IOSR Journal of Computer Engineering (IOSRJCE)
ISSN: 2278-0661, ISBN: 2278-8727Volume 7, Issue 4 (Nov. - Dec. 2012), PP 16-23
www.iosrjournals.org

The Negative Impact of Missing Value Imputation in Classification of Diabetes Dataset and Solution for Improvement

Angeline Christobel. Y[1], Dr.P.SivaPrakasam[2]
[1]*(Research Scholar, Department of Computer Science, Karpagam University, Coimbatore, India)*
[2]*(Professor, Department of Computer Science, Sri Vasavi College, Erode, India)*

Abstract : *The common problem for data quality is missing data. The real datasets have lot of missing values. Missing values imputation is a challenging issue in machine learning and data mining. Missing data should be carefully handled; otherwise it affects the quality of the mining process or the performance of classification algorithms. Mean method of imputation is the most common method to replace the missing values. In this paper, we address the negative impact of missing value imputation and solution for improvement while evaluating the performance of kNN algorithm for classification of Diabetes data. We selected diabetes dataset because it contains lot of missing values and the impact of imputation is very obvious. To measure the performance, we used Accuracy and Error rate as the metrics.*

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

I. Introduction

The common difficulty encountered in many real world situations is missing data. Missing data creates difficulties with data analysis, study and visualization [3]. Missing data is categorized into three types by the statisticians [5].

Missing completely at random (MCAR).
It is the highest level of randomness. The probability of missing data on any attribute does not depend on any value of attribute.

Missing at random (MAR).
The probability of missing data on any attribute does not depend on its own particular value, but on the values of other attributes.

Not missing at random (NMAR)
Missing data depends on the values that are missing.

In KDD process, the treatment of missing values is an important task. If the dataset contains large amount of missing values, the treatment of missing data can improve the quality of mining process. There are loads of imputation methods (Little and Rubin 1987) like Mean Imputation, regression imputation, Expectation maximization etc are available. Imputation of missing data minimizes bias and allows for analysis using a reduced dataset. In [21] the different approaches to handle missing values in dataset are given as:

- Ignore the tuple
- Fill in the missing values manually
- Use a global constant to fill in the missing values
- Use attribute mean to fill in the missing values
- Use the attribute mean for all samples belonging to the same class as the given tuple

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

In general, the imputation methods can be classified into single and multiple imputations. The single imputation method always imputes the same value, thereby ignoring the variance associated with the imputation process. The multiple imputation method imputes several imputed values and the effect of the chosen imputed values on the variance can be taken into account.

A lot of research work has been done on missing value imputation. In [6], the authors show the effect of missing data imputation using five single imputation methods and one multiple imputation method on classification accuracy for six popular classifiers (RIPPER, C4.5, K-nearest-neighbor, support vector machine with polynomial and RBF kernels, and Nai¨ve-Bayes) on 15 datasets. The empirical study shows that imputation with the tested methods on average improves classification accuracy when compared to classification without imputation. In [7], the authors propose a class mean imputation (CMI) method based on the k-NN hot deck imputation method (MINI) to impute both continuous and nominal missing data in small data sets. Bhekisipho Twala, Michelle Cartwright [9] proposed BAMINNSI that constructs ensembles based on two imputation methods (Bayesian multiple imputation and nearest neighbour single imputation).In [10], the authors studied the impact of different missing value imputation methods on the classification accuracy. They implemented three popular imputation methods such as Singular Value Decomposition (SVD), weighted K-nearest neighbors (KNNimpute), and Zero replacement. After applying the three imputation methods, they found slight variations in the classifier accuracy. Twala, Jones, and Hand [11] proposed a method for creating a separate class for missing values, and found that its performance was competitive with that of likelihood-based multiple imputation. Kim and Yates [12] did a simulation study of seven popular missing value methods but did not find any dominant method.

The objective of this paper is to address the negative impact of missing value imputation and solution for improvement while evaluating the performance of kNN algorithm for classification of Diabetes data which may contain lot of missing values. We found that data imputation method will not lead to higher accuracy and imputation along with a suitable data preprocessing method can increase the accuracy.

II. DATA Preprocessing

Today’s real-world datasets are highly subject to noisy, missing, and inconsistent due to their larger size and their likely origin from multiple, heterogeneous sources. Low-quality data will lead to low-quality mining results [21]. So we need to preprocess the dataset before using it.

Data preprocessing techniques can be divided into following categories.

- Data cleaning
- Data integration
- Data reduction
- Data transformations

To remove noise and correct inconsistencies in data, *data cleaning* is applied.

Data integration combines data from multiple sources into a common data store such as a data warehouse.

Data reduction is applied to reduce the size by aggregating, eliminating redundant features, or clustering.

Data transformations (normalization) can be applied, where data are scaled to fall within a range 0.0 to 1.0. This will improve the accuracy and efficiency of mining algorithms involving distance measurements.

Depending on the problem domain and the goal for the data mining process, the right technique should be selected. In this paper, we used mean substitution to impute missing values and data scaling algorithms to increase the accuracy of KNN.

The Mean Substitution, Data scaling and KNN algorithms are discussed below.

1. The Mean Substitution Algorithm

The popular imputation method to fill in the missing data values is to use a variable’s mean or median.

The following algorithm explains the very commonly used form of mean substitution method [9]

```
Let
 $D = \{ A_1, A_{2,} A_{3, \ldots\ldots} A_n \}$
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 {
                $a_i \leftarrow A_i \cap m_i$
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. The Simple Data Scaling Algorithm

The following algorithm explains the data scaling method.

Let
$D = \{ A_1, A_2, A_3, \ldots\ldots A_n \}$
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 {
                $Max_i \leftarrow max(A_i)$
                $Min_i \leftarrow min(A_i)$
                For r =1 to m {
                        $A_{ir} \leftarrow A_{ir}-Min_i$
                        $A_{ir} \leftarrow A_{ir}/ Max_i$
                        Where
$A_{ir}$ is the element of $A_i$ at row r
}
        }
        Finally we will have the scaled data set.
End
```

3. K-Nearest Neighbor 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
```

4. Validating the Performance of the Classification Algorithm

Classifier performance depends on the characteristics of the data to be classified. Performance of the selected algorithms is measured for Accuracy and Error rate. 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

III. Experimental Results

Pima Indians Diabetes Database

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 [20]. According to World Health Organization (WHO) [21], 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

Sl No	Attribute	Explanation
1	pregnant	Number of times pregnant
2	glucose	Plasma glucose concentration (glucose tolerance test)
3	pressure	Diastolic blood pressure (mm Hg)
4	triceps	Triceps skin fold thickness (mm)
5	insulin	2-Hour serum insulin (mu U/ml)
6	mass	Body mass index (weight in kg/(height in m)^2)
7	pedigree	Diabetes pedigree function
8	age	Age (years)
9	diabetes	Class variable (test for diabetes)

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.

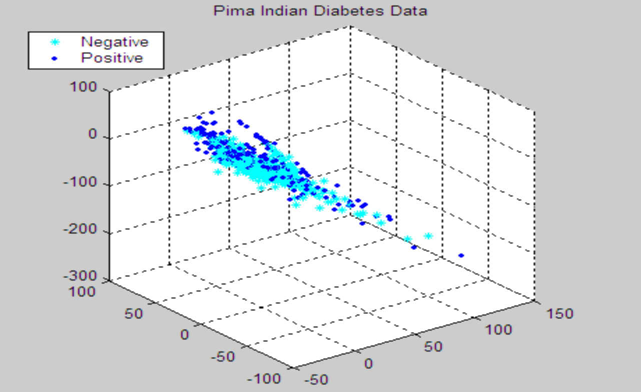

Fig 1: The 3D plot of original Pima Indians Diabetes Database

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 1. 10 Fold Validation Results

Performance with different Metrics	Actual Data	After Imputation	After Imputation and Scaling
Accuracy	71.71	71.32	71.84
Error	28.29	28.68	28.16
Time	0.2	0.28	0.23

Fig.2 shows the performance of the classifier in terms of accuracy. The performance after imputation is reduced compared with the actual data. This shows the negative impact of imputation. 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 we can say that only imputation cannot increase the accuracy of the classifier. The solution to improve the accuracy is along with imputation another data preprocessing technique can be used.. The technique used here is imputation and scaling. Fig 2 clearly shows that after imputation and scaling the performance of the classifier has been improved.

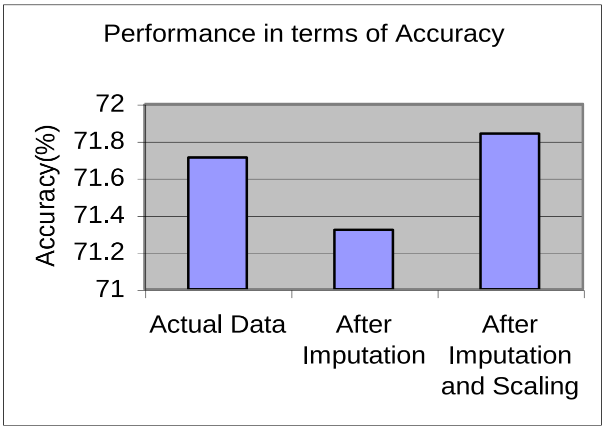

Fig 2 : Comparison of Accuracy –10 Fold Validation

The accuracy of classification has been reduced after imputation and significantly improved after imputation and scaling.

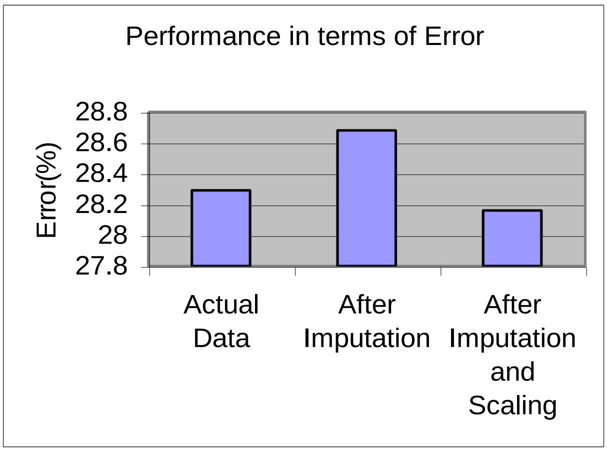

Fig 3.: Comparison of Error –10 Fold Valuation

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

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

Table 2. 2 to 10 Fold Validation Results

Performance with different Metrics	Actual Data	After Imputation	After Imputation and Scaling
Accuracy	71.94	71.26	73.38
Error	28.06	28.74	26.62
Time	0.19	0.19	0.19

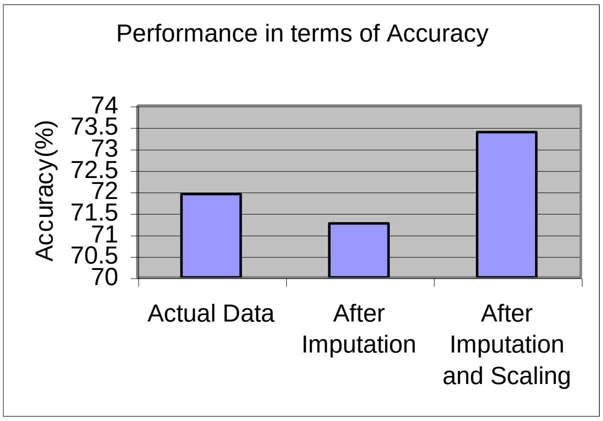

Fig 4: Comparison of Accuracy
Avg. of 2-10 Fold Validations

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

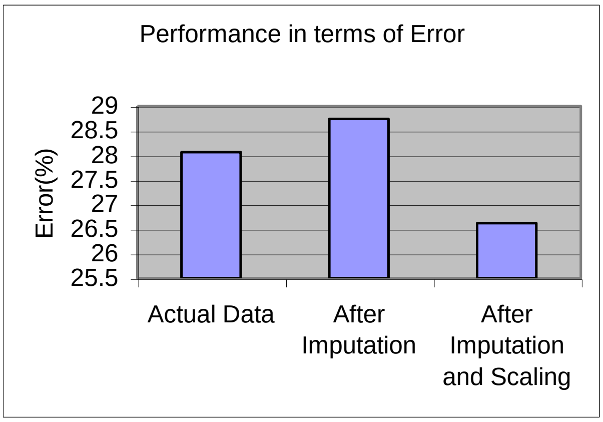

Fig 5. Comparison of Error
Avg. of 2-10 Fold Valuations

The classification error has been increased after imputation and significantly reduced after imputation and scaling. Figure 4 clearly shows the difference in classification Error.

IV. Conclusion And Scope For Further Enhancements

The result shows that the performance after imputation is lower than original data. This shows the negative impact of imputation. 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. Imputation along with a suitable data preprocessing method can improve the accuracy of the 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 data normalization had obvious impact on classification performance and significantly improved the performance 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. 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 tables 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

Table 2. Results With Actual Pima Indian Diabetes Dataset

k	Sensitivity	Specificity	Accuracy	Error	Time
2	56.45	81.30	72.53	27.47	0.13
3	57.83	83.14	74.35	25.65	0.20
4	53.96	81.32	71.61	28.39	0.16
5	51.77	81.39	70.98	29.02	0.17
6	54.91	81.69	72.40	27.60	0.19
7	51.94	81.08	70.90	29.10	0.25
8	52.51	82.97	72.14	27.86	0.20
9	50.12	81.81	70.85	29.15	0.19
10	51.48	82.63	71.71	28.29	0.20
avg	53.44	81.93	71.94	28.06	0.19

Table 3. Results With Imputed Data

k	Sensitivity	Specificity	Accuracy	Error	Time
2	55.81	79.67	71.22	28.78	0.16
3	54.60	79.07	70.44	29.56	0.19
4	53.72	80.19	70.83	29.17	0.16
5	54.95	80.05	71.24	28.76	0.14
6	55.24	80.75	71.88	28.13	0.19
7	52.34	81.78	71.56	28.44	0.20
8	53.90	81.07	71.61	28.39	0.17
9	54.55	80.24	71.24	28.76	0.19
10	52.73	81.40	71.32	28.68	0.28
Avg	54.21	80.47	71.26	28.74	0.19

Table 4. Results With Imputed and Scaled/Normalized Data

k	Sensitivity	Specificity	Accuracy	Error	Time
2	61.02	82.33	74.74	25.26	0.22
3	58.55	82.31	74.09	25.91	0.13
4	55.55	80.63	71.88	28.13	0.16
5	61.60	80.27	73.59	26.41	0.17
6	59.35	80.42	72.92	27.08	0.17
7	62.34	80.95	74.31	25.69	0.16
8	60.38	81.93	74.09	25.91	0.22
9	61.46	79.73	72.94	27.06	0.22
10	57.34	79.88	71.84	28.16	0.23
Avg	59.73	80.94	73.38	26.62	0.19

References

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

[2] World Health Organization. Available: http://www.who.int

[3] G. Ssali, T. Marwala. “Estimation of missing data using computational intelligence and decision trees.” Proceedings of IEEE International Joint Conference On Neural Networks, Hong Kong.

[4] R. J. Little and D. B. Rubin. “Statistical Analysis With Missing Data”, Hoboken, NJ: Wiley,(1987).

[5] Gustavo E. A. P. A. Batista and Maria Carolina Monard, An Analysis of Four Missing Data Treatment Methods for Supervised Learning, Applied Artificial Intelligence 17(5-6): 519-533 , 2003

[6] Alireza Farhangfar, Lukasz Kurgan, Jennifer Dy,”Impact of imputation of missing values on classification error for discrete data”, Pattern Recognition, Volume 41, Issue 12, December 2008, Pages 3692–3705

[7] Qinbo Song, Martin Shepperd.”A new imputation method for small software project data sets”, Journal of Systems and Software, Volume 80, Issue 1, January 2007, Pages 51–62

[8] José M. Jerez Ignacio Molina ,, Pedro J. García-Laencina ,Emilio Alba, Nuria Ribelles, Miguel Martín, Leonardo Franco,” Missing data imputation using statistical and machine learning methods in a real breast cancer problem”, Artificial Intelligence in Medicine, Volume 50, Issue 2, October 2010, Pages 105–115

[9] Bhekisipho Twala, Michelle Cartwright,”Ensemble Imputation Methods for Missing Software Engineering Data”, 11th IEEE International Software Metrics Symposium (METRICS'05)

[10] Vidan Fathi Ghoneim, Nahed H. Solouma, Yasser M. Kadah,” Evaluation of Missing Values Imputation Methods in cDNA Microarrays Based on Classification Accuracy”, 978-1-4244-7000-6/11 IEEE.

[11] Twala, M.C. Jones, and D.J. Hand.” Good methods for coping with missing data in decision trees”,Pattern Recognition Letters, 29:950–956, 2008.

[12] H. Kim and S. Yates. Missing value algorithms in decision trees. In H. Bozdogan, editor, Statistical Data Mining and Knowledge Discovery, pages 155–172. Chapman & Hall/CRC, Boca Raton, Fla, 2003.

[13] K.T. Chuang, K. P. Lin, and M. S. Chen. “Quality-Aware Sampling and Its Applications in Incremental Data Mining”, IEEE Transactions on knowledge and data engineering,vol.19,no. 4,2007.

[14] Li.Liu, Y. Tu, Y. Li, and G. Zou. “Imputation for missing data and variance estimation whenauxiliary information is incomplete”, Model Assisted Statistics and Applications, 83-94,2005

[15] Y Shi, Z Cai, G Lin, Classification accuracy based microarray missing value imputation. in Bioinformatics Algorithms: Techniques and Applications, ed. by Mandoiu I, Zelikovsky A (Wiley-Interscience, Hoboken, NJ, USA, 2007), pp. 303–328

[16] J Hua, T Waibhav, ER Dougherty, Performance of feature-selection methods in the classification of high-dimension data. Pattern Recognition **42**(3), 409–424 (2009).

[17] Brian A. Cattle1, Paul D. Baxter, Darren C. Greenwood, Christopher P. Gale1, Robert M. West, “Multiple imputation for completion of a national clinical audit dataset”, Statistics in Medicine Volume 30, Issue 22, pages 2736–2753, 30 September 2011

[18] Alan Olinsky, Shaw Chen, Lisa Harlow ,” The comparative efficacy of imputation methods for missing data in structural equation modeling”, European Journal of Operational Research, Volume 151, Issue 1, 16 November 2003, Pages 53–79

[19] Awawtam. "Pima Stories of the Beginning of the World." The Norton Anthology of American Literature. 7th ed. Vol. A. New York: W. W. Norton &, 2007. 22-31

[20] World Health Organization. Available: http://www.who.int

[21] Han and M. Kamber, Data Mining Concepts and Techniques, Morgan Kauffman Publishers, USA, 2006.

[22] Angeline Christobel, SivaPrakasam,”An Empirical Comparison of Data mining Classification Methods”, International Journal of Computer Information Systems, Vol. 3, No. 2, 2011

[23] UCI Machine Learning Repository [http://archive.ics.uci.edu/ml/datasets]

[24] G.E.A.P.A. Batista and M.C. Monard, “ An analysis of four Missing Data treatment methods for supervised learning”, Applied Arti¯cial Intelli-gence 17 (2003)

[25] M. Ganji and M. Abadeh, "Using fuzzy ant colony optimization fordiagnosis of diabetes disease," IEEE, 2010