Research Article

XGBoost in handling missing values for life insurance risk prediction

Deandra Aulia Rusdah[1] · **Hendri Murfi**[1]

Received: 10 March 2020 / Accepted: 22 June 2020 / Published online: 6 July 2020

Abstract
Insurance risk prediction is carried out to classify the levels of risk in insurance industries. From the machine learning point of view, the problem of risk level prediction is a multi-class classification. To classify the risk, a machine learning model will predict the level of applicant's risk based on historical data. In the insurance applicant's historical data, there will be the possibility of missing values so that it is necessary to deal with these problems to provide better performance. XGBoost is a machine learning method that is widely used for classification problems and can handle missing values without an imputation preprocessing. This paper analyzed the performance of the XGBoost model in handling the missing values for risk prediction in life insurance. The simulations show that the XGBoost model without any imputation preprocessing gives a comparable accuracy to one of the XGBoost models with an imputation preprocessing.

Keywords Life insurance · Machine learning · Missing values · Multi-class classification · Risk prediction · XGBoost

1 Introduction

Insurance risk prediction is conducted by life insurance companies to identify risk classifications and eligibility submitted by applicants. Suppose that someone registers himself to an insurance company, then the insurance company will make a risk prediction for him based on his historical data to classify his risk level. In the insurance applicant's historical data, there will be the possibility of missing data or missing values so that it is necessary to deal with these problems to provide better performance.

The most widely used methods that can be used to handle missing value problems fall into three main categories; those are deletion, single imputation, and model-based methods [1]. Each method has some drawbacks. The disadvantage of the deletion method is that it can lead to biased parameter estimates. Moreover, by deleting some data with missing value make the data lost some of its information. The second method, single imputation, like mean and median imputation, is straightforward that replaces each missing value with the mean or median of non-missing values of the variable or item. The simplicity of this method is also its disadvantage; that is, the distribution of the imputed variables can get highly distorted, and the variance is underestimated because each missing value is assigned the same imputation value. There are two main drawbacks in the third category, model-based method, like *k*-nearest neighbors (KNN), i.e., the model estimated values are usually more well behaved than the true values, and the models perform poorly if the observed and missing variables are independent [2].

There also some advanced methods proposed for the problem of the missing values. Doreswamy and Vastrad [3] proposed and evaluated an iterative imputation method called MiFoImpute. This method is based on a random forest method for incomplete molecular descriptor data. Bertsimas et al. [4] propose and analyze a general imputation algorithm opt.impute that produces the best imputation than five other methods, i.e., Bayesian PCA, mean impute, predictive mean matching, iterative KNN, and *k*-nearest neighbors. Kim et al. [5] analyze missing values in

✉ Deandra Aulia Rusdah, deandra.aulia@sci.ui.ac.id | [1]Department of Mathematics, Universitas Indonesia, Depok 16424, Indonesia.

SN Applied Sciences (2020) 2:1336 | https://doi.org/10.1007/s42452-020-3128-y

the day-ahead PV generation with the imputation method using *k*-nearest neighbors.

From a machine learning point of view, the risk level is represented as a class, and if there is more than one class, it is referred to as multi-class. Therefore, the problem of insurance risk with eight levels of risk is included in the multi-class classification. Given the historical data, it is necessary to build a machine learning model that predicts the risk levels of the applicants. XGBoost is a machine learning method that is widely used for classification problems. XGBoost is a gradient tree boosting-based method with some extensions. One of the extensions is the sparsity awareness that can handle the possibility of missing values. Therefore, XGBoost can process data with missing values without doing imputation first [6].

Fauzan and Murfi [7] applied and analyzed the accuracy of XGBoost for claim prediction, where the missing values were being imputed by the standard methods of mean and median. This XGBoost method was also compared to other ensemble learning with similar imputation methods. The result showed that the XGBoost provided better accuracies than the other methods. Mustika et al. [8] analyzed the accuracy of XGBoost for risk assessment. The results show that XGBoost also gave better accuracy than other methods, i.e., Bayesian Ridge, random forest, and decision tree.

This paper analyzed the performance of the XGBoost model in handling the missing values for risk prediction in life insurance. It also compares the performance with standard imputation methods, i.e., mean and KNN method, to know how well XGBoost in handling data that contain missing value without doing the imputation preprocessing first.

The next section is organized as follows: Sect. 2 presents the materials and methods of this paper. In Sect. 3, the results of the simulations are described. In Sect. 4, the conclusion of this paper is given.

2 Materials and methods

2.1 Dataset

This research used the level of risk data from the Prudential Life Insurance Company from the Kaggle. The data can be obtained from https://www.kaggle.com/c/prudential-life-insurance-assessment/data. The data show life insurance applicant information that can be used to predict the level of risk.

In the data, there are 59,381 observations or individuals, 128 features that show life insurance applicant attributes consisting of 60 features of categorical variables, 13 continuous variable features, 5 discrete variable features, 48 dummy variable features, 1 Id feature that shows the id for a region, and 1 response feature that shows life insurance risk category. There are also 13 features that contain missing values, which is presented in Table 1.

Table 1 Percentage of missing values

Feature	Missing value (%)
Employment_Info_1	0.032
Employment_Info_4	11.42
Employment_Info_6	18.28
Insurance_History_5	42.77
Family_Hist_2	48.26
Family_Hist_3	57.67
Family_Hist_4	32.31
Family_Hist_5	70.42
Medical_History_1	14.97
Medical_History_10	99.07
Medical_History_15	75.11
Medical_History_24	93.60
Medical_History_32	98.14

Table 2 Level of risk

Risk level	Observation
1	6207
2	6552
3	1013
4	1428
5	5432
6	11,233
7	8027
8	19,489

Table 1 shows that the highest percentage of missing values is in Medical_History_10 with 99.07% missing values. All data on features that contain missing values are float data type.

Another problem that occurs is an imbalance in the target. Target in the data is response, which shows the level of life insurance risk. The imbalance of the target data can be seen from the amount of observation at each risk level. Table 2 shows the amount of observation at each risk level.

Based on Table 2, there are differences in the amount of data at the 8 levels of risk. There is such a big difference in the amount of observation between risk level 8 and the others. The difference in the amount of data indicates that there is an imbalance problem in the data.

2.2 Missing value

A missing value is information that is not available in an object or case. Missing value occurs when information for something about an object is not given, challenging to find, or, indeed, the information is not available. If the historical data of the applicant contain a large percentage of missing values, then it is necessary to handle it.

According to the mechanism of missingness, there are three types of missing values, i.e., missing completely at random (MCAR), when the values of missing data have not related to the values in the observed dataset, missing at random (MAR), when response opportunity of the missing depends on the observed dataset, but not associated with specific expected missing values, and missing not at random (MNAR), when data characteristics are not included in the MCAR and MAR to get an unbiased estimate of parameters such as the case of missing data in the model [8].

Several papers are dealing with the problem of missing values. Little and Rubbin [9] researched missing values with statistical analysis. Zhang et al. [10] used a missing value model that cannot be ignored with the MCMC sampling algorithm in the Bayesian method. Ma and Chen [11] conducted similar research with the Bayesian approach for handling missing values. Dewi et al. [12] researched handling missing values by replacing missing values with 0 (zero), mean values, medians, and values that often arise from data in the same column. The research represents that the XGBoost method can work on data that contain missing values without handling the data. Stephen researched handling missing data in numeric analyses [13]. Wijasekara and Liyanage [14] made a comparison of imputation methods for missing values in air pollution data. Sanjar et al. [15] researched missing data imputation for geolocation-based price prediction using KNN-MCF method.

2.3 Imbalance dataset

An imbalance dataset or imbalance in data is one of the classification problems. When the dataset is imbalanced, the classification algorithm does not have sufficient information relating to the minority class to get an accurate prediction [16]. In the multi-class classification case, data imbalance can happen between one and another class. It said to be unbalanced when there are significant differences in data in each class. It can occur when classes that have relatively little data (minority classes) much smaller or less frequent than classes that have the most data (majority classes) [17].

Machine learning and deep learning algorithms are strongly affected by the class imbalance problem. Thus, the class imbalance problem needs to be handled [18]. Imbalance of data between classes in a multi-class classification can be handled with several techniques. One of them is the oversampling technique. Oversampling is a technique used to adjust the class distribution of datasets. One popular oversampling method is the random over sampler, which works by randomly duplicating minority class samples to achieve a balanced distribution in both classes [19].

2.4 XGBoost

XGBoost is a method that used the development of the basic gradient tree boosting model to become extreme gradient boosting. Based on a paper that is written by [6], the model used classification and regression tree (CART). CART is a binary decision tree that divides a node into two leaf nodes repeatedly. CART tree formation is done by selecting the best splitting for each node of the tree. For example, given $\mathbf{D}$ as follows:

$$\mathbf{D} = \{(\mathbf{x}_i, y_i)\}, \quad 1 \le i \le n \tag{1}$$

$$\mathbf{x}_i = [x_{ij}], \quad 1 \le j \le m \tag{2}$$

where n and m are the number of applicants and features in the data consecutively, $\mathbf{x}_i$ is the i life insurance applicant data, y_i is the level of risk insurance claim of the i applicant for life insurance or the actual target, and x_{ij} is the i individual data with jth feature.

In the multi-class classification problem, the process of forming a CART is done by grouping the same class in each branch of the tree that is created, starting with a node containing all the data; then, splitting the process continues until it reaches the stopping criteria. In the splitting rule, there is a node a which becomes two vertices left (L) and right (R) with $s_{ij} \in \{a_{ij} | i \in I_a, 1 \le j \le m\}$ as a splitting point with the results of splitting

$$I_a = I_\mathrm{L} \cup I_\mathrm{R} \tag{3}$$

$$I_\mathrm{L} = \{i \in I_a | a_{ij} < s_{ij}\} \tag{4}$$

$$I_\mathrm{R} = \{i \in I_a | a_{ij} \ge s_{ij}\} \tag{5}$$

the best determination s_{ij} is to test all possible combinations of s_{ij}, i.e., the minimum Gini index value for the two vertices $G_\mathrm{L} + G_\mathrm{R}$. If the data at the node consist of only one level or a similar level, then the Gini value of a node will be minimum. Gini index is used to determine the purity of a node that is defined by

$$G = 1 - \sum_{i=1}^{M} (p_i)^2 \tag{6}$$

where M is a class that has the probability of each p_i where $i = 1, 2, \ldots, M$.

In gradient tree boosting, the splitting process uses all values in data that are used as splitting points. There is a split determination algorithm for each feature that is the exact greedy algorithm [6].

Algorithm 1. Exact Greedy Algorithm for Split Finding

Input: $\boldsymbol{I}$, instance set of current node
Input: $\boldsymbol{d}$ feature dimension
$gain \leftarrow 0$
$G \leftarrow \sum_{i \in I} g_i$, $H \leftarrow \sum_{i \in I} h_i$
for $k = 1$ $\boldsymbol{to}$ m $\boldsymbol{do}$
  $G_L \leftarrow 0$, $H_L \leftarrow 0$
  for j in sorted $(I, by\ x_{jk})$ do
    $G_L \leftarrow G_L + g_j$, $H_L \leftarrow H_L + h_j$
    $G_R \leftarrow G - G_L$, $H_R \leftarrow H - H_L$
    $score \leftarrow max\left(score, \frac{{G_L}^2}{H_L+\lambda} + \frac{{G_R}^2}{H_R+\lambda} - \frac{G^2}{H+\lambda}\right)$
  end
end
Output: Split with max score

Problems will arise if missing values occur in the dataset. On XGBoost, it can be handled with a sparsity-aware split finding algorithm that can accurately handle missing values on XGBoost. The algorithm helps in the process of creating a CART on XGBoost to work out missing values directly. Here is the algorithm of sparsity-aware split finding algorithm found on XGBoost [6].

Algorithm 2. Sparsity Aware Split Finding Algorithm

Input: $\boldsymbol{I}$, instance set of current node, $\boldsymbol{I_k} = \{\boldsymbol{i} \in \boldsymbol{I} | \boldsymbol{x_{ik}} \neq \boldsymbol{missing\ value}\}$, $\boldsymbol{d}$ feature dimension.
Also applies to the approximate setting, only collect statistic of non-missing entries into buckets.
$gain \leftarrow 0$
$G \leftarrow \sum_{i \in I} g_i$, $H \leftarrow \sum_{i \in I} h_i$
for $k = 1$ to m do
  // enumerate missing values go to right
  $G_L \leftarrow 0$, $H_L \leftarrow 0$
  for j in sorted do
    $G_L \leftarrow G_L + g_j$, $H_L \leftarrow H_L + h_j$
    $G_R \leftarrow G - G_L$, $H_R \leftarrow H - H_L$
    $score \leftarrow max\left(score, \frac{{G_L}^2}{H_L+\lambda} + \frac{{G_R}^2}{H_R+\lambda} - \frac{G^2}{H+\lambda}\right)$
  end
  // enumerate missing values go to left
  $G_R \leftarrow 0$, $H_R \leftarrow 0$
  for j in sorted do
    $G_R \leftarrow G_R + g_j$, $H_R \leftarrow H_R + h_j$
    $G_L \leftarrow G - G_R$, $H_L \leftarrow H - H_R$
    $score \leftarrow max\left(score, \frac{{G_L}^2}{H_L+\lambda} + \frac{{G_R}^2}{H_R+\lambda} - \frac{G^2}{H+\lambda}\right)$
  end
end
Output: Split and default direction with max gain

2.5 Machine learning

Machine learning is a method used to process data and information to be applied to further data. Machine learning implements various algorithms that iteratively learn to improve data, describe data, and predict results [20]. Several stages need to be done when processing data using machine learning, including vectorization, preprocessing, learning, and evaluation.

2.5.1 Vectorization

Vectorization is the process of representing data in the form of vectors. The process is carried out after the collection of data to be used, which is raw data in the form of Excel, Ms. Access, and text files. After the vectorization process is completed, the data in the form of this vector are identified and processed at a later stage.

2.5.2 Preprocessing

In general, the data that have been collected cannot be handled directly with machine learning because it has several problems, such as data containing missing values, imbalances, or inconsistencies of data. So it is necessary to do preprocessing before the data can be used to make a model.

In this research, data preprocessing is done by changing categorical variables to dummy variables with one hot encoder and standardization for continuous variables. Techniques used for standardization are standard scaler, which is standardization using normal distribution or statistical normalization (*Z*-score) [21]. This formula can be written as follows:

$$x' = \frac{(x_i - \mu_i)}{\sigma_i} \tag{7}$$

where μ_i is the mean and σ_i is the standard deviation of the ith data.

2.5.3 Learning

Learning is a process of determining the values of the model's parameters based on data. In the learning process will be performed a fitting model which is a matching machine learning model that is suitable for the problem to be solved. There are two kinds of learning in machine learning. Those are unsupervised learning and supervised learning.

In supervised learning, there was a target feature that is included in training data so that each of the training data is in the form of data pairs. The purpose of supervised learning is to make a model that can give output that best suits the target for all training data [22]. There are two kinds of problems that can be solved using supervised learning, i.e., classification and regression. Classification will produce output in the form of classes or categories to classify data accurately, while regression will produce output in the form of continuous or real values. In unsupervised learning, there is no target feature in training data. Unsupervised learning builds a model that can describe hidden structures in data. The problems included in this research are clustering and dimensionality reduction.

The problem in this research is a kind of supervised learning problem, which is a multi-class classification. The dataset in this research contains input and output pairs that show historical data for life insurance applicants, and the target is the class of the life insurance applicant's risk.

2.5.4 Evaluation

In evaluation, the model accuracy estimation process is evaluated using cross-validation to see how well the model fits the data.

2.6 Risk assessment as a machine learning problem

Life insurance risk assessment is a multi-class classification problem that is included in supervised learning. The performance of supervised learning can be measured using evaluation metrics. The evaluation metric that is used in this research is a confusion matrix. Confusion matrix is a performance measurement of the classification model. In a confusion matrix, the predicted class will be compared with the actual class. Each column in the matrix shows the predicted results for the class corresponding to the column, while each row shows the actual class.

Figure 1 shows the confusion matrix for multi-class classification, where *ck* is a positive class, and other than *ck* is a negative class. TN (true negative) is the frequencies of results where the model predicts negative classes correctly. FP (false positive) is the frequencies of results where the model incorrectly predicts a positive class. TP (true positive) is the frequencies of results where the model predicts positive classes correctly. FN (false negative) is the frequencies of results where the model incorrectly predicts a negative class.

Confusion matrix is used to calculate some performance metrics, such as accuracy, precision, and recall. Accuracy

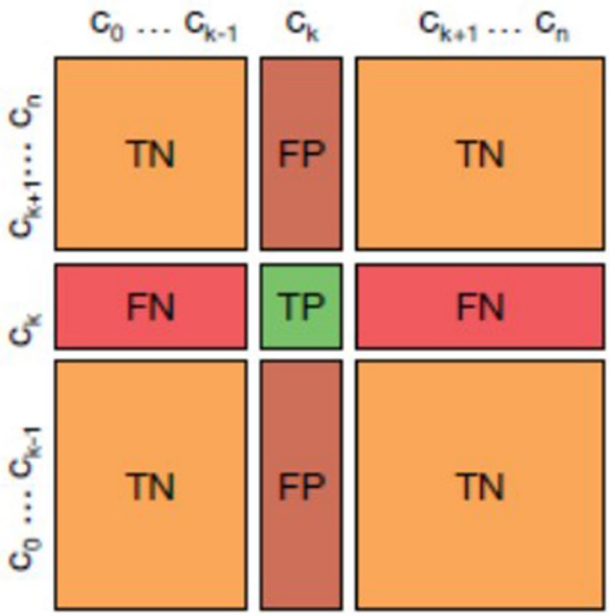


Fig. 1 Confusion matrix for multi-class

is an evaluation metric to measure the total number of predictions a model gets right. The formula for accuracy is given below:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}. \quad (8)$$

Precision evaluates how precise a model is in predicting positive labels. Precision is the percentage of the results which are relevant. The formula for precision can be written as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (9)$$

Recall calculates the percentage of actual positives a model correctly identified (true positive). The formula for the recall is given below:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (10)$$

Table 3 Best XGBoost hyperparameter for data with mean and KNN imputation

Hyperparameter	Value
Max_depth	24
Min_child_weight	0
Subsample	0.8
Colsample_bytree	0.8
Learning Rate	0.1
Alpha	0
Lamda	0

Table 4 Best XGBoost hyperparameter for data without imputation

Hyperparameter	Value
Max_depth	24
Min_child_weight	0
Subsample	0.8
Colsample_bytree	0.9
Learning rate	0.2
Alpha	0
Lamda	0

3 Simulation

3.1 Preprocessing

Preprocessing is done in this case, including making feature data and target data, overcoming missing values, changing categorical variables into dummy variables, standardizing continuous variables, and overcoming imbalanced datasets. First is creating feature data and target data. Feature data are obtained by removing the Id and response features of the dataset so that the dataset becomes 126 features. Target data are the response in the dataset. Second is checking the missing values. In this research, no missing value features are omitted because the XGBoost model can overcome the missing value problem without doing the imputation process first. That is because, in the XGBoost model, there is a sparsity awareness algorithm that can handle missing values. Therefore, in this preprocessing process, two data are prepared for this study, data that overcome the missing value problem using the imputation process on features that contain missing values using the mean and KNN of data and data without imputation on the missing values. Next is changing categorical variables into dummy variables. In this research, one hot encoder is used to transform categorical into dummy variables. After the one hot encoder process, the feature data become 943 features. Then is the standardization of continuous variables. Standardization is carried out due to differences in measurement units in continuous data using Eq. (7). The last is to solve the issue of the imbalance in the dataset. In this research, an oversampling technique was performed using random over sampler to overcome the imbalance dataset problem.

3.2 Model fitting, hyperparameter model, and results

Model fitting is performed using training data. The best hyperparameter values of the model used for the model fitting process were obtained from the optimization process in the XGBoost model with imputation and without imputation presented in Tables 3 and 4.

Based on [23, 24], the XGBoost model hyperparameter that needs to be optimized is: maximum depth (max_depth), a parameter of tree depth level; minimum child weight which is the minimum number of hessian weights; the subsample, a sample ratio of training data; colsample by a tree, the ratio of the sample column when creating each tree; the learning rate, a shrinkage parameter; alpha, the L1 regularization parameter; and lamda, the L2 regularization parameter.

After fitting the model using training data, the XGBoost model that is obtained from data with mean and KNN imputation will be compared with the model that is derived from data without imputation. Model performance is measured using an accuracy metric. The accuracy of the model is obtained from the evaluation of the XGBoost model using the best hyperparameters. A cross-validation procedure with five folds is used to evaluate each model's performance. Boxplot from the results of cross-validation is shown in Fig. 2.

The confusion matrix from the model's results is shown in Figs. 3, 4 and 5.

From the confusion matrix in Figs. 3, 4 and 5, the precision and recall of the model for each class and also the accuracy score can be calculated. The results are presented as the graph in Figs. 6, 7 and 8.

Figures 5, 6, and 7 show precision and recall of each class from the model without imputation higher than the model with mean and KNN imputation.

The accuracy of the XGBoost model is presented in Table 5.

Furthermore, it can be seen that model which learned from data without imputation produces better accuracy scores than that with standard imputation.

From the results of accuracy, precision, and recall score, it can be said that the XGBoost model can handle the missing value problem. It means XGBoost can work with missing values without first carrying out the data imputation process.

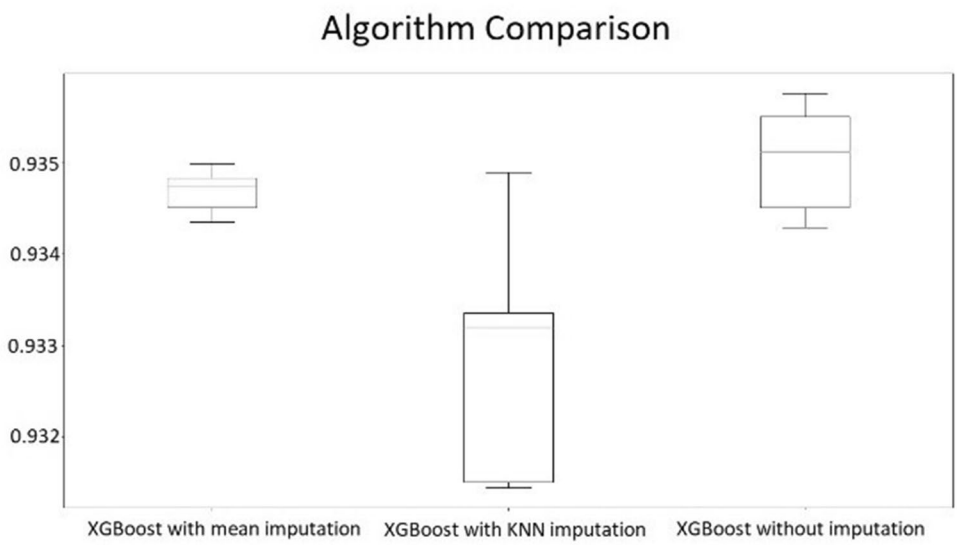


Fig. 2 Boxplot from cross-validation results of model with mean (left), KNN (center), and without imputation (right)

Fig. 3 Confusion matrix of XGBoost model from data with mean imputation

$$\begin{bmatrix} 3868 & 42 & 0 & 0 & 4 & 48 & 12 & 17 \\ 56 & 3698 & 0 & 0 & 11 & 61 & 6 & 22 \\ 7 & 9 & 3880 & 0 & 0 & 3 & 0 & 0 \\ 6 & 18 & 0 & 3907 & 0 & 3 & 0 & 1 \\ 21 & 45 & 0 & 0 & 3906 & 61 & 2 & 6 \\ 69 & 72 & 0 & 0 & 51 & 3223 & 82 & 122 \\ 49 & 35 & 0 & 0 & 14 & 210 & 3552 & 237 \\ 78 & 64 & 0 & 0 & 25 & 256 & 207 & 3299 \end{bmatrix}$$

Fig. 4 Confusion matrix of XGBoost model from data with KNN imputation

$$\begin{bmatrix} 3688 & 38 & 0 & 0 & 10 & 54 & 9 & 22 \\ 47 & 3697 & 0 & 0 & 12 & 67 & 9 & 12 \\ 6 & 8 & 3880 & 0 & 0 & 1 & 0 & 0 \\ 9 & 22 & 0 & 3907 & 0 & 5 & 0 & 1 \\ 37 & 52 & 0 & 0 & 3898 & 73 & 1 & 3 \\ 75 & 68 & 0 & 0 & 51 & 3164 & 104 & 179 \\ 33 & 28 & 0 & 0 & 15 & 206 & 3513 & 189 \\ 77 & 70 & 0 & 0 & 25 & 295 & 225 & 3298 \end{bmatrix}$$

Fig. 5 Confusion matrix of XGBoost model from data without imputation

3701	39	0	0	7	47	16	19
42	3704	0	0	13	48	12	8
6	7	3880	0	0	2	0	0
7	19	0	3907	0	2	0	1
27	45	0	0	3901	59	2	4
79	74	0	0	54	3217	98	126
41	27	0	0	11	197	3516	199
69	68	0	0	25	293	217	3347

Fig. 6 Precision and recall from model with mean imputation

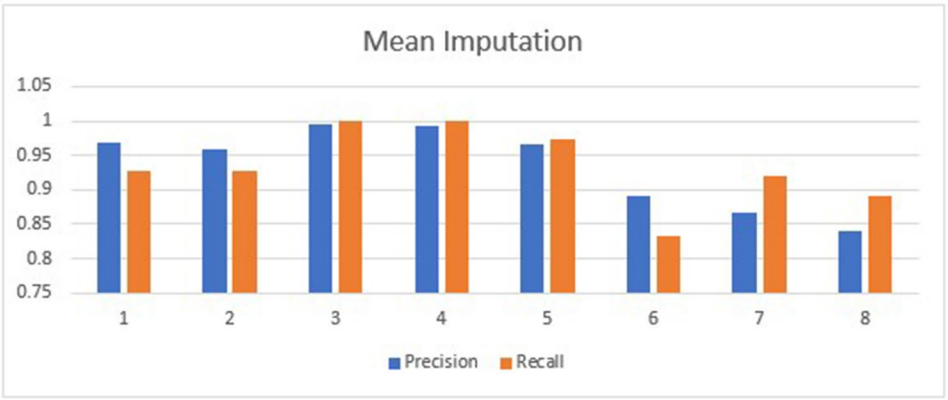
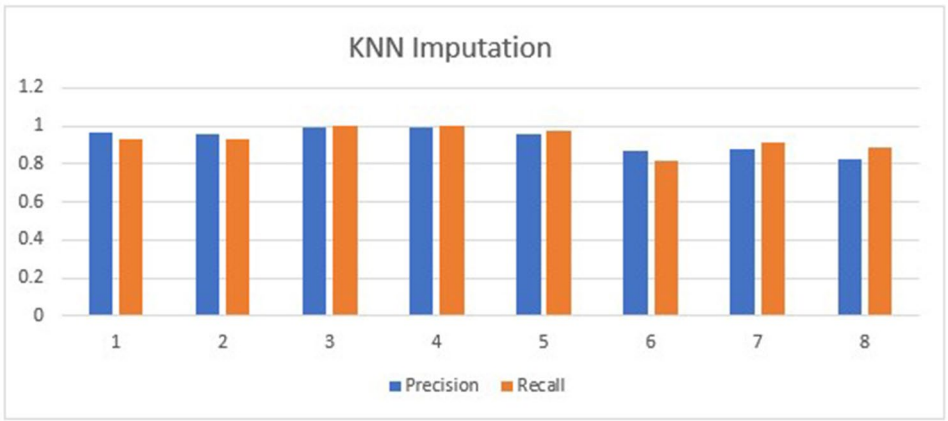


Fig. 7 Precision and recall from model with KNN imputation



4 Conclusion

Life insurance risk prediction is an important way to classify the risk of insurance applicants based on historical data of each life insurance applicant. Historical data will be different for each insurance applicant, so there is a possibility of missing historical data. Therefore, XGBoost is needed to handle missing values because there is a sparsity-aware finding algorithm for dealing with missing values in life insurance risk prediction. In the multiclass classification problem, there can be a possibility of an imbalance of data between classes, so it needs to be done for the issue of class imbalance using oversampling techniques.

In this research, an accuracy analysis was performed on the XGBoost model that is obtained from data with the standard imputation of the missing values and comparing it with the accuracy of the XGBoost model from data without imputation process. The XGBoost model that is obtained from data without imputation of missing values gives comparable accuracy than the XGBoost model that is obtained from data with standard imputation.

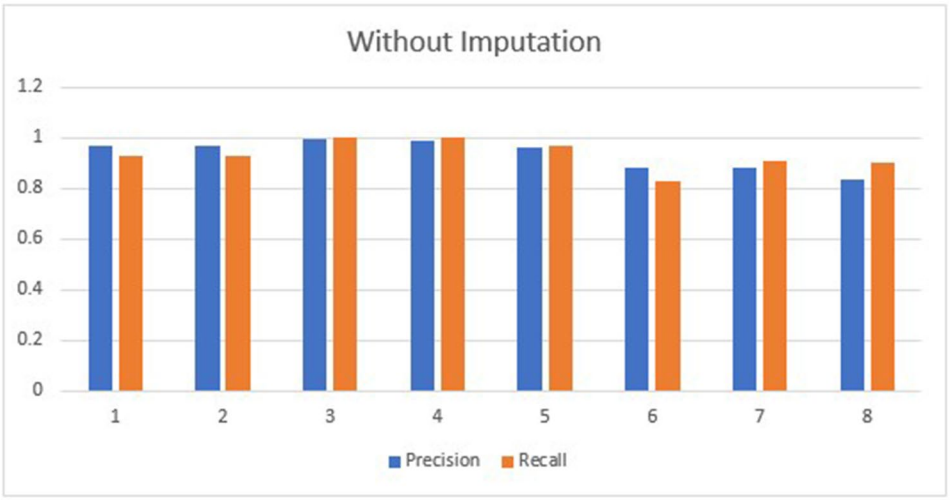


Fig. 8 Precision and recall from model without imputation

Table 5 Accuracy of XGBoost model

XGBoost	Accuracy
Without imputation	0.94
With mean imputation	0.93
With KNN imputation	0.93

Acknowledgements This work was supported by Universitas Indonesia under the 2020 PUTI Prosiding grant. Any opinions, findings, and conclusions or recommendations are the author's and do not necessarily reflect those of the sponsor.

References

1. Salgado CM, Azevedo C, Proença H, Vieira SM (2016) Secondary analysis of electronic health records. Springer Nature, Cambridge
2. Lodder P (2013) To impute or not impute: that's the question. In: Mellenbergh JG, Ader HJ (eds) Advising on research methods: selected topics. Johannes van Kessel Publishing, Huizen
3. Doreswamy H, Vastrad CM (2013) A robust missing value imputation method MiFoImpute for incomplete molecular descriptor data and comparative analysis with other missing value imputation methods. Int J Comput Sci Appl (IJCSA) 3(4):63–74
4. Bertsimas D, Pawlowski C, Zhuo YD (2018) From predictive methods to missing data imputation: an optimization approach. J Mach Learn Res 18:1–39
5. Kim T, Ko W, Kim J (2019) Analysis and impact evaluation of missing data imputation in day-ahead PV generation forecasting. Appl Sci 9(1):1–18. https://doi.org/10.3390/app9010204
6. Chen T, Guestrin C (2016) XGBoost: a scalable tree boosting system. In: KDD'16 Proceedings of the 22nd SCM SIGKDD international conference on knowledge discovery and data mining, pp 785–794
7. Fauzan MA, Murfi H (2018) The accuracy of XGBoost for insurance claim prediction. Int J Adv Soft Comput Appl 10(2):159–171
8. Mustika WF, Murfi H, Widyaningsih Y (2019) Analysis accuracy of XGBoost model for multiclass classification—a case study of applicant level risk prediction for life insurance. In: 5th International conference on science in information technology (ICSITech)
9. Little R, Rubin D (2002) Statistical analysis of missing data, 2nd edn. Wiley, New York
10. Zhang X, Boscardin WJ, Belin TR, Wan X, He Y, Zhang K (2015) A Bayesian method for analyzing combinations of continuous, ordinal, and nominal categorical data with missing values. J Multivar Anal 135:43–58. https://doi.org/10.1016/j.jmva.2014.11.007
11. Ma Z, Chen G (2017) Bayesian methods for dealing with missing data problems. J Korean Stat Soc 47(3):297–313. https://doi.org/10.1109/LGRS.2013.2286078
12. Dewi KC, Mustika WF, Murfi H (2019) Ensemble learning for predicting mortality rates affected by air quality. J Phys Conf Ser 1192(1):012021. https://doi.org/10.1088/1742-6596/1192/1/012021
13. Gorard S (2020) Handling missing data in numeric analyses. Int J Soc Res Methodol 00(00):1–10. https://doi.org/10.1080/13645579.2020.1729974
14. Wijesekara WMLKN, Liyanage L (2020) Comparison of imputation methods for missing values in air pollution data: case study on Sydney Air Quality index. In: Arai K, Kapoor S, Bhatia R (eds) Advances in information and communication. FICC 2020. Advances in intelligent systems and computing, vol 1130. Springer, Cham
15. Sanjar K, Bekhzod O, Kim J, Paul A, Kim J (2020) Missing data imputation for geolocation-based price prediction using KNN-MCF method. ISPRS Int J Geo-Inf 9(4):227. https://doi.org/10.3390/ijgi9040227
16. Bejjanki KK, Gyani J, Gugulothu N (2020) Class imbalance reduction (CIR): a novel approach to software defect prediction in the presence of class imbalance. Symmetry (Basel) 12(3):407. https://doi.org/10.3390/sym12030407
17. Ren F, Cao P, Li W, Zhao D, Zaiane O (2017) Ensemble based adaptive over-sampling method for imbalanced data learning in computer aided detection of microaneurysm. Comput Med Imaging Graph 55:54–67
18. Buda M, Maki A, Mazurowski MA (2017) A systematic study of the class imbalance problem in convolutional neural networks. Neural Netw 106:249–259
19. Syukron A, Subekti A (2018) Penerapan Metode Random Over-Under Sampling dan Random Forest Untuk

Klasifikasi Penilaian Kredit. J Inform 5(2):175–185. https://doi.org/10.31311/ji.v5i2.4158
20. Hurwitz J, Kirsch D (2018) Machine learning for dummies. Wiley, New York
21. Jayalakshmi T, Santhakumaran A (2011) Statistical normalization and back propagation for classification. Int J Comput Theory Eng 3(1):89–93. https://doi.org/10.7763/ijcte.2011.v3.288
22. Bishop CM (2006) Pattern recognition and machine learning. Springer, New York. ISSN : 1613-9011
23. Lim S, Chi S (2019) Xgboost application on bridge management systems for proactive damage estimation. Adv Eng Inform 41:100922. https://doi.org/10.1016/j.eswa.2019.01.083
24. Martinez-de-Pison FJ, Gonzalez-Sendino R, Aldama A, Ferreiro-Cabello J, Fraile-Garcia E (2018) Hybrid methodology based on Bayesian optimization and GA-PARSIMONY to search for parsimony models by combining hyperparameter optimization and feature selection. Int Conf Hybrid Artif Intell Syst 10334:52–62. https://doi.org/10.1016/j.bdr.2017.07.003