International Journal of Recent Development in Computer Technology & Software Applications
Vol. 2, Issue 1 - 2018

A COMPARATIVE REVIEW STUDY FOR THE EFFICIENCY OF DIFFERENT ALGORITHMS IN DATA MINING

VISHAL DUTT*, DEEPALI JAIN*, POOJA TRIPATHI*, NEETIKA SHARMA*, VIBHUTI BANSAL*

ABSTRACT

Data mining is completely based on mathematical rigors and algorithms in order to get the specific pattern and better classifications. Different algorithms are applied over datasets and having same parameters in order to get more accurate results. Our work primarily focuses on results obtained after applying different algorithms on datasets. This Chapter focuses on different algorithms and methodologies along with their implementations results. The work has been applied for getting better accuracy for the algorithms comparatively with respect to the results generated over the datasets with same or different parameters. This chapter aims at getting the specific pattern for getting better classifications with the application results of different algorithms.

C4.5 ALGORITHM

INTRODUCTION OF C4.5 ALGORITHM

It is a decision tree generation algorithm which is introduced by Ross Quinlan. It is an extension of the ID3 algorithm. [1] We can perform classification on the dataset using the C4.5 algorithm by generating the Decision Tree, and therefore, C4.5 is also called a statistical classifier.

Algorithm

Step 1: all the samples in the list belongs to some class.

Step 2: some of the features of sample provide any information gain.

a. Check for the base class (empty class).
b. For each attribute, 'A', find the normalized important gain ratio.

Step 3: let 'A' and 'B' the attribute with the highest important gain value.

Step 4: let's create a decision node which splits 'A' Best node.

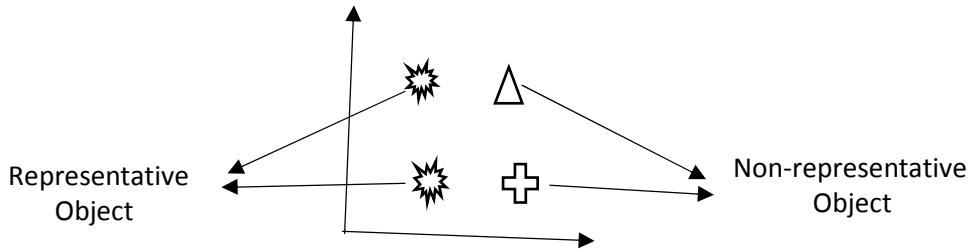
Step 5: reoccurrences on sub list obtained by splitting and add new nodes as children of the existing node.

K-MEDOID ALGORITHM

INTRODUCTION OF MEDOID ALGORITHM

The basis of a k-medoid algorithm is to group to the objects into k cluster. The initial representative object is choosing randomly them. We have to calculate the distance of the other data point from that representative element. This algorithm performs k partition for n object.[3] The quality of resulting clustering is calculated for each such combination.

* MCA Scholar, MDS University, Ajmer, India. ***Correspondence E-mail Id:*** editor@eurekajournals.com



Algorithm

Input: D dataset of n objects, k the no. of clusters.

Output: A set of k clusters.

Step 1: choose k object in D dataset randomly & initially.

Step 2: repeat.

Step 3: assign each remaining object to the cluster with a nearest representative object.

Step 4: select arbitrary to non-representative object.

Step 5: calculate the total cost and swap representative object with the non-representative object.

Step 6: if s<0 then swap object to object.

Step 7: repeat step suntil no change in the dataset.

PINCER SEARCH ALGORITHM

INTRODUCTION OF PINCER SEARCH ALGORITHM

It is used to get frequent item sets. In this method, the steps can be approached from top to bottom and bottom-to-top as well.[7] The subsets of frequent itemsets are also frequent.

Algorithm

Step 1: firstly, scan the dataset D for the transaction.

Step 2: where the candidate sets can be-

(L=0; K=1{}{i} (where I € item set))

Step 3: set MFCS = {1, 2,…… to n items}. // MFCS(Maximum Frequent Candidate Set)

Step 4: while (C_K = 0) read dataset count support for C_K.

Step 5: removing frequent item set from MFCS and add them in MFS (Minimum Frequent Set).

Step 6: Determine frequent set and infrequent set(S_K).

Step 7: useS_K to generate new item set and update MFS.

Step 8: K = K+1 return MFS;

Step 9: algorithm terminated when MFS = MFCS.

This Algorithm work on the basis of S input. MDCS is termed as total support of the items present in the database. After scanning MFCS, a new item set database is generated which is called MFS. Which holds all the frequent item sets. [2]

A motive of the algorithm is to generate maximum frequent item set possible where the minimum support is user define.

FP-GROWTH ALGORITHM

INTRODUCTION OF FP-GROWTH ALGORITHM

It is a powerful technique for searching the frequent pattern in the given database.[8] This algorithm proceeds in two steps:-

- In the first pass, it will perform the following operations:-

Step 1:

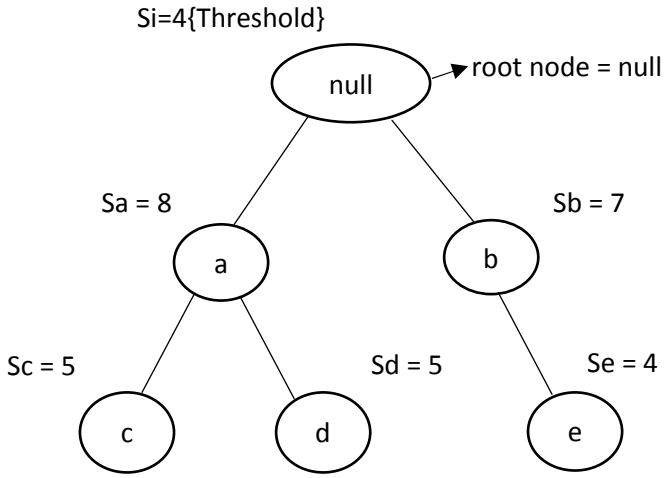
a) Scan the database and search the support for every item.
b) Remove the infrequent dataset and sort in decreasing order.

Step 2:

a) Build a compact data structure for FP-Tree(FP-Tree generation). Extract frequent item set directly from FP-Tree representation.

Example:-

T_{ID}	Items
1	ab
2	bcd
3	acdc
4	ade
5	abc
6	abcd
7	ab
8	abb
9	bcc
10	ac



- **Best Case:** All transaction have same no. of items.
- **Worst Case:** All transaction has a unique set of item.

BORDER ALGORITHM

INTRODUCTION OF BORDER ALGORITHM

This Algorithm proposed a method for searching frequent item set as in case of an A-prior algorithm, but there is a difference in the approach of this algorithm. [2]

The bigger dataset D is not frequent whereas the subset of the bigger database is frequent item set.[5] It uses the border method to set a threshold value which is no frequent.

Algorithm

Suppose item set will be generated as:

D – {a, b, c, d}.

b the bigger database and D – (a, b) is a subset of D.

Now using border algorithm we have to set a threshold value or minimum support value, which is called a border. In the context of D database, is supposed to be frequent but it is not necessary that the D is also frequent. This is not efficient as Apriori algorithm because it is not applicable to every dataset. Its limitation is greater than all the other applied algorithms for finding the frequent item set. [1] This algorithm usually applied for smaller datasets.

BOAT ALGORITHM

INTRODUCTION OF BOAT ALGORITHM

This is the first algorithm of a decision tree that produces various levels of the tree in a single scan. This is fastest than best existing algorithms because all other algorithms have need to be max more than one scan to a dataset.[11] It is used for construct the same decision tree but the complexity is very much less as compared to other algorithms.

Boat algorithm improved the functionality of existing algorithm. It is the first algorithm for maintenance of decision tree incrementally when the training dataset changing dynamically.[10]

Algorithm

Step 1: read the given sample of bigger dataset D.

Step 2: learn the m nodes of given dataset.

Step 3: keep any subset of node m exactly the same.

Step 4: verify the subset and scanning is performed.

Step 5: if failed repeat the process.

Example:

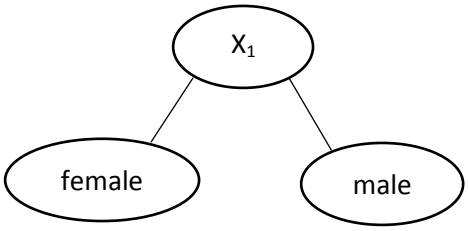
age between 61, 65

f age >= 61 than

female;

if age <= 65 than

male;



APRIORI ALGORITHM

INTRODUCTION OF APRIORI ALGORITHM

This algorithm is introduced by R. Agarwal and R. Shreekant in 1994 for frequent itemsets for association rules. [3] This algorithm is based on the fact that algorithm uses prior knowledge of the frequent item set.

This algorithm engages an attractive approach. In this algorithm, nitemsets are used to explore n+1 item sets. [15]

Algorithm

Input: dataset D for a transaction.

Output: frequent item set in D.

Process

L_n = find frequent items(D).

for(n=3;$L_{n-1} \neq 0$;n++)

for each transaction scan(D)

c_t= subset of C//end

for each candidate c € C

c.count++;

L_n= count >= min_supoort;

return L = L_n;//end

for each set

join P_1 |><| P_2 |><| P_3 |><|……..Pn items

if(L1 = L2)(L2 ≠ L3)……n items

prune(delete);

(min_supoort) (prune stop)

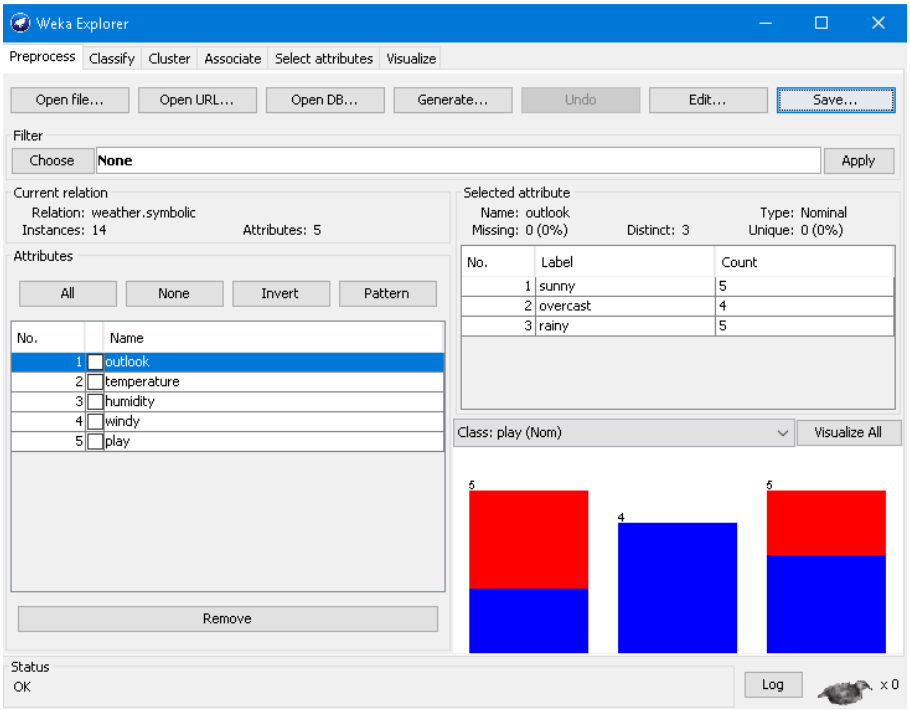
return;

end.

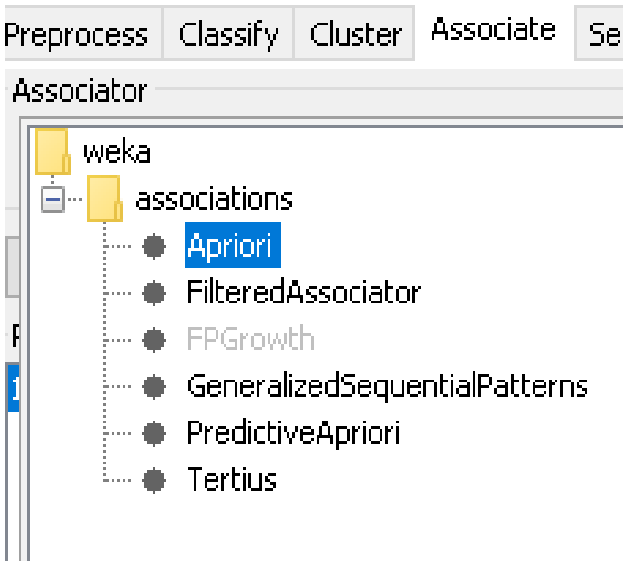
Note: Pruning is discarded less gain ration.

Example:

We will select process menu into menu bar and will choose open file tab then we will select a CSV file into given list like asweather.nominal.arff and will click on open button. So the implementation of our dataset are shown like this: [13]



We will select associate menu into the menu bar and then we will click on choose button and extend association base folder, and will select Apriori algorithm into given list and will click on close button. Finally, we will click on start button for implementation.



This is an "arff (Attribute-Relation File Format)" file of our dataset. The algorithm only applies to this file.

weather.nominal.arff

```
@relation weather.symbolic

@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,mild,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no
```

Result of Apriori algorithm:

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Relation:     weather.symbolic
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.15 (2 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 17

Generated sets of large itemsets:

Size of set of large itemsets L(1): 12

Size of set of large itemsets L(2): 47

Size of set of large itemsets L(3): 39

Size of set of large itemsets L(4): 6

Best rules found:
```


1. outlook=overcast 4 ==> play=yes 4 conf:(1)
2. temperature=cool 4 ==> humidity=normal 4 conf:(1)
3. humidity=normal windy=FALSE 4 ==> play=yes 4 conf:(1)
4. outlook=sunny play=no 3 ==> humidity=high 3 conf:(1)
5. outlook=sunny humidity=high 3 ==> play=no 3 conf:(1)
6. outlook=rainy play=yes 3 ==> windy=FALSE 3 conf:(1)
7. outlook=rainy windy=FALSE 3 ==> play=yes 3 conf:(1)
8. temperature=cool play=yes 3 ==> humidity=normal 3 conf:(1)
9. outlook=sunny temperature=hot 2 ==> humidity=high 2 conf:(1)
10. temperature=hot play=no 2 ==> outlook=sunny 2 conf:(1)

This result shows that there is 5 attribute to find out weather report such as outlook, temperature, humidity, windy, play and we have also respective values to them. It will calculate minimum support. It finds out best rules for dataset by applying the appropriate algorithm. The Number of cycle defines that how many times the same rule has repeated.

ID3 ALGORITHM

INTRODUCTION OF ID3 ALGORITHM

Advanced version is a C4.5 algorithm. In decision tree learning algorithm, the decision tree is generated from a dataset.[12] This is used in machine learning and neural language processing. It begins with original set 'S' as a root node.

It iterates through each set of attributes used in datasets and entropy is calculated. Select that type of attribute who having lowest entropy.[14] Now, 'S' is split by selecting attributes. This will produce the bigger dataset of the database. This algorithm repeats itself.

Considering the attributes, where repetition can stop:

- When there is no example in the subset.
- When there are no attributes to select.
- If every element belongs to the same class.

This algorithm does not ensure an optional solution. It follows greedy approach for selecting the issued attribute.

Formula: $H(J) = \sum \frac{P(x) \log_2 P(x)}{n€x}$ [where H(j) = 0]

where,

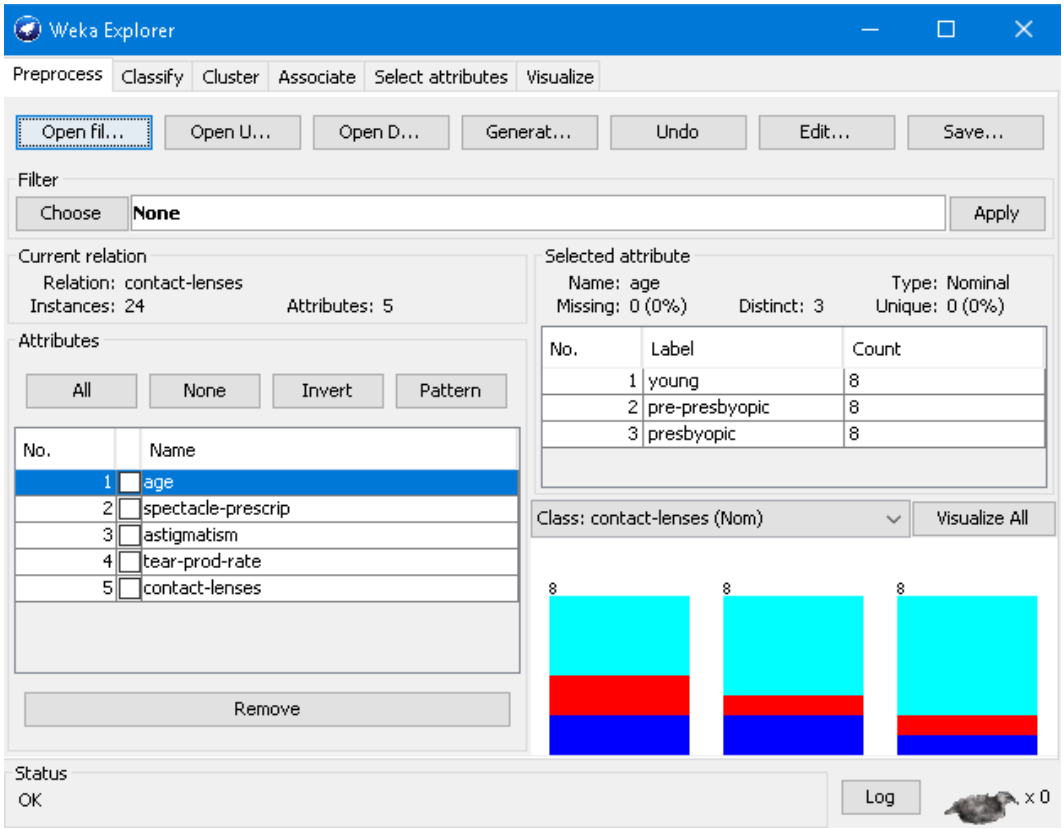
n = set of classes.

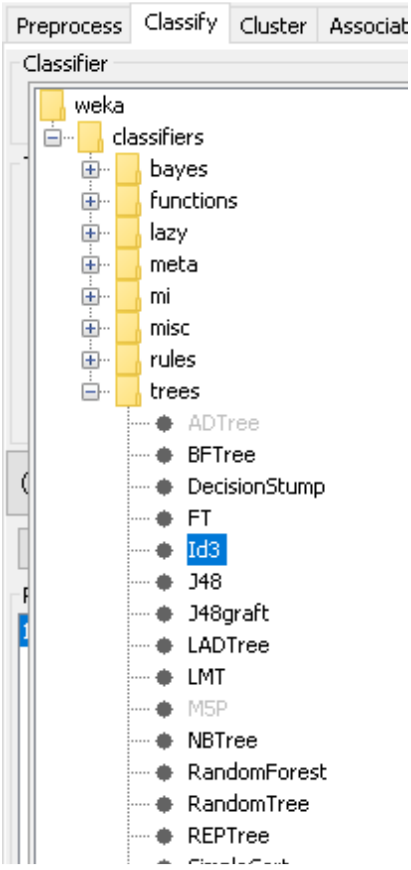
P(x) = the proportion of number of element in class x.

Example:

We will select process menu into menu bar and will choose open file tab then we will select a CSV file into given list like as contact-lenses.arff and will click on open button. So the implementation of our dataset are shown like this: [6]

We will select classify menu into the menu bar and then we will click on choose button and extend trees base folder, and will select ID3 algorithm into given list and will click on close button. Finally, we will click on start button for implementation.





contact-lenses.arff

```
% 1. Title: Database for fitting contact lenses
%
% 2. Sources:
%    (a) Cendrowska, J. "PRISM: An algorithm for inducing modular rules",
%        International Journal of Man-Machine Studies, 1987, 27, 349-370
%    (b) Donor: Benoit Julien (Julien@ce.cmu.edu)
%    (c) Date: 1 August 1990
%
% 3. Past Usage:
%    1. See above.
%    2. Witten, I. H. & MacDonald, B. A. (1988). Using concept
%       learning for knowledge acquisition. International Journal of
%       Man-Machine Studies, 27, (pp. 349-370).
%
% Notes:  This database is complete (all possible combinations of
%       attribute-value pairs are represented).
%
%       Each instance is complete and correct.
%
%       9 rules cover the training set.
%
% 4. Relevant Information Paragraph:
%   The examples are complete and noise free.
%   The examples highly simplified the problem. The attributes do not
%   fully describe all the factors affecting the decision as to which type,
%   if any, to fit.
%
% 5. Number of Instances: 24
%
% 6. Number of Attributes: 4 (all nominal)
%
% 7. Attribute Information:
%   -- 3 Classes
%    1: the patient should be fitted with hard contact lenses,
%    2: the patient should be fitted with soft contact lenses.
%    1: the patient should not be fitted with contact lenses.
%
%    1. age of the patient: (1) young, (2) pre-presbyopic, (3) presbyopic
%    2. spectacle prescription:  (1) myope, (2) hypermetrope
%    3. astigmatic:    (1) no, (2) yes
%    4. tear production rate:  (1) reduced, (2) normal
%
% 8. Number of Missing Attribute Values:  0
%
% 9. Class Distribution:
%   1. hard contact lenses: 4
%   2. soft contact lenses: 5
%   3. no contact lenses: 15

@relation contact-lenses

@attribute age                  {young, pre-presbyopic, presbyopic}
@attribute spectacle-prescrip   {myope, hypermetrope}
@attribute astigmatism          {no, yes}
@attribute tear-prod-rate   {reduced, normal}
@attribute contact-lenses   {soft, hard, none}

@data
%
% 24 instances
%
young,myope,no,reduced,none
young,myope,no,normal,soft
young,myope,yes,reduced,none
young,myope,yes,normal,hard
young,hypermetrope,no,reduced,none
young,hypermetrope,no,normal,soft
young,hypermetrope,yes,reduced,none
young,hypermetrope,yes,normal,hard
pre-presbyopic,myope,no,reduced,none
pre-presbyopic,myope,no,normal,soft
pre-presbyopic,myope,yes,reduced,none
```


pre-presbyopic,myope,yes,normal,hard
pre-presbyopic,hypermetrope,no,reduced,none
pre-presbyopic,hypermetrope,no,normal,soft
pre-presbyopic,hypermetrope,yes,reduced,none
pre-presbyopic,hypermetrope,yes,normal,none
presbyopic,myope,no,reduced,none
presbyopic,myope,no,normal,none
presbyopic,myope,yes,reduced,none
presbyopic,myope,yes,normal,hard
presbyopic,hypermetrope,no,reduced,none
presbyopic,hypermetrope,no,normal,soft
presbyopic,hypermetrope,yes,reduced,none
presbyopic,hypermetrope,yes,normal,none

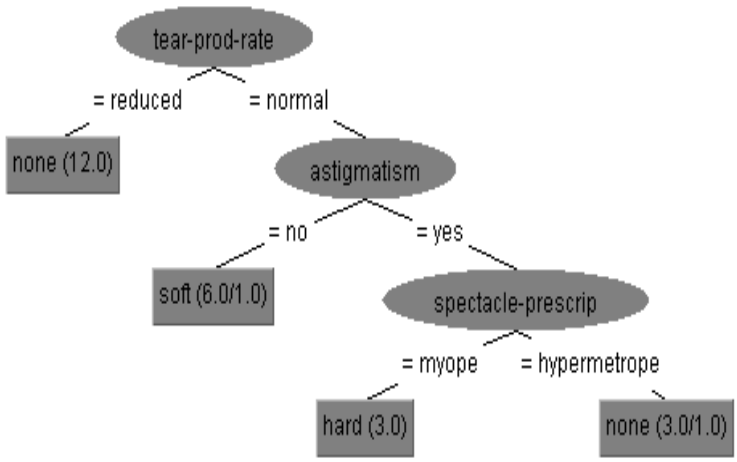
Result of ID3 Algorithm:

```
=== Run information ===

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     contact-lenses
Instances:    24
Attributes:   5
              age
              spectacle-prescrip
              astigmatism
              tear-prod-rate
              contact-lenses
Test mode:10-fold cross-validation

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

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

tear-prod-rate = reduced: none (12.0)
tear-prod-rate = normal
|   astigmatism = no: soft (6.0/1.0)
|   astigmatism = yes
|   |   spectacle-prescrip = myope: hard (3.0)
|   |   spectacle-prescrip = hypermetrope: none (3.0/1.0)

Number of Leaves  :         4
```



```
Size of the tree :      7


Time taken to build model: 0 seconds

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

Correctly Classified Instances          20               83.3333 %
Incorrectly Classified Instances         4               16.6667 %
Kappa statistic                          0.71
Mean absolute error                      0.15
Root mean squared error                  0.3249
Relative absolute error                 39.7059 %
Root relative squared error             74.3898 %
Total Number of Instances               24

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  ROC Area  Class
                 1         0.053      0.833     1         0.909      0.947    soft
                 0.75      0.1        0.6       0.75      0.667      0.813    hard
                 0.8       0.111      0.923     0.8       0.857      0.811    none
Weighted Avg.    0.833     0.097      0.851     0.833     0.836      0.84

=== Confusion Matrix ===

  a  b  c   <-- classified as
 5  0  0 |  a = soft
 0  3  1 |  b = hard
 1  2 12 |  c = none
```

This is a decision tree algorithm. So it also produces the virtualized tree shown below:



This result shows that there is 5 attribute to find out contact-lenses report such as age, spectacle-prescription, astigmatism, tear-prod-rate, contact-lenses and we have also respective values to them. It has tree size is 7 it means nodes are 7 and number of leaves are 4 it means branches are 4. It describes accuracy in detailed by the class.

CONCLUSION

This chapter has concluded with comparative results of primary algorithms of data mining and working efficiencies of different algorithms of data mining and their results.Right from FP tree to Apriori algorithms has been applied over same datasets which resulted in better classifications figure and better pattern analysis.Our work primarily goals at getting better efficiency results for different algorithms with respects to their results and finally comparative results have been generated using Weka tool.The future scope can be using the same approach for different dynamic datasets for getting much better classifications and get solves huge data in more efficient manner.

REFERENCES

[1]. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules.

[2]. AhmedS,CoenenF,LengPH (2006) Tree-basedpartitioningofdateforassociationrule mining.

[3]. Banerjee A, Merugu S, Dhillon I, Ghosh J (2005) Clustering with Bregman divergences.

[4]. Devroye L, Gyorfi L, Lugosi G (1996) Aprobabilistic theory of pattern recognition.

[5]. DhillonI S, GuanY, Kulis B (2004) Kernelk-means: spectralclu stering and normalizedcuts.

[6]. Dietterich TG (1997) Machine learning: Four current directions.

[7]. Domingos P (1999) MetaCost: A general method for making classifiers cost-sensitive. In: Proceedings of the fifth international conference on knowledge discovery and data mining.

[8]. McLachlan GJ, Krishnan T (1997) The EM algorithm and extensions. Wiley, New York

[9]. McLachlan GJ, Peel D (2000) Finite Mixture Models. Wiley, New York

[10]. Messenger RC, Mandell ML (1972) A model search technique for predictive nominal scale multivariate analysis.

[11]. Morishita S, Sese J (2000) Traversing lattice itemset with statistical metric pruning.

[12]. Olshen R (2001) A conversation with Leo Breiman.

[13]. Page L, Brin S, Motwami R, Winograd T (1999) The PageRank citation ranking: bringing order to the Web.

[14]. Quinlan JR (1993) C4.5: Programs for machine learning.

[15]. Washio T, Nakanishi K, Motoda H (2005) Association rules based on levels subspace clustering.

[16]. Wasserman S, Raust K (1994) Social network analysis.

