CS 422 Assignment – 3

Association Rule Mining

Name : Dhayalini Nagaraj

Student ID :A20359686

1. Use Weka
- Use "Associate" tab
1.1 Supermarket dataset: Show the top 4 association rules with Apriori using the default parameters. Discuss what are the main parameters for Apriori that you can modify. Modify support/confidence 3 times: low, medium high, experiment and report the summary of what your learned - explain briefly how your modifications affected the generation of the rules

Supermarket dataset:

There are about 4627 and 217 attributes related to the super market data set. There are 216 nominal attributes which represent the variety of departments that were available in the supermarket, apart from the class attribute 'total' which denotes whether the final sum price was higher or lower. The 4627 records represent the individual customer transactions. A value of 't' denotes the presence of item of that particular department in the market basket and the missing values indicate the items that the customer did not buy.

We use Apriori associative rule mining to find various associative rules for the super market data set. If an itemset is frequent, then all of its subsets must also be frequent.

Important parameters of Apriori algorithms.

delta: Iteratively decrease support by this factor. Reduces support until min support is reached or required number of rules has been generated.
Default value: 0.05

lowerBoundMinSupport: This value specifies the lower bound for minimum support.
Default value: 0.1

metricType: This parameter allows the user to choose the metric that the minMetric will work on. The different measures for ranking rules are confidence, lift, leverage and conviction.
Default value: Confidence

minMetric: This parameter takes the minimum value for the selected metric type.
Default value: 0.9

outputItemSets: This parameter prints the itemsets when set to true.
Default Value : False

removeAllMissingCols: When set to true, all the columns with missing values are removed and helps in generating rules for an ideal scenario.

Default value: 0.9

upperBoundMinSupport: This value specifies the upper bound for minimum support. Start iteratively decreasing minimum support from this value.
Default value: 1.0

numRules: This parameter tells Weka how many rules to print to the console.
Default value: 10

car: If enabled class association rules are mined instead of (general) association rules.
Default value: false

classIndex: Index of the class attribute. If set to -1, the last attribute is taken as class attribute.
Default value: -1

significanceLevel: Significance level. Significance test (confidence metric only).
Default value: -1.0

verbose: If enabled the algorithm will be run in verbose mode.
Default value: False

Top 4 association rules using default parameters.

Main parameters with default value:
lowerBoundMinSupport (support): 0.1
minMetric (confidence): 0.9

1. biscuits=t frozen foods=t fruit=t total=high 788 ==> bread and cake=t 723 <conf:(0.92)> lift:(1.27) lev:(0.03) [155] conv:(3.35)

2. baking needs=t biscuits=t fruit=t total=high 760 ==> bread and cake=t 696 <conf:(0.92)> lift:(1.27) lev:(0.03) [149] conv:(3.28)

3. baking needs=t frozen foods=t fruit=t total=high 770 ==> bread and cake=t 705 <conf:(0.92)> lift:(1.27) lev:(0.03) [150] conv:(3.27)

4. biscuits=t fruit=t vegetables=t total=high 815 ==> bread and cake=t 746 <conf:(0.92)> lift:(1.27) lev:(0.03) [159] conv:(3.26)

The number of cycles performed are 17 and count of large itemset levels generated was 6.

Bread and cake parameter is the most frequent attribute of supermarket dataset and it is clearly seen that the top rules always have bread and cake attribute. Similarly, total is another attribute. Total is a class attribute in the supermarket dataset.

Rule 1 suggests that the presence of biscuits, frozen foods, fruits and total attributes indicate a strong consequence on bread and cake. This shows that the default parameters are not very helpful in determining the actual association rule because it is dominated by the frequent items rather than a meaningful association.

The association rule results for modified support and count values.

To analyze the impact of the support and confidence values in mining for association rules using Apriori algorithm, the values of lowerBoundMinSupport and minMetric values were altered

lowerBoundMinSupport (Support)	minMetric (Confidence)	Cycle count	Large itemset levels	Top 10 association rules
Default lowerBoundMinSupport : 0.1 and High, Medium and Low confidence				
0.1	0.8	14	3	1. biscuits=t vegetables=t 1764 ==> bread and cake=t 1487 <conf:(0.84)> lift:(1.17) lev:(0.05) [217] conv:(1.78) 2. total=high 1679 ==> bread and cake=t 1413 <conf:(0.84)> lift:(1.17) lev:(0.04) [204] conv:(1.76) 3. biscuits=t milk-cream=t 1767 ==> bread and cake=t 1485 <conf:(0.84)> lift:(1.17) lev:(0.05) [213] conv:(1.75) 4. biscuits=t fruit=t 1837 ==> bread and cake=t 1541 <conf:(0.84)> lift:(1.17) lev:(0.05) [218] conv:(1.73) 5. biscuits=t frozen foods=t 1810 ==> bread and cake=t 1510 <conf:(0.83)> lift:(1.16) lev:(0.04) [207] conv:(1.69) 6. frozen foods=t fruit=t 1861 ==> bread and cake=t 1548 <conf:(0.83)> lift:(1.16) lev:(0.05) [208] conv:(1.66) 7. frozen foods=t milk-cream=t 1826 ==> bread and cake=t 1516 <conf:(0.83)> lift:(1.15) lev:(0.04) [201] conv:(1.65) 8. baking needs=t milk-cream=t 1907 ==> bread and cake=t 1580 <conf:(0.83)> lift:(1.15) lev:(0.04) [207] conv:(1.63) 9. milk-cream=t fruit=t 2038 ==> bread and cake=t 1684 <conf:(0.83)> lift:(1.15) lev:(0.05) [217] conv:(1.61) 10. baking needs=t biscuits=t 1764 ==> bread and cake=t 1456 <conf:(0.83)> lift:(1.15) lev:(0.04) [186] conv:(1.6)
0.1	0.5	11	2	1. biscuits=t 2605 ==> bread and cake=t 2083 <conf:(0.8)> lift:(1.11) lev:(0.04) [208] conv:(1.4) 2. milk-cream=t 2939 ==> bread and cake=t 2337 <conf:(0.8)> lift:(1.1) lev:(0.05) [221] conv:(1.37) 3. fruit=t 2962 ==> bread and cake=t 2325 <conf:(0.78)> lift:(1.09) lev:(0.04) [193] conv:(1.3) 4. baking needs=t 2795 ==> bread and cake=t 2191 <conf:(0.78)> lift:(1.09) lev:(0.04) [179] conv:(1.29) 5. frozen foods=t 2717 ==> bread and cake=t 2129 <conf:(0.78)> lift:(1.09) lev:(0.04) [173] conv:(1.29) 6. vegetables=t 2961 ==> bread and cake=t 2298 <conf:(0.78)> lift:(1.08) lev:(0.04) [167] conv:(1.25)

				7. vegetables=t 2961 ==> fruit=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 8. fruit=t 2962 ==> vegetables=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 9. bread and cake=t 3330 ==> milk-cream=t 2337 <conf:(0.7)> lift:(1.1) lev:(0.05) [221] conv:(1.22) 10. bread and cake=t 3330 ==> fruit=t 2325 <conf:(0.7)> lift:(1.09) lev:(0.04) [193] conv:(1.19)
0.1	0.2	11	2	1. biscuits=t 2605 ==> bread and cake=t 2083 <conf:(0.8)> lift:(1.11) lev:(0.04) [208] conv:(1.4) 2. milk-cream=t 2939 ==> bread and cake=t 2337 <conf:(0.8)> lift:(1.1) lev:(0.05) [221] conv:(1.37) 3. fruit=t 2962 ==> bread and cake=t 2325 <conf:(0.78)> lift:(1.09) lev:(0.04) [193] conv:(1.3) 4. baking needs=t 2795 ==> bread and cake=t 2191 <conf:(0.78)> lift:(1.09) lev:(0.04) [179] conv:(1.29) 5. frozen foods=t 2717 ==> bread and cake=t 2129 <conf:(0.78)> lift:(1.09) lev:(0.04) [173] conv:(1.29) 6. vegetables=t 2961 ==> bread and cake=t 2298 <conf:(0.78)> lift:(1.08) lev:(0.04) [167] conv:(1.25) 7. vegetables=t 2961 ==> fruit=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 8. fruit=t 2962 ==> vegetables=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 9. bread and cake=t 3330 ==> milk-cream=t 2337 <conf:(0.7)> lift:(1.1) lev:(0.05) [221] conv:(1.22) 10. bread and cake=t 3330 ==> fruit=t 2325 <conf:(0.7)> lift:(1.09) lev:(0.04) [193] conv:(1.19)
Low lowerBoundMinSupport : 0.3 and High, Medium and Low confidence				
0.3	0.8	14	3	1. biscuits=t vegetables=t 1764 ==> bread and cake=t 1487 <conf:(0.84)> lift:(1.17) lev:(0.05) [217] conv:(1.78) 2. total=high 1679 ==> bread and cake=t 1413 <conf:(0.84)> lift:(1.17) lev:(0.04) [204] conv:(1.76) 3. biscuits=t milk-cream=t 1767 ==> bread and cake=t 1485 <conf:(0.84)> lift:(1.17) lev:(0.05) [213] conv:(1.75) 4. biscuits=t fruit=t 1837 ==> bread and cake=t 1541 <conf:(0.84)> lift:(1.17) lev:(0.05) [218] conv:(1.73) 5. biscuits=t frozen foods=t 1810 ==> bread and cake=t 1510 <conf:(0.83)> lift:(1.16) lev:(0.04) [207] conv:(1.69) 6. frozen foods=t fruit=t 1861 ==> bread and cake=t 1548 <conf:(0.83)> lift:(1.16) lev:(0.05) [208] conv:(1.66) 7. frozen foods=t milk-cream=t 1826 ==> bread and cake=t 1516 <conf:(0.83)> lift:(1.15) lev:(0.04) [201] conv:(1.65) 8. baking needs=t milk-cream=t 1907 ==> bread and cake=t 1580 <conf:(0.83)> lift:(1.15) lev:(0.04) [207]

				conv:(1.63) 9. milk-cream=t fruit=t 2038 ==> bread and cake=t 1684 <conf:(0.83)> lift:(1.15) lev:(0.05) [217] conv:(1.61) 10. baking needs=t biscuits=t 1764 ==> bread and cake=t 1456 <conf:(0.83)> lift:(1.15) lev:(0.04) [186] conv:(1.6)
0.3	0.5	11	2	1. biscuits=t 2605 ==> bread and cake=t 2083 <conf:(0.8)> lift:(1.11) lev:(0.04) [208] conv:(1.4) 2. milk-cream=t 2939 ==> bread and cake=t 2337 <conf:(0.8)> lift:(1.1) lev:(0.05) [221] conv:(1.37) 3. fruit=t 2962 ==> bread and cake=t 2325 <conf:(0.78)> lift:(1.09) lev:(0.04) [193] conv:(1.3) 4. baking needs=t 2795 ==> bread and cake=t 2191 <conf:(0.78)> lift:(1.09) lev:(0.04) [179] conv:(1.29) 5. frozen foods=t 2717 ==> bread and cake=t 2129 <conf:(0.78)> lift:(1.09) lev:(0.04) [173] conv:(1.29) 6. vegetables=t 2961 ==> bread and cake=t 2298 <conf:(0.78)> lift:(1.08) lev:(0.04) [167] conv:(1.25) 7. vegetables=t 2961 ==> fruit=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 8. fruit=t 2962 ==> vegetables=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 9. bread and cake=t 3330 ==> milk-cream=t 2337 <conf:(0.7)> lift:(1.1) lev:(0.05) [221] conv:(1.22) 10. bread and cake=t 3330 ==> fruit=t 2325 <conf:(0.7)> lift:(1.09) lev:(0.04) [193] conv:(1.19)
0.3	0.2	11	2	1. biscuits=t 2605 ==> bread and cake=t 2083 <conf:(0.8)> lift:(1.11) lev:(0.04) [208] conv:(1.4) 2. milk-cream=t 2939 ==> bread and cake=t 2337 <conf:(0.8)> lift:(1.1) lev:(0.05) [221] conv:(1.37) 3. fruit=t 2962 ==> bread and cake=t 2325 <conf:(0.78)> lift:(1.09) lev:(0.04) [193] conv:(1.3) 4. baking needs=t 2795 ==> bread and cake=t 2191 <conf:(0.78)> lift:(1.09) lev:(0.04) [179] conv:(1.29) 5. frozen foods=t 2717 ==> bread and cake=t 2129 <conf:(0.78)> lift:(1.09) lev:(0.04) [173] conv:(1.29) 6. vegetables=t 2961 ==> bread and cake=t 2298 <conf:(0.78)> lift:(1.08) lev:(0.04) [167] conv:(1.25) 7. vegetables=t 2961 ==> fruit=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 8. fruit=t 2962 ==> vegetables=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 9. bread and cake=t 3330 ==> milk-cream=t 2337 <conf:(0.7)> lift:(1.1) lev:(0.05) [221] conv:(1.22) 10. bread and cake=t 3330 ==> fruit=t 2325 <conf:(0.7)> lift:(1.09) lev:(0.04) [193] conv:(1.19)
Medium lowerBoundMinSupport : 0.5 and High, Medium and Low confidence				
0.5	0.8			No best rules were achieved
0.5	0.5	10	2	1. milk-cream=t 2939 ==> bread and cake=t 2337

				<conf:(0.8)> lift:(1.1) lev:(0.05) [221] conv:(1.37) 2. fruit=t 2962 ==> bread and cake=t 2325 <conf:(0.78)> lift:(1.09) lev:(0.04) [193] conv:(1.3) 3. bread and cake=t 3330 ==> milk-cream=t 2337 <conf:(0.7)> lift:(1.1) lev:(0.05) [221] conv:(1.22) 4. bread and cake=t 3330 ==> fruit=t 2325 <conf:(0.7)> lift:(1.09) lev:(0.04) [193] conv:(1.19)
0.5	0.2	10	2	1. milk-cream=t 2939 ==> bread and cake=t 2337 <conf:(0.8)> lift:(1.1) lev:(0.05) [221] conv:(1.37) 2. fruit=t 2962 ==> bread and cake=t 2325 <conf:(0.78)> lift:(1.09) lev:(0.04) [193] conv:(1.3) 3. bread and cake=t 3330 ==> milk-cream=t 2337 <conf:(0.7)> lift:(1.1) lev:(0.05) [221] conv:(1.22) 4. bread and cake=t 3330 ==> fruit=t 2325 <conf:(0.7)> lift:(1.09) lev:(0.04) [193] conv:(1.19)
High lowerBoundMinSupport : 0.7 and High, Medium and Low confidence				
0.7	0.8			No large itemsets and rules found!
0.7	0.5			No large itemsets and rules found!
0.7	0.2			No large itemsets and rules found!

Inferences on association rule generated by modifying support and confidence values

- When we set low lowerBoundMinimum support (S) and high minMetric confidence(C) e.g S = 0.1 and C = 0.8. In these cases, top 10 rules are generated. The confidence value is high for the rules but they are not interesting and would not be reliable since their support is comparatively low and they map to the most frequent item which is bread and cake.

- The top 10 rules are not of great importance because their confidence was high and the lower minimum support criteria was very low. In general, a lot of rules can get high confidence but the associations are not always true.

- When we set support to low value and increase the confidence to medium and low value we see that the cycle count has been reduced and the large item sets is also reduced. New association rules is also generated.

- When we slightly increase the support value and use high value for confidence best rules were not achieved. When we use medium support value and medium or low confidence value, instead of 10 rules only 4 best rules were achieved. The association rules were completely new when compared to the above rules. The number of cycles performed and large itemset also differes.

- When confidence value is medium or lower, same set of association rules were generated most of the times.

- When we increase the support, the rules are generated with greater support and acceptable confidence. Rules of this case seem more valuable than the types of rules addressed in the above points. If both support and confidence are high, rules are not

generated sometimes. This is because we cannot expect interesting associations to happen all the time. A certain optimum confidence on these associations would suffice.

- High support values (eg. S=0.7) does not produce any rules because there are no such itemset that is massively common among the transactions. No large itemset and rules are found.

- The results clearly shows that the low support values result in least interesting rules and high confidence values results implies good association rules but are influenced by frequent itemsets. Thus medium value for support and confidence achieves more reasonable association rules.

1.2 Use the attribute selection tab to remove attributes that appear in most rules. Run the same experiments as above and report what you observe.

<u>Results after removing attributes that appear in most rules</u>.

lowerBoundMinSupport (Support)	minMetric (Confidence)	Cycle count	Large itemset levels	Top 10 association rules
Removed 'bread and cake' lowerBoundMinSupport : 0.1 and Confidence 0.9				
0.1	0.9	18	6	1. baking needs=t beef=t fruit=t total=high 527 ==> vegetables=t 485 <conf:(0.92)> lift:(1.44) lev:(0.03) [147] conv:(4.41) 2. milk-cream=t beef=t fruit=t total=high 512 ==> vegetables=t 464 <conf:(0.91)> lift:(1.42) lev:(0.03) [136] conv:(3.76) 3. biscuits=t beef=t fruit=t total=high 514 ==> vegetables=t 465 <conf:(0.9)> lift:(1.41) lev:(0.03) [136] conv:(3.7) 4. frozen foods=t beef=t fruit=t total=high 543 ==> vegetables=t 491 <conf:(0.9)> lift:(1.41) lev:(0.03) [143] conv:(3.69)
Removed 'bread and cake' lowerBoundMinSupport : 0.1 and High, Medium, Low Confidence				
0.1	0.8	16	4	1. fruit=t total=high 1243 ==> vegetables=t 1050 <conf:(0.84)> lift:(1.32) lev:(0.06) [254] conv:(2.31) 2. beef=t fruit=t 1186 ==> vegetables=t 986 <conf:(0.83)> lift:(1.3) lev:(0.05) [227] conv:(2.12) 3. vegetables=t total=high 1270 ==> fruit=t 1050 <conf:(0.83)> lift:(1.29) lev:(0.05) [237] conv:(2.07) 4. biscuits=t milk-cream=t vegetables=t 1236 ==> fruit=t 1015 <conf:(0.82)> lift:(1.28) lev:(0.05) [223] conv:(2) 5. biscuits=t total=high 1228 ==> frozen foods=t 1006 <conf:(0.82)> lift:(1.4) lev:(0.06) [284] conv:(2.27) 6. baking needs=t frozen foods=t fruit=t 1281 ==> vegetables=t 1040 <conf:(0.81)> lift:(1.27) lev:(0.05)

				[220] conv:(1.91) 7. baking needs=t biscuits=t vegetables=t 1263 ==> fruit=t 1021 <conf:(0.81)> lift:(1.26) lev:(0.05) [212] conv:(1.87) 8. tissues-paper prd=t total=high 1156 ==> baking needs=t 933 <conf:(0.81)> lift:(1.34) lev:(0.05) [234] conv:(2.04) 9. biscuits=t frozen foods=t vegetables=t 1290 ==> fruit=t 1039 <conf:(0.81)> lift:(1.26) lev:(0.05) [213] conv:(1.84) 10. dairy foods=t vegetables=t 1176 ==> fruit=t 947 <conf:(0.81)> lift:(1.26) lev:(0.04) [194] conv:(1.84)
0.1	0.5	12	2	1. vegetables=t 2961 ==> fruit=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 2. fruit=t 2962 ==> vegetables=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 3. baking needs=t 2795 ==> vegetables=t 1949 <conf:(0.7)> lift:(1.09) lev:(0.03) [160] conv:(1.19) 4. milk-cream=t 2939 ==> fruit=t 2038 <conf:(0.69)> lift:(1.08) lev:(0.03) [156] conv:(1.17) 5. frozen foods=t 2717 ==> vegetables=t 1882 <conf:(0.69)> lift:(1.08) lev:(0.03) [143] conv:(1.17) 6. milk-cream=t 2939 ==> vegetables=t 2025 <conf:(0.69)> lift:(1.08) lev:(0.03) [144] conv:(1.16) 7. fruit=t 2962 ==> milk-cream=t 2038 <conf:(0.69)> lift:(1.08) lev:(0.03) [156] conv:(1.17) 8. frozen foods=t 2717 ==> fruit=t 1861 <conf:(0.68)> lift:(1.07) lev:(0.03) [121] conv:(1.14) 9. vegetables=t 2961 ==> milk-cream=t 2025 <conf:(0.68)> lift:(1.08) lev:(0.03) [144] conv:(1.15) 10. baking needs=t 2795 ==> milk-cream=t 1907 <conf:(0.68)> lift:(1.07) lev:(0.03) [131] conv:(1.15)
0.1	0.2	12	2	1. vegetables=t 2961 ==> fruit=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 2. fruit=t 2962 ==> vegetables=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 3. baking needs=t 2795 ==> vegetables=t 1949 <conf:(0.7)> lift:(1.09) lev:(0.03) [160] conv:(1.19) 4. milk-cream=t 2939 ==> fruit=t 2038 <conf:(0.69)> lift:(1.08) lev:(0.03) [156] conv:(1.17) 5. frozen foods=t 2717 ==> vegetables=t 1882 <conf:(0.69)> lift:(1.08) lev:(0.03) [143] conv:(1.17) 6. milk-cream=t 2939 ==> vegetables=t 2025 <conf:(0.69)> lift:(1.08) lev:(0.03) [144] conv:(1.16) 7. fruit=t 2962 ==> milk-cream=t 2038 <conf:(0.69)> lift:(1.08) lev:(0.03) [156] conv:(1.17) 8. frozen foods=t 2717 ==> fruit=t 1861 <conf:(0.68)> lift:(1.07) lev:(0.03) [121] conv:(1.14) 9. vegetables=t 2961 ==> milk-cream=t 2025 <conf:(0.68)> lift:(1.08) lev:(0.03) [144] conv:(1.15)

				10. baking needs=t 2795 ==> milk-cream=t 1907 <conf:(0.68)> lift:(1.07) lev:(0.03) [131] conv:(1.15)
Removed 'bread and cake' lowerBoundMinSupport : 0.3 and High, Medium, Low Confidence				
0.3	0.9			No best rules were achieved
0.3	0.5	12	2	1. vegetables=t 2961 ==> fruit=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 2. fruit=t 2962 ==> vegetables=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 3. baking needs=t 2795 ==> vegetables=t 1949 <conf:(0.7)> lift:(1.09) lev:(0.03) [160] conv:(1.19) 4. milk-cream=t 2939 ==> fruit=t 2038 <conf:(0.69)> lift:(1.08) lev:(0.03) [156] conv:(1.17) 5. frozen foods=t 2717 ==> vegetables=t 1882 <conf:(0.69)> lift:(1.08) lev:(0.03) [143] conv:(1.17) 6. milk-cream=t 2939 ==> vegetables=t 2025 <conf:(0.69)> lift:(1.08) lev:(0.03) [144] conv:(1.16) 7. fruit=t 2962 ==> milk-cream=t 2038 <conf:(0.69)> lift:(1.08) lev:(0.03) [156] conv:(1.17) 8. frozen foods=t 2717 ==> fruit=t 1861 <conf:(0.68)> lift:(1.07) lev:(0.03) [121] conv:(1.14) 9. vegetables=t 2961 ==> milk-cream=t 2025 <conf:(0.68)> lift:(1.08) lev:(0.03) [144] conv:(1.15) 10. baking needs=t 2795 ==> milk-cream=t 1907 <conf:(0.68)> lift:(1.07) lev:(0.03) [131] conv:(1.15)
0.3	0.2	12	2	1. vegetables=t 2961 ==> fruit=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 2. fruit=t 2962 ==> vegetables=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 3. baking needs=t 2795 ==> vegetables=t 1949 <conf:(0.7)> lift:(1.09) lev:(0.03) [160] conv:(1.19) 4. milk-cream=t 2939 ==> fruit=t 2038 <conf:(0.69)> lift:(1.08) lev:(0.03) [156] conv:(1.17) 5. frozen foods=t 2717 ==> vegetables=t 1882 <conf:(0.69)> lift:(1.08) lev:(0.03) [143] conv:(1.17) 6. milk-cream=t 2939 ==> vegetables=t 2025 <conf:(0.69)> lift:(1.08) lev:(0.03) [144] conv:(1.16) 7. fruit=t 2962 ==> milk-cream=t 2038 <conf:(0.69)> lift:(1.08) lev:(0.03) [156] conv:(1.17) 8. frozen foods=t 2717 ==> fruit=t 1861 <conf:(0.68)> lift:(1.07) lev:(0.03) [121] conv:(1.14) 9. vegetables=t 2961 ==> milk-cream=t 2025 <conf:(0.68)> lift:(1.08) lev:(0.03) [144] conv:(1.15) 10. baking needs=t 2795 ==> milk-cream=t 1907 <conf:(0.68)> lift:(1.07) lev:(0.03) [131] conv:(1.15)
Removed 'bread and cake' lowerBoundMinSupport : 0.5 and High, Medium, Low Confidence				
0.5	0.8			No large itemsets and rules found!
0.5	0.5			No large itemsets and rules found!

0.5	0.2			No large itemsets and rules found!
Removed 'bread and cake' lowerBoundMinSupport : 0.8 and High, Medium, Low Confidence				
0.8	0.8			No large itemsets and rules found!
0.8	0.5			No large itemsets and rules found!
0.8	0.2			No large itemsets and rules found!
Removed 'baking needs' along with 'bread and cake' lowerBoundMinSupport : 0.1 and High, Medium, Low Confidence				
0.1	0.9	18	6	1. milk-cream=t beef=t fruit=t total=high 512 ==> vegetables=t 464 <conf:(0.91)> lift:(1.42) lev:(0.03) [136] conv:(3.76) 2. biscuits=t beef=t fruit=t total=high 514 ==> vegetables=t 465 <conf:(0.9)> lift:(1.41) lev:(0.03) [136] conv:(3.7) 3. frozen foods=t beef=t fruit=t total=high 543 ==> vegetables=t 491 <conf:(0.9)> lift:(1.41) lev:(0.03) [143] conv:(3.69)
0.1	0.8	17	5	1. margarine=t fruit=t total=high 818 ==> vegetables=t 711 <conf:(0.87)> lift:(1.36) lev:(0.04) [187] conv:(2.73) 2. sauces-gravy-pkle=t fruit=t total=high 848 ==> vegetables=t 733 <conf:(0.86)> lift:(1.35) lev:(0.04) [190] conv:(2.63) 3. frozen foods=t fruit=t total=high 969 ==> vegetables=t 834 <conf:(0.86)> lift:(1.34) lev:(0.05) [213] conv:(2.57) 4. tissues-paper prd=t fruit=t total=high 878 ==> vegetables=t 754 <conf:(0.86)> lift:(1.34) lev:(0.04) [192] conv:(2.53) 5. biscuits=t vegetables=t total=high 950 ==> fruit=t 815 <conf:(0.86)> lift:(1.34) lev:(0.04) [206] conv:(2.51) 6. juice-sat-cord-ms=t fruit=t total=high 855 ==> vegetables=t 731 <conf:(0.85)> lift:(1.34) lev:(0.04) [183] conv:(2.46) 7. biscuits=t fruit=t total=high 954 ==> vegetables=t 815 <conf:(0.85)> lift:(1.33) lev:(0.04) [204] conv:(2.45) 8. milk-cream=t beef=t fruit=t 839 ==> vegetables=t 716 <conf:(0.85)> lift:(1.33) lev:(0.04) [179] conv:(2.44) 9. frozen foods=t beef=t fruit=t 842 ==> vegetables=t 717 <conf:(0.85)> lift:(1.33) lev:(0.04) [178] conv:(2.41) 10. milk-cream=t fruit=t total=high 951 ==> vegetables=t 808 <conf:(0.85)> lift:(1.33) lev:(0.04) [199] conv:(2.38)
0.1	0.5	12	2	1. vegetables=t 2961 ==> fruit=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 2. fruit=t 2962 ==> vegetables=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41)

				3. milk-cream=t 2939 ==> fruit=t 2038 <conf:(0.69)> lift:(1.08) lev:(0.03) [156] conv:(1.17) 4. frozen foods=t 2717 ==> vegetables=t 1882 <conf:(0.69)> lift:(1.08) lev:(0.03) [143] conv:(1.17) 5. milk-cream=t 2939 ==> vegetables=t 2025 <conf:(0.69)> lift:(1.08) lev:(0.03) [144] conv:(1.16) 6. fruit=t 2962 ==> milk-cream=t 2038 <conf:(0.69)> lift:(1.08) lev:(0.03) [156] conv:(1.17) 7. frozen foods=t 2717 ==> fruit=t 1861 <conf:(0.68)> lift:(1.07) lev:(0.03) [121] conv:(1.14) 8. vegetables=t 2961 ==> milk-cream=t 2025 <conf:(0.68)> lift:(1.08) lev:(0.03) [144] conv:(1.15) 9. vegetables=t 2961 ==> frozen foods=t 1882 <conf:(0.64)> lift:(1.08) lev:(0.03) [143] conv:(1.13) 10. fruit=t 2962 ==> frozen foods=t 1861 <conf:(0.63)> lift:(1.07) lev:(0.03) [121] conv:(1.11)
0.1	0.2	12	2	1. vegetables=t 2961 ==> fruit=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 2. fruit=t 2962 ==> vegetables=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 3. milk-cream=t 2939 ==> fruit=t 2038 <conf:(0.69)> lift:(1.08) lev:(0.03) [156] conv:(1.17) 4. frozen foods=t 2717 ==> vegetables=t 1882 <conf:(0.69)> lift:(1.08) lev:(0.03) [143] conv:(1.17) 5. milk-cream=t 2939 ==> vegetables=t 2025 <conf:(0.69)> lift:(1.08) lev:(0.03) [144] conv:(1.16) 6. fruit=t 2962 ==> milk-cream=t 2038 <conf:(0.69)> lift:(1.08) lev:(0.03) [156] conv:(1.17) 7. frozen foods=t 2717 ==> fruit=t 1861 <conf:(0.68)> lift:(1.07) lev:(0.03) [121] conv:(1.14) 8. vegetables=t 2961 ==> milk-cream=t 2025 <conf:(0.68)> lift:(1.08) lev:(0.03) [144] conv:(1.15) 9. vegetables=t 2961 ==> frozen foods=t 1882 <conf:(0.64)> lift:(1.08) lev:(0.03) [143] conv:(1.13) 10. fruit=t 2962 ==> frozen foods=t 1861 <conf:(0.63)> lift:(1.07) lev:(0.03) [121] conv:(1.11)
Removed ‘baking needs’ along with ‘bread and cake’ lowerBoundMinSupport : 0.3 and High, Medium, Low Confidence				
0.3	0.9			No best rules were achieved
0.3	0.6	12	2	1. vegetables=t 2961 ==> fruit=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 2. fruit=t 2962 ==> vegetables=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 3. milk-cream=t 2939 ==> fruit=t 2038 <conf:(0.69)> lift:(1.08) lev:(0.03) [156] conv:(1.17) 4. frozen foods=t 2717 ==> vegetables=t 1882 <conf:(0.69)> lift:(1.08) lev:(0.03) [143] conv:(1.17) 5. milk-cream=t 2939 ==> vegetables=t 2025 <conf:(0.69)> lift:(1.08) lev:(0.03) [144] conv:(1.16) 6. fruit=t 2962 ==> milk-cream=t 2038 <conf:(0.69)> lift:(1.08) lev:(0.03) [156] conv:(1.17)

				7. frozen foods=t 2717 ==> fruit=t 1861 <conf:(0.68)> lift:(1.07) lev:(0.03) [121] conv:(1.14) 8. vegetables=t 2961 ==> milk-cream=t 2025 <conf:(0.68)> lift:(1.08) lev:(0.03) [144] conv:(1.15) 9. vegetables=t 2961 ==> frozen foods=t 1882 <conf:(0.64)> lift:(1.08) lev:(0.03) [143] conv:(1.13) 10. fruit=t 2962 ==> frozen foods=t 1861 <conf:(0.63)> lift:(1.07) lev:(0.03) [121] conv:(1.11)
0.3	0.3	12	2	1. vegetables=t 2961 ==> fruit=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 2. fruit=t 2962 ==> vegetables=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 3. milk-cream=t 2939 ==> fruit=t 2038 <conf:(0.69)> lift:(1.08) lev:(0.03) [156] conv:(1.17) 4. frozen foods=t 2717 ==> vegetables=t 1882 <conf:(0.69)> lift:(1.08) lev:(0.03) [143] conv:(1.17) 5. milk-cream=t 2939 ==> vegetables=t 2025 <conf:(0.69)> lift:(1.08) lev:(0.03) [144] conv:(1.16) 6. fruit=t 2962 ==> milk-cream=t 2038 <conf:(0.69)> lift:(1.08) lev:(0.03) [156] conv:(1.17) 7. frozen foods=t 2717 ==> fruit=t 1861 <conf:(0.68)> lift:(1.07) lev:(0.03) [121] conv:(1.14) 8. vegetables=t 2961 ==> milk-cream=t 2025 <conf:(0.68)> lift:(1.08) lev:(0.03) [144] conv:(1.15) 9. vegetables=t 2961 ==> frozen foods=t 1882 <conf:(0.64)> lift:(1.08) lev:(0.03) [143] conv:(1.13) 10. fruit=t 2962 ==> frozen foods=t 1861 <conf:(0.63)> lift:(1.07) lev:(0.03) [121] conv:(1.11)
Removed 'baking needs' along with 'bread and cake' lowerBoundMinSupport : 0.5 and High, Medium, Low Confidence				
0.5	0.8			No large itemsets and rules found
0.5	0.5			No large itemsets and rules found
0.5	0.3			No large itemsets and rules found
Removed 'milk-cream' , 'baking needs' and 'bread and cake' lowerBoundMinSupport : 0.1 and High, Medium, Low Confidence				
0.1	0.9	18	6	1. biscuits=t beef=t fruit=t total=high 514 ==> vegetables=t 465 <conf:(0.9)> lift:(1.41) lev:(0.03) [136] conv:(3.7) 2. frozen foods=t beef=t fruit=t total=high 543 ==> vegetables=t 491 <conf:(0.9)> lift:(1.41) lev:(0.03) [143] conv:(3.69)
0.1	0.5	13	2	1. vegetables=t 2961 ==> fruit=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 2. fruit=t 2962 ==> vegetables=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 3. biscuits=t 2605 ==> fruit=t 1837 <conf:(0.71)> lift:(1.1) lev:(0.04) [169] conv:(1.22) 4. biscuits=t 2605 ==> frozen foods=t 1810 <conf:(0.69)> lift:(1.18) lev:(0.06) [280] conv:(1.35) 5. frozen foods=t 2717 ==> vegetables=t 1882

				<conf:(0.69)> lift:(1.08) lev:(0.03) [143] conv:(1.17) 6. frozen foods=t 2717 ==> fruit=t 1861 <conf:(0.68)> lift:(1.07) lev:(0.03) [121] conv:(1.14) 7. juice-sat-cord-ms=t 2463 ==> fruit=t 1672 <conf:(0.68)> lift:(1.06) lev:(0.02) [95] conv:(1.12) 8. biscuits=t 2605 ==> vegetables=t 1764 <conf:(0.68)> lift:(1.06) lev:(0.02) [96] conv:(1.11) 9. juice-sat-cord-ms=t 2463 ==> vegetables=t 1658 <conf:(0.67)> lift:(1.05) lev:(0.02) [81] conv:(1.1) 10. frozen foods=t 2717 ==> biscuits=t 1810 <conf:(0.67)> lift:(1.18) lev:(0.06) [280] conv:(1.31)
0.1	0.3	13	2	1. vegetables=t 2961 ==> fruit=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 2. fruit=t 2962 ==> vegetables=t 2207 <conf:(0.75)> lift:(1.16) lev:(0.07) [311] conv:(1.41) 3. biscuits=t 2605 ==> fruit=t 1837 <conf:(0.71)> lift:(1.1) lev:(0.04) [169] conv:(1.22) 4. biscuits=t 2605 ==> frozen foods=t 1810 <conf:(0.69)> lift:(1.18) lev:(0.06) [280] conv:(1.35) 5. frozen foods=t 2717 ==> vegetables=t 1882 <conf:(0.69)> lift:(1.08) lev:(0.03) [143] conv:(1.17) 6. frozen foods=t 2717 ==> fruit=t 1861 <conf:(0.68)> lift:(1.07) lev:(0.03) [121] conv:(1.14) 7. juice-sat-cord-ms=t 2463 ==> fruit=t 1672 <conf:(0.68)> lift:(1.06) lev:(0.02) [95] conv:(1.12) 8. biscuits=t 2605 ==> vegetables=t 1764 <conf:(0.68)> lift:(1.06) lev:(0.02) [96] conv:(1.11) 9. juice-sat-cord-ms=t 2463 ==> vegetables=t 1658 <conf:(0.67)> lift:(1.05) lev:(0.02) [81] conv:(1.1) 10. frozen foods=t 2717 ==> biscuits=t 1810 <conf:(0.67)> lift:(1.18) lev:(0.06) [280] conv:(1.31)
Removed ‘milk-cream’ , ‘baking needs’ and ‘bread and cake’ lowerBoundMinSupport : 0.5 and High, Medium, Low Confidence				
0.5	0.8			No large itemsets and rules found!
0.5	0.5			No large itemsets and rules found!
0.5	0.3			No large itemsets and rules found!
Removed ‘milk-cream’ , ‘baking needs’ and ‘bread and cake’ lowerBoundMinSupport : 0.8 and High, Medium, Low Confidence				
0.8	0.8			No large itemsets and rules found!
0.8	0.5			No large itemsets and rules found!
0.8	0.3			No large itemsets and rules found!
Removed ‘vegetables’, ’biscuits’, ‘frozenfoods’, ‘milk-cream’ , ‘baking needs’ and ‘bread and cake’ lowerBoundMinSupport : 0.1 and High, Medium, Low Confidence				
0.1	0.8	18	5	1. laundry needs=t margarine=t total=high 613 ==> tissues-paper prd=t 507 <conf:(0.83)> lift:(1.7) lev:(0.05) [209] conv:(2.95) 2. juice-sat-cord-ms=t laundry needs=t total=high 603 ==> tissues-paper prd=t 489 <conf:(0.81)> lift:(1.67) lev:(0.04) [196] conv:(2.7) 3. sauces-gravy-pkle=t laundry needs=t total=high 621 ==> tissues-paper prd=t 501 <conf:(0.81)> lift:(1.66) lev:(0.04) [199] conv:(2.64)

				4. party snack foods=t dairy foods=t total=high 597 ==> fruit=t 481 <conf:(0.81)> lift:(1.26) lev:(0.02) [98] conv:(1.84)
0.1	0.5	14	2	1. tissues-paper prd=t 2247 ==> fruit=t 1567 <conf:(0.7)> lift:(1.09) lev:(0.03) [128] conv:(1.19) 2. party snack foods=t 2330 ==> fruit=t 1592 <conf:(0.68)> lift:(1.07) lev:(0.02) [100] conv:(1.13) 3. juice-sat-cord-ms=t 2463 ==> fruit=t 1672 <conf:(0.68)> lift:(1.06) lev:(0.02) [95] conv:(1.12) 4. sauces-gravy-pkle=t 2201 ==> fruit=t 1490 <conf:(0.68)> lift:(1.06) lev:(0.02) [81] conv:(1.11) 5. margarine=t 2288 ==> fruit=t 1538 <conf:(0.67)> lift:(1.05) lev:(0.02) [73] conv:(1.1) 6. party snack foods=t 2330 ==> juice-sat-cord-ms=t 1482 <conf:(0.64)> lift:(1.19) lev:(0.05) [241] conv:(1.28) 7. juice-sat-cord-ms=t 2463 ==> party snack foods=t 1482 <conf:(0.6)> lift:(1.19) lev:(0.05) [241] conv:(1.25) 8. total=low 2948 ==> fruit=t 1719 <conf:(0.58)> lift:(0.91) lev:(-0.04) [-168] conv:(0.86) 9. fruit=t 2962 ==> total=low 1719 <conf:(0.58)> lift:(0.91) lev:(-0.04) [-168] conv:(0.86) 10. fruit=t 2962 ==> juice-sat-cord-ms=t 1672 <conf:(0.56)> lift:(1.06) lev:(0.02) [95] conv:(1.07)
0.1	0.3	14	2	1. tissues-paper prd=t 2247 ==> fruit=t 1567 <conf:(0.7)> lift:(1.09) lev:(0.03) [128] conv:(1.19) 2. party snack foods=t 2330 ==> fruit=t 1592 <conf:(0.68)> lift:(1.07) lev:(0.02) [100] conv:(1.13) 3. juice-sat-cord-ms=t 2463 ==> fruit=t 1672 <conf:(0.68)> lift:(1.06) lev:(0.02) [95] conv:(1.12) 4. sauces-gravy-pkle=t 2201 ==> fruit=t 1490 <conf:(0.68)> lift:(1.06) lev:(0.02) [81] conv:(1.11) 5. margarine=t 2288 ==> fruit=t 1538 <conf:(0.67)> lift:(1.05) lev:(0.02) [73] conv:(1.1) 6. party snack foods=t 2330 ==> juice-sat-cord-ms=t 1482 <conf:(0.64)> lift:(1.19) lev:(0.05) [241] conv:(1.28) 7. juice-sat-cord-ms=t 2463 ==> party snack foods=t 1482 <conf:(0.6)> lift:(1.19) lev:(0.05) [241] conv:(1.25) 8. total=low 2948 ==> fruit=t 1719 <conf:(0.58)> lift:(0.91) lev:(-0.04) [-168] conv:(0.86) 9. fruit=t 2962 ==> total=low 1719 <conf:(0.58)> lift:(0.91) lev:(-0.04) [-168] conv:(0.86) 10. fruit=t 2962 ==> juice-sat-cord-ms=t 1672 <conf:(0.56)> lift:(1.06) lev:(0.02) [95] conv:(1.07)
Removed ‘vegetables’, ’biscuits’, ‘frozenfoods’, ‘milk-cream’ , ‘baking needs’ and ‘bread and cake’ lowerBoundMinSupport : 0.5 and High, Medium, Low Confidence				
0.5	0.8			No large itemsets and rules found!

0.5	0.5			No large itemsets and rules found!
0.5	0.3			No large itemsets and rules found!
Removed 'vegetables', 'biscuits', 'frozenfoods', 'milk-cream' , 'baking needs' and 'bread and cake' lowerBoundMinSupport : 0.3 and High, Medium, Low Confidence				
0.8	0.8			No large itemsets and rules found!
0.8	0.5			No large itemsets and rules found!
0.8	0.3			No large itemsets and rules found!

Inferences on association rule generated by removing some attributes and modifying support and confidence values

- When support and confidence are set to default values 0.1 and 0.9, 10 association rules were generated and it was based on frequent items. It is clearly seen from the rules. These rules are not interesting because there are more number of chances that the rules will be affected if the frequent items and itemsets that are associated with them are removed.

- When the most frequent item bread and cake was removed and default values for support and confidence are set only 4 association rules were generated and the cycles count was 18 and large itemset was 6.This clearly shows how removing frequent item from data affects the association rules.

- When the confidence value was set to 0.8, 10 association rules were generated and new association rules were generated. When the confidence value was decreased to 0.5 and 0.2 same set of association rules were generated but different from the above rules. The cycles and large itemset was also reduced. When support values are increase to 0.5 and more no best rules were fount and no large itemsets.

- When 'baking need' the next frequent item was removed along with 'bread and cake' and with default values only four rules were generated. When confidence value is det to 0.8 different set of rules were generated and decreasing the confidence and increasing support again produces some association rules which were produced earlier. When support is increased more, no more rules were generated nor large itemsets.

- When 'milk – cream' was removed along with 'baking needs' and 'bread and cake' only four rules were generated and altering suppot and confidence values altered the rules and the cycle count and large itemset.

- These analysis was done by removing the other frequent items like vegetables, biscuits and frozen foods and some interesting association rules were generated. When support and confidence are increased no large itemset or rules were found.

- Eg: vegetables=t 2961 ==> fruit=t 2207 , frozen foods=t 2717 ==> fruit=t 1861. These kind of rules are more interesting than with the frequent items. The absence of total attribute provides more meaningful rules. As the number of frequent items are removed the number of best rules found was decreasing. On removal of items there was no are very little change in the rules at some point.

1.3 Use vote dataset: Select all attributes in the attribute selection tab. Note that the class label will be used as one of the attributes or "items" and each record is a "market basket". Compute the association rules with the default parameter settings. Look at the right hand size of the top rules, compare the attributes in the association rules to the most important attributes that you computed for the decision tree classifier in HW 2.

Vote Dataset:

The Vote dataset has vote details of the Congressmen in the U.S House of representatives. It has 435 instances. There are 16 attributes and a class attribute which is 17 in total. All the attributes are of type Nominal.

lowerBoundMinSupport (Support)	minMetric (Confidence)	Cycle count	Large itemset levels	Top 10 association rules
Default lowerBoundMinSupport : 0.1 and Confidence 0.9				
0.1	0.9	11	4	1. adoption-of-the-budget-resolution=y physician-fee-freeze=n 219 ==> Class=democrat 219 <conf:(1)> lift:(1.63) lev:(0.19) [84] conv:(84.58) 2. adoption-of-the-budget-resolution=y physician-fee-freeze=n aid-to-nicaraguan-contras=y 198 ==> Class=democrat 198 <conf:(1)> lift:(1.63) lev:(0.18) [76] conv:(76.47) 3. physician-fee-freeze=n aid-to-nicaraguan-contras=y 211 ==> Class=democrat 210 <conf:(1)> lift:(1.62) lev:(0.19) [80] conv:(40.74) 4. physician-fee-freeze=n education-spending=n 202 ==> Class=democrat 201 <conf:(1)> lift:(1.62) lev:(0.18) [77] conv:(39.01) 5. physician-fee-freeze=n 247 ==> Class=democrat 245 <conf:(0.99)> lift:(1.62) lev:(0.21) [93] conv:(31.8) 6. el-salvador-aid=n Class=democrat 200 ==> aid-to-nicaraguan-contras=y 197 <conf:(0.98)> lift:(1.77) lev:(0.2) [85] conv:(22.18) 7. el-salvador-aid=n 208 ==> aid-to-nicaraguan-contras=y 204 <conf:(0.98)> lift:(1.76) lev:(0.2) [88] conv:(18.46) 8. adoption-of-the-budget-resolution=y aid-to-nicaraguan-contras=y Class=democrat 203 ==> physician-fee-freeze=n 198 <conf:(0.98)> lift:(1.72) lev:(0.19) [82] conv:(14.62) 9. el-salvador-aid=n aid-to-nicaraguan-contras=y 204 ==> Class=democrat 197 <conf:(0.97)> lift:(1.57) lev:(0.17) [71] conv:(9.85) 10. aid-to-nicaraguan-contras=y Class=democrat 218 ==> physician-fee-freeze=n 210 <conf:(0.96)> lift:(1.7) lev:(0.2) [86] conv:(10.47)
High, Medium and Low lowerBoundMinSupport and High, Medium, Low Confidence				
0.1	0.5	10	3	1. adoption-of-the-budget-resolution=y physician-fee-

				freeze=n 219 ==> Class=democrat 219 <conf:(1)> lift:(1.63) lev:(0.19) [84] conv:(84.58) 2. physician-fee-freeze=n 247 ==> Class=democrat 245 <conf:(0.99)> lift:(1.62) lev:(0.21) [93] conv:(31.8) 3. adoption-of-the-budget-resolution=y Class=democrat 231 ==> physician-fee-freeze=n 219 <conf:(0.95)> lift:(1.67) lev:(0.2) [87] conv:(7.68) 4. Class=democrat 267 ==> physician-fee-freeze=n 245 <conf:(0.92)> lift:(1.62) lev:(0.21) [93] conv:(5.02) 5. adoption-of-the-budget-resolution=y 253 ==> Class=democrat 231 <conf:(0.91)> lift:(1.49) lev:(0.17) [75] conv:(4.25) 6. aid-to-nicaraguan-contras=y 242 ==> Class=democrat 218 <conf:(0.9)> lift:(1.47) lev:(0.16) [69] conv:(3.74) 7. physician-fee-freeze=n Class=democrat 245 ==> adoption-of-the-budget-resolution=y 219 <conf:(0.89)> lift:(1.54) lev:(0.18) [76] conv:(3.8) 8. physician-fee-freeze=n 247 ==> adoption-of-the-budget-resolution=y 219 <conf:(0.89)> lift:(1.52) lev:(0.17) [75] conv:(3.56) 9. physician-fee-freeze=n 247 ==> adoption-of-the-budget-resolution=y Class=democrat 219 <conf:(0.89)> lift:(1.67) lev:(0.2) [87] conv:(3.99) 10. adoption-of-the-budget-resolution=y 253 ==> physician-fee-freeze=n 219 <conf:(0.87)> lift:(1.52) lev:(0.17) [75] conv:(3.12)
0.1	0.3	10	3	1. adoption-of-the-budget-resolution=y physician-fee-freeze=n 219 ==> Class=democrat 219 <conf:(1)> lift:(1.63) lev:(0.19) [84] conv:(84.58) 2. physician-fee-freeze=n 247 ==> Class=democrat 245 <conf:(0.99)> lift:(1.62) lev:(0.21) [93] conv:(31.8) 3. adoption-of-the-budget-resolution=y Class=democrat 231 ==> physician-fee-freeze=n 219 <conf:(0.95)> lift:(1.67) lev:(0.2) [87] conv:(7.68) 4. Class=democrat 267 ==> physician-fee-freeze=n 245 <conf:(0.92)> lift:(1.62) lev:(0.21) [93] conv:(5.02) 5. adoption-of-the-budget-resolution=y 253 ==> Class=democrat 231 <conf:(0.91)> lift:(1.49) lev:(0.17) [75] conv:(4.25) 6. aid-to-nicaraguan-contras=y 242 ==> Class=democrat 218 <conf:(0.9)> lift:(1.47) lev:(0.16) [69] conv:(3.74) 7. physician-fee-freeze=n Class=democrat 245 ==> adoption-of-the-budget-resolution=y 219 <conf:(0.89)> lift:(1.54) lev:(0.18) [76] conv:(3.8) 8. physician-fee-freeze=n 247 ==> adoption-of-the-budget-resolution=y 219 <conf:(0.89)> lift:(1.52) lev:(0.17) [75] conv:(3.56) 9. physician-fee-freeze=n 247 ==> adoption-of-the-budget-resolution=y Class=democrat 219 <conf:(0.89)> lift:(1.67) lev:(0.2) [87] conv:(3.99) 10. adoption-of-the-budget-resolution=y 253 ==> physician-fee-freeze=n 219 <conf:(0.87)> lift:(1.52) lev:(0.17) [75] conv:(3.12)

0.2	0.8	10	3	1. adoption-of-the-budget-resolution=y physician-fee-freeze=n 219 ==> Class=democrat 219 <conf:(1)> lift:(1.63) lev:(0.19) [84] conv:(84.58) 2. physician-fee-freeze=n 247 ==> Class=democrat 245 <conf:(0.99)> lift:(1.62) lev:(0.21) [93] conv:(31.8) 3. adoption-of-the-budget-resolution=y Class=democrat 231 ==> physician-fee-freeze=n 219 <conf:(0.95)> lift:(1.67) lev:(0.2) [87] conv:(7.68) 4. Class=democrat 267 ==> physician-fee-freeze=n 245 <conf:(0.92)> lift:(1.62) lev:(0.21) [93] conv:(5.02) 5. adoption-of-the-budget-resolution=y 253 ==> Class=democrat 231 <conf:(0.91)> lift:(1.49) lev:(0.17) [75] conv:(4.25) 6. aid-to-nicaraguan-contras=y 242 ==> Class=democrat 218 <conf:(0.9)> lift:(1.47) lev:(0.16) [69] conv:(3.74) 7. physician-fee-freeze=n Class=democrat 245 ==> adoption-of-the-budget-resolution=y 219 <conf:(0.89)> lift:(1.54) lev:(0.18) [76] conv:(3.8) 8. physician-fee-freeze=n 247 ==> adoption-of-the-budget-resolution=y 219 <conf:(0.89)> lift:(1.52) lev:(0.17) [75] conv:(3.56) 9. physician-fee-freeze=n 247 ==> adoption-of-the-budget-resolution=y Class=democrat 219 <conf:(0.89)> lift:(1.67) lev:(0.2) [87] conv:(3.99) 10. adoption-of-the-budget-resolution=y 253 ==> physician-fee-freeze=n 219 <conf:(0.87)> lift:(1.52) lev:(0.17) [75] conv:(3.12)
0.2	0.5	10	3	1. adoption-of-the-budget-resolution=y physician-fee-freeze=n 219 ==> Class=democrat 219 <conf:(1)> lift:(1.63) lev:(0.19) [84] conv:(84.58) 2. physician-fee-freeze=n 247 ==> Class=democrat 245 <conf:(0.99)> lift:(1.62) lev:(0.21) [93] conv:(31.8) 3. adoption-of-the-budget-resolution=y Class=democrat 231 ==> physician-fee-freeze=n 219 <conf:(0.95)> lift:(1.67) lev:(0.2) [87] conv:(7.68) 4. Class=democrat 267 ==> physician-fee-freeze=n 245 <conf:(0.92)> lift:(1.62) lev:(0.21) [93] conv:(5.02) 5. adoption-of-the-budget-resolution=y 253 ==> Class=democrat 231 <conf:(0.91)> lift:(1.49) lev:(0.17) [75] conv:(4.25) 6. aid-to-nicaraguan-contras=y 242 ==> Class=democrat 218 <conf:(0.9)> lift:(1.47) lev:(0.16) [69] conv:(3.74) 7. physician-fee-freeze=n Class=democrat 245 ==> adoption-of-the-budget-resolution=y 219 <conf:(0.89)> lift:(1.54) lev:(0.18) [76] conv:(3.8) 8. physician-fee-freeze=n 247 ==> adoption-of-the-budget-resolution=y 219 <conf:(0.89)> lift:(1.52) lev:(0.17) [75] conv:(3.56) 9. physician-fee-freeze=n 247 ==> adoption-of-the-budget-resolution=y Class=democrat 219 <conf:(0.89)> lift:(1.67) lev:(0.2) [87] conv:(3.99)

				10. adoption-of-the-budget-resolution=y 253 ==> physician-fee-freeze=n 219 <conf:(0.87)> lift:(1.52) lev:(0.17) [75] conv:(3.12)
0.5	0.8	10	3	1. adoption-of-the-budget-resolution=y physician-fee-freeze=n 219 ==> Class=democrat 219 <conf:(1)> lift:(1.63) lev:(0.19) [84] conv:(84.58) 2. physician-fee-freeze=n 247 ==> Class=democrat 245 <conf:(0.99)> lift:(1.62) lev:(0.21) [93] conv:(31.8) 3. adoption-of-the-budget-resolution=y Class=democrat 231 ==> physician-fee-freeze=n 219 <conf:(0.95)> lift:(1.67) lev:(0.2) [87] conv:(7.68) 4. Class=democrat 267 ==> physician-fee-freeze=n 245 <conf:(0.92)> lift:(1.62) lev:(0.21) [93] conv:(5.02) 5. adoption-of-the-budget-resolution=y 253 ==> Class=democrat 231 <conf:(0.91)> lift:(1.49) lev:(0.17) [75] conv:(4.25) 6. aid-to-nicaraguan-contras=y 242 ==> Class=democrat 218 <conf:(0.9)> lift:(1.47) lev:(0.16) [69] conv:(3.74) 7. physician-fee-freeze=n Class=democrat 245 ==> adoption-of-the-budget-resolution=y 219 <conf:(0.89)> lift:(1.54) lev:(0.18) [76] conv:(3.8) 8. physician-fee-freeze=n 247 ==> adoption-of-the-budget-resolution=y 219 <conf:(0.89)> lift:(1.52) lev:(0.17) [75] conv:(3.56) 9. physician-fee-freeze=n 247 ==> adoption-of-the-budget-resolution=y Class=democrat 219 <conf:(0.89)> lift:(1.67) lev:(0.2) [87] conv:(3.99) 10. adoption-of-the-budget-resolution=y 253 ==> physician-fee-freeze=n 219 <conf:(0.87)> lift:(1.52) lev:(0.17) [75] conv:(3.12)
0.5	0.3	10	3	1. adoption-of-the-budget-resolution=y physician-fee-freeze=n 219 ==> Class=democrat 219 <conf:(1)> lift:(1.63) lev:(0.19) [84] conv:(84.58) 2. physician-fee-freeze=n 247 ==> Class=democrat 245 <conf:(0.99)> lift:(1.62) lev:(0.21) [93] conv:(31.8) 3. adoption-of-the-budget-resolution=y Class=democrat 231 ==> physician-fee-freeze=n 219 <conf:(0.95)> lift:(1.67) lev:(0.2) [87] conv:(7.68) 4. Class=democrat 267 ==> physician-fee-freeze=n 245 <conf:(0.92)> lift:(1.62) lev:(0.21) [93] conv:(5.02) 5. adoption-of-the-budget-resolution=y 253 ==> Class=democrat 231 <conf:(0.91)> lift:(1.49) lev:(0.17) [75] conv:(4.25) 6. aid-to-nicaraguan-contras=y 242 ==> Class=democrat 218 <conf:(0.9)> lift:(1.47) lev:(0.16) [69] conv:(3.74) 7. physician-fee-freeze=n Class=democrat 245 ==> adoption-of-the-budget-resolution=y 219 <conf:(0.89)> lift:(1.54) lev:(0.18) [76] conv:(3.8) 8. physician-fee-freeze=n 247 ==> adoption-of-the-budget-resolution=y 219 <conf:(0.89)> lift:(1.52) lev:(0.17) [75] conv:(3.56)

				9. physician-fee-freeze=n 247 ==> adoption-of-the-budget-resolution=y Class=democrat 219 <conf:(0.89)> lift:(1.67) lev:(0.2) [87] conv:(3.99) 10. adoption-of-the-budget-resolution=y 253 ==> physician-fee-freeze=n 219 <conf:(0.87)> lift:(1.52) lev:(0.17) [75] conv:(3.12)
0.8	0.8			No large itemsets and rules found!
0.8	0.5			No large itemsets and rules found!
0.8	0.3			No large itemsets and rules found!
Removed 'adoption of the budget resolution'				
0.1	0.9	11	3	1. physician-fee-freeze=n aid-to-nicaraguan-contras=y 211 ==> Class=democrat 210 <conf:(1)> lift:(1.62) lev:(0.19) [80] conv:(40.74) 2. physician-fee-freeze=n education-spending=n 202 ==> Class=democrat 201 <conf:(1)> lift:(1.62) lev:(0.18) [77] conv:(39.01) 3. physician-fee-freeze=n 247 ==> Class=democrat 245 <conf:(0.99)> lift:(1.62) lev:(0.21) [93] conv:(31.8) 4. el-salvador-aid=n Class=democrat 200 ==> aid-to-nicaraguan-contras=y 197 <conf:(0.98)> lift:(1.77) lev:(0.2) [85] conv:(22.18) 5. el-salvador-aid=n 208 ==> aid-to-nicaraguan-contras=y 204 <conf:(0.98)> lift:(1.76) lev:(0.2) [88] conv:(18.46) 6. el-salvador-aid=n aid-to-nicaraguan-contras=y 204 ==> Class=democrat 197 <conf:(0.97)> lift:(1.57) lev:(0.17) [71] conv:(9.85) 7. aid-to-nicaraguan-contras=y Class=democrat 218 ==> physician-fee-freeze=n 210 <conf:(0.96)> lift:(1.7) lev:(0.2) [86] conv:(10.47) 8. el-salvador-aid=n 208 ==> Class=democrat 200 <conf:(0.96)> lift:(1.57) lev:(0.17) [72] conv:(8.93) 9. el-salvador-aid=n 208 ==> aid-to-nicaraguan-contras=y Class=democrat 197 <conf:(0.95)> lift:(1.89) lev:(0.21) [92] conv:(8.65) 10. education-spending=n Class=democrat 213 ==> physician-fee-freeze=n 201 <conf:(0.94)> lift:(1.66) lev:(0.18) [80] conv:(7.08)
Removed 'adoption of the budget resolution' & 'Physician fee freeze'				
0.1	0.9	12	4	1. el-salvador-aid=n Class=democrat 200 ==> aid-to-nicaraguan-contras=y 197 <conf:(0.98)> lift:(1.77) lev:(0.2) [85] conv:(22.18) 2. el-salvador-aid=n anti-satellite-test-ban=y Class=democrat 182 ==> aid-to-nicaraguan-contras=y 179 <conf:(0.98)> lift:(1.77) lev:(0.18) [77] conv:(20.19) 3. el-salvador-aid=n education-spending=n 177 ==> aid-to-nicaraguan-contras=y 174 <conf:(0.98)> lift:(1.77)

				lev:(0.17) [75] conv:(19.63) 4. el-salvador-aid=n 208 ==> aid-to-nicaraguan-contras=y 204 <conf:(0.98)> lift:(1.76) lev:(0.2) [88] conv:(18.46) 5. el-salvador-aid=n anti-satellite-test-ban=y 188 ==> aid-to-nicaraguan-contras=y 184 <conf:(0.98)> lift:(1.76) lev:(0.18) [79] conv:(16.68) 6. el-salvador-aid=n mx-missile=y 179 ==> aid-to-nicaraguan-contras=y 175 <conf:(0.98)> lift:(1.76) lev:(0.17) [75] conv:(15.88) 7. el-salvador-aid=n anti-satellite-test-ban=y aid-to-nicaraguan-contras=y 184 ==> Class=democrat 179 <conf:(0.97)> lift:(1.58) lev:(0.15) [66] conv:(11.84) 8. el-salvador-aid=n anti-satellite-test-ban=y 188 ==> Class=democrat 182 <conf:(0.97)> lift:(1.58) lev:(0.15) [66] conv:(10.37) 9. el-salvador-aid=n aid-to-nicaraguan-contras=y 204 ==> Class=democrat 197 <conf:(0.97)> lift:(1.57) lev:(0.17) [71] conv:(9.85) 10. el-salvador-aid=n 208 ==> Class=democrat 200 <conf:(0.96)> lift:(1.57) lev:(0.17) [72] conv:(8.93)
Removed ‘adoption of the budget resolution’ , ‘Physician fee freeze’ ,‘el-salvator aid’				
0.1	0.9	12	3	1. aid-to-nicaraguan-contras=y education-spending=n 194 ==> Class=democrat 186 <conf:(0.96)> lift:(1.56) lev:(0.15) [66] conv:(8.32) 2. mx-missile=y Class=democrat 188 ==> aid-to-nicaraguan-contras=y 179 <conf:(0.95)> lift:(1.71) lev:(0.17) [74] conv:(8.34) 3. anti-satellite-test-ban=y Class=democrat 200 ==> aid-to-nicaraguan-contras=y 189 <conf:(0.94)> lift:(1.7) lev:(0.18) [77] conv:(7.39) 4. aid-to-nicaraguan-contras=y mx-missile=y 192 ==> Class=democrat 179 <conf:(0.93)> lift:(1.52) lev:(0.14) [61] conv:(5.3) 5. mx-missile=y 207 ==> aid-to-nicaraguan-contras=y 192 <conf:(0.93)> lift:(1.67) lev:(0.18) [76] conv:(5.74) 6. education-spending=n 233 ==> Class=democrat 213 <conf:(0.91)> lift:(1.49) lev:(0.16) [69] conv:(4.29) 7. mx-missile=n 206 ==> religious-groups-in-schools=y 188 <conf:(0.91)> lift:(1.46) lev:(0.14) [59] conv:(4.06) 8. mx-missile=y 207 ==> Class=democrat 188 <conf:(0.91)> lift:(1.48) lev:(0.14) [60] conv:(4) 9. aid-to-nicaraguan-contras=y 242 ==> Class=democrat 218 <conf:(0.9)> lift:(1.47) lev:(0.16) [69] conv:(3.74) 10. anti-satellite-test-ban=y aid-to-nicaraguan-contras=y 210 ==> Class=democrat 189 <conf:(0.9)> lift:(1.47) lev:(0.14) [60] conv:(3.69)

Inferences on association rule generated by modifying support and confidence values and by removing some attributes

- When support and confidence are set to default values 0.1 and 0.9, 10 association rules were generated and it was based on frequent items. It is clearly seen from the association rules. These rules are not interesting because there are more number of chances that the rules will be affected if the frequent items and itemsets that are associated with them are removed.

- When the support and confidence level was altered the cycles count decreased and large items set decreased and there was slight change in the association rules. When support is set to high values then no best rules were found.

- When some of the frequent items were removed from data then there was change in association rules which are interesting than the above rules.

- When certain support and confidence values were altered, same set of association rules were generated often.

Decision Stump:

This algorithm produces a Decision Stump, which is a variant of decision tree where there is only one level. This implies that the size of a decision tree (the number of nodes in a tree) is always 3 and the internal nodes are absent. There is a single root node, as with all other decision trees, where the attribute with maximal information gain is checked for a condition based on the result of which instances are split into two groups and different class labels are assigned to them.

Decision Stump	10 Cross Validation	Training Set
Correctly Classified Instance	416	416
Incorrectly classified Instance	19	19
Correctly classified Percentage	95.6322	95.6322
Incorrectly classified Percentage	4.3678	4.3678
Relative absolute error	16.693%	16.5385
Root relative squared error	41.2142%	40.6726
Total Number of instances	435	435

J48:

For the given data-set it generates a classification-decision tree by recursive partitioning of data. Using Depth-first strategy the decision is grown. The algorithm considers all the possible tests that can split the data set and selects a test that gives the best information gain. For each discrete attribute, one test with outcomes as many as the number of distinct values of the attribute is considered.

J48	10 Cross Validation	Training Set
Time Taken	0.01sec	0.
Number of Leaves	6	6
Size of Tree	11	11

Correctly Classified Instance	419	423
Incorrectly classified Instance	16	12
Correctly classified Percentage	96.3218	97.2414%
Incorrectly classified Percentage	3.6782%	2.7586%
Relative absolute error	12.887%	10.9481%
Root relative squared error	35.9085%	30.9353%
Total Number of instances	435	435

RANDOM FOREST:

Random Forest is an ensemble of unpruned classification or regression trees, induced from bootstrap samples of the training data, using random feature selection in the tree induction process. Prediction is made by aggregating (majority vote for classification or averaging for regression) the predictions of the ensemble. Random forest generally exhibits a substantial performance improvement over the single tree classifier

Random	10 Cross Validation	Training Set
Time Taken	0.11sec	0.1
Correctly Classified Instance	418	432
Incorrectly classified Instance	17	3
Correctly classified Percentage	96.092%	99.3103%
Incorrectly classified Percentage	3.908%	0.6897%
Relative absolute error	15.0587%	6.9673%
Root relative squared error	35.7776%	17.733%
Total Number of instances	435	435

Top 4 association rules using default parameters.

Main parameters with default value:
lowerBoundMinSupport (support): 0.1
minMetric (confidence): 0.9

1. adoption-of-the-budget-resolution=y physician-fee-freeze=n 219 ==> Class=democrat 219 <conf:(1)> lift:(1.63) lev:(0.19) [84] conv:(84.58)

2. adoption-of-the-budget-resolution=y physician-fee-freeze=n aid-to-nicaraguan-contras=y 198 ==> Class=democrat 198 <conf:(1)> lift:(1.63) lev:(0.18) [76] conv:(76.47)

3. physician-fee-freeze=n aid-to-nicaraguan-contras=y 211 ==> Class=democrat 210 <conf:(1)> lift:(1.62) lev:(0.19) [80] conv:(40.74)

4. physician-fee-freeze=n education-spending=n 202 ==> Class=democrat 201 <conf:(1)> lift:(1.62) lev:(0.18) [77] conv:(39.01)

5. physician-fee-freeze=n 247 ==> Class=democrat 245 <conf:(0.99)> lift:(1.62) lev:(0.21) [93] conv:(31.8)

6. el-salvador-aid=n Class=democrat 200 ==> aid-to-nicaraguan-contras=y 197 <conf:(0.98)> lift:(1.77) lev:(0.2) [85] conv:(22.18)

7. el-salvador-aid=n 208 ==> aid-to-nicaraguan-contras=y 204 <conf:(0.98)> lift:(1.76) lev:(0.2) [88] conv:(18.46)

8. adoption-of-the-budget-resolution=y aid-to-nicaraguan-contras=y Class=democrat 203 ==> physician-fee-freeze=n 198 <conf:(0.98)> lift:(1.72) lev:(0.19) [82] conv:(14.62)

9. el-salvador-aid=n aid-to-nicaraguan-contras=y 204 ==> Class=democrat 197 <conf:(0.97)> lift:(1.57) lev:(0.17) [71] conv:(9.85)

10. aid-to-nicaraguan-contras=y Class=democrat 218 ==> physician-fee-freeze=n 210 <conf:(0.96)> lift:(1.7) lev:(0.2) [86] conv:(10.47)

Number of cycles performed was 11 and large itemset produced was 4

Comparing the attributes in the association rules to the important attributes used for decision tree classifier.

- Let us consider the association rule with default values. The association rules is used to find the antecedent and consequent item relations between the attributes. It does not focus on single class attribute. The classifier tries to map the attributes to a specific class value.

- The top 4 association rules tries to map the attributes to a specific class values.
  Eg: adoption-of-the-budget-resolution=y physician-fee-freeze=n 219 ==> Class=democrat 219. From top four rules, it is clearly shown that the decision tree classifier and the association rules both behave in a similar way. They predict whether the representative will be a democrat or republican based on the vote patterns.

- In case of association rule it usually takes all the items in the dataset and analyze to form the rule based on it. In Decision tree classifiers, it is classified based on one attribute to assign the class value for the attributes. The model is built based on one attribute depending on Information gain, correlation and then the model is used to classify.

- This shows that association rules are good to analyze the data and classifiers are suited for mapping of many values to class attributes. Association analysis does not predict the class alone, it can show the relation between items and how it affects the other. eg: If a person supports el-salvador-aid then the person will most likely support aid-to-nicaraguan-contras

- In all the three classifiers the attribute used to classify the data depends on the correlation ranking and information ranking filter and based on missing class values of the attributes. In general the attributes with high correlation and information gain will be selected by the classifiers.

- In association rules, the rules are based on frequent items. The most frequent attributes physician-fee-freeze, adoption-of-the-budget-resolution and el-salvador-aid are the

attributes which has high correlation and information gain raking and does not consider about the missing values.

- When comparing the most important attributes used in classifier and in association rules, we find that the same attributes are used. This implies that both association rule and the classifiers work in a similar manner except for small differences.

2. Use R
The goal for this part of the assignment is to learn some basics of R.

Follow these commands, after each step describe in a few words what output you get:

> str(Titanic)

```
 table [1:4, 1:2, 1:2, 1:2] 0 0 35 0 0 0 17 0 118 154 ...
- attr(*, "dimnames")=List of 4
  ..$ Class   : chr [1:4] "1st" "2nd" "3rd" "Crew"
  ..$ Sex     : chr [1:2] "Male" "Female"
  ..$ Age     : chr [1:2] "Child" "Adult"
  ..$ Survived: chr [1:2] "No" "Yes"
```

The str function is the most useful function in R. It provides great information about the structure of some object. It displays the internal structure of an R object. The attributes and datatypes of Titanic dataset is displayed.

```
> df <- as.data.frame(Titanic)
> head(df)
  Class    Sex   Age Survived Freq
1   1st   Male Child       No    0
2   2nd   Male Child       No    0
3   3rd   Male Child       No   35
4  Crew   Male Child       No    0
5   1st Female Child       No    0
6   2nd Female Child       No    0
```

The dataset set is converted to dataframe and returns the first part of the dataframe.

```
> titanic.raw <- NULL
> for(i in 1:4) {
+ titanic.raw <- cbind(titanic.raw, rep(as.character(df[,i]), df$Freq))
+ }
> titanic.raw <- as.data.frame(titanic.raw)
> names(titanic.raw) <- names(df)[1:4]
> dim(titanic.raw)
[1] 2201    4
```

The frequent items of the Titanic dataset is calculated and the dimension of it is displayed.

```
> str(titanic.raw)
'data.frame':   2201 obs. of  4 variables:
 $ Class   : Factor w/ 4 levels "1st","2nd","3rd",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ Sex     : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
```

```
 $ Age     : Factor w/ 2 levels "Adult","Child": 2 2 2 2 2 2 2 2 2 2 ...
 $ Survived: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
```
The structure of the dataset after finding the frequency of items is displayed.

```
> head(titanic.raw)
  Class  Sex   Age Survived
1   3rd Male Child       No
2   3rd Male Child       No
3   3rd Male Child       No
4   3rd Male Child       No
5   3rd Male Child       No
6   3rd Male Child       No
```
The first part or several rows of data frame is displayed.

```
> summary(titanic.raw)
  Class         Sex         Age       Survived
 1st :325   Female: 470   Adult:2092   No :1490
 2nd :285   Male  :1731   Child: 109   Yes: 711
 3rd :706
 Crew:885
```
After executing all the functions the summary of the Titanic dataset is displayed.

Use association rule mining with apriori().

```
> library(arules)
Loading required package: Matrix
Attaching package: 'arules'
The following objects are masked from 'package:base':
abbreviate, write
```
The required packages are loaded from arules.

```
>  rules.all <- apriori(titanic.raw)
Apriori
Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen
       0.8        0.1      1     none FALSE          TRUE              5     0.1      1
 maxlen target   ext
    10  rules FALSE

Algorithmic control:
 filter  tree   heap  memopt  load  sort  verbose
   0.1 TRUE TRUE   FALSE    TRUE    2    TRUE

Absolute minimum support count: 220

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[10 item(s), 2201 transaction(s)] done [0.00s].
sorting and recoding items ... [9 item(s)] done [0.00s].
```

```
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [27 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```
The association rules are generated based on the default parameters.

```
> quality(rules.all) <- round(quality(rules.all), digits=3)
> rules.all
set of 27 rules
```
Totally 27 association rules are achieved and it is displayed.

```
> inspect(rules.all)
    lhs                          rhs          support confidence
[1]  {}                       => {Age=Adult}   0.950   0.950
[2]  {Class=2nd}              => {Age=Adult}   0.119   0.916
[3]  {Class=1st}              => {Age=Adult}   0.145   0.982
[4]  {Sex=Female}             => {Age=Adult}   0.193   0.904
[5]  {Class=3rd}              => {Age=Adult}   0.285   0.888
[6]  {Survived=Yes}           => {Age=Adult}   0.297   0.920
[7]  {Class=Crew}             => {Sex=Male}    0.392   0.974
[8]  {Class=Crew}             => {Age=Adult}   0.402   1.000
[9]  {Survived=No}            => {Sex=Male}    0.620   0.915
[10] {Survived=No}            => {Age=Adult}   0.653   0.965
[11] {Sex=Male}               => {Age=Adult}   0.757   0.963
[12] {Sex=Female,Survived=Yes} => {Age=Adult}  0.144   0.919
[13] {Class=3rd,Sex=Male}     => {Survived=No} 0.192   0.827
[14] {Class=3rd,Survived=No}  => {Age=Adult}   0.216   0.902
[15] {Class=3rd,Sex=Male}     => {Age=Adult}   0.210   0.906
[16] {Sex=Male,Survived=Yes}  => {Age=Adult}   0.154   0.921
[17] {Class=Crew,Survived=No} => {Sex=Male}    0.304   0.996
[18] {Class=Crew,Survived=No} => {Age=Adult}   0.306   1.000
[19] {Class=Crew,Sex=Male}    => {Age=Adult}   0.392   1.000
[20] {Class=Crew,Age=Adult}   => {Sex=Male}    0.392   0.974
[21] {Sex=Male,Survived=No}   => {Age=Adult}   0.604   0.974
[22] {Age=Adult,Survived=No}  => {Sex=Male}    0.604   0.924
[23] {Class=3rd,Sex=Male,Survived=No}  => {Age=Adult}   0.176   0.917
[24] {Class=3rd,Age=Adult,Survived=No} => {Sex=Male}    0.176   0.813
[25] {Class=3rd,Sex=Male,Age=Adult}    => {Survived=No} 0.176   0.838
[26] {Class=Crew,Sex=Male,Survived=No} => {Age=Adult}   0.304   1.000
[27] {Class=Crew,Age=Adult,Survived=No} => {Sex=Male}   0.304   0.996
    lift
[1]  1.000
[2]  0.964
[3]  1.033
[4]  0.951
[5]  0.934
[6]  0.968
```

[7] 1.238
[8] 1.052
[9] 1.164
[10] 1.015
[11] 1.013
[12] 0.966
[13] 1.222
[14] 0.948
[15] 0.953
[16] 0.969
[17] 1.266
[18] 1.052
[19] 1.052
[20] 1.238
[21] 1.025
[22] 1.175
[23] 0.965
[24] 1.034
[25] 1.237
[26] 1.052
[27] 1.266

All the 27 rules that are achieved and lift values for it are displayed.

Describe the value of lift that you see and explain how they help you evaluate the quality of the rules.

Lift is an interesting parameter in association rules analysis. Lift is a performance measure of a target model at classification and prediction or classifying based on the response measured against a random choice target model. Lift is the ratio of target response divided by average response.

Lift is found by dividing the confidence by the unconditional probability of the consequent. That is ratio of Confidence to Expected Confidence. It can also be found using dividing the support by the probability of the antecedent times the probability of the consequent.

According to arules package, Lift is the probability (support) of the itemset over the product of the probabilities of all items in the itemset, i.e., $supp(X) / \prod_{x \in X} supp(X)$. This is a measure of dependence similar to lift for rules. Range: $[0, \infty]$ (1 indicated independence)

- If the association rule has a lift value of 1 then it implies the probability of the occurrence of antecedent and consequent are independent of each other. If the two attributes are independent then no association rule can be formed.

- If the lift value for a rule is greater than one then it implies the relation between the antecedent and consequent is more significant. If the lift ratio is larger then the significance of the association rule is more.

- If lift is smaller than 1 indicates that the rule body and the rule head appear less often together than expected, the occurrence of the rule body has a negative effect on the occurrence of the rule head.

The lift value of the first association rule is 1. This implies that the antecedent and consequent are independent of each other. So this association rule is of no use and can be neglected.

[1] {} => {Age=Adult} s 0.950 C 0.950

The lift values of association rule 2, 4, 5, 6, 12, 14, 15, 16, 23 have values less than 1. This means the association rules appear less often and rule of body has negative effect on the rule of head.

[2] {Class=2nd} => {Age=Adult} s 0.119 C 0.916
[4] {Sex=Female} => {Age=Adult} s 0.193 C 0.904
[5] {Class=3rd} => {Age=Adult} s 0.285 C 0.888
[6] {Survived=Yes} => {Age=Adult} s 0.297 C 0.920
[12] {Sex=Female,Survived=Yes} => {Age=Adult} s 0.144 C 0.919
[14] {Class=3rd,Survived=No} => {Age=Adult} s 0.216 C 0.902
[15] {Class=3rd,Sex=Male} => {Age=Adult} s 0.210 C 0.906
[16] {Sex=Male,Survived=Yes} => {Age=Adult} s 0.154 C 0.921
[23] {Class=3rd,Sex=Male,Survived=No} => {Age=Adult} s 0.176 C 0.917

The remaining lift values of the association rules are greater than 1 and these rules are interesting and can be used to analyze and classify. If the lift value is greter than 1 and if it is the high score then significance of that rule is more.

[3] {Class=1st} => {Age=Adult} 0.145 0.982
[7] {Class=Crew} => {Sex=Male} 0.392 0.974
[8] {Class=Crew} => {Age=Adult} 0.402 1.000
[9] {Survived=No} => {Sex=Male} 0.620 0.915
[10] {Survived=No} => {Age=Adult} 0.653 0.965
[11] {Sex=Male} => {Age=Adult} 0.757 0.963
[13] {Class=3rd,Sex=Male} => {Survived=No} 0.192 0.827
[17] {Class=Crew,Survived=No} => {Sex=Male} 0.304 0.996
[18] {Class=Crew,Survived=No} => {Age=Adult} 0.306 1.000
[19] {Class=Crew,Sex=Male} => {Age=Adult} 0.392 1.000
[20] {Class=Crew,Age=Adult} => {Sex=Male} 0.392 0.974
[21] {Sex=Male,Survived=No} => {Age=Adult} 0.604 0.974
[22] {Age=Adult,Survived=No} => {Sex=Male} 0.604 0.924
[24] {Class=3rd,Age=Adult,Survived=No} => {Sex=Male} 0.176 0.813
[25] {Class=3rd,Sex=Male,Age=Adult} => {Survived=No} 0.176 0.838
[26] {Class=Crew,Sex=Male,Survived=No} => {Age=Adult} 0.304 1.000
[27] {Class=Crew,Age=Adult,Survived=No} => {Sex=Male} 0.304 0.996

The rule 17 and 27 has the greater lift value so the best association rules are 17 and 27. The other significant rules are 7, 20, 25, 13, 22, 9, 8, 18, 19, 26, 24, 3, 21, 10, 11 in order.

The other rules whose significance is less are 2, 4 , 5, 6, 12, 14, 15, 16, 23. The lift values of these rules are nearly equal to 1 so these rules does not have more significance but these rules will appear less often.

Based on the given lift values the quality and the significance of the rules are identified.

```
> # rules with rhs containing "Survived" only
> 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"))
> quality(rules) <- round(quality(rules), digits=3)
> rules.sorted <- sort(rules, by="lift")
> inspect(rules.sorted)
```

```
    lhs                                    rhs          support confidence
[1]  {Class=2nd,Age=Child}            => {Survived=Yes} 0.011   1.000
[2]  {Class=2nd,Sex=Female,Age=Child} => {Survived=Yes} 0.006   1.000
[3]  {Class=1st,Sex=Female}           => {Survived=Yes} 0.064   0.972
[4]  {Class=1st,Sex=Female,Age=Adult} => {Survived=Yes} 0.064   0.972
[5]  {Class=2nd,Sex=Female}           => {Survived=Yes} 0.042   0.877
[6]  {Class=Crew,Sex=Female}          => {Survived=Yes} 0.009   0.870
[7]  {Class=Crew,Sex=Female,Age=Adult} => {Survived=Yes} 0.009   0.870
[8]  {Class=2nd,Sex=Female,Age=Adult} => {Survived=Yes} 0.036   0.860
[9]  {Class=2nd,Sex=Male,Age=Adult}   => {Survived=No}  0.070   0.917
[10] {Class=2nd,Sex=Male}             => {Survived=No}  0.070   0.860
[11] {Class=3rd,Sex=Male,Age=Adult}   => {Survived=No}  0.176   0.838
[12] {Class=3rd,Sex=Male}             => {Survived=No}  0.192   0.827
    lift
[1]  3.096
[2]  3.096
[3]  3.010
[4]  3.010
[5]  2.716
[6]  2.692
[7]  2.692
[8]  2.663
[9]  1.354
[10] 1.271
[11] 1.237
[12] 1.222
```

The rhs of the association rules is set to survived. In order to avoid the empty set in rule minlen is set to 2 and the rules are sorted based of the lift value in decreasing order .The rules and lift values are displayed. The greater lift value, the significance of association rule is greater.

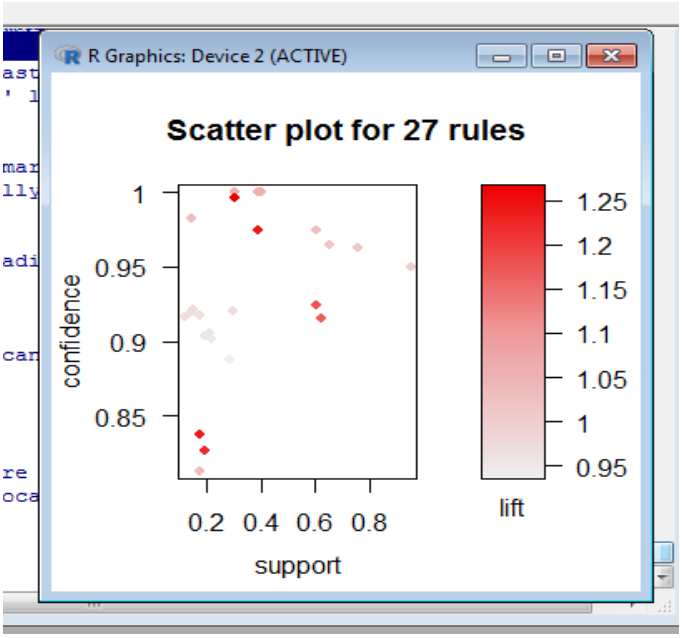
```
> subset.matrix <- is.subset(rules.sorted, rules.sorted)
> subset.matrix[lower.tri(subset.matrix, diag=T)] <- NA
```

```
> redundant <- colSums(subset.matrix, na.rm=T) >= 1
> which(redundant)

 {Class=2nd,Sex=Female,Age=Child,Survived=Yes}
                                2
 {Class=1st,Sex=Female,Age=Adult,Survived=Yes}
                                4
{Class=Crew,Sex=Female,Age=Adult,Survived=Yes}
                                7
 {Class=2nd,Sex=Female,Age=Adult,Survived=Yes}
                                8
```
The redundant rules are checked and displayed.

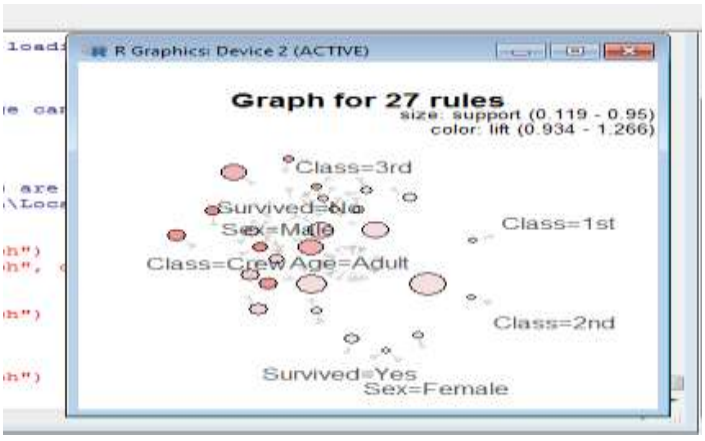
```
> rules.pruned <- rules.sorted[!redundant]
> inspect(rules.pruned)
   lhs                          rhs          support confidence lift
[1] {Class=2nd,Age=Child}        => {Survived=Yes} 0.011   1.000     3.096
[2] {Class=1st,Sex=Female}       => {Survived=Yes} 0.064   0.972     3.010
[3] {Class=2nd,Sex=Female}       => {Survived=Yes} 0.042   0.877     2.716
[4] {Class=Crew,Sex=Female}      => {Survived=Yes} 0.009   0.870     2.692
[5] {Class=2nd,Sex=Male,Age=Adult} => {Survived=No}  0.070   0.917     1.354
[6] {Class=2nd,Sex=Male}         => {Survived=No}  0.070   0.860     1.271
[7] {Class=3rd,Sex=Male,Age=Adult} => {Survived=No}  0.176   0.838     1.237
[8] {Class=3rd,Sex=Male}         => {Survived=No}  0.192   0.827     1.222
```
The redundant rules are pruned and displayed.

```
> rules <- apriori(titanic.raw,
+ parameter = list(minlen=3, supp=0.002, conf=0.2),
+ appearance = list(rhs=c("Survived=Yes"),
+ lhs=c("Class=1st", "Class=2nd", "Class=3rd",
+ "Age=Child", "Age=Adult"),
+ default="none"),
+ control = list(verbose=F))
> rules.sorted <- sort(rules, by="confidence")
> inspect(rules.sorted)

   lhs                  rhs          support     confidence lift
[1] {Class=2nd,Age=Child} => {Survived=Yes} 0.010904134 1.0000000  3.0956399
[2] {Class=1st,Age=Child} => {Survived=Yes} 0.002726034 1.0000000  3.0956399
[3] {Class=1st,Age=Adult} => {Survived=Yes} 0.089504771 0.6175549  1.9117275
[4] {Class=2nd,Age=Adult} => {Survived=Yes} 0.042707860 0.3601533  1.1149048
[5] {Class=3rd,Age=Child} => {Survived=Yes} 0.012267151 0.3417722  1.0580035
[6] {Class=3rd,Age=Adult} => {Survived=Yes} 0.068605179 0.2408293  0.7455209
```
The lhs of the rules are set to Class1, Class 2, Class 3 and Age to Child and Adult and rhs is set to survived = yes. The association rules are generated accordingly and sorted in decreasing order of lift and displayed.

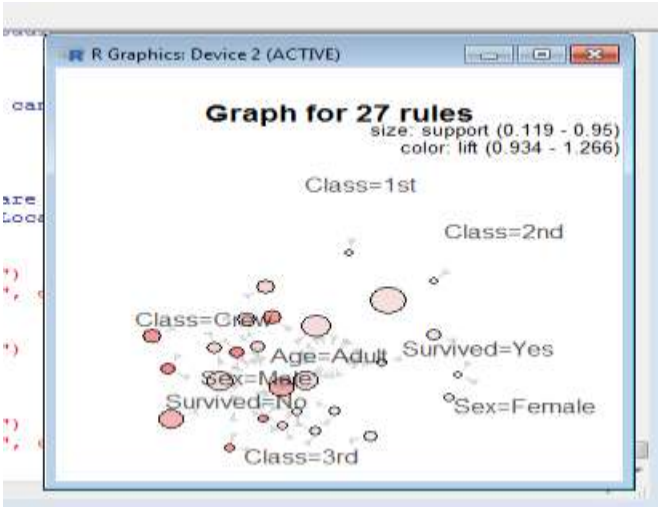
> library(arulesViz)
> plot(rules.all)



> plot(rules.all, method="graph")



> plot(rules.all, method="graph", control=list(type="items"))



Graphs are generated based on the association rules.

Describe the final set of rules (how do they differ from the first rules that you computes and discuss what you see on the visualization plots and what information we can learn from them.

The first set of rules had 27 rules in which many rules had very less significant. The most important rules were found based on lift values. This will not work if the dataset is too large. The interesting facts about the dataset found in these rules were very less and redundant. It had more rules which has no significance. The comparison based on these rules would be improper.

In order to generate more interesting association rule a medium level support and confidence value was set and to avoid empty set minlen was set to two. The association rules achieved more interesting than the rules generated first time. Even these rules were more and inorder to avoid redundancy pruning was done recursively and then a set of rules were generated.
After finding the significant rules, understanding the rules is challenging.

Eg: "{Class=2nd, Age=Child}=> {Survived=Yes}" .This has a confidence of one and a lift of three and there are no rules on children of the 1st or 3rd classes. Therefore, it might be interpreted by users as children of the 2nd class had a higher survival rate than other children. This is wrong. This rule provides information that children of class2 survived but no other information to compare the survival rates. So the final set of rules were generated by setting lower threshold values for support and confidence and lhs and rhs are set accordingly which can be compared.

The final set of rules are more significant and not redundant. It provides more information about the items and dataset. These rules can be used for comparison and analysis whereas the rules generated before was less interesting and the analysis could not be done properly.

Graph Visualization

Graph 1: From the first graph scatter plots for 27 rules, The colour represents the lift values, the x axis represent the support and y axis represent confidence. More rules are generated when the support value is low and confidence value is high. Most of the rules generated when

confidence value is high and support value less is of less significance. It does not show any interesting facts about the items and dataset. The rules with medium confidence and medium support had more significance and some rules with minimum confidence and minimum support had high significance. In general only seven rules were significant out of 27. It is not useful very much.

Graph 2: From second graph Graph for 27 plots, The size of the node denotes the support. If the size of the node is big then the support of the rule is high. The colour denotes the lift value. The nodes with high lift values are of more significance. The nodes with more support is less . The nodes with medium support is more. More rules were generated when the support is set to a medium value and it is of greater importance. The nodes with high lift values specify the more significant rules. The denser area in the graph denotes the frequent items of the dataset which forms certain association rules. The nodes which are not in the denser area and which are alone denotes the less frequent items in the dataset. These nodes which are made of non frequent items has very small support value. More significant rules are less but rules which has less significance is more in these and redundant rules can be found.

Graph 3: This graph represents focuses on how the rules are composed of individual items and shows which rules share items. Control is a list with further control arguments to customize the plot. Based on the items the plots in the graph2 is customised. In this it shows how individual items are grouped to form the association rules. The other results are similar to graph 2.

3 Exercises from the course book "Introduction to Data mining". Note that the solutions are available on the web, so only few points will be assigned to each exercise but there will be a penalty if an exercise is missing. I encourage you to work on the solutions yourself, of course. It is ok if you verify your answer with the solutions from the web. However, you have to prepare the answers in YOUR OWN WORDS, cut-and-paste from the available solutions will result in penalty. Explain every answer briefly in your own words, just a short answer without explanation will be zero point. This is a very good preparation for the midterm and final.

3.1 Chapter -4

Exercise 5. Consider the following data set for a binary class problem.

A	B	Classs Label
T	F	+
T	T	+
T	T	+
T	F	-
T	T	+
F	F	-
F	F	-
F	F	-
T	T	-
T	F	-

(a) Calculate the information gain when splitting on A and B. Which attribute

would the decision tree induction algorithm choose?

The overall entropy og the data before splitting the attributes is

$E_{orignal}$ = - $P(C_A)$ $\log_2$ $P(C_A)$- $P(C_B)\log_2$ $P(C_B)$ = - 4/10 $\log_2$ 4/10 - 6/10 $\log_2$ 6/10
= - 0.4 $\log_2$ 0.4 – 0.6$\log_2$ 0.6 = 0.52876 + 0.44214 = 0.9710

The contingency table after splitting the attributes

Class	A = T	A =F
+	4	0
-	3	3

Class	B = T	B =F
+	3	1
-	1	5

The Information Gain of A after splitting is

$E_{A=T}$ = -4/7 $\log_2$ 4/7 -3/7 $\log_2$ 3/7 = -0.571 $\log_2$ 0.571 -0.4285$\log_2$ 0.4285= 0.4615 + 0.5236
= 0.9851

$E_{A=F}$ = -3/3 $\log_2$ 3/3 -0/3 $\log_2$ 0/3 = -1$\log_2$ 1 -0 $\log_2$ 0 = 0

Δ = $E_{original}$ – 7/10 $E_{A=T}$ - 3/10$E_{A=F}$ = 0.9710 -0.6895 = 0.2813

The Information Gain of B after splitting is

$E_{B=T}$ = -3/4 $\log_2$ 3/4 -1/4 $\log_2$ 1/4 = -0.75 $\log_2$ 0.75 -0.25$\log_2$ 0.25= 0.31125+0.5= 0.8113

$E_{B=F}$ = -1/6 $\log_2$ 1/6 -5/6 $\log_2$ 5/6 = -0.1666 $\log_2$ 0.1666 -0.8333 $\log_2$0.8333
= 0.4307 + 0.2912 = 0.64993 = 0.6500

Δ = $E_{original}$ – 4/10 $E_{B=T}$ - 6/10$E_{B=F}$ = 0.9710 -0.3245 – 0.39= 0.2565

Since attribute A value is more it will be chosen to split the node.

(b) Calculate the gain in tbe Gini index when splitting on A and B. Which attribute would the decision tree induction algorithm choose?

The GINI before splitting is

$G_{original}$ =1- $P(C_A)^2$ - $P(C_B)^2$ = 1- $(4/10)^2$ - $(6/10)^2$ = 1 - 0.16 - 0.36 = 0.48

The gain in GINI Index after Splitting A is

$G_{A=T}$ = 1- $(4/7)^2$ - $(3/7)^2$ = 1 – 0.32649 – 0.18367 = 0.4898

$G_{A=F}$ = 1- $(3/3)^2$ - $(0/3)^2$ = 1-1 = 0

Δ = $G_{original}$ – 7/10 $G_{A=T}$ - 3/10$G_{A=F}$ = 0.48 – 0.3428 = 0.1372

$G_{B=T} = 1- (1/4)^2 - (3/4)^2 = 1- 0.0625 – 0.5625 = 0.3750$

$G_{B=F} = 1- (1/6)^2 - (5/6)^2 = 1- 0.02777 – 0.69438 = 0.2778$

$\Delta = G_{original} – 4/10\ G_{B=T} - 6/10 G_{B=F} = 0.48 – 0.15 – 0.1666 = 0.1634$

Since attribute B Gini Index is more , attribute B will be chosen to split the node.

(c) The figure shows that entropy and the Gini inde.x are both monot-onously increasing on the range [0. 0.5] and they are both monotonously decreasing on the range [0.5, 1]. Is it possible that information gain and the gain in the Gini index favor different attributes? Explain.

Yes, it is possible that information gain and the gain in the Gini index can favor different attributes. Even though the measures have similar range and monotonous behaviour and their gains are scaled differences of the measure, it is not necessary that they should behave in the same way.

3.2 Chapter – 6

Exercise 2

Customer ID	Transaction ID	Items Bought
1	0001	{a,d,e}
1	0024	{a,b,c,e}
2	0012	{a,b,d,e}
2	0031	{a,c,d,e}
3	0015	{b,c,e}
3	0022	{b,d,e}
4	0029	{c,d}
4	0040	{a,b,c}
5	0033	{a,d,e}
5	0038	{a,b,e}

(a)Compute the support for itemsets {e}, {b, d}, and {b,d, e} by treating each transaction ID as a market basket.

Support count of an itemset = No.of transactions that contain the itemset/ Total no. of transactions. s{itemset} = σ{itemset} / Number of instances

$s(\{e\}) = 8/10 = 0.8$

$s(\{b,d\}) = 2/10 = 0.2$

$s(\{b,d,e\}) = 2/10 = 0.2$

(b) Use the results in part (a) to compute the confidence for the association rules {b,d} →{e} and {e} →{b,d). Is confidence a symmetric measure?

Confidence X→Y

- How often the transaction that contain X also contain Y
- C (X →Y) = σ (X ∪ Y)/σ(X)

C(bd→e) = 0.2/ 0.2 = 100 %

C (e →bd) = 0.2/0.8 = 1 / 4 = 25%

From the above results it is clear that confidence is not a symmetric measure.

Exercise -6

Transaction ID	Items Bought
1	{Milk,Beer,Diapers}
2	{Bread,Butter,Milk}
3	{Milk, Diapers,Cookies}
4	{Bread,Butter,Cookies}
5	{Beer,Cookies,Diapers}
6	{Milk,Diapers,Bread,Butter}
7	{Bread,Butter,Diapers}
8	{Beer,Diapers}
9	{Milk,Diapers,Bread,Butter}
10	{Beer,Cookies}

(a) What is the maximum number of association rules that can be extracted from this data (including rules that have zero support)?

The number of possible rules that contains d items
$R = 3^d - 2^{(d+1)} + 1$

There are 6 items in the data set.
$R = 3^6 - 2^{(6+1)} + 1 = 729 - 128 + 1 = 602$

The total number of rules is 602

(c) Write an expression for the maximum number of size-3 itemsets that can be derived from this data set.

The maximum number of k-itemsets that can be achieved from a dataset of size d is expressed as $\begin{bmatrix} d \\ k \end{bmatrix}$ which is combinations of size k that a d sized dataset can give rise to.

Thus the expression can be alternately written as ${}_dC_k$.

$$\begin{bmatrix} 6 \\ 3 \end{bmatrix}$$

= 6*5*4 / 1*2*3 = 20

(d) Find an itemset (of size 2 or larger) that has the largest support.

The 1-itemset with largest support in the given dataset is diapers (7). The second largest are Milk, Bread and Butter with support of 5 each, Beer and Cookies support is 4. First let us find the itemset between these items and find their support.

s{Diapers, Milk} – 4
s{Milk, Bread} - 3
s{Diapers, Bread} – 3
s{Milk, Butter} - 3
s{Diapers, Butter} – 3
s{Bread, Butter} – 5
s{Diapers, Cookies} – 2
s{Diapers, Beer} – 3
s{Milk, Cookies} – 1
s{Milk, Beer} – 1
s{Bread, Cookies} – 1
s{Bread, Beer} – 0
s{ Butter, Cookies} – 1
s{Butter, Beer} – 0
s{Cookies, Beer} – 2

Thus the {Bread, Butter} itemset has a maximum support of 5 among the others.

Answer: {Bread,Butter} has largest support.

(e) Find a pair of items, a and b, such that the rules {a} →{b} and {b} →{a} have the same confidence.

- $C(X \rightarrow Y) = \sigma(X \cup Y)/\sigma(X)$

The relationship c(a->b)=c(b->a) will hold good for any pair of items a and b, if their individual supports are equal. Equal support leads to equal denominators in confidence expression and thus yield same confidence, given that the numerator is always equal for c(a->b) and c(b->a). Thus, the item pairs that will satisfy the given relationship are (Bread, Butter), (Milk, Butter), (Milk,Bread) and (Beer, Cookies) which have same individual supports pairwise.

C(Beer → Cookies) = 2/4 = 0.5
C(Cookies → Beer) = 2/4 =0.5

C(Bread→ Butter) = 5/5 = 1
C(Butter → Bread) = 5/5 = 1

C(Milk→ Butter) = 5/5 = 1
C(Butter→ Milk) = 5/5 = 1

C(Milk→ Bread) = 5/5 = 1
C(Bread→ Milk) = 5/5 = 1

Answer: (Beer, Cookies), (Bread,Butter), (Milk, Butter), (Milk,Bread) have the same confidence pairwise.

Exercise 8

The Apriori algorithm uses a generate-and-count strategy for deriving frequent itemsets. Candidate itemsets of size k + 1 are created by joining a pair of frequent itemsets of size k (this is known as the candidate generation step). A candidate is discarded if any one of its subsets is found to be infrequent during the candidate pruning step. Suppose the Apri01'i algorithm is applied to the data set shown in Table with min sup = 30%, i.e.1 any itemset occurring in less than 3 transactions is considered to be infrequent.

Transaction ID	Items Bought
1	$\{a, b, d, e\}$
2	$\{b, c, d\}$
3	$\{a, b, d, e\}$
4	$\{a, c, d, e\}$
5	$\{b, c, d, e\}$
6	$\{b, d, e\}$
7	$\{c, d\}$
8	$\{a, b, c\}$
9	$\{a, d, e\}$
10	$\{b, d\}$

(a) Draw an itemset lattice representing the data set given in Table .Label each node in the lattice with the following letter(s):

- ***N: If the it.emset is not considered to be a candidate itemset by the A priori algorithm. There are two reasons for an itemset not to be considered as a candidate itemset: (1) it is not generated at all during the candidate generation step, or (2) it is generated during the candidate generation step hut is subsequently removed during the candidate pruning step because one of its subsets is found to be infrequent.***
- ***F: If the candidate itemset is found to be frequent by the Apriori algorithm.***
- ***I: If the candidate itemset is found to be infrequent after support counting.***

Using the given dataset, enumerating the support all k-itemsets where k ranges from 1 to 4,
s{a}=5, s{b}=7, s{c}=5, s{d}=9, s{e}=6
All of the 1-itemsets satisfy the minimum support count, so we take all of them for next level calculations.
s{a,b}=3, s{a,c}=2, s{a,d}=4, s{a,e}=4
s{b,c}=3, s{b,d}=6, s{b,e}=4
s{c,d}=4, s{c,e}=2, s{d,e}=6

In this level, {a,c} and {c,e} can be pruned as infrequent since they have a support less than minimum support. All of supersets containing {a,c} and {c,e} can be ignored as infrequent rightaway by Apriori principle which states that if an itemset is infrequent, then all of its supersets will also be infrequent.

Proceeding to next level with only the valid 2-itemsets,
s{a,b,d}=2, s{a,b,e}=2, s{a,d,e}=4
s{b,c,d}=2, s{b,d,e}=4

In this level, all itemsets can be pruned except {a,d,e} and {b,d,e}.

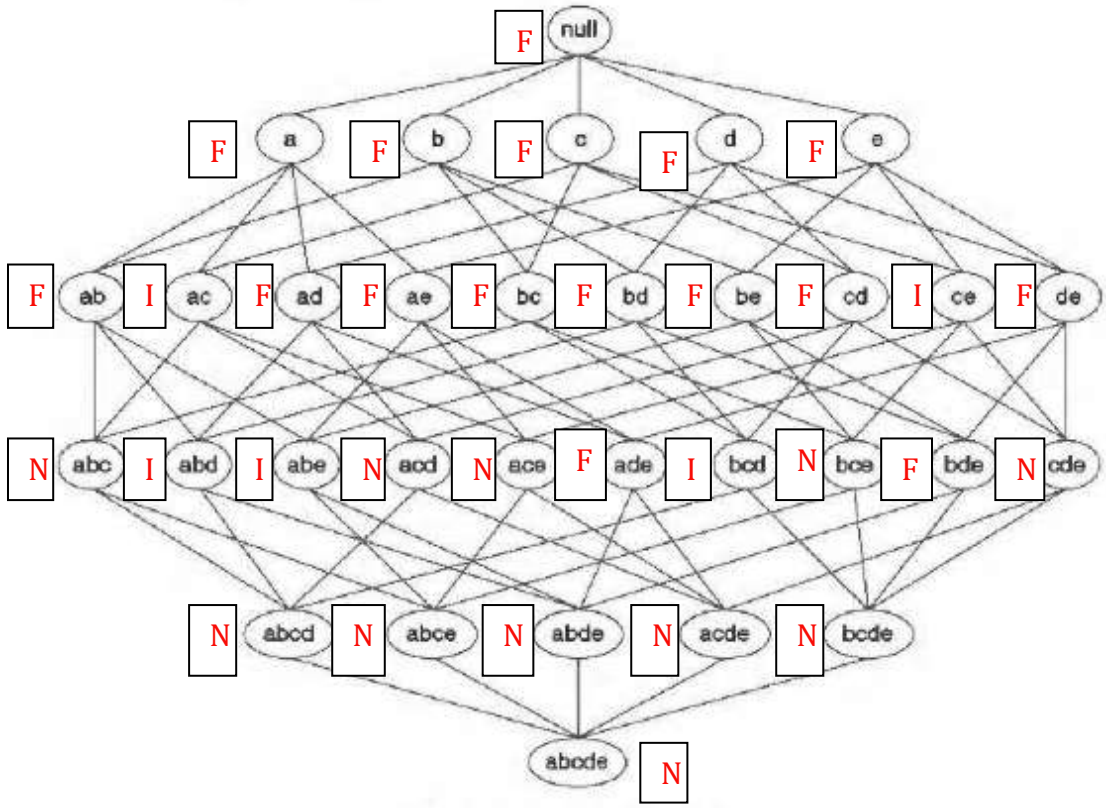
There is no need to calculate support for other k-itemsets where k>3 since we know that all of them will be infrequent too by Apriori principle.

The frequent itemsets {null}, {a}, {b}, {c}, {d}, {e}, {a,b}, {a,d}, {a,e}, {b,c}, {b,d}, {b,e}, {c,d}, {d,e}, {a,d,e}, {b,d,e} can be assigned a label of F.

The infrequent itemsets {a,c}, {c,e}, {a,b,d}, {a,b,e}, {b,c,d} can be assigned the label of I.

All other itemsets {a,b,c}, {a,c,d}, {a,c,e}, {b,c,e}, {c,d,e}, {a,b,c,d}, {a,b,c,e}, {a,b,d,e}, {a,c,d,e}, {b,c,d,e}, {a,b,c,d,e} which were ignored as infrequent without calculating support can be assigned alabel of N.

The resultant itemset lattice with the N, F and I labels added is as follows.



(b) What is the percentage of frequent itemsets (with respect to all itemsets in the lattice)?

Percentage of frequent itemsets = Count of F / total no.of itemsets = 16/32 =0.5 = 50%.
This includes null set.

(c) What is the pruning ratio of the Apriori algorithm on this data set?(Pruning ratio is defined as the percentage of itemsets not considered to be a candidate because (1) they are not generated during candidate generation or (2) they ru:e pruned during the candidate pruning step.)

Pruning Ratio is the ration of N to the total number of itemsets.
Pruning ratio = Count of N/ Total no. of itemsets = 11/32 = 34.375 %

(d) What is the false alarm rate (i.e percentage of candidate itemsets that are found to be infrequent after performing support counting)?

False alarm rate is the Ratio of I to the total no. of itemsets.
Count of I = 5
Total no.of Itemsets = 32
Therefore, False alarm rate = 5/32 = 15.625 %

Exercise 9

The Apriori algorithm uses a hash tree data structure to efficiently count the support of candidate itemsets. Consider the hash tree for candidate 3-itemsets shown in Figure.

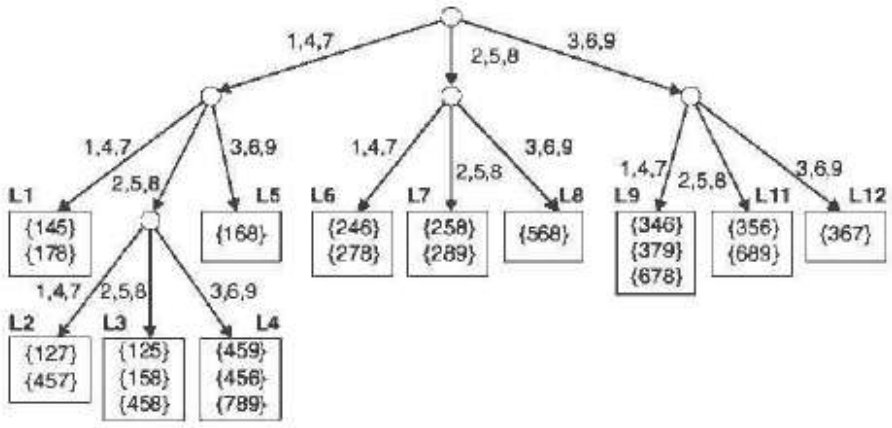
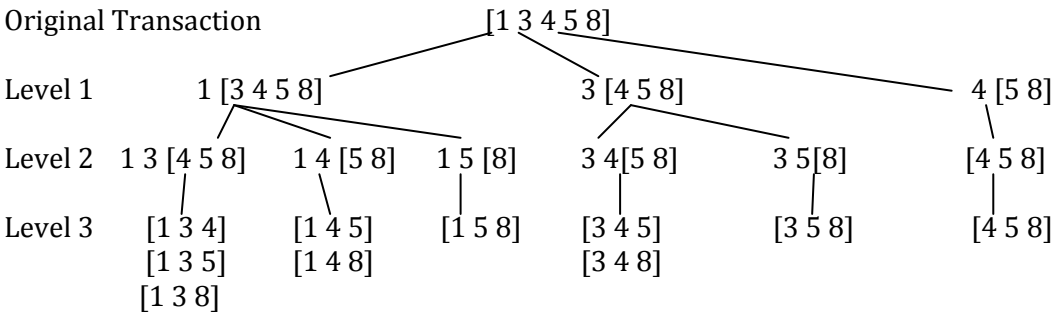


Figure 6.32. An example of a hash tree structure.

(a)Given a transaction that contains items {1,3,4,5,8}, which of the hash tree leaf nodes will be visited when finding the candidates of the transaction?

For the given transaction, subset enumeration can be done as follows.



The hash tree leaf nodes that will be visited when finding the candidates of the transaction are L1, L3, L5, L9 and L11

(b) Use the visited leaf nodes in part (b) to determine the candidate itemsets that are contained in the transaction {1,3,4,5,8} .

Itemset	Leaf Node
{1, 4, 5}	L1
{1, 5, 8} {4, 5, 8}	L3

The candidate itemsets that are contained in the transaction are {1,4,5}, {1,5,8} and {4,5,8}

Reference: Definition of Lift and concepts about Lift
https://en.wikipedia.org/wiki/Lift_(data_mining),
http://www.ibm.com/support/knowledgecenter,
http://www.solver.com/xlminer/help/association-rules