DATA MINING

COSC 2111– Assignment 1

Part 1: Classification

3.1.1

Run No	Classifier	Default Parameters	Training Error	Cross-valid Error	Over - Fitting
1	ZeroR	None	7.7147	7.7147	No
2	OneR	B-6	2.9692	3.685	Yes (0.7158)
3	J48	C - 0.25, M - 2	0.1856	0.4242	Yes (0.2386)
4	IBK	K - 1, W - 0	0.00	8.4836	Yes (8.4836)

- When cross validation error is higher than training error Overfitting occurs.
- As the errors are same ZeroR does not show overfitting.
- The difference between the errors is low for J48 and hence it has the least overfitting
- Overfitting is seen more in IBK with the least training error and highest training error.
- Zero error has the highest training error where as J48 has the least training error.
- Similarly, IBK has the highest cross validation error where as J48 has the least.
- Therefore, we can conclude that J48 is the better classifier as it has the least train and cross validation errors with minimum overfitting.

3.1.2

Run No	Classifier	Parameters	Training Error	Cross-valid Error	Over - Fitting
1	J48	C-0.25, M-10	0.6363	0.6893	Yes (0.053)
2	J48	C-0.25, M-20	0.8484	0.8484	None
3	J48	C-0.20, M-2	0.1856	0.4242	Yes (0.2386)
4	J48	C-0.15, M-2	0.2386	0.4242	Yes (0.1856)
5	J48	C-0.15, M-20	0.8484	0.8484	None

- Lower the confidence factor, the higher is the pruning. Pruning helps to reduce the size and complexity of decision trees. It also gives the better predictive accuracy by reducing overfitting.
- For simpler tree, the minNumObj (M) should be maximun.
- Therefore, J48 with confidence factor (C) = 0.15 and minNumObj (M) = 20, has no overfitting.

3.1.3

Run No	Classifier	Parameters	Average Precision	Average Recall	Incorrect Classified Instances (%)
1	J48	C-0.25 M-2 (Percentage split: 66%)	0.993	0.995	0.5460
2	J48	C-0.25 M-2 (Percentage split: 50%)	0.994	0.995	0.5302
3	J48	C-0.25 M-2 (Percentage split: 40%)	0.994	0.995	0.5303
4	J48	C-0.25 M-2 (Percentage split: 30%)	0.993	0.995	0.7197
5	J48	C-0.25 M-2 (Percentage split: 20%)	0.988	0.988	1.2260

- The percentage of data at which the training and testing is made is known as Percentage split.
- The decrease in the percentage split results in the decrease in average precision as well as recall.

- Therefore, the higher the percentage split the incorrect classified instances are higher.

3.1.4

Run No	Classifier	Parameters	Training Error	Cross-valid Error	Over - Fitting
1	IBK	K - 10	6.5217	6.7603	Yes (0.2386)
2	IBK	K – 25	7.1315	7.2110	Yes (0.08)
3	IBK	K - 50	7.6882	7.8208	Yes (0.1326)
4	IBK	K – 75	7.8473	7.8208	None
5	IBK	K – 100	7.7147	7.7147	None

- For K = 1, The training error will be 0 with overfitting.
- The increase in K value helps in better predictions as more data is considered.
- K value that can minimize or reduce the overfitting to the large extent, but with a small value to avoid oversimplification of the distribution is to be chosen. Therefore K – 100 minimises the overfitting.

3.1.5

Run No.	Classifier	Parameters	Training Accuracy %	Cross-valid Accuracy %	Over-fitting	Time taken (seconds)
1	PART	C = 0.25, M = 2	99.87	99.42	Yes (0.45)	0.03
2	PART	C = 0.9, M = 2	99.89	99.39	Yes (0.5)	0.01
3	LWL	K = 0	95.39	95.39	Yes (0)	43.95
4	LWL	K = 2	100	90.45	Yes (9.55)	3.03
5	LWL	K = 6	99.97	93.53	Yes (6.44)	3.45

- Part classifier and LWL classifier shown high training and validation accuracies.
- Part at C 0.25 shown less over fitting than C 0.9. Suggesting that the increase in C values increases overfitting, even though train accuracy increases.
- For LWL K 0 shown the optimal accuracy with 0 overfitting but taken more time to compile.
- Part, which shows overfitting, compiled in very less time.
- Overall, Part is best. But LWL also performed better but has more training time.
- LWL is both best and worst depending on the parameters but PART seems to be better than LWL.

3.1.6

Classifier	Default Parameters	Training Accuracy	Cross-valid Accuracy	Over - Fitting
ZeroR	None	92.2853	92.2853	No
OneR	B-6	97.0308	96.315	Yes (0.7158)
J48	C - 0.25, M - 2	99.8144	99.5758	Yes (0.2386)

- Among the three classifiers ZeroR, OneR and J48, J48 has the highest training accuracy of 99.81% and cross validation accuracy of 99.58%.
- Also, an overfitting of 0.23% suggests that J48 has better performance compared to ZeroR and OneR.

3.1.7

➢ If the value of TSH is less than 6, we can say that the person or patient has no chance of having hypothyroid. Since TSH is false.

- Similarly, When TSH > 6 and FTI < 64, TSH is false. Therefore, the patient does not suffer from hypothyroid.

3.1.8

Run No	Classifier	Parameters	Attribute	Average Precision (%)	Average Recall (%)	Average F-Measure (%)	Cross-valid Error
1	J48	C-0.25, M-2	All	99.5	99.6	99.5	0.4242
2	J48	C-0.25, M-2	3, 8, 18, 22, 26	99.6	99.6	99.6	0.3977

- The attribute evaluator with J48 classifier selected the following attributes thyroxine, thyroid surgery, TSH, TT4, and FTI.
- This means that the J48 classifier performs well on these attributes.
- Therefore, we can observe that with the decrease in number of attributes the Average Precision, Average Recall, and Average F - Measure are increased, whereas the Cross - validation error decreased.

Part 2: Numeric Prediction

3.2.1

Run No	Classifier	Parameters	Mean Absolute Error (Training set) %	Relative Absolute error (Training set) %	Mean Absolute Error (Cross Validation set) %	Relative Absolute Error (Cross Validation set) %	Over - Fitting
1	ZeroR	None	87.3828	100	87.6583	100	Yes
2	M5P	M - 4.0	12.9306	14.7976	13.6917	15.6194	Yes
3	IBk	K – 1, W - 0	0	0	20.8278	23.7602	Yes

- The training error is highest for ZeroR and least for IBk. Whereas Cross validation error is more for IBk and least for M5P.
- The overfitting is minimal in ZeroR.
- Finally, we can conclude that M5P has better prediction ability compared to IBk and ZeroR as it has average values for both the errors and overfitting.

3.2.2

Run No	Classifier	Parameters	Mean Absolute Error (Training set) %	Relative Absolute error (Training set) %	Mean Absolute Error (Cross Validation set) %	Relative Absolute Error (Cross Validation set) %	Over - Fitting
1	M5P	M - 4.0	12.93	14.8	13.69	15.61	Yes
2	M5P	M - 6.0	14.77	16.9	13.07	14.9	Yes
3	M5P	M - 10.0	14.76	16.89	15.97	18.2	Yes
4	M5P	M - 16.0	16.69	19.1	18.02	20.56	Yes

- Overall, M5P shows better performance than IBk. M5P model has lower Mean absolute error than IBk model, which means M5P has highest performance than IBk.
- In all runs, for both the models, accuracy of validation set differs from the accuracy of training set. Thereby, there is overfitting in both the models.

- On increasing M (minNuminstances) value in M5P model, accuracy of a model is decreasing. Similarly, on increasing KNN value in IBk model, accuracy decreases.

3.2.3

Run No	Classifier	Parameters	Mean Absolute Error (Training set) %	Relative Absolute error (Training set) %	Mean Absolute Error (Cross Validation set) %	Relative Absolute Error (Cross Validation set) %	Over - Fitting
1	M5P	M - 4.0	12.93	14.8	13.69	15.61	Yes
2	M5P	M - 6.0	14.77	16.9	13.07	14.9	Yes
3	M5P	M - 10.0	14.76	16.89	15.97	18.2	Yes
4	M5P	M - 16.0	16.69	19.1	18.02	20.56	Yes

3.2.4

- CHMAX attribute has a minimum value of 0 and maximum value of 176 with mean of 18.27.
- CHMIN attribute has a minimum value of 0 and maximum value of 52 with mean of 4.69.
- Vendor with label has highest count.
- M5P model performs better on this data set compared to other classifiers.

Part 3: Clustering

3.3.1

- On increasing the K value, sum of squared errors within the clusters are decreasing gradually.
- The value of K is to be decided properly. Even though there is decrease in error or the improvement of quality in clustering process with the increase in clusters, there will be a decrease in the overall performance of the clusters. The more the clusters are the more clusters become useless.
- Let us consider k = 20, as there will be more than one clusters with the same value for example say age of 64 is to be grouped into the mean age of 64 of a cluster. There we got 2 clusters of the same value. The age 64 is kept into only one cluster leaving the other cluster useless. Also, the decision is not significant in this case. Since both clusters defines the group of that age value (64).

3.3.2

- Seed determines the randomly generated cluster centroids.
- Here, on this data or the clustering process, the percentage split of cluster changes when seed values are changed with default K or number of clusters value 2.
- Seed from 1 to 3 gives the same results with sum of squares error (SSE) 38.46 and 70% and 30% of two clusters, respectively. When seed is 4 and 5 it gives the 76% and 24% respectively with greater SSE of 1168.15.
- This is due to the even distribution of data.

3.3.3

EM (Expectation Maximization):

1. On executing EM algorithm with default parameters, six clusters are selected by the cross validation.
2. Cluster 5 holds 50% of the whole data. Remaining 50% splits among the clusters 0 to 4.
3. All the clusters have highest number of females than males.
4. Log likelihood value recorded as -11.87592.

```
EM
==

Number of clusters selected by cross validation: 6
Number of iterations performed: 84


                Cluster
Attribute            0          1          2          3          4          5
                  (0.24)     (0.06)     (0.16)     (0.03)     (0.02)      (0.5)
=============================================================================
age
  mean            49.5944    52.7328    51.9588    47.6349    47.0337    52.988
  std. dev.       19.7506    19.7055    17.6174    47.0711    16.8983   18.6518

sex
  F              679.1847   179.2248   454.3139    89.7395    56.3661  1177.171
  M              237.2221    50.5307   134.2373     6.9611    21.3861   697.6628
  [total]        916.4067   229.7554   588.5512    96.7006    77.7522  1874.8338
TSH
  mean             4.4569    14.8677     0.1109     3.1982   118.0143     1.2911
  std. dev.        1.4166     9.0444     0.0958     2.1816   114.5321     0.7265

TT4
  mean           100.2776    89.0489   137.1098   172.0312    33.8318   105.3636
  std. dev.       18.2157    33.9594    48.3878    28.9964    23.9997    22.8037
```

3.3.4

EM algorithm observations after normalizing the numeric data,

- Six clusters are divided with the same percentage as before normalization run.
- Age and TSH attributes are normalized.
- Although TT4 is a numerical data, there is no change in the values after normalizing.
- Since sex is a factor variable there is no effect of normalization on it.
- Log likelihood value increases from -11.87592 to 0.51504.

```
EM
==

Number of clusters selected by cross validation: 6
Number of iterations performed: 84


                Cluster
Attribute            0          1          2          3          4          5
                  (0.24)     (0.06)     (0.16)     (0.03)     (0.02)      (0.5)
=============================================================================
age
  mean             0.107      0.1139     0.1122     0.1027     0.1014     0.1145
  std. dev.        0.0435     0.0434     0.0388     0.1037     0.0372     0.0411

sex
  F              679.1847   179.2248   454.3139    89.7395    56.3661  1177.171
  M              237.2221    50.5307   134.2373     6.9611    21.3861   697.6628
  [total]        916.4067   229.7554   588.5512    96.7006    77.7522  1874.8338
TSH
  mean             0.0084     0.028      0.0002     0.006      0.2227     0.0024
  std. dev.        0.0027     0.0171     0.0002     0.0041     0.2161     0.0014

TT4
  mean           100.2776    89.0489   137.1098   172.0312    33.8318   105.3636
  std. dev.       18.2157    33.9594    48.3878    28.9964    23.9997    22.8037
```

3.3.5

minLogLikelihoodImprovementCV:

- On increasing the value of minLogLikelihoodImprovementCV from 1.0E-6, 0.1 and 1.0 respectively, time taken to build the EM model decreases from 60.47, 25.85, and 6.13 seconds, respectively. Whereas Log likelihood value decreased significantly.
- Number of clusters decreases from 6, 4 and 2 respectively on increasing minLogLikelihoodImprovementCV value.

minStdDev:

On increasing the value of minStdDev from 1.0E-6, 0.1 and 1.0 respectively, time taken by EM model for execution also increases drastically from 43.05, 61.81 and 78.71 seconds, respectively. On the other hand, Log likelihood value decreased significantly.

minLogLikelihoodImprovementIterating

- On increasing the value of minLogLikelihoodImprovementIterating from 1.0E-6, 0.1 and 1.0 respectively,
  - EM model takes more time for execution (64.53, 68.15 and 91.44 seconds respectively).
  - At the same time, Log likelihood value decreases from 0.51504, 0.23054 and 0.14542, respectively.

Number of clusters (6, 8 and 11 respectively) also increased on increasing minLogLikelihoodImprovementIterating.

3.3.6

- Training data provides the total number of clusters.
- The number of clusters is determined by considering some hypothesis.
- One of the known methods to determine the number of clusters is k = (n / 2) ^ 0.5

  where n -> num of data points

  Therefore,

  k = (n / 2) ^ 0.5

  = (3772 / 2) ^ 0.5

  = 43.43

  = 43

(Reference : https://stats.stackexchange.com/questions/55215/way-to-determine-best-number-of-clusters-weka/77218)

Attribute	Cluster 0 (0.24)	1 (0.06)	2 (0.16)	3 (0.03)	4 (0.02)	5 (0.5)
age						
mean	49.5944	52.7328	51.9588	47.6349	47.0337	52.988
std. dev.	19.7506	19.7055	17.6174	47.0711	16.8983	18.6518
sex						
F	679.1847	179.2248	454.3139	89.7395	56.3661	1177.171
M	237.2221	50.5307	134.2373	6.9611	21.3861	697.6628
[total]	916.4067	229.7554	588.5512	96.7006	77.7522	1874.8338
TSH						
mean	4.4569	14.8677	0.1109	3.1982	118.0143	1.2911
std. dev.	1.4166	9.0444	0.0958	2.1816	114.5321	0.7265
TT4						
mean	100.2776	89.0489	137.1098	172.0312	33.8318	105.3636
std. dev.	18.2157	33.9594	48.3878	28.9964	23.9997	22.8037

Cluster 1 has more Females than males, mostly under the age group of 50 to 55 with an average of 14.87 TSH and 89.05 TT4.

3.3.7

- K-Means execution is faster and quicker the EM. For example, K-Means with default parameters took 0.04 seconds to execute hypothyroid data. On the other hand, EM with default parameters took 44.26 seconds.
- Even though EM is slow it can handle both Nominal and Numeric data.
- So, it depends on the type of data chosen and the person or the decision maker himself.
- Here, for these clusters I prefer K-Means as it is faster and works based on Euclidean distance, whereas EM works on density probability.
- Also, in EM based on membership probability, an instance belongs to many clusters, whereas in K-Means an instance is the member of the single cluster.

3.3.8

- Females data is double than the Males data in a dataset.
- Maximum value of age is 455, which states that age recorded as incorrect. Which needs to be preprocessed before applying models on data.

Part 4: Association Finding

3.4.1

- supermarket1-small.arff has the data with labels f -138 and t - 43
- supermarket2-small.arff has the data with label t – 299 and has no f label.

3.4.2

- Initially, the unwanted department attributes are removed.
- Then the Attribute Evaluator is set to ClassifierSubsetEval along with J48 Classifier and the Best First Search algorithm is used to find the variables or attributes required for running the Apriori algorithm with default filters.
- The obtained attributes are 1, 2, 5, 6, 8, 15, 19, 24, 27, 28, 30, 31, 35, 46, 49, 50, 72 and 83 which are 18 in number.

- These 18 attributes are now sent to apriori algorithm and 10 best rules are obtained.

```
Best rules found:

 1. chickens=f 181 ==> coupons=f 181     <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 2. coupons=f 181 ==> chickens=f 181     <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 3. flowers=f 181 ==> coupons=f 181     <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 4. coupons=f 181 ==> flowers=f 181     <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 5. flowers=f 181 ==> chickens=f 181     <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 6. chickens=f 181 ==> flowers=f 181     <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 7. chickens=f flowers=f 181 ==> coupons=f 181     <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 8. coupons=f flowers=f 181 ==> chickens=f 181     <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 9. coupons=f chickens=f 181 ==> flowers=f 181     <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
10. flowers=f 181 ==> coupons=f chickens=f 181     <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
```

For example: From the rule “chickens=f 181 ==> coupons=f 181 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)”

“The customer who do not buy chickens mostly do not buy coupons” can be predicted with 100% Confidence.

- It has the best support of 95%.

3.4.3

Metric Type	<confidence>: 0.9	<lift>: 1.1	<leverage>: 0.1	<conviction>: 1.1
Num of Rules	10	15	12	16
Support	0.95 or 95%	0.5 or 50%	0.35 or 35%	0.85 or 85%
Example	From the rule “chickens=f 181 ==> coupons=f 181 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)” “The customer who do not buy chickens mostly do not buy coupons” can be predicted with 100% Confidence.	From the rule laundry needs=f 107 ==> baby needs=f dishcloths-scour=f health food other=f 90 conf:(0.84) < lift:(1.17)> lev:(0.07) [13] conv:(1.67)” “The customer who do not buy laundry needs are more likely do not buy baby needs and who do not buy dishcloth-scour also do not buy health food other” can be predicted with 84% Confidence and with lift 1.17.	From the rule “dishcloths-scour = f laundry needs = f 103 ==> coffee = f tissues-paper prd = f 63 conf:(0.61) lift:(1.5) < lev:(0.12) [20]> conv:(1.49)” “The customer who do not buy dishcloths-scour and laundry needs are more likely do not coffee and tissues-paper prd” can be predicted with 61% Confidence, lift 1.5 and leverage 0.12 [20].	For example: From the rule “fuels-garden aids=f cough-cold-pain=f 156 ==> preserving needs=f 154 conf:(0.99) lift:(1.01) lev:(0.01) [1] < conv:(1.15)>” “The customer who do not buy fuels-garden aids and mostly do not buy health food” can be predicted with 96% Confidence, lift 1.03, lev 0.02 and conv 1.43.

- The support is high for the metric type confidence which implies the number of frequency of items that appear in the data. Later followed by metric of type conviction with the best rules 16.
- The confidence on an average equal to 90%. This suggests that the frequency of the rule is 90%.
- Lift which shows the performance of the association is on an average 1.1 with a support of 50%.

3.4.4

The same process is applied on the second data set and the unnecessary attributes are removed. The apriori algorithm is applied on the data set with default metrics and attributes. The algorithm does not return any results on metric type. This might be due to a single label (t) in the data and hence the algorithm has no confidence on the data predictions.

3.4.5

Metric Type	<lift>: 1.1	<conviction>: 1.1
Num of Rules	10	20
Support	0.2 or 20%	0.2 or 20%

Example	From the rule "baking needs=t 757 ==> jams-spreads=t 265 conf:(0.35) < lift:(1.25)> lev:(0.04) [52] conv:(1.11)" "The customer who buys baking needs also buys jams-spreads" can be predicted with 35% Confidence and with lift 1.25.	For example: From the rule "jams-spreads=t cheese=t 181 ==> baking needs=t 144 conf:(0.8) lift:(1.35) lev:(0.03) [37] < conv:(1.95)>" "The customer who buys jams-spreads also buys baking needs" can be predicted with 80% Confidence and with lift 1.35, lev 0.03[37] and conv 1.95.

- The support is least for all the metric types.
- The confidence on an average equal to 35% when lift and 75% when convition. This low confidence might be due to single data label as mentioned previously.
- Lift which shows the performance of the association is on an average 1.1.

3.4.6

FP Growth:

supermarket1-small.arff

On running it gave 3 rules.

```
FPGrowth found 3 rules (displaying top 3)

1. [juice-sat-cord-ms=t, tissues-paper prd=t, pet foods=t]: 24 ==> [biscuits=t]: 23   <conf:(0.96)> lift:(1.45) lev:(0.04) conv:(4.04)
2. [biscuits=t, juice-sat-cord-ms=t, cleaners-polishers=t]: 21 ==> [tissues-paper prd=t]: 19   <conf:(0.9)> lift:(2) lev:(0.05) conv:(3.83)
3. [party snack foods=t, tissues-paper prd=t, pet foods=t]: 21 ==> [biscuits=t]: 19   <conf:(0.9)> lift:(1.36) lev:(0.03) conv:(2.36)
```

For example: From the rule "[juice-sat-cord-ms=t, tissues-paper prd=t, pet foods=t]: 24 ==> [biscuits=t]: 23 <conf:(0.96)> lift:(1.45) lev:(0.04) conv:(4.04)"

"The customer who buys juice-sat-cord-ms, tissues-paper, pet foods also buys biscuits" can be predicted with 96% Confidence, lift 1.45, leverage 0.04, conviction 4.04.

supermarket2-small.arff

The algorithm returned "No results found".

FilteredAssociator

supermarket1-small.arff

```
Best rules found:

 1. chickens=f 181 ==> coupons=f 181     <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 2. coupons=f 181 ==> chickens=f 181     <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 3. flowers=f 181 ==> coupons=f 181     <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 4. coupons=f 181 ==> flowers=f 181     <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 5. flowers=f 181 ==> chickens=f 181     <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 6. chickens=f 181 ==> flowers=f 181     <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 7. chickens=f flowers=f 181 ==> coupons=f 181     <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 8. coupons=f flowers=f 181 ==> chickens=f 181     <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 9. coupons=f chickens=f 181 ==> flowers=f 181     <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
10. flowers=f 181 ==> coupons=f chickens=f 181     <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
```

For example: From the rule "chickens=f 181 ==> coupons=f 181 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)"

"The customer who do not buy chickens mostly doesn't buy coupons" can be predicted with 100% Confidence, lift 1, leverage 0, conviction 4.

supermarket2-small.arff

Since the associator is Apriori by default and run on a default metric Confidence, this algorithm does not return any rules.

3.4.7

- On the dataset supermarket1-small.arff both Apriori and Associator Model which is related to Apriori gave the similar rule of "chickens=f 181 ==> coupons=f 181 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)" with 100% confidence and the similar readings of each metric.
- This will be same when the algorithms are run on the supermarket2-small.arff dataset also.
- This is because the associator for the model is also Apriori.
- Coming to the second data set the support and confidence of the algorithms is far too low. This might be due to the reason of single label in supermarket2-small.arff, where there are two labels in supermarket1-small.arff.
- Also, the FpGrowth returned few rules for both the data sets. 3 for the former and two for the later.

3.4.8

Yes.

- The numerical attributes are removed as association cannot be perform on numeric data.
- Also, the remaining data of the hypothyroid data set is nominal. So, Filtered Association is applicable.

```
Best rules found:

 1. hypopituitary=f 3771 ==> TBG measured=f 3771    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 2. lithium=f 3754 ==> TBG measured=f 3754    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 3. lithium=f hypopituitary=f 3753 ==> TBG measured=f 3753    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 4. goitre=f 3738 ==> TBG measured=f 3738    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 5. goitre=f hypopituitary=f 3737 ==> TBG measured=f 3737    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 6. on antithyroid medication=f 3729 ==> TBG measured=f 3729    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 7. on antithyroid medication=f hypopituitary=f 3728 ==> TBG measured=f 3728    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 8. query on thyroxine=f 3722 ==> hypopituitary=f 3722    <conf:(1)> lift:(1) lev:(0) [0] conv:(0.99)
 9. query on thyroxine=f 3722 ==> TBG measured=f 3722    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
10. query on thyroxine=f TBG measured=f 3722 ==> hypopituitary=f 3722    <conf:(1)> lift:(1) lev:(0) [0] conv:(0.99)
```

For example: From the rule “hypopituitary=f 3771 ==> TBG measured=f 3771 <conf:(1)> lift:(1) lev:(0) [0] conv:(0)”

“The patients who does not have hypopituitary are more likely to be measured TBG as False” can be predicted with 100% Confidence, lift 0, lev 0 and conv 0.

- All the rules have 100% confidence. Therefore, the associator predicted the most accurate results on the hypothyroid data.
- It has a support of 95% which indicates that the items appear more frequently in the data.

4.1

Team Members:	**Contribution**
1. Sri Venkata Manideepu Maddipati (S3820822)	50%
2. Vishal Patnaik Damodarpatruni (S3811521)	50%

We both worked equally on each part of the assignment. We shared our knowledge on each concept of Data Mining covered in the assignment and did our best in answering each question.