Journal of Theoretical and Applied Information Technology
10th February 2015. Vol.72 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

FUZZY RULE BASED CLASSIFIER FOR SOFTWARE QUALITY DATA

[1] JAYA PAL, [2] VANDANA BHATTACHERJEE
[1]Asstt Prof., Department of CS & Engineering, Birla Institute of Technology, Ranchi, India
[2]Prof. , Department of CS & Engineering Birla Institute of Technology, Ranchi, India
E-mail: [1]jayapal@bitmesra.ac.in,[2]vbhattacharya@bitmesra.ac.in

ABSTRACT

Software quality estimation based on measured attributes from previous similar products is an active field of research. Such estimation models must inevitably handle imprecision and uncertainty and hence soft computing techniques are gaining popularity. This paper presents a Fuzzy rule based classifier for software quality data and its performance is compared with Bayesian classifier. The fuzzy rules have been generated using Fuzzy C-Means clustering. The objectives of this paper are threefold. First, Fuzzy C-Means algorithm is applied to a set of Software Quality data and clusters are generated. The nearest data points to each cluster centroid are used to generate optimum set of fuzzy rules which are refined with the help of train data. These rules are used to label the clusters generated by Fuzzy C-Means algorithm Second, these fuzzy rules are used to classify the test data. Third, naïve Bayes classification method is applied to classify the test data. Confusion matrix is generated and results show that the performance of both the classification methods is comparable.

Keywords: *Fuzzy Clustering, Fuzzy C-Means Algorithm, Fuzzy Rules, Software Quality, Bayes Theorem, Naive Bayes Classification, Laplacian Correction.*

1. INTRODUCTION

Fuzzy logic integrated with data mining techniques becomes one of the key constituents of soft computing in handling the challenges posed by massive collections of natural data [6]. For solving a problem using hard computing its mathematical formulation is required. Most of the real-world problems are too complex to model mathematically, in such cases, hard computing may not be suitable to solve such problems. To solve above problems involving inherent imprecision and uncertainties, soft computing is used. Fuzzy logic control algorithm solves problems that are difficult to address with traditional control techniques. Clustering is a powerful technique of data mining for extracting useful information from a set of data and classifies the data into several clusters based on similarity of the pattern. The nature of the clusters may be either crisp or fuzzy. The boundaries of the crisp clusters are well defined and fixed among themselves whereas fuzzy clusters have vague boundaries [1]. K-means is a popular clustering technique and its variations have proposed to overcome its inherent limitations [10] [11]. The clusters formed by K-means technique are crisp clusters. This technique has been used for software fault prediction [12].Several methods of fuzzy clustering, such as fuzzy ISODATA [2], Fuzzy C-Means [3], Fuzzy K-nearest neighborhood Algorithm [4], potential based clustering [5] and others, have been proposed by various researches

The non-unique partitioning of the data in collection of clusters is the central idea in fuzzy clustering. The membership values of data points are assigned for each of the clusters. The membership value of zero indicates that the data point is not a member of the cluster under consideration. Handling of extreme outliers in many crisp techniques are difficult but the tendency of fuzzy clustering algorithms is to give them very small membership value in surrounding clusters [7].

The membership values with a maximum of one show the degree to which the data point represents a cluster. At the centre of the cluster, data points have maximum membership values and the membership value continuously decreases when we move away from the cluster's centre. Thus fuzzy clustering provides a flexible and robust method for handling natural data with vagueness and uncertainty [8]. In fuzzy clustering, for each cluster each data point will have an associated membership value. The membership value in the range [0, 1] indicates the strength of association in that cluster. The compactness and distinctness of

Journal of Theoretical and Applied Information Technology
10th February 2015. Vol.72 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the clusters are decided based on the intra cluster and inter cluster distances of elements respectively [13]. The nearest data points to each cluster centroid are used to generate optimum set of fuzzy rules.

Software quality is controlled by many types of uncertainties that occur during software development process which makes it difficult for the designer to evaluate the software quality. Software reuse has become a topic of much interest in the software community due to its potential benefits, which include increased product quality and decreased product cost and schedule. Although a large number of metrics have been proposed by researchers to access object-oriented design quality via reusability, they pose problems of their own, the most important being the ability to give a relevant interpretation of the measurement results.

In our research work, estimation of software quality is based on uncertainty, hence we have used Fuzzy Logic and result is compared with Naïve Bayes Classification. In previous similar works, in [16] authors have shown the effectiveness of the incremental estimation of project failure risk using Naïve Bayes classifier. They conclude that Naïve Bayes classifier can provide more stable and robust prediction than Poisson regression by increasing accuracy incrementally. In [17] authors use frequent patterns to estimate the Bayesian probability. They have proposed Enbay, the Entropy Based Bayesian classifier and experimental evaluation performed on real and synthetic datasets shows that its performance is comparable or better than most of the classifiers. In [18] authors use hierarchical clustering and fuzzy clustering using Min-Max method to estimate the quality for students' projects data. From the experimental results it is seen that the fuzzy clustering and hierarchical clustering technique prove to be useful tools in obtaining clusters which can be meaning fully interpreted. The rest of the paper is organized as follows: The rest of the paper is organized as follows: Section 2 presents Fuzzy C-Means clustering, Section 3 presents Naïve Bayes Classification, Section 4 presents Research method, Section 5 presents Experimental Results and Section 6 presents Conclusion.

2. FUZZY C-MEANS CLUSTERING

Fuzzy C-Means (FCM) clustering is one of the most popular fuzzy clustering techniques, in which a particular data point of the set may be member of several clusters with different membership values [1]. The aim of this algorithm is to minimize dissimilarity measure using Euclidean distance of the data points with the pre defined clusters. The similarity of two data points residing in the same cluster will have high value and two points belonging to two different clusters will be dissimilar in nature. It is an iterative algorithm to minimize the dissimilarity measure by updating the cluster centers and membership values of the data points with the clusters.

Consider n-dimensional N data points x_i (i=1, 2,.N) which are to be clustered. Each data point x_i is represented by a vector on n values x_{i1}, x_{i2} …x_{in} . Let C is the number of clusters in which these data points are clustered, where $2 \le C \le N$ and *m* (>1) is a factor indicating the level of cluster fuzziness. Let us consider a membership matrix [μ] with dimension $N \times C$. Thus membership value of the i-th data point with j-th cluster is represented by μ_{ij} which lies in the range [0, 1] as

$$\sum_{j=1}^{C} \mu_{ij} = 1$$

2.1 Fuzzy C-Means Algorithm

The FCM consists of the following steps:

Step 1: Initialize the number of clusters C, $2 \le C \le N$.

Step 2: Select an appropriate level of cluster fuzziness m>1.

Step 3: Initialize membership matrix [μ] of order $N \times C$ *at random, such that* $\mu_{ij} \in [0,1]$

and

$$\sum_{j=1}^{C} \mu_{ij} = 1 \quad \text{for each i.}$$

Step 4: Calculate k-th dimension of j-th cluster centre CC_{jk}

$$CC_{jk} = \frac{\sum_{i=1}^{N} \mu_{ij}^{m} \, x_{ik}}{\sum_{i=1}^{N} \mu_{ij}^{m}} \quad \text{..... (1)}$$

5: Calculate the Euclidean distance between i-th data point and j-th cluster centre as

$$d_{ij} = \left\| CC_j - x_i \right\|$$

Step 6: According to d_{ij} *, update fuzzy membership matrix [μ]*

If $d_{ij} > 0$ *then*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$\mu_{ij} = \frac{1}{\sum_{l=1}^{C} (\frac{d_{ij}}{d_{il}})^{\frac{2}{m-1}}} \quad \text{..... (2)}$$

If $d_{ij} = 0$ then data point coincides with j-th cluster centre CCj and it will have full membership valuei.e. $\sum_i \mu_{ji} = 1$

Step 7: Repeat Step 4 to Step 6 until the change in [μ] will be less than some pre defined termination criteria say $\mathcal{E}$ i.e $\Delta\mu < \mathcal{E}$.

Thus on applying this algorithm, fuzzy clusters of data points will be obtained. The boundaries of the clusters are such that there will be some overlapping of two or more clusters [1]. It is important to note that the performance (quality of clusters) of this algorithm depends on initial membership matrix selected at random.

As the Euclidean distance is considered for dissimilarity measure the cost (objective) function to be minimized is

$$f(\mu, C) = \sum_{j=1}^{C} \sum_{i=1}^{N} \mu_{ij} d_{ij}^2 \quad \text{..... (3)}$$

where dij is the Euclidean distance between i-th data point and j-th cluster.

$$d_{ij} = \| C_j - x_i \|$$

3. NAIVE BAYES CLASSIFICATION

Bayes Theorem/Bayes Rule: Bayes rule is a technique to estimate the likelihood of a property given the set of data as input. Suppose that either hypothesis h_1 orhypothesis h_2 must occur, but not both. Also suppose that x_i is an observable event. Thus Baye's Theorem is

$$P(h_1 / x_i) = \frac{P(x_i / h_1) P(h_1)}{P(x_i / h_1) P(h_{i1}) + P(x_i / h_2) P(h_2)}$$

Here $P(h_i / x_i)$ is called the posterior probability, while $P(h_1)$ ia the prior probability associated with hypothesis h_1. $P(x_i)$ is the probability of the occurrence of data value x_i and $P(x_i / h_1)$ is the conditional probability that , given a hypothesis, the tuple satisfies it. When we have m different hypotheses we have:

$$P(x_i) = \sum_{j=1}^{m} P(x_i / h_j) P(h_j)$$

Thus we have

$$P(h_1 / x_i) = \frac{P(x_i / h_1) P(h_1)}{P(x_i)}$$

Bayes rule allow us to assign probabilities of hypothesis give a data value, $P(h_j / x_i)$.

A simple classification scheme called naïve Bayes classification has been proposed that is based on Bayes rule of conditional probability as stated in above. A classification is made by combining the impact that the different attributes have on the prediction to be made. Given a data value x_i the probability that related tuple t_i is in class C_j is described by $P(C_j / x_i)$.Training data can be used to determine $P(x_i)$, $P(x_i / C_j)$ and $P(C_j)$. From these values Bayes theorem allows us to estimate the posterior probability $P(C_j / x_i)$ and $P(C_j / t_i)$.

Given a training set, the naïve Bayes algorithm first estimates the prior probability $P(C_j)$ for each class by counting how often each class occurs in the training data. For each attribute, x_i, the number of occurrences of each attribute value x_i can be counted to determine $P(x_i)$. Similarly, the probability $P(x_i / C_j)$ can be estimated by counting how often each value occurs in the class in the training data. A tuple in the training data may have many different attributes each with many values. This must be done for all attributes and all values of attributes. The probabilities are used to predict the class membership for a target tuple.

When classifying a target tuple, the conditional and prior probabilities generated from the training set are used to make the prediction . Suppose that tuple t_i has p independent attributes values $\{x_{i1}, x_{i2}, ..., x_{ip}\}$. Knowing $P(x_{ik} / C_j)$, for each class C_j and attribute, x_{ik} we estimate $P(t_i / C_j)$ by

Journal of Theoretical and Applied Information Technology
10th February 2015. Vol.72 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$P(t_i / C_j) = \prod_{k=1}^{p} P(x_{ik} / C_j)$$

At this point in the algorithm, we require prior probabilities for each class and the conditional probability. To calculate, we can estimate the likelihood that is in the class. This can be done by finding the likelihood that this tuple is in each class and then adding all these values. The probability that is in a class is the product of the conditional probabilities for each attribute value. The posterior probability is then found for each class. The class with the highest probability is one chosen for the tuple [14].

Laplacian Correction: In Bayesian classification, a zero probability cancels the effects of all other (posteriori) probabilities involved in the product. To avoid this problem, we can assume that our training data base D, is so large that adding one to each count that we need only make a negligible difference in the estimated probability value in order to avoid the case of probability values of zero. This technique for probability estimation is knownasLaplacian Correction [15].

4. RESEARCH METHOD

4.1 Metrics Used

Software quality factors may be enumerated as follows: portability, usability, reusability, correctness, maintainability etc. This paper focuses on usability of software and metrics were designed and / or adapted from Pal and Bhattacherjee [9] where the authors have developed a Fuzzy Logic System for prediction of software quality.

Description of metrics:

1. GUI (GRAPHICAL USER INTERFACE): GUI was measured as the relative number of forms which were clearly displayed, on a scale of 0-10.

2. MEM (MEANINGFUL ERROR MESSAGE): MEM was measured as the relative number of meaningful error messages displayed by the software, on a scale of 0-1.

3. UM (USER MANUAL): UM was measured as the completeness of the user manual or help file, on a scale of 1-20.

The quality of the ultimate product (program) has been judged by team of three experts who ranked the various projects on a scale of 50-100 for usability and this served as the predicted output.

4.2 Data Gathered

The hundred and ten projects are used to obtain the data. A snapshot of the dataset related to them is depicted in Table 1.

Table 1: Projects id and metrics, Graphical User Interface (GUI), Meaningful Error Message (MEM),User Manual(UM), Software Usability (SU)

Sl No	Project Description	GUI	MEM	UM	SU
1	Proj 1	0	0.5	9	75
2	Proj 22	5	0.5	14	80
3	Proj 33	1	0.4	8	72
4	Proj 41	7	0.7	12	82
5	Proj 52	7	0.7	16	82
6	Proj 60	6	0.6	14	83
7	Proj 71	7	0.8	18	91
8	Proj 82	1	0.2	9	62
9	Proj 93	7	0.5	14	82
10	Proj100	8	0.8	17	92
11	Proj 101	7	0.5	15	90
12	Proj 102	5	0.8	18	89
13	Proj 103	5	0.4	11	75
14	Proj 104	2	0.2	12	61
15	Proj 105	5	0.2	12	75
16	Proj 106	1	0.4	7	59
17	Proj 107	9	0.5	14	89
18	Proj 108	6	0.4	12	81
19	Proj 109	9	0.8	18	91
20	Proj 110	3	0.6	13	76

Input and output Membership Functions (MF) are depicted in Table 2.

Table 2: Membership Function Characteristics

Inputs

Variable Name	Range	MF	Parameters		
			a	B	c
Graphics User Interface	0-10	Low	0	2	4
		Average	2	4	6
		High	4	7	10
Meaningful Error Message	0-1	Low	0.1	0.25	0.4
		Average	0.3	0.5	0.7
		High	0.5	0.85	1
User Manual	1-20	Small	2	6	10
		Medium	8	11	14
		Big	12	16	20

Journal of Theoretical and Applied Information Technology
10th February 2015. Vol.72 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Variable Name	Range	MF	Parameters		
			a	b	c
Software Quality	50-100	Low	50	60	70
		Average	60	80	90
		High	80	90	100

Outputs

4.3 Evaluation Parameters

The performance of a classification model is represented by Confusion Matrix to illustrate the accuracy of the solution to a classification problem.

A confusion matrix contains information about actual and predicted classification system. It gives the picture of errors made by a classification model. Given m classes, a confusion matrix is a m x m matrix where entry ci j indicates the number of tuples from data set D that were assigned to class Cj but where the correct class is Ci. The rows correspond to the known class of the data i.e. the labels in the data. The columns correspond to the predictions made by the model. The value of each of element in the matrix is the number of predictions made with the class corresponding to the column for examples with the correct value as represented by the row. Thus, the diagonal entries show the number of correct classifications made for each class, and the off-diagonal entries show the errors made. The best solution will have only zero values outside the diagonal. Consider three class problems with three classes A, B and C. The abstract confusion matrix of these classes for notation is as shown below:

Actual Class (class level in data)	Predicted Class		
	A	B	C
A	tp_A	e_{AB}	e_{AC}
B	e_{BA}	tp_B	e_{BC}
C	e_{CA}	e_{CB}	tp_C

Where, tp_A,, tp_B and tp_C represent true positive cases of A, B and C. e_{AB} represents error in prediction that an instance is B, similarly e_{AC}, e_{BA}, ..., e_{CB}.

Performance measures from confusion matrix:

Accuracy: It is the overall correctness of the model and is calculated as the sum of correct classifications divided by the total number of classifications. It is defined by:

$$Accuracy = \frac{tp_A + tp_B + tp_C}{tp_A + e_{AB} + e_{AC} + \ldots\ldots\ldots\ldots + e_{CB} + tp_C}$$

Precision: Itis the measure of the accuracy provided thata specific class has been predicted. It is defined by:

$$\Pr ecision = \frac{tp}{tp + fp}$$

where tp and fp are the numbers of true positive and false positive predictions for considered class. The precision for the class A would be defined as:

$$\Pr ecision_A = \frac{tp_A}{tp_A + e_{BA} + e_{CA}}$$

Recall / Sensitivity: It is a measure of the ability of a prediction model to select instances of certain class from a data set. It corresponds to the true positive rate. It is defined by:

$$\mathrm{Re}\, call / Sensitivity = \frac{tp}{tp + fn}$$

where , tp and fn are the numbers of the true positive and false negative predictions for the considered class. is the total number of test examples of the considered class. The recall for the class A would be defined as:

$$\mathrm{Re}\, call_A = Sensitivity_A = \frac{tp_A}{tp_A + e_{AB} + e_{AC}}$$

Specificity: Recall/Sensitivity is related to specificity, which is a measure that is commonly used in two class problems where one is more interested in particular class. Specificity corresponds to the true-negative rate. It is defined by:

$$Specificity = \frac{tn}{tn + fp}$$

where tn and fp are true negative and false positive prediction for the considered class. For class A, the specificity to the true –negative rate for class A (not being a member of class A) and be defined as:

$$Specificity_A = \frac{tn_A}{tn_A + e_{BA} + e_{CA}},$$

$$where\ tn_A = tp_B + e_{BC} + e_{CB} + tp_C$$

Journal of Theoretical and Applied Information Technology
10th February 2015. Vol.72 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5. EXPERIMENTAL RESULTS

5.1 Applying FCM Algorithm

In this paper we are using FCM for cluster numbers 2 to N/2, where N is the number of training data points (number of projects) and then finding which number of clusters is the best for level of fuzziness m. Here, we are getting best cluster number = 3 for $m = 4.5$ on using N=100 data points. We applied the data of hundred projects in FCM and algorithm ended with the cost of clusters as in Table 3.

Table 3: Cost of clusters

Number of clusters	Cost
2	1262.362
3	**1200.869**
4	1494.687
5	1300.411

Choosing the minimum cost of clusters 3(as shown in bold font in the Table 3), their centroids are as in Table 4.

Table 4: Centroids of the minimum cost cluster

Cluster Centers	W	X	Y
C1	1.0211	0.2194	8.9742
C2	6.5816	0.6580	14.9536
C3	5.3948	0.5914	13.4132

Where W values represent GUI, X values represent MEM, Y values represent UM, C1, C2 and C3 are three final cluster centers. A snapshot of the membership values for train data set as obtained by FCM algorithm for best cluster are given in Table 5.

Table 5: Prediction of membership values of train data

Data points	C1	C2	C3
P1	0.3029	0.3164	0.3806
P22	0.1226	0.2839	0.5936
P33	0.3809	0.2863	0.3327
P41	0.1384	0.3683	0.4933
P52	0.1350	0.4552	0.4098
P60	0.1068	0.4011	0.4920
P71	0.1901	0.4547	0.3551
P82	0.7497	0.1196	0.1306
P93	0.1192	0.3987	0.4820
P100	0.1986	0.4440	0.3574

5.2 Allocation of Train Data Points to Clusters

On the basis of values of Table 5, the data points are allotted to cluster as follows. Table 6 presents the cluster number and distance from the cluster centroid.

Table 6: Cluster number and distance

Data points	Cluster Number	Distance from centroid
P1	C3	8.99
P22	C3	**1.21**
P33	C1	9.42
P41	C3	2.26
P52	C2	**2.59**
P60	C3	1.48
P71	C2	6.48
P82	C1	**0.98**
P93	C3	1.84
P100	C2	7.62

where bold font represents the nearest data point to the clusters.

5.3 Generation of Fuzzy Rules

The initial fuzzy rule base consists of $3^3 = 27$ rules. To generate an optimum set of rules, we identify the nearest data points to each cluster centroid .With the help of these points (in this case P_{22}, P_{52} *and* P_{82}) the optimized set of Fuzzy Rules (Rule 1 to Rule 10) are generated. Further with respect to data coverage of all points, Rule 11 to Rule 15 is generated as follows:

Rule 1: If GUI is high, MEM is average and UM is big then SU is high.
Rule 2: If GUI is high, MEM is high and UM is big then SU is high.
Rule 3: If GUI is low, MEM is low and UM is small then SU is low.
Rule 4: If GUI is low, MEM is low and UM is medium then SU is low.
Rule 5: If GUI is average, MEM is average and UM is medium then SU is average.
Rule 6: If GUI is average, MEM is high and UM is medium then SU is average.
Rule 7: If GUI is average, MEM is average and UM is big then SU is average.
Rule 8: If GUI is average, MEM is high and UM is big then SU is high.
Rule 9: If GUI is high, MEM is average and UM is medium then SU is average.
Rule 10: If GUI is high, MEM is high and UM is medium then SU is high.
Rule 11: If GUI is low, MEM is high and UM is medium then SU is average.
Rule 12: If GUI is low, MEM is average and UM is

Journal of Theoretical and Applied Information Technology
10th February 2015. Vol.72 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

medium then SU is average.

Rule 13: If GUI is low, MEM is average and UM is small then SU is low.

Rule 14: If GUI is low, MEM is high and UM is small then SU is average.

Rule 15: If GUI is average, MEM is low and UM is medium then SU is average.

5.4 Labeling Of Clusters For Software Usability Values

On the basis of centroid points we apply fuzzy rules to find the membership values for various clusters and label the clusters accordingly.

Example

For cluster C1, the rule strengths using the fuzzy membership values illustrated as follows:

Rule 1: min (0, 0) = 0.0
Rule 2: min (0, 0) = 0.0
Rule 3: min (0.5, 0.6, 0.27) = 0.27
Rule 4: min (0.5, 0.6, 0.3) = 0.3
Rule 5: min (0, 0) = 0.0
Rule 6: min (0, 0) = 0.0
.
.
Rule 15: min (0, 0) = 0.0

The fuzzy output is the 'fuzzy OR' of all fuzzy outputs of the rules with non-zero strengths.
The competing fuzzy output is the Rule 4 with strength 0.3.
Therefore, μ_{low} **(SQ) = 0.3**

For cluster C2, the rule strengths using the fuzzy membership values illustrated as follows:

Rule 1: min (0.85, 0.21, 0.73) = 0.21
Rule 2: min (0.85, 0.45, 0.73) = 0.85
Rule 3**:** min (0, 0) = 0.0
Rule 4: min (0, 0) = 0.0
.
.
Rule 15: min (0, 0) = 0.0

The fuzzy output is the 'fuzzy OR' of all fuzzy outputs of the rules with non-zero strengths.
Thus the competing fuzzy output is the Rule 2 with strength 0.85.
Therefore, μ_{high} **(SQ) = 0.85**

Similarly, for cluster C3 we get $\mu_{average}$ **(SQ) = 0.46.**

Hence label the clusters by the label (low, average, high) of the centroid data points as

C1 → low
C2 → high

Cluster _id	Cluster label	Train data
Cluster1	Low	P33,P82
Cluster2	High	P52,P71,P100
Cluster3	Average	P1,P22,P41,P60,P93

C3 → average

A snapshot of prediction of cluster analysis of training data points as shown in Table 7.

Table7: Prediction of cluster analysis

5.5 Testing Phase

For the testing phase, data are classified using the optimized fuzzy rules and using Bayes classification. Classification of test data using Bayes classification Fuzzy rules is shown in Table 8 as follows:

Test data	Bayes classification	Optimized Fuzzy rules
Proj 101	average	high
Proj 102	high	high
Proj 103	average	average
Proj 104	low	low
Proj 105	average	average
Proj 106	low	low
Proj 107	average	high
Proj 108	average	average
Proj 109	high	high
Proj 110	average	average

Table 8: Classification of test data using Bayes classification and Fuzzy rules

Actual Class (Bayesian Classification)	PredictedClass (Fuzzy Rules)		
	Low	Average	High
Low	2	0	0
Average	0	4	2
High	0	0	2

The Confusion matrix between Bayesian Classification and Fuzzy Rules is depicted in Table 9 as follows:

Table 9: Confusion Matrix

Performance measures of classes Low, Average and High in the confusion matrix (Table 9) is depicted by Table 10 as follows:

Table 10: Performance measures of classes

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

		Performance measures		
		Accuracy = 0.8		
Classes		Precision	Recall / Sensitivity	Specificity
	Low	1	1	1
	Average	1	0.66	1
	High	0.5	1	0.75

6. CONCLUSION

In this paper Fuzzy C-Means algorithm is applied to a set of Software quality data and clusters are generated. The nearest data points to each cluster centroid are used to generate fuzzy rules which are refined with the help of train data. These clusters centroids are used to generate the initial set of fuzzy rules which may be refined with the help of train data. These fuzzy rules are used to label the clusters with the software quality values. Further, the generated fuzzy rules are used to classify the test data into different software quality categories. Bayes classification is also used to classify the same and results are presented in Table 8.The performance measures shown in Table10 indicate that overall accuracy of prediction by optimized fuzzy rules is 0.8. It should be further noted that precision has the lowest value 0.5 for the "High" class while recall/sensitivity has a minimum of 0.66 for "Average" class and specificity has a minimum of 0.75 for "High" class. The advantage of fuzzy clustering is that even though we can allocate an absolute cluster membership to a data point, it can be done at a more fine granularity level and we can provide the percentage usability. As part of our ongoing work, we are collecting exhaustive set of data so as to develop a model which can be for generalized use.

REFERENCES:

[1] D. K. Pratihar, "*Soft Computing*", Narosa Publishing House Pvt. Ltd., 2009.

[2] J.C. Dunn, "*A fuzzy relative of theISODATA process and its use in detectingcompact well separated clusters*", Journal of Cybernetics, 3, pp.32-57. 1973.

[3]J.C. Bezdek, "*Fuzzy mathematics in pattern classification*", Ph.D.thesis, Applied Mathematics Centre, Cornell University, Ithaca, 1973.

[4] J. Keller, M. R. Gray and J.A. Givens, "*A Fuzzy K-nearest neighbor algorithm*", IEEE Trans. on Systems, Man and Cybernetics, SMC-15, 4, PP. 580-585, 1985.

[5] S. L. Chiu, "*Fuzzy model identification based on cluster estimation*", Journal of Intelligent Fuzzy Systems,2, pp.267-278, 1994.

[6] S. K. Pal, and P. Mitra, "*Data Mining in Soft Computing Framework: A survey*", IEEE transactions on neural networks, vol13, no1, January 2002.

[7] Carl G. Looney, "*A Fuzzy clustering and Fuzzy MergingAlgorithm*", Available: http://citeseer.ist.psu.edu/399498.html

[8] B. Thomas, G., Raju, and S. Wangmo, "*A modified Fuzzy C-Means Algorithm for Natural Data Exploration"*, World Academy of Science, Engineering and Technology 49 2009.

[9] J. Pal, and V. Bhattacherjee, "*A Fuzzy Logic System for Software Quality Estimation*", in proceedings ICIT 2009, pp 183-187, 2009.

[10] P.S. Bishnu,,and V. Bhattacherjee "A *new Initialization Method for K-Means Algorithm using Quad Tree*" NCM2C, 2008, JNU, New Delhi.

[11] P.S. Bishnu and V. Bhattacherjee, "*Divide and conquer based clustering using K-Means algorithm",* Conf. on Information Science, Technology and Management, 2009.

[12] V. Bhattacherjee, and P. S. Bishnu "*Unsupervised learning approach to fault prediction in Software Module*", in proceedings of National Conference on Computing and Systems 2010, Burdwan, India,January2010, pp 91-94, 2010.

[13]J. Pal, and V. Bhattacherjee,, "Entropy-based Fuzzy clustering: A Comparison", in proceedings ofSecondInternational Conference on Computing and Systems, Burdawan University, pp. 321-325, Sept 21-22, 2013.

[14] M. H. Dunhum,, "Data Mining –Introductory and Advanced Topics", Dorling Kindersley(India)Pvt Ltd, Pearson Education,2006.

[15] A. Hand and M., Kamber " Data Mining Concept and Technique", Morgan Kauffman Publishers, Elsevier Inia, New Delhi, 2003.

[16] T.Mori, S. Tamura, S. Kakui, "Incremental Estimation of Project Failure Risk with Naïve Bayes Classifier", ACM / IEEE International Symposium on Empirical Software Engineering and Measurement, pp.283-286, 2013.

[17] E. Baralis, L. Cagliero, and P. Garza, "EnBay: A Novel Pattern-Based Bayesian Classifier", IEEE Transactions on Knowledge and Data Engineering, Vol. 25, No. 12, December 2013.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[18] J. Pal, and V. Bhattacherjee , "Hierarchical Cluster Generation for Software Quality: A Comparative Approach", International Journal of Engineering and Technology (IJET),vol 6 ,No 4, Aug-Sep pp. 1827- 1839,2014.