www.ijecs.in
International Journal Of Engineering And Computer Science
Volume 8 Issue 11 November 2019, Page No. 24869-24877
ISSN: 2319-7242 DOI: 10.18535/ijecs/v8i10.4383

Modification of KNN Algorithm

Shubham Pandey, Vivek Sharma, Garima Agrawal

#Computer Science Department, VIT University Near Katpadi Road, Vellore, Tamil Nadu 632014,India

Abstract
K-Nearest Neighbor (KNN) classification is one of the most fundamental and simple classification methods. It is among the most frequently used classification algorithm in the case when there is little or no prior knowledge about the distribution of the data. In this paper a modification is taken to improve the performance of KNN. The main idea of KNN is to use a set of robust neighbors in the training data. This modified KNN proposed in this paper is better from traditional KNN in both terms: robustness and performance. Inspired from the traditional KNN algorithm, the main idea is to classify an input query according to the most frequent tag in set of neighbor tags with the say of the tag closest to the new tuple being the highest. Proposed Modified KNN can be considered a kind of weighted KNN so that the query label is approximated by weighting the neighbors of the query. The procedure computes the frequencies of the same labeled neighbors to the total number of neighbors with value associated with each label multiplied by a factor which is inversely proportional to the distance between new tuple and neighbours. The proposed method is evaluated on a variety of several standard UCI data sets. Experiments show the significant improvement in the performance of KNN method.

Index Terms—KNN,Modified KNN,Weighted KNN, KNN Classification, Modified KNearest Neighbor, Weighted K-Nearest Neighbor, Neighbor Validity

1 Introduction

Nowadays, recognition system is used in many applications which are related to different fields that have different nature. Pattern recognition is about assigning labels to objects which are described by a set of values named attributes or features[21]. K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique [1].There are three major types of pattern recognition trends:unsupervised, semi-supervised and supervised learning. Inthe supervised category, also called classification or regression, each object of the data comes with a preassigned class label. In other hand, there is a teacher saying the true answer. The task is to train a classifier to perform the labeling, using the teacher. A procedure which tries to leverage the teacher's answer to generalize the problem and obtain his knowledge is learning algorithm. Most often this procedure cannot be described in a human understandable form, like Artificial Neural Networks classifiers. In these cases, the data and the teacher's labelings are supplied to the machine to run the procedure of learning over the data. Although the classification knowledge learned by the machine in this process might be obscure, the recognition accuracy of the classifier will be the judge of its quality of learning or its performance [1].In some new classification systems, it is tried to investigate the errors and propose a solution to compensate them [2-5]. There are many classification and clustering methods as SAwell as the combinational approaches [6-8]. While the supervised learning tries to learn from the true labels or answers of the teacher, in semi-supervised the learner conversely uses teacher just to approve or not to approve the data in total. It means that in semi-supervised learning there is not really available teacher or supervisor. The procedure first starts with fully random manner, and when it reaches the state of final, it looks to the condition whether he wined or losed. For example in the chess game, take it in consideration, that there may be none supervisor, but you gradually train to play better by trail-and-error process and looking at the end of the game to find you wined or losed.KNearest Neighbor (KNN) classification is one of the most fundamental and simple classification methods. When there is little or no prior knowledge about the distribution of the data, the KNN method should be one of the first choices for classification. It is a powerful non-parametric classification system which bypasses the problem of probability densities completely [9]. The main idea is to classify an input query x into the most frequent tag in the set of its neighbor tags. The KNN rule classifies x by assigning it the label most frequently represented among the K nearest samples; this means that, a decision is made by examining the labels on the K-nearest neighbors and taking a vote. It is first introduced by Fix and Hodges in 1957 [10]. Later in 1967, KNN is looked at in theoretic perspective [11].

Once such consideration of KNN classification were established, a long line of investigation ensued including new rejection approaches [12], refinements with respect to Bayes error rate [13], distance weighted approaches [14, 15], soft computing [16] methods and fuzzy methods [17, 18]. ITQON et al. in [19] proposed a classifier, TFkNN, aiming at upgrading of distinction performance of KNN classifier and combining plural KNNs using testing characteristics. Their method not only upgrades distinction performance of the KNN but also brings an effect stabilizing variation of recognition ratio; and on recognition time, even when plural KNNs are performed in parallel, by devising its distance calculation it can be done not so as to extremely increase on comparison with that in single KNN. Some KNN advantages can be as follows: simplicity, robustness to noisy training data, and effectiveness in the adequate training data. It has some disadvantages such as: high computation cost in a test query, the large memory to implement, low accuracy rate in multi-dimensional data sets, parameter K, unclearness of distance type. Shall we use all attributes or certain attributes only [20]? In this paper a new interesting algorithm is proposed which partially overcomes the low accuracy rate of KNN. Beforehand, it preprocesses the train set, computing the validity of any train samples. Then the final classification is executed using weighted KNN which is employed the validity as the multiplication factor..

2 Proposed Method

2.1 Methodology

The main idea of the presented method is assigning the class label of the queried instance into K validated data training points nearest to it.First, the validity of all data samples in the train set is computed. Then, a weighted KNN is performed on any test samples. Here is the pseudo code of the proposed algorithm.

Pseudo-code of the MKNN Algorithm: For all tuples in trainSet,Find its k nearest neighbours,then find validity by, Validity= (Number of neighbours with same class)/k

For all tuples in the test set, do: Find its k nearest neighbours

Find the weight of each class by adding the weights of individual neighbours having same class,where wieght of individual neighbours is,

WeighIndiv = (validity of neighbour)*(1-(Its normalised distance))

The predicted class will be class with maximum weight.

2.2 Calculating the validity

In the MKNN algorithm, every training sample must be validated at the first step. The validity of each point is computed according to its neighbors. The validation process is performed for all train samples once. After assigning the validity of each train sample, it is used at the second step as impact or weight of the points in the ensembles of neighbors which the point is selected to attend.To validate a sample point in the train set, the *H* nearest neighbors of the point is considered. Among the *H* nearest neighbors of a train sample *x*, *validity(x)* counts the number of points with the same label to the label of x divided by *H* and this count is associated with that tuple for any further use as validity.

2.3 Obataining the result using weighted KNN

Weighted KNN is one of the variations of KNN method which uses the K nearest neighbors, regardless of their classes, but then uses weighted votes from each sample rather than a simple majority or plurality voting rule. Each of the K samples is given a weight based on their own validity and their distance from the new data.

These weighted votes are then summed for each class, and the class with the largest total vote is chosen.

Vote for class of each new tuple is done by multiplying the validity of k nearest neighbours with 1-(normal distance between the new tuple and neighbors).This technique has the effect of giving greater importance to the reference samples that have greater validity and closeness to the test sample. So, the decision is less affected by reference samples which are not very stable in the feature space in comparison with other samples. In other hand, the multiplication of the validity measure on distance based measure can overcome the weakness of any distance based weights which have many problems in the case of outliers. So, the proposed MKNN algorithm is significantly stronger than the traditional KNN method which is based just on votes forn the training set tuples.

Pseudo code for prediction: Begin For Neighbour with same class do Vote += Neighour's Validity * (Normal Distance Of the neighbour from the new tuple) Return Class With Maximum Vote End

3 Result Analysis

Result Obtained on Using the Proposed Method on four different UCI data sets is shown in this section.

3.1 How Accuracy is obtained?

Data set is sampled into train set and test test by randomly assigning tuples to each based on the split factor assigned in the program.

For each data set 10 iterations are performed on randomly selected test and train sets to generalize the result obtaine

3.2 Accuracy Tables

1. Credit Card Data Set

Data Set Used	No Of Instances	No Of Attributes	No of classes
credit card clients	2000	24	2

Accuracy recorded for Credit Data Set

Instance	KNN Accuracy	Modified KNN Accuracy
1	63.58974359	70.67307692
2	70.6185567	70
3	73.65853659	73.42995169
4	70.58823529	72.37569061
5	65.87677725	72.51184834
6	63.82978723	72.63681592
7	73.76237624	69.3877551
8	69.56521739	73.97959184
9	75.52083333	75.6097561
10	64.79591837	71.11111111
Average:	69.1805982	72.17155976

2.Soybean Data Set

Data Set Used	No Of Instances	No Of Attributes	No of classes
Soybean	307	35	19

Accuracy recorded for Soyabean Data Set

Instance	KNN Accuracy	Modified KNN Accuracy
1	86.66666667	83.33333333
2	93.10344828	84.84848485
3	80	92.85714286
4	86.20689655	88.46153846
5	85	92.10526316
6	76.47058824	91.66666667
7	86.66666667	84.375
8	72.22222222	92
9	92	91.89189189
10	96.875	86.20689655
Average	85.52114886	88.77462178

3. Teaching Assitant Data Set

Data Set Used	No Of Instances	No Of Attributes	No of classes
Teaching Assistant Evaluation	151	5	2

Accuracy Recorded For Teaching Assitant Evalutaion Data Set

Instance	KNN Accuracy	Modified KNN Accuracy
1	30	78.57142857
2	47.05882353	70
3	33.33333333	53.84615385
4	55.55555556	44.44444444
5	47.05882353	47.05882353
6	25	60
7	63.63636364	78.57142857
8	46.15384615	45.45454545
9	50	55
10	20	57.89473684
Average:	41.77967457	59.08415613

4.Iris Data Set

Data Set Used	No Of Instances	No Of Attributes	No of classes
Iris	150	4	3

Accuracy Recorded for Iris data Set

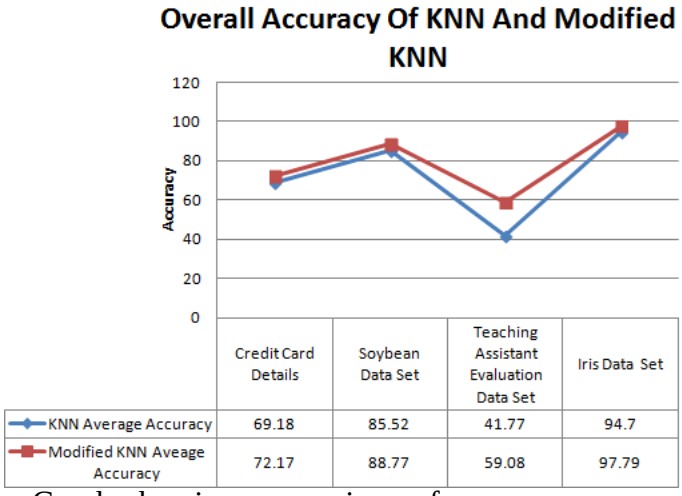
Instance	KNN Accuracy	Modified KNN Accuracy
1	100	100
2	93.75	93.75
3	90.47619048	94.73684211
4	92.85714286	100
5	100	100
6	86.66666667	100

7	100	100
8	90	89.47368421
9	100	100
10	93.33333333	100
Average:	94.70833333	97.79605263

From the above tables it is evident that the proposed modified KNN has significant improvement in average accuracy of prediction for new tuples.Another point that should be noted is modified KNN has better result than KNN in almost 90% of the instances.We have run these algorithms for data set with different number of attributes , instances and classes and obtained favourable results.

Graphical view of the above mentioned results can be seen in appendix-A.

3.3Graph



Graph showing comparison of average accuracy of KNN and modified KNN on the four data sets.(Fig 1)

All the other graphs are attached at the end of the paper.

(Please see appendix A)

FIG 1.Comparision Of KNN and Modified KN

4 Conclusion

In this paper, a new algorithm for improving the performance of KNN classifier is proposed which is called Modified K-Nearest Neighbor. The proposed method which considerably improves the performance of KNN method employs a kind of preprocessing on train data. It adds a new value named Validity to train samples which cause to more information about the situation of training data samples in the feature space. The validity takes into accounts the value of stability and robustness of the any train samples regarding with its neighbors.Applying the weighted KNN which employs validity as the multiplication factor along with the Euclidian diatance of the new tuple and its neighbours yields to more robust classification rather than simple KNN method, efficiently. The method evaluated on four different data sets: Credit Card Details, Soyabean, Teaching Assistant, Iris . The results confirm authors' claim about its robustness and accurateness unanimously. So this method is better in noisy datasets and also in the case of outliers. Since the outliers usually gain low value of validity, it considerably yields to robustness of the MKNN method facing with outliers.

5 Acknowledgments

The authors wish to thank Prof Thendral P for his support and guidance throughout this project.

6 References

[1] L. I. Kuncheva, Combining Pattern Classifiers, Methods and Algorithms, New York: Wiley, 2005

[2] H. Parvin, H. Alizadeh, B. Minaei-Bidgoli and M. Analoui, A n Scalable Method for Improving the Performance of Classifiers in Multiclass Applications by Pairwise Classifiers and GA‖, In Proc. of the Int. Conf. on Networked Computing and advanced Information Management byIEEE CS, (NCM08), Sep. 2008

[3] H. Parvin, H. Alizadeh, M. Moshki, B. Minaei- Bidgoli and N. Mozayani, D ivide & Conquer Classification and Optimization by Genetic Algorithm‖, In Proc. of the Int. Conf. on Convergence and hybrid Information Technology by IEEE CS, (ICCIT08), Nov. 11-13, 2008

[4] H. Parvin, H. Alizadeh, B. Minaei-Bidgoli and M. Analoui, CC HR: Combination of Classifiers using Heuristic Retraining‖, In Proc. of the Int. Conf. on Networked Computing and advanced Information Management by IEEE CS, (NCM 2008), Korea Sep. 2008.

[5] H. Parvin, H. Alizadeh and B. Minaei-Bidgoli, New Approach to Improve the Vote-Based Classifier Selection‖, In Proc. of the Int. Conf. on Networked Computing and advanced Information Management by IEEE CS, (NCM 2008), Korea Sep. 2008

[6] H. Alizadeh, M. Mohammadi and B. Minaei-Bidgoli, Neu ral Network Ensembles using Clustering Ensemble and Genetic Algorithm‖, In Proc. of the Int. Conf. on Convergence and hybrid Information Technology by IEEE CS, (ICCIT08), Nov. 11-13, 2008, Busan, Korea.

[7] H. Parvin, H. Alizadeh and B. Minaei-Bidgoli, A New Method for Constructing Classifier Ensembles, International Journal of Digital Content: Technology and its Application, JDCTA, ISSN: 1975-9339, 2009 (in press).

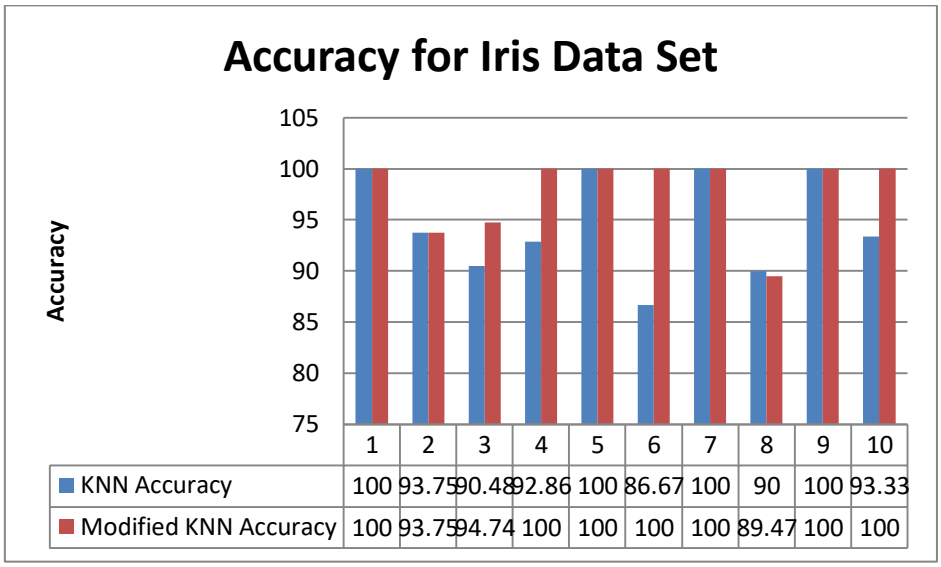
[8] H. Parvin, H. Alizadeh and B. Minaei-Bidgoli,Using Clustering for Generating Diversity in Classifier Ensemble, Internation-

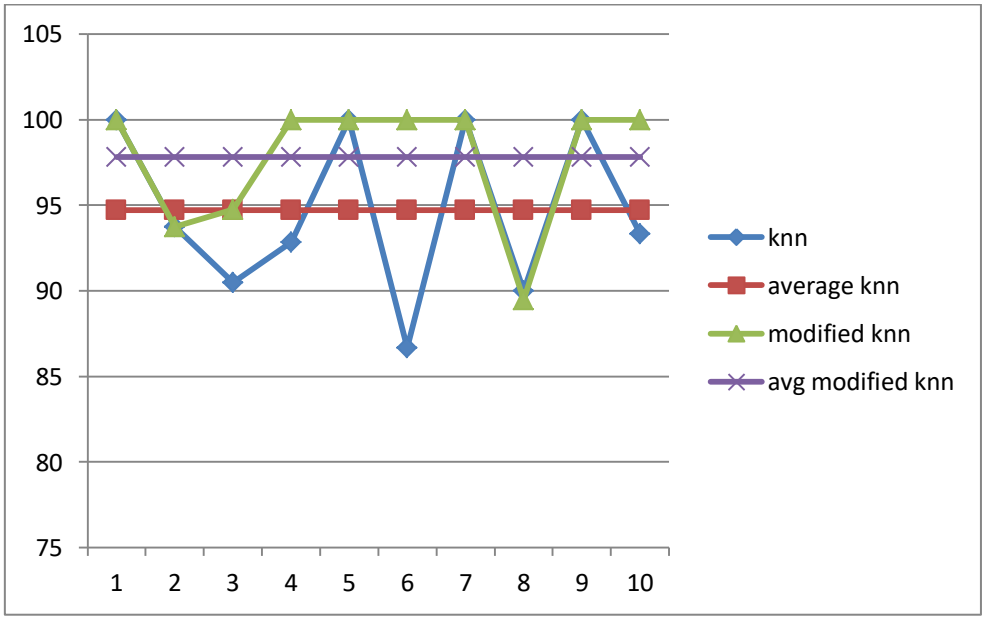
Journal of Digital Content: Technology and its ApplicationJDCTA, ISSN: 1975-9339, 2009 (in press).
[9] B.V. Darasay, Nearest Neighbor pattern classification techniques, Las Alamitos, LA: IEEE CS Press
[10] E. Fix, J.L. Hodges, Discriminatory analysis, nonparametric discrimination: Consistency properties. Technical Report 4, USAF School of Aviation Medicine, Randolph Field, Texas, 1951
[11] Cover, T.M., Hart, P.E. Nearest neighbor pattern classification. IEEE Trans. Inform. Theory, IT-13(1):21–27, 1967.
[12] Hellman, M.E. The nearest neighbor classification rule with a reject option. IEEE Trans. Syst. Man Cybern., 3:179–185, 1970.
[13] K. Fukunaga,, L. Hostetler, k-nearest-neighbor bayes-risk estimation. IEEE Trans. Information Theory, 21(3), 285-293, 1975.
[14] S.A. Dudani, The distance-weighted k-nearestneighbor rule. IEEE Trans. Syst. Man Cybern. SMC-6:325–327, 1976.
[15] T. Bailey, A. Jain, A note on distance-weighted knearest neighbor rules. IEEE Trans. Systems, Man, Cybernetics, Vol. 8, pp. 311-313, 1978.
[16] S. Bermejo, J. Cabestany, Adaptive soft k-nearestneighbour classifiers. Pattern Recognition, Vol. 33, pp. 1999-2005, 2000.
[17] Jozwik, A learning scheme for a fuzzy k-nn rule. Pattern Recognition Letters, 1:287–289, 1983.
[18] J.M. Keller, M.R. Gray, J.A. Givens, A fuzzy k-nn neighbor algorithm. IEEE Trans. Syst. Man Cybern., SMC-15(4):580–585, 1985.
[19] K. ITQON, Shunichi and I. Satoru, Improving Performance of k-Nearest Neighbor Classifier by Test Features, Springer Transactions of the Institute of Electronics, Information and Communication Engineers 2001.
[20] R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification , John Wiley & Sons, 2000.
[21] Hamid Parvin,Hoseinali Alizadeh,Behrouz Minati A Modification on K-Nearest Neighbor Classifier *GJCST Vol.10 Issue 14 (Ver.1.0) November 2010* UCI data set used:
[22] 1.default of credit card clients Data Set http://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients
2.Soybean Data Set http://archive.ics.uci.edu/ml/datasets/Soybean+%28Large%29
3.Teaching Assistant Evaluation Data Set http://archive.ics.uci.edu/ml/datasets/Teaching+Assistant+Evaluation
4.Iris Data Set http://archive.ics.uci.edu/ml/datasets/Iris

APPENDIX-A

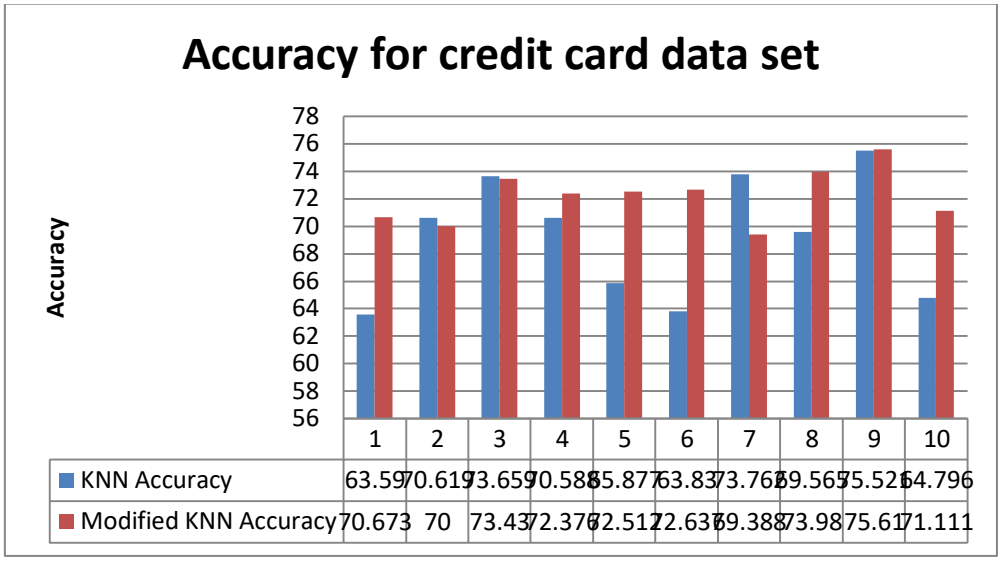
Graph showing results for different algorithms for

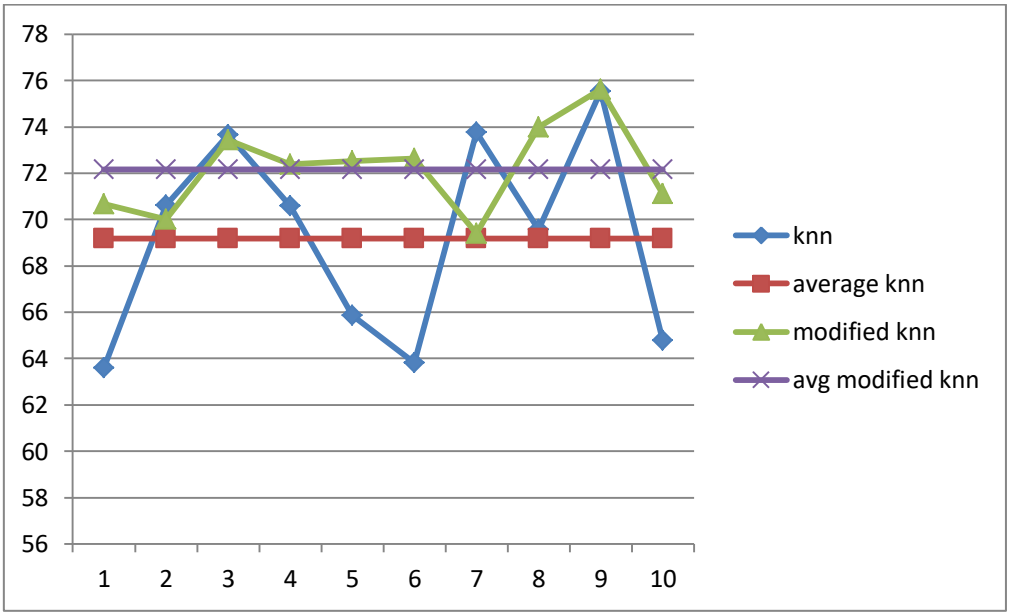
1.Iris Data Set



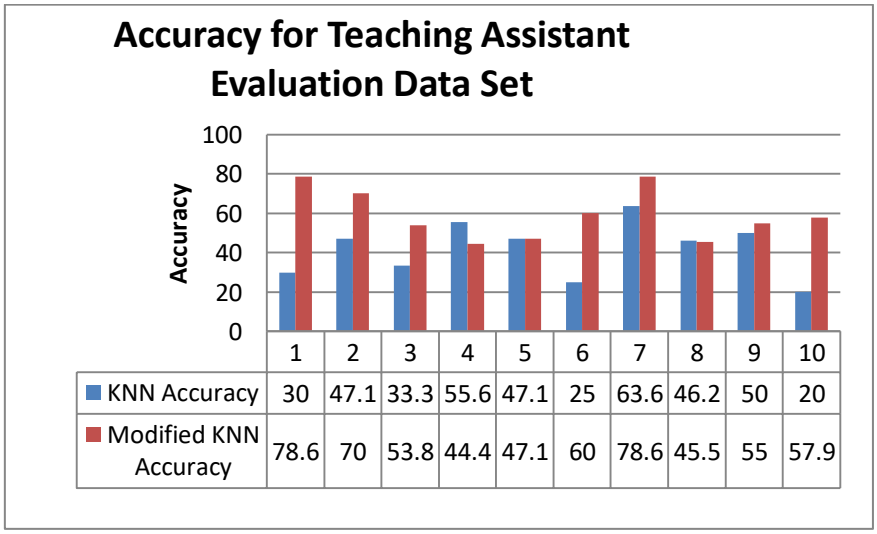


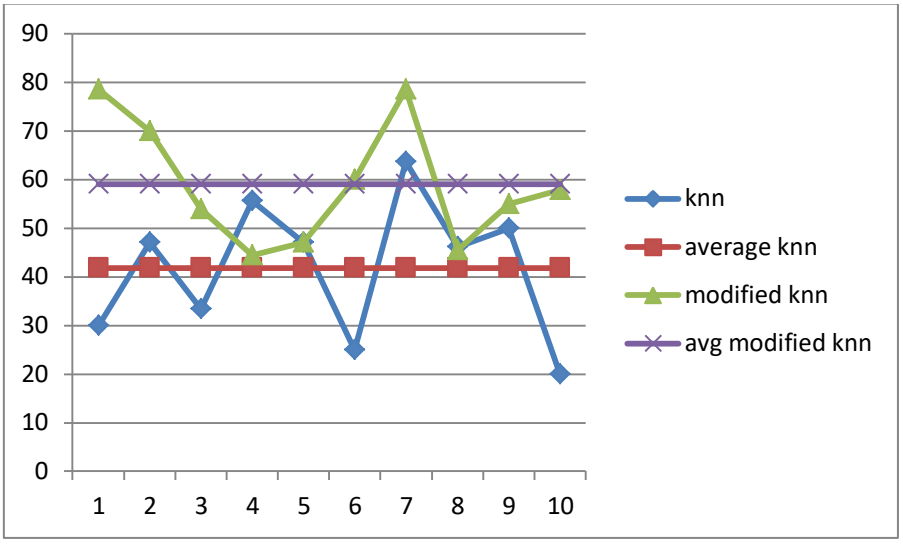
2.Credit *Data Set*



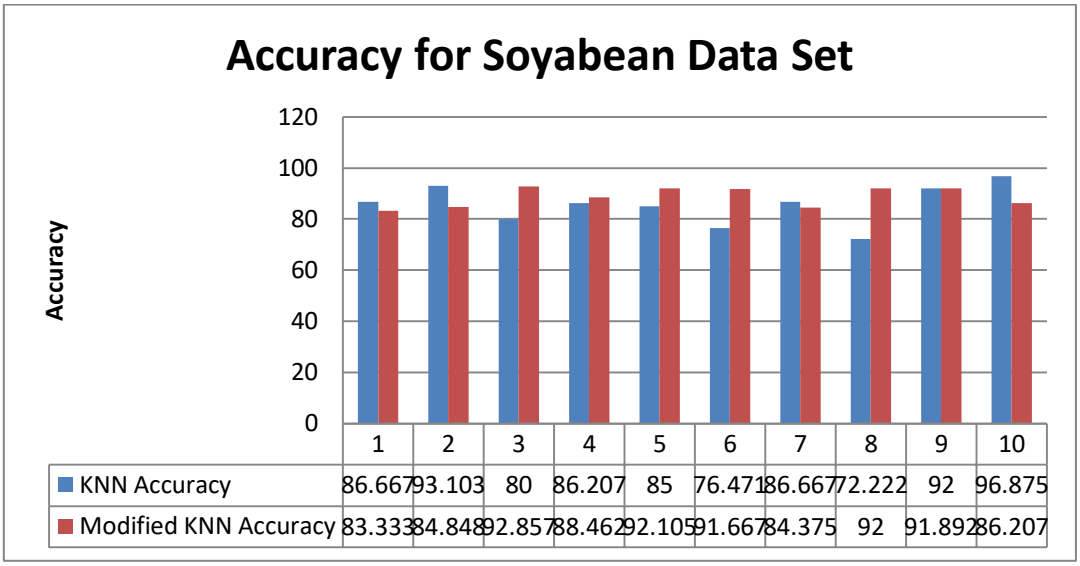


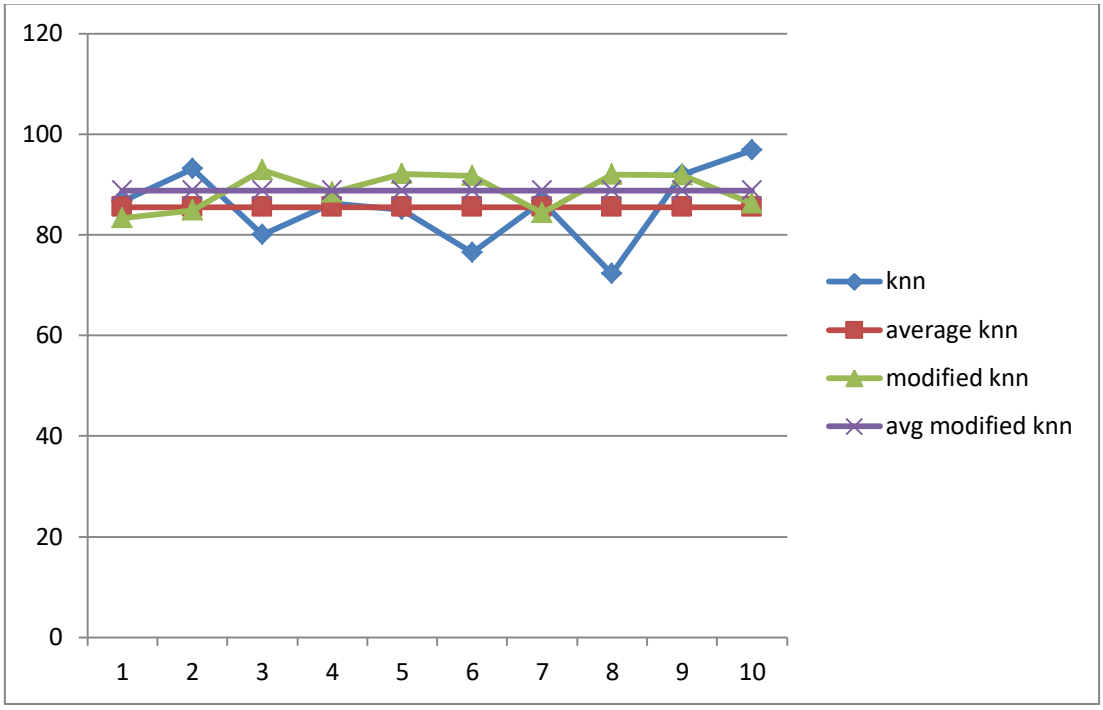
3.Teaching Assistant Data Set





4.Soyabean Data Set




120
100
80
60
40
20
0
1
2
3
4
5
6
7
8
9
10
knn
average knn
modified knn
avg modified knn