continues values.

3.2 Data Mining

Association Rules
Mining association rules from transactional databases was introduced in [1]. The idea is to extract frequent patterns from a set of transactions and use them to generate association rules. The interesting parameters of generated rules are support (reflecting usefulness), confidence (reflecting certainty), and also correlation or lift (measuring dependency between items).

Classification and Prediction
Generally, classification is the process of organizing data into distinct groups whereas prediction assesses the class of unlabeled sample or, in other words, forecasts the value of an attribute based on the values of other attributes in a given sample. However, in data mining, classification is used to predict discrete or nominal values, and prediction is used to predict continues values.
There are several techniques that can be applied to the task of classification and/or prediction, such as *decision tree induction*, *Bayesian classification*, *classification by backpropagation*, *associative classification* (based on association rule mining), *k-nearest neighbor*, *genetic algorithms*, *fuzzy set approaches*, *support vector machines*, *hidden Markov models*, *radial basis functions*, and others [3] [5].

Clustering
Clustering is a process of grouping similar objects into classes. There are several methods of clustering such as *partitioning*, *hierarchical* methods, *density-based* methods, *grid-based* methods, or *model-based* methods [3]. Partitioning utilizes k-*means* algorithm or k-*medoids* algorithm to find clusters. Hierarchical methods can be classified as being either *agglomerative* (the bottom-up strategy that starts with clusters represented by single objects and merges them to create less but bigger groups) or *divisive* (the top-down strategy that starts with one cluster representing all the objects and divides it into smaller groups). Density-based methods create clusters based on the number of objects in the neighbourhood. Grid-based methods divide the object space into cells forming a grid structure whereas model-based methods hypothesize a model for each cluster and try to find the best fit of the data to the model.

3.3 Visualization

There are several graphs for display of data summaries and distribution. A *box plot* is used for the visual representation of distribution. *Histograms* reflect the count or frequencies of objects or classes. A *scatter plot* is a graphical method for determining relationships, patterns, or trends between two variables.

4 Functionalities

The general procedure of using tool's methods consists of several steps as follows:

1. Opening a sheet containing data to be processed.
2. Choosing functionality from the menu (see Figure 1).

3. Setting parameters adequate to the chosen functionality in a dialog box or a set of dialog boxes (see Figure 2).
4. Retrieving results in separate sheets (see Figure 3).

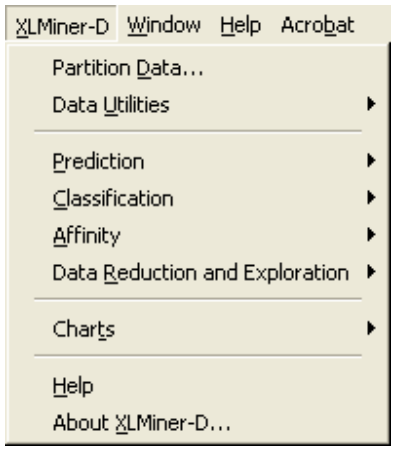

Figure 1 XLMiner embedded menu

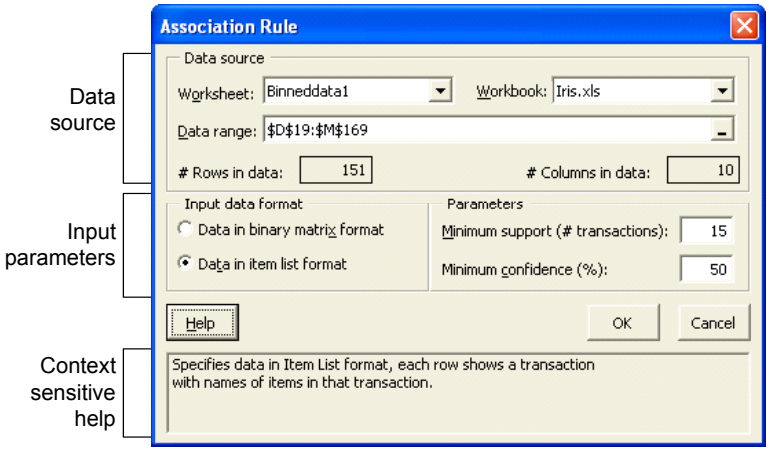

Figure 2 Dialog box structure (example)

Output Navigator

Inputs | Bin Interval | Output

Inputs

Data				
Data Source	Data!$A$1:$F$151			
# records	150			
Binning variables	Petal_width	Petal_length	Sepal_width	Sepal_length
Binning variable name	Binned_Petal_width	Binned_Petal_length	Binned_Sepal_width	Binned_Sepal_length
#bins of selected variable	5	5	13	8
Binning type	Equal count	Equal count	Equal count	Equal interval
Binning value type	Rank	Mean	Median	Rank

Output

Row Id.	Species_No	Petal_width	Petal_length	Sepal_width	Sepal_length	Species_name
1	1	2	14	33	50	Setosa
2	1	2	10	36	46	Setosa
3	1	2	16	31	48	Setosa
4	1	1	14	36	49	Setosa
5	1	2	13	32	44	Setosa
6	1	2	16	38	51	Setosa
7	1	2	16	30	50	Setosa

Figure 3 Output sheet (example)

4.1 Data Utilities

4.1.1 Sampling

Input data can be loaded from either worksheet file or MS Access database file. The sampling dialog box allows a user to choose the fields to include in the results. The tool requires specifying a primary key for the loaded data.

A user specifies a desired sample size and chooses between two different methods of sampling, either *Simple Random Sampling* or *Stratified Random Sampling*. Simple Random Sampling supports replacements i.e. a single record may be chosen more than once (this feature is optional). Choosing Stratified Random Sampling enables a box with additional parameters from which a user can set a stratum variable (i.e. the field in a record on which similarity comparison will be performed) and the method of picking records from strata. There are three different methods of picking records: proportionate to the size of each stratum, equal from each stratum specified by a fixed number, and equal from each stratum indicated by the size of the smallest stratum.
The output is simply the input cropped to the desired sample size and the chosen fields.

4.1.2 Missing Data Handling

The parameters of Missing Data Handling method allow a user to specify how missing values are represented in the input data (by default, empty cells are considered to be invalid).
A user can choose which method to use to handle missing values. The method can be chosen from either deleting a record, using calculated mean/median/mode, or specifying the value manually. Each attribute can be solved by individual method.
The output sheet contains a table of fixed records as well as details regarding the parameters used in the process. Each affected value is visually marked and the number of affected values per attribute is displayed.

4.1.3 Bin Continues Data

The tool allows a user to specify an individual binning technique to each attribute. The number of bins is automatically set based on the range of each attribute. This number can be changed manually. The tool supports both *equidepth* and *equiwidth* technique named here *equal count* and *equal interval* respectively. There are three methods of smoothing values: the rank of the bin (initial number and increment value can be set manually), the mean of the bin, and the median of the bin.
The output sheet consists of information about the input parameters; a table containing processed records (every attribute that has been marked to be binned has its corresponding binned attribute); and binning intervals information (including the number of records in a bin) for each binned attribute.

Output

Row Id.	Species_No	Petal_width	Petal_length	Sepal_width	Binned_Petal_width	Binned_Petal_length	Binned_Sepal_width	Binned_Sepal_length
1	1	2	14	33	1	13.87	34	2
2	1	2	10	36	1	13.87	37	1
3	1	2	16	31	1	25.5	31	2
4	1	1	14	36	1	13.87	37	2
5	1	2	13	32	1	13.87	32	1
6	1	2	16	38	1	25.5	37	2
7	1	2	16	30	1	25.5	30	2
8	1	4	19	38	2	25.5	37	2

Binning interval

Petal_width			Petal_length			Sepal_width			Sepal_length		
Intervals value		#records	Intervals value		#records	Intervals value		#records	Intervals value		#records
From	To		From	To		From	To		From	To	

1	2	34	10	15	37	20	25	19	43	48	11
3	11	26	16	39	24	26	26	6	48	53	34
12	15	39	40	47	31	27	28	23	53	58	28
16	19	23	48	54	29	29	30	35	58	63	26
20	25	28	55	69	29	31	31	11	63	68	31
						32	32	13	68	73	12
						33	34	18	73	78	7
						35	35	6	78	79	1
						36	38	13			
						39	44	6			

Figure 4 Output sheet for Bin Continues Data

4.1.4 Transform Categorical Data

The tool provides two methods of transforming categorical data: creating *dummies* i.e. a set of binary exclusive attributes (only one attribute in a tuple is set to 1 and this attribute represents processed categorical value in the tuple); and creating *category scores* i.e. each distinct categorical value of an attribute is numbered.
A user can choose more than one attribute to be transformed.
The output sheet displays the input data extended by the additional attributes.

4.2 Partition Data

XLMiner provides two methods for partitioning data into training, validation, and test sets: *random partitioning* and *user-defined partitioning*. The former method allows for entering the value for random number seed, which can be helpful if user wants to have the data partitioned in the same way for successive experiments. User can also enter the desired percentage of examples of each type. The latter method requires data having additional column for denoting the type of the desired set (t – training, v – validating, s – testing). The drawback is that these symbols are fixed and cannot be customized.
The result of this operation is the dataset divided into the sets: training, validation, and testing.

4.3 Prediction

This group consists of four tools designed for prediction the value of the continuous outcome variable. All the process involves several steps that depend on chosen prediction method (see subsections 4.3.1 - 4.3.4). However, regardless of the selected method, the first step requires setting a number of parameters, which can be divided into two groups. The first one consists of *input parameters*, such as data source, input variables, output variable, and, in case of multiple linear regression, weight variable (if in the data exists a variable that assigns weights to the different rows (cases)), see Figure 5

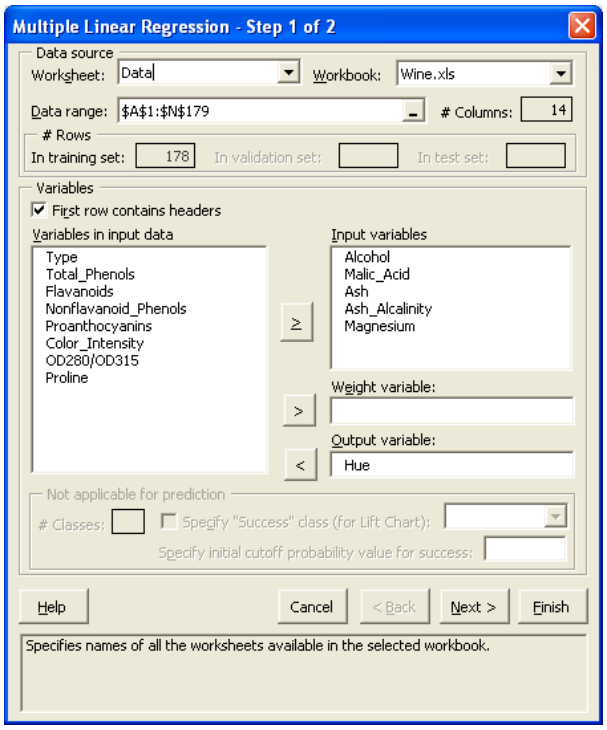

Figure 5 Dialog box for setting input parameters for prediction

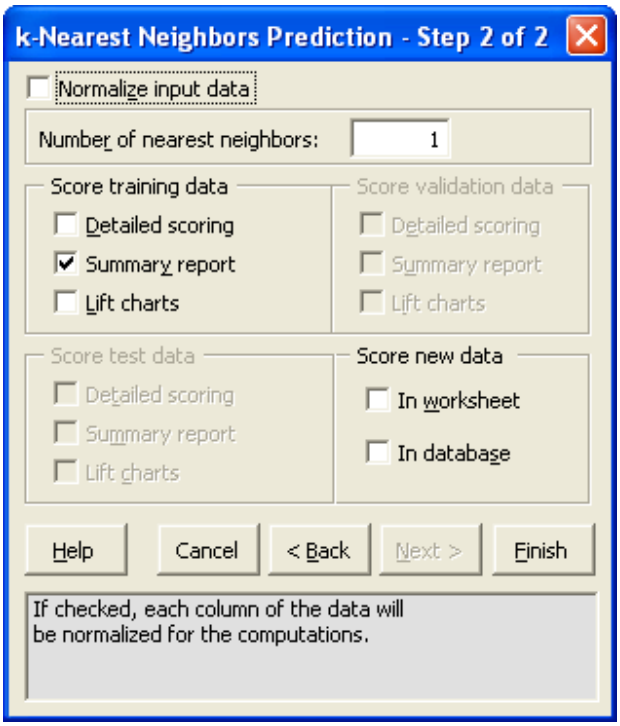

Figure 6 Dialog box for setting output parameters for prediction

The second group of options, which are uniform regardless of the prediction method, are those regarding *output format* (see Figure 6). The elements, which are common for all prediction methods, are as follows:

1. *Detailed scoring* lists all examples and matches to them predicted values,
2. *Summary report* in each case includes information on
    - Input - data, variables, parameters/options (they depend on the chosen model) and output options;
    - Data scoring – three errors are calculated, i.e. total sum of squared errors (deviations between predicted and actual values), the root mean square error (square root of the average squared error), and the average error (deviations between predicted and actual values);
    - Elapsed time – time that was necessary to perform the prediction.
3. *Lift chart* - visual aids for measuring model performance

The subsections 4.3.1- 4.3.4 present the prediction methods available in XLMiner. The descriptions are divided into two categories*, input parameters,* which includes all parameters that can be set for a particular model, and *output format*, which shows options available using particular model. Both of these groups include only additional features that are not common for all models, since they are outlined above.

4.3.1 Multiple Linear Regression

- **Input parameters**
    - the constant term in the equation can be set to zero;

- additional options that includes elements that are shown as a result of regression, such as ANOVA table, fitted values, variance-covariance matrix, and residuals (standardized and unstandarized). Advanced options allows for setting up additional statistics to be displayed, such as Cook's distance or Hat matrix diagonals, collinearity diagnostics and studentized/deleted residuals.

- **Output format**
  - a table including standard regression output (coefficients, standard error, p-value, sum of squared error for each input variable).
  - tables that correspond to user-chosen options, such as ANOVA table or variance-covariance matrix, see Figure 7.

The Regression Model

Input variables	Coefficient	Std. Error	p-value	SS
Alcohol	0.06957125	0.01173211	0	162.1984406
Malic_Acid	-0.11641926	0.01459007	0	2.7424531
Ash	0.07249416	0.07151553	0.31214902	0.16086084
Ash_Alcalinity	0.00263285	0.00517368	0.61147487	0.0086948
Magnesium	0.00098253	0.00116366	0.39964238	0.03000516

Residual df	173
Multiple R-squared	0.957771096
Std. Dev. estimate	0.20515293
Residual SS	7.28117657

ANOVA

Source	df	SS	MS	F-statistic	p-value
Regression	4	165.140454	41.28511361	980.9300168	1.0933E-117
Error	173	7.28117657	0.042087726		
Total	177	172.421631			

Variance-Covariance Matrix

Figure 7 Example of multiple linear regression result

4.3.2 k-Nearest Neighbors

- **Input parameters**
  - number of nearest neighbors;
  - decision if the data should be normalized (this expresses all data in terms of standard deviations so that the distance measure is not dominated by variables with a large scale);

4.3.3 Regression Tree

- **Input parameters**
  - decision if the data should be normalized;
  - maximum number of splits for input variables;
  - minimum number of records in a terminal node;
  - decision if the tree should be displayed in graphical way (user can select either full, pruned tree or both and maximum number of levels being displayed);
- **Output format**
  - set of decision rules;

- graphical tree representation (if selected);

4.3.4 Neural Network (Multilayer feedforward)

- **Input parameters**
  - decision if the data should be normalized;
  - network architecture (number of hidden layers up to 4 and number of nodes in each layer);
  - training options (number of epochs, step size for gradient descent, weight change momentum, error tolerance, and weight decay);
- **Output format**
  - tables including set of weights among nodes in the neural network, which represent network architecture;

4.4 Classification

This group consists of six tools designed for classifying the discrete or categorical outcome variable into the discrete classes or categories.
The usage of the classification tools from the user perspective is very similar to usage of those for prediction. The dialog box for setting input parameters looks exactly the same, see Figure 5, except that the hidden option that shows number of classes is now visible (the user can also specify "success class"). All common options for output format are available in this case as well, (predicted values are replaced with class name).
The subsections 4.4.1 - 4.4.6 describe the available classification methods following the same convention as it was done for prediction techniques, see section 4.3.

4.4.1 Discriminant Analysis

- **Input parameters**
  - prior class probability that will be used by discriminant analysis procedure (it can either exploit knowledge about number of occurrence of each class in the data set, or assume that classes occur with equal probability)
  - decision if canonical variate loadings should be included in the result (the *canonical variates* are created for the data which is based on an orthogonal representation of the original variates. This has the effect of choosing a representation, which maximizes the distance between the different groups. For a k class problem there are k-1 Canonical variates. Very often only a subset (say g) of the canonical variates is sufficient to discriminate between the classes)
- **Output format**
  - Classification function;
  - canonical variate loadings (optional)

.

Prior class probabilities

According to relative occurrences in training data

Class	Prob.
A	0.331460674
B	0.398876404
C	0.269662921

Classification Function

Variables	Classification Function		
	A	B	C
Constant	-38.3051758	-37.5625382	-38.284996
Malic_Acid	1.62582397	1.26700199	2.74496078
Ash_Alcalinity	1.85481739	2.31832266	2.42944956
Total_Phenols	13.91881466	10.58930111	7.60512352

Canonical Variate Loadings

Variables	Variate1	Variate2
Malic_Acid	0.02402827	0.07360768
Ash_Alcalinity	0.01313573	-0.01524041
Total_Phenols	-0.14215432	0.02308882

Training Data scoring - Summary Report

Classification Confusion Matrix			
	Predicted Class		
Actual Class	A	B	C
A	47	12	0
B	13	46	12
C	0	7	41

Error Report			
Class	# Cases	# Errors	% Error
A	59	12	20.34
B	71	25	35.21
C	48	7	14.58
Overall	178	44	24.72

Figure 8 Result of discriminant analysis classification

4.4.2 Logistic Regression

- **Input parameters**
  - the constant term in the equation can be set to zero;
  - confidence level for odds (it is used to alter the level of confidence for the confidence intervals displayed in the results for the odds ratio)
  - advanced options make possible to set the maximum number of iterations (for iterative process of estimating the coefficients), initial marquardt overshoot factor (which is a part of the above-mentioned iterative procedure), and collinearity diagnostics (it helps dealing with situations, in which variables are highly correlated with one another resulting in large standard errors for the affected coefficients);
  - best subset feature allows for selecting subset of variables, which can be useful in case that this subset (instead of all variables) does the best job of classification;
- **Output format**
  - regression model;
  - covariance matrix of coefficients;
  - residuals;

4.4.3 Classification Tree

- **Input parameters**
  - decision if the data should be normalized;
  - decision if the tree can be pruned using validation data;
  - options for drawing the tree in the output (available options are: full tree, best pruned tree, minimum error tree, and tree with specified number of decision nodes);
- **Output format**
  - set of decision rules;

- graphical version of tree are also available

4.4.4 Naïve Bayes Classifier

- **Input parameters**
  - option for calculating prior class probabilities (equal or according to the relative occurrences in training data);
- **Output format**
  - table with conditional probabilities for each class

4.4.5 Neural Networks (Multilayer feedforward)

- **Input parameters**
  - decision if the data should be normalized;
  - network architecture (number of hidden layers up to 4 and number of nodes in each layer);
  - training options (number of epochs, step size for gradient descent, weight change momentum, error tolerance, and weight decay);
  - cost function (squared error, cross entropy, maximum likelihood and perceptron convergence), hidden and output layer sigmoid function (logistic or symmetric);
- **Output format**
  - tables including set of weights among nodes in the neural network, which represent network architecture;

4.4.6 k-Nearest Neighbors

- **Input parameters**
  - number of nearest neighbors;
  - decision if the data should be normalized (this expresses all data in terms of standard deviations so that the distance measure is not dominated by variables with a large scale);

4.5 Association Rules

There are two types of data the tool can deal with: data in a binary list format (i.e. each transaction is represented as a binary string, where "1" denotes the existence of an item in a transaction) and data in an item list format (i.e. each transaction contains a set of item identifiers). The two parameters a user can set are a minimum support (as a number of transactions) and minimum confidence (in percentage).
The result consists of information about the input data and the table of generated rules including antecedent, consequent, their individual support, combined support, confidence, and correlation (lift) between them.

Rule 6: If item(s) ChildBks, RefBks= is / are purchased, then this implies item(s) CookBks, DoltYBks is / are also purchased. This rule has confidence of 61.11%.

Rule #	Conf. %	Antecedent (a)	Consequent (c)	Support(a)	Support(c)	Support(a U c)	Lift Ratio
1	80	ArtBks, DoltYBks=>	ChildBks, CookBks	30	54	24	2.962963
2	73.33	ArtBks, GeogBks=>	ChildBks, CookBks	30	54	22	2.716049
3	73.33	DoltYBks, GeogBks=>	ChildBks, CookBks	30	54	22	2.716049
4	63.16	ArtBks, ChildBks=>	CookBks, DoltYBks	38	47	24	2.68757
5	51.06	CookBks, DoltYBks=>	ArtBks, ChildBks	47	38	24	2.68757
6	**61.11**	**ChildBks, RefBks=>**	**CookBks, DoltYBks**	**36**	**47**	**22**	**2.600473**

Figure 9 The output sheet of Association Rules mining

4.6 Data Reduction and Exploration

4.6.1 *k*-Means Clustering

This process consists of three steps. The first step is simply choosing the input data and specifying the input attributes to be processed (by default, all the numerical attributes are included). The second step allows a user to specified the desired number of clusters and the number of iterations i.e. how many times the process will start with an initial partition (the iteration that minimizes the distance measure between clusters will be chosen as the result of operation). A user can also choose to normalize the input data before processing. The last step allows for specifying additional output information such as *data summary* and *distances from each cluster center*.

The output sheets consist of: input information; the table of cluster centers for each processed attribute; distances between clusters; data summary such as the number of objects in each cluster and average distance in a cluster; and of course the clustered data itself as a set of records with an additional attribute – cluster ID.

4.6.2 Hierarchical Clustering

The method consists of three steps. The first step requires choosing the input data and specifying the type of data. There is a choice between *raw data* and *distance matrix*. A user also has to choose attributes to be processed. The second step allows a user to specify whether the input data should be normalized. One of the following similarity measures has to be selected: Euclidean distance, Jaccard's coefficients, and matching coefficients. The latter two can be selected only if the input data is binary. A user can also decide which clustering method should be involved in the process. It can be either single linkage, complete linkage, average linkage, average group linkage, or Ward's method. The last step allows a user to set whether the output sheets should include dendrogram and specify the number of clusters.

The output sheets consist of: input information, a clustering stages table, predicted clusters, and the dendrogram.

Row Id.	Cluster Id	Sub Cluster Id	x1	x2	x3	x4
1	1	1	1.06	9.2	151	54.4
2	1	2	0.89	10.3	202	57.9
3	1	3	1.43	15.4	113	53
4	1	4	1.02	11.2	168	56
5	2	5	1.49	8.8	192	51.2
6	1	6	1.32	13.5	111	60
7	3	7	1.22	12.2	175	67.6
8	4	8	1.1	9.2	245	57
9	1	9	1.34	13	168	60.4
10	1	10	1.12	12.4	197	53
11	4	11	0.75	7.5	173	51.5
12	3	12	1.13	10.9	178	62
13	1	13	1.15	12.7	199	53.7

Figure 10 Hierarchical clustering: predicted clusters

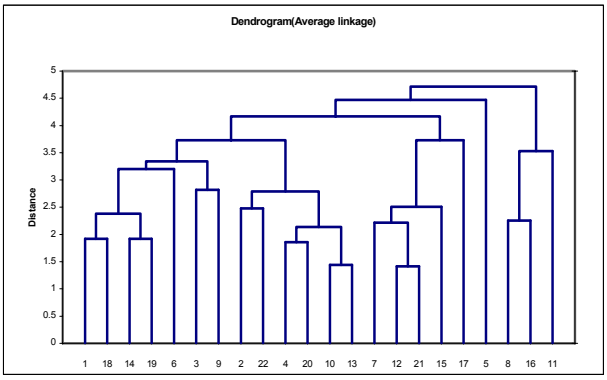

Figure 11 Hierarchical clustering: dendogram

4.7 Charts

All the three kinds of chart, box plot, histogram, and matrix plot (a set of scatter plots), require specifying input variables and (optionally) a chart description. The box plot chart is extended of an option indicating whether the chart should display the notch of confidence interval around the mean of the input set. The box plot chart allows also for choosing more than one set of input variables. The matrix plot requires at least two input sets.
The result of usage of these three kinds of chart is shown in Figure 12, Figure 13, and Figure 14.

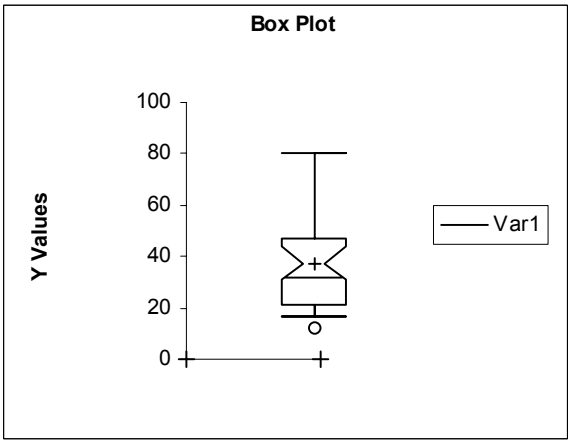

Figure 12 Box plot

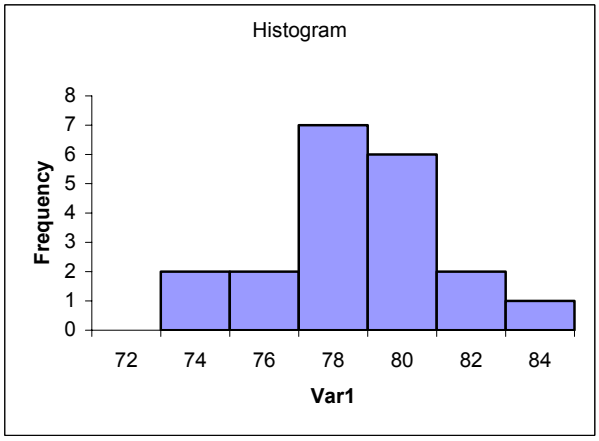

Figure 13 Histogram

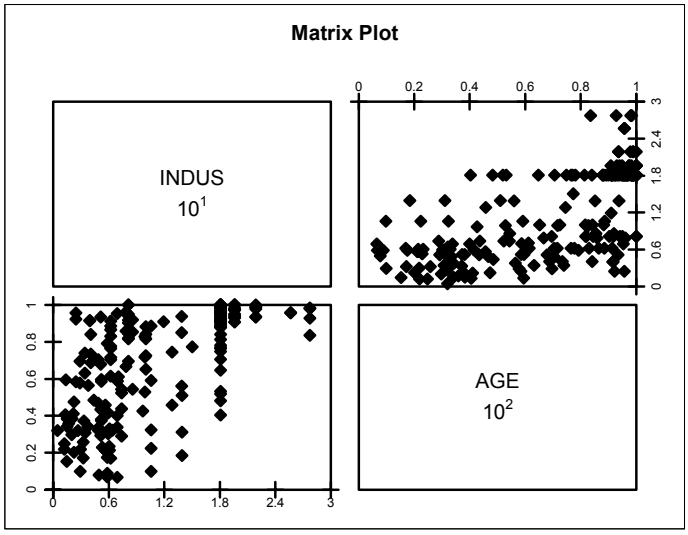

Figure 14 Matrix plot for two sets of input variables

5 Evaluation

Due to the fact that we tested the demo version of XLMiner (see Appendix B Limitations) we were not able to evaluate such parameters as scalability or efficiency. All experiments performed on size-constrained datasets raised the output within few seconds. As a result, we decided to focus our attention to such criteria as applicability, usability, and functionality of the tool. The following subsections summarize our observations.

5.1 Applicability

XLMiner is dedicated to work with Excel sheet data. A sheet-based representation gives a user flexible access to data and possibility to use the data with other Excel built-in tools. On the other hand, that kind of data representation leads to substantial constraints, as nowadays more and more data is stored in database systems (or data warehouses), which makes this tool useless, or costly because of necessity of transferring data between these instances. Some functionalities supports data loaded from external databases, yet, surprisingly, this feature is limited only to few of them. Moreover, the only supported database is Microsoft Access.

Unlike other commercial products, such as PolyAnalyst [6], XLMiner is not focused on particular application (e.g. *text analysis* or *finding dependencies*). Rather, this is a set of methods to deal with different tasks in data mining, limited to only raw data representation without the interpretation of results.

Although the tool includes elements that can be applied in different stages throughout the knowledge data discovery process (such as data preprocessing, data mining, and visualization), XLMiner does not provide the functionality to combine all those elements into a single process (as it is done e.g. in Tanagra [8]).

5.2 Usability

As an add-in for Microsoft Excel, XLMiner derives all the Microsoft Office suite facilities.

A user who is familiar with Microsoft Office environment will find usage of XLMiner easy and intuitive. The tool is installed as an additional MS Excel menu item. This item is divided into several parts corresponding to different stages of the KDD process, which makes it ease and fast to navigate between data mining procedures.
All the procedures follow the same scenario of usage. Procedure dialog boxes are standardized and contain well-separated parts grouping different types of parameters. Each dialog box additionally is equipped with *context sensitive help* that gives a user quick idea of how to set the chosen parameter. Majority of parameters has its default values set, often adjusted to input data. Every attempt of setting parameters that are in contradiction to each other or have improper values is reported and a possible solution is suggested.
We also encountered several unexpected behaviours that resulted in program crashing; though we cannot imply for sure that it was caused by XLMiner itself as we run other programs simultaneously. In terms of usability we found several minor problems, like not supporting of all keyboard shortcuts at every stage of program usage, e.g. CTRL+A for selecting all items, yet they do not distort the positive general impression of usability of this tool.

5.3 Functionality

We found the serious drawback of the tool is a lack of procedures chaining, i.e. it is not possible to create all the steps of the KDD process in a single run. Rather, if a user wants to go through the whole process, he/she has to perform each step manually.
Number of available DM tools is limited and does not include several powerful ones, e.g. Support Vector Machines, Hidden Markov Model, or Radial Basis Functions. XLMiner is a commercial product and in the opposite to some free open-source programs (such as Weka or Tanagra) does not allow for creating our own procedures and include them to the tool.
Moreover, several built-in methods have limited functionalities due to the fact that some parameters are fixed and cannot be changed, e.g. in case of Neural Networks is not possible to choose transfer function or network learning procedure. This, in consequence, limits application of these techniques and makes impossible to take advantage of all of their features.
Association Rules seems to be developed only for basket analysis. The results are confined only to the list of rules and do not contain the explicit list of frequent itemsets, which could be used in other techniques of mining using frequent patterns.

5.4 Correctness

In order to check the correctness of the results, we also tested XLMiner on small datasets performing limited number (no more than two per each method). We chose small datasets to be able to assess the results by ourselves. The results turned out to be correct in each examined case. However, since the lack of comprehensive experiments we cannot generalize this hypothesis. Moreover, the tests involved only methods, which raise the same output regardless of the number of experiments, e.g. neural network learning does not satisfy this condition.

5.5 Support

The web page of the program is available at [11]. The service provides such information as product description, ordering details, online support.
The web service seems to be a comprehensible resource; however, some pages are not up-to-date (e.g. User Guide, Bug reports & Patches, Capabilities).

Although the program support is limited to e-mail correspondence only, we experienced it to be very fast (few hours) and reliable.

6 Conclusions

Evaluating several different data mining tools raises some reflections on how a perfect mining tool should look like. An ideal tool should at least satisfy the following characteristics.

1. Support different sources of data, especially databases and warehouses.
2. Allow a user to create the whole KDD process from components and perform it in a single run. The process should also be easy to control and modify.
3. Support several well-known KDD strategies/applications (e.g. *basket analysis*)
4. Provide a broad set of tools on each level of the KDD process.
5. Allow a user to create his/her own algorithms/modules/procedures.
6. Provide flexible and user-friendly interface.
7. Produce comprehensible and easy-to-interpret results.
8. Perform procedures efficiently.
9. Assure reliable support from the producer.
10. Provide complete documentation with tutorials.
11. Support different platforms (e.g. Windows, Unix)

Comparing XLMiner to our "perfect tool" we can count out several shortages as it is shown in the previous section. The ratio "price to capabilities" of XLMiner in comparison to other products on the market (especially free yet more powerful ones such as Weka or Tanagra) is not promising.
However, XLMiner as a commercial product has a full support from the producer, which is very rare in the case of free open-source tools.
As in most cases, this product is addressed to the certain group of users. XLMiner seems to be dedicated for educational purposes and small business solutions, rarely as a professional tool used in research and business analysis. XLMiner may turn out to be insufficient for those who expect an advanced, flexible, and powerful data mining tool.

Appendix A Supported Data Types

Table 1 shows the data types of both input and output variables a user can provide or retrieved while performing a particular method.

Table 1 Data types supported in XLMiner (source: [10])

Method	Output variable			Input variables		
	Continuous	Categorical		Continuous	Categorical	
		Ordinal	Nominal		Ordinal	Nominal
Multiple Linear Regression	Y	Y	N	Y	Y	N
k-Nearest Neighbor Prediction	Y	Y	N	Y	Y	N
Regression Tree	Y	Y	N	Y	Y	N
Discriminant Analysis	N	Y	Y	Y	Y	N
Logistic Regression	N	Y (D)	N	Y	Y	N
Classification Tree	N	Y	Y	Y	Y	N
Naïve Bayes	N	Y	Y	N	Y	Y
Neural Networks	N	Y	Y	Y	Y	N
k-Nearest Neighbor Classification	N	Y	Y	Y	Y	N
Association Rules	-	-	-	N	Y (D)	Y
Principal Component Analysis	-	-	-	Y	Y	N
k-Means Clustering	-	-	-	Y	Y	N
Hierarchical Clustering	-	-	-	Y	Y	N

Legend: Y - Supported; N - Not supported; D - Dichotomous - Binary value (0,1)

Appendix B Limitations

XLMiner Demo Version 2 (Build # 19.11, 29-Mar-2004) has several limitations gathered in Table 2.

Table 2 Limitations on the demo version of XLMiner (source: [10])

Partitioning	# Rows	Original data : No limit
		Output : 600, subject to training partition being not more than 200.
	# Columns	Original data : No limit
		Output : 200
Sampling from worksheet	# Rows	Original data: No limit
		Sample output: 200
	# Columns	Original data : No limit
		Output : 200
	# categories for Stratum variable (in Stratified Sampling)	30 (Stratum values are not case sensitive)
Sampling from database	# Fields	In the table: No limit
		Sample output: 200
	# Records	In the table : 200
		Sample Output : 200
	# categories for Stratum field (in Stratified Sampling)	30 (Stratum values are not case sensitive)
Handle Missing values	#Columns in the data range	199
	# Rows	200
	# Columns	200
Binning	#Columns in the data range	199
	# Rows	200
	# Columns	200 (Inclusive of all columns in the data range and output binned columns)
Transform Categorical Data	#Rows	200
	# Columns	200 (Inclusive of all columns in the data range and the ones added in the outpuut.)
Classification and Prediction	# Rows	200 for each partition (Training, Validation, Test) if partitioning is used.
		200 if partitioning is not used.
		200 in new data used as Scoring target
	# Columns (input variables)	30 (The dataset can contain upto 200 columns, out of which, upto 30 can be selected for the model as input variables)
	# Distinct classes for a categorical variable	30 (Class values are not case sensitive)
	# Distinct values for any input variable for Naive Bayes classification	30 (Values are not case sensitive)
	# Nearest neighbors for k-Nearest Neighbors	20 (or # Training rows whichever is smaller)
	# Splits for Regression Tree	5000 (or [# Training rows -1] whichever is smaller)
	# Levels in Tree drawing for Regression and Classification trees	7 (Actual tree may contain more levels)
	# Epochs for Neural Networks	200
	# Iterations for Logistic Regression	50
Affinity - Association Rules	# Transactions	200
	# Distinct items in dataset (In item list format)	1000
	# Columns (In Item List Format)	30. These are maximum #items in a transaction 30.
	# Columns (In Binary Matrix Format)	30
	# Rules	60000 (Additional rules may exist, they are not displayed)
Data Exploration & Reduction	# Rows	200
	# Columns (variables)	30. (The dataset can contain upto 200 columns, out of which, upto 30 can be selected for the model as variables.)

	# Clusters displayed in a Dendrogram	30 (The solution may involve a higher number of clusters, but the Dendrogram shows a maximum of 30 top-level clusters)
	Size of Distance Matrix (if specified) for Hierarchical Clustering	200 x 200
	# Clusters for k-Means clustering	20 (or # Training rows whichever is smaller)
	# Iterations for k-Means clustering	50
Charts	# Rows	200
	# Columns	Original Data : 200
		For charts drawing : 5 (For Box & Matrix plots)
	# Distinct values X-variable can take	5 (for Box plot)
General	# Worksheets in workbook (Excel File)	245, before running any XLMiner™ procedure (Count includes any hidden/very hidden sheets also which may be present in the workbook)

Appendix C Program Cost

The cost of the program depends on the purpose and period of using (see Table 3).

Table 3 Cost of XLMiner – valid on the day of writing this report (source: [11])

Education	
Single User Education version, download. Licensed for one year	$99.00
Single User Education version, CD delivery. Licensed for one year	$109.00
Single User Classroom version. Minimum pack of 10 copies adopted for a classroom	$49.00
Classroom site license for computer lab. Licensed for one year	$1499.00
Standard	
Single User Commercial License. Full feature, 2-year license	$899.00
Single User Academic License. Full feature, 1-year license. CALL Network (multi-user) License	$199.00

References

[1] Agrawal R., Imielinski T., and Swami A., Mining Association Rules between Sets of Items in Large Databases, *Proc. ACM Conf. on Management of Data*, pp. 207-216, 1993

[2] Cytel – Statistical Software, http://www.cytel.com/home/default.asp

[3] Han J., and Kamber M., Data Mining: Concepts and Techniques, *Morgan Kaufmann,* 2001

[4] Hand D., Mannila H., and Smyth P, Principles of Data Mining, *The MIT Press*, 2001

[5] Hastie T., Tibshirani R., Friedman J., The Elements of Statistical Learning, Data Mining, Inference, and Prediction, *Springer,* 2003

[6] PolyAnalyst Web Page, http://www.megaputer.com/products/pa/index.php3.

[7] Resampling Stats - http://www.resample.com/

[8] Tanagra Web Page, http://chirouble.univ-lyon2.fr/~ricco/tanagra/en/tanagra.html.

[9] Weka Web Page, http://www.cs.waikato.ac.nz/ml/weka/index.html

[10] XLMiner - *Online Manual* for the program package (Build # 19, Demo Version)

[11] XLMiner Web Page, http://www.xlminer.net/