Various data imputation techniques in R

Authors: Jan Borowski, Filip Chrzuszcz, Piotr Fic (Warsaw University of Technology)

Abstract

There are many suggestions how to deal with missing values in data sets problem. Some solutions are offered in publicly available packages for the R language. In our study, we tried to compare the quality of different methods of data imputation and their impact on the performance of machine learning models. We scored different algorithms on various data sets imputed by chosen packages. Results summary presents packages which enabled to achieve the best models predictions metrics. Moreover, duration of imputation was measured.

Introduction and Motivation

Background an related work

Missing observations in data sets is a common and difficult problem. In the field of machine learning, one of the key objects is the data set. Real-world data are often incomplete, which prevents the usage of many algorithms. Most implementations of machine learning models, available in popular packages, are not prepared to deal with missing values. Before creating a machine learning model, it is essential to solve the problem of missing observations. This requires user pre-processing of data. Some researches examined similarity between original and imputed data, in terms of descriptive statistics (Musil et al. 2002). Missing data are common in medical sciences and impact of different imputation methods on analysis was measured (Bono et al. 2007). Some studies show that imputation can improve the results of machine learning models and that more advanced techniques of imputation outperform basic solutions (Batista and Monard 2003) (Su, Khoshgoftaar, and Greiner 2008).

Motivation

Various imputation techniques are implemented in different packages for the R language. Their performance is often analyzed independently and only in terms of imputation alone. Because of variety of available tools, it becomes uncertain which one package and method to use, when complete data set is needed for machine learning model. In our study we would like to examine, how methods offered by some popular packages perform on various data sets. We want to consider flexibility of these packages to deal with different data sets. The most important issue for us is the impact of performed imputation on later machine learning model performance. We are going to consider one specific type of machine learning tasks: *supervised binary classification*. Our aim is a comparison of metrics scores achieved by various models depending on chosen imputation method.

Definition of missing data

At the beginning, clariyfing the definition of missing data is necessary. Missing data means, that one or more variables have no data values in observations. This can be caused by various reasons, which we can formally define as follows, referring to Rubin (1976):

- MCAR (Missing completely at random)
  Values are missing completely at random if the events that lead to lack of value are independent both of observable variable and of unobservable parameters. The missing data are simply a random subset of the data. Analysis performed on MCAR data is unbiased. However, data are rarely MCAR.

- MAR (Missing at random)
  Missingness of the values can be fully explained by complete variables. In other words, missing data are not affected by their characteristic, but are related to some or all of observed data. This is the most common assumption about missing data.
- MNAR (Missing not at random)
  When data are missing not at random, the missingness is related to the characteristic of variable itself.

Techniques of dealing with missing data

In case of preparing a data set for machine learning models we can generally distinguish two approaches. The first method is **omission**. From the data set we can remove observations with at least one missing value or we can remove whole variables where missing values are present. This strategy is appropriate if the features are MCAR. However, it is frequently used also when this assumption is not met. It is also useless when the percentage of missing values is high. The second approach is **imputation**, where values are filled in the place of missing data. There are many methods of imputation, which we can divide into two groups. **Single imputation** techniques use information of one variable with missing values. Popular method is filling missings with mean, median or mode of no missing values. More advanced are predictions from regression models which are applied on the mean and covariance matrix estimated by analysis of complete cases. The main disadvantage of single imputation is treating the imputed value as true value. This method does not take into account the uncertainty of the missing value prediction. For this reason **multiple imputation** was proposed. This method imputes k values, which leads to creating k complete data sets. The analysis or model is applied on each complete data set and finally results are consolidated. This approach keeps the uncertainty about the range of values which the true value could have taken. Additionally, multiple imputation can be used in both cases of MCAR and MAR data.

Methodology

Experiment like this one can be performed involving many techniques we decide to divide our tests into 4 steps:

- Data Preparation,
- Data Imputation,
- Model Training,
- Model Evaluation.

Bellow we will explain every step in detail.

1. Data Preperation

For test purposes we used 14 datasets form OpenML library (Casalicchio et al. 2019). Every dataset is designed for binary classification and most of them contain numerical and categorical features. Most of the sets have a similar number of observations in both classes but some of them were very unbalanced. Before data imputation data set was prepared specific preparation are different for each data set but we commonly do:

- Removing features which didn't contain useful information (for example all observation have the same value)
- Correcting typos and converting all string to lower case to reduce the number of categories
- Converting date to more than one column (for example "2018-03-31" can be converted to three column year, month and day)
- Removing or converting columns with too many categories

After cleaning data sets were transferred to the next step.

2.Data Imputation

Clean datasets were split into two data sets train and test in proportion 1/4 respectively. This split was performed randomly and only once for every dataset that's mean every imputation technique used the same split. Imputation was performed separately for train and test sets. Before split we also remove the target column to avoid using it in imputation. For our study, we decided to choose five packages designed for missing data imputation in the R language and one self-implemented basic technique:

- **Mode and median**: Simple technique of filling missing values with mode (for categorical variables) and median (for continuous variables) of complete values in a variable. Implemented with basic R language functions.
- **mice**(van Buuren and Groothuis-Oudshoorn 2011): Package allows to perform multivariate imputation by chained equations (MICE), which is a type of multiple imputation. The method is based on Fully Conditional Specification, where each incomplete variable is imputed by separate model.
- **missMDA**(Josse and Husson 2016): Package for multiple missing values imputation. Data sets are imputed with the principal component method, regularized iterative FAMD algorithm (factorial analysis for mixed data). First estimation of the number of dimensions for factorial analysis is essential.
- **missFOREST**(Stekhoven and Buehlmann 2012): Package can be used for imputation with predictions of random forest model, trained on complete observations. Package works on data with complex interactions and non-linear relations. Enables parallel calculations.
- **softImpute**(Hastie and Mazumder 2015): Package for matrix imputation with nuclear-norm regularization. Algorithm works like EM, solving an optimization problem using a soft-thresholded SVD algorithm. Works only with continuous variables.
- **VIM**(Kowarik and Templ 2016): Package for visualization and imputation of missing values. It offers iterative robust model-based imputation (IRMI). In each iteration, one variable is used as a response variable and the remaining variables as the regressors.

First we use mode/median imputation, which is a very simple method and it is used as a base result for more complex algorithms to compare. Imputation method form mice package don't require any form of help because can impute numeric and categorical features. SoftImpute package works only with numeric features. To compare it with other algorithms on the same data we use SoftImput for numeric variables and mode for categorical variables. Alternatively, it is possible to use SoftImpute for numeric features and different algorithms for categorical variables, but we decided that this approach may lead to unreliable results. MissForest algorithm can be used on both numeric and categorical features and is capable of performing imputation without any help of other methods. Imputation method from Mice package also can be run on all types of data. Iterative Robust Model-Based Imputation method from VIM package can impute all types of data. This method additionally creates new columns with information whether observation was imputed or not. We decided to do not use these columns, because other methods do not create them. Last method which we covered is missMDA which also can be used to input numeric and categorical features. After imputation we add back target variable to both sets. All methods work on the same parameters for all data sets. In case when for some reason method can't input some data set it was treated like "worst result" more information about it can be found in section 4.Model evaluation.

3.Model traing

For classification task we use four classification algorithms:

- Extreme Gradient Boosting,
- Random Forest,
- Support Vector Machines,
- Linear Regression

All methods were implemented in **mlr** package (**???**) for hyperparameters tuning, we also used methods from the same package. For all data sets four classifiers were trained and tuned on the same train sets.

To select parameters we used Grid Search. We will not focus on this part of the experiment. The most important part in this step, is that every model training was proceeded the same way. This mean that differences in results can be caused only by influence of previously used imputation technique.

4.Model evaluation

After previous steps, we have got trained models and test sets. In the final step we evaluate model and imputation. For every imputation and algorithm we calculate F1 score expressed by formula $2\frac{(precision)*(recall)}{precision+recall}$ and accuracy. In case when imputation algorithm fail to impute some data set results for this set are thread as "the worst result". It means if u try to create a ranking it is always last (if more then one imputation fail all of them is placed last). A detailed discussion about results in the next section.

5.Results

After the long and tedious process of data imputation it is finally time to evaluate our methods of imputation. Methods were trained and tested on 11 datasets and evaluated strictly using 2 methods mentioned earlier. Before score analysis it is worth mentioning that all algorithms were tested with optimal parameters, so score should be quite meaningful. Table below presents average metrics achieved by model, depending on imputation method. As described earlier, we decided to use two measures of algorithm effectiveness:

- F1 score
- Accuracy

These two measures complement each other well because they allow us to measure well the effectiveness of our imputations and algorithms on both balanced and unbalanced sets.

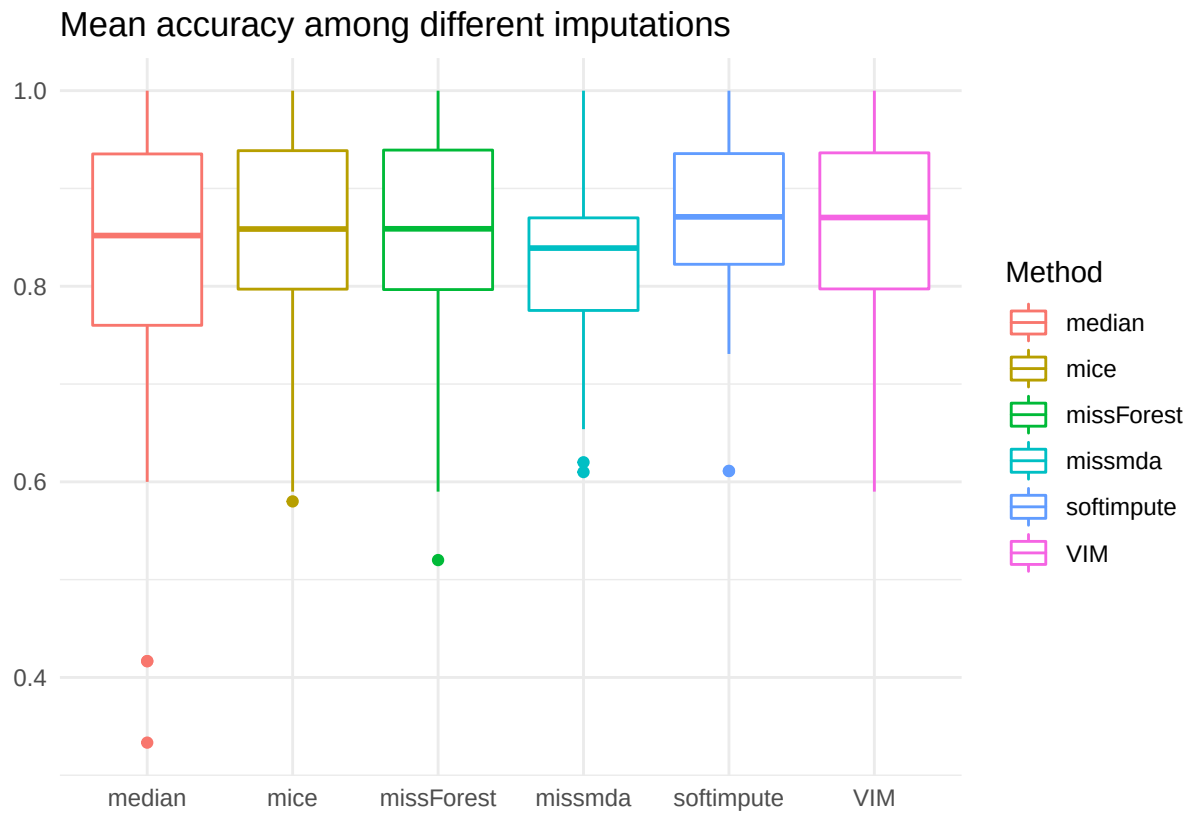
Table 1: Average scores of model on imputation methods

Method	Accuracy	F1
median	0.82	0.75
mice	0.85	0.77
missForest	0.84	0.75
missmda	0.82	0.80
softimpute	0.87	0.75
VIM	0.85	0.75

Our experiment definately did not find out the best imputation algoriithm, but we can derive some interesting conclusions from it. First of all, we are going to treat median imputation as kind of a baseline for our algorithms. Suprisngly it seems to perform quite well, amongst the others, often quite sophisticated methods. It achieved over 80% of accuracy on average. However, it is hard to decide whether it is a high score or not, because we are only able to compare algorithms between the others. To say anything more meanigful about our methods we shall look at the distribution of the scores in order to make a better analysis of performance.

Distributions of scores

Accuracy



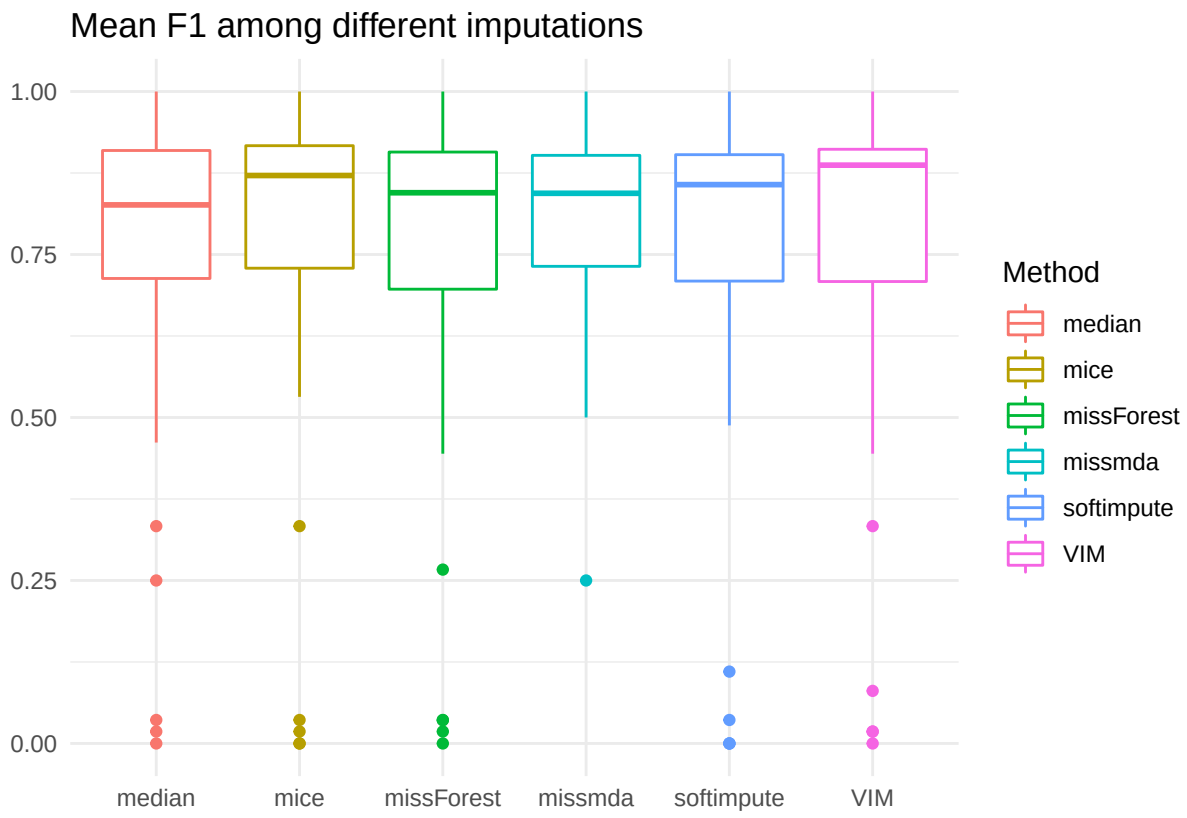
Taking a brief look at the distributions of accuracy score it seems quite unclear which algorithm performs the best. All medians seem to be aproximately on the same level and also first and third quartile of almost all of the plots have the same value. Only missmda is a bit lower than the others, but this is too early to derive any conclusions.

Let's check standard deviation of scores.

Method	acc_std
median	0.1508157
mice	0.1132738
missForest	0.1147474
missmda	0.1037811
softimpute	0.0887034
VIM	0.1105295

Deviation of scores seem to be quite equal for all of the methods so it will not be an issue.

F1



F1 scores give us much more informations. First thing that becomes apparent after looking at that plot is fact that we do have some very low values for every type of imputation. However this is simply because some of our datasetes were very small, saw neither of our algorithms were able to achieve recall above 0. Beside from that once again scores of all algorithms were quite close to each other. This is why we decide to use two different methods of comparing and classifying these algortihms, so that we will be able to choose our winner.

The same like before let's check standard deviation to ensure that our measurments will not be biased by it.

Method	F1_std
median	0.2511548
mice	0.2631994
missForest	0.2623674
missmda	0.1584238
softimpute	0.2802600
VIM	0.2875142

Once again scores are quite equal, but the raw numbers are a bit higher, purely because of the reasons that we described earlier, which was the problem of not balanced datasets.

Median baseline score

As a more reliable sample of imputation assessment we use to arrange them according to the application of the scheme. Let's take the result of each imputation method on each set. We define this result as the average of all machine processing algorithms released on data collected using imputation. Choose, we sort the result from the highest and we will assign algorithms points for each place. Finally, we add points and the method of imputation which points are the most important will win our ranking.

Method	wynik
missForest	41
softimpute	43
median	48
mice	51
VIM	61
missmda	63

After these comparisons we seem to have measure of algorithms perfromarnce. We cannot clearly compare by numbers, but the order which we present here may be quite useful to present how well algortihms work.

Method	wynik
missForest	36
softimpute	47
median	48
mice	54
VIM	56
missmda	61

And the same thing for F1 score. Because of the fact, that plenty of datasets we used were not balanced we decide to use F1 score as our main measure, because it is able to operate better on such datasets.

Best model by F1 measure

The process which we described earlier may be also quite useful to find the best classification alghoritm among tested. That process may

Algorithm	wynik
svm	27
rf	33
log_reg	37
xgb	45

Results which we gained are quite interesing, but we are afraid that some scores may be biased because of the fact that we had some very small datasets available in training which may have forced some algorithms to overfit. Nevertheless SVM scored the best with a reasonable margin above the others.

Second approach

As the second form of ranking imputation algorithms, we decided to use the following formula. We treat the median as the basic form of imputation and we will compare it to all other methods. We want to check how much the average prediction measured made by other algorithms differed from the median. As one prediction, we understand the average of all imputation methods

Method	score
missForest	0.0397342
softimpute	0.0134121
mice	-0.0415292
VIM	-0.1047341
missmda	-0.1123986

Result are quite simmilar to what we have achieved earlier, but here we can capture the difference which some algorithms have compared with median imputing. Difference can be even up to 0.1 in F1 score which is an considerable amount.

Summary and conclusions

Considering metrics scores achieved by evaluated model few important conlusions about imputation methods can be made. Basic approach with median and mode imputation allows to get decent results of machine learning model. Its disadvatage is large discrepancy in scores. More complex techniques available in packages, ensure more stable results of model among all data sets. From them, we can distinguish missFOREST, VIM irmi and mice. These packages have carried out the best imputation in terms of further model performance. They proved to be the most versatile, achiving the best average results of accuracy. VIM and missFOREST AUC scores show also the stability of model performance on imputed data set. SoftImpute package caused too low scores of recall and precision which suggests its poor performance on unbalanced data. MissMDA package performed worse than basic mode/median and all other methods, being able to proceed only on 6 from 11 data sets.
Second aspect to consider is time needed for imputation. MissFOREST achieved good metrics scores of model, but its very time consuming. Both VIM irmi and mice packages not only provided well imputed data sets but also were time efficient.
Choosing the best imputation tool may obviously depend on characteristic of the specific data sets, however our research should help to find flexible solution for everyday usage.

Bibliography

Batista, Gustavo E. A. P. A., and Maria Carolina Monard. 2003. "An Analysis of Four Missing Data Treatment Methods for Supervised Learning." *Applied Artificial Intelligence* 17 (5-6). Taylor & Francis: 519–33. https://doi.org/10.1080/713827181.

Bono, Christine, L. Ried, Carole Kimberlin, and Bruce Vogel. 2007. "Missing Data on the Center for Epidemiologic Studies Depression Scale: A Comparison of 4 Imputation Techniques." *Research in Social & Administrative Pharmacy : RSAP* 3 (April): 1–27. https://doi.org/10.1016/j.sapharm.2006.04.001.

Casalicchio, Giuseppe, Bernd Bischl, Dominik Kirchhoff, Michel Lang, Benjamin Hofner, Jakob Bossek, Pascal Kerschke, and Joaquin Vanschoren. 2019. *OpenML: Open Machine Learning and Open Data Platform.* https://CRAN.R-project.org/package=OpenML.

Hastie, Trevor, and Rahul Mazumder. 2015. *SoftImpute: Matrix Completion via Iterative Soft-Thresholded Svd.* https://CRAN.R-project.org/package=softImpute.

Josse, Julie, and François Husson. 2016. "missMDA: A Package for Handling Missing Values in Multivariate Data Analysis." *Journal of Statistical Software* 70 (1): 1–31. https://doi.org/10.18637/jss.v070.i01.

Kowarik, Alexander, and Matthias Templ. 2016. "Imputation with the R Package VIM." *Journal of Statistical Software* 74 (7): 1–16. https://doi.org/10.18637/jss.v074.i07.

Musil, Carol, Camille Warner, Piyanee Klainin-Yobas, and Susan Jones. 2002. "A Comparison of Imputation Techniques for Handling Missing Data." *Western Journal of Nursing Research* 24 (December): 815–29. https://doi.org/10.1177/019394502762477004.

Rubin, DONALD B. 1976. "Inference and missing data." *Biometrika* 63 (3): 581–92. https://doi.org/10.1093/biomet/63.3.581.

Stekhoven, Daniel J., and Peter Buehlmann. 2012. "MissForest - Non-Parametric Missing Value Imputation for Mixed-Type Data." *Bioinformatics* 28 (1). Oxford Univ Press: 112–18.

Su, Xiaoyuan, Taghi Khoshgoftaar, and Russ Greiner. 2008. "Using Imputation Techniques to Help Learn Accurate Classifiers." *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI* 1 (December): 437–44. https://doi.org/10.1109/ICTAI.2008.60.

van Buuren, Stef, and Karin Groothuis-Oudshoorn. 2011. "mice: Multivariate Imputation by Chained Equations in R." *Journal of Statistical Software* 45 (3): 1–67. https://www.jstatsoft.org/v45/i03/.