Department Of Computer Science

08338
Data Mining and Decision Systems

Weka ACW Workshop

Dr Darryl N. Davis

2015 - 2016

Tutorial: Data Mining and Decision Support using WEKA

Weka Freeware: http://www.cs.waikato.ac.nz/ml/weka/

Module Site for Tutorial:

http://intra.net.dcs.hull.ac.uk/student/modules/08338/weka%20Material/Forms/AllItems.aspx

Aims:

1. Gain practical experience in systematic manipulation of data in Excel and weka through the adoption of a data mining methodology.
2. Understand how to manipulate data using MS Excel using xlsx and csv files
3. Be able to generate input data (CSV) files for weka
4. Gain an understanding of Decision Trees, Classifier, Clustering and Association Rules from practical experience using weka.
5. Be able to use results from weka to make decisions on data

1. Background

In this study pack (from the *DMDS-Tutorial folder* on the module SharePoint site), you will follow a data mining methodology using a variety of data mining techniques within Excel and weka to classify some data on furniture. This weka workshop is a simplified version of the ACW for 08338. It should be possible to complete this Tutorial in the first Two Hour lab but if not you are expected to complete in your own time before the second lab which will be focused on the ACW.

Each item (or *exemplar*) has been encoded as a set of 5 input features and one outcome label (plus one attribute irrelevant for classification), as shown in the file *chairs-workshop.xlsx*. The question asked is, *according to the data given is the item a table or a chair (Target class is Chair)*? The features (with a variety of data types) to be used for classification purposes are:

• Record	Integer	*(reference)*
• Number of legs	Integer	*(input)*
• Backrest	Nominal	*(input)*
• Arms	Nominal	*(input)*
• FlatSurface	Nominal	*(input)*
• Round	Nominal	*(input)*
• Class	Nominal	(*outcome or target*)

The sixth worksheet (*chairs-raw* in *chairs-workshop.xlsx*) contains not only the data defining 42 examples (each record represents an exemplar), but is also annotated with comments. Note each pattern is labeled with a class (*Table* or *Chair*).

The first worksheet in the given data file is *Methodology*. This should be used to describe the data mining and decision steps and all worksheets in your final spreadsheet. This is partially completed. A second worksheet (*Description-Given* in the file *chairs-workshop.xlsx*) contains a partially completed data description table. A third worksheet (*Description-Final)* contains a partially started data description table. The fourth worksheet (*DataCleaning*) will need completing. The fifth worksheet (*Performance*) is for storing classifier performance information. Other worksheets will be created for this workshop.

The process outlined below shows how to move from the raw data to classification of unknown data records using a systematic data mining and decision support methodology. The process is very similar to that which is expected to be used in the ACW for this module.

2. Data Mining Methodology

To ensure consistent and accurate results in data mining you should employ a data mining methodology to guide you through the separate stages. Figure 1 gives an example methodology that is adopted for this module. The following sub-sections take you through this in simple steps. Not the feedback loops – this is not a linear process.

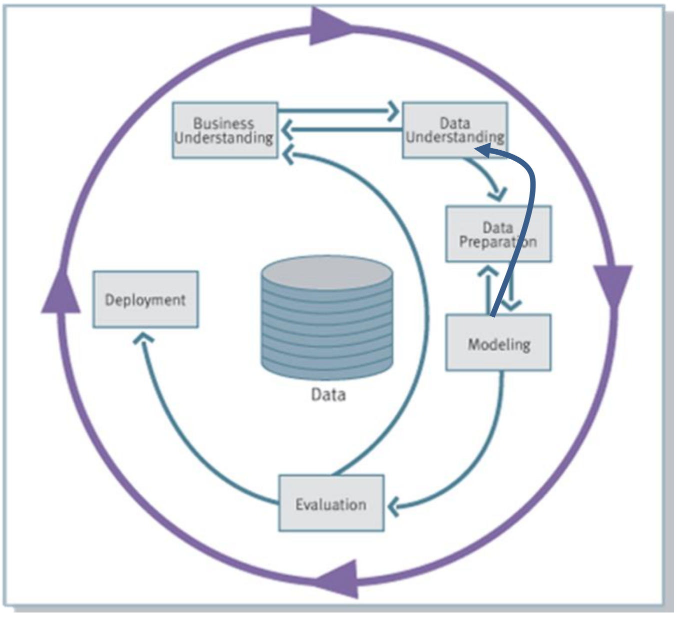


Figure1. A Data Mining Methodology (based on CRISP-DM)

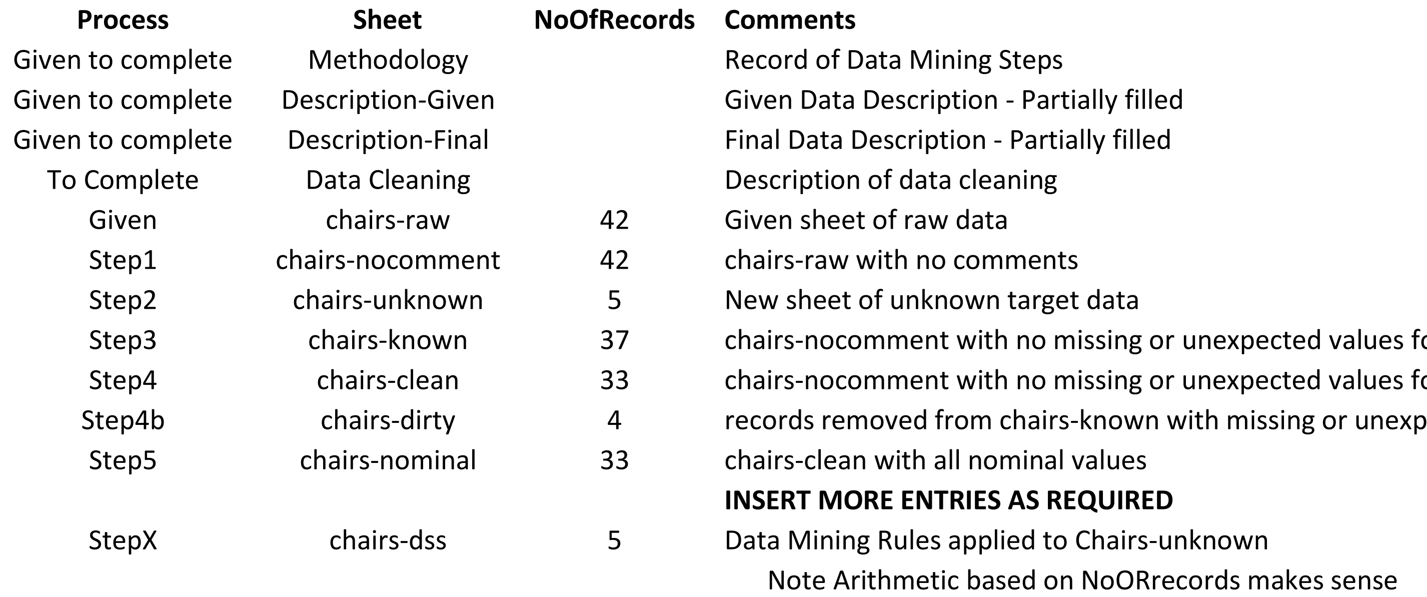
- Business Understanding: Understand what the problem to be solved is
- Data Understanding: Understand and describe the given data
- Data Preparation: The various (documented) steps in preparing the data
- Modeling: Prepare and describe the finalised data for classification
- Evaluation: Performance evaluation of the prepared modelled data
- Deployment: Application of data mining results to further data (DSS)

2.1 Managing the Data Files

Note you should maintain **ONLY ONE** xlsx file throughout the entire process. In manipulating this data you should use multiple worksheets within one Excel file, plus csv files for weka use. A single worksheet in Excel can be saved as a csv file but beware of editing saved files as you may be left editing the csv file and not the Excel xlsx file. All your data and text editing should be in the single Excel (xlsx) file. You can store weka results as text in worksheets in the Excel file. You will also need to build decision tables in the Excel spreadsheet (again as separate worksheets).

The given data file (*chairs-workshop.xlsx*) contains four worksheets. Three data sheets and one methodology note page:

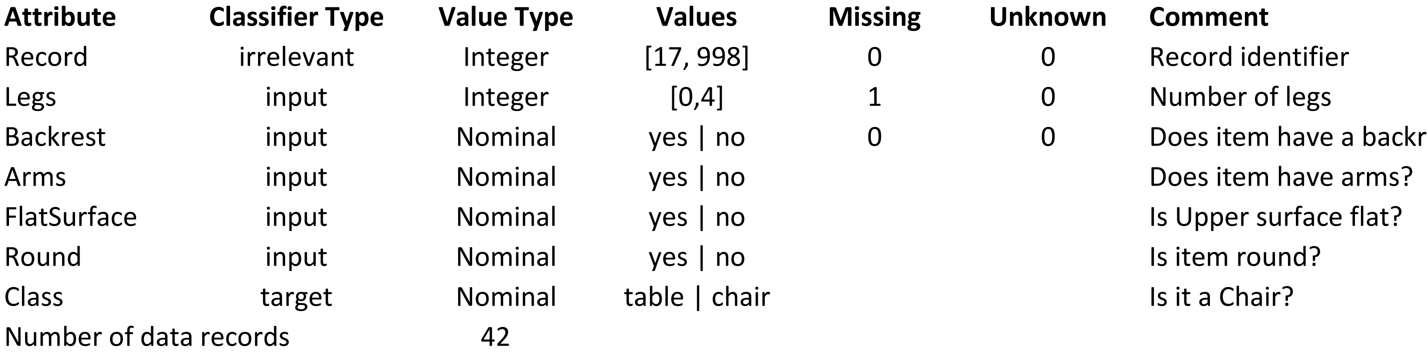
- **Methodology.** This worksheet (as in the ACW) should be used to record data transformations and record what is in ALL the remaining worksheets (see Figure 2).



Process	Sheet	NoOfRecords	Comments
Given to complete	Methodology		Record of Data Mining Steps
Given to complete	Description-Given		Given Data Description - Partially filled
Given to complete	Description-Final		Final Data Description - Partially filled
To Complete	Data Cleaning		Description of data cleaning
Given	chairs-raw	42	Given sheet of raw data
Step1	chairs-nocomment	42	chairs-raw with no comments
Step2	chairs-unknown	5	New sheet of unknown target data
Step3	chairs-known	37	chairs-nocomment with no missing or unexpected values fo
Step4	chairs-clean	33	chairs-nocomment with no missing or unexpected values fo
Step4b	chairs-dirty	4	records removed from chairs-known with missing or unexp
Step5	chairs-nominal	33	chairs-clean with all nominal values
			INSERT MORE ENTRIES AS REQUIRED
StepX	chairs-dss	5	Data Mining Rules applied to Chairs-unknown
			Note Arithmetic based on NoORrecords makes sense

Figure2. Table showing ***Methodology*** worksheet as given (shows many of the steps required)

- **Description-Given**: This worksheet contains a partially completed data description table. You need to complete it (See Figure 3)

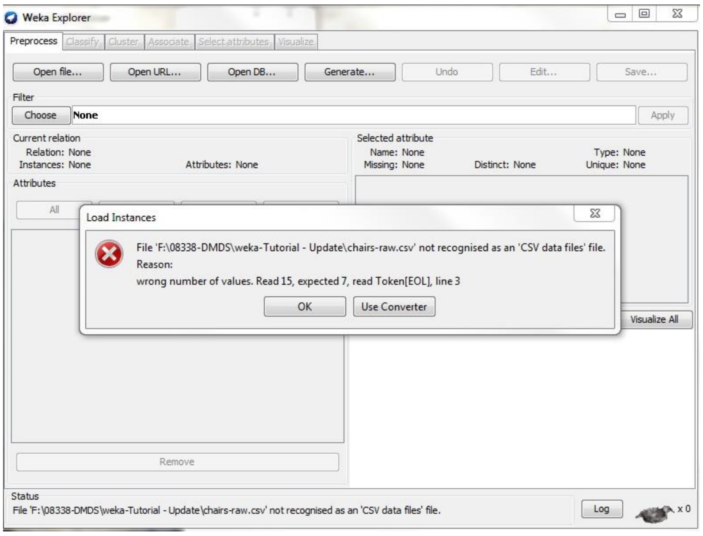


Attribute	Classifier Type	Value Type	Values	Missing	Unknown	Comment
Record	irrelevant	Integer	[17, 998]	0	0	Record identifier
Legs	input	Integer	[0,4]	1	0	Number of legs
Backrest	input	Nominal	yes | no	0	0	Does item have a backr
Arms	input	Nominal	yes | no			Does item have arms?
FlatSurface	input	Nominal	yes | no			Is Upper surface flat?
Round	input	Nominal	yes | no			Is item round?
Class	target	Nominal	table | chair			Is it a Chair?
Number of data records		42				

Figure3. Table showing ***Description-Given*** worksheet as given (partially completed)

- **Performance**: This worksheet contains an uncompleted classifier performance evaluation table. You need to complete it for section 2.7 of this document.
- **chairs-raw**: This worksheet contains not only the data defining 42 examples (each record represents an exemplar), but is also annotated with comments. This worksheet contains the raw training set with many issues to be resolved. The steps to do this are described below and the same as to be used in the ACW.

2.2 Stage 1: Initial Data Cleaning



Filter: None

Current relation
Relation: None
Instances: None
Attributes: None

Selected attribute
Name: None
Type: None
Missing: None
Distinct: None
Unique: None

Attributes

Load Instances

File 'F:\08338-DMDS\weka-Tutorial - Update\chairs-raw.csv' not recognised as an 'CSV data files' file.
Reason:
wrong number of values. Read 15, expected 7, read Token[EOL], line 3

OK | Use Converter

Status
File 'F:\08338-DMDS\weka-Tutorial - Update\chairs-raw.csv' not recognised as an 'CSV data files' file.

Figure4. Weka load error

To understand how particular weka is about data matching column headings perform this little experiment. Open the Excel spreadsheet and save chairs-raw to a CSV file (use the saveas option and the CSV file type and save as chairs-raw.csv). Start weka (see Section 2.3) and load the CSV file. It will fail to load with an error message stating: wrong number of values (see Figure 4). Be aware of this when manipulating data both in this tutorial and the ACW.

To prepare the data follow these steps:

Column Heading to Data Consistency: Copy *chairs-raw* to a new worksheet and rename the new worksheet (e.g. *chairs-nocomment*). Save the xlsx file. Remove all comments (i.e. material to the right of the attribute-value columns) and save file again. You do this not by editing each line but by selecting the column headings. Select column H (mouser left click and drag right until all comments covered by blue highlight). Now use the Delete key. Now with the *chairs-nocomment* worksheet active save as a CSV file (Note you need to use the SaveAs menu and select csv file). Make sure after you navigate the menus that you remain in Excel mode and not CSV (the Excel header panel will tell you this) – **YOU should only use CSV files for loading into weka and never to edit data**

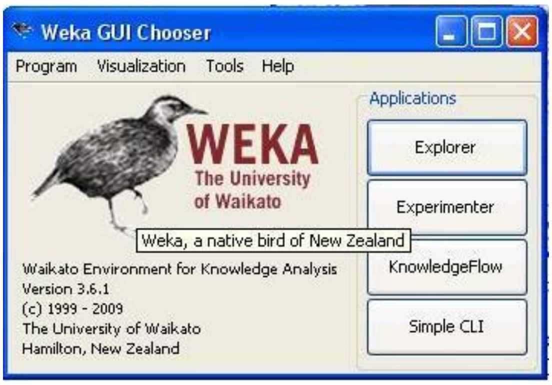
At this point you can start weka (see next section) and visualize the data. If the csv file fails to load into weka, you have made an error in the cleaning process. Go back to the Excel spreadsheet (NOT csv) and clean again, saving both the spreadsheet and the CSV output file. Further data cleaning steps will be required as detailed later.

2.2 Stage 2: Data Understanding (Data Visualisation and Description)

You should complete the data taxonomy table (worksheet **Description** in the Excel spreadsheet) that describes this data in the worksheet (i.e. for the *chairs-nocomment* in the file *chairs-workshop.xlsx*). You can do this from within Excel and using weka. The end result will be a data description table similar to Figure 3, but with all the data cells complete.

2.2.1 Stage 2a Data Understanding - Data Visualisation in weka

WEKA is a data mining tool written in Java. It builds on the theory described in (Witten. Frank & Hall, 2011). Part II of that text describes the package and how to use it. WEKA is available as a download for your own laptop/PC use. Information on the toolkit (and downloads) are at: http://www.cs.waikato.ac.nz/ml/weka/

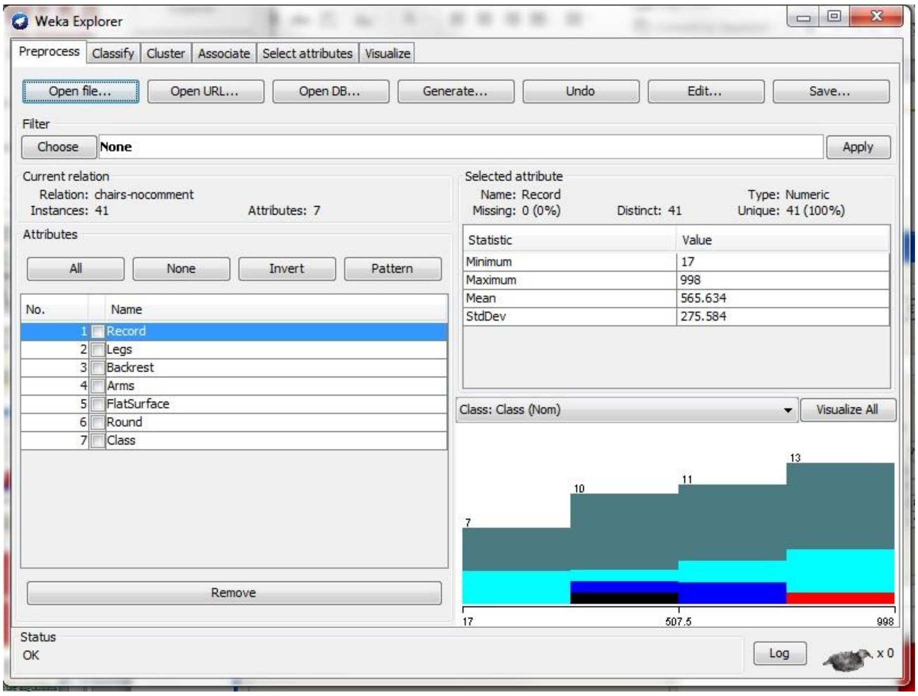


Waikato Environment for Knowledge Analysis
Version 3.6.1
(c) 1999 - 2009
The University of Waikato
Hamilton, New Zealand

Figure 5 Start-Up Window in WEKA

WEKA is a stand-alone piece of software running on Java. Starting it should result in the window shown below (Figure5). Select "Explorer" from this menu.

From Explorer CSV data files can be loaded, visualized and classifiers used. The figure below (Figure 4) shows the WEKA Explorer Window, with Chairs-clean.csv data file loaded. Select "***Open File***" to load your data. CSV format for data files are suggested.



Filter: None

Current relation
Relation: chairs-nocomment
Instances: 41 Attributes: 7

Selected attribute
Name: Record Type: Numeric
Missing: 0 (0%) Distinct: 41 Unique: 41 (100%)

Statistic	Value
Minimum	17
Maximum	998
Mean	565.634
StdDev	275.584

Attributes

No.	Name
1	□ Record
2	□ Legs
3	□ Backrest
4	□ Arms
5	□ FlatSurface
6	□ Round
7	□ Class

Class: Class (Nom)

Status
OK

Figure6 WEKA Explorer Window

Note that by selecting different attributes (click on the attribute name) a data summary for that attribute is shown to the right (Figure 6). This is useful for building data description tables. Selecting "***Visualise All***" enables you to see how the data is distributed across the output classes (The menu to the left of the "***Visualise All***" button selects the output attribute).

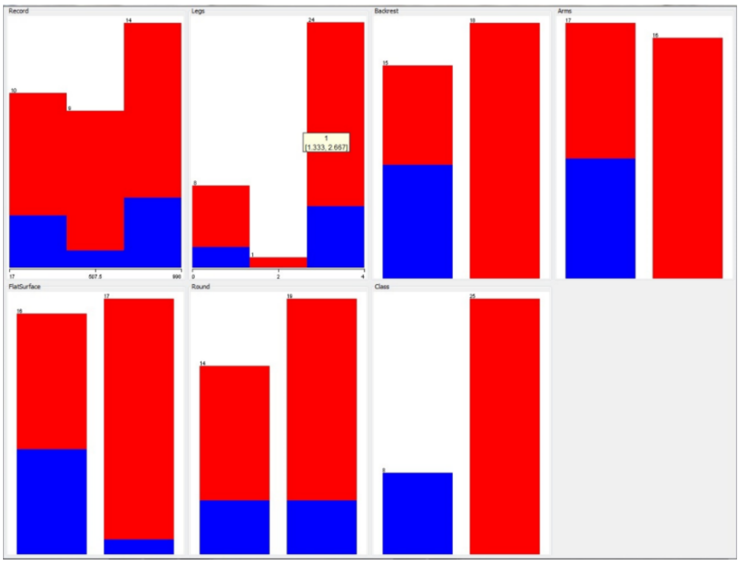
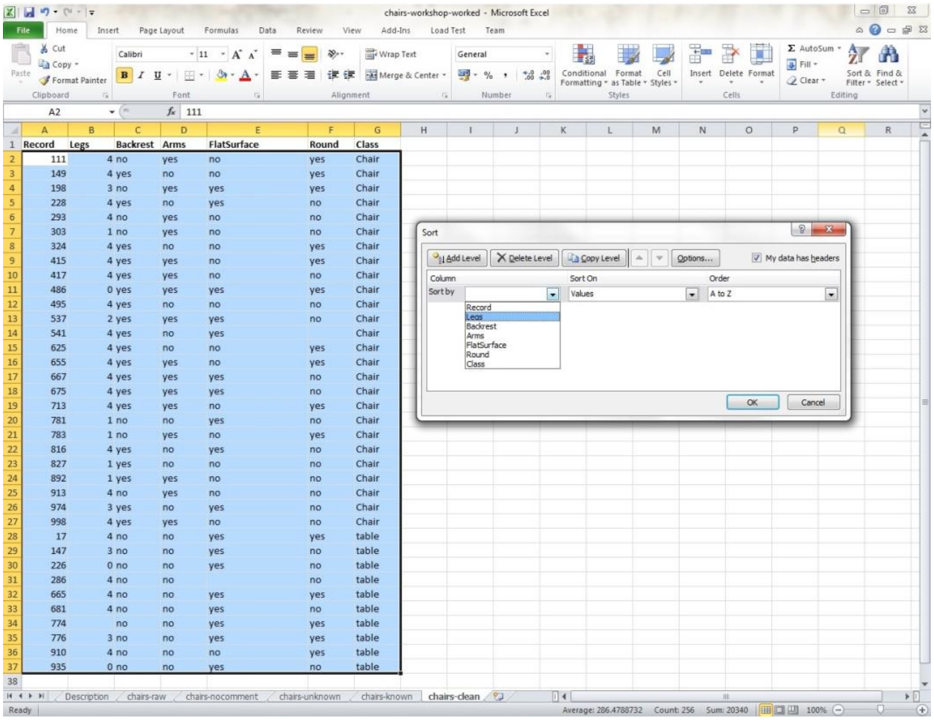


Figure7 Example WEKA Visualisation Window

Figure 7 depicts the resulting window. This window is colour-coded and interactive. Moving the mouse cursor enables you to see values for the different attributes. It provides an easy way to check that your data and its values are as expected (and noted in the *Description* worksheet). You should note a number missing or Unknown values for some attributes.

2.3 Stage 3: Data Preparation - Data Value Removal (No Replacement here)

We can use Excel to deal with missing and unknown values. Copy *chairs-nocomment* to a new worksheet (name it *chairs-known*). Select all the data and "Custom sort" on Class in a similar way as shown in Figure 8 (make sure you click on "My data has headers".



A2 *fx* 111

	A	B	C	D	E	F	G
1	Record	Legs	Backrest	Arms	FlatSurface	Round	Class
2	111	4	no	yes	no	yes	Chair
3	149	4	yes	no	no	yes	Chair
4	198	3	no	yes	yes	yes	Chair
5	228	4	yes	no	yes	no	Chair
6	293	4	no	yes	no	no	Chair
7	303	1	no	yes	no	no	Chair
8	324	4	yes	no	no	yes	Chair
9	415	4	yes	yes	no	yes	Chair
10	417	4	yes	yes	no	no	Chair
11	486	0	yes	yes	yes	yes	Chair
12	495	4	yes	no	no	no	Chair
13	537	2	yes	yes	yes	no	Chair
14	541	4	yes	no	yes		Chair
15	625	4	yes	no	no	yes	Chair
16	655	4	yes	yes	no	yes	Chair
17	667	4	yes	yes	yes	no	Chair
18	675	4	yes	yes	yes	no	Chair
19	713	4	yes	yes	no	yes	Chair
20	781	1	no	no	yes	no	Chair
21	783	1	no	yes	no	yes	Chair
22	816	4	yes	no	yes	no	Chair
23	827	1	yes	no	no	no	Chair
24	892	1	yes	yes	no	no	Chair
25	913	4	no	yes	no	no	Chair
26	974	3	yes	no	yes	no	Chair
27	998	4	yes	yes	no	no	Chair
28	17	4	no	no	yes	yes	table
29	147	3	no	no	yes	no	table
30	226	0	no	no	yes	no	table
31	286	4	no	no		no	table
32	665	4	no	no	yes	yes	table
33	681	4	no	no	yes	no	table
34	774		no	no	yes	yes	table
35	776	3	no	no	yes	yes	table
36	910	4	no	no	no	yes	table
37	935	0	no	no	yes	no	table
38							

Sort

☑ My data has headers

Column	Sort On	Order
Sort by	Values	A to Z

Record
Legs
Backrest
Arms
FlatSurface
Round
Class

Figure8 Example Custom Sort in Excel

The data records will be sorted in alphabetical order (A-Z) on the Class values. Scroll down to the base of the data and you will see four records with unknown and one with missing values. Delete these records entirely and save the Excel File.

Next Copy *chairs-nocomment* to a new worksheet (name it *chairs-unknown*). Select all the data and "Custom sort" on Class in a similar way as described above. Scroll down to the base of the data and you will see four records with unknown and one with missing values. Delete all the records bar these records and save the Excel File.

There are further missing missing values in the data. For this create a new worksheet (*chairs-dirty*) and copy the header line to the top. Copy *chairs-known* to *chairs-clean* and use this worksheet as follows. Use the Select all data and Custom Sort (A-Z) multiple times; once for each attribute. This time instead of deleting the record with any missing, null or unknown value cut it from *chairs-clean* and paste into *chairs-dirty*.

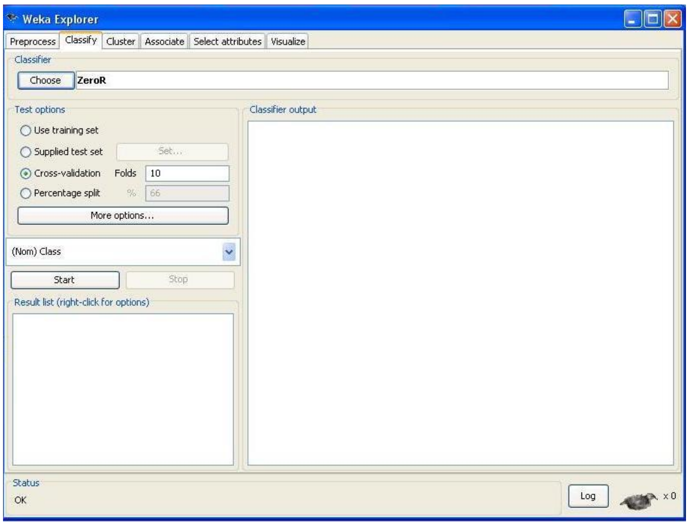
The final part of cleaning is dealing with duplicated records. I have used Conditional Formatting to highlight cells in the Record attribute that have the same value. There should be one such record. Cut this record from chairs-clean and paste into *chairs-dirty*. You never completely delete any data in the Data Warehouse; merely place it to some location (e.g. worksheet) as record of a process.

Complete the Data Cleaning worksheet as a record of these data cleaning steps. You can now start to complete the worksheet that describes the "clean" final data (i.e. *Description-Final*).

Once complete save the Excel file and document your edits in Methodology (I have already done this but check your data files agree) and in Description (this will need completing). Save the *chairs-clean* as a CSV file of that name. If you have done this correctly you are now ready to do the data mining in weka.

2.4 Stage 4: Modeling: Data Preparation for Classifiers

At this point, you can use some of the classifiers in weka. Others will require data transformations (as explained in subsequent sections). Load *chairs-clean.csv* into weka. Remove attribute "Record" within weka – it is not needed within the classifiers.



ZeroR

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) Class

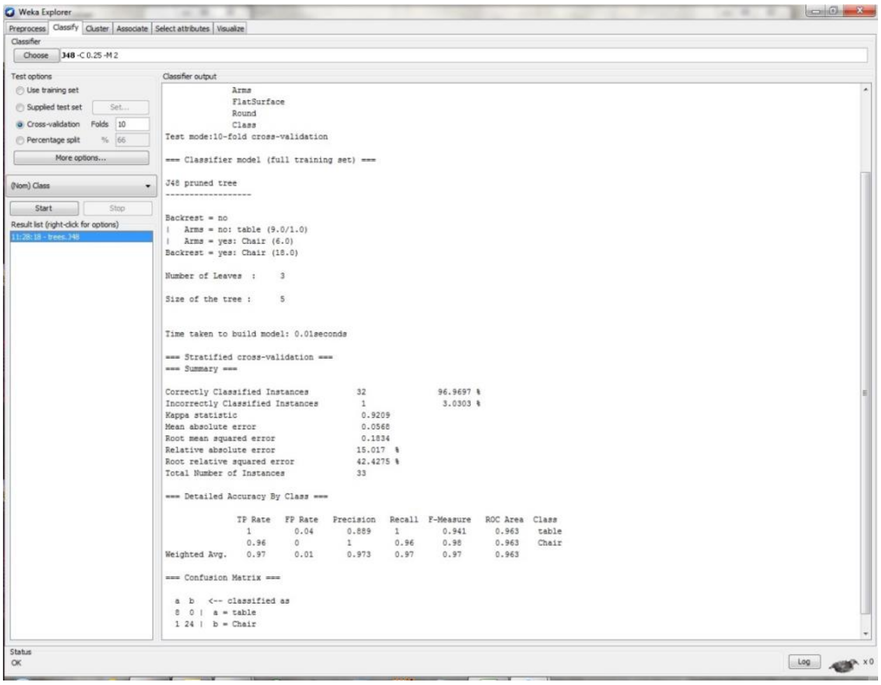
Status
OK

Figure9 WEKA Classify Window (before choosing any classifier)

Selecting "***Classify***" menu on the top Function Button sequence allows the (supervised learning) classifier window to be accessed. The "***Choose***" button (in figure 9) allows you to select between classifiers from a hierarchical menu (***Bayes***, ***Function***, ***Lazy***, ***Meta***, ***mi***, ***misc***, ***Rules, Trees***). The classifiers are organized to type. Hence ID3 and J48 (Decision Trees) are to be found under the

Trees Selection. Neural Nets (e.g. MLP, RBF) are within the ***Function*** selection. The WEKA support documentation describes this tool in more detail.

Figure 10 shows how J48 has been used to generate a Decision Tree. Select all the text in the scrollable window and save to a new worksheet in the Excel file. Use easily identifiable worksheet names as you will be saving many weka outputs (e.g. *chairs-clean-j48*)



J48 -C 0.25 -M 2

● Cross-validation Folds 10

○ Percentage split % 66

(Nom) Class

Result list (right-click for options)
11:28:18 - trees.J48

```
              Arms
              FlatSurface
              Round
              Class
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
------------------

Backrest = no
|   Arms = no: table (9.0/1.0)
|   Arms = yes: Chair (6.0)
Backrest = yes: Chair (18.0)

Number of Leaves  :     3

Size of the tree :      5


Time taken to build model: 0.01seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          32               96.9697 %
Incorrectly Classified Instances         1                3.0303 %
Kappa statistic                          0.9209
Mean absolute error                      0.0568
Root mean squared error                  0.1834
Relative absolute error                 15.017  %
Root relative squared error             42.4275 %
Total Number of Instances               33

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0.04      0.889     1         0.941      0.963    table
                 0.96      0         1         0.96      0.98       0.963    Chair
Weighted Avg.    0.97      0.01      0.973     0.97      0.97       0.963

=== Confusion Matrix ===

  a  b   <-- classified as
  8  0 |  a = table
  1 24 |  b = Chair
```

Status
OK

Figure10 WEKA Classify Window after using j48 tree

Note that by clicking on the command line you can change the classifier parameters. You can change the use of data by changing the options on the left of the screen. Here 10-fold data folding is used. You can visualize the tree (Figure 11) by clicking on the appropriate J48 menu after right clicking the blue highlighted j48 selection.

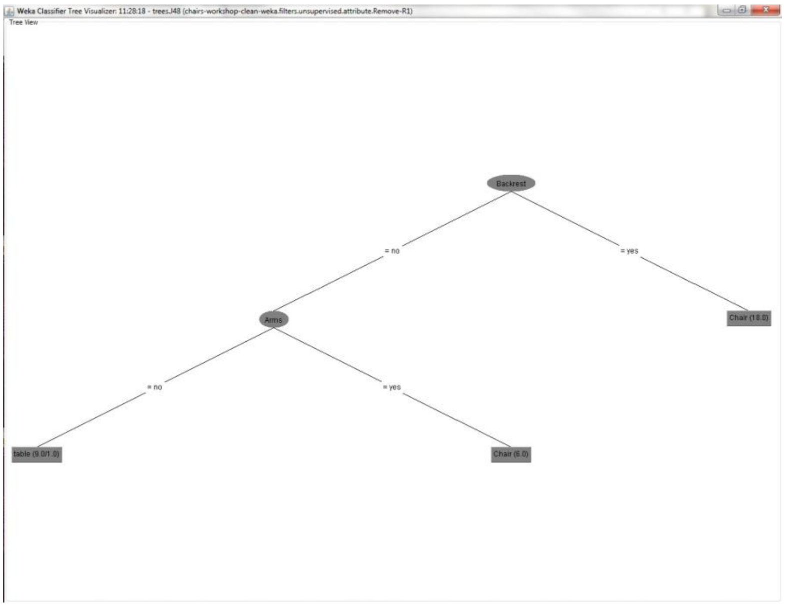
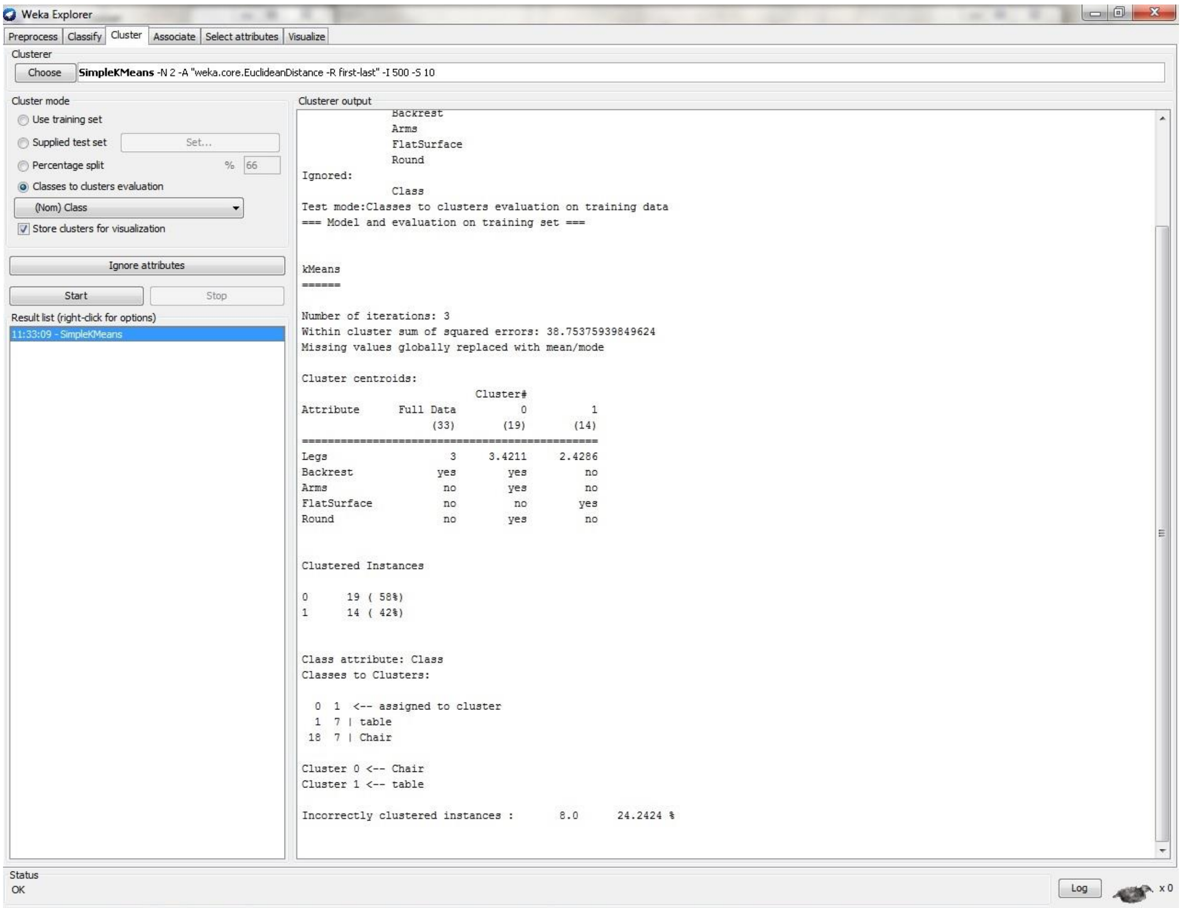

Weka Classifier Tree Visualizer: 11:28:18 - trees.J48 (chairs-workshop-clean-weka.filters.unsupervised.attribute.Remove-R1)
Tree View
Backrest
= no
= yes
Arms
Chair (18.0)
= no
= yes
table (9.0/1.0)
Chair (6.0)

Figure11 WEKA visualization Window for a j48 tree

It is recommended that you use at least three other classifiers on the data. Four recommended classifiers to try are **Trees-J48**, **Functions-SMO**, **Rules-Ridor** and **Bayes-NaiveBayes** (other good classifiers include Rules-PART, Rules-NNge). In each case save the output as text in an appropriately named worksheet in the Excel file.

2.5 Modeling: Stage 4b: Data Preparation for Clustering

You can use the data as loaded for use with J48 with the clustering algorithms in weka. Simply switch to the clustering menu and select a clustering algorithm (for example, **SimpleKmeans**). Pressing start will run this algorithm without using the Outcome labels to form clusters. To see how it performs on the given outcome labels, select Classes to cluster evaluation and run again (see Figure 12). This will produce a confusion matrix. This tells us that the Data Model is not perfect and 100% agreement with the Class labels is not found.



Clusterer: SimpleKMeans -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10

Cluster mode:
○ Use training set
○ Supplied test set
○ Percentage split % 66
● Classes to clusters evaluation
(Nom) Class
☑ Store clusters for visualization

Result list: 11:33:09 - SimpleKMeans

```
              Backrest
              Arms
              FlatSurface
              Round
Ignored:
              Class
Test mode:Classes to clusters evaluation on training data
=== Model and evaluation on training set ===


kMeans
======

Number of iterations: 3
Within cluster sum of squared errors: 38.75375939849624
Missing values globally replaced with mean/mode

Cluster centroids:
                          Cluster#
Attribute     Full Data          0          1
                   (33)       (19)       (14)
==============================================
Legs                  3     3.4211     2.4286
Backrest            yes        yes         no
Arms                 no        yes         no
FlatSurface          no         no        yes
Round                no        yes         no



Clustered Instances

0      19 ( 58%)
1      14 ( 42%)


Class attribute: Class
Classes to Clusters:

  0  1  <-- assigned to cluster
  1  7 | table
 18  7 | Chair

Cluster 0 <-- Chair
Cluster 1 <-- table

Incorrectly clustered instances :       8.0      24.2424 %
```

Status: OK

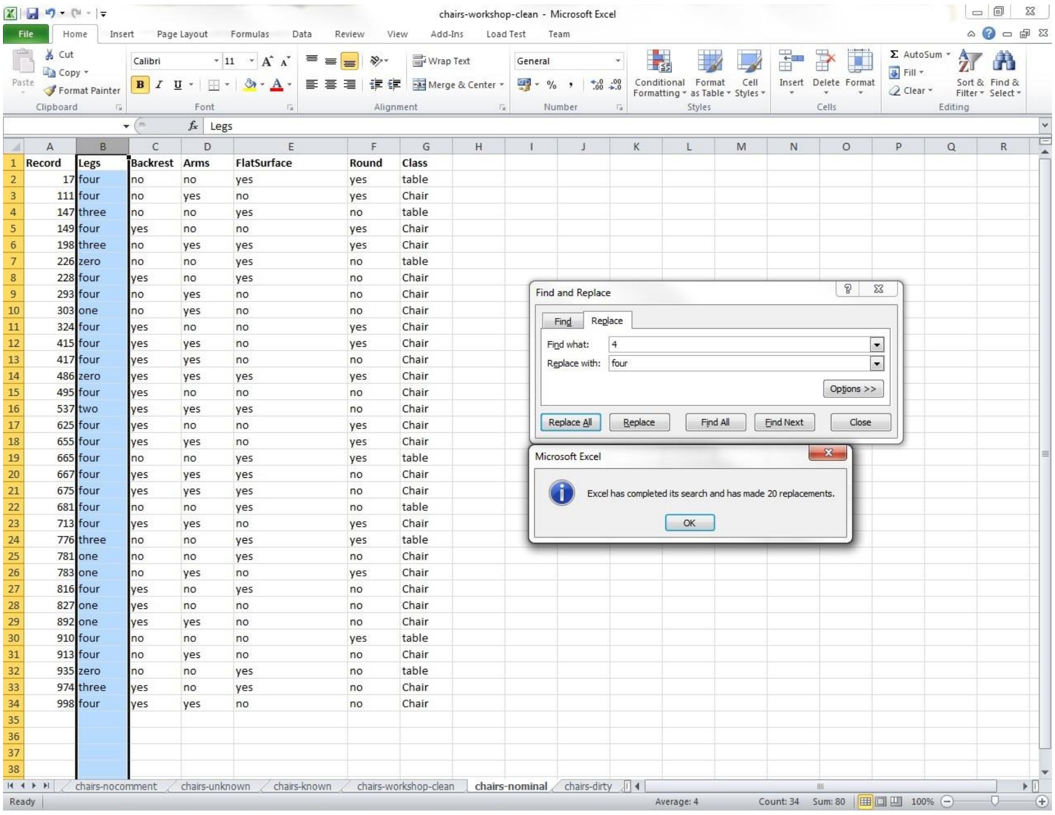
Figure12 WEKA Classify Window (after using a SimpleKmeans)

2.6 Modeling: Stage 4c: Data Preparation for Association Rules

If you now switch to the Associate Menu, and select **Tertius** (the only recommended AR generator in weka) and run, you will see no results. This is because you need to transform the data to be nominal. To do this, go back to the Excel spreadsheet. Copy the worksheet chairs-clean and rename the new worksheet (e.g. *chairs-nominal*). Transform all the numeric values to text (e.g. 0 to zero, 1 to one etc.). By selecting each column in turn using ReplaceAll (see figure 13) this is easily done. Now save the spreadsheet and save worksheet as a csv file. Be careful with these edits as replacing 1 with One will not only make a 1 a One but will change every 15 (for example) into a One5

Load this new csv file into weka and remove the Record attribute. Now select **Associate-Tertius** again. You need to select the command-line options and change some parameters (see Figure 14).

Change ***classIndex*** from 0 to -1 and ***classification*** from False to True. You can reduce the number of rules generated by changing the 10 in confirmationValues from 10 to 5. It should now generate five association rules with Class as the conclusions. Save the output (see Figure 15).



chairs-workshop-clean - Microsoft Excel

Record	Legs	Backrest	Arms	FlatSurface	Round	Class
17	four	no	no	yes	yes	table
111	four	no	yes	no	yes	Chair
147	three	no	no	yes	no	table
149	four	yes	no	no	yes	Chair
198	three	no	yes	yes	yes	Chair
226	zero	no	no	yes	no	table
228	four	yes	no	yes	no	Chair
293	four	no	yes	no	no	Chair
303	one	no	yes	no	no	Chair
324	four	yes	no	no	yes	Chair
415	four	yes	yes	no	yes	Chair
417	four	yes	yes	no	no	Chair
486	zero	yes	yes	yes	yes	Chair
495	four	yes	no	no	no	Chair
537	two	yes	yes	yes	no	Chair
625	four	yes	no	no	yes	Chair
655	four	yes	yes	no	yes	Chair
665	four	no	no	yes	yes	table
667	four	yes	yes	yes	no	Chair
675	four	yes	yes	yes	no	Chair
681	four	no	no	yes	no	table
713	four	yes	yes	no	yes	Chair
776	three	no	no	yes	yes	table
781	one	no	no	yes	no	Chair
783	one	no	yes	no	yes	Chair
816	four	yes	no	yes	no	Chair
827	one	yes	no	no	no	Chair
892	one	yes	yes	no	no	Chair
910	four	no	no	no	yes	table
913	four	no	yes	no	no	Chair
935	zero	no	no	yes	no	table
974	three	yes	no	yes	no	Chair
998	four	yes	yes	no	no	Chair

Find and Replace

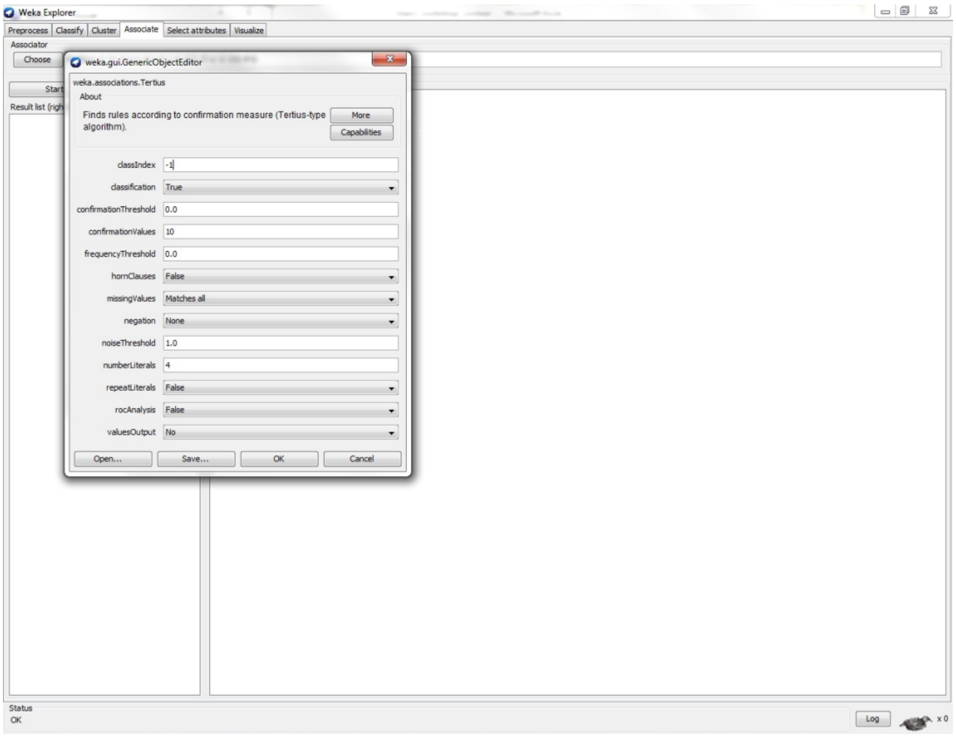
Find what: 4

Replace with: four

Microsoft Excel

Excel has completed its search and has made 20 replacements.

Figure13 Using ReplaceAll in Excel



weka.gui.GenericObjectEditor

weka.associations.Tertius

About

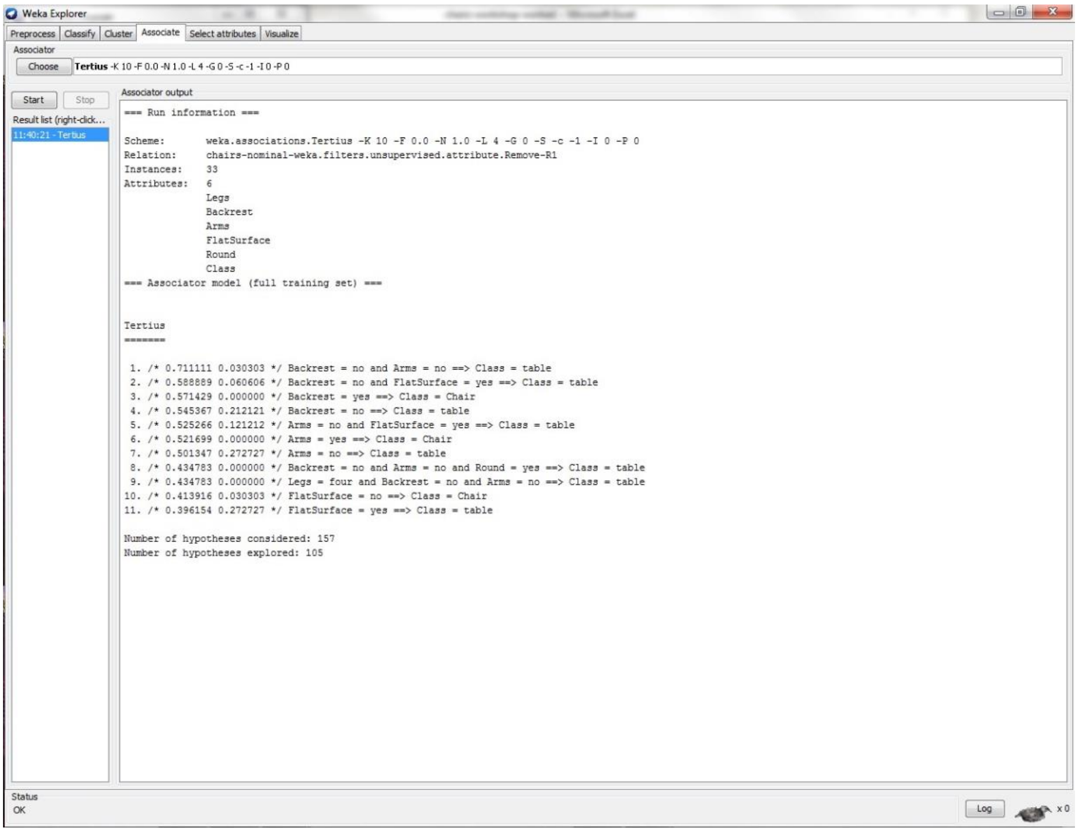
Finds rules according to confirmation measure (Tertius-type algorithm).

classIndex: -1
classification: True
confirmationThreshold: 0.0
confirmationValues: 10
frequencyThreshold: 0.0
hornClauses: False
missingValues: Matches all
negation: None
noiseThreshold: 1.0
numberLiterals: 4
repeatLiterals: False
rocAnalysis: False
valuesOutput: No

Status

OK

Figure14 Changing Tertius parameters in weka



Tertius -K 10 -F 0.0 -N 1.0 -L 4 -G 0 -S -c -1 -I 0 -P 0

```
=== Run information ===

Scheme:       weka.associations.Tertius -K 10 -F 0.0 -N 1.0 -L 4 -G 0 -S -c -1 -I 0 -P 0
Relation:     chairs-nominal-weka.filters.unsupervised.attribute.Remove-R1
Instances:    33
Attributes:   6
              Legs
              Backrest
              Arms
              FlatSurface
              Round
              Class
=== Associator model (full training set) ===


Tertius
=======

 1. /* 0.711111 0.030303 */ Backrest = no and Arms = no ==> Class = table
 2. /* 0.588889 0.060606 */ Backrest = no and FlatSurface = yes ==> Class = table
 3. /* 0.571429 0.000000 */ Backrest = yes ==> Class = Chair
 4. /* 0.545367 0.212121 */ Backrest = no ==> Class = table
 5. /* 0.525266 0.121212 */ Arms = no and FlatSurface = yes ==> Class = table
 6. /* 0.521699 0.000000 */ Arms = yes ==> Class = Chair
 7. /* 0.501347 0.272727 */ Arms = no ==> Class = table
 8. /* 0.434783 0.000000 */ Backrest = no and Arms = no and Round = yes ==> Class = table
 9. /* 0.434783 0.000000 */ Legs = four and Backrest = no and Arms = no ==> Class = table
10. /* 0.413916 0.030303 */ FlatSurface = no ==> Class = Chair
11. /* 0.396154 0.272727 */ FlatSurface = yes ==> Class = table

Number of hypotheses considered: 157
Number of hypotheses explored: 105
```

Figure 15 WEKA Classify Window (after using Tertius Association Rule generator)

2.7 Evaluation Stage 5: Determining Performance from Classifier Output

The third worksheet in the given spreadsheet (Performance) can be easily completed for all the Classifiers you use. Note Tertius does not give Performance Data suitable for this. It is possible to generate this yourself but it is suggested to leave that for advanced users in the ACW. The Performance table is empty bar the headers and some formula (Figure 16).

Classifier	Data	RMSE	Accuracy	TP	FP	TN	FN	Sensitivity	Specificity
j48	chairs-clean							#DIV/0!	#DIV/0!
Ridor	chairs-clean							#DIV/0!	#DIV/0!
K-Means	chairs-clean							#DIV/0!	#DIV/0!

Figure 16 Incomplete Performance Spreadsheet

Note that it requires both classifier and data file name to be entered as the same classifier will perform differently on different data; and different classifiers will perform differently on the same data. Classifier performance is dealt in detail in the Classifiers lecture. For this tutorial/workshop you can simply complete this table to gain an initial insight into Classifier performance. The following bullet points cover the Columns. Note Chairs is the Target class.

- **RMSE (Root Mean Square Error).** Weka lists the RMSE so copy that figure into the appropriate cell – typically it is a number from 0 to 1.0 - the bigger the better.
- **Accuracy.** Weka lists this as “Correctly Classified Instances” – typically a percentage.
- **TP (True Positives).** This is from the confusion matrix - Chairs classified as Chairs
- **FP (False Positives).** This is from the confusion matrix - Tables classified as Chairs
- **TN (True Negatives).** This is from the confusion matrix - Tables classified as Tables

- **FN (False Negatives).** This is from the confusion matrix - Chairs classified as Tables
- **Sensitivity**. This details what the classifier does with Chairs: Formula = TP/(TP+FN)
- **Specificity**. This details what the classifier does with Tables: Formula = TN/(TN+FP)

2.8 Deployment Stage 6: Classifying Unseen Data using Classifier Output

You can now close weka, and use the output from J48, Ridor, and the Association Rules to classify the data in the **chairs-unknown** spreadsheet. Copy *chairs-unknown* to chairs-dss. Make fresh columns for Outcomes (one per rule used) and name them according to the classifier used (you can use more than one). Figure 17 shows a completed Decision Support table. Here the process is described for j48, Ridor, and Tertius. We will not be using K-Means to classify here. **Note in all cases when applying rules if the rule does not match an example you cannot say anything for that rule on the given example**.

Using j48 to make Decisions

To the right of Class attribute add a new Header j48 – this will be used to store j48 decisions. Below the data, paste in the j48 decision tree from chairs-clean-j48. It should look similar to this:

```
j48 clean
Backrest = no
| Arms = no: table (9.0/1.0)
|   Arms = yes: Chair (6.0)
Backrest = yes: Chair (18.0)
```

This given tree is to be read as follows:

IF (Backrest = no) AND (Arms = no) THEN Prediction is table

IF (Backrest = no) AND (Arms = yes) THEN Prediction is Chair

IF (Backrest = yes) THEN Prediction is Chair

For each pattern, follow the decision tree path to generate a prediction for the pattern. Store the prediction in the spreadsheet (in the new j48 column).

Using Ridor to make Decisions

Ridor works differently. It produces a Default Rule with a number of rules that detail exceptions. The Default rule typically is the Target value with the greatest frequency. For this data that will be Chairs (in the clean known data, there are 26 chair examples and 10 Table examples). Ridor output looks similar to this:

```
Ridor-clean
Class = Chair  (33.0/8.0)
    Except (Backrest = no) and (Arms = no) and (Legs > 2) => Class = table  (5.0/0.0) [1.0/0.0]
```

This equates to TWO rules. Make new columns with Headers (Ridor-0 for default rule), then Ridor-1 (for next rule) etc. All Unknowns are predicted as Chair by Ridor-0, but only those matching the conditions of Ridor-1 (etc.) can be tables. Exceptions over-ride Default (Ridor-0) classifications in Ridor.

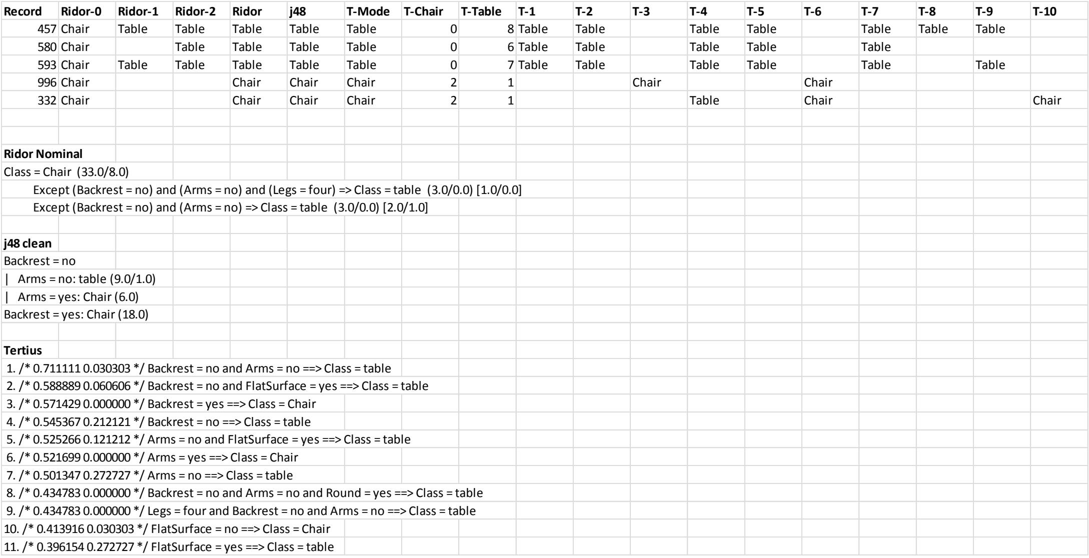
Using Tertius to make Decisions

Tertius produces a set of rules – you can control the number as given in section 2.6. You will need 3 columns (T-mode, T-Chair and T-table) plus one column per Tertius rule. First complete the application of each rule for each record. Figure 17 shows the use of 11 rules but you need only generate 5 rules using Tertius. Tertius rules are easy to understand:

1. /* 0.711111 0.030303 */ Backrest = no and Arms = no ==> Class = table
2. /* 0.588889 0.060606 */ Backrest = no and FlatSurface = yes ==> Class = table
3. /* 0.571429 0.000000 */ Backrest = yes ==> Class = Chair
4. /* 0.545367 0.212121 */ Backrest = no ==> Class = table
5. /* 0.525266 0.121212 */ Arms = no and FlatSurface = yes ==> Class = table

For the time being you can ignore the information metrics (the numbers) given at the start of each rule. Once you have completed the application of all Tertius Rules to all Unknown examples, you need to use Formula in Excel to count the Number of Chairs and the number of tables. You should become proficient in this over the module so take time to understand the use of Formula.

- Select the first cell under T-Chair and insert =COUNTIF(P2:Z2,"Chair"), select the P2:Z2 in the formula and select the cells in that row that cover all the Tertius rules. The resulting number should agree with a manual count
- Select the first cell under T-Table and insert =COUNTIF(P2:Z2,"Table"). Do the same as above. Again the Count should correspond to a manual count of Table along that row.
- Now select both these cells, copy and paste the formula to the corresponding cells for all the other examples.
- Finally enter Table or Chair in T-Mode according to the information given by these formulas.



Record	Ridor-0	Ridor-1	Ridor-2	Ridor	j48	T-Mode	T-Chair	T-Table	T-1	T-2	T-3	T-4	T-5	T-6	T-7	T-8	T-9	T-10
457	Chair	Table	Table	Table	Table	Table	0	8	Table	Table		Table	Table		Table	Table	Table	
580	Chair		Table	Table	Table	Table	0	6	Table	Table		Table	Table		Table			
593	Chair	Table	Table	Table	Table	Table	0	7	Table	Table		Table	Table		Table		Table	
996	Chair			Chair	Chair	Chair	2	1			Chair			Chair				
332	Chair			Chair	Chair	Chair	2	1				Table		Chair				Chair

Ridor Nominal

Class = Chair (33.0/8.0)
Except (Backrest = no) and (Arms = no) and (Legs = four) => Class = table (3.0/0.0) [1.0/0.0]
Except (Backrest = no) and (Arms = no) => Class = table (3.0/0.0) [2.0/1.0]

j48 clean

Backrest = no
| Arms = no: table (9.0/1.0)
| Arms = yes: Chair (6.0)
Backrest = yes: Chair (18.0)

Tertius

1. /* 0.711111 0.030303 */ Backrest = no and Arms = no ==> Class = table
2. /* 0.588889 0.060606 */ Backrest = no and FlatSurface = yes ==> Class = table
3. /* 0.571429 0.000000 */ Backrest = yes ==> Class = Chair
4. /* 0.545367 0.212121 */ Backrest = no ==> Class = table
5. /* 0.525266 0.121212 */ Arms = no and FlatSurface = yes ==> Class = table
6. /* 0.521699 0.000000 */ Arms = yes ==> Class = Chair
7. /* 0.501347 0.272727 */ Arms = no ==> Class = table
8. /* 0.434783 0.000000 */ Backrest = no and Arms = no and Round = yes ==> Class = table
9. /* 0.434783 0.000000 */ Legs = four and Backrest = no and Arms = no ==> Class = table
10. /* 0.413916 0.030303 */ FlatSurface = no ==> Class = Chair
11. /* 0.396154 0.272727 */ FlatSurface = yes ==> Class = table

Figure 17. chairs-dss - decisions shown

If you have done all the above correctly and made no data transcription errors the j48, Ridor and T-mode predictions should agree as shown in Figure 17.

References

Weka 3 (2009), Data Mining software available at http://www.cs.waikato.ac.nz/~ml/weka/index.html, [Accessed 2 September 2011]

Witten, I.H., Frank, E. & Hall, M.A. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, 3/e, Morgan Kaufmann; ISBN: 978-0-12-374856-0, 2011. (website: http://www.cs.waikato.ac.nz/ml/weka/)