ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 9, September 2014

Design and Development of Data Mining System to Estimate Cars Promotion using Improved ID3 Algorithm

M.Jayakameswaraiah[1], S.Ramakrishna[2]

Research Scholar, Department of Computer Science, Sri Venkateswara University, Tirupati, India[1]

Professor, Department of Computer Science, Sri Venkateswara University, Tirupati, India[2]

Abstract: Now a day's customers are required comfort and their loving brand & color. With the arrival of the Internet and Data Mining Algorithms has definitely contributed to the altar of marketing focus. Here, we proposed Improved ID3 algorithm for best car market analysis. We are executed the similar in Weka Tool with Java code. We have analyzed the graphical performance study between ID3 and our novel improved ID3 clustering algorithms with Classes to Clusters evaluation purchase, safety, luggage booting, persons, doors, maintenance and buying attributes of customer's requirements for unacceptable/acceptable/good/very good ratings of a car to purchase. Conservative way of business is a challenging in car market due to many competitors are there around the world for providing aggressive products. The car manufacturers categorizes the car users and have to discover a suitable car; the seller correctly groups the buyers and he sells a right car; and the customers selects best car by analyzing more brands of cars with 'N' number of sellers. These three cases they spent too much of time for analyzing old or statistical records for choosing a right product.

Keywords: Supervised Learning, Weka, Classification, ID3, Improved ID3.

I. INTRODUCTION

This section will briefly sketch the underlying theoretical frameworks, after which we will present and discuss successfully apply error analysis, planning and development methods, all of which have been rolled out to production sites of two large automobile manufacturers. Section 2 provides some notations needed for the rest of the article. Section 3 discusses an ID3 Algorithm. Section 4 deals with the Improved ID3 Algorithm. Section 5 deals with comparison of ID3 and Improved ID3 algorithms. Time is very important in human life. We are able to recover the lost things, but never get even a single second. Effectively utilization of time majority of individuals are utilizing transport facility to their daily needs, which are help to save time in transportation. Firms today are worried with rising customer value through study of the customer life cycle. The tools and technologies of data mining, data warehousing and other customer association techniques afford new opportunities for businesses to act on the concepts of relationship marketing. The older model of "design-build-sell" (a product-oriented view) is being replaced by "sell-build-redesign" (a customer-oriented view). It is a spiral model of software engineering [27]. The conventional process of mass marketing is being challenged by the new approach of one-to-one marketing. In the conventional process, the marketing goal is to reach more customers and expand the customer base [32, 11, 5]. But known the high cost of get fresh customers; it makes better sense to perform business with current customers.

ID3 is uncomplicated decision tree learning algorithm which uses the greedy top to down search to build the tree which will choose the decision rules. For this there is a necessity for some mathematical concepts. The two concepts which are mostly involved in ID3 are Entropy and Information Gain. We will present examples of data mining applications from all these three stages, that are development, manufacture planning and error analysis. All assistance share the property that we use (or extract) rule patterns to explain the domain under analysis to the user. Rules (in form of association rules) are a well-understood means of representing knowledge and data dependencies. The natural interdisciplinary quality of the automobile development and manufacturing process requires models that are easily understood across application area boundaries. The understanding of patterns can be greatly enhanced by providing powerful visualization methods along with the analysis tools.

II. BACK GROUND

Selecting the suitable car is extremely tricky job if parameters (color, comfort, seating capacity, maintenance, price, and so on) are known otherwise it is difficult task. If the customer knows these all things then also sometimes it is hard to choose the right car.

The problem is unmanageable in the perspective of manufacturer and seller, because they must work with different categories of people [32]. Some people preferred only high expenditure cars, some are small price with all features and others are in between these classes. One more category of people are only knowing information about different brands but they never buys.

The need to increase the productivity of manufacturer, raises the seller transactions and customer satisfy of the selected car comforts. Comfort transportation is encourages frequency of vehicle usage, it increases Economic growth as well as it decreases wastage of time

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 9, September 2014

in journey. Car is the icon of comfortable.
If this system is available then there are no capabilities required to assist with telecom expense management, i.e., the administrator can find out the number of calls and text messages used as well as cellular and WiFi data usage, both for home and roaming networks [19].
We will now in brief discuss the notational foundation that is needed to present the ideas and results from the industrial applications.

A. Graphical Models

As we have pointed out in the beginning, here are dependencies and independencies that have to be taken into account when reasoning in complex domains shall be doing well. Graphical models are interesting since they provide a framework of modeling independencies between attributes and influence variables. The term "graphical model" is derived from an analogy between stochastic independence and node separation in graphs. Let V = {A1 ,... , An } be a set of unsystematic variables. If the fundamental probability distribution P (V) satisfies some criteria (see e. g. (CGH97; Pea93)), then it is possible to capture some of the independence relations between the variables in V using a graph G = (V, E), where E denotes the set of boundaries. The fundamental idea is to decompose the joint distribution P (V) into lower-dimensional marginal or conditional distributions from which the original distribution can be reconstructed with no or at least as few errors as possible (LS88; Pea88). The named independence relations allow for a simplification of these factor distributions. We claim, that every independence that can be read from a graph also holds in the corresponding joint allocation. The graph is then called a freedom map.

B. Supervised Learning

In the supervised learning problems, the machine is given a training set $Z = \{z_n\}_{n=1}^{N}$, which contains training examples $z_n = (x_n, y_n)$.We assume that each feature vector $x_n \in X \subseteq \mathbb{R}^D$, each label $y_n \in y$, and each training example z_n is drawn independently from an unknown probability measure dF(x, y) on X × Y. We focus on the case where dF(y | x), the random process that generates y from x, is governed by

$$y = g_*(X) + \epsilon_x$$

Here g_*: $x \longrightarrow y$ is a deterministic but unknown component called the target function, which denotes the best function that can predict y from x. The exact notion of "best" varies by application needs and will be formally defined later in this section. The other part of y, which cannot be perfectly explained by $g_*(x)$, is represented by a random component ϵ_x.
With the given training set, the machine should return a decision function $\hat{g}$as the inference of the objective function. The result function is chosen from a learning model G = {g}, which is a collection of candidate functions$g : X \rightarrow Y$. Briefly speaking, the task of supervised learning is to use the information in the training set Z to find some decision function $\hat{g} \in \mathcal{G}$ that is almost as good as g_* under dF (x, y).
For instance, we may want to build a recognition system that transforms an image of a written digit to its proposed meaning. We can initially ask somebody to write down N digits and represent their images by the feature vectors xn. We then label the images by $y_n \in \{0, 1, \ldots, 9\}$ according to their meanings. The target function g_x here encodes the process of our human-based recognition system and ϵ_x represents the mistakes we may formulate in our brain. The assignment of this learning problem is to set up an automatic recognition system (decision function) $\hat{g}$ that is almost as good as our own recognition system, even on the yet unseen images of written digits in the future.
The machine conquers the task with a learning algorithm A. Generally speaking, the algorithm takes the learning model G and the training set Z as inputs. It then returns a decision function g∈G by minimizing a predefined objective function E(g, Z) over g ∈ G.
Let us take one step back and look at what we mean by g_* being the "best" function to forecast y from x. To estimate the predicting ability of any $: X \rightarrow Y$, we define its out-of-sample cost

$$\pi(g, F) = \int C\big(y, g(x)\big) dF(x, y)$$

Here C (y, k) is called the expenditure function, which quantify the price to be paid when an example of label y is predicted as k. The value of π (g, F) reflects the estimated test cost on the (mostly) unseen examples tired from dF(x, y).
In this thesis, we assume that such a g_*exists with ties in argmin arbitrarily broken, and denote $\pi(g, F)$ by $\pi(g)$ when F is clear from the context.
Recall that the task of supervised learning is to find some $\hat{g}$ G that is almost as good as g_* under dF(x, y). Since $\pi(g*)$ is the lower bound, we desire $\pi(\hat{g})$ to be as small as possible. Note that A minimizes E (g, Z) to get ĝ, and hence ideally we want to set E(g, Z) = π(g). Nevertheless, because dF(x, y) is unknown, it is not possible to compute such an E(g, Z) nor to minimize it directly. A substitute quantity that depends only on Z is called the in-sample cost

$$\nu(g) = \sum_{i=1}^{N} C\big(y_n, g(x_n)\big).\frac{1}{N}.$$

Note that ν(g) can also be defined by π(g, Zu) where Zu denotes a identical distribution over the training set Z . Because ν(g) is an unbiased estimate of π(g) for any given single g, many learning algorithms take ν(g) as a major component of E(g, Z). A small ν(g), however, does not always imply a small π(g).
One important type of supervised learning problem is regression, which deals with the case when Y is a metric space isometric to R. For simplicity, we shall restrict ourselves to the case where Y = R. Although not strictly required, common regression algorithms usually not only

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 9, September 2014

work on some G that contains continuous functions, but also desires gˆ to be reasonably smooth as a control of its complexity. The metric information is thus important in determining the smoothness of the function.

Another important type of supervised learning problem is called classification, in which Y is a finite set Yc = {1, 2, . . . , K }. Each label in Yc represents a different category. For instance, the digit recognition system described earlier can be formulated as a classification problem. A function of the form X → Yc is called a classifier. In the special case where |Yc | = 2, the classification problem is called dual classification, in which the classifier g is called a binary classifier.

C. Classification

Classification problems try to determine the characteristics which correctly identify the class to which each instance belongs to. Thus, the scope is to learn a model from the training set which describes the class Y, i.e. predict y from the values of (a subset of) (x(1), ..., x(n)). The resulting model can be employed either for descriptive, or predictive tasks. Classification is similar to clustering, the major variation being that, in classification, the class to which each instance in the dataset belongs to is known a priori.

The most intense research efforts in DM and connected fields (machine learning, statistics) have focused on finding efficient classification algorithms, such that at the present there is a large collection of state-of-the-art methods available in the literature. This thesis focuses on DM classification tasks.

D. C4.5 Algorithm

Systems that construct classifiers are one of the commonly used tools in data mining. Such system obtain as input a set of cases, each belonging to one of a small number of classes and described by its values for a fixed set of attribute, and output a classifier that can exactly predict the class to which a new case belongs. Related to CLS and ID3, C4.5 generates classifiers expressed as result trees, but it can also construct classifiers in more understandable rule set form [2, 3, 4].

a) Decision Tree

Decision trees are trees that classify instances by sorting them based on feature values given a set S of cases, C4.5 first grow an original tree using the divide-and-conquer algorithm as follows:

- If all the cases in S belong to the identical class or S is small, the tree is a leaf labeled with the most frequent class in S.
- Otherwise, choose a test based on a single attribute with two or more outcome. Construct this test the root of the tree with one branch for each result of the test, partition S into resultant subsets S1, S2… according to the outcome for each case, and apply the same process to each subset.

Decision trees are usually unvaried since they use based on a single feature at each internal node. Most decision tree algorithms cannot accomplish well with problems that require diagonal partition.

C4.5 uses two heuristic criteria to rank possible tests: information gain, which minimize the entire entropy of the subsets {Si} and the default gain ratio that divides information gain by the information provided by the test outcome. Attributes can be whichever numeric or nominal and this determines the format of the test outcomes. For a numeric attribute A they are $\{A \leq h, A > h\}$ where the threshold h is found by sorting S on the values of A and choosing the split between successive values that maximizes the principle above. An attribute A with isolated values has by default one outcome for every value, but an alternative allows the values to be grouped into two or more subsets with one outcome for every subset. The primary tree is then pruned to avoid over fitting. The pruning algorithm is based on a pessimistic estimate of the error rate correlated with a set of N cases, E of which do not belong to the most recurrent class. In its place of E/N, C4.5 determines the upper limit of the binomial probability when E events have been practical in N trials, using a user-specified assurance whose default value is 0.25. Pruning is accepted from the leaves to the root. The probable error at a leaf with N cases and E errors is N times the pessimistic error rate as beyond. For a sub tree, C4.5 adds the estimated errors of the branches and compares this to the estimated error if the sub tree is replaced by a leaf; if the latter is no higher than the former, the sub tree is pruned. Similarly, C4.5 checks the estimated error if the sub tree is replaced by one of its branches and when this appears beneficial the tree is modified accordingly.

The pruning process is completed in one pass all the way throughout the tree C4.5's tree-construction algorithm differs in other than a few respects from CART [5].

- Tests in CART are forever binary, but C4.5 allows two or extra outcomes.
- CART uses the Gini diversity index to rank tests, while C4.5 uses information-based criterion.
- CART prunes trees using a cost-complexity model whose parameters are estimated by cross-validation; C4.5 uses a single-pass algorithm resulting from binomial assurance limits.
- This brief discussion has not mentioned what happens when some of a case's values are unidentified. CART looks for substitute tests that estimated the outcomes when the tested attribute has an unidentified value, but C4.5 apportions the case probabilistically among the outcomes.

b) *Limitations of C4.5 Algorithm:* The restrictions of C4.5 are discussed

(a) Empty branches: Constructing tree with meaningful value is one of the crucial steps for rule invention by C4.5 algorithm. In our trial, we have originated several nodes with zero values or close to zero values. These values neither contribute to make rules nor help to make any class for classification task. Slightly it makes the tree

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 9, September 2014

larger and more complex.

(b) Insignificant branches: Numbers of selected discrete attributes create equal number of potential branches to build a decision tree. But every single one of them are not significant for classification task. These irrelevant branches not only decrease the usability of decision

(c) Over fitting: Over fitting happens when algorithm model picks up data with infrequent individuality. This cause many fragmentations is the process allocation. Statistically irrelevant nodes with very few samples are known as fragmentations. Generally C4.5 algorithm constructs trees and grows it branches 'immediately deep adequate to perfectly categorize the exercise examples'. This approach performs well through noise free data. But the largest part of the time this approach over fits the training examples with noisy data. Presently there are two approaches are generally using to bypass this over-fitting in decision tree learning. Those are:

- If tree grows extremely large, stop it before it reaches maximal point of perfect classification of the training data.
- Permit the tree to over-fit the training data then post-prune tree.

E. About Weka tool

There are many tools available for data mining and machine learning, but in this research work we use the open source software suite WEKA which stands for Waikato Environment for Knowledge Analysis. The main reason why we selected to use WEKA was because of its flexibility. WEKA is a well-liked tool used for data analysis, machine learning and predictive modeling that was developed by the University of Waikato in New Zealand using the programming language JAVA.

1. Main Features

Some of WEKAs most important features are the following:

Data preprocessing - WEKA supports a couple of popular text file formats such as CSV, JSON and Matlab ASCII files to import data along with their own file format ARFF. They also have support to import data from databases with JDBC. Besides importing data, they have a wide collection of supervised as well as unsupervised filters to apply on your data to facilitate further analysis.

Data classification - A huge collection of algorithms have been implemented to perform classification scheduled data sets. These comprise Bayesian algorithms, mathematical functions such as support vector machines, lazy classifiers implementing nearest-neighbor calculations; Meta based algorithms as well as rule and tree-based classifiers.

Data clustering - A couple of algorithms for clustering exist such as variations of the k-mean method as well as density and hierarchical based clustering algorithms.

Attribute association - Methods to analyze data using association rule learners. Association rules can be seen as rules describing relations between attributes in a data set.

Attribute selection - Methods to evaluate which attribute contribute the most when predicting an outcome.

Data visualization - Depending on the methods used to analyze the data, this view can to plot data against suitable variables as well as give tools to analyze specific points further.

The WEKA file format

The main file format used in WEKA is .arff format called Attribute Relationship file format. It is basically a normal text file with the following structure:

```
@relation car

@attribute buying {vhigh, high, med, low}
@attribute maint {vhigh, high, med, low}
@attribute doors {2, 3, 4, 5more}
@attribute persons {2, 4, more}
@attribute lug_boot {small, med, big}
@attribute safety {low, med, high}
@attribute purchase {unacc, acc, good, vgood}
```

The last attribute is the "label" class by default which is used when training the data to know if it was classified correctly

% @DATA means the start of the actual data. Every row is single entry, and the values are comma separated. The principles are entered in the arrangement that the attributes are declared above. Also note that string and date values have to be quoted since they can include whitespaces [49,54].

```
@data

vhigh,vhigh,2,2,small,low,unacc
vhigh,vhigh,2,2,small,med,unacc
vhigh,vhigh,2,2,small,high,unacc
vhigh,vhigh,2,2,med,low,unacc
vhigh,vhigh,2,2,med,med,unacc
```

II. ID3 ALGORITHM

The ID3 algorithm was originally developed by J. Ross Quinlan at the University of Sydney, and he first presented it in the 1975 book "Machine Learning". The ID3 algorithm induces classification models, or decision trees, from data. It is a supervised learning algorithm that is trained by examples for different classes. After being trained, the algorithm should be able to predict the class of a new item.

ID3 identifies attributes that differentiate one class from another. All attributes must be known in advance, and must also be either continuous or selected from a set of known values. For instance, temperature (continuous), and country of citizenship (set of known values) are valid attributes. To determine which attributes are the most important, ID3 uses the statistical property of entropy. Entropy measures the amount of information in an attribute. This is how the decision tree, which will be used in testing upcoming cases, is built.

The principle of the ID3 algorithm is follows. The hierarchy is constructed top-down in a recursive approach. At the root, every attribute is tested to find out

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 9, September 2014

how well it alone classifies the transactions. The "best" attribute (to be discussed below) is then chosen and the remaining transactions are partitioned by it.

A. Entropy

In information theory, entropy is a measure of the uncertainty about a source of messages. The other uncertain a recipient is about a source of messages, the additional information that recipient will need in order to know what message has been sent.

First, let's assume, without loss of generality, that the resulting decision tree classifies instances into two categories, we'll call them P(positive)and N(negative).

Given a set S, containing these positive and negative targets, the entropy of S connected to this Boolean classification is:

Entropy(S) = - P (positive) log2P (positive) -
P (negative) log2P (negative)

P (positive): proportion of positive examples in S

P (negative): proportion of negative examples in S

For example, if S is (0.5+, 0.5-) then Entropy(S) is 1, if S is (0.67+, 0.33-) then Entropy(S) is 0.92, if P is (1+, 0 -) then Entropy(S) is 0. Note that the additional uniform is the probability distribution; the bigger is its information.

B. Information gain

Now consider what happens if we partition the set on the basis of an input attribute X into subsets$T_1, T_2, T_3, \ldots, T_N$. The information needed to identify the class of an element of T is the weighted average of the information needed to identify the class of an element of each subset:

$$H(X,T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} H(T_i)$$

In the context of building a decision tree, we are paying attention in how greatly information about the output attribute can be gained by knowing the value of an input attribute . This is just the difference between the information needed to classify an element of before knowing the value of X, H(T), and the information needed after partitioning the dataset T on the basis of knowing the value of X, H(X, T). We define the information gain due to attribute X for set T as:

$$\text{Gain}(X,T) = H(T) - H(X,T)$$

In order to choose which attribute to divide upon, the ID3 algorithm computes the information gain for each attribute, and selects the one with the highest gain.

The simple ID3 algorithm above can have difficulties when an input attribute has many possible values, because Gain(X, T) tends to favor attributes which have a large number of values. It is easy to understand why if we consider an extreme case.

Imagine that our dataset contains an attribute that has a different value for every element of T. This could arise in practice if a unique record ID was retain while extract, from a database.

The problem also arises when an attribute can take on many values, even if they are not unique to each element. Quinlan (1986) suggests a solution based on considering the amount of information required to determine the value of an attribute X for a set T. This is given by H(PX,T), where PX,T is the probability distribution of the values of X:

$$P_{X,T} = \left(\frac{|T_1|}{|T|}, \frac{|T_2|}{|T|}, \ldots, \frac{|T_n|}{|T|}\right)$$

The quantity H(PX,T) is known as the divide information for attribute X and set T.

III. IMPROVED ID3 ALGORITHM

This Algorithm is characterized by the ability to deal with the explosion of business data and accelerated promote changes, these uniqueness help providing powerful tools for judgment makers, such tools can be used by industry users (not only statisticians) for analyzing huge amount of data for patterns and trends. Consequently, data mining has become a research area with increasing importance and it involved in determining useful patterns from collected data or determining a model that fits best on the collected data[37,40].

It is used to investigate the attributes of car in the perspective of manufacturer – whether a new product is able to launch or not, seller – whether a customer may purchase a car or not and customer – whether a manufacturer provided suitable and comfortable car or not and the seller is able to guide for best car i.e., suitable to me. It is essential to analyze the car in short span of time, consider cases when all parties (i.e. manufacturer, seller and customer) selecting a right product.

Algorithm for Improved ID3
Input: Training data set
Output: Decision Making approach

Algorithm

ImprovedID3 (Learning Sets S, Attributes values, V Attribute Sets A)
Begin

1. Load learning sets first, generate root node 'rootNode add learning set S into root node as its subset.
2. Load training data set for training.
3. If attributes are exclusively identified in data set, rem it from training set.
4. With the source of distance metric split the specified training data into subsets.

3.1. Calculate the distance for n objects, each instance in available dataset.

$$D(x,y) = \sum_{i=1}^{n} |X_i - Y_i|$$

Where X is selected instance and Y is comparing instance.

5. if D>55% then instance is belong to same group and into new set and remove from original data set. Otherwise do nothing.
6. Repeat the steps 3.1 and 4 for each instance until all matched it not found.
7. On each split apply ID3 algorithm recursively.

- If the entire examples are positive, return the single-node tree root with mark is positive.
- If the entire examples are negative, return the single-

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 9, September 2014

node tree root with mark is negative.

- If number predicting attributes are unfilled, then return the single node tree root, with the mark is most familiar value of the target attribute in the examples.
- Otherwise

Begin

For rootNode, we compute Entropy(rootNode.subset) first

$$\text{Entropy}(S) = \sum_{i=1}^{c} P_i \log_2 P_i$$

- If Entropy(rootNode.subset)==0, then the subset consists of proceedings all with the same value for the unqualified attribute, return a leaf node with decision attribute:value;
- If Entropy(rootNode.subset)!=0, then calculate information gain for every attribute left(have not been used in splitting), discover attribute A with Maximum(Gain(S,A)).
- generate child nodes of this rootNode and add to rootNode in the decision hierarchy.
- For every child of the rootNode, apply ID3(S,V,A) recursively until reach node that has entropy=0 or reach leaf node.

End

End

V. RESULT ANALYSIS

A. Compare and contrast Test Results with Graphical performance Analysis of ID3 and Improved ID3 Algorithms:

We are considered car data set from UCI repository to compare ID3 and Improved ID3 Algorithms.

In this section we are compared ID3 and Improved ID3 algorithms. The following parameters are used to compare and contrast these two algorithms.

- Correctly Classified Instances (%)
- Incorrectly Classified Instances (%)
- Kappa statistic
- Mean absolute error
- Root mean squared error
- Relative absolute error (%)
- Root relative squared error (%)
- Coverage of cases (0.95 level) (%)
- Mean rel. region size (0.95 level) (%)
- Total Number of Instances

ID3 Algorithm								
S.No	Parameter	Percentage split 33%	Percentage split 66%	Percentage split 99%	20 Cross Fold	10 Cross Fold	5 Cross Fold	Training Set
1	Correctly Classified Instances (%)	84.2832	89.1156	88.2353	89.1782	89.3519	88.8889	100
2	Incorrectly Classified Instances (%)	11.5717	5.7823	5.8824	2.7199	3.5301	4.8032	0
3	Kappa statistic	0.7289	0.8549	0.8182	0.9263	0.9071	0.8768	1
4	Mean absolute error	0.0604	0.0305	0.0313	0.0148	0.019	0.0256	0
5	Root mean squared error	0.2457	0.1745	0.1768	0.1216	0.1379	0.1601	0
6	Relative absolute error (%)	28.1063	14.6401	16.9933	7.5435	9.4937	12.5661	0
7	Root relative squared error (%)	76.3184	55.3612	64.2209	40.3727	45.0502	51.5442	0
8	Coverage of cases (0.95 level) (%)	84.2832	89.1156	88.2353	89.1782	89.3519	88.8889	100
9	Mean rel. region size (0.95 level) (%)	23.9637	23.7245	23.5294	22.9745	23.2205	23.423	25
10	Total Number of Instances	1158	588	17	1728	1728	1728	1728

Table 1: Execution of ID3 Algorithm in various parameters

Improved ID3 Algorithm								
S.No	Parameter	Percentage split 33%	Percentage split 66%	Percentage split 99%	20 Cross Fold	10 Cross Fold	5 Cross Fold	Training Set
1	Correctly Classified Instances (%)	86.7876	91.8367	94.1176	93.0556	93.4606	92.0139	100
2	Incorrectly Classified Instances (%)	13.2124	8.1633	5.8824	6.9444	6.5394	7.9861	0
3	Kappa statistic	0.705	0.8218	0.8496	0.8491	0.8579	0.8245	1
4	Mean absolute error	0.1057	0.0797	0.0566	0.0762	0.0741	0.0767	0.0173
5	Root mean squared error	0.2167	0.1834	0.182	0.171	0.1675	0.1751	0.0512
6	Relative absolute error (%)	45.9512	34.7459	27.3814	33.2642	32.3612	33.4924	7.5667
7	Root relative squared error (%)	64.1666	54.106	60.0271	50.5601	49.5369	51.7995	15.1418
8	Coverage of cases (0.95 level) (%)	99.0501	99.4898	100	99.8843	99.9421	99.9421	100
9	Mean rel. region size (0.95 level) (%)	46.3731	40.5612	36.7647	41.5365	41.4641	40.7552	31.4236
10	Total Number of Instances	1158	588	17	1728	1728	1728	1728

Table 2: Execution of Improved ID3 Algorithm in various parameters

1. Correctly Classified Instances

Correctly Classified Instances tells you that your guess was correct. The labels on the test set are supposed to be the actual accurate classification. The performance is computed by asking the classifier to give its best guess about the classification for each instance in the test set. After that the predicted classifications are compared to the genuine classifications to verify accuracy. So I have a training set with tweets that I gave the label for and a test set with tweets that all have the label "positive". When I ran Naive Bayes, I get the following results:

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 9, September 2014

Correctly classified instances: 69 92% Incorrectly classified instances: 6 8%

Then if I change the labels of the tweets in the test set to "negative" and ran once more Naive Bayes, the results are inversed:

In Correctly classified instances: 6 8% incorrectly classified instances: 69 92%

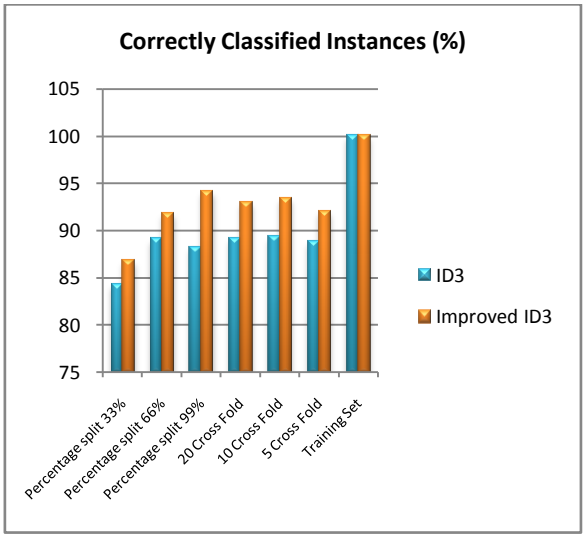

Figure 1: Comparison of ID3 and Improved ID3 with Correctly Classified Instances (%)

2) *Incorrectly Classified Instances*

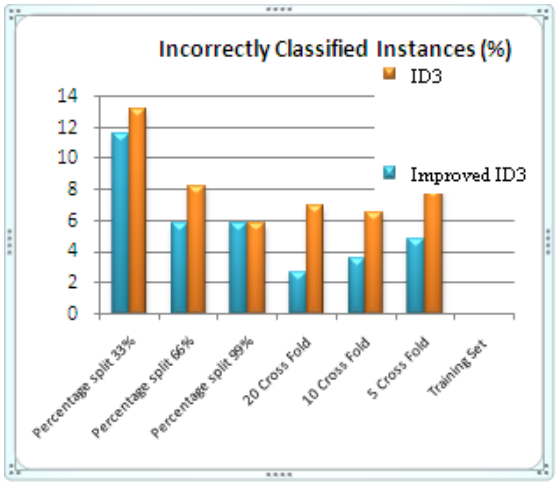

Figure 2: Comparison of ID3 and Improved ID3 Algorithm with Incorrectly Classified Instances (%)

3) *Kappa statistic*

Kappa is for calculation of agreement normalized for possibility agreement.

$$K = \frac{P(A) - P(E)}{1 - P(E)}$$

Where P(A) is the percentage agreement (e.g., between your classifier and ground truth) and P(E) is the possibility agreement. K=1 indicates faultless agreement, K=0 indicates chance agreement.

Kappa is a chance-corrected measure of agreement between the classifications and the accurate classes. It's calculated by taking the contract expected by chance away from the observed agreement and dividing by the maximum feasible agreement. A value bigger than 0 means that your classifier is doing improved than chance.

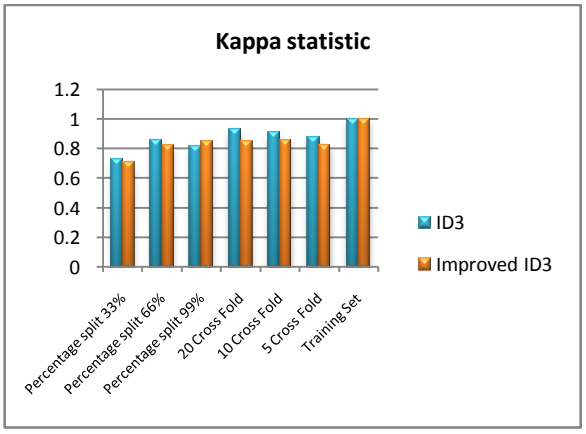

Figure 3: Comparison of ID3 and Improved ID3 Algorithm with Kappa *statistic*

4) *Mean absolute error(MAE)*

The MAE measures the average magnitude of the errors in a set of forecasts, without allowing for their direction. It measures accuracy for nonstop variables. The equation is given in the documents references. Expressed in words, the MAE is the average over the verification sample of the absolute values of the differences between forecast and the equivalent observation. The MAE is a linear achieves which means that all the individual differences are weighted equally in the average.

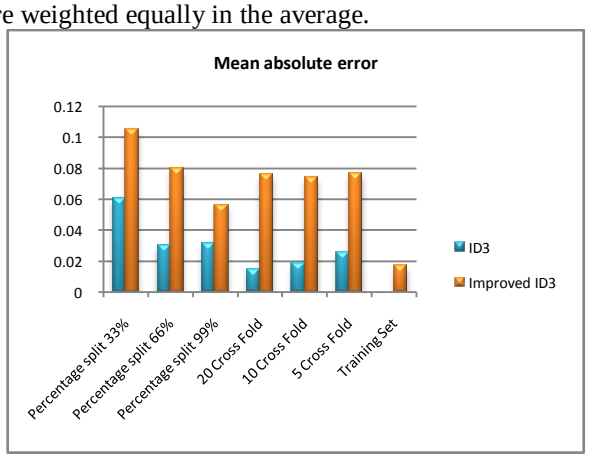

Figure 4: Comparison of ID3 and Improved ID3 Algorithm with Mean absolute error

5) *Root mean squared error (RMSE)*

The RMSE is a quadratic scoring rule which measures the average magnitude of the inaccuracy. The equation for the RMSE is given in commonly of the references. Expressing the principle in words, the difference between forecast and similar observed values are each squared and then averaged over the example. At last, the square root of the standard is taken. Since the errors are squared before they are averaged, the RMSE gives a comparatively high weight to outsized errors. This means the RMSE is mainly useful when large errors are particularly undesirable.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 9, September 2014

The MAE and the RMSE can be used together to diagnose the variation in the errors in a set of forecasts. The RMSE will constantly be larger or equal to the MAE; the greater difference between them, the bigger the variance in the individual errors in the example. If the RMSE=MAE, then the entire errors are of the same magnitude

Both the MAE and RMSE can range from 0 to ∞. They are negatively-oriented scores: Lower values are better.

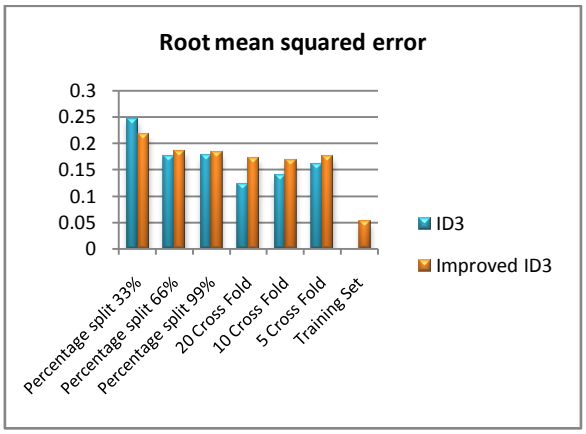

Figure 5: Comparison of ID3 and Improved ID3 Algorithm with Root mean squared error

6) Relative absolute error (RAE)

The formula for Root Relative Squared Error is actually the formula for the Relative Squared Error. You have to take the square root of this formula to get what Weka outputs.

$$\frac{|p_1 - a_1| + \ldots + |p_n - a_n|}{|\bar{a} - a_1| + \ldots + |\bar{a} - a_n|}$$

With Actual target values: a1 a2 … an, Predicted target values: p1 p2 … pn

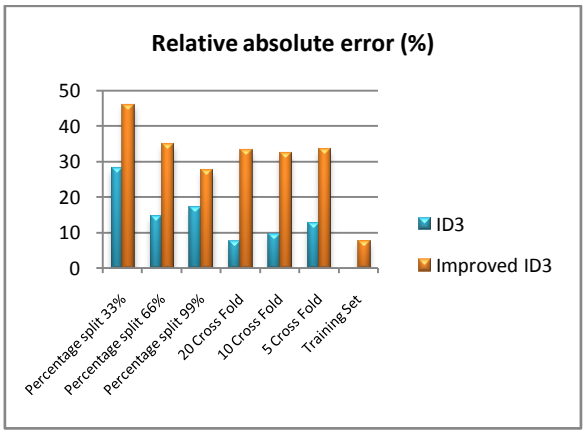

Figure 6: Comparison of ID3 and Improved ID3 Algorithm with Relative absolute error (%)

7) Root relative squared error (RRSE)

RRSE is computed by dividing the RMSE by the RMSE obtained by just predicting the mean of target values (and then multiply by 100). As a result, smaller values are better and values > 100% indicate a scheme is doing worse than just predicting the mean.

$$\frac{(p_1 - a_1)^2 + \ldots + (|p_n - a_n|^2)}{(\bar{a} - a_1)^2 + \ldots + (|p_n - a_n|^2)}$$

With Actual target values: a1 a2 … an, Predicted target values: p1 p2 … pn

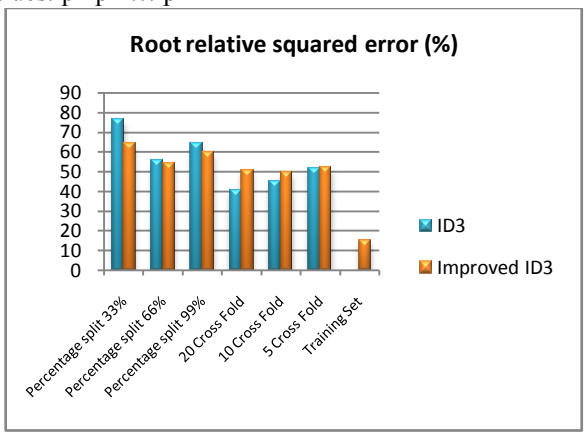

Figure 7: Comparison of ID3 and Improved ID3 Algorithm with Root relative squared error (%)

8) Coverage of cases (0.95 level)

Interval estimation statistics, namely coverage and average relative width of intervals at a 95% assurance level, are now output for any regression plan that implements Interval Estimator. The relative width is the interval width normalized by the range of target values in the training data (i.e. relative width >= 100% corresponds to inadequate intervals). The experiential coverage for an interval estimator should be >= the confidence level (i.e. 95%). A reasonable interval estimator is one that exhibits coverage at or above the 95% level while producing narrow intervals.

These two statistics are *also* output for any nominal classification scheme, based on predicted probabilities. An "interval" in this case is the smallest set of class values such that the cumulative class probability for these values exceeds the 0.95 level. The relative width is the number of class values in the set divided by the total number of class values.

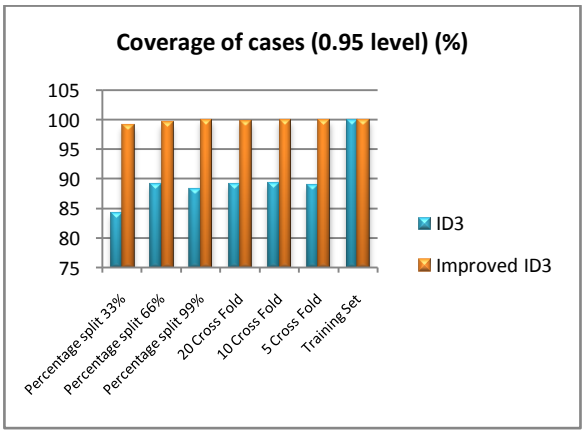

Figure 8: Comparison of ID3 and Improved ID3 Algorithm with Coverage of cases (0.95 level) (%)

9) Mean rel. region size (0.95 level)

This is part of the evaluation of computed confidence bounds on the predictions made by the classifier. The Javadocs for the method that is used to produce the

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 9, September 2014

output.Gets the average size of the predicted regions, virtual to the range of the target in the training data, by the confidence level precise when evaluation was performed. This we can fetch through sizeOfPredictedRegions() method.

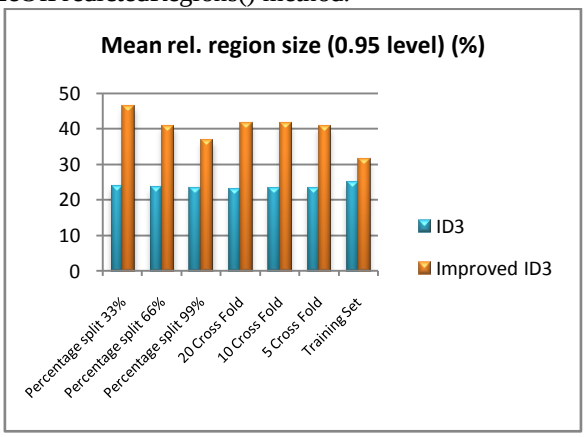

Figure 9: Comparison of ID3 and Improved ID3 Algorithm with Mean rel. region size (0.95 level) (%)

10) Total Number of Instances

The total number of instances that are used in classifier output.

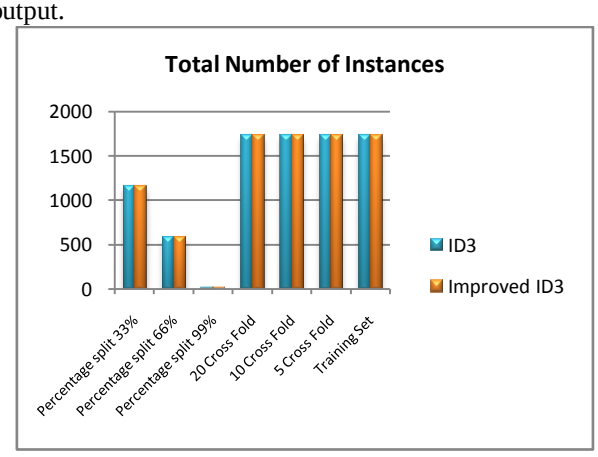

Figure 10: Comparison of ID3 and Improved ID3 Algorithm with Total Number of Instances

VI. CONCLUSION

In this section we presented ID3 and our improved ID3 classification Algorithms. We have also executed the same in Weka Tool with Java code and compared the performance of two algorithms based on Percentage Split and Cross Validation for Correctly Classified Instances (%), Incorrectly Classified Instances (%), Kappa statistic, Mean absolute error, Root mean squared error, Relative absolute error (%), Root relative squared error (%), Coverage of cases (0.95 level) (%), Mean rel. region size (0.95 level) (%) .The Improved ID3 algorithm gives accurate results compared to ID3. The Total Number of Instances result gives promotion and help to the car seller/manufacturer for analyzing their customers. We analyzed the graphical performance investigation between ID3 and our improved ID3 classification Algorithms with Visualize Classifier error, Visualize margin curve, Visualize Threshold Curve, Cost/ Benefit Analysis and Visualize Cost Curve on the cars promotion with color, purchase, safety, luggage boot, persons (seating capacity), doors, maintenance and buying attributes of customers requirements for unacceptable/acceptable/good/very good ratings of a car to purchase.

In the future work, if we will integrate and apply the Improved ID3 algorithm with clustering technique to the real world data sets in the UCI Machine learning repository for better accuracy and performance.

REFERENCES

[1] A. Feelders, H. Daniels, M. Holsheimer, "Methodological and Practical Aspects of Data Mining", Information & Management, 271-281, 2000.
[2] Aggarval, C. C., & Yu, P. S,"Finding localized associations in market basket data", IEEE Transactions on Knowledge and Data Engineering, 14, 51–62, 2002.
[3] Alex Berson, Kurt Thearling, Stephen J.Smith, "Building Data Mining Applications for CRM" , kindle edition,eBook, 488 pages,2002.
[4] B. Sun and Morwitz, V.G,"Stated intentions and purchase behavior: A unified mode", International Journal of Research in Marketing,Volume 27(4), 356-366,2010.
[5] B. V. Dasarathy., "Nearest neighbor (nn) norms: Nn pattern classi_cation tech-niques", IEEE Computer Society Press, 1991.
[6] C-L. Huang, M-C. Chen and, C-J. Wang, "Credit scoring with a data mining approach based on support vector machines", Expert System with Applications, 37, 847-856, 2007.
[7] Cabibbo and R. Torlone, "An architecture for data warehousing supporting data independence and interoperability: an architecture for data warehousing", International Journal of Cooperative Information Systems, vol. 10, no. 3, 2001.
[8] Calvanese, G. D. Giacomo, M. Lenzerini, D. Nardi, and R. Rosati, "Data integration in data warehousing", International Journal Of Cooperative Information Systems, vol. 10, no. 3, pp. 237, 2001.
[9] Christoph F. Eick, Nidal Zeidat, and Zhenghong Zhao, "Supervised Clustering – Algorithms and Benefits", 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004), 2004.
[10] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze, "Introduction to information retrieval", Cambridge University Press, 2009.
[11] D. Olson and S. Yong, Introduction to Business Data Mining. McGraw Hill International Edition, 2006.
[12] Data mining for Business Intelligence. [www.dataminingbook.com]
[13] David J. Hand, Heikki Mannila and Padhraic Smyth, "Principles of Data Mining", MIT Press,Fall 2000.
[14] Douglas, S.Agarwal, D., & Alonso, T,"Mining customer care dialogs for ''daily news". IEEE Transactions on Speech and Audio Processing, 13, 652–660, 2005.
[15] Dunham, M.H.," Data Mining: Introductory and Advanced Topics", Pearson Education Inc.2003.
[16] Fayyad U, "From Data Mining to Knowledge Discovery: An overview", In Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press.1996.
[17] Fong, Q.Li, and S. Huang, "Universal data warehousing based on a meta-data modeling approach", International Journal Of Cooperative Information Systems, vol. 12, no. 3, pp. 325, 2003.
[18] Ha, S.H,"Helping online customers decide through web personalization". IEEE Intelligent Systems, 17, 34–43.2002.
[19] IDC & Cap Gemini. "Four elements of customer relationship management". Cap Gemini White Paper.ISBN: 978-0-387-79419-8, 2009.
[20] James A.O. Brien, "Management Information System", Tata Mc Graw Hill publication company Limited, New Delhi, 2009.
[21] Jiang, T., & Tuzhilin, A," Segmenting customers from population to individuals: Does 1-to-1 keep your customers forever", IEEE Transactions on Knowledge and Data Engineering, 18, 1297–1311, 2006.

[22] K. Hian and K.L. Chan, "Going concern prediction using data mining techniques", Managerial Auditing Journal, Vol 19, No 3, 462-476, 2004.
[23] Kalton, K. Wagstaff, and J. Yoo, "Generalized Clustering,Supervised Learning, and Data Assignment," Proceedings of the Seventh International Conference on Knowledge Discovery and DataMining, ACM Press, 2001.
[24] Kantardzic, M.," Data Mining: Concepts, Models, Methods and Algorithms", Wiley-IEEE Press, 2011.
[25] Kerdprasop, and K. Kerdpraso, "Moving data mining tools toward a business intelligence system", Enformatika, vol. 19, pp. 117-122, 2007.
[26] Kilian Q.Weinberger, Lawrence K. Saul, "Distance Metric Learning for Large Margin Nearest Neighbor Classification", Journal of Machine Learning Research, 207-244, 2009.
[27] Kubat, M., Hafez, A., Raghavan, V. V., Lekkala, J. R., & Chen, W. K," Item set trees for targeted association querying". IEEE Transaction on Knowledge and Data Engineering, 15, 1522–1534, 2003.
[28] Lapersonne, G. Laurent and J-J Le Goff, "Consideration sets of size one: An empirical investigation of automobile purchases", International Journal of Research in Marketing 12, 55-66, 1995.
[29] Leontin, T. L., Moldovan, D., Rusu, M., Secară, D., Trifu, C,"Data mining on the real estate market", Revista Informatica Economică, nr. 4 (36), 2005.
[30] Liqiang and J. Howard, "Interestingness measures for data mining: a survey", ACM Computing Surveys, vol. 38, no. 3, pp. 1-32, 2006.
[31] M. Panda and M. Patra,"A novel classification via clustering method for anomaly based network intrusion detection system", International Journal of Recent Trends in Engineering, 2:1–6, 2009.
[32] M.R.Lad, R.G.Mehta, D.P.Rana, "A Noval Tree Based Classification", [IJESAT] International Journal of Engineering and Advanced Technology Volume-2, Issue-3, 581 – 586 may 2012.
[33] Moutinho, L., Davies, F. and Curry, B,"The impact of gender on car buyer satisfaction and loyalty". Journal of Retailing and Consumer Services 3(3), 135-144, 1996.
[34] M. Matteucci,"A Tutorial on Clustering Algorithms", 2008. [http://home.dei.polimi.it/matteucc/Clustering/tutorial_html].
[35] Ming, H., Wenying, N. and Xu, L, "An improved decision tree classification algorithm based on ID3 and the application in score analysis", Chinese Control and Decision Conference (CCDC), pp1876-1879, 2009.
[36] Mitra, S., Pal, S. K., & Mitra, P,"Data mining in soft computing framework: A survey". IEEE Transactions on Neural Networks, 13, 3–14, 2002.
[37] R. Krakovsky and R. Forgac,"Neural network approach to multidimensional data classification via clustering", Intelligent Systems and Informatics (SISY), 2011 IEEE 9th International Symposium on, 169–174, IEEE2011.
[38] Rahul A. Patil, Prashant G. Ahire, Pramod. D. Patil, Avinash L. Golande ,"A Modified Approach to Construct Decision Tree in Data Mining Classification" , International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 1, July 2012.
[39] Rosset, S., Neumann, E., Eick, U., & Vatnik, N,"Customer lifetime value models for decision support", Data Mining and Knowledge Discovery, 7, 321–339, 2003.
[40] R. Nayak and T. Qiu, "A data mining application: analysis of problems occurring during a software project development process", International Journal Of Software Engineering & Knowledge Engineering, vol.15, no. 4, pp. 647-663, 2005.
[41] S. Bongsik, "An exploratory investigation of system success factors in data warehousing", Journal of the Association for Information Systems, vol. 4, pp. 141-168, 2003.
[42] S. Lee, S. Hong, and P. Katerattanakul, "Impact of data warehousing on organizational performance of retailing firms", International Journal of Information Technology & Decision Making, vol. 3, no. 1, pp. 61-79, 2004.
[43] Sen and A. P. Sinha, "A comparison of data warehousing methodologies", Communications of The ACM, vol. 48 Issue 3, pp. 79- 84, 2005.
[44] Sun and Morwitz, V.G., Stated intentions and purchase behavior: A unified model. International Journal of Research in Marketing, Volume 27(4), 356-366, 2010.
[45] Su, C. T., Hsu, H. H., & Tsai, C. H,"Knowledge mining from trained neural networks", Journal of Computer Information Systems, 42, 61–70,2002.
[46] T. Hertz, A. Hillel, and D. Weinshall, "Learning a Kernel Function for Classification with Small Training Samples," Proc. ACM Int'l Conf. Machine Learning, 2006.
[47] Transportation and Economic Development. [https://people.hofstra.edu/geotrans/eng/ch7en/conc7en/ch7c1en.html]
[48] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases", American Association For Artificial Intelligence. AI Magazine, pp. 37-54, 1996.
[49] UCI Machine Learning Repository – [http://mlearn.ics.uci.edu/databases]
[50] W. Smith, "Applying data mining to scheduling courses at a university", Communications Of AIs; vol. 2005, no. 16, pp. 463-474, 2005.
[51] W. Hugh, A. Thilini, Jr. Matyska, and J. Robert, "Data warehousing stages of growth", Information Systems Management; vol. 18, no. 3, pp.42-51, 2001.
[52] Wai-Ho Au, Member, IEEE, Keith C. C. Chan, Andrew K.C. Wong, Fellow, IEEE,and Yang Wang, Member, IEEE ,"Attribute Clustering for Grouping, Selection, and Classification of Gene Expression Data", Sep. 15, 2004.
[53] W. Smith, "Applying data mining to scheduling courses at a university", Communications Of AIs; vol. 2005, no. 16, pp. 463-474, 2005.
[54] WEKA Software, The University of Waikato. [http://www.cs.waikato.ac.nz/ml/weka].