

CSCI 5408, Winter 2017

Assignment 5: Association Rule Mining and Application

Submission date: 26TH March 2017

Student ID: B00741912

Name: Yamuna Jayabalan

Student ID: B00762641

Name: Bharat Jain

Division of Task:

Yamuna Jayabalan – Worked on R Script with the relevant report & README

Bharat Jain – Worked on Weka Tool with relevant report & README

List of Figures:

Figures	Page no.
Fig 1. Pre-Processing 6
Fig 2. Applying Filter 7
Fig 3. Set Configurations 8
Fig 4. Result Buffer 8
Fig 5. Result instance 9
Fig 6. Instance of R script11
Fig 7. Visualization Plot 111
Fig 8. Visualization Plot 212

- **Application scenario and dataset:**

Association rule mining is basically a enhanced field in the Data Mining World. This Assignment focuses on in-depth learning of association rule mining, method and applications. To get the overview of Association rule mining let's get a glimpse about it.

Association rule learning is a rule based learning method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness. We will demonstrate how to perform the association rule mining using different tools i.e. WEKA & R.

WEKA is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. [1]

Data Description:

Column	Description
state	The name of state in US
year	Year of forecast
candidate	Candidate Name
forecast_prob	Probability of the candidate becoming a senate based on the forecast
result	Result of the candidate becoming a senate or not
winflag	Flag set based on the result (win =1, lose=0)

Sample data from dataset:

state	year	candidate	forecast_prob	result	winflag
Alabama	2008	Sessions	1	Win	1
Alabama	2008	Figures	0	Lose	0
Alabama	2010	Shelby	1	Win	1
Alabama	2010	Barnes	0	Loss	0
Alaska	2008	Begich	1	Win	1
Alaska	2008	Stevens	0	Lose	0
Alaska	2010	Miller	0.71	Lose	0
Alaska	2010	Murkowski	0.21	Win	1
Alaska	2010	McAdams	0.08	Loss	0

- **Data processing and target dataset:**

No pre-processing of data sets was required.

- **Association mining solution:**

1. **Description of Association rule mining algorithm & creation:**

Association rules analysis is a technique to predict the way items are associated to each other. Three measures are used to find the association.

Transaction 1	A, B, C, D
Transaction 2	A, B, C
Transaction 3	A, B
Transaction 4	A, E
Transaction 5	F, B, C, D
Transaction 6	F, B, C
Transaction 7	F, B
Transaction 8	F, E

Measure 1. (Support). This measure is used to describe the popularity of an Item set. It is measured by the proportion of transactions in which an item set appears. In some cases, item-sets can also contain multiple items. For instance, the support of {A, B, C} is 2 out of 8, or 25% for example. If you discover that sales of items beyond a certain proportion tend to have a significant impact on your profits, you might consider using that proportion as your support threshold. You may then identify item sets with support values above this threshold as significant item sets.

Measure 2. (Confidence): This measure is used to describe the likeliness of an item Y being purchased when item X is purchased, expressed as {X → Y}. This is measured by the proportion of transactions with item X, in which item Y also appears. For example, the confidence of {A → B} is 3 out of 4, or 75%. One drawback of the confidence measure is that it might misrepresent the importance of an association. This is because it only accounts for how popular A is, but not B. If B is also very popular in general, there will be a higher chance that a transaction containing A will also contain B, thus inflating the confidence measure. To account for the base popularity of both constituent items, a third measure called lift is used.

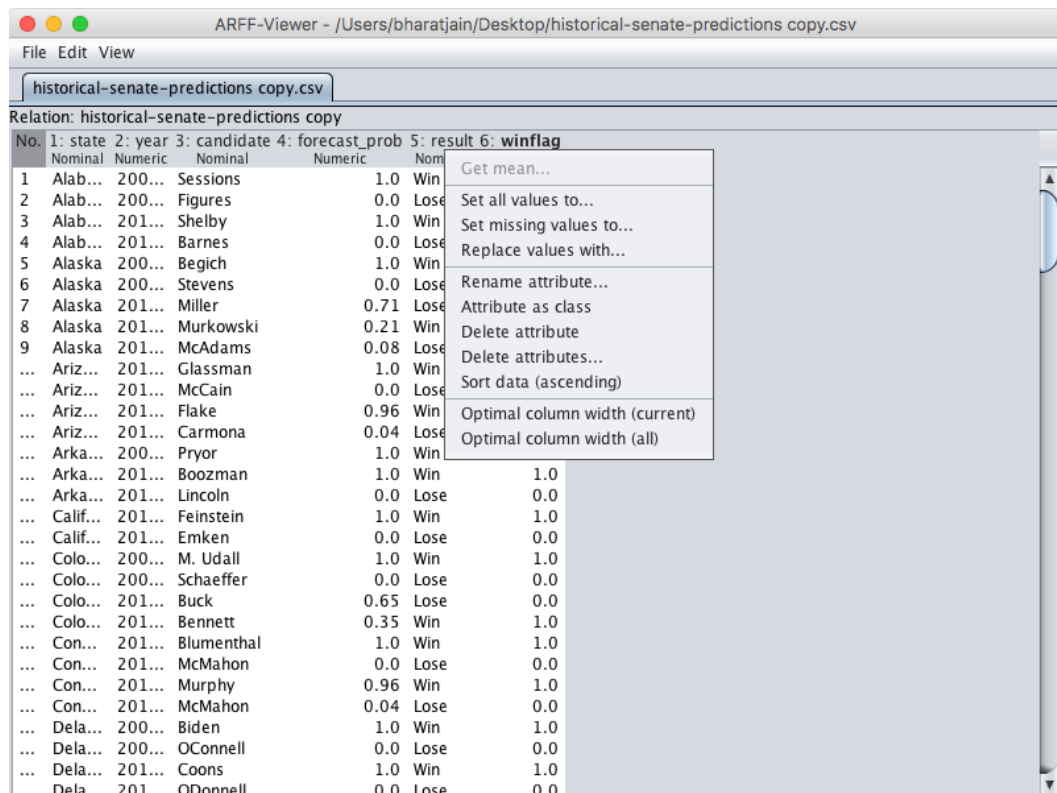
Measure 3. (Lift): This measure describes the likeliness of an item Y being purchased when item X is purchased, while controlling the popularity of item Y. For example, the lift of {A → B} is 1, which implies no association between items. A lift value greater than 1 says that item Y is likely to be bought if item X is bought, while a value less than 1 means that item Y is unlikely to be bought if item X is bought.

2. Steps taken to perform Association rule mining in WEKA:

As already described in the introduction about what WEKA tool is all about. Let's directly get into the task as specified. To begin with we will specify the steps taken:

Step 1.

The first step should be pre-processing & cleaning of data sets. To clean the data sets there are two ways. One can Either use SSIS tool or the Weka tool itself. The below screen shot depicts how the cleaning process looks like.



The screenshot shows the ARFF-Viewer application window titled "ARFF-Viewer - /Users/bharatjain/Desktop/historical-senate-predictions copy.csv". The menu bar includes "File", "Edit", and "View". The toolbar shows "historical-senate-predictions copy.csv". The main area displays a table with the following columns: "No.", "1: state", "2: year", "3: candidate", "4: forecast_prob", "5: result", and "6: winflag". The table contains 20 rows of data. A context menu is open over the table, listing the following options: "Get mean...", "Set all values to...", "Set missing values to...", "Replace values with...", "Rename attribute...", "Attribute as class", "Delete attribute", "Delete attributes...", "Sort data (ascending)", "Optimal column width (current)", and "Optimal column width (all)".

No.	1: state	2: year	3: candidate	4: forecast_prob	5: result	6: winflag
	Nominal	Numeric	Nominal	Numeric	Nominal	
1	Alab...	200...	Sessions	1.0	Win	
2	Alab...	200...	Figures	0.0	Lose	
3	Alab...	201...	Shelby	1.0	Win	
4	Alab...	201...	Barnes	0.0	Lose	
5	Alaska	200...	Begich	1.0	Win	
6	Alaska	200...	Stevens	0.0	Lose	
7	Alaska	201...	Miller	0.71	Lose	
8	Alaska	201...	Murkowski	0.21	Win	
9	Alaska	201...	McAdams	0.08	Lose	
...	Ariz...	201...	Glassman	1.0	Win	
...	Ariz...	201...	McCain	0.0	Lose	
...	Ariz...	201...	Flake	0.96	Win	
...	Ariz...	201...	Carmona	0.04	Lose	
...	Arka...	200...	Pryor	1.0	Win	
...	Arka...	201...	Boozman	1.0	Win	1.0
...	Arka...	201...	Lincoln	0.0	Lose	0.0
...	Calif...	201...	Feinstein	1.0	Win	1.0
...	Calif...	201...	Emken	0.0	Lose	0.0
...	Colo...	200...	M. Udall	1.0	Win	1.0
...	Colo...	200...	Schaeffer	0.0	Lose	0.0
...	Colo...	201...	Buck	0.65	Lose	0.0
...	Colo...	201...	Bennett	0.35	Win	1.0
...	Con...	201...	Blumenthal	1.0	Win	1.0
...	Con...	201...	McMahon	0.0	Lose	0.0
...	Con...	201...	Murphy	0.96	Win	1.0
...	Con...	201...	McMahon	0.04	Lose	0.0
...	Dela...	200...	Biden	1.0	Win	1.0
...	Dela...	200...	OConnell	0.0	Lose	0.0
...	Dela...	201...	Coons	1.0	Win	1.0
...	Dela...	201...	O'Donnell	0.0	Lose	0.0

Fig 1. Pre-Processing

As it can be seen that once you import the data set all the Tuples are displayed. **Data Cleaning** can be performed here itself. These are the options which can be performed to do data pre-processing & cleaning:

- **Set all values to...** - This means if you would like to alter the whole column to any other specific values it can be done using this option.
- **Set missing values to...** - This option enables us to assign the missing values to some standard values which may further result in correct output.

- **Replace values with...** - If in case we would like to alter the values for particular column then it's really important to have such functionality where all the values can be replaced with one go. May be millions of transactions.
- **Attribute as class...** - This is one of the important functionality where we can set the attribute as class which will move the particulate column at the last.
- Similarly there are lot different functionality provided by WEKA tool.

Step 2.

Next step is to apply the filter on numeric values. **REMEMBER** that WEKA tool does not work with default integral values. So, to proceed further we need to apply the **filter** of **NumericToNominal** and click apply.

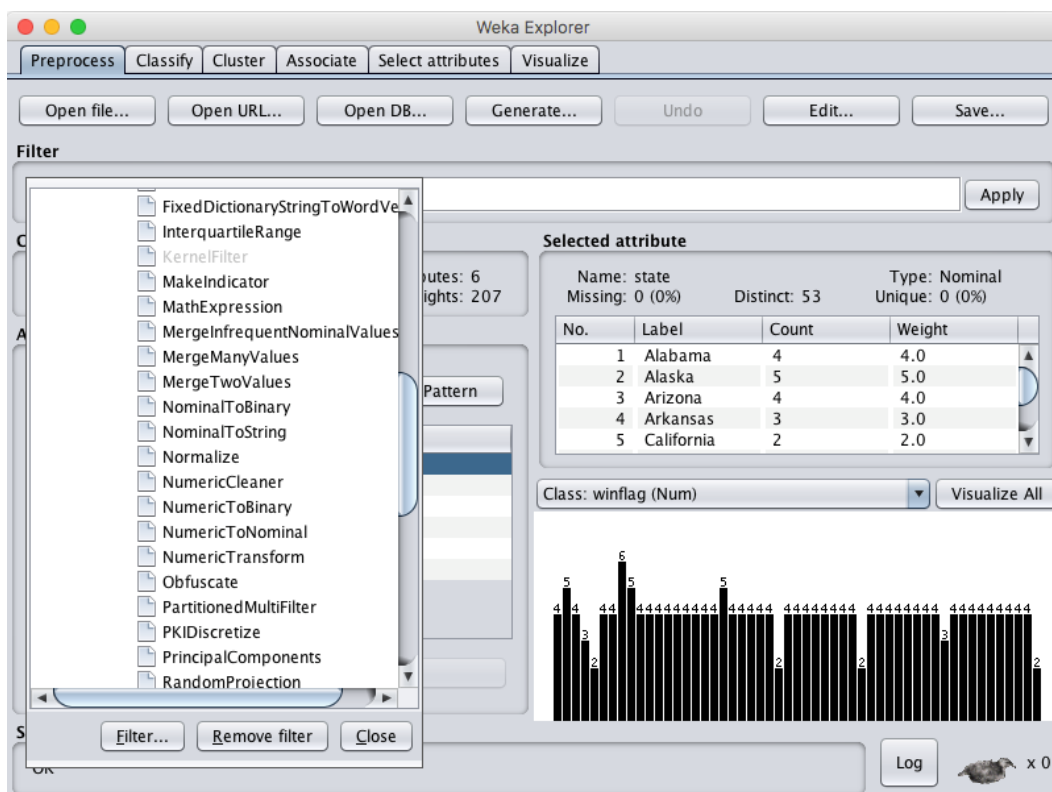


Fig 2. Applying Filter

Before the actual association is performed all the integral values is converted to Nominal. In our data set there are columns which consists of integral values.

Step 3.

Next step is to navigate to Associate **Tab** and select the **Associator as Apriori**. Open the settings to modify Support Count and minimum confidence as seen in below screen shot.

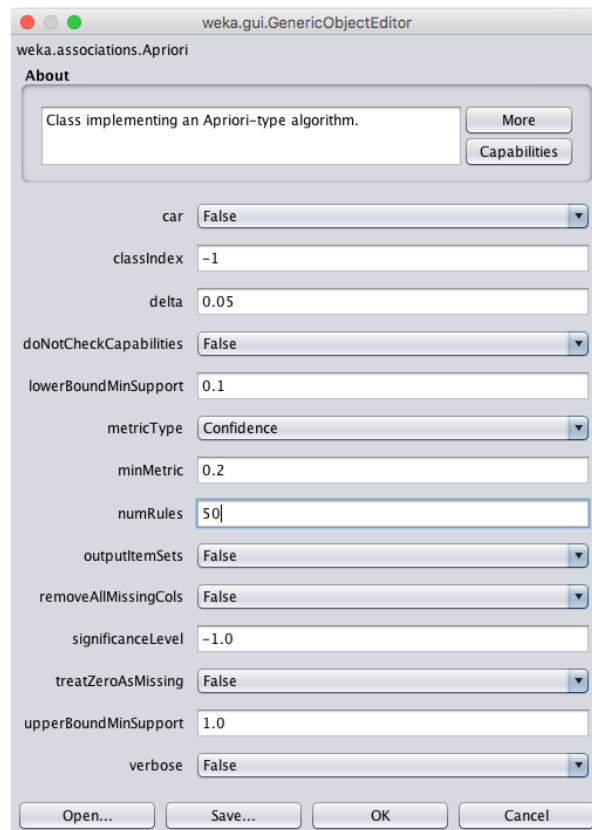


Fig 3. Set Configurations

It can be seen that we have selected the minimum support count as 0.1 and Confidence as 0.2. Click on ok and proceed with the operations. A result buffer will be generated as seen in below screen shot.

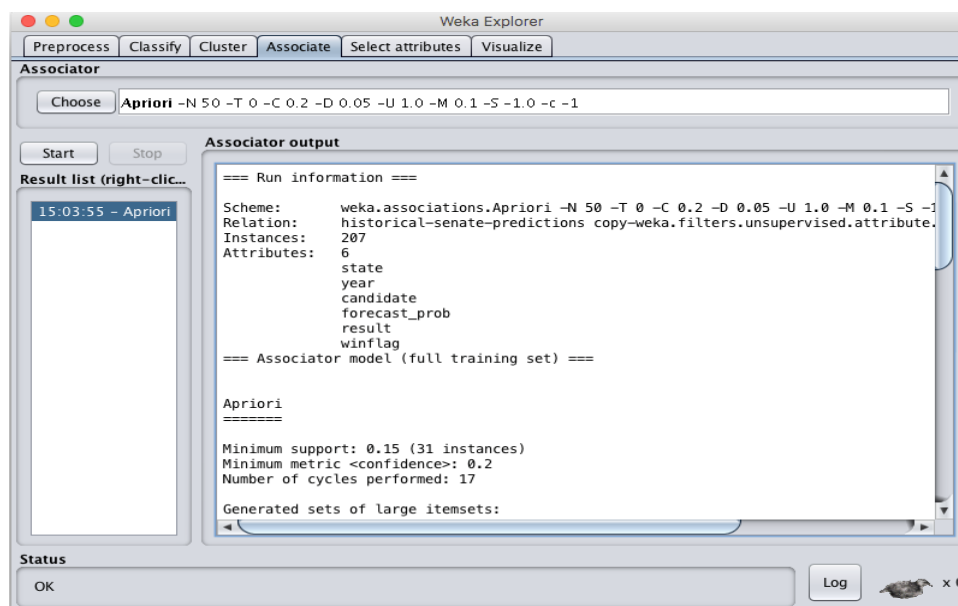


Fig 4. Result Buffer

The result buffer is stored in **Apriori.txt**

Result Instance:

Best rules found:

```

1. winflag=0 104 ==> result=Lose 104    <conf:(1)> lift:(1.99) lev:(0.25) [51] conv:(51.75)
2. result=Lose 104 ==> winflag=0 104    <conf:(1)> lift:(1.99) lev:(0.25) [51] conv:(51.75)
3. winflag=1 103 ==> result=Win 103     <conf:(1)> lift:(2.01) lev:(0.25) [51] conv:(51.75)
4. result=Win 103 ==> winflag=1 103     <conf:(1)> lift:(2.01) lev:(0.25) [51] conv:(51.75)
5. forecast_prob=1 68 ==> result=Win 68  <conf:(1)> lift:(2.01) lev:(0.17) [34] conv:(34.16)
6. forecast_prob=1 68 ==> winflag=1 68   <conf:(1)> lift:(2.01) lev:(0.17) [34] conv:(34.16)
7. forecast_prob=1 winflag=1 68 ==> result=Win 68  <conf:(1)> lift:(2.01) lev:(0.17) [34] conv:(34.16)
8. forecast_prob=1 result=Win 68 ==> winflag=1 68  <conf:(1)> lift:(2.01) lev:(0.17) [34] conv:(34.16)
9. forecast_prob=1 68 ==> result=Win winflag=1 68  <conf:(1)> lift:(2.01) lev:(0.17) [34] conv:(34.16)
10. forecast_prob=0 67 ==> result=Lose 67  <conf:(1)> lift:(1.99) lev:(0.16) [33] conv:(33.34)
11. forecast_prob=0 67 ==> winflag=0 67   <conf:(1)> lift:(1.99) lev:(0.16) [33] conv:(33.34)
12. forecast_prob=0 winflag=0 67 ==> result=Lose 67  <conf:(1)> lift:(1.99) lev:(0.16) [33] conv:(33.34)
13. forecast_prob=0 result=Lose 67 ==> winflag=0 67  <conf:(1)> lift:(1.99) lev:(0.16) [33] conv:(33.34)
14. forecast_prob=0 67 ==> result=Lose winflag=0 67  <conf:(1)> lift:(1.99) lev:(0.16) [33] conv:(33.34)
15. year=2010 winflag=0 37 ==> result=Lose 37  <conf:(1)> lift:(1.99) lev:(0.09) [18] conv:(18.41)
16. year=2010 result=Lose 37 ==> winflag=0 37   <conf:(1)> lift:(1.99) lev:(0.09) [18] conv:(18.41)
17. year=2010 winflag=1 36 ==> result=Win 36   <conf:(1)> lift:(2.01) lev:(0.09) [18] conv:(18.09)
18. year=2010 result=Win 36 ==> winflag=1 36   <conf:(1)> lift:(2.01) lev:(0.09) [18] conv:(18.09)
19. year=2008 winflag=1 34 ==> result=Win 34   <conf:(1)> lift:(2.01) lev:(0.08) [17] conv:(17.08)
20. year=2008 result=Win 34 ==> winflag=1 34   <conf:(1)> lift:(2.01) lev:(0.08) [17] conv:(17.08)
21. year=2012 winflag=0 34 ==> result=Lose 34   <conf:(1)> lift:(1.99) lev:(0.08) [16] conv:(16.92)
22. year=2012 result=Lose 34 ==> winflag=0 34   <conf:(1)> lift:(1.99) lev:(0.08) [16] conv:(16.92)
23. year=2008 winflag=0 33 ==> result=Lose 33   <conf:(1)> lift:(1.99) lev:(0.08) [16] conv:(16.42)

```

Fig 5. Result instance

Let's Discuss about the output generated by WEKA Tool. An important factor here in. i.e. **Delta**.

- **What does Delta means ?**

The default value of Delta is 0.05. During the association rule calculation there are cycles associated with each and every iterations. So for every iteration the value of minimum support count is reduced by 0.05. By default there will be more number of rules generated based on number of cycles which related to value of Delta.

- There were 17 cycles & overall 207 instances with 6 Attribute were found.
- The size of item sets are divided in 3 different category i.e. L1, L2 & L3.
- The schema followed is: **weka.associations.Apriori -N 50 -T 0 -C 0.2 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1**

3. Steps taken to perform Association rule mining in R:

Step 1:

If the result of code has to be saved in a file, sink() function is used at the beginning of the code. The first parameter being file name, it is the name of file in which output of code is saved. Split parameter when set “TRUE”, will display the output value in the console as well.

Code snippet:

```
sink("output.txt", split=TRUE)
```

Step 2:

Importing the arule library to implement apriori algorithm and arulesViz to perform visualizations of rules.

Code snippet:

```
library(arules)
library(arulesViz)
```

Step 3:

The required dataset is read into the variable trans, as next step

Code snippet:

```
trans<-read.transactions("historical-senate-predictions.csv",format = "basket",
sep=",", rm.duplicates=TRUE)
```

Step 4:

Apriori function is called and trans variable is passed, along with other parameters. The parameter “minlen=2” will display the LHS which are non-empty, support and confidence parameter values are passed as desired. In this step, the rules are created and stored in rules variable.

Code snippet:

```
rules<-apriori(trans,parameter=list(minlen=2,supp=0.1,conf=0.2))
```

Step 5:

The upcoming lines of code is used to view and visualize the rules. The graph is also stored as image files. “sink()” function at the end of the code, closes the file which was opened in first line to save the code output.

Code snippet

```
summary(rules)
inspect(rules)
jpeg('plot1.jpg')
plot(rules)
dev.off()
jpeg('plot2.jpg')
plot(rules, method="graph", control=list(type="items"))
dev.off()
sink()
```

Below is the execution screen capture for R Script in R Studio.

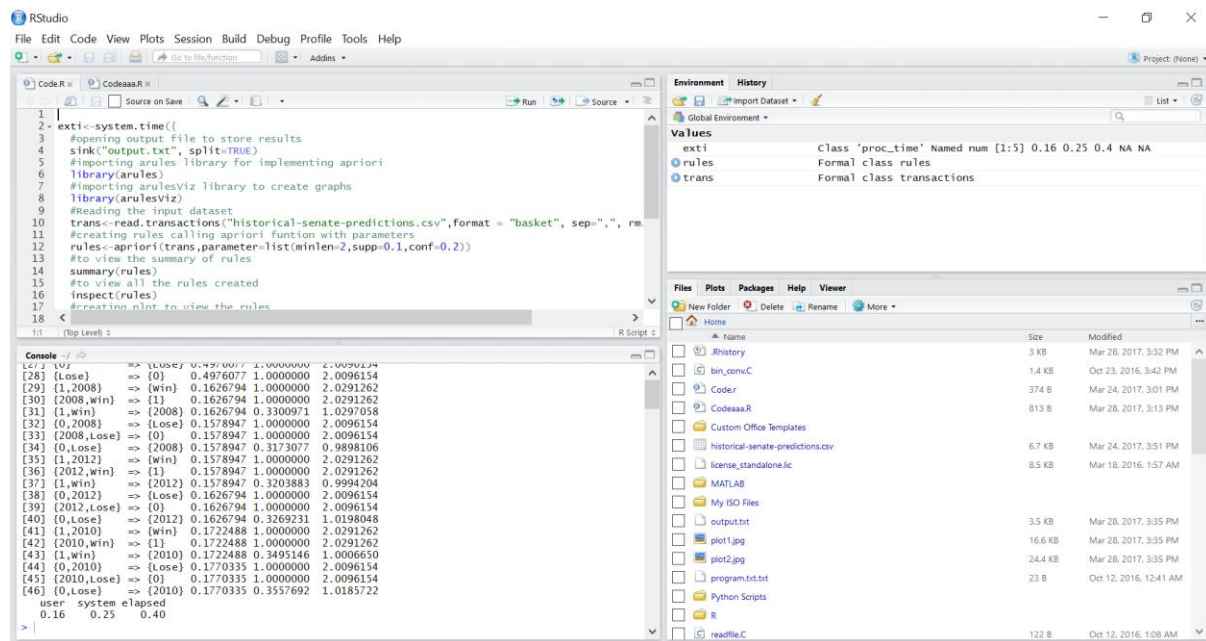


Fig 6. Instance of R script

The above screen shot has the execution time printed at the end using system.time() function. (Refer 4th link in reference section). The output has user, system and elapsed time. User time is the time taken by CPU to execute user instructions and system time is the time taken by CPU to execute system process. Elapsed time is the total execution time of the code.

As mentioned in our script we have plotted the graph based on the rules generated. We will list down the visual representation of the graph as below.

1. Support vs Confidence graph for 46 Rules generated by R.

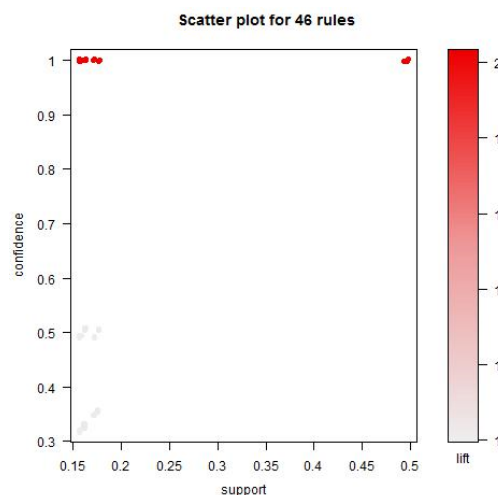


Fig 7. Visualization Plot 1

2. Node connection graph for Target attribute for Lose & win for 3 different years.

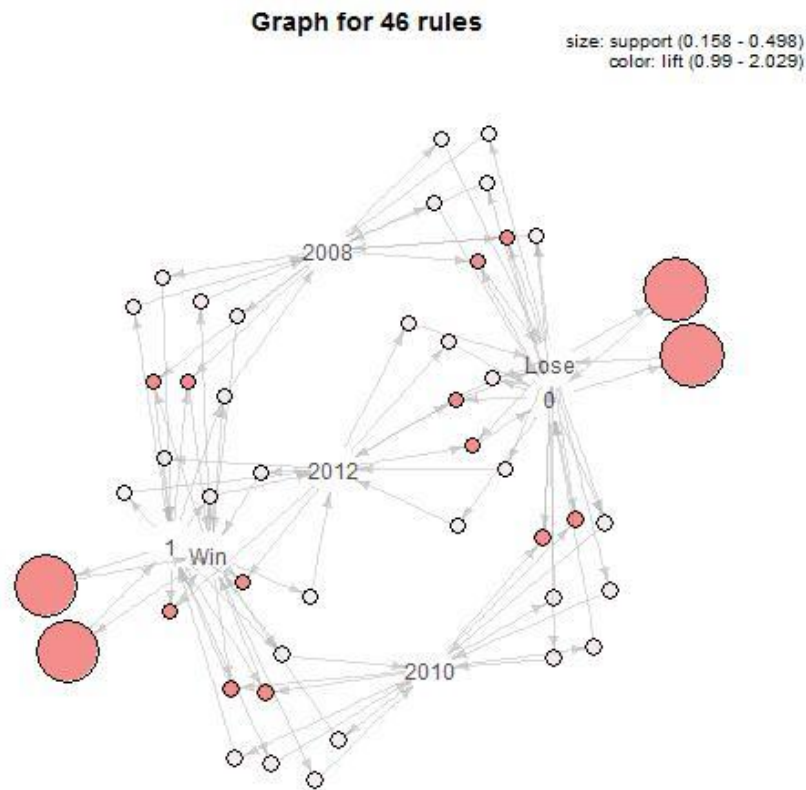


Fig 8. Visualization Plot 2

Visualization always plays an important role in better understanding of result & decision making. The above graph clearly depicts the density based on quantity.

Note: The R script is present in the file named **Code.R** and the Result Buffer is stored in file named **output.txt**.

Comparison:

As we have already mentioned in each Section of R & Weka related to the output and performance. We will get an overview of what exactly the difference is.

Time taken by R is (0.4 seconds) comparatively less than WEKA. This is because R script by default does not have any number of cycles to recursively generate the rules whereas in WEKA the factor called **Delta** is responsible for number of cycles. More the number of cycles more the rules will be generated and more time taken.

Observation Summary:

The assignment focuses on different tools to compare the get keen to variant of available tools to do the same task. Well, it's quite noticeable that the output generated by both the tools are significantly different or we can say that the **“output generated by R is subset of output generated by WEKA Tool in general”**.

There are many tools available in market to pre-process and clean the data before the actual task to be performed. SSIS is one of the tool which allows us to do all types of pre-processing. Here in with WEKA we can pre-process the similar kind with intact functionality related to pre-processing during the process of importing the data. As already discussed that Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.

Using R, code is written to read the given dataset and to create rules as output using Apriori algorithm. The code is written to eliminate the rules with empty LHS value by specifying the minimum length is as 2. The support value given is 0.1 and the confidence value is 0.2. For the given dataset, the apriori algorithm has created 46 rules based on the transactions. The code takes roughly 0.4 seconds to run. RStudio IDE has a simple and interactive user interface. It has a code window, a console to view output, a project window and viewer to view files/graph outputs.

References:

1. <http://www.cs.waikato.ac.nz/ml/weka/>
2. <http://www.kdnuggets.com/2016/04/association-rules-apriori-algorithm-tutorial.html>
3. <https://www.r-bloggers.com/implementing-apriori-algorithm-in-r/>
4. http://www.cookbook-r.com/Scripts_and_functions/Measuring_elapsed_time/