CHAPTER-9
COMPARATIVE STUDY OF DIFFERENT DATA MINING PREDICTION ALGORITHMS
CHAPTER 9
COMPARATIVE STUDY OF DIFFERENT DATA MINING PREDICTION ALGORITHMS

9.1 Introduction

The major objective of this chapter is to make a comparative study of existing data mining algorithms and their strengths and weaknesses in different applications. The main algorithms involved in this study include classification using decision tree, clustering algorithm, Apriori algorithm and association rules. All the algorithms were analyzed with different data sets which were collected from real life applications.

9.2 Decision tree

Decision tree is a tree shaped arrangement that stand for sets of decisions. These decisions produce rules for the categorization of a dataset. The purpose of the decision tree is to identify certain classes of objects, such as software modules that are likely to be fault prone or costly to develop. The decision tree leaf nodes contain a “Yes” and “No” to indicate whether an object is likely to be in a certain class based on historical data. Decision trees are useful structures since they are straightforward to build and interpret. They use different object attributes to classify different subsets of objects. They can also be naturally decomposed to form a set of production rules. The rules may then be used to form the basis of an expert system. Since the tree generation process is automated, the cost of knowledge acquisition is greatly reduced.

Training set

In distinctive supervised learning situation, a training set is known and the objective is to form an explanation that can be used to forecast unknown examples previously. The training dataset can be explained in many ways. It is described as an occurrence of a certain schema. An instance is a collection of tuples that may have duplicates. Every tuple is described by a vector of attributed values. The schema provides the explanation of the attributes and their domains. Attributes is typically of nominal or numeric types. [grumbach and milo (1996)].

9.3 Definition of the classification problem

The machine learning community was among the first to begin the problem of concept learning. Concepts are intellectual categories for objects, ideas or events that can have a common set of characteristics. According to marek kretowsi “each concept can be viewed as relating some subset of objects or events defined over a larger set”.

210
An idea can be stared as a function from the occurrence space to that Boolean set, that is C: X→[-1,1]. Otherwise any one can pass on a notion c as a subset of X, i.e. {x∈X:C(x)=1}.

A concept can be officially regarded as a task from the set of all probable cases to the Boolean set {True, False}. The communities, such as the KDD community desire to deal with a straightforward extension of concept learning, called as the classification problem.

We look for a task that maps the set of different possible examples into a already defined set of class labels that are not restricted to the Boolean set.

9.4 Algorithmic framework for decision tree inducers

Decision tree inducers are algorithms that involuntarily build a decision tree from a dataset. Naturally the aim is to find the best decision tree by reducing the simplification error. Still, other target functions such as reducing the number of nodes or reducing the average depth can also be described.

Optimal decision tree algorithms are feasible only in minute problems. Therefore, heuristics methods are necessary for solving the above problem. These methods can be classified into two groups: bottom up and top down with preference to the second group.

There is a range of top down decision trees generates such as ID3 C4.5, CART. Some in inducers consist of two intangible phases: Pruning (C4.5 and CART) and Growing. Other inducers execute only the growing phase. These algorithms are insatiable by nature and build the decision tree in a top down, repetitive manner (it is called divide and conquer). In every iteration of the algorithm consists the division of the training dataset using the result of separate input aspects. The choice of the most suitable attribute is prepared according to a few splitting measures. After the choice of a suitable split, each node auxiliary subdivides the training dataset into lesser subsets, till a stopping principle is satisfied.

Stopping criteria

The growing phase continues till a stopping condition is triggered. The below conditions are widespread stopping rules:

1. All occurrences in the training dataset feel right to a specific value of y.
2. The utmost tree depth has been achieved.
3. The amount of cases in the leaf node is lesser than the minimum amount of cases meant for parent nodes.
4. If the node had been split, the amount of cases within one or more child nodes should be lesser than the minimum amount of child nodes.

5. The superlative splitting condition is not greater than a convinced threshold

**Algorithm for decision tree**

TreeGrowing(S,A,Y SplittingCriterion, StopCriterion)

Where:
- S Training Set
- A Input Feature Set
- Y Target Feature
- Splitting criterion the method for evaluating a certain split
- Stop criterion the criteria to stop the growing process
- Create a new tree T through a single root node.
- IF Stopping Criterion(S) THEN
  - Mark T as a leaf with the most frequent value of y in S as a label.
- ELSE
  - For every element, ai €A find a that obtain the best splitting Criterian(Ai,S).
    - Label t with a
    - FOR each outcome Vi of a:
      - Set Subtreei=TreeGrowing(σa=viS,A,y).
      - Connect the root node of T to Subtree with an edge that is labeled as vi
  - END FOR
- END IF
- RETURN Tree Pruning (S,T,and y)

**Tree Pruning (S,T,y)**

Where:
- S Training Set
- Y object Feature
- T the tree to be puruned
- DO
  - Select a node t in T such that purning it
    - Maximally improve some evaluation criteria
  - IF t_=ϕ THEN T=pruned(T,t)
- UNTIL t=ϕ
- RETURN T

**Construction of decision trees**

Most of the algorithms that have been created for learning decision trees are different on a core algorithm that utilizes a top down, greedy search during the space of probable decision trees. Decision tree programs build a decision tree T from a set of training cases. J. Ross Quinlan initially extended ID3 at the Sydney University. He first explained ID3 in 1975 in a Machine Learning book vol. 1, no. 1. The Concept Learning System (CLS) algorithm is used in ID3.
ID3 investigates through the elements of the training instances and hauls out the attribute that finest separates the given examples. If the element is perfectly classifies the trained sets then ID3 discontinues, otherwise it recursively on the \(m\) (where \(m=\) number of probable values of an element) divided subsets to obtain their “best” attribute. This algorithm utilizes a greedy search, that is, it chooses the best attribute and never walks back to reconsider earlier choices.

The main focus of the decision tree rising algorithm is choosing the attributes to test at every node in the tree. The choice of the attribute with the more inhomogeneous class sharing the algorithm utilizes the concept of entropy.

A well quantitative compute of the value of an attribute is a arithmetical property known as information gain that gain measure how well a known attribute splits the training samples according to their required classification. This compute is used to choose among the candidate elements at each step even as growing the tree.

**Issues within data mining by decision trees**

Practical issues in erudition decision trees consists of, determining how intensely to develop the decision tree, choosing an appropriate attribute selection measure, handing attributes with differing costs, handling continuous attributes, handling training data with missing attribute values [21], and improving computational efficiency.

**9.5 Strength and Weakness of Decision Tree Methods**

The decision tree has its own advantages and limitations.

1. The strength of Decision tree Methods are:
   - The Decision trees are capable to generate understandable rules.
   - The Decision trees execute classification without necessitating much computation.
   - The Decision trees are able to handle both permanent and uncompromising variables.
   - The Decision trees provide a clear sign of that fields are most significant for classification or prediction.

2. The weakness of decision tree methods
   - The Decision trees are less suitable for inference tasks where the aim is to forecast the value of a permanent attribute.
   - The Decision trees are flat to errors in classification problems with more class and moderately small number of training samples.
Table 9.1 Strengths and weakness of decision tree

<table>
<thead>
<tr>
<th>Strengths</th>
<th>Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision tree provides clear suggestions of the fields are most significant for forecast or classification. So error rate is less</td>
<td>Decision trees typically require certain information of quantitative or arithmetic experience to complete the process correctly. Failing to correctly understand decision trees can guide to a confused outcome of business opportunities or decision possibilities. Decision trees are flat to errors in classification problems so many class and comparatively small number of training examples.</td>
</tr>
<tr>
<td>Decomposition is earlier as compared with other techniques. Decision trees perform classification exclusive of much computation.</td>
<td>It can also be hard to include variables on the decision tree, exclude replica information in a logical, consistent manner. The incapability to finish the decision tree using only one set of information can be rather difficult. Decision trees are fewer appropriate for estimation jobs where the aim is to forecast the value of a permanent attribute.</td>
</tr>
<tr>
<td>Represent the knowledge in the form of IF-THEN rules. Rules are easier for human to understand. Decision trees are able to produce understandable rules.</td>
<td>While imperfect information can generate difficulties in the decision tree process, more information can also be an issue.</td>
</tr>
<tr>
<td>Decision trees are able to hold both continuous and categorical variables.</td>
<td>Decision trees do not take care of nonrectangular regions. Most decision tree algorithms only check a single record at a time. It leads to rectangular categorization boxes that cannot communicate well with the original distribution of fields in the decision space.</td>
</tr>
</tbody>
</table>

The Decision tree can be computationally elite to train. The procedure of increasing a decision tree is computationally elite. At each node, every candidate splitting field should be stored before, to find out best split. In some algorithms, mixtures of fields are used and an exploration must be made for finest combining weights. Pruning algorithms could also be elite since several candidate sub trees should be formed and contrasted.

The Decision trees not look after non rectangular regions. Many decision tree algorithms observe a single field at a time. This directs to rectangular classification boxes may not communicate fine with the actual allocation of records in the decision space.

9.6 Clustering Algorithm

Clustering can be considered the most significant unsupervised learning problem. So, as every new problem of this kind, it deals with discovery a structure in a collection of unlabelled data. A classification of clustering could be “the method of organizing things into groups” whose members are related in some way”. A cluster is
thus a collection of “similar” objects between them and “dissimilar” with objects belonging to all other cluster. The goal of clustering is to resolve the basic grouping in a set of unlabeled data [124].

**Applications of Clustering**

Clustering algorithms are used in various fields, such as

- **Marketing:** finding group of customers with related behavior given a huge database of client data containing their functions and precedent buying records.
- **Libraries:** Book ordering and grouping.
- **Biology:** categorization of animals and plants give their features.
- **Insurance:** identifying group of motor insurance policy holders with high average claim cost, identify frauds.
- **City-planning:** identifying group of houses according to their house nature, price and geographical location.
- **Earthquake studies:** the clustering pragmatic earthquake epicenters to recognize dangerous zones.
- **World Wide Web:** document classification, clustering weblog information to find out group of similar access patterns.

**Requirements of clustering**

The major requirements that a clustering algorithm should assure are:

- Scalability
- Dealing with different types of attributes
- Discovering clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Ability to deal with noise and outliers
- Infectivity to order of input records
- High dimensionality
- Interpretability and usability

**Issues in clustering**

There are a number of issues with clustering. They are

- Existing clustering Methods do not deal with all necessities adequately and concurrently
- Dealing with large number of proportions and huge number of data items can be challenging as of time complexity.
- The efficiency of the technique depends on the classification of “distance” (for distance-based clustering).
- If a clear distance measure doesn’t survive we must “define” it, i.e. not always simple, particularly in multi-dimensional spaces.
- The end result of the clustering algorithm (in many cases it can be arbitrary itself) can interpret in different ways.

**9.7 Types of clustering**

1. **Partitioning algorithm**
   - Construct various separations and then assess them by some principle.
Partitioning method: construct partition of database D with n number of objects into a set of k clusters that optimizes the selected partitioning principle. t is global optimal.

E.g.: k-means clustering algorithm.

1.1 Strength and weakness of k-means methods

Strength
- Relatively efficient: n # objects, k # clusters and t # iterations, usually, k, t<<n.
- Frequently terminates at a local optimum.

Weakness
- Appropriate for only when mean is defined, concerning categorical data.
- Require to identify k, the amount of clusters, in advance.
- Not capable to hold outliers and noisy data.
- Not suitable to find out clusters among non-convex shapes.

2. Hierarchy Algorithm

Generate a hierarchy corrosion of the set of information (or objects) using specific principle. It uses the distance matrix as clustering criterion. This method does not need the amount of clusters k as an input, but needs a loss condition [124].

Eg:- AGNES(Agglomerative Nesting)

2.1 AGNES (Agglomerative Nesting)

AGNES was introduced by Kaufmann and Rousseau (1990) and it was implemented in statistical analysis packages.

Eg.- SPSS. AGNES uses the dissimilarity matrix and the single-Link method. Here the least dissimilarity nodes are combined. It goes in a non-descending fashion. Ultimately all the nodes belong to the same cluster.

3. Density-Based Algorithm

It is based on connectivity and density functions. Clustering be based on density (local cluster criterion), such like density-connected points. It can find out cluster of arbitrary shape and can easily handle noise. This algorithm desires density parameters as termination condition.

Eg: DBSCAN (Density based spatial clustering of applications with noise.)

4. Grid Based- Algorithm

This algorithm is based on a multiple-level granularity structure. This algorithm uses multi-resolution grid data structure.

Eg:- Egesting (a Statistical Information Grid approach)

The spatial area is separated into rectangular cells and there numerous levels of cells equivalent to different levels of decision. Each cell at a elevated level is
partitioned into a many smaller cells in the subsequently lower level. Statistical information of each cell is considered and stored earlier and is needed to answer queries. Parameters of superior level cells can be simply calculated from parameter of lower level cell. It uses a top-down method to answer spatial data inquiry. The algorithm starts from a pre-selected layer-typically with a small number of cells. The assurance interval is calculating for each cell in the current level. Remove the irrelevant cells from further deliberation. When the current layer is examined, precede to the next lower level. Replicate this process until the bottom layer reach [124].

Advantages:
☆ Easy to parallelize, incremental update, query-independent.
☆ O(g), where g is the number of grid cubicle at the buck level

Disadvantages:
✓ All the cluster boundaries are both vertical and horizontal, and no transverse boundary is detected.

5. Model-based Algorithm

Here a model is hypothesized for every cluster and the thought is to find the finest robust of that model to each other. An effort is made to optimize the robust between the data and some mathematical model. This algorithm mainly uses statistical and AI approach [125].

6. Conceptual Clustering

It is a form of clustering in machine learning. This type of clustering produce a classification method for a set of unlabeled items and finds attribute report for each concept (class). E.g.: COWEB

6.1 COWEB

It is admired & easy method of incremental conceptual learning which creates a hierarchical grouping in the form of a categorization tree. Each node mentions to a concept and holds a probabilistic report of that concept [126].

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relatively capable and easy to implement</td>
<td>Receptive to initialization</td>
</tr>
<tr>
<td>Terminate at local optimum</td>
<td>Limiting case of fixed data</td>
</tr>
<tr>
<td>Relate even large data sets</td>
<td>Difficult to balance with different numbers of clusters</td>
</tr>
<tr>
<td>The clusters are non-hierarchical and they do not overlap</td>
<td>Needs to identify the number of clusters in progress</td>
</tr>
<tr>
<td>With a large number of variables, may be Computationally faster</td>
<td>Unable to hold noisy or outliers</td>
</tr>
<tr>
<td>It may produce tighter clusters especially if the clusters are spherical</td>
<td>Not suitable to discover clusters with non-convex shapes</td>
</tr>
</tbody>
</table>
9.8 Association rules

This algorithm is finding common patterns, associations, correlations, or relational databases and other information repositories, fundamental structures among sets of items or objects in transaction databases [125].

✓ Examples.
  o Rule form: “Body Head [support, confidence]”.
  o Buys(x, ”computer”) buys(x,”software”) [0.5%,60%]
  o Major(x,”CS”) Takes(x,”DB”) grade(x,”A”) [1%,75%]

The two procedures of interestingness of association rule mining are Support and confidence wherever support is usefulness of discovered pattern and confidence is pertinacity of discovered pattern.

The rules are measured as exciting if they convince both a least support threshold & minimum confidence threshold.

It is two steps processes which include finding all frequent item sets and then generating strong association rules for frequent item sets.

The classification of frequent pattern mining:
  ➢ Completeness of patterns to be mined.
  ➢ Level of abstraction involved.
  ➢ Number of data dimension involved.
  ➢ Types of values handled and kinds of rules to be mined.

9.9 Apriori Algorithm - An efficient and scalable frequent item set mining method

The Apriori algorithm is a powerful algorithm for mining frequent item sets (the items sets which have least support) or Boolean association rules. The key concept included in this algorithm are frequent item sets (the items sets which have least support), Apriori Property (Any subset of frequent item set must be recurrent), join Operation (a set of candidate k-item sets is generated by union $L_{k-1}$ with itself, To find $L_k$), Prune Operation (Generate $L_k$, by pruning $C_k$ with minimum support count) etc, it uses the frequent item sets to generate association rules [124].

The Pseudo code of Apriori Algorithm

$C_k$: Candidate Itemset of size k
$L_k$: frequent itemset of size k
$L_1$: {frequent items} ;
For $(k=1; L_k \neq \emptyset; k++)$ do begin
  $C_{k+1}=Candidates$ generate from $L_k$ ;
end
For each transaction \( t \) in database does
Increment the count of every candidates in \( C_{k+1} \) that are contained in \( t \)
\[ L_{k+1} = \text{Candidate in } C_{k+1} \text{ with min\_support} \]
End, return \( \bigcup_k L_k \);

Methods to improve Apriori’s Efficiency

There are different methods to improve the efficiency of Apriori.

They are as follows

- Hash-based Itemset counting: A \( k \)-Itemset whose equivalent hashing bucket count is less than the threshold cannot be recurrent.
- Transaction reduction: A transaction which does not have any frequent \( k \)-Item set \( u \) is useless in subsequent scans.
- Partitioning: Partitioning the data to find candidate item sets-local frequent Item set, global candidate Item set.
- Sampling: mining on a subset of given data, a method to determine the completeness + least support threshold.
- Dynamic Item set counting: add new candidate item sets only where all of their subsets are expected to be frequent

Applications of Apriori Algorithm

The main applications of Apriori algorithm are

- Purchase Domain application- where it is used for profit maximization, time minimization and optimal layouts, conveyance shopping, combination purchasing bundling etc.
- Diagnosis- used for machine diagnosis as well as medical diagnosis
- Inventory management- to identify items using unique id statistics.
- Fraud Detection- to find forged credit card usage and stolen cards, money laundering activities, anomaly detection etc.

It can also be applicable to Market Basket data analysis, classification, cross-marketing, clustering, catalog design, etc [126]

<table>
<thead>
<tr>
<th>ADVANTAGES</th>
<th>LIMITATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>To look after the sequential patterns the Association rule algorithms are formed</td>
<td>By using the dependent variable Association rules do not identify logical patterns and the number of independent variables cannot be decreased by removing</td>
</tr>
<tr>
<td>The most relevant to association rules are the methods of data achievement and integration and integrity checks</td>
<td>If the information do not provide support and confidence of rule are correct then Association rules cannot be useful</td>
</tr>
</tbody>
</table>
### Table 9.4 Algorithm Validation Metrics - Classification Algorithm

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Formula</th>
<th>Male</th>
<th>Female</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision (P)</td>
<td>Correct Values/Prediction Values * 100</td>
<td>86.08</td>
<td>85.42</td>
<td>85.76</td>
</tr>
<tr>
<td>Recall (R)</td>
<td>Correct Values/Actual Values * 100</td>
<td>74.73</td>
<td>75.93</td>
<td>75.29</td>
</tr>
<tr>
<td>F-measure</td>
<td>$2^\times (P \times R)/(P + R)$</td>
<td>80.00</td>
<td>80.39</td>
<td>80.19</td>
</tr>
<tr>
<td>Error Rate</td>
<td>(Actual Values - Correct Values)/Actual Values * 100</td>
<td>25.27</td>
<td>24.07</td>
<td>24.71</td>
</tr>
<tr>
<td>Accuracy Rate</td>
<td>Correct Values/Actual Values * 100</td>
<td>74.73</td>
<td>75.93</td>
<td>75.29</td>
</tr>
</tbody>
</table>

### Table 9.5 Algorithm Validation Metrics - Clustering Algorithm

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Formula</th>
<th>Male</th>
<th>Female</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision (P)</td>
<td>Correct Values/Prediction Values * 100</td>
<td>91.08</td>
<td>84.21</td>
<td>87.70</td>
</tr>
<tr>
<td>Recall (R)</td>
<td>Correct Values/Actual Values * 100</td>
<td>78.57</td>
<td>79.01</td>
<td>78.78</td>
</tr>
<tr>
<td>F-measure</td>
<td>$2^\times (P \times R)/(P + R)$</td>
<td>84.37</td>
<td>81.53</td>
<td>83.00</td>
</tr>
<tr>
<td>Error Rate</td>
<td>(Actual Values - Correct Values)/Actual Values * 100</td>
<td>21.43</td>
<td>20.99</td>
<td>21.22</td>
</tr>
<tr>
<td>Accuracy Rate</td>
<td>Correct Values/Actual Values * 100</td>
<td>78.57</td>
<td>79.01</td>
<td>78.78</td>
</tr>
</tbody>
</table>

### Table 9.6 Algorithm Validation Metrics - Apriori Algorithm

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Formula</th>
<th>Male</th>
<th>Female</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision (P)</td>
<td>Correct Values/Prediction Values * 100</td>
<td>88.69</td>
<td>88.00</td>
<td>88.36</td>
</tr>
<tr>
<td>Recall (R)</td>
<td>Correct Values/Actual Values * 100</td>
<td>81.87</td>
<td>81.48</td>
<td>81.69</td>
</tr>
<tr>
<td>F-measure</td>
<td>$2^\times (P \times R)/(P + R)$</td>
<td>85.14</td>
<td>84.62</td>
<td>84.89</td>
</tr>
<tr>
<td>Error Rate</td>
<td>(Actual Values - Correct Values)/Actual Values * 100</td>
<td>18.13</td>
<td>18.52</td>
<td>18.31</td>
</tr>
<tr>
<td>Accuracy Rate</td>
<td>Correct Values/Actual Values * 100</td>
<td>81.87</td>
<td>81.48</td>
<td>81.69</td>
</tr>
</tbody>
</table>

### Table 9.7 Algorithm Validation Metrics - Association Rules (Decision Tree)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Formula</th>
<th>Male</th>
<th>Female</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision (P)</td>
<td>Correct Values/Prediction Values * 100</td>
<td>90.64</td>
<td>88.46</td>
<td>89.60</td>
</tr>
<tr>
<td>Recall (R)</td>
<td>Correct Values/Actual Values * 100</td>
<td>85.16</td>
<td>85.19</td>
<td>85.17</td>
</tr>
<tr>
<td>F-measure</td>
<td>$2^\times (P \times R)/(P + R)$</td>
<td>87.82</td>
<td>86.79</td>
<td>87.33</td>
</tr>
<tr>
<td>Error Rate</td>
<td>(Actual Values - Correct Values)/Actual Values * 100</td>
<td>14.84</td>
<td>14.81</td>
<td>14.83</td>
</tr>
<tr>
<td>Accuracy Rate</td>
<td>Correct Values/Actual Values * 100</td>
<td>85.16</td>
<td>85.19</td>
<td>85.17</td>
</tr>
</tbody>
</table>

### Table 9.8 Algorithm Validation Parameters - Hybrid Algorithm

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Formula</th>
<th>Male</th>
<th>Female</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision (P)</td>
<td>Correct Values/Prediction Values * 100</td>
<td>96.05</td>
<td>95.54</td>
<td>95.81</td>
</tr>
<tr>
<td>Recall (R)</td>
<td>Correct Values/Actual Values * 100</td>
<td>93.41</td>
<td>92.59</td>
<td>93.02</td>
</tr>
<tr>
<td>F-measure</td>
<td>$2^\times (P \times R)/(P + R)$</td>
<td>94.71</td>
<td>94.04</td>
<td>94.40</td>
</tr>
<tr>
<td>Error Rate</td>
<td>(Actual Values - Correct Values)/Actual Values * 100</td>
<td>06.59</td>
<td>07.41</td>
<td>06.98</td>
</tr>
<tr>
<td>Accuracy Rate</td>
<td>Correct Values/Actual Values * 100</td>
<td>93.41</td>
<td>92.59</td>
<td>93.02</td>
</tr>
</tbody>
</table>

### Table 9.9 Comparison of Algorithms with No. of people Effected

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Actual</th>
<th>Classification</th>
<th>Clustering</th>
<th>Apriori</th>
<th>Decision Tree</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>344</td>
<td>302</td>
<td>309</td>
<td>318</td>
<td>327</td>
<td>334</td>
</tr>
</tbody>
</table>
Figure-9.1 Comparison of Algorithms - victims

Table-9.10 Comparison of Algorithms - Gender wise

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Gender</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M</td>
<td>F</td>
</tr>
<tr>
<td>Actual</td>
<td>182</td>
<td>162</td>
</tr>
<tr>
<td>Classification</td>
<td>158</td>
<td>144</td>
</tr>
<tr>
<td>Clustering</td>
<td>157</td>
<td>152</td>
</tr>
<tr>
<td>Apriori</td>
<td>168</td>
<td>150</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>171</td>
<td>156</td>
</tr>
<tr>
<td>Hybrid</td>
<td>177</td>
<td>157</td>
</tr>
</tbody>
</table>

Figure-9.2 Comparison of Algorithms - Gender wise

Table-9.11 Comparison of Algorithms Area wise

<table>
<thead>
<tr>
<th>Area</th>
<th>Actual</th>
<th>Classification</th>
<th>Clustering</th>
<th>Apriori</th>
<th>Decision Tree</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tribal</td>
<td>49.1</td>
<td>49.0</td>
<td>49.2</td>
<td>50.0</td>
<td>48.9</td>
<td>49.4</td>
</tr>
<tr>
<td>Hill</td>
<td>25.9</td>
<td>24.8</td>
<td>26.2</td>
<td>26.1</td>
<td>25.4</td>
<td>25.4</td>
</tr>
<tr>
<td>Rural</td>
<td>15.7</td>
<td>16.2</td>
<td>16.2</td>
<td>14.5</td>
<td>16.2</td>
<td>15.6</td>
</tr>
<tr>
<td>Urban</td>
<td>9.3</td>
<td>9.9</td>
<td>8.4</td>
<td>9.4</td>
<td>9.5</td>
<td>9.6</td>
</tr>
</tbody>
</table>

Figure-9.3 Comparison of Algorithms - Area wise
### Table-9.12 Comparison of Algorithms - Validation Parameters

<table>
<thead>
<tr>
<th></th>
<th>Classification</th>
<th>Clustering</th>
<th>Apriori</th>
<th>Decision Tree</th>
<th>Hybrid</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Precision</strong></td>
<td>84.75</td>
<td>85.12</td>
<td>83.87</td>
<td>84.38</td>
<td>94.78</td>
</tr>
<tr>
<td><strong>Recall</strong></td>
<td>70.42</td>
<td>72.54</td>
<td>73.24</td>
<td>76.06</td>
<td>89.44</td>
</tr>
<tr>
<td><strong>F-measure</strong></td>
<td>76.92</td>
<td>78.33</td>
<td>78.20</td>
<td>80.00</td>
<td>92.03</td>
</tr>
<tr>
<td><strong>Error Rate</strong></td>
<td>29.58</td>
<td>27.46</td>
<td>26.76</td>
<td>23.94</td>
<td>10.56</td>
</tr>
<tr>
<td><strong>Accuracy Rate</strong></td>
<td>70.42</td>
<td>72.54</td>
<td>73.24</td>
<td>76.06</td>
<td>89.44</td>
</tr>
</tbody>
</table>

![Figure-9.4 Comparison of Algorithms - Validation Parameters](image)

**Figure-9.4 Comparison of Algorithms - Validation Parameters**

![Figure-9.5 Comparison of Algorithms - Precision](image)

**Figure-9.5 Comparison of Algorithms - Precision**

![Figure-9.6 Comparison of Algorithms - Recall](image)

**Figure-9.6 Comparison of Algorithms - Recall**
Summary
Each algorithm has its individual strengths and limitations. The application of each algorithm may vary based on the dataset. By considering the strength and weakness of the existing algorithms, a new hybrid algorithm (EDPI) was proposed and developed in PHP language. The experimental results of Classification, Clustering, Apriori Algorithm, Decision Tree Models and Hybrid Algorithm are validated with the matrices (Precision, Recall, F-measure, Accuracy and Error rates). The newly developed Hybrid Algorithm proved to be accurate when compared with a standard statistical tool.