Manjeet Kumar al. International Journal of Recent Research Aspects ISSN: 2349-7688, Vol. 2, Issue 2, June 2015, pp. 62-66

Classification Rule Discovery for Diabetes Patients Using k-NN

Raj Kumar[1], Manjeet Kumar[2]

[1]Assistant Professor, CSE, Jind Institute of Engg. & Tech., Jind, India
[2]M.Tech Scholar, CSE, Jind Institute of Engg. & Tech., Jind, India

***Abstract*—Classification is a Data Mining function that assigns items in a collection to target categories or classes. There are various types of the traditional classification techniques like Naïve Bayesian, ID3, C4.5, CART, k-NN, k-mean, SVM etc. In some algorithms eager approach is used while some algorithms follow the lazy approach. One of algorithm which uses the Lazy approach in k-NN. In this paper evolution of the k-NN algorithm for the classification rule discovery is done. The algorithm is applied on the diabetes patient data for the purpose of classification rule discovery.**

***Keywords*— Classification, DM, k-NN, KDD.**

I. INTRODUCTION

Data mining or knowledge discovery is needed to make sense and use of data. Knowledge discovery in the data is the non-trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns of data [1]. Data mining is the core step of Knowledge Discovery in Database (KDD) and interdisciplinary field include database management system, machine learning, statistics, neural network, fuzzy logic etc. Any of the technique may be integrated depending on the kind of data to be mined. The research in KDD is expected to generate a large variety of systems because diversity of disciplines to be contributed. Therefore a comprehensive classification system is required able to distinguish between the systems and identify the most required by the user. The major issue involved with the classification rule mining is to identify a dataset for a small number of rules to serve as classifier for predicting the class of any new instance. The classification algorithm should be accurate, simple and efficient. The existing classification algorithm assuming that the input data is drawn from a pre-defined distribution having stationary majors. Therefore these algorithms perform poorly when used to infer real world datasets. [2].

II. K-NEAREST NEIGHBOR:

The nearest-neighbor method is perhaps the simplest of all algorithms for predicting the class of a test example. The training phase is trivial: simply store every training example, with its label. To make a prediction for a test example, first compute its distance to every training example. Then, keep the k closest training examples, where k=1 is a fixed integer. Look for the label that is most common among these examples. This label is the prediction for this test example. Using the same set notation as above, the nearest-neighbor method is a function of type (X × Y) n × X ! Y. A distance function has type X × X! R. This basic method is called the k-NN algorithm. There are two major design choices to make: the value of k, and the distance function to use. Ties can also arise when two distance values are the same. An implementation of k-NN needs a sensible algorithm to break ties; there is no consensus on the best way to do this [4]. The k-NN algorithm was originally suggested by Cover and Hart [5], nowadays, it is the most usable classification algorithm **k-nearest neighbors algorithm (k-NN)** is a method for classifying objects based on closest training examples in the feature space. k-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification.

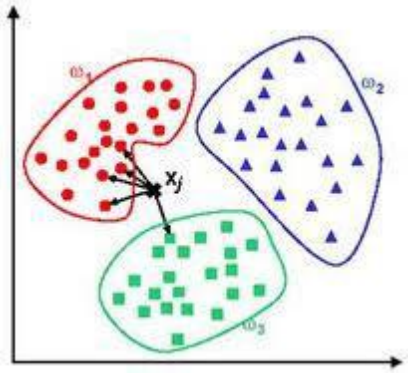

Figure 1 k-NN classifier

Various distance functions for k-NN

A Euclidean distance with different scaling factors c (a1…ad) for different dimensions: [GƆ1 (xj(1) - xnew(1)) 2+ … + GƆd (xj(d) - xnew(d))2]1/2

The k-nearest n neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of its nearest neighbor. The same method can be used for regression, by simply assigning the property value for the object to be the average of the values of its k nearest neighbors. It can be

useful to weight the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. (A common weighting scheme is to give each neighbor a weight of 1/d, where d is the distance to the neighbor. This scheme is a generalization of linear interpolation). This algorithm operation is based on comparing a given new record with training records and finding training records that are similar to it. Each record with n attributes represents a point in an n-dimensional space. In this algorithm nearest is defined in terms of a distance metric such as Euclidean distance [6]-[8].

Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sqrt{\sum_{i=1}^{k}(|x_i - y_i|)^q}\right)^{1/q}$

procedure Instance Base Learner(Testing Instances) for each testing instance

{
find the k most nearest instances of the training set according to a distance metric Resulting Class= most frequent class label of the k nearest instances
}

III. IMPLEMENTATION & EVALUATION OF K-NN:

The k-NN algorithm is implemented on the diabetes patient data for classification rule discovery on rapid miner.

A. DIABETES PATIENTS DATA SET:

The proposed algorithm is applied on the data of the diabetes patients [3]. The given dataset of the diabetes patients has 768 instances with the following nine attributes.

1. Pregnancies (PRG): Number of pregnancies.
2. PG Concentration (PGC): Plasma glucose at two hours in an oral glucose tolerance test.
3. Diastolic BP (DBP): Diastolic Blood Pressure (mm Hg).
4. Tri Fold Thick (TFT): Triceps Skin Fold Thickness (mm).
5. Serum Ins (SI): 2-Hour Serum Insulin (mu U/ml).
6. BMI: Body Mass Index: (weight in kg/ (height in m)^2).
7. DP Function (DPF): Diabetes Pedigree Function.
8. Age: Age (years).
9. Diabetes(DBT): Whether or not the person has diabetes However in the diabetes patients data set there are 768 instances are there but in this paper we have used only 20 instances , the dataset used for the evaluation is given in the below table 1.

PRG	PGC	DBP	TFT	SI	BMI	DPF	Age	DBT
6	148	72	35	0	33.6	0.627	50	Sick
1	85	66	29	0	26.6	0.351	31	Healthy
8	183	64	0	0	23.3	0.672	32	Sick
1	89	66	23	94	28.1	0.167	21	Healthy
0	137	40	35	168	43.1	2.288	33	Sick
5	116	74	0	0	25.6	0.201	30	Healthy
3	78	50	32	88	31	0.248	26	Sick
10	115	0	0	0	35.3	0.134	29	Healthy
2	197	70	45	543	30.5	0.158	53	Sick
8	125	96	0	0	0	0.232	54	Sick
4	110	92	0	0	37.6	0.191	30	Healthy
10	168	74	0	0	38	0.537	34	Sick
10	139	80	0	0	27.1	1.441	57	Healthy
1	189	60	23	846	30.1	0.398	59	Sick
5	166	72	19	175	25.8	0.587	51	Sick
7	100	0	0	0	30	0.484	32	Sick
0	118	84	47	230	45.8	0.551	31	Sick
7	107	74	0	0	29.6	0.254	31	Sick
1	103	30	38	83	43.3	0.183	33	Healthy
1	115	70	30	96	34.6	0.529	32	Sick

TABLE I

This data set is divided in the following two data sets.

(i) **Training Data Set:** Training data set is the subset of data set which is used to make the model

(ii) **Applied data set:** The applied data set is the whole data set which is applied on the model tested by the training data set.

Confusion Matrix:

A confusion matrix is accuracy measurement tool for data mining classification. Accuracy of a classifier on a given test set is the percentage of test tuples that are correctly classified by a classifier [9]. A confusion matrix is generally of the form

		Actual class	
Predicted class		C1	C2
	C1	TP	FP
	C2	FN	TN

Table 2: layout of confusion matrix

Where

TP: True positive
FP: False Positive
FN: False Negative
TN: True negative

B. Model:

Model of k-NN for our data set is designed in rapid miner is shown in below figure.

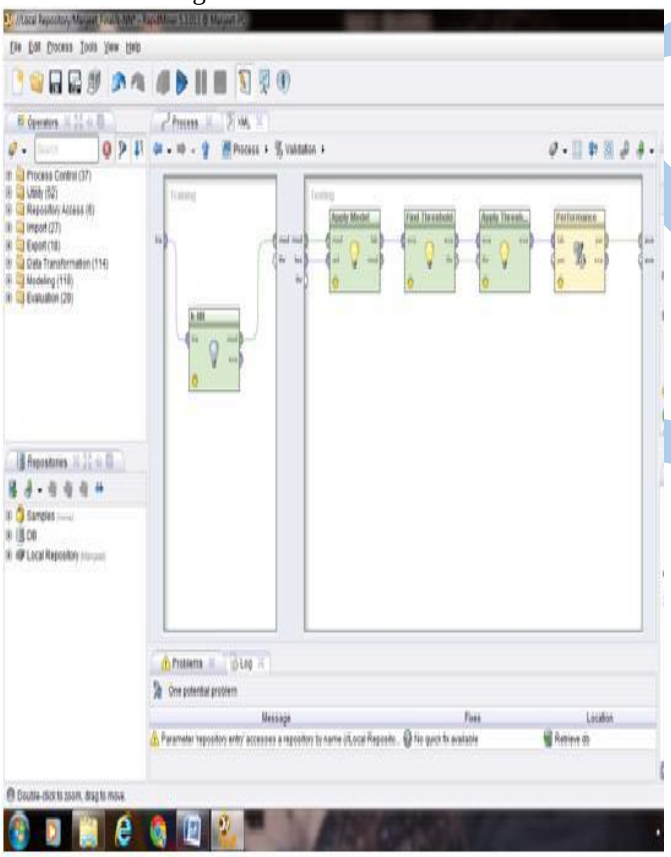

Figure 2: Model for k-NN

In k-NN only two parameters are there to take in consideration. First one is size of nearest neighbors i.e. value of k and other one is distance function. Mainly we take in consideration Euclidian distance function to find out nearest neighbors. We start with minimum value of k according to our data that is taken as 1 here, i.e. k=1.

C. Output in Table View (k=1)

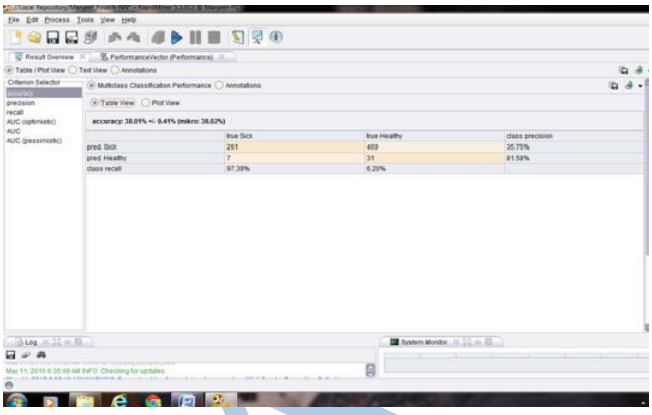

Figure 3 Accuracy result of k-NN (k=1)

Out of the 768 instances of data set 261 instances of class sick are predicted true i.e. they were of class sick and also predicated as class sick. 469 instances of class healthy are predicted as sick. 31 instances of class healthy are predicted as healthy. 7 instances of class sick are predicted as healthy. Total Number of Instance in data-set - 768 Accuracy Percentage – ((True Positive + True Negative)/ Total no. of Instance)*100 According to our data-set ((261 + 31)/768)*100 = 38.02 % (Accuracy) So we can say that accuracy for the classification is 38.02%. So the proposed algorithm is able to get a prediction error about 61.98%. Precision Percentage – (True Negative/ (True Negative + False Negative))*100

According to our data-set (31/ (31+7)*100) = 81.58 % (Precision)

So we can say that precision for the classification is 81.58% (Healthy class). So the proposed algorithm is able to get a precision ratio about 18.42% (Sick class).

Recall Percentage – ((True Negative)/ (False Positive + True Negative)) *100

According to our data-set (31/ (469+31)*100) = 6.20 % (Recall)

So we can say that recall for the classification is 6.20% (Healthy class). So the proposed algorithm is able to get a recall ratio about 93.80% (Sick class).

D. Output in Table View (k=5)

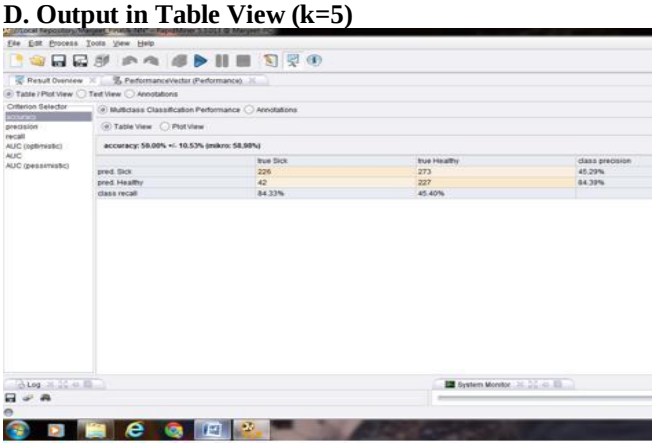

Figure 4 Accuracy result of k-NN (k=5)

Out of the 768 instances of data set 226 instances of class sick are predicted true i.e. they were of class sick and also predicated as class sick. 273 instances of class healthy are predicted as sick. 227 instances of class healthy are predicted as healthy. 42 instances of class sick are predicted as healthy.
Total Number of Instance in data-set - 768
Accuracy Percentage – ((True Positive + True Negative)/ Total no. of Instance)*100 According to our data-set ((226 + 227)/768)*100 = 58.98 % (Accuracy) So we can say that accuracy for the classification is 58.98%. So the proposed algorithm is able to get a prediction error about 41.02%. Precision Percentage – (True Negative/ (True Negative + False Negative))*100 According to our data-set (227/ (227+42)*100) = 84.39 % (Precision) So we can say that precision for the classification is 84.39% (Healthy class). So the proposed algorithm is able to get a precision ratio about 15.61% (Sick class). Recall Percentage – ((True Negative)/ (False Positive + True Negative)) *100 According to our data-set (227/ (273+227)*100) = 45.40 % (Recall) So we can say that recall for the classification is 45.40% (Healthy class). So the proposed algorithm is able to get a recall ratio about 54.60% (Sick class).

E. Output in Table View (k=10)

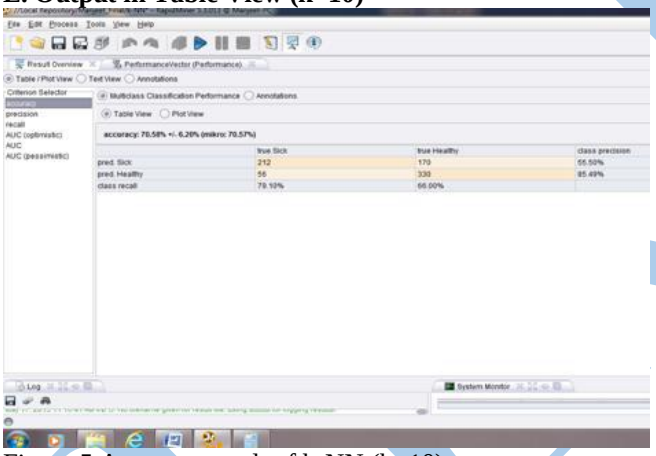

Figure 5 Accuracy result of k-NN (k=10)

Out of the 768 instances of data set 212 instances of class sick are predicted true i.e. they were of class sick and also predicated as class sick. 170 instances of class healthy are predicted as sick. 330 instances of class healthy are predicted as healthy. 56 instances of class sick are predicted as healthy.
Criteria

Criteria	K	K=1	K=5	K=10
Accuracy	38.01 %	59.00%	70.58%	
Precision	81.58%	84.39%	85.79%	
Recall	6.20%	45.40%	66.00%	

Table 3: Binary classification using k-NN

Total Number of Instance in data-set - 768
Accuracy Percentage – ((True Positive + True Negative)/ Total no. of Instance)*100 According to our data-set ((212 + 330)/768)*100 = 70.57 % (Accuracy) So we can say that accuracy for the classification is 70.57%. So the proposed algorithm is able to get a prediction error about 29.43%. Precision Percentage – (True Negative/ (True Negative + False Negative))*100 According to our data-set (330/ (330 + 56)*100) = 85.49 % (Precision) So we can say that precision for the classification is 85.49% (Healthy class). So the proposed algorithm is able to get a precision ratio about 14.51% (Sick class). Recall Percentage – ((True Negative)/ (False Positive + True Negative)) *100 According to our data-set (330/ (170+330)*100) = 66.00 % (Recall) So we can say that recall for the classification is 66.00% (Healthy class). So the proposed algorithm is able to get a recall ratio about 34.00% (Sick class).

IV. CONCLUSION:

We can observe that the area under the curve going on decreasing as we increase the value of nearest neighbours. Hence we can say that smaller the size of neighbourhood circle grater will be the accuracy, as the area under the curve shows that more members related to that particular class in case of binary as well as multiclass classification. As k-NN is considered as a lazy algorithm still its performance for binary labeled attribute is quite better then multiclass. We have observed that for k=10 accuracy of classification is good. For K=10 accuracy is 70.58%. So a prediction error of 29.42% is found. We apply k-NN for the binary classification rule discovery. Accuracy can be enhanced by using hybrid approaches with k-NN. Also work can be done to improve the accuracy of k-NN for multiclass classification.

V. REFERENCES:

[1]. Fayyad, Piatetsky-Shapiro, Smyth and Uthurusamy", Advances in Knowledge Discovery and Data Mining", (Chapter 1), AAAI/MIT Press 1996.
[2]. Raj Kumar, Dr. Anil Kr. Kapil, Anupam Bhatia ,"Modified Tree Classifiation in Data Mining Global Journal of Computer Science and Technology, Vol. 12, Issue 2 (Ver. 1.0), 2012, pp. 59-62
[3]. http://www.liacc.up.pt/ML/statlog/datasets/diabetes/diabetes.doc.html
[4]. Charles Elkan, "Nearest Neighbor Classification", elkan@cs.ucsd.edu, January 11, 2011.
[5]. Cover & Hart , "Nearest neighbor pattern classification", IEEE Transactions on Information Theory, 1967.
[6]. Hastie, T., & Tibshirani, "Discriminant adaptive nearest neighbor classification", R. (1996), IEEE Trans. PAMMI, 18(6), 607-616.
[7]. Vincent, P., & Bengio, Y. "K-local hyper plane & convex distance nearest neighbor algorithms", (2001), Adv Neural Inf Process Syst, 14, 985-992.
[8]. Domeniconi, C., & Gunopulos, D. "Efficient local flexible nearest neighbor classification ",(2002). In proceedings of the 2nd SIAM International Conference on Data Mining.
[9]. Jaiwei Han, Micheline Kkamber, "Data Mining

Concepts and Techniques", Morgan Kaufmann Publishers, 2006, pp 360-361

BIOGRAPHY.

Raj Kumar is working as Asst. Prof. in the deptt. Of Computer Sc. & Engg. at Jind Institute of Engg. & Technology, Jind(Haryana), India. He obtained his M.Tech. degree from Chaoudary Devi Lal University,Sirsa and MCA degrees from Krurukshetra University, Kurukshetra. His area of Research is Data Mining

Manjeet Kumar is a resarch student. His interested area broadly within Data-mining. He received his B.Tech from Sri Shukmani Institute of Engineering & Technology, Dera-bassi,Mohali. Affiliated to Punjab Technical University, Jallander, India in 2005-08 and presently pursing his M.Tech (Data-mining) from Jind Institute of Engineering & Technology, jind (Haryana) Approved by AICTE, New Dlhi & Affiliated to Kurukshetra University, Kurukshetra, Haryana, India

