

SOFTWARE TOOLS FOR TEACHING UNDERGRADUATE DATA MINING COURSE

Ashwin Satyanarayana
asatyanarayana@citytech.cuny.edu
New York City College of Technology,
N-913, 300 Jay Street, Brooklyn, NY - 11201

Abstract: Data mining, a growingly popular field in Computer Science, is the transformation of large amounts of data into meaningful patterns and rules. Recent studies have noted the rise of data mining as a career path with increasing opportunities for graduates. Data mining introduces new challenges for faculty in universities who teach courses in this area. One of the main challenges for faculty would be to identify which software tool to use to introduce this subject in a one semester undergraduate course. In this paper, we compare and contrast three popular commercial and three popular open source tools that are available for faculty.

Keywords: Data Mining, SAS, IBM SPSS Modeler, MATLAB, R, WEKA, RapidMiner

Introduction:

Enormous amounts of data are generated every minute. Some sources of data, such as those found on the Internet are obvious. Social networking sites, search and retrieval engines, media sharing sites, stock trading sites, and news sources continually store enormous amounts of new data throughout the day [5].

We are in a new era in modern information technology - the “Big Data” era. In March, 2012, the U.S. Government announced a “Big Data Research and Development Initiative” -- a \$200 million dollar commitment to improve our ability to “extract knowledge and insights from large and complex collections of digital data.” Government agencies such as NSF, NIH, and DoD are investing hundreds of millions of dollars toward the development of systems that can help them extract knowledge from their data.

The career potential for our graduates continue to blossom in this field. A recent study released by Gartner projects that in 2013, “big data is forecast to drive \$34 billion of IT spending,” with a total of \$232 billion to be spent through 2016 [1].

Over the last 10 years, a number of commercial and open source tools have been developed to examine and transform data. In this paper, an overview of some tools are presented that can be used to teach data mining in a one semester course. We present 3 of the most popular commercial tools and 3 open source tools.

Tools:

The ultimate goal of data mining is to create a model, a model that can improve the way you read and interpret your existing data and your future data. Since there are so many commercial and open source tools available for data mining, the major step is to choose the right tools to create a good model. From there, the tools are then used to refine the model to make it more useful.

There are 3 main needs that we would like to achieve for the tools that we pick for the course for teaching a single semester basic data mining course:

1. Easily available for students to download
2. Easily to load data, build models, and interpret results
3. Should work with big data

According to a survey by KDnuggets [2], the most popular tools used by 900 data miners, here are the most popular data mining tools used by real projects:

Name of the Tool	% of users	Commercial (C)/Open Source (O)
RapidMiner	37.80%	O
R	29.80%	O
Weka	14.30%	O
SAS	12.00%	C
Matlab	9.20%	C
IBM SPSS Statistics	7.90%	C

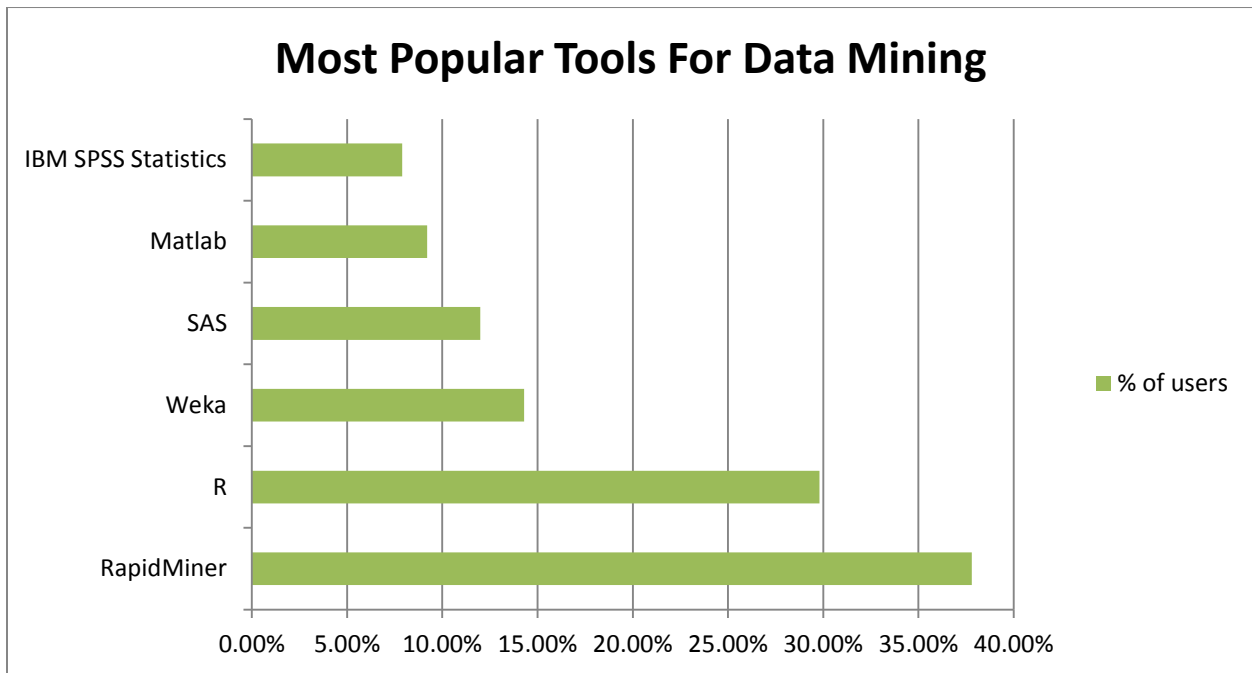


Figure 1. Most popular tools for Data Mining (according to KDnuggets poll 2010)

Commercial Tools:

In this section we will discuss the most popular commercial tools that are available for real projects. One of the benefits of using commercial tools in universities is that students will get hands on experience with working with tools that are used in industry. However one of the drawbacks of using commercial tools is that they are expensive and not easily accessible to students to download. Considering this, it would be best to use commercial tools for an advanced data mining course and not for an introductory course. We will see later in this paper, that using open source tools for introductory courses is ideal.

In this paper we will discuss 3 popular commercial tools:

- (a) SAS Enterprise Miner
- (b) MATLAB
- (c) IBM SPSS Modeler

SAS Enterprise Miner

SAS Enterprise Miner software streamlines the data mining process to create highly accurate predictive and descriptive models. The models are based on analysis of vast amounts of data from across an enterprise. Interactive statistical and visualization tools help you better search for trends and anomalies and help you focus on the model development process.



Here are good sources of information that can be used for getting an introduction to this tool:







1. <http://liqing.cai.swufe.edu.cn/SAS%20Mining.pdf>
2. <http://support.sas.com/publishing/pubcat/chaps/59829.pdf>

Enterprise Miner nodes are arranged into the following categories according the SAS process for data mining: SEMMA.

- Sample — identify input data sets (identify input data; sample from a larger data set; partition data set into training, validation, and test data sets).
- Explore — explore data sets statistically and graphically (plot the data, obtain descriptive statistics, identify important variables, perform association analysis).
- Modify — prepare the data for analysis (create additional variables or transform existing variables for analysis, identify outliers, replace missing values, modify the way in which variables are used for the analysis, perform cluster analysis, analyze data with self-organizing maps (known as SOMs) or Kohonen networks).
- Model — fit a predictive model (model a target variable by using a regression model, a decision tree, a neural network, or a user-defined model).
- Assess — compare competing predictive models (build charts that plot the percentage of respondents, percentage of respondents captured, lift, and profit).

Since most of the work of SAS Enterprise mining deals with Nodes, here are some sample notes:

 Input Data Source	The Input Data Source node reads data sources and defines their attributes for later processing by Enterprise Miner.
 Sampling	The Sampling node enables you to perform random sampling, stratified random sampling, and cluster sampling. Sampling is recommended for extremely large databases because it can significantly decrease model-training time.

 Data Partition	<p>The Data Partition node enables you to partition data sets into training, test, and validation data sets. The training data set is used for preliminary model fitting. The validation data set is used to monitor and tune the model weights during estimation and is also used for model assessment. The test data set is an additional data set that you can use for model assessment.</p>
 Distribution Explorer	<p>The Distribution Explorer node enables you to explore large volumes of data in multidimensional histograms.</p>
 Multiplot	<p>The Multiplot node enables you to explore large volumes of data graphically</p>
 Association	<p>The Association node enables you to identify association relationships within the data. For example, if a customer buys a loaf of bread, how likely is the customer to buy a gallon of milk as well?</p>
 Clustering	<p>The Clustering node enables you to segment your data; that is, it enables you to identify data observations that are similar in some way.</p>
 Regression	<p>The Regression node enables you to fit both linear and logistic regression models to your data. You can use both continuous and discrete variables as inputs.</p>

Here is a sample of how to combine nodes to create a model:

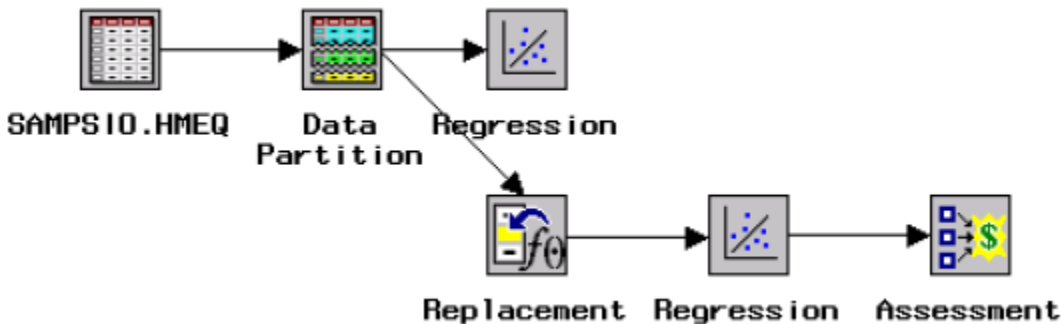


Figure 2: Combining Nodes to Create a Model

Once the nodes are created, all we need to do is right-click on the Assessment node and click Run. As each node finishes, it will turn green.

In Summary, SAS Enterprise Miner, is a simple commercial tool, which will help students play with different aspects of Data Mining: Data Preparation and Investigation, Fitting and Comparing Candidate Models and Generating Reports.

MATLAB:

“MATLAB has excellent built-in support for many data analysis and visualization routines,” [3] in particular, one of its most useful facilities is that of efficient exploratory data analysis , which is a natural fit in the context of data mining.

Advantages of MATLAB: Two major advantages are that of portability and domain specific representations. MATLAB's portability comes from the fact that all MATLAB users will have the same range of basic functions at their disposal. The representation which MATLAB implements, is dealing with all data in the form of matrices. This allows for many varied algorithmic implementations [4], which, as we shall see, is crucial for any data mining package.

Other advantages of MATLAB include its interactive interface, debugging facilities, object oriented nature and in particular its high quality graphics and visualization. facilities [Burton 2006]. Lastly and perhaps most importantly, MATLAB's add on feature, in the form of toolboxes, makes it possible to extend the existing capabilities of the language with ease [Burton 2006]. Data mining, which, for the most part, consists of numerical methodologies [Woolf 2005], is thus a natural fit for implementation with the MATLAB package.

Disadvantages: The main drawback of MATLAB is the fact that it is an interpreted language, which leads to performance cuts, as compared with third generation languages such as C, upon which MATLAB is built. In the context of data mining, this can be a very serious issue, particularly when one is dealing with enormous quantities of data. Dwinnell [3] points out that 4GL's with their own compilers are able to largely overcome this disadvantage. MATLAB does in fact possess its own compiler but it is distributed as a toolbox by The MathWorks (distributors of MATLAB).

One of the central themes of MATLAB is to synthesize many different data mining tools to discover the potential of MATLAB for data mining. Here are a list of MATLAB toolboxes that are used for Data Mining:

Name of the Toolbox	Link:
Classification toolbox	http://stuff.mit.edu/afs/sipb.mit.edu/user/arolfe/matlab/
ARMADA Data Mining Tool	http://www.jamesmalone.plus.com/software.htm
NetLab - Neural Network Toolbox	http://netlab.apponic.com/
Bootstrap Matlab Toolbox	http://www.mathworks.com/help/stats/bootstrp.html
The Spider	http://www.kyb.tuebingen.mpg.de/bs/people/spider/index.html
Bayes Net Toolbox for Matlab	http://www.ai.mit.edu/~murphyk/Software/BNT/bnt.html
SVM and Kernel Methods Matlab Toolbox	http://asi.insa-rouen.fr/~arakotom/toolbox/index

IBM SPSS Modeler [7]:

IBM SPSS Modeler is a data mining software application from IBM. It is a data mining and text analytics workbench used to build predictive models. It has a visual interface which allows users to leverage statistical and data mining algorithms without programming.

IBM SPSS Modeler offers a variety of modeling methods taken from machine learning, artificial intelligence, and statistics. The methods available on the Modeling palette allow you to derive new information from your data and to develop predictive models. Each method has certain strengths and is best suited for particular types of problems.

Modeling methods are divided into three categories:

- Classification
- Association
- Segmentation

Classification Models: *Classification models* use the values of one or more **input** fields to predict the value of one or more output, or **target**, fields. Some examples of these techniques are: decision trees (C&R Tree, QUEST, CHAID and C5.0 algorithms), regression (linear, logistic, generalized linear, and Cox regression algorithms), neural networks, support vector machines, and Bayesian networks.

Classification nodes



The Auto Classifier node creates and compares a number of different models for binary outcomes (yes or no, churn or do not churn, and so on), allowing you to choose the best approach for a given analysis. A number of modeling algorithms are supported, making it possible to select the methods you want to use, the specific options for each, and the criteria for comparing the results. The node generates a set of models based on the specified options and ranks the best candidates according to the criteria you specify. See the topic [Auto Classifier Node](#) for more information.



The Auto Numeric node estimates and compares models for continuous numeric range outcomes using a number of different methods. The node works in the same manner as the Auto Classifier node, allowing you to choose the algorithms to use and to experiment with multiple combinations of options in a single modeling pass. Supported algorithms include neural networks, C&R Tree, CHAID, linear regression, generalized linear regression, and support vector machines (SVM). Models can be compared based on correlation, relative error, or number of variables used. See the topic [Auto Numeric Node](#) for more information.



The Classification and Regression (C&R) Tree node generates a decision tree that allows you to predict or classify future observations. The method uses recursive partitioning to split the training records into segments by minimizing the impurity at each step, where a node in the tree is considered "pure" if 100% of cases in the node fall into a specific category of the target field. Target and input fields can be numeric ranges or categorical (nominal, ordinal, or flags); all splits are binary (only two subgroups). See the topic [C&R Tree Node](#) for more information.



The QUEST node provides a binary classification method for building decision trees, designed to reduce the processing time required for large C&R Tree analyses while also reducing the tendency found in classification tree methods to favor inputs that allow more splits. Input fields can be numeric ranges (continuous), but the target field must be categorical. All splits are binary. See the topic [QUEST Node](#) for more information.

Association Models: *Association models* find patterns in your data where one or more entities (such as events, purchases, or attributes) are associated with one or more other entities. The

models construct rule sets that define these relationships. Here the fields within the data can act as both inputs and targets. You could find these associations manually, but association rule algorithms do so much more quickly, and can explore more complex patterns. Apriori and Carma models are examples of the use of such algorithms. One other type of association model is a sequence detection model, which finds sequential patterns in time-structured data.

Association nodes



The Apriori node extracts a set of rules from the data, pulling out the rules with the highest information content. Apriori offers five different methods of selecting rules and uses a sophisticated indexing scheme to process large data sets efficiently. For large problems, Apriori is generally faster to train; it has no arbitrary limit on the number of rules that can be retained, and it can handle rules with up to 32 preconditions. Apriori requires that input and output fields all be categorical but delivers better performance because it is optimized for this type of data. See the topic [Apriori Node](#) for more information.



The CARMA model extracts a set of rules from the data without requiring you to specify input or target fields. In contrast to Apriori the CARMA node offers build settings for rule support (support for both antecedent and consequent) rather than just antecedent support. This means that the rules generated can be used for a wider variety of applications—for example, to find a list of products or services (antecedents) whose consequent is the item that you want to promote this holiday season. See the topic [CARMA Node](#) for more information.



The Sequence node discovers association rules in sequential or time-oriented data. A sequence is a list of item sets that tends to occur in a predictable order. For example, a customer who purchases a razor and aftershave lotion may purchase shaving cream the next time he shops. The Sequence node is based on the CARMA association rules algorithm, which uses an efficient two-pass method for finding sequences. See the topic [Sequence Node](#) for more information.

Segmentation Models: *Segmentation models* divide the data into segments, or clusters, of records that have similar patterns of input fields. As they are only interested in the input fields, segmentation models have no concept of output or target fields. Examples of segmentation models are Kohonen networks, K-Means clustering, two-step clustering and anomaly detection.

Segmentation nodes



The Auto Cluster node estimates and compares clustering models, which identify groups of records that have similar characteristics. The node works in the same manner as other automated modeling nodes, allowing you to experiment with multiple combinations of options in a single modeling pass. Models can be compared using basic measures with which to attempt to filter and rank the usefulness of the cluster models, and provide a measure based on the importance of particular fields. See the topic [Auto Cluster Node](#) for more information.



The K-Means node clusters the data set into distinct groups (or clusters). The method defines a fixed number of clusters, iteratively assigns records to clusters, and adjusts the cluster centers until further refinement can no longer improve the model. Instead of trying to predict an outcome, *k*-means uses a process known as unsupervised learning to uncover patterns in the set of input fields. See the topic [K-Means Node](#) for more information.



The Kohonen node generates a type of neural network that can be used to cluster the data set into distinct groups. When the network is fully trained, records that are similar should be close together on the output map, while records that are different will be far apart. You can look at the number of observations captured by each unit in the model nugget to identify the strong units. This may give you a sense of the appropriate number of clusters. See the topic [Kohonen Node](#) for more information.

Open Source Tools:

In this section we will discuss 3 tools which are available free for download. The benefit of using Open Source tools is that students can download these tools at home. Here are the 3 most popular tools that we will be discussing in this paper:

- (a) RapidMiner
- (b) R
- (c) WEKA

RapidMiner: RapidMiner [8][9] (formerly Yale) is an environment for machine learning and data mining processes. A modular operator concept allows the design of complex nested operator chains for a huge number of learning problems. The data handling is transparent to the operators. They do not have to cope with the actual data format or different data views - the RapidMiner core takes care of the necessary transformations. Today, RapidMiner is the world-wide leading open-source data mining solution and is widely used by researchers and companies.

Working with RapidMiner fundamentally consists in defining analysis processes by indicating a succession of individual work steps. In RapidMiner these process components are called operators. An operator is defined by several things:

- The description of the expected inputs,
- The description of the supplied outputs,
- The action performed by the operator on the inputs, which ultimately leads to the supply of the outputs,
- A number of parameters which can control the action performed.

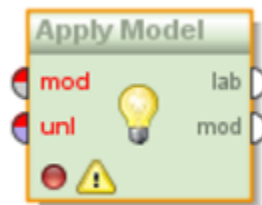


Figure 4: Operator used in RapidMiner

If several operators are interconnected, then we speak of an analysis process or process for short. Such a succession of steps can for example load a data set, transform the data, compute a model and apply the model to another data set.

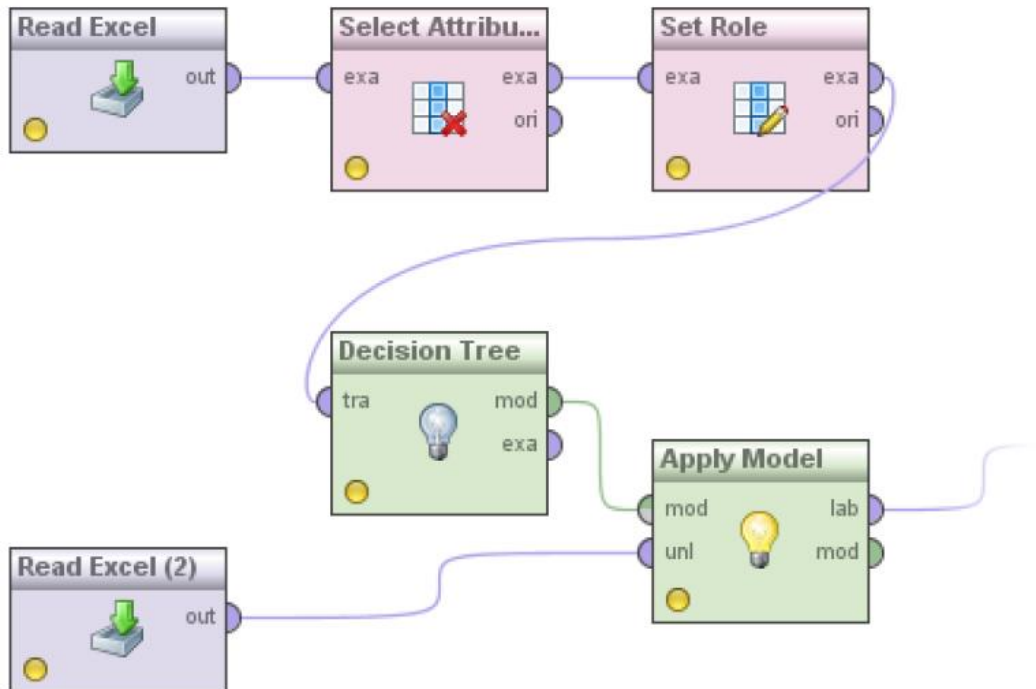


Figure 5: An analysis process consisting of several operators.

Such processes can easily grow to several hundred operators in size in RapidMiner and spread over several levels or subprocesses. The process inspections continually performed in the background as well as the process navigation aids shown below ensure that you do not lose track and that you define correct processes, even for more complex tasks.

RapidMiner is the top voted tool according to KDnuggets. Students would find the animated operators easy to drag and drop, and learn and understand data mining. We would strongly recommend this tool for an introductory undergraduate course.

R:

R [10] is a free software environment for statistical computing and graphics. R can be easily extended with 4,728 packages available on CRAN(as of Sept 6, 2013). Some of the important Data mining tasks (Classification, Clustering and Association Rules) along with the R commands for them are shown below:

Classification with R:

Decision Trees	<i>rparty, party</i>
Random Forest	<i>randomForest, party</i>
SVM	<i>e1071, kernlab</i>
Neural Network	<i>nnet, neuralnet, RSNNs</i>

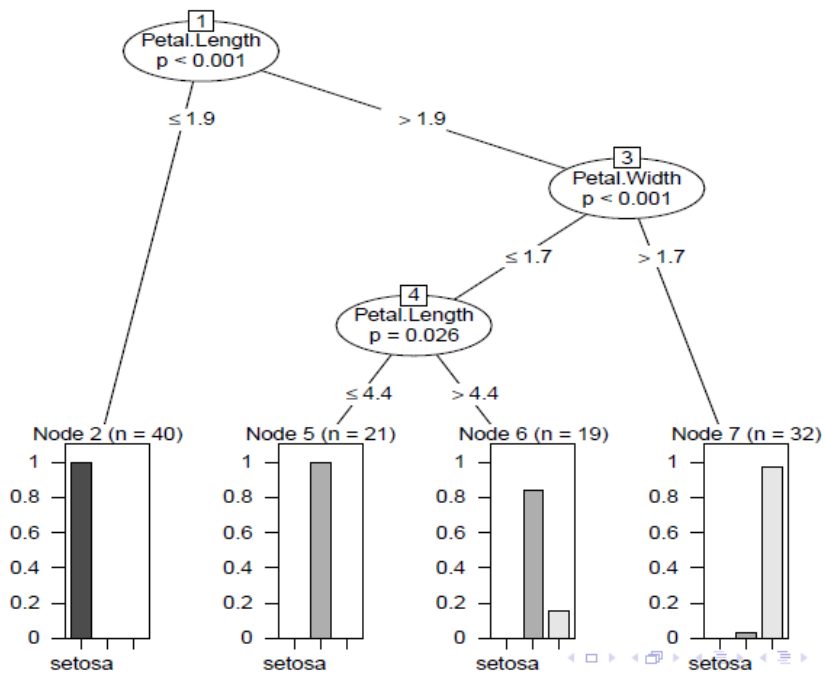
Building a Decision Tree can be very simple using a few commands. Here is a sample of the commands:

```

# build a decision tree
library(party)
iris.formula <- Species ~ Sepal.Length + Sepal.Width +
Petal.Length + Petal.Width
iris.ctree <- ctree(iris.formula, data=iris.train)

plot(iris.ctree)

```



Clustering:

<i>k</i> -means	<i>kmeans()</i> , <i>kmeansruns()</i>
<i>k</i> -medioids	<i>pam()</i> , <i>pamk()</i>
Hierarchical clustering	<i>hcust()</i> , <i>agnes()</i> , <i>diana()</i>
BIRCH	<i>Birch</i>

To do *k*-means clustering on the Iris dataset involves the following simple steps:

```

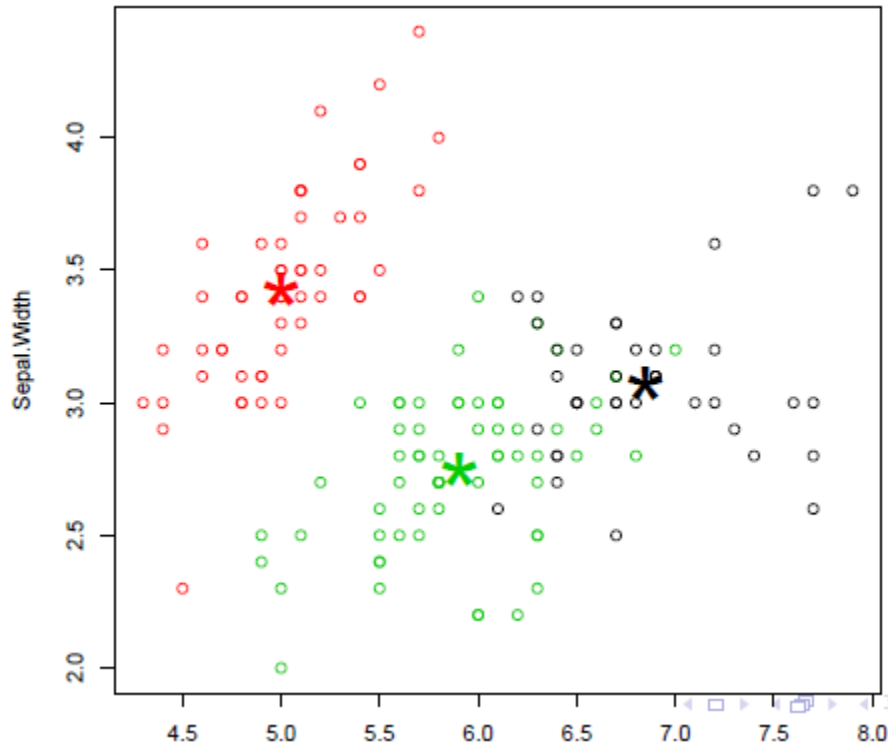
set.seed(8953)
iris2 <- iris
# remove class IDs
iris2$Species <- NULL
# k-means clustering
iris.kmeans <- kmeans(iris2, 3)
# check result
table(iris$Species, iris.kmeans$cluster)
##
## 1 2 3
## setosa 0 50 0
## versicolor 2 0 48
## virginica 36 0 14

```

```

# plot clusters and their centers
plot(iris2[c("Sepal.Length", "Sepal.Width")], col=iris.kmeans$cluster)
points(iris.kmeans$centers[, c("Sepal.Length", "Sepal.Width")],
col=1:3, pch="*", cex=5)

```



Association Rule Mining:

Association Rules	<i>apriori()</i> , <i>eclat()</i> in package <i>arules</i>
Sequential Patterns	<i>arulesSequence</i>
Visualisation of associations	<i>arulesViz</i>

A simple association rule on the titanic dataset can be done as follows:

```

# find association rules with the APRIORI algorithm
library(arules)
rules <- apriori(titanic.raw, control=list(verbose=F),
parameter=list(minlen=2, supp=0.005, conf=0.8),
appearance=list(rhs=c("Survived=No", "Survived=Yes"),
default="lhs"))
# sort rules
quality(rules) <- round(quality(rules), digits=3)
rules.sorted <- sort(rules, by="lift")
# have a look at rules
# inspect(rules.sorted)

# lhs rhs support confidence lift

```

```
# 1 fClass=2nd,  
# Age=Childg => fSurvived=Yesg 0.011 1.000 3.096  
# 2 fClass=2nd,  
# Sex=Female,  
# Age=Childg => fSurvived=Yesg 0.006 1.000 3.096  
# 3 fClass=1st,  
# Sex=Female => fSurvived=Yesg 0.064 0.972 3.010  
# 4 fClass=1st,  
# Sex=Female,  
# Age=Adultg => fSurvived=Yesg 0.064 0.972 3.010  
# 5 fClass=2nd,  
# Sex=Male,  
# Age=Adultg => fSurvived=Nog 0.070 0.917 1.354  
# 6 fClass=2nd,  
# Sex=Femaleg => fSurvived=Yesg 0.042 0.877 2.716  
# 7 fClass=Crew,  
# Sex=Femaleg => fSurvived=Yesg 0.009 0.870 2.692  
# 8 fClass=Crew,  
# Sex=Female,  
# Age=Adultg => fSurvived=Yesg 0.009 0.870 2.692  
# 9 fClass=2nd,  
# Sex=Maleg => fSurvived=Nog 0.070 0.860 1.271
```

In summary, R is a very useful tool for dataminers. It comes with a lot of packages which make it very simple for data mining operations. One of the challenges of teaching R in an introductory course is that it requires a steep learning curve. Students should be encouraged to learn R for an advanced course.

WEKA:

WEKA [11] is the product of the University of Waikato (New Zealand) and was first implemented in its modern form in 1997. It uses the GNU General Public License (GPL). The software is written in the Java™ language and contains a GUI for interacting with data files and producing visual results (think tables and curves). It also has a general API, so you can embed WEKA, like any other library, in your own applications to such things as automated server-side data-mining tasks.

We will briefly look at how you can do the following using WEKA:

- (a) Building a dataset
- (b) Loading the dataset
- (c) Creating the model
- (d) Interpreting the model

To load data into WEKA, we have to put it into a format that will be understood. WEKA's preferred method for loading data is in the Attribute-Relation File Format (ARFF), where you can define the type of data being loaded, then supply the data itself. In the file, you define each column and what each column contains.

Here is an example of building a dataset using WEKA:

In the case of the regression model, you are limited to a NUMERIC or a DATE column. Finally, you supply each row of data in a comma-delimited format. The ARFF file we'll be using with WEKA appears below. Notice in the rows of data that we've left out my house. Since we are creating the model, we cannot input my house into it since the selling price is unknown.

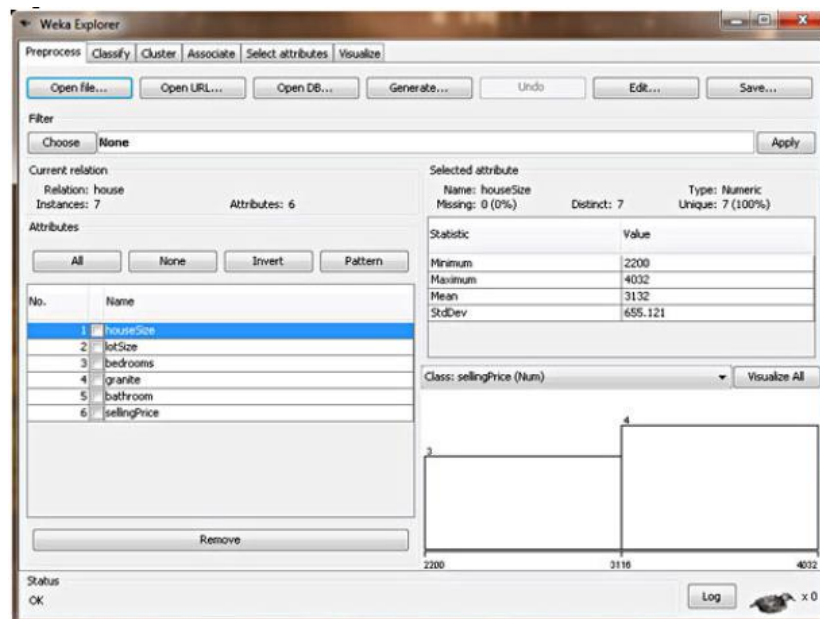
```
@RELATION house

@ATTRIBUTE houseSize NUMERIC
@ATTRIBUTE lotSize NUMERIC
@ATTRIBUTE bedrooms NUMERIC
@ATTRIBUTE granite NUMERIC
@ATTRIBUTE bathroom NUMERIC
@ATTRIBUTE sellingPrice NUMERIC

@DATA
3529,9191,6,0,0,205000
3247,10061,5,1,1,224900
4032,10150,5,0,1,197900
2397,14156,4,1,0,189900
2200,9600,4,0,1,195000
3536,19994,6,1,1,325000
```

(b) Loading the dataset into WEKA:

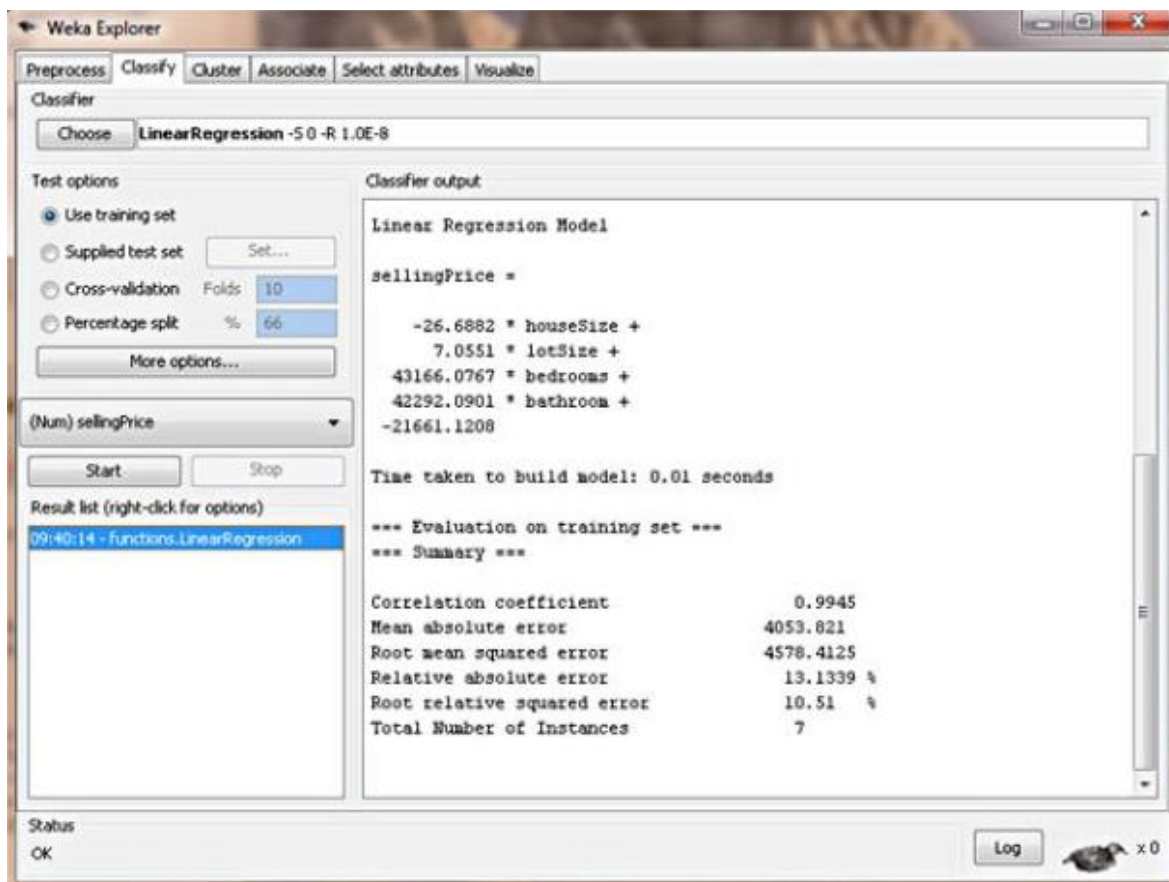
Now that the data file has been created, it's time to create our regression model. Start WEKA, and then choose the **Explorer**. You'll be taken to the Explorer screen, with the **Preprocess** tab selected. Select the **Open File** button and select the ARFF file you created in the section above. After selecting the file, your WEKA Explorer should look similar to the screenshot in Figure below.



(c) Creating a model (regression model): To create the model, click on the **Classify** tab. The first step is to select the model we want to build, so WEKA knows how to work with the data, and how to create the appropriate model:

1. Click the **Choose** button, then expand the **functions** branch.
2. Select the **LinearRegression** leaf.

This tells WEKA that we want to build a regression model. As you can see from the other choices, though, there are lots of possible models to build. Lots! This should give you a good indication of how we are only touching the surface of this subject. Also of note: There is another choice called **SimpleLinearRegression** in the same branch. Do not choose this because simple regression only looks at one variable, and we have six. When you've selected the right model, your WEKA Explorer should look like Figure below:



WEKA doesn't mess around. It puts the regression model right there in the output, as shown in below.


```
sellingPrice = (-26.6882 * houseSize) +  
              (7.0551 * lotSize) +  
              (43166.0767 * bedrooms) +  
              (42292.0901 * bathroom)  
              - 21661.1208
```

In summary, this section discussed the first data-mining model, the regression model (specifically, the linear regression multi-variable model), and showed how to use it in WEKA. This regression model is easy to use and can be used for myriad data sets. In summary, WEKA can be used as an introductory tool for datamining students mainly because of its GUI and its explorer which make is easy for students to explore the different aspects of data mining.

Conclusion:

In this paper, we have presented 3 commercial tools (SAS Enterprise Miner, IBM SPSS Modeler, MATLAB), and 3 open source tools (RapidMiner, R, WEKA), and discussed how to build models using them with pros and cons of each. We feel that using open source tools in a university setting is the best for teaching a basic undergraduate class in one semester. If however, it is an advanced data mining class, the commercial tools can be considered for teaching this course.

References:

- [1] Gartner Press Release. “Gartner Says Big Data Will Drive \$28 Billion of IT Spending in 2012.” October 17, 2012 <http://www.gartner.com/it/page.jsp?id=2200815>
- [2] <http://www.kdnuggets.com/polls/2010/data-mining-analytics-tools.html>
- [3] Murphy, K., *Bayes Net Toolbox for MATLAB*, 2005, Accessed: 10 May 2006, <<http://bnt.sourceforge.net>>.
- [4] Dwinnell, W., *Modeling Methodology 5: Mathematical Programming Languages*, 1998, <<http://will.dwinnell.com/will/willTechnicalPublications.html>>.
- [5] B. King, A.Satyanarayana “Teaching Data Mining in the Era of Big Data”. 2013 ASEE Annual Conference. <http://www.asee.org/public/conferences/20/papers/7580/view>
- [6] IBM SPSS Modeler: www.ibm.com/software/analytics/spss/products/modeler/
- [7] IBM SPSS Modeler Documentation: http://pic.dhe.ibm.com/infocenter/spssmodl/v15r0m0/index.jsp?topic=%2Fcom.ibm.spss.modeler.help%2Fabout_cl ementine_documentation.htm
- [8] Rapid Miner: Installation, Tutorial: <http://rapid-i.com/content/view/36/209/>
- [9] Rapid Miner Manual: http://docs.rapid-i.com/files/rapidminer/rapidminer-5.0-manual-english_v1.0.pdf
- [10] R: <http://www.r-project.org>
- [11] WEKA: Download, Documentation: www.cs.waikato.ac.nz/ml/weka/