Performance Analysis of XGBoost Classifier with Missing Data

Zeliha Ergul Aydin[1], and Zehra Kamisli Ozturk[1]

[1]Department of Industrial Engineering, Eskisehir Technical University, Eskisehir, Turkey

Corresponding author: Zeliha Ergul Aydin (e-mail: zergul@eskisehir.edu.tr).

Article Info

Received: December 02, 2020
Accepted: March 28, 2021
Published: May 01, 2021

Keywords

Missing Data
Missing Data Imputation
Classification
XGBoost
KEEL

ABSTRACT XGBoost algorithm has become popular due to its success in data science competitions, especially Kaggle competitions. Missingness in a dataset is a challenging problem and needs extra processing. Missing data imputation, which is filling missing data with plausible values, is one of the solutions to this problem. One of the most important reasons why researchers prefer XGBoost is that it can work with missing data without processing. With this study, we aim to show the impact of missing data imputation methods on the XGBoost classifier. We compare the performance of the XGBoost classifier trained on non-imputed data with the XGBoost classifier trained on imputed data. For comparative analysis, we choose K-nearest neighbor (KNN), Soft-Impute, multivariate imputation by chained equations (MICE), mean, and class-mean as the imputation methods, and ten datasets from KEEL repository as the datasets. We perform the Friedman test to compare the classification models' F-score statistically. Our analysis shows that the missing data imputation methods don't have any effect on XGBoost classifier performance.

I. INTRODUCTION

XGBoost also called eXtreme Gradient Boosting, is a machine learning algorithm that is becoming widespread as it won many Kaggle data science competitions. It provides satisfactory results in many applications such as disease prediction (Budholiya et al., 2020), diesel fuel brands identification (Wang et al., 2020), estimation of the tunnel boring machine advance rate (Zhou et al., 2020), prediction of concrete electrical resistivity for structural health monitoring (Dong et al., 2020), hotel reviews sentiment analysis (Zhang & Yu, 2017), star/galaxy classification (Chao et al., 2019), prediction of vehicle occupants injury at signalized intersections (Kidando et al,2020). Researchers widely prefer XGBoost because of its pros given below:

- handling missing data internally
- does not require data scaling and normalizing
- high computational speed by using parallel processing
- avoiding the overfitting problem with regularization
- high prediction accuracy

XGBoost's handling of missing data internally is one of the essential factors in the widespread use of XGBoost because missing data handling is a challenging problem and needs extra processing. There is no comprehensive study analyzing the XGBoost performance in handling missing data to the best of our knowledge. To fill this gap, we perform a comprehensive experiment to show the impact of missing data imputation methods on the XGBoost classifier with this study. We use K- nearest neighbor (KNN), Soft-Impute, multivariate imputation by chained equations (MICE), mean, and class-mean imputations methods, and ten datasets from KEEL repository as the datasets for comparison. The rest of the paper is organized as follows. In Section II, we give a brief synopsis of the related works. Section III describes the XGBoost classifier and missing data imputation methods used in this study. In Section IV, we present our experimental design, results and our discussion about the results. Finally, the conclusion and future works are given in Section V.

II. RELATED WORKS

There are many comparative missing data imputation analyzes for traditional classifiers in the literature. Most of these studies proved that missing data imputation methods increase the classifier's performance. Farhangfar et al. (2008) showed that imputation methods, except for the mean imputation, improve the performance of RIPPER, C4.5, KNN, support vector machine with polynomial and RBF kernels, and Naive-Bayes classifier, with paired t-test. Luengo et al. (2012) performed a comprehensive analysis with 23 different classification methods and 14 different imputation methods. They concluded that missing values imputation methods could improve the classification accuracy. Missing data imputatio1n gave similar results on ANN, decision tree, and random forest classifiers (Poulos & Valle, 2018). Contrary to these findings, Acuña and Rodriguez (2004) showed that the case deletion method, mean imputation, median imputation, and the KNN imputation don't have a significant effect on the accuracy of Linear Discriminant Analysis and the KNN classifier.

There are also some comparison based studies for classifiers C4.5, CN2, and XGBoost that can handle missing data internally. Batista and Monard (2003) indicated that missing data KNN imputation method could outperform the internal methods used by C4.5 and CN2 classifier. However, they didn't perform any statistical test to compare the classifiers' performance on imputed and non-imputed datasets. Rusdah and Murfi (2020) showed that the XGBoost without any imputation gives a comparable classification accuracy score to one of the XGBoost with KNN and mean imputation method for risk prediction in life insurance. They performed the analysis on only one dataset from an insurance company without any statistical test. Hence, their findings cannot be generalized. Based on the literature, it is seen that there is a need for a comprehensive analysis to mention general results.

III. METHODS

The idea behind ensemble classification is to construct multiple classifiers to obtain better classification performance. Bagging and boosting are the two forms of ensemble classification. There are three types of boosting algorithm; Adaptive Boosting (AdaBoost), Gradient Boosting, and eXtreme Gradient Boosting (XGBoost). We focus on XGBoost classifier in subsection A. Besides, data missingness is a critical issue in the classification task because most classification algorithms work with complete datasets. We briefly explain missing data imputation methods used in this paper, in subsection B.

A. XGBOOST

XGBoost presented by Chen and Guestrin (2016) is a gradient boosted decision tree model. XGBoost algorithm trains decision trees on training data sequentially. The algorithm adds a new decision tree at each iteration to the previous decision trees to improve the objective function's value. The objective function, which is aimed to minimize, consists of the loss term (*l*) and the regularization term (*Ω*). Equation 1 defines the objective function of the *t*-th iteration (L^t), where y_i is the actual class label of instance *i*, $\hat{y}_i$ is the predicted class label of instance *i*, f_k is the function of tree, *n* is the number of instances in the training set, and *Ω* is regularization term.

$$L^t = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{t-1} + f_t(x_i)\right) + \Omega\,(f_t) \tag{1}$$

$\Omega(f_t)$ in Equation 2 penalizes the model complexity to avoid overfitting, where γ and λ are the hyperparameters, *T* is the number of leaves in the tree, and *w* is the weight of each leaf.

$$\Omega(f_t) = \gamma T_t + \frac{1}{2}\lambda \sum_{j=1}^{T} {w_j}^2 \tag{2}$$

The second-order Taylor expansion is applied to approximate the value of the loss function in Equation 3, where g_i represents the first order gradient statistics on the loss in Equation 4, and h_i represents the second order gradient statistics on the loss in Equation 5.

$$L^t \cong \sum_{i=1}^{n} \left[l(y_i, \hat{y}_i^{t-1}) + g_i f_t(x_i) + \frac{1}{2} h_i {f_t}^2(x_i) \right] + \Omega(f_t) \tag{3}$$

$$g_i = \partial_{\hat{y}_i^{t-1}} l(y_i, \hat{y}_i^{t-1}) \tag{4}$$

$$h_i = \partial^2_{\hat{y}_i^{t-1}} l(y_i, \hat{y}_i^{t-1}) \tag{5}$$

For a fixed tree structure, the optimal leaf weight in leaf node *j*, and the corresponding optimal value of objective function are given by Equation 6 and 7 respectively, where I_j denotes the instance set of leaf *j*. Equation 7 is used as a scoring function to evaluate the quality of tree structure *q*.

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \tag{6}$$

$$L(q) = -\frac{1}{2} \sum_{j=1}^{T} \frac{\left(\sum_{i \in I_j} g_i\right)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \tag{7}$$

Nevertheless, all possible tree structures must be evaluated to obtain the optimal tree structure. However, it is impossible to consider all possible tree structures because of computational cost. In practice, XGBoost adopts a greedy algorithm to find an optimal tree structure.

Missing data are handled internally in XGBoost without requiring any imputing and deleting process. XGBoost classifies an instance with missing feature in to default direction at each node as shown in Figure 1. There are two options as right and left nodes for default direction. The direction with the maximum gain in training set is selected as a default direction.

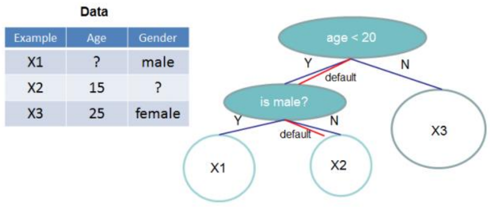

FIGURE 1. An example of how XGBoost handle missing data (Chen, & Guestrin, 2016)

B. MISSING DATA IMPUTATION METHODS

Little and Rubin (2002) classified missing data mechanisms that lead to missingness as missing completely at random (MCAR), missing at random (MAR), or not missing at random (NMAR). In a nutshell, missingness on MCAR is independent of both missing and known data. The missingness on MAR depends on known data, and missingness on NMAR depends on missing data. Missing data imputation methods fill the missing data with plausible data under the assumptions of different mechanisms. Missing data imputation, which is filling missing data with plausible values, is one of the solutions to this problem.

In this paper, we consider five commonly used imputation methods that can work with the MCAR mechanism. These are K-nearest neighbor (KNN), Soft-Impute, multivariate imputation by chained equations (MICE), mean, and class-mean. KNN imputation replaces missing data in an instance with the mean, mode, or median of the k nearest neighbors of this instance. The nearest neighbors are determined using Euclidean, Manhattan, Hamming, or Jaccard similarity measures. Mean imputation fills the missing data in a feature with the mean, median, or mode of this feature. In class mean imputation, missing data in a feature are completed with the mean, median, or the mode of this feature instance, which belongs to the same class label as the missing data. Mean and median are used for continuous features, the mode is used for discrete features in mean, KNN on and class mean imputation. MICE (Van Buuren & Groothuis-Oudshoorn, 2011) imputes missing data by chained regressions. A regression model is trained on other features to predict missing data in a feature, and these missing data replace with the predictions. This process continues iteratively with a chain until it reaches the number of multiple imputations and iteration parameters. Soft-Impute (Mazumder et al., 2010) iteratively replaces the missing data with values obtained from a soft-thresholded singular value decomposition.

IV. EXPERIMENTS AND RESULTS

A. DATASETS

Ten datasets which include missing value were taken from KEEL repository (Alcalá-Fdez et al., 2011) for comparative analysis. In the KEEL repository, Datasets are split into training and test sets with 10-folds cross-validation procedure. They are artificially introduced missing data according to MCAR mechanism in training sets. Table I summarizes the characteristic of these datasets.

TABLE I
DESCRIPTION OF DATASETS

Dataset	Features	Feature Type	Instances	Classes	Missing Percentage (%)
Iris	4	Continous	150	3	32.67
Pima	8	Continous	768	2	50.65
Wine	13	Continous	178	3	70.22
Australian	14	Mixed	690	2	70.58
Newtyroid	5	Mixed	215	3	35.35
Ecoli	7	Continous	336	8	48.21
Satimage	36	Discrete	6435	7	87.80
German	20	Mixed	1000	2	80.00
Magic	10	Continous	1902	2	58.20
Shuttle	9	Discrete	2175	7	55.95

B. CLASSIFICATION MODELS

We constructed six different XGBoost classifier models by combining different missing data imputation methods for each dataset. Table II shows the description and name of these models. Fancyimpute (fancyimpute, 2020) and Scikit-learn (Pedregosa et al., 2011) Python packages are used for the implementation of these methods.

TABLE II
DESCRIPTION OF CLASSIFICATION MODELS

Model Name	Description
XGB	XGBoost classifier trained on non-imputed data
XGB+KNN	XGBoost classifier trained on imputed data with KNN
XGB+MICE	XGBoost classifier trained on imputed data with MICE
XGB+SI	XGBoost classifier trained on imputed data with Soft-Impute
XGB+MI	XGBoost classifier trained on imputed data with mean
XGB+CMI	XGBoost classifier trained on imputed data with class-mean

C. PARAMETER SETTINGS

We set the *k* parameter as 1 in KNN, the number of multiple imputations parameters and iterations as 5, and optimized the XGBoost classifiers' hyperparameters by using grid search method with nested 10-fold cross-validation procedure.

D. EVALUATION METRICS

The macro-averaged F-score computed with 10-fold cross-validation is used to evaluate the models' performance. F-score is calculated based on the confusion matrix, which is given in Table III.

TABLE III
CONFUSION MATRIX

	Actual Positive	**Actual Negative**
Predicted Positive	TP	FP
Predicted Negative	FN	TN

The calculation of F-score is given in Equation 8.

$$F - score = \frac{TP}{TP + \frac{(FP + FN)}{2}} \tag{8}$$

E. STATISTICAL COMPARISON OF CLASSIFIERS

We performed Friedman test (Friedman, 1940) to compare the classification models' F-score statistically. The Friedman test is a useful non-parametric statistical test to compare multiple classification models over multiple data sets (Demsar, 2006). The null hypothesis of this test states that there is no difference between the classification models. It ranks the classification models' performance for each dataset in ascending order, then computes each classification model's average rank over all datasets. Equation 9 shows the Friedman test statistic calculation with *K* denotes the number of the classification model, *N* denotes the number of datasets, and AR_i denotes the average rank of each classification model over all datasets. If the test statistic value exceeds the critical value obtained from the chi-squared distribution table with K-1 degrees of freedom, we can reject the null hypothesis. The *p*-value approach can also be used for hypothesis testing. If the corresponding *p*-value of the test statistic is less than or equal to the significance level, we can reject the null hypothesis.

$$\chi_F^2 = \frac{12N}{K(K+1)}\left[\sum_{i=1}^{K} AR_i^2 - \frac{K(K+1)^2}{4}\right] \tag{9}$$

F. RESULTS AND DISCUSSIONS

Average macro F-scores of six classification models using 10-fold cross-validation are tabulated in Table IV. The bold values indicate the best F-score for each dataset in this table. It is worth noting that XGB+CMI does not give the best F-score for any data set. We can say that the reason for this is the imbalanced structure of the data sets used. The mean of the minor classes' features may not be informative due to the small instance size in this class. The XGB+MICE models provide the best F-score for three datasets. The XGB, XGB+KNN, and XGB+MI models give the best F-score for two datasets, and the XGB+SI model provides the best F-score for just one dataset.

TABLE IV
F-SCORES OF THE CLASSIFICATION MODELS

Dataset	**XGB**	**XGB +KNN**	**XGB +MICE**	**XGB +SI**	**XGB +MI**	**XGB +CMI**
Iris	0.9463	0.9463	0.9463	0.9596	**0.9597**	0.9530
Pima	0.7122	0.7172	**0.7227**	0.7003	0.6982	0.7089
Wine	0.9778	0.9721	**0.9788**	0.9635	0.9726	0.9666
Australian	0.8560	**0.8692**	0.8562	0.8545	0.8618	0.8647
Newtyroid	0.9286	**0.9620**	0.9325	0.9448	0.9393	0.9270
Ecoli	**0.7272**	0.6815	0.7121	0.6786	0.7251	0.6880
Satimage	0.8942	0.8980	**0.9015**	0.8982	0.8928	0.8840
German	**0.7054**	0.6843	0.6799	0.6836	0.6736	0.6893
Magic	0.7999	0.7982	0.7898	**0.8039**	0.7898	0.7928
Shuttle	0.9227	0.9404	0.8974	0.8963	**0.9471**	0.9223

We also visualize the results in Figure 2. The models' F-score values are very close to each other; hence, it is difficult to say that one model is superior to the others. Here, it is necessary to use a statistical test to mention a statistical difference between models.

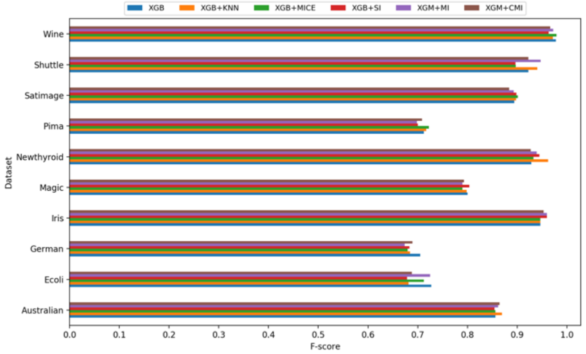

FIGURE 2. **F-scores of the classification models on all datasets.**

We used the Friedman test for statistical comparison of the six classification models. The *p*-value of the Friedman test is 0.708, so we fail to reject the null hypothesis at 95% confidence level. There is no strong evidence to support that these classification models are significantly different. So, we can conclude classification models used in this study have the same performance. Our findings show that missing data imputation methods don't have an impact on XGBoost classifier performance. The findings also support Rusdah and Murfi (2020).

V. CONCLUSIONS

XGBoost is a commonly used classifier that can handle missing data internally and doesn't require any missing data treatment for missing data. This study analyzes the impact of missing data imputation methods on the XGBoost classifier. Firstly, we impute missing data in ten datasets from the KEEL repository with five imputation methods, namely KNN, Soft-Impute, MICE, mean, and class mean, to analyze. We train XGBoost classifiers on these imputed datasets and non- imputed datasets. Finally, we compare the F-score of the classifiers statistically. This study shows that the XGBoost classifier trained on non-imputed datasets gives statistically the same results as the XGBoost classifier trained on imputed datasets.

Future works can consider including different missing data imputation methods and perform analysis in more data sets with XGBoost classifier. In this study, we only consider the artificially generated MCAR mechanism, limiting the generalization of our results. We will analyze the XGBoost classifier performance on MAR and NMAR mechanism as future work to overcome this limitation.

ACKNOWLEDGMENTS

This study is supported by Eskisehir Technical University Scientific Research Projects Committee (ESTUBAP-20DRP025).

REFERENCES

Acuña, E. and Rodriguez, C., 2004, The Treatment of Missing Values and its Effect on Classifier Accuracy, In: Banks D., McMorris F.R., Arabie P., Gaul W. (eds) Classification, Clustering, and Data Mining Applications, Springer Berlin Heidelberg, 639–647. doi:10.1007/978-3-642-17103-1_60.

Alcalá-Fdez, J., Fernandez, A., Luengo, J., Derrac, J., Garc´ıa, S., Sa´nchez, L. and Herrera, F., 2011, KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework, Journal of Multiple-Valued Logic and Soft Computing, 17(2-3), 255–287.

Batista, G. E. and Monard, M. C., 2003, An analysis of four missing data treatment methods for supervised learning, Applied Artificial Intelligence,17 (5-6), 519–533. doi:10.1080/713827181.

Budholiya, K., Shrivastava, S.K. and Sharma, V., 2020, An optimized XGBoost based diagnostic system for effective prediction of heart disease, Journal of King Saud University - Computer and Information Sciences. doi:10.1016/j.jksuci.2020.10.013.

Chao, L., Wen-hui, Z. and Ji-ming L., 2019, Study of Star/Galaxy Classification Based on the XGBoost Algorithm, Chinese Astronomy and Astrophysics, 43(4), 539-548. doi:10.1016/j.chinastron.2019.11.005.

Chen, T. and Guestrin, C., 2016, XGBoost, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. doi:10.1145/2939672.2939785.

Demsar, J., 2006, Statistical Comparisons of Classifiers over Multiple Data Sets, Journal of Machine Learning Research, 7(1), 1-30.

Dong, W., Huang, Y., Lehane B. and Ma, G., 2020, XGBoost algorithm-based prediction of concrete electrical resistivity for structural health monitoring, Automation in Construction, 114, 103155. doi:10.1016/j.autcon.2020.103155.

fancyimpute, 2020, https://pypi.org/project/fancyimpute/

Farhangfar, A., Kurgan, L. and Dy, L., 2008, Impact of imputation of missing values on classification error for discrete data, Pattern Recognition,41(12), 3692–3705..

Friedman, M., 1940, A Comparison of Alternative Tests of Significance for the Problem of m Rankings, The Annals of Mathematical Statistics, 11, 86-92.doi:10.1214/aoms/1177731944.

Kidando, E., Kitali, A. E., Kutela, B., Ghorbanzadeh, M., Karaer, A., Koloushani, M., Moses, R., Ozguven, E. E. and Sando, T., 2021, Prediction of vehicle occupants injury at signalized intersections using real-time traffic and signal data, Accident Analysis & Prevention, 149, 105869. doi:10.1016/j.aap.2020.105869.

Little, R. J. A. and Rubin, D. B., 2002, Statistical Analysis with Missing Data, John Wiley & Sons, Inc., Hoboken, NJ, USA. doi:10.1002/9781119013563.

Luengo, J., García, S. and Herrera, F., 2012, On the choice of the best imputation methods for missing values considering three groups of classification methods, Knowledge and Information Systems, 32(1), 77–108. doi:10.1007/s10115-011-0424-2.

Mazumder, R., Hastie,T. and Tibshirani, R., 2010, Spectral Regularization Algorithms for Learning Large Incomplete Matrices, Journal of Machine Learning Research, 11(80), 2287–2322.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E, 2011, Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, 12(85), 2825–2830.

Poulos, J. and Valle, R., 2018, Missing Data Imputation for Supervised Learning, Applied Artificial Intelligence, 32(2), 186–196. doi:10.1080/08839514.2018.1448143.

Rusdah, D. A. and Murfi, H., 2020, XGBoost in handling missing values for life insurance risk prediction, SN Applied Sciences, 2, 1336. doi:10.1007/s42452-020-3128-y.

Van Buuren, S. and Groothuis-Oudshoorn, K., 2011, mice: Multivariate imputation by chained equations in R, Journal of Statistical Software, 45(3), 1–67. doi:10.18637/jss.v045.i03.

Zhang, X. and Yu, Q., 2017, Hotel reviews sentiment analysis based on word vector clustering, 2017, 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA), 260-264. doi:10.1109/CIAPP.2017.8167219.

Zhou, J., Qiu, Y., Zhu, S., Armaghani, D. J., Khandelwal, M., and Mohamad, E. T, 2020, Estimation of the TBM advance rate under hard rock conditions using XGBoost and Bayesian optimization, Underground Space. doi:10.1016/j.undsp.2020.05.008.

Wang, S., Liu, S., Zhang, J., Che, X.,Yuan, Y., Wang, Z. and Kong, D., 2020,A new method of diesel fuel brands identification: SMOTE oversampling combined with XGBoost ensemble learning, Fuel, 282, 118848. doi:10.1016/j.fuel.2020.118848.