IMPROVING THE PERFORMANCE OF K-NEAREST NEIGHBOR ALGORITHM FOR THE CLASSIFICATION OF DIABETES DATASET WITH MISSING VALUES

Y. Angeline Christobel\textsuperscript{1}, P. Sivaprakasam\textsuperscript{2},
\textsuperscript{1}Research Scholar, Department of Computer Science, Karpagam University, Coimbatore, India
\textsuperscript{2}Professor, Department of Computer Science, Sri Vasavi College, Erode, India

ABSTRACT
In today’s world, people get affected by many diseases which cannot be completely cured. Diabetes is one such disease and is now a big growing health problem. It leads to the risk of heart attack, kidney failure and renal disease. The techniques of data mining have been widely applied to extract knowledge from medical databases. In this paper, we evaluated the performance of k-nearest neighbor(kNN) algorithm for classification of Diabetes data. We considered the data imputation, scaling and normalization techniques to improve the accuracy of the classifier while using diabetes data, which may contain lot of missing values. We selected to explore KNN because, it is very simple and faster than most of the complex classification algorithms. To measure the performance, we used Accuracy and Error rate as the metrics. We found that data imputation method will not lead to higher accuracy; instead it will give correct accuracy for missing values and imputation along with a suitable data preprocessing method increases the accuracy.

Keywords: Data Mining, Classification, kNN, Data Imputation Methods, Data Normalization and Scaling

1. INTRODUCTION
Hospitals and medical institutions are finding difficult to extract useful information for decision support due to the increase of medical data. So the methods for efficient computer based analysis are essential. It has been proven that the benefits of introducing machine learning into medical analysis are to increase diagnostic accuracy, to reduce costs and to reduce human resources. Earlier detection of disease will aid in increased exposure to required patient care and improved cure rates [17]. Diabetes is one of the most deadly diseases observed in many of the nations. According to the World Health Organization (WHO) [18] there are approximately millions of
people in this world are suffering from diabetes. Data classification is an important problem in engineering and scientific disciplines such as biology, psychology, medicines, marketing, computer vision, and artificial intelligence. The goal of the data classification is to classify objects into a number of categories or classes.

A lot of research work has been done on Pima Indian diabetes dataset. In [8], Jayalakshmi and Santhakumaran used the ANN method for diagnosing diabetes, using the Pima Indian diabetes dataset without missing data and obtained 68.56% classification accuracy.

Manaswini Pradhan et al.[12] suggested an Artificial Neural Network (ANN) based classification model for classifying diabetic patients. Best accuracy being 72% with average accuracy of 72.2%.

Umar Sathic Ali, Dr.C.Jothi Ventakeswaran[13] proposed an improved evidence theoretic KNN algorithm which combines Dempster Shafer theory of evidence and K nearest neighbouring rule with distance metric based neighborhood with the average accuracy of 73.57%.

Pengyi Yang, LiangXu, Bing B Zhou, Zili Zhang, and Albert Y Zomaya [14] proposed a particle swarm based hybrid system for remediying the class imbalance problem in medical and biological data mining. To guide the sampling process, multiple classification algorithms are used. The classification algorithms used in the hybrid system composition includes Decision Tree (J48), k-Nearest Neighbor (kNN), Naive Bayes (NB), Random Forest (RF) and Logistic Regression (LOG). Four medical datasets such as blood, survival, breast cancer and Pima Indian Diabetes from UCI Machine Learning Repository and a genome wide association study (GWAS) dataset from the genotyping of Single Nucleotide Polymorphisms (SNPs) of Age-related Macular Degeneration (AMD) are used. The different methods of the study includes PSO (Particle Swarm based Hybrid System), RO(Random OverSampling), RU (Random UnderSampling) and Clustering. The metrics used are AU(Area under ROC curve), F-measure and Geometric mean. Particle Swarm based Hybrid System (PSO) gives an average accuracy of 71.6%, RU 67.3%, RO 65.8% and cluster 64.9% with KNN Classifier for Pima Indian Diabetes. Also with other classifiers and dataset, the study shows that PSO gives better accuracy.

In [15], Biswadip Ghosh applies FCP (Fuzzy Composite Programming) to build a diabetes classifier using the PIMA diabetes dataset. He evaluated the performance of the Fuzzy classifier using Receiver Operating Characteristic (ROC) Curves and compared the performance of the Fuzzy classifier against a Logistic Regression classifier. The results show that FCP classifier is better when the calculated AUC values are compared. The classifier based on logistic regression has an AUC of 64.8%, while the classifier based on FCP has an AUC of 72.2%. He proved that the performance of the fuzzy classifier was found to be better than a logistic regression classifier. Quinlan applied C4.5 and it was 71.1% accurate (Quinlan 1993).

In [3], Wahba's group at the University of Wisconsin applied penalized log likelihood smoothing spline analysis of variance (PSA). They eliminated patients with glucose and BMIs of zero leaving n=752. They used 500 for the training set, and the remaining 252 as the evaluation set which showed an accuracy of 72% for the PSA model and 74% for a GLIM model (Wahba, Gu et al. 1992).

Michie et al. used 22 algorithms with 10-fold cross validation reported the following accuracy rates on the test set and:  Discrimum 77.5%, Quadisc 73.8%, Logdisc 77.7%, SMART 76.8%, ALLOC80 69.9%, k-NN 67.6%, CASTLE 74.2%, CART 74.5%, IndCART 72.9%, NewID 71.1%, AC2 72.4%, Baytree 72.9%, NaiveBay 73.8%, CN2 71.1%, C4.5 73%, Itrule 75.5%, Cal5 75%, Kohonen 72.7%, DIPOL92 77.6%, Backprop 75.2%, RBF 75.7%, and LVQ 72.8% (Michie, Spiegelhalter et al. 1994, p. 158)
J. Oates (1994) used Multi-Stream Dependency Detection (MSDD) algorithm on two-thirds of the dataset for training. Accuracy on the one-third of evaluation was 71.33%.

Although the cited articles use different techniques, the average accuracy is 70.64% for Pima Indian Diabetes dataset.

The representation of lot of missing values by zeros itself forms a characteristic and hence affects the accuracy of the classifier. The main objective of this paper is to explore the ways to improve the classification accuracy of a classification algorithm using suitable techniques such as data imputation, scaling and normalization for the classification of diabetes data which may contain lot of missing values.

2. DATA PREPROCESSING

Data in the real world is incomplete, noisy, inconsistent and redundant. The knowledge discovery during the training phase is difficult if there is much irrelevant and redundant information. Data preprocessing is important to produce quality mining results. The main tasks in data preprocessing are data cleaning, data integration, data transformation, data reduction and data discretization. The final dataset used will be the product of data preprocessing.

2.1. Missing Values and Imputation of Missing Values

Many existing, industrial and research datasets contain missing values. They are introduced due to reasons, such as manual data entry procedures, equipment errors and incorrect measurements. The most important task in data cleaning is processing missing values. Many methods have been proposed in the literature to treat missing data [21,22]. Missing data should be carefully handled; otherwise bias might be introduced into the knowledge induced and affect the quality of the supervised learning process or the performance of classification algorithms [3, 4]. Missing values imputation is a challenging issue in machine learning and data mining [2]. Due to the difficulty with missing values, most of the learning algorithms are not well adapted to some application domains (for example Web Applications) because the existed algorithms are designed with the assumption that there are no missing values in datasets. This shows the necessity for dealing with missing values. The satisfactory solution to missing data is good database design but good analysis can help to reduce the problems [1]. Many techniques can be used to deal missing data. Depending on the problem domain and the goal for the data mining process, the right technique should be selected.

In literature, the different approaches to handle missing values in dataset are given as:
1. Ignore the tuple
2. Fill in the missing values manually
3. Use a global constant to fill in the missing values
4. Use attribute mean to fill in the missing values
5. Use the attribute mean for all samples belonging to the same class as the given tuple

2.2. Mean substitution

The popular imputation method to fill in the missing data values is to use a variable’s mean or median. This method is suitable only if the data are MCAR(Missing Completely At Random). This method creates a spiked distribution at the mean in frequency distributions. It lowers the correlations between the imputed variables and the other variables and underestimates variance.
The Simple Data Scaling Algorithm
The following algorithm explains the data scaling method.
Let
\[ D = \{ A_1, A_2, A_3, \ldots, A_n \} \]
Where
\( D \) is the set of unnormalized data
\( A_i \) – is the \( i^{th} \) attribute column of values of
\( m \) - is the member of rows (records)
n - is the number of attributes.

Function Normalize(D)
Begin
For i=1 to n {
Max_i ← max(A_i)
Min_i ← min(A_i)
For r =1 to m {
A_{ir} ← A_{ir} - Min_i
A_{ir} ← A_{ir} / Max_i
Where
\( A_{ir} \) is the element of \( A_i \) at row \( r \)
}
}
Finally we will have the scaled data set.
End

The Mean Substitution Algorithm
The following algorithm explains the very commonly used form of mean substitution method [9]
Let
\[ D = \{ A_1, A_2, A_3, \ldots, A_n \} \]
Where
\( D \) is the set of data with missing values
\( A_i \) – is the \( i^{th} \) attribute column of values of \( D \) with missing values in some or all columns
n - is the number of attributes.

Function MeanSubstitution(D)
Begin
For i=1 to n {
a_i ← A_i \cap m_i
where
\( a_i \) is the column of attributes without missing values
\( m_i \) is the set of missing values in \( A_i \) (missing values denoted by a symbol)
Let \( \mu_i \) be the mean of \( a_i \)
Replace all the missing elements of \( A_i \) with \( \mu_i \)
}
Finally we will have the imputed data set.
End
2.3. k-Nearest Neighbor (knn) Classification Algorithm

KNN classification classifies instances based on their similarity. It is one of the most popular algorithms for pattern recognition. It is a type of Lazy learning where the function is only approximated locally and all computation is deferred until classification. An object is classified by a majority of its neighbors. K is always a positive integer. The neighbors are selected from a set of objects for which the correct classification is known.

The kNN algorithm is as follows:

1. Determine k i.e., the number of nearest neighbors
2. Using the distance measure, calculate the distance between the query instance and all the training samples.
3. The distance of all the training samples are sorted and nearest neighbor based on the k minimum distance is determined.
4. Since the kNN is supervised learning, get all the categories of the training data for the sorted value which fall under k.
5. The prediction value is measured by using the majority of nearest neighbors.

The Pseudo Code of kNN Algorithm

Function kNN(train_patterns, train_targets, test_patterns)

   Uc   - a set of unique labels of train_targets;
   N    - size of test_patterns
   for i = 1:N,
      dist:=EuclideanDistance(train_patterns, test_patterns(i))
      idxs := sort(dist)
      topkClasses := train_targets(idxs(1:Knn))
      c := DominatingClass (topkClasses)
      test_targets(i)  :=  c
   end

2.4. Validating the Performance of the Classification Algorithm

Classifier performance depends on the characteristics of the data to be classified. Performance of the selected algorithm is measured for Accuracy and Error rate. Various empirical tests can be performed to compare the classifier like holdout, random sub-sampling, k-fold cross validation and bootstrap method. In this study, we have selected k-fold cross validation for evaluating the classifiers. In k-fold cross validation, the initial data are randomly partitioned into k mutually exclusive subset or folds d1,d2,…,dk, each approximately equal in size. The training and testing is performed k times. In the first iteration, subsets d2, …, dk collectively serve as the training set in order to obtain a first model, which is tested on d1; the second iteration is trained in subsets d1, d3,…, dk and tested on d2; and so no[19]. The accuracy of the classifier refers to the ability of a given classifier to correctly predict the class label of new or previously unseen data [19].

The Accuracy and Error rate can be defined as follows:

Accuracy = (TP+TN) / (TP + FP + TN + FN)
Error rate  = (FP+FN) / (TP + FP + TN + FN)
Where
TP is the number of True Positives
TN is the number of True Negatives
FP is the number of False Positives
FN is the number of False Negatives

3. EXPERIMENTAL RESULTS

3.1. Pima Indians Diabetes Dataset

The Pima (or Akimel O'odham) are a group of American Indians living in southern Arizona. The name, "Akimel O'odham", means "river people". The short name, "Pima" is believed to have come from the phrase pi 'añi mac or pi mac, meaning "I don't know," used repeatedly in their initial meeting with Europeans [25]. According to World Health Organization (WHO) [18], a population of women who were at least 21 years old of Pima Indian Heritage was tested for diabetes. The data were collected by the US National Institute of Diabetes and Digestive and Kidney Diseases.

Number of Instances: 768
Number of Attributes: 8 (Attributes) plus 1 (class label)
All the attributes are numeric-valued

<table>
<thead>
<tr>
<th>Sl No</th>
<th>Attribute</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Pregnant</td>
<td>Number of times pregnant</td>
</tr>
<tr>
<td>2</td>
<td>Glucose</td>
<td>Plasma glucose concentration (glucose tolerance test)</td>
</tr>
<tr>
<td>3</td>
<td>Pressure</td>
<td>Diastolic blood pressure (mm Hg)</td>
</tr>
<tr>
<td>4</td>
<td>Triceps</td>
<td>Triceps skin fold thickness (mm)</td>
</tr>
<tr>
<td>5</td>
<td>Insulin</td>
<td>2-Hour serum insulin (mu U/ml)</td>
</tr>
<tr>
<td>6</td>
<td>Mass</td>
<td>Body mass index (weight in kg/(height in m)^2)</td>
</tr>
<tr>
<td>7</td>
<td>Pedigree</td>
<td>Diabetes pedigree function</td>
</tr>
<tr>
<td>8</td>
<td>Age</td>
<td>Age (years)</td>
</tr>
<tr>
<td>9</td>
<td>Diabetes</td>
<td>Class variable (test for diabetes)</td>
</tr>
</tbody>
</table>

Class Distribution: Class value 1 is interpreted as "tested positive for diabetes"
Class Value : 0 - Number of instances - 500
Class Value : 1 - Number of instances – 268

While the UCI repository index claims that there are no missing values, closer inspection of the data shows several physical impossibilities, e.g., blood pressure or body mass index of 0. They should be treated as missing values to achieve better classification accuracy.

The following 3D plot clearly shows the complex distribution of the positive and negative records in the original Pima Indians Diabetes Database which will complicate the classification task.
3.2. Results of the 10 Fold Validation
The following table shows the results in terms of accuracy, error and run time in the case of 10 fold validation.

<table>
<thead>
<tr>
<th>Performance with different Metrics</th>
<th>Actual Data</th>
<th>After Imputation</th>
<th>After Imputation and Scaling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>71.71</td>
<td>71.32</td>
<td>71.84</td>
</tr>
<tr>
<td>Error</td>
<td>28.29</td>
<td>28.68</td>
<td>28.16</td>
</tr>
<tr>
<td>Time</td>
<td>0.2</td>
<td>0.28</td>
<td>0.23</td>
</tr>
</tbody>
</table>

As shown in the following graphs, the performance was considerably improved while using imputed and scaled data. The performance after imputation is almost equal or little bit lower than using normal data. The reason is, the original data contains lot of missing values and the missing values represented by 0 are treated as a significant feature by the classifier. Even though the performance of original data is higher than that of imputed data, we should not consider because it will not be accurate. If we train the system with original data which has lot of missing values and use it for classifying real world data which may not contain any missing value, then the system will not classify the ideal data correctly.
The accuracy of classification has been significantly improved after imputation and scaling.

The classification error has been significantly reduced after imputation and scaling.

### 3.3. Results of the Average of 2 to 10 Fold Validation

The following table shows the average of results in terms of accuracy, error and run time in the case of 2 to 10 fold validation

<table>
<thead>
<tr>
<th>Performance with different Metrics</th>
<th>Actual Data</th>
<th>After Imputation</th>
<th>After Imputation and Scaling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>71.94</td>
<td>71.26</td>
<td>73.38</td>
</tr>
<tr>
<td>Error</td>
<td>28.06</td>
<td>28.74</td>
<td>26.62</td>
</tr>
<tr>
<td>Time</td>
<td>0.19</td>
<td>0.19</td>
<td>0.19</td>
</tr>
</tbody>
</table>
The accuracy of classification has been significantly improved after imputation and scaling. The average of 2 to 10 fold validations clearly shows the difference in accuracy.

The classification error has been significantly reduced after imputation and scaling. The average of 2 to 10 fold validations clearly shows the difference in classification Error.

The graphs above show that imputation alone does not give higher accuracy but imputation with a data preprocessing technique can improve the accuracy of a dataset which has missing values.

We also compared our improved KNN with Weka’s Implementation of KNN. The results show that improved kNN’s accuracy with imputation and scaling method is 3.2% higher than the standard Weka’s Implementation of kNN.

4. CONCLUSION AND SCOPE FOR FURTHER ENHANCEMENTS
The result shows that the characteristics and distribution of data and the quality of input data had a significant impact on the performance classifier. Performance of the kNN classifier is measured in terms of Accuracy, Error Rate and run time using 10-fold validation method and average of 2-10 fold validation. The application of imputation method and data normalization had obvious
impact on classification performance and significantly improved the performance of kNN. The performance after imputation is almost equal or little bit lower than using original data. The reason is, the original data contains lot of missing values and the missing values represented by 0 are treated as a significant feature by the classifier. So the data imputation method not always lead to so called “good results” they only give “correct” results. It means imputation will not always lead to higher accuracy for imputed missing values. But Imputation along with a suitable data preprocessing method increases the accuracy. Our improved kNN’s accuracy with imputation and scaling method is 3.2% higher than the standard Weka's Implementation of kNN. To improve the overall accuracy, we have to improve the performance of the classifier or use a good feature selection method. Future works may address hybrid classification model using kNN with other techniques. Even though the achieved accuracy of kNN based method is little bit lower than some of the previous methods, still there are scope for improving the kNN based method. For example, we may use another distance metric function instead of the standard Euclidean distance function in the distance calculation part of kNN for improving accuracy. Even we may use any evolutionary computing technique such as GA or PSO for feature selection or Feature weight selection for attaining maximum possible accuracy of classification with kNN. So, future work may address the ways to incorporate these ideas along with the standard kNN.

**ANNEXURE – RESULTS OF k-FOLD VALIDATION**

The following table shows the k-fold validation results where k is changed from 2 to 10. We used the average of this results as well as the results corresponding to k=10 for preparing graphs in the previous section IV.

**Table 3. Results With Actual Pima Indian Diabetes Dataset**

<table>
<thead>
<tr>
<th>k</th>
<th>Accuracy</th>
<th>Error</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>72.53</td>
<td>27.47</td>
<td>0.13</td>
</tr>
<tr>
<td>3</td>
<td>74.35</td>
<td>25.65</td>
<td>0.20</td>
</tr>
<tr>
<td>4</td>
<td>71.61</td>
<td>28.39</td>
<td>0.16</td>
</tr>
<tr>
<td>5</td>
<td>70.98</td>
<td>29.02</td>
<td>0.17</td>
</tr>
<tr>
<td>6</td>
<td>72.40</td>
<td>27.60</td>
<td>0.19</td>
</tr>
<tr>
<td>7</td>
<td>70.90</td>
<td>29.10</td>
<td>0.25</td>
</tr>
<tr>
<td>8</td>
<td>72.14</td>
<td>27.86</td>
<td>0.20</td>
</tr>
<tr>
<td>9</td>
<td>70.85</td>
<td>29.15</td>
<td>0.19</td>
</tr>
<tr>
<td>10</td>
<td>71.71</td>
<td>28.29</td>
<td>0.20</td>
</tr>
<tr>
<td>avg</td>
<td>71.94</td>
<td>28.06</td>
<td>0.19</td>
</tr>
</tbody>
</table>
Table 4. Results With Imputed Data

<table>
<thead>
<tr>
<th>k</th>
<th>Accuracy</th>
<th>Error</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>71.22</td>
<td>28.78</td>
<td>0.16</td>
</tr>
<tr>
<td>3</td>
<td>70.44</td>
<td>29.56</td>
<td>0.19</td>
</tr>
<tr>
<td>4</td>
<td>70.83</td>
<td>29.17</td>
<td>0.16</td>
</tr>
<tr>
<td>5</td>
<td>71.24</td>
<td>28.76</td>
<td>0.14</td>
</tr>
<tr>
<td>6</td>
<td>71.88</td>
<td>28.13</td>
<td>0.19</td>
</tr>
<tr>
<td>7</td>
<td>71.56</td>
<td>28.44</td>
<td>0.20</td>
</tr>
<tr>
<td>8</td>
<td>71.61</td>
<td>28.39</td>
<td>0.17</td>
</tr>
<tr>
<td>9</td>
<td>71.24</td>
<td>28.76</td>
<td>0.19</td>
</tr>
<tr>
<td>10</td>
<td>71.32</td>
<td>28.68</td>
<td>0.28</td>
</tr>
<tr>
<td>Avg</td>
<td>71.26</td>
<td>28.74</td>
<td>0.19</td>
</tr>
</tbody>
</table>

Table 5. Results With Imputed and Scaled/Normalized Data

<table>
<thead>
<tr>
<th>k</th>
<th>Accuracy</th>
<th>Error</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>74.74</td>
<td>25.26</td>
<td>0.22</td>
</tr>
<tr>
<td>3</td>
<td>74.09</td>
<td>25.91</td>
<td>0.13</td>
</tr>
<tr>
<td>4</td>
<td>71.88</td>
<td>28.13</td>
<td>0.16</td>
</tr>
<tr>
<td>5</td>
<td>73.59</td>
<td>26.41</td>
<td>0.17</td>
</tr>
<tr>
<td>6</td>
<td>72.92</td>
<td>27.08</td>
<td>0.17</td>
</tr>
<tr>
<td>7</td>
<td>74.31</td>
<td>25.69</td>
<td>0.16</td>
</tr>
<tr>
<td>8</td>
<td>74.09</td>
<td>25.91</td>
<td>0.22</td>
</tr>
<tr>
<td>9</td>
<td>72.94</td>
<td>27.06</td>
<td>0.22</td>
</tr>
<tr>
<td>10</td>
<td>71.84</td>
<td>28.16</td>
<td>0.23</td>
</tr>
<tr>
<td>Avg</td>
<td>73.38</td>
<td>26.62</td>
<td>0.19</td>
</tr>
</tbody>
</table>

5. REFERENCES


in IEEE EMBS Conference on Biomedical Engineering & Sciences, 2010.


[17] Roshawnna Scales, Mark Embrechts, “Computational intelligence techniques for medical diagnostics”.


[34] UCI Machine Learning Repository http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

[35] Xindong Wu · Vipin Kumar · J. Ross Quinlan · Joydeep Ghosh · Qiang Yang · Hiroshi Motoda · Geoffrey J. McLachlan · Angus Ng · Bing Liu · Philip S. Yu · Zhi-Hua Zhou · Michael Steinbach · David J. Hand · Dan Steinberg, “Top 10 algorithms in data mining” Springer 2007


[37] Thair Nu Phyu, “Survey of Classification Techniques in Data Mining MultiConference of Engineers and Computer Scientists”, 2009 Vol I IMECS 2009, Hong Kong

[38] Arbach, L.; Reinhardt, J.M.; Bennett, D.L.; Fallowh, G.; Iowa Univ., Iowa City, IA, USA “Mammographic masses classification: comparison between backpropagation neural network (BNN), K nearest neighbors (KNN), and human readers”, 2003 IEEE CCECE