DE GRUYTER
OPEN

STUDIES IN LOGIC, GRAMMAR
AND RHETORIC 43 (56) 2015
DOI: 10.1515/slgr-2015-0041

Classification of Patients Treated for Infertility Using the IVF Method

Paweł Malinowski[1], Robert Milewski[1], Piotr Ziniewicz[1], Anna Justyna Milewska[1], Jan Czerniecki[2], Teresa Więsak[3], Allen Morgan[4], Dariusz Surowik[5], Sławomir Wołczyński[6]

[1] Department of Statistics and Medical Informatics, Medical University of Bialystok, Poland

[2] Department of Biology and Pathology of Human Reproduction, Institute of Animal Reproduction and Food Research of Polish Academy of Sciences, Olsztyn, Poland

[3] Department of Gamete and Embryo Biology, Institute of Animal Reproduction and Food Research, Polish Academy of Sciences, Olsztyn, Poland

[4] Shore Institute for Reproductive Medicine, Lakewood, USA

[5] The Chair of Logic, Informatics and Philosophy of Science, University of Bialystok, Poland

[6] Department of Reproduction and Gynecological Endocrinology, Medical University of Bialystok, Poland

Abstract. One of the most effective methods of infertility treatment is in vitro fertilization (IVF). Effectiveness of the treatment, as well as classification of the data obtained from it, is still an ongoing issue. Classifiers obtained so far are powerful, but even the best ones do not exhibit equal quality concerning possible treatment outcome predictions. Usually, lack of pregnancy is predicted far too often. This creates a constant need for further exploration of this issue. Careful use of different classification methods can, however, help to achieve that goal.

Introduction

It is estimated that infertility is a problem that affects approximately 15% of couples who wish to have a child (Radwan, 2011). This phenomenon is the subject of many research teams around the world. The most effective method of infertility treatment, and often the only one that gives hope for the success of fertilization, is the in vitro fertilization method (IVF). The effectiveness of this method of treatment has grown steadily in recent years, exceeding the level of 40% (Milewski et al., 2013a). But it is not a sufficient level to allow the use of the transfer of a single embryo, thereby reducing the risk of multiple pregnancy.

ISBN 978–83–7431–464–0 ISSN 0860-150X

An important step to increasing the effectiveness of infertility treatment is the ability to predict the results of treatment. This allows one to adjust the parameters of the treatment to the individual case depending on the prognosis, which in turn translates to a higher percentage of success. There are a number of studies, with the aim of creating predictive models, allowing for prediction of pregnancy during the treatment process.

Basic statistical methods often turn out to be insufficient in the analysis of this type of data. Therefore, data-mining classification methods are very popular. Interesting results can be obtained by using artificial neural networks (Milewski et al., 2009, 2013b). There are also reports on the application of data-mining analysis in reproductive medicine, such as the use of basket analysis (Milewska et al., 2011) and principal component analysis (Milewska et al., 2014). Interesting results have also been obtained by applying the concept of the closest neighborhood (Milewski et al., 2011). Some of the better results were obtained by applying the Random Forest algorithm, trying to predict the final outcome of a treatment even during its course (Malinowski et al., 2013). These and other concepts have been widely described by Malinowski et al. (2014).

The purpose of this study was to find a good classifier for data obtained from treatment using IVF. Following the previously obtained results and recommendations, the effectiveness of three classifiers – SVM, RkNN and RandomForest – was examined. Next, the effectiveness of the target classifier was examined. The second objective was to obtain a high quality classification for both possible outcomes of treatment.

Materials and Methods

The analyzed dataset was collected from 1,995 patients from an infertility treatment clinic in the USA. Pregnancy, defined by > 5 IU HCG/ml on 10–12 days after embryo transfer, was the dependent feature. There were also 26 independent features – 14 numerical, 11 categorical and 1 ordinal. Among numerical values were: age, number of oocytes retrieved and cultured, semen parameters and hormone levels. HCG dose to induce ovulation was an ordinal variable, and diagnosis, type of treatment and stimulation protocol were categorical ones.

Dataset analysis was performed in a few steps. Standard data recognition was performed in the first step. Dependent features were selected and their basic properties were checked, including their range and occurrence of any abnormalities. In addition, the rate of missingness was analyzed.

Due to danger of overfitting, all further algorithms, if possible, were run inside a cross-validation loop. The whole dataset was divided randomly into validation and learning parts (ratio 3:7). If it was possible, algorithms were trained and evaluated on the learning part, and the validation one was left for final algorithm evaluation.

The next phase after feature recognition was missing value imputation. Three algorithms were selected:

- standard imputation,
- "kNN" imputation,
- "missForest" (Stekhoven et al., 2012).

The standard algorithm impute values using basic statistics are: mean (for a continuous dependent variable), median (for an ordinal value) and mode (for a categorical value). The kNN imputation algorithm fills in missing data in the same way as the standard imputation algorithm, but using information obtained from k – a particular number of – nearest neighbors. The missForest algorithm iteratively builds a Random Forest (in classification or regression mode – depending on the type of feature) to fill in missing data.

Table 1. Tuned parameters – imputation algorithms

Algorithm	Tuned parameter	Description
"kNN" imputation	k	Number of neighbors
"missForest"	mtry	Number of randomly drawn features at each split inside tree
	ntree	Number of trees

The standard algorithm has no free parameters to tune, but the kNN imputation and missForest do (Table 1), and require some form of cross-validation. While it is possible to train the first two algorithms on observations different than those actually imputed, it is not so in the case of the latter one because of its iterative nature. Therefore, after choosing the best parameters, the "missForest" algorithm was performed using them on the whole dataset.

Tuning of "kNN" and "missForest" algorithms was performed by ranking their parameters against two measures, as defined by Oba et al. (2003):

- normalized mean root square error (NRMSE) – for continuous features,
- percent of false classified data (PFC) – for non-continuous features.

To calculate those values, some random part of the dataset was marked as missing, imputed, and then compared with the original dataset. The parameter set that minimalized the sum of rank was chosen as the best one.

If there was more than one such set, the minimum among the parameter(s) itself was chosen to obtain the final value. This rule of minimum-amongst-minimum ensures choosing the simplest possible algorithm.

After imputation, a classification was performed, also using three methods:

- SVM (Boser et al., 1992),
- Random Forest (Breiman, 2001),
- Random kNN (Li, 2009).

The first two algorithms were among the most powerful classifiers at the time of their introduction, and inspired a great deal of research. Even now, they are used as standard tools for classification. SVM builds a simple linear classifier in a transformed space of features by using an optimization procedure. By utilizing special additional constraint on the final form of that classifier, the possible space of solutions is greatly reduced, and, in most cases, only one such solution exists. Random Forest builds a set of decision trees, each on a separate bootstrap sample of data. Features to choose from are also selected randomly on each tree split. The decision of the whole forest is somewhat "democratic" in its nature – the final class is the one chosen by the majority of trees.

Random kNN is a randomized generalization of the kNN classifier algorithm. The algorithm builds a set of kNN classifiers, each trained on a random subspace of features from the original dataset. Similarly to Random Forest, the result of the whole set of kNNs is taken from the majority of votes. Among possible benefits over Random Forest, the algorithm uses kNN – a more stable and non-hierarchical algorithm (simpler) than a decision tree – as its base, and therefore does not require bootstrapping.

Each of the previously mentioned classification algorithms have free parameters to tune (presented in Table 2).

Even the best algorithms cannot help with data analysis without proper implementation. Whole analysis was performed in the R environment (R version 3.2.1 (2015–06–18) *"World-Famous Astronaut"*), with the packages presented in Table 3.

Some algorithms were written manually, including:

- cross-validation of imputation algorithms, including:
  - calculation of NRMSE and PFC over randomly removed data,
  - standard and kNN imputation over validation data,
- the random kNN algorithm,
- Manual Random kNN implementation in R, adapted directly for tune function from the e1071 package, took only 50 lines of code, which is another indicator of algorithm simplicity.

Table 2. Tuned parameters – classification algorithms

Algorithm	Tuned parameter	Description
SVM – RBF kernel	gamma	The single free parameter of kernel function
	cost	Cost of linear separability violation
Random Forest	mtry	Number of randomly drawn features at each split inside tree
	ntree	Number of trees
	nodesize	Minimum number of observations in terminal tree node
Random kNN	k	Number of nearest neighbors
	mtry	Number of randomly drawn features for each kNN
	r	Number of kNN trained

Table 3. Used packages

Package Name	Version	URL
e1071 (Meyer et al., 2014)	1.6–4	http://CRAN.R-project.org/package=e1071
VIM (Templ et al., 2015)	4.3.0	http://CRAN.R-project.org/package=VIM
randomForest (Liaw et al., 2002)	4.6–7	http://CRAN.R-project.org/package=randomForest
missForest (Stekhoven, 2013)	1.4	http://CRAN.R-project.org/package=missForest

Results

The analyzed dataset is presented in Figure 1. The first feature is a dependent one, then the next follows. Missing values are marked as black, while other values are gray-scaled. At the right side, there is a cross-validation indicator – observations marked as black (white) are training (validation) data. There were 598 observations in the validation dataset and 1,397 in the learning dataset.

Among 27 analyzed, missing values were in only 4 features, and comprised only 2.3% of the whole dataset. The cross-validation procedure for imputation randomly removed 2.5% of data in each step. In Table 4, ranges of imputation algorithm parameters are described. They were selected par-

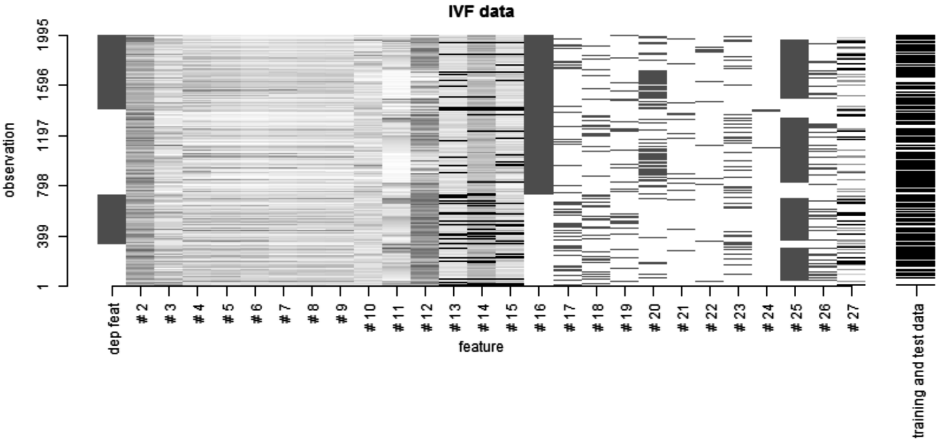

Figure 1. The dataset

Table 4. Tuned parameter range and result of imputation algorithms

Algorithm	Tuned parameters	Range	Best parameters		
			value	NRMSE	PFC
"kNN" imputation	k	1–100, by 1	21	0.344	0.138
"missForest"	mtry	1–12, by 1	10	0.309	0.118
	ntree	30–360, by 30	180		

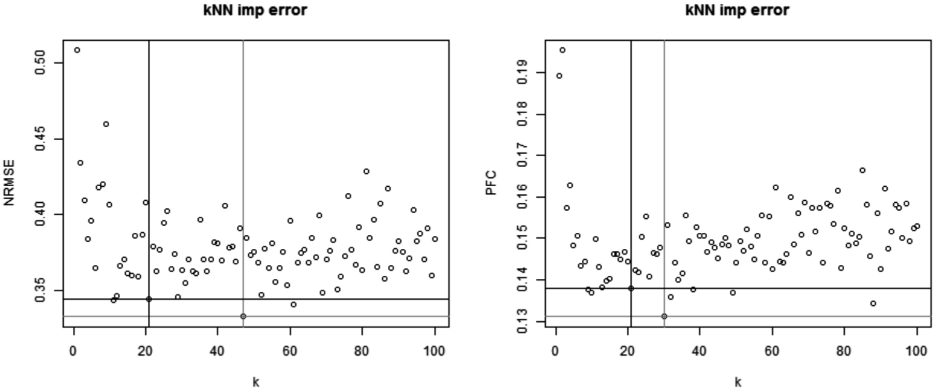

Figure 2. Results of tuning kNN imputation algorithm parameter

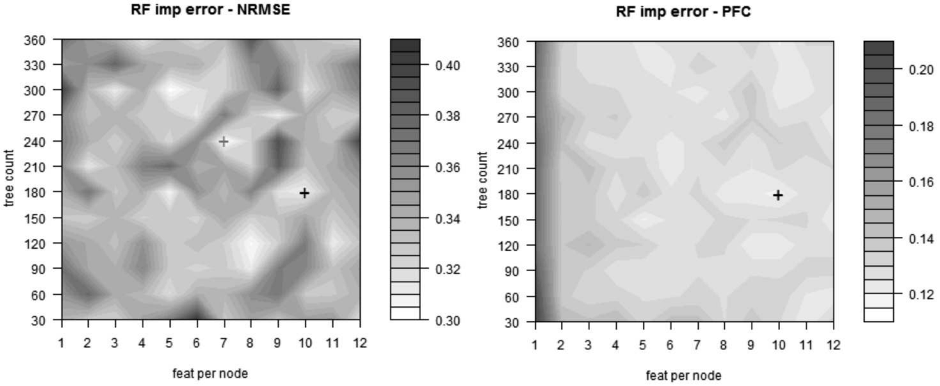

Figure 3. Results of tuning missForest imputation algorithm parameters

tially by each algorithm's authors' recommendation. The results of training are presented in Figure 2 and Figure 3. Gray lines (or '+' signs) indicate best values of NRMSE and PFC, and the black ones are those that minimized the sum of rank criterion. For Random Forest, both minima for PFC are for the same parameter set.

After obtaining parameters, imputation of the whole dataset was performed. Such prepared datasets were ready for classification. Again, using recommendations from the algorithm authors, parameters from Table 5 were chosen to tune.

Table 5. Tuned parameter ranges and results for classification algorithms

Algorithm	Tuned parameters	Range	Best parameters		
			value	imputation	error
SVM – RBF kernel	gamma	–20 to 21, by 1, on log2 scale	–18	"kNN"-based	0.413
	cost	–20 to 21, by 1, on log2 scale	17		
	gamma	–48 to –7, by 1, on log2 scale	–12	"kNN"–based	0.411
	cost	–48 to –7, by 1, on log2 scale	9		
Random Forest	mtry	1–13, by 1	2	missForest	0.364
	ntree	100–1700, by 100	700		
	nodesize	1–8, by 1	1		
Random kNN	k	1–13, by 2	1	missForest	0.408
	mtry	2–13, by 1	8		
	r (ntree)	101–1201, by 100	801		

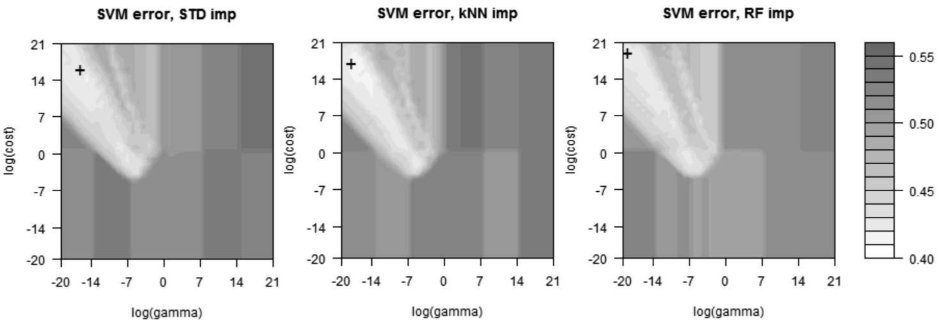

Figure 4. Results for SVM classifier – first parameter set

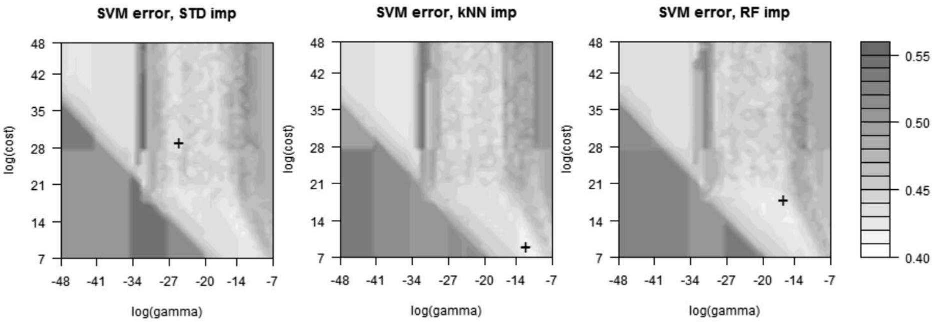

Figure 5. Results for SVM classifier – second parameter set

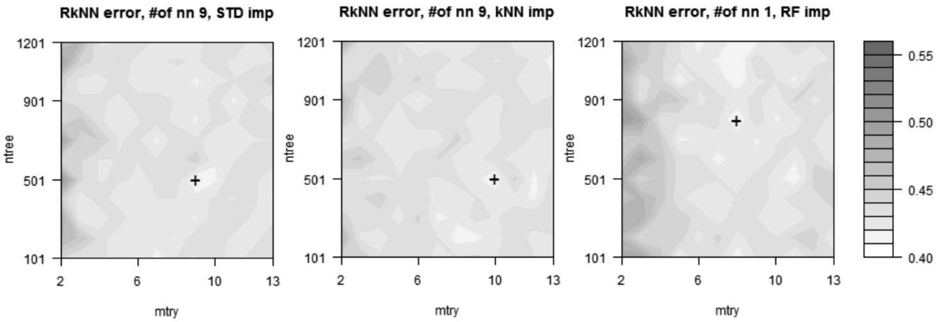

Figure 6. Results for RkNN classifier

Due to the curious pattern of SVM classifier performance (Figure 4), an additional range was selected for analysis to cover specific regions of parameter space (Figure 5). Rectangle areas appeared in both figures in places of rather poor classification quality. They are the result of random number generation that took place on different desktops (the whole calculation was

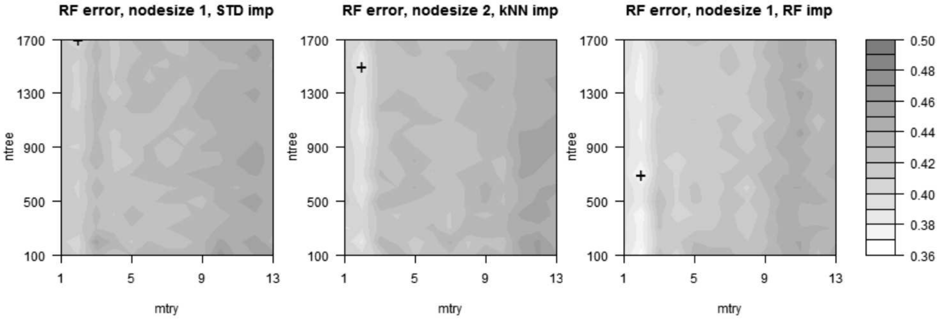

Figure 7. Results for Random Forest algorithm

performed on a cluster of computers). Due to randomness, results can be – and in many cases are – different in overlapping areas of figures, including the best one for SVM.

Results from the RkNN classifier were very similar among datasets obtained by different methods of imputation. In Figure 6, results for best number of neighbors are shown.

The Random Forest algorithm had a slightly lower error rate than other algorithms. In Figure 7, results for best nodesize are shown.

The best classifier algorithm was used again, this time for the validation part of the dataset. Results are shown in Table 6.

Table 6. Results for validation set

Results for validation set only		Predicted values		Correctness
		no pregnancy	pregnancy	
Empirical values	no pregnancy	162	126	0.563
	pregnancy	94	216	0.697
Correctness		0.633	0.632	0.632

Conclusions

SVM has a curious pattern of classification quality. Best classification was obtained using very flat kernel and high cost of misclassification. Despite simplicity, RkNN shows higher quality than SVM. This is another confirmation of the high predictive power of the simple nearest neighbor concept. Random Forest shows superior quality, compared to SVM and RkNN. Very similar results, obtained in the training and validation phases, illustrate the generalization power of the trained classifier. The most important result is almost equal predictive power for pregnancy and lack of it. Usually, lack of pregnancy is detected far more correctly than positive outcome of treatment. This very important result gives hope for the possibility of finding an even more effective classifier. By carefully studying its internal structure, it might even be possible to find relevant features determining the result of IVF treatment, which will be the next logical step for future analysis.

REFERENCES

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers, In D. Haussler (Ed.), *5th Annual ACM Workshop on COLT* (pp. 144–152). Pittsburgh, PA, USA: ACM Press.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32.

Li, S. (2009). *Random KNN Modeling and Variable Selection for High Dimensional Data.* (Doctoral Dissertations, West Virginia University). Available from Proquest, Umi Dissertation Publishing (AAI3381197).

Liaw, A., & Wiener, M. (2002). Classification and Regression by randomForest. *R News*, 2(3), 18–22.

Malinowski, P., Milewski, R., Ziniewicz, P., Milewska, A. J., Czerniecki, J., & Wołczyński, S. (2013). Classification Issue in the IVF ICSI-ET Data Analysis: Early Treatment Outcome Prognosis. *Studies in Logic, Grammar and Rhetoric. Logical, Statistical and Computer Methods in Medicine*, 35(48), 103–115.

Malinowski, P., Milewski, R., Ziniewicz, P., Milewska, A. J., Czerniecki, J., & Wołczyński, S. (2014). The use of data mining methods to Predict the Result of Infertility Treatment Using the IVF ET Method. *Studies in Logic, Grammar and Rhetoric. Logical, Statistical and Computer Methods in Medicine*, 39(52), 103–115.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., & Leisch, F. (2014). e1071: Misc Functions of the Department of Statistics (e1071), TU Wien (R package version 1.6–4). Retrieved from http://CRAN.R-project.org/package=e1071

Milewska, A. J., Górska, U., Jankowska, D., Milewski, R., & Wołczyński, S. (2011). The use of the basket analysis in a research of the process of hospitalization in the gynecological ward. *Studies in Logic, Grammar and Rhetoric. Logical, Statistical and Computer Methods in Medicine*, 25(38), 83–98.

Milewska, A. J., Jankowska, D., Citko, D., Więsak, T., Acacio, B., & Milewski, R. (2014). The use of principal component analysis and logistic regression in prediction of infertility treatment outcome. *Studies in Logic, Grammar and Rhetoric. Logical, Statistical and Computer Methods in Medicine*, 39(52), 7–23.

Milewski, R., Jamiołkowski, J., Milewska, A. J., Domitrz, J., Szamatowicz, J., & Wołczyński, S. (2009). Prognosis of the IVF ICSI/ET procedure efficiency with the use of artificial Neural networks among patients of the Department of Reproduction and Gynecological Endocrinology. *Ginekologia Polska*, 80(12), 900–906.

Milewski, R., Malinowski, P., Milewska, A. J., Czerniecki, J., Ziniewicz, P., & Wołczyński, S. (2011). Nearest neighbor concept in the study of IVF ICSI/ET treatment effectiveness. *Studies in Logic, Grammar and Rhetoric. Logical, Statistical and Computer Methods in Medicine*, 25(38), 49–57.

Milewski, R., Milewska, A. J., Czerniecki, J., Leśniewska, M., & Wołczyński, S. (2013). Analysis of the demographic profile of patients treated for infertility using assisted reproductive techniques in 2005–2010. *Ginekologia Polska*, 84(7), 609–614.

Milewski, R., Milewska, A. J., Więsak, T., & Morgan, A. (2013). Comparison of artificial neural networks and logistic regression analysis in pregnancy prediction using in the in vitro fertilization treatment. *Studies in Logic, Grammar and Rhetoric. Logical, Statistical and Computer Methods in Medicine*, 35(48), 39–48.

Oba, S., Sato, M., Takemasa, I., Monden, M., Matsubara, K., & Ishii, S. (2003). A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, 19(16), 2088–2096.

Radwan, J. (2011). Epidemiologia niepłodności. In J. Radwan, & S. Wołczyński (Eds.), *Niepłodność i rozród wspomagany* (pp. 11–14). Poznań, Polska: Termedia.

Stekhoven, D. J., & Bühlmann, P. (2012). MissForest – non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 1(28), 112–118.

Stekhoven, D. J. (2013). missForest: Nonparametric Missing Value Imputation using Random Forest (R package version 1.4). Retrieved from http://CRAN.R-project.org/package=missForest

Templ, M., Alfons, A., Kowarik, A., & Prantner, B. (2015). VIM: Visualization and Imputation of Missing Values (R package version 4.3.0). Retrieved from http://CRAN.R-project.org/package=VIM