Evaluation of Data Mining Classification Methodologies for Single Label Learning on benchmark data sets Using R and WEKA

Madana Mohana R
Associate Professor, CSE Department Bharat Institute of Engineering and Technology
Hyderabad, Telangana, India
rmmnaidu@gmail.com

Satyanarayana V
Associate Professor, CSE Department Bharat Institute of Engineering and Technology
Hyderabad, Telangana, India
satya.vujjaini@gmail.com

Abstract: - The objective of this paper is to introduce, explain and compare the performance of single-labeled supervised learning algorithms in R language on benchmark single labeled datasets. Data is rich with hidden information that can be used for intelligent decision making. Classification is a form of data analysis that extracts models describing important data classes, the traditional classification algorithms like decision tree, random forest, support vector machine, naïve-bayes are used under inspection. We have considered four measures (sensitivity, specificity, accuracy, F-measure) of performance here, the observations of all dataset accuracies lead to infer that Random Forest outperforms the other classification methods. For more justification of our result we have implemented the same algorithms with same datasets in weka tool also.

Key-Words: - Decision Tree, Random Forest, SVM, Naïve Bayes, R, Weka

1 Introduction

Data classification is a two step process consisting of a learning step and classification step. Learning step is where classification model is constructed. Classification step is where the model is used to predict the class label for given data. Classification predicts categorical labels [4]. Classification algorithms aim at finding regularities in patterns of training data, this is one of the familiar and popular techniques in machine learning. The dataset is initially partitioned in to training set and testing set randomly and classifier is trained on the former. The testing set is used to evaluate the generalized capability of classifier.

1.1 R Language in Data Mining

The R language which is effective in statistical Analysis also provides effective handling of data mining algorithms [1]. It provides simple decision tree via Rattle package, Naïve Bayesian classifier via e1071 package, Random Forest classifier, and Support Vector Machines classifier is also done using Rattle package.

1.2 WEKA

Weka is a collection of machine learning algorithms for data mining tasks [8]. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes. J48 algorithm is used for decision tree, random forest algorithm is used for random forest, smo algorithm is used for support vector machine and naïve bayes algorithm is used for naïve bayes classifier.

2 Literature Review

There are many classification techniques that are developed for classification problems. Here are few discussions about considered algorithms.

2.1 Decision Tree

A decision tree is a flowchart like structure, where each internal node denotes a test on an attribute [6]. Each branch represents an outcome of the test and each leaf node holds a class label. Given a tuple X for which the associated class labels is unknown; the attributes values of tuple are tested against the decision tree. A path is traced from root to leaf node which holds the class prediction for that tuple. Decision trees can easily be converted to classification trees. During tree construction attribute, attribute selection measures are used to select the attribute that best partition the tuples into

distinct classes. Decision tree's representation of acquired knowledge in tree form is intuitive and generally easy to assimilate by humans.
Measures developed for selecting the best split or often based on the degree of impurity of the child nodes. Examples of impurity measures include:

$$Entropy(t) \quad \sum_{i\ 0}^{c\ 1} p(\frac{i}{t}) \log_2 p(\frac{i}{t}) \qquad (1)$$

$$Gini(t) \quad 1 \quad \sum_{i\ 0}^{c\ 1} [p(\frac{i}{t})]^2 \qquad (2)$$

$$\text{Classification Error } CE(t) \quad 1 \quad \max_i [(p(\frac{i}{t})] \qquad (3)$$

Observe that all three measures attain their maximum value when the class distribution is uniform the minimum values for the measures are attained when all the records belong to the same class because the smaller the degree of impurity, the more the skewed the class distribution [5].

2.2 Naïve Bayes Classifier

Bayesian classifiers are statistical classifiers. They can predict class membership probabilities that is the probability that a given tuple belongs to a particular class. Bayesian classification is based on bayes theorem. Studies comparing classification algorithms have found a simple Bayesian classifier known the naïve bayes classifier to be comparable in performance with decision tree and selected neural network classifiers. Naïve bayes in classifiers assume that the affect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence. Let X be a data tuple, H be a hypothesis such that the data tuple X belongs to a specified class C. for classification problems we want to determine p(H/X), in other words the probability the tuple X belongs to class C ,given that we know the attribute description of X ,P(H/X)is posterior probability of H conditioned on X

$$P(\frac{H}{X}) \quad \frac{P(\frac{X}{H})P(H)}{P(X)} \qquad (4)$$

2.3 Support Vector Machine

Support Vector Machines are the newest supervised machine learning technique. SVMs revolve around the notion of a "margin"- either side of a hyper plane that separates two data classes. Maximizing the margin and thereby creating the largest possible distance between the separating hyper plane and the instances on either side of it has been proven to reduce an upper bound on the expected generalization error. If the training data is linearly separable, then a pair (W, b) exists such that

$$W^T X_i \quad b \quad 1, \quad x_i \quad P \qquad (5)$$

$$W^T X_i \quad b \quad 1, \quad x_i \quad N \qquad (6)$$

With decision rule give by,

$$f_{w,b}(X) \quad \text{sgn}(W^T X_i \quad b) \qquad (7)$$

Where W is termed as the weight vector and b is termed as the bias or threshold. When it is possible to linearly separate two classes, an optimum separating hyper plane can be found by minimizing the squared norm of the separating hyper plane. The minimization can be set up as a convex quadratic programming (QP) problem:

$$\text{Minimize (w, b)} \quad (W) \quad \frac{1}{2}\|W\|^2 \qquad (8)$$

Subject to $y_i(W^T X_i + b) >= 1$, i = 1, ……, l. In the case of linearly separable data, once the optimum separating hyper plane is found, data points that lie on its margin are known as support vector points and the solution is represented as a linear combination of only these points. Other data points are ignored.

2.4 Random Forest

Random forest is a class ensemble methods specifically designed for decision tree classifiers [2]. If combines the prediction made by multiple decision trees where each tree is generated based on the values of independent set of random vectors. Random vectors are generated from a fixed probability distribution. It is theoretically proven that the upper bound for generalization error of a

random forest converges to the following expression. When number of trees is sufficiently large.

$$\text{Generalization Error} \quad \frac{\bar{\rho}(1 - S^2)}{S^2} \qquad (9)$$

$\bar{\rho}$ is the average correlation among the trees and s is a quantity that measures "strengths" of the tree classifiers. The strength of a set of classifiers refers to the average performance of the classifiers. Randomization helps to reduce the co-relation among decision trees so that the generalization error of the ensemble can be improved. A random vector can be incorporated into the tree growing process in many ways. The algorithm for inducing a random forest was developed by Leo Breiman and Adele Cutler.

3 Dataset Description

All datasets are taken from uci repository [7].

3.1 Credit Approval Data

This data set concerns credit card applications. All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data. This dataset is interesting because there is a good mix of attributes -- continuous, nominal with small numbers of values, and nominal with larger numbers of values this dataset contains 690 instances and 16 attributes. The target variable is class. This is a binary class problem having two classes namely '+' and '-'.

3.2 Heart Disease

The dataset contains 303 instances and 14 attributes. The attribute num is the target variable which says about the angiographic disease status. It has 2 classes namely <50 which means the diagnosis status as 0 and >50 means the diagnosis status as 1.

3.3 Diabetes Data

The dataset contains 768 instances and 9 attributes. The attribute class is the target variable which says about the disease status. It has 2 classes namely tested-negative and tested-positive.

3.4 Ionosphere Data

Classification of radar returns from the ionosphere using neural networks. This radar data was collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. The targets were free electrons in the ionosphere."Good" radar returns are those showing evidence of some type of structure in the ionosphere."Bad" returns are those that do not; their signals pass through the ionosphere. There are total 351 instances and 34 variables the target variable is class which is either g (good) or b (bad).

Table 1
SUMMARY OF ALL DATASETS

Dataset	#instances	#attributes	#classes
Credit	690	16	2
Heart	303	14	2
Ionosphere	351	34	2
Diabetes	768	9	2

4 Experimental Setup

The experimental setup is as shown in Fig.1

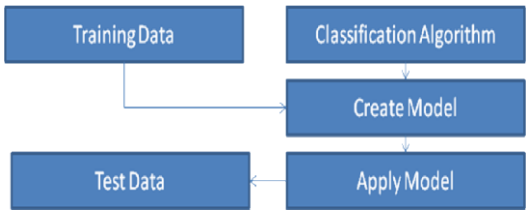

Fig. 1: Overall experimental setup

The experimental setup consists of the following steps

1. Read the data in R Workspace.
2. Split the data into training data (70%) and testing data (30%).
3. Use classification algorithm to generate rules.
4. For each sample in the testing data, use the model to fit the test data (i.e. find the corresponding class label and assign the test sample to that class label).
5. Calculate the classifier performance in terms of sensitivity, specificity, accuracy and F - measure.

5 Proposed Method

The proposed method compares the traditional 4 classification algorithms in R language. R provides many packages out of them Rattle, e107, are used for the analysis of the classification algorithms.

5.1 Rattle (Decision Tree)

To analyze simple decision tree in R language the Rattle (*R Analytical Tool to learn easily)* package is used. Rattle is a GUI based package [3]. It uses simple decision tree. The decision tree generated for the benchmark data set is

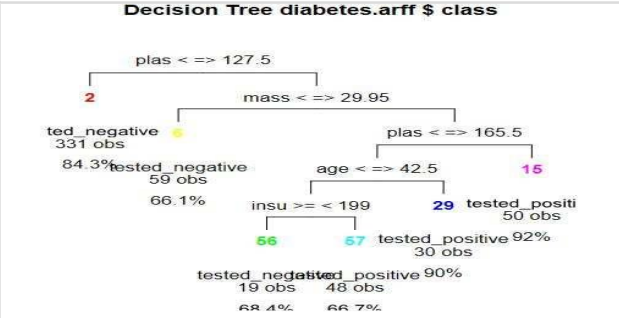

Fig. 2: Tree generated for diabetes dataset using rattle

5.2 Rattle (Random Forest)

Rattle uses the Random Forest package (Liaw and Weiner, 2002) to build a forest of trees. This is an interface to the original random forest code from the original developers of the technique. The summary of Random Forest model is as follows:

Number of observations used to build the model: 537
Missing value imputation is active.

Call:
randomForest (formula = class ~ ., data = crs$dataset[crs$sample, c(crs$input, crs$target)],ntree = 500, mtry =2, importance = TRUE, replace = FALSE, na.action =na.roughfix)
Type of random forest: classification
Number of trees: 500
No. of variables tried at each split: 2

OOB estimate of error rate: 23.46%
Confusion matrix:

	tested_negative	*tested_positive*	*class.error*
tested_negative	*304*	*50*	*0.1412429*
tested_positive	*76*	*107*	*0.4153005*

5.3 Rattle (SVM)

Rattle supports the building of support vector machine (SVM) models using the ksvm package for R. The summary of the SVM model build for diabetes is given below.

Summary of the SVM model (built using ksvm):Support Vector Machine object of class "ksvm" SV type: C-svc (classification)
Parameter: cost C = 1
Linear (vanilla) kernel function.
Number of Support Vectors: 262
Objective Function Value: -257.4034
Training error: 0.206704
Probability model included.

5.4 e1071: Naïve Bayes

Naïve Bayes in e1071 package in R language computes the conditional a-posterior probabilities of a categorical class variable given independent predictor variables using the Bayes rule. Coding for e1071 package for building naïveBayes classifier for Heart Disease dataset is as follows:

```
> ind <- sample(2, nrow(hart), replace=TRUE, prob=c(0.7, 0.3))
> trainData <- hart[ind==1,]
> testData<-hart[ind==2,]
> m <- naiveBayes(num ~ ., data = trainData)
> table(predict(m,as.data.frame(trainData)),trainData$num)

       <50 >50_1
  <50    97    18
  >50_1  14    82
> n<-predict(m,newdata=testData)
> table(n,testData$num)

n       <50 >50_1
  <50    49    12
  >50_1   5    26
```

6 Results

6.1 Measuring the performance

Accuracy:
The accuracy of a classifier is the percentage of the test set tuples that are correctly classified by the classifier.

$$Accuracy \quad \frac{TP \quad TN}{TP \quad TN \quad FP \quad FN} \qquad (10)$$

TP -> number of true positives
TN -> number of true negatives

FN -> number of false negatives
FP -> number of false positive

Sensitivity:
Sensitivity is also referred as True positive rate i.e. the proportion of positive tuples that are correctly identified.

$$Sensitivity \quad \frac{TP}{TP \quad FN} \quad (11)$$

Specificity:
Specificity measures the proportion of negative tuples that are correctly identified.

$$Specificity \quad \frac{TN}{TN \quad FP} \quad (12)$$

F-Measure:
The F- score is used as a single measure of performance of the test. It is harmonic mean of precision and recall.

$$F \quad 2 \quad \frac{\text{Precision} \quad \text{Recall}}{\text{Precision} \quad \text{Recall}} \quad (13)$$

6.2 Credit approval data

The following table.2 show overall efficiency of classifiers for credit approval data in terms of 4 measures:

Table 2
PERFORMANCE OF CLASSIFIERS FOR CREDIT APPROVAL DATA

classifier	*Sensitivity (%)*	*Specificity (%)*	*Accuracy (%)*	*F-measure*
DT	82.79	82.60	82.692	0.8104914
RF	82.60	88.34	85.641	0.84438
SVM	91.30	74.75	82.564	0.87716004
NB	88.88	73.35	79.512	0.72718388

DT->Decision Tree, RF-> Random Forest, SVM->support vector machine, NB->Naïve Bayes

From Fig.3 it is clear that Random Forest classifier performs well for Credit Approval Data.

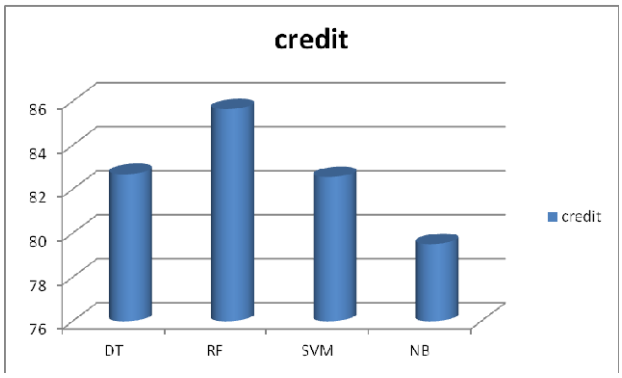

Fig 3: Performance of classifiers for Credit Approval Data

6.3 Diabetes Data

The following table.3 show overall efficiency of classifiers for Diabetes data in terms of 4 measures

Table 3
PERFORMANCE OF CLASSIFIERS FOR DIABETES DATA

classifier	*Sensitivity (%)*	*Specificity (%)*	*Accuracy (%)*	*F-measure*
DT	38.82	91.78	72.294	0.50765414
RF	61.17	82.87	74.892	0.64192853
SVM	47.05	84.93	70.996	0.54413687
NB	66.19	80.74	76.293	0.63080472

DT->Decision Tree, RF-> Random Forest, SVM->support vector machine, NB->Naïve Bayes

From Fig.4 it is clear that Naïve Bayes classifier performs well for Diabetes Data.

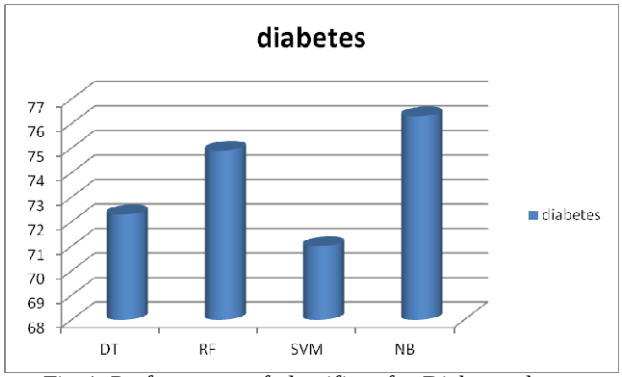

Fig 4: Performance of classifiers for Diabetes data

6.4 Heart Disease Data

The following table.4 show overall efficiency of classifiers for Diabetes data in terms of 4 measures.

Table 4
PERFORMANCE OF CLASSIFIERS FOR HEART DISEASE DATA

classifier	*Sensitivity (%)*	*Specificity (%)*	*Accuracy (%)*	*F-measure*
DT	76.19	75.51	77.824	0.74394103
RF	85.95	87.5	84.444	0.82925579
SVM	80.95	81.25	81.111	0.79993837
NB	83.87	80.32	81.522	0.75321289

DT->Decision Tree, RF-> Random Forest, SVM->support vector machine, NB->Naïve Bayes

From Fig.5 it is clear that Random Forest classifier performs well for Heart Disease Data.

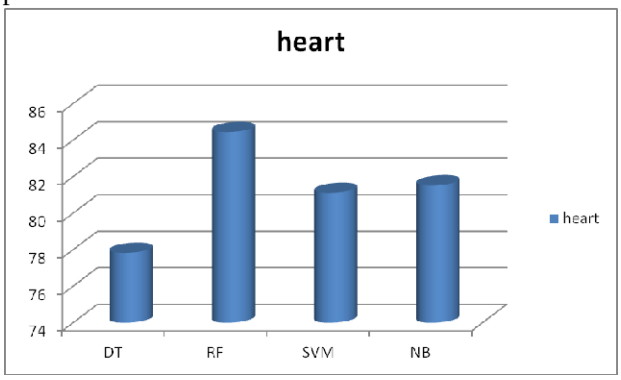

Fig 5: Performance of classifiers for Heart Disease Data

6.4 Ionosphere Data

The following table.5 show overall efficiency of classifiers for Ionosphere data in terms of 4 measures:

Table 5
PERFORMANCE OF CLASSIFIERS FOR IONOSPHERE DATA

classifier	*Sensitivity (%)*	*Specificity (%)*	*Accuracy (%)*	*F-measure*
DT	91.17	86.84	89.623	0.91844965
RF	97.05	86.84	93.396	0.94955763
SVM	91.17	63.15	81.132	0.86103240
NB	95.77	82.85	91.509	0.93789889

DT->Decision Tree, RF-> Random Forest, SVM->support vector machine, NB->Naïve Bayes

From Fig.6 it is clear that Random Forest classifier performs well for Ionosphere Data.

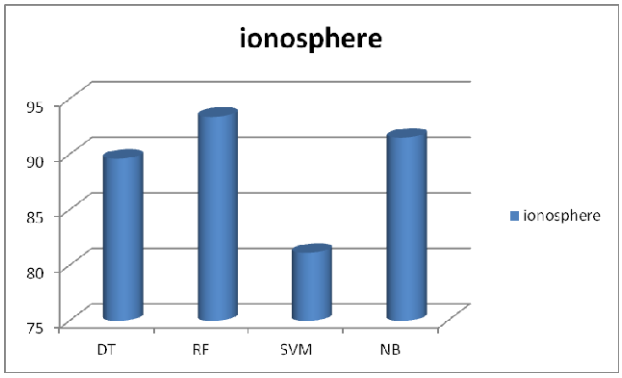

Fig 6: Performance of classifiers for Ionosphere Data

6.5 summary of performance of all classifiers

Table.6 gives the accuracy of 4 classifiers for the chosen datasets in R.

Table 6
ACCURACY OF ALL CLASSIFIERS IN R

classifier	*Diabetes*	*Heart Disease*	*Ionosphere*	*Credit approval*
DT	72.294	77.824	89.623	82.692
RF	74.892	84.444	93.396	85.641
SVM	70.996	81.111	81.132	82.564
NB	76.293	81.522	91.509	79.512

Out of four traditional classifiers random forest performs well and it is shown in fig.7

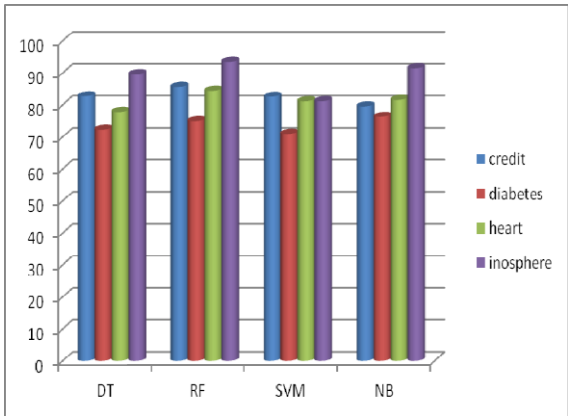

Fig 7: Accuracy of all classifiers in R

For more justification of the result obtained we took the same algorithms, same datasets and with same partitioning of data (70% training and 30% testing as in R) in Weka.
Table .7 gives the accuracy of 4 classifiers for the chosen datasets in Weka.

Table 7
ACCURACY OF ALL CLASSIFIERS IN WEKA

classifier	*Diabetes*	*Heart Disease*	*Ionosphere*	*Credit approval*
DT	76.5217	76.9231	80	85.9903
RF	78.2609	89.011	90.4762	85.5072
SVM	79.1304	85.7143	85.7143	85.5072
NB	76.9565	85.7143	80.9524	75.3623

Out of four traditional classifiers random forest performs well and it is shown in Fig.8

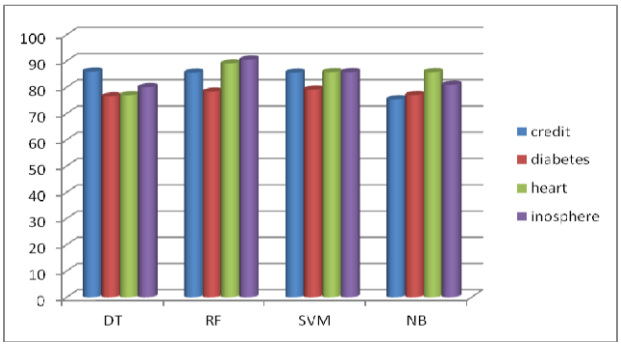

Fig 8: Accuracy of all classifiers in weka

6.6 Averaged performance of all classifiers

Table.8 and Fig.9 gives averaged performances of all classifiers in R and weka.

Table 8
AVERAGED PERFORMANCE IN R AND WEKA

	Decision Tree	*Random Forest*	*SVM*	*Naïve Bayes*
R	79.859	85.814	84.166	79.7463
weka	81.609	84.593	78.951	82.209

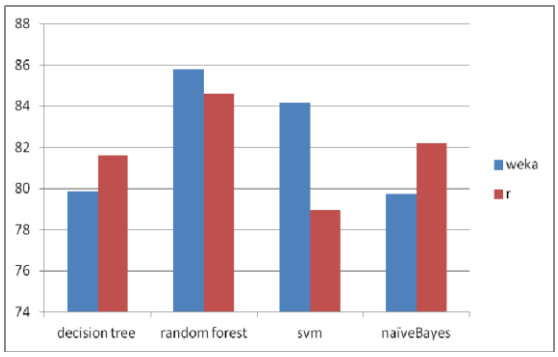

Fig 9: Averaged performance of all classifiers

7 Conclusion

Combining predictions of an ensemble is often more accurate than the individual classifiers that make them up. Statistical, Computational and Representation are three fundamental reasons why an ensemble may work better than a single classifier. By observing fig.9 we can clearly prove the above statements. So that it is clear that in many cases Random Forest gives better accuracy compared to other traditional classifiers.

References:

[1] R Development Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http: //www.R-project.org/., 2013.

[2] Breiman,Leo *Random Forest-Random Features*, 1999.

[3] Graham J Williams, *Rattle: A Data Mining GUI for R*, The R journal Vol. ½,2009.

[4] Jiawei Han and Micheline kamber,book on *Data mining:Concepts and Techniques*, 3rd edition, morgan Kaufinann Publishers,March 2012.ISBN 1-55860-901-6.

[5] Pang-Ning Tan, Vipin Kumar, Michael Steinbach , *Introduction to Data Mining* ,ISBN 978-81-317-1472-0

[6] J. R. Quinlan. *Induction of decision trees*. Machine learning, 1-81-106,1986.

[7] Blake, C. L., & Merz, C. J. *UCI repository of machine learning databases.* http://www.ics.uci.edu/- _mlearn /MLRRepository.html, 1998.

[8] Remco R. Bouckaert,eibe frank,Mark Hall,Richard Kirkby,Peter reutemann,alex seewald,david Scuse *Weka Manual for version 3-6-0*, , December 18,2008.

[9] K. Suresh, R. Madana Mohana, A. Rama Mohan Reddy. *Improved FCM algorithm for Clustering on Web Usage Mining*, IJCSI International Journal of Computer Science Issues (IJCSI), Vol. 8, Issue 1, January 2011.

[10] Yan chang Zhao, *Introduction to Data Mining with R*, UAS, 2014.

[11] Vapnik V, *Statistical Learning Theory*. Wiley, New York, 1998.

[12] Therneau TM, Atkinson B. RPART: recursive partitioning. RPART by B. Ripley. R package version 3.1-41, 2008.