VERSITA

STUDIES IN LOGIC, GRAMMAR AND RHETORIC 35 (48) 2013

DOI: 10.2478/slgr-2013-0034

Classification Issue in the IVF ICSI/ET Data Analysis: Early Treatment Outcome Prognosis

Paweł Malinowski[1], Robert Milewski[1], Piotr Ziniewicz[1], Anna Justyna Milewska[1], Jan Czerniecki[2], Sławomir Wołczyński[3]

[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 Polish Academy of Sciences, Olsztyn, Poland

[3] Department of Reproduction and Gynaecological Endocrinology, Medical University of Bialystok, Poland

Abstract. Infertility is a serious social problem. Very often the only treatment possibility are IVF methods. This study explores the possibility of outcome prediction in the early stages of treatment. The data, collected from the previous treatment cycles, were divided into four subsets, which corresponded to the selected stages of treatment. On each such subset, sophisticated data mining analysis was carried out, with appropriate imputations and classification procedures. The obtained results indicate that there is a possibility of predicting the final outcome at the beginning of treatment.

Introduction

Infertility is a problem that affects a growing number of couples that wish to have a child. Based on current statistics, approximately 18–20% of the couples in Poland suffer from infertility (Radwan, 2011). Currently, there are many known causes of infertility, including the crucial age of the woman (Milewski et al., 2008). In a significant proportion of cases, a direct cause of infertility cannot be determined, and the results of both women and men are in the norm, a so-called idiopathic infertility.

Depending on the identified causes, there are many treatments for infertility, but in many cases the only way to obtain offspring is by using In Vitro Fertilization (IVF) methods. In spite of constant improvement in Assisted Reproductive Technology (ART) that continues to enhance efficacy of the treatment, the pregnancy rate is still low and remains in the range of 40% (Milewski et al., 2013). Hence, there is a continuing need for an in-depth analysis of the data obtained in the treatment process, to find predictors for

ISBN 978–83–7431–392–6 ISSN 0860-150X

pregnancy and other factors that contribute to the next stages of treatment outcomes.

An extensive database was created in the Department of Reproduction and Gynecological Endocrinology at the Medical University of Bialystok, using dedicated software. The database covers the 6 year period from 2005 to 2010. Earlier studies analyzing the database focused mainly on a whole feature set. Those included classification alone (Milewski et al., 2012), selecting relevant features (Milewski et al., 2010) and using them to help classification (Milewski et al., 2011). This study focuses on classification by using subsets of features available at three selected phases of treatment. For comparison only, the same methodology was applied to the full set of features, available at the end of treatment. The purpose was to explore the possibility of successful classification with information available at the end of each selected phase of treatment. Like in the previous work (Milewskiet al., 2012), final results are shown for data not used in the learning phase, so that the results are as unbiased as possible.

Material and Methods

The analyzed database contains 1445 observations (single cycle of treatment) and 150 features (which include the outcome – pregnancy or no pregnancy), 22% of data is missing. Only about a third of the treatment cycles ended up inpregnancy. This high rate of missingness and relatively high outcome imbalance (2:1) can be linked to the nature of IVF ICSI/ET treatment.

Data analysis was carried out using the R environment (R version 3.0.1 (2013-05-16) "Good Sport"). There were used the packages presented in Table 1.

Table 1. Packages used

Package Name	Version	URL
e1071	1.6-1	http://CRAN.R-project.org/package=e1071
VIM	3.0.3.1	http://CRAN.R-project.org/package=VIM
randomForest	4.6-7	http://CRAN.R-project.org/package=randomForest
missForest	1.3	http://CRAN.R-project.org/package=missForest

On top of classification algorithm a cross-validation procedure was used. The whole dataset was divided randomly (on observations), at a 7:3 ratio,

to create a learning and validation part. This division was retained during further data split according to features. All algorithms were trained on the learning part only using k-fold cross-validation ($k = 10$). In almost all cases learning observations were partitioned into k subsamples. One of them was retained, while the rest were used for the training process for a specific algorithm and its set of parameters. The result of the training was then tested against the retained part, producing an error estimate. This process was repeated k times, and at the end, a single estimate of error was produced by averaging. This basic k-fold cross-validation procedure, implemented in package "e1071" (Meyer et al., 2012), was used for analysis.

For classification, a Random Forest (Breiman, 2001) algorithm was used, which proved to be one of the best in analyzing the database (Milewski et al., 2012). During the learning process, the algorithm builds a set of decision trees, based on available data. Each tree chooses, "gives a vote for", an observation class. Whole forest chooses the class with the majority of votes. Given N observations and M features, each tree is grown on samples of N observations selected with replacement (there are copies of observations). At each tree node some of the features are randomly selected (their number should be much smaller than M, square root by default for classification). The tree node splits data after the best possible split is found, but using previously selected features only as a criterion. By default each tree is built until the next split would result in an empty node (default for classification), but a minimum number of observations can be set in the terminal node.

The following parameters were selected for training random forest:

- number of trees (*ntree*),
- number of features at each split node (*mtry*),
- minimum number of observations in terminal node (*nodesize*).

These parameters were tuned using a grid search and 10-fold cross-validation. Random Forest algorithm, implemented in package "randomForest" (Liaw et al., 2002), which based on the original Fortran 77 implementation by Breiman, was used.

Since information on treatment outcome is not available before classification, unsupervised imputation procedures had to be used. This is the one of the main differences from the previous study (Milewski et al., 2012), where 2 of 3 used imputation methods were, in fact, supervised. Three, single-imputation algorithms were used before classification:

- a "standard" one,
- kNN-based,
- Random Forest based – "missForest".

A "standard" algorithm imputes missing values based on the mean (for numerical features), median (ordinal) or mode (categorical) of all other training observations. The kNN-based algorithm, implemented in the "VIM" package (Templ et al., 2013), tries to fill in missing data in a similar way to the standard algorithm, but utilizes only the k nearest observations – "the neighbors". Distances are calculated by using a version of Gover metric. The number of neighbors (k) is a free parameter.

The "missForest" algorithm (Stekhoven et al., 2012a), implemented in the package with the same name (Stekhoven et al., 2012b), starts with "standard" imputation. Then features are sorted by amounts of missing values, starting with the lowest amount. In the next step, an iterative procedure is used. For each feature, a random forest is trained, treating the selected feature as the dependent one and observations with filled values in that feature as the learning set. After training, missing values are imputed using predictions from the previously trained random forest. Depending on the type of feature (can be both – categorical or continuous), random forest for either classification or regression is used. After all features are imputed, the stopping criterion is evaluated, and depending on it the algorithm stops or continues to the next iteration. The stopping criterion is met as soon as the difference between the newly imputed data matrix and the previous one increases for the first time with respect to both variable types, if present (Stekhoven et al., 2012a).

A standard imputation procedure was used on the whole dataset, using the learning part only as a base. This was implemented manually, due to the lack of such a procedure in R.

For the "missForest" algorithm, the *ntree* and *mtry* parameters were chosen to tune. Note, that it is difficult to simply use a cross-validation on top of "missForest", due to its iterative nature. Instead, a ten step, pseudo cross-validation procedure was used to estimate error of imputation for each subset of tuned parameters. In each step, an additional 5% of the learning data were marked as missing, and the algorithm was trained on such data. After that, imputation error was calculated. For continuous variables, a normalized mean root square error ($NRMSE$) was used to calculate error. For categorical ones, a percent of false classified (PFC) data was used. Those measures were defined as follows (Oba et al., 2003):

$$NRMSE = \sqrt{\frac{mean\Big((X_{con,true} - X_{con,imp})^2\Big)}{\mathrm{var}(X_{con,true})}} \tag{1}$$

$$PFC = \frac{count(X_{cat,true} \neq X_{cat,imp})}{\#NA_{cat}} \tag{2}$$

where: $X_{con,true}$, $X_{cat,true}$ are original data with continuous (categorical) features; $X_{con,imp}$, $X_{cat,imp}$ are imputed data with continuous features, $\#NA_{cat}$ equals the number of added missing values

Note, that values originally missing are not counted in those measures (that is the missing values in the $X_{con,true}$, $X_{cat,true}$ datasets). Those two values were calculated at each step, and averaged at the end for each pair of *ntree* and *mtry*. To get an optimal pair, each such pair was ranked based on $NRMSE$ and PFC values separately. The pair that minimized the sum of ranks was chosen as the final one. If there were more such pairs, the one with less *ntree*, and at the end, less *mtry* (by minimizing those parameters, the algorithm became more simple, computational and logical) was chosen. After the best parameters were obtained, the whole dataset was imputed based on all data (again, this is the result of the iterative nature of this algorithm). This pseudo cross-validation procedure, and functions used to calculate $NRMSE$ and PFC (where the original dataset contains missing values, and we do not want to count them) were implemented manually due to lack of such procedures in R.

The same pseudo cross-validation procedure was used to tune the number of neighbors in the kNN imputation algorithm. This time, if there were two or more k, which minimizes the sum of ranks, the lesser one was chosen (by minimizing this parameter, the algorithm became more simple, computational and logical).

Data Preparation and Imputation

In further analysis, only features filled in more than 20% were used. Features containing only one value were also removed. After reduction, 108 features remained and only 5% of data were missing. The dataset was then divided into 4 sets corresponding to 4 selected treatment phases:

- medical history (data available before treatment),
- beginning of the treatment,
- gametes selection and moment of fertilization,
- from fertilization to embryo transfer.

Those 4 sets are presented in Figure 1. The first feature is the dependent one, then the features from the next sets follow. Each set is divided from the others by a black line. Missing values are marked as black, other values

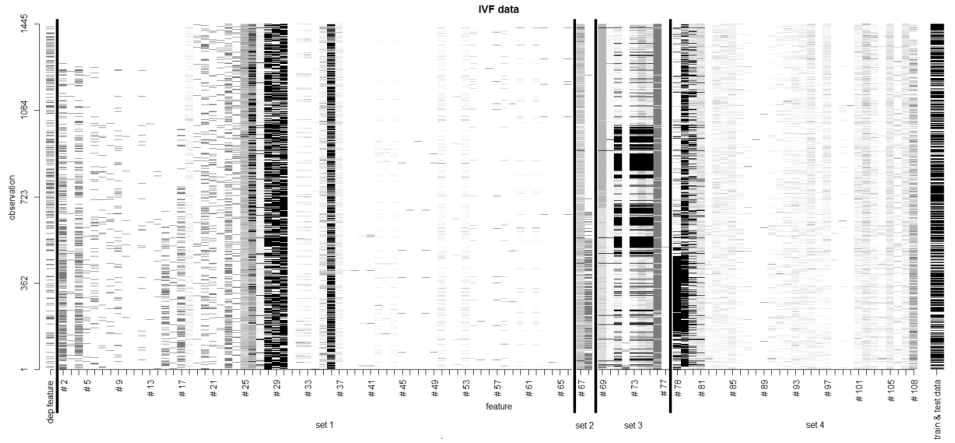

Figure 1. Division of the dataset

are gray-scaled. At the right side, there is a cross-validation indicator – observations marked as black (white) are training (validation) data.

Based on those four sets, four new datasets were built by utilizing the additive rule; that is, each new dataset contained the previous one and some additional features. Table 2 symbolically presents this (+ means, that given subset is present in dataset). This way, each new dataset contains information available at a chosen treatment phase.

Table 2. Datasets creation

New dataset id	Number of features	Sets			
		medical history	begin of the treatment	gametes selection and fertilization moment	from fertilization to embryo transfer
1	66	+			
2	68	+	+		
3	77	+	+	+	
4	108	+	+	+	+

After creation of the four subsets, they were imputed by three previously described procedures. During the pseudo cross-validation phase, the following ranges of parameters were tuned:

- kNN imputation: k (number of neighbors) in 1–100 range,
- missForest imputation: *ntree* in 30–240 by 30 range, *mtry* in 3–14 range.

Ranges for missForest were chosen partially according to recommendations (Stekhoven et al., 2012a). Results for kNN imputation are presented in Figure 2 (dataset 1 and 2) and Figure 3 (dataset 3 and 4). The gray lines present k optimizing specified measures. The black lines present k, which optimizes the sum of ranks – the final ones. Results for missForest imputation are presented in Figure 4 (dataset 1 and 2) and Figure 5 (dataset 3 and 4). Similar to earlier pictures, a gray '+' sign presents coordinates (pair *ntree* and *mtry*) optimizing specified measures (which are printed also). The black '+' presents a pair that optimizes the sum of ranks. If a gray mark (either for kNN or missForest) is not present, then it is the same as the black one (the set of parameters optimizing a particular measure is the same as the set optimizing the sum of ranks).

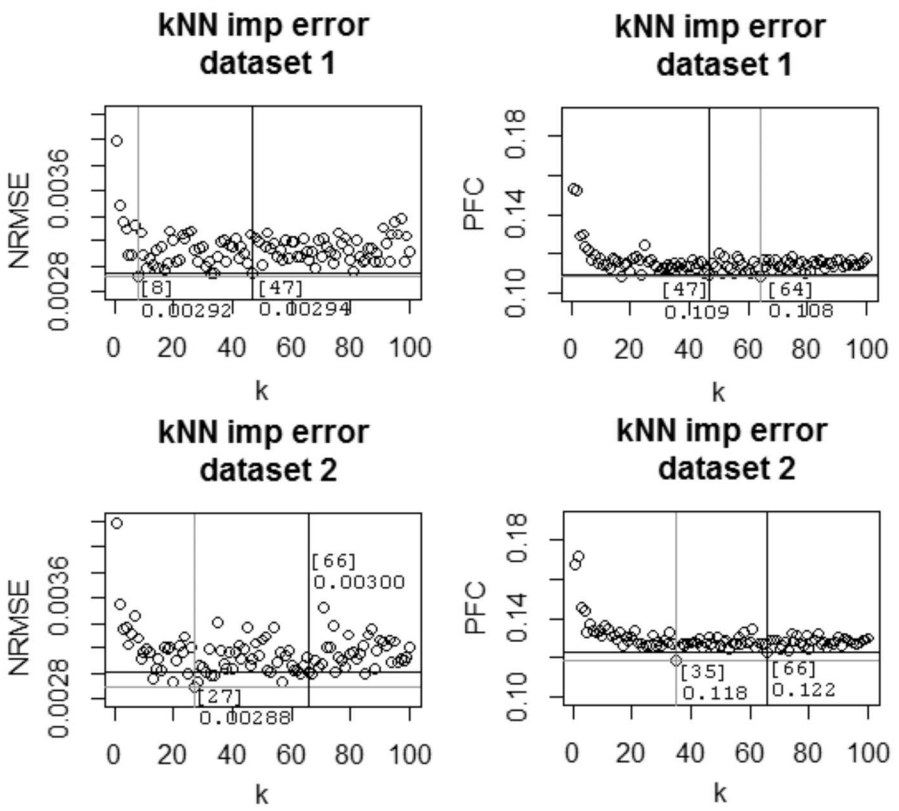

Figure 2. Results from pseudo cross-validation imputation for kNN method; datasets 1 and 2

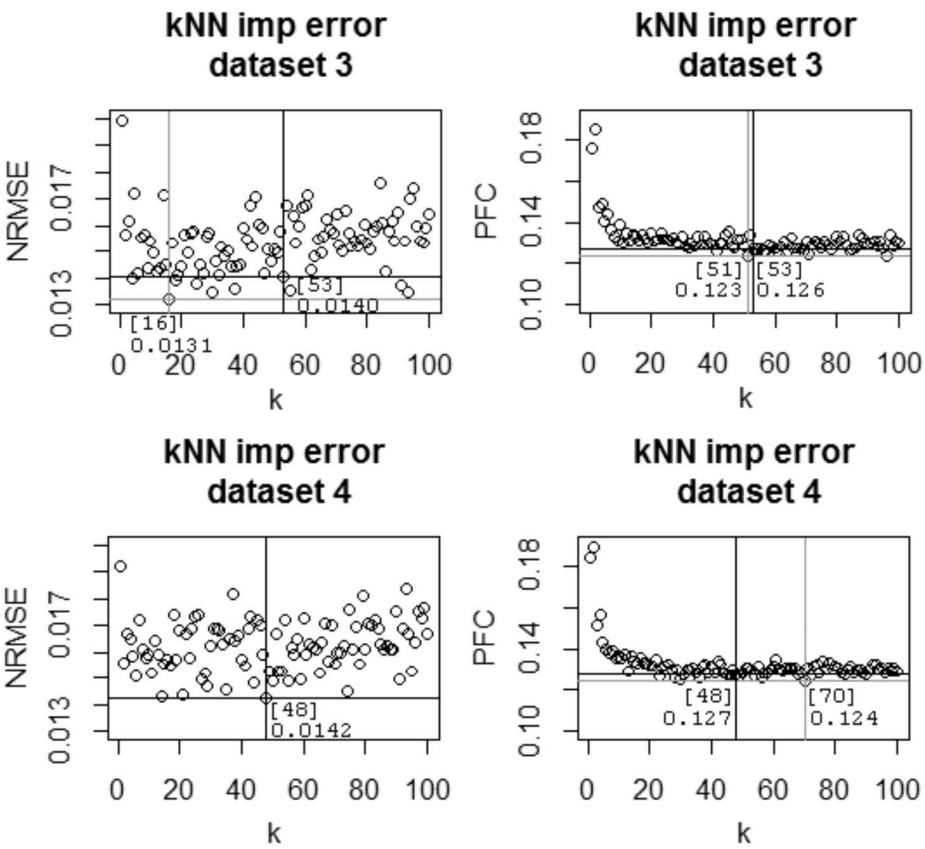

Figure 3. Results from pseudo cross-validation imputation for kNN method; datasets 3 and 4

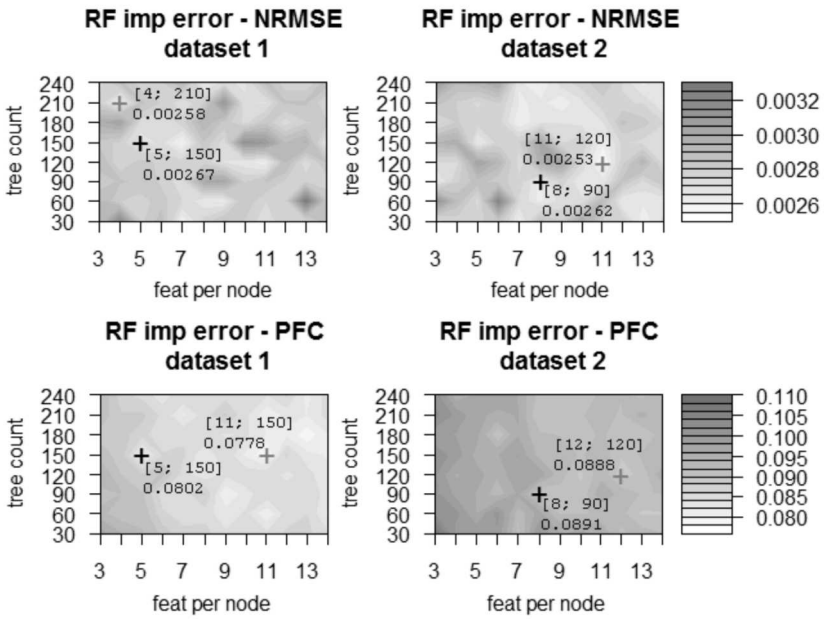

Figure 4. Results from pseudo cross-validation imputation for missForest method; datasets 1 and 2

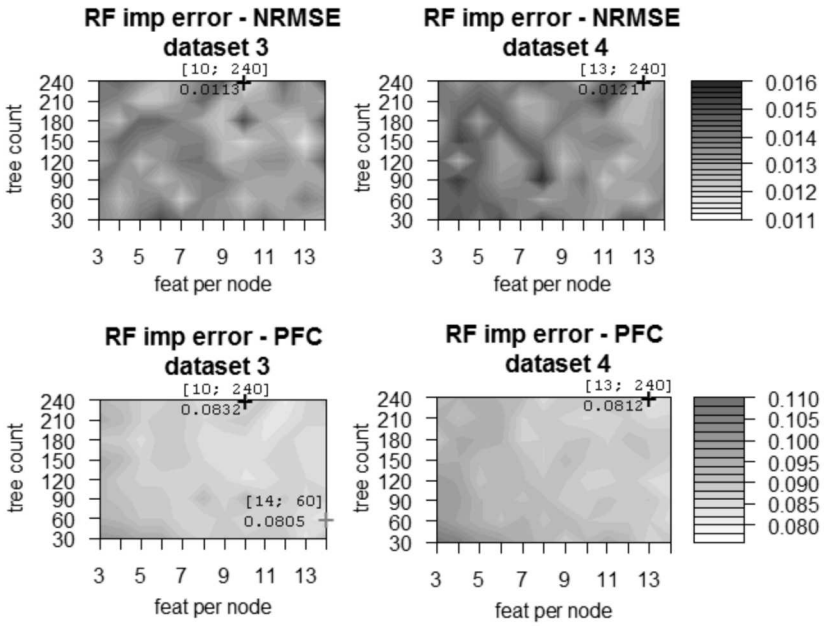

Figure 5. Results from pseudo cross-validation imputation for missForest method; datasets 3 and 4

Classification Results

After imputation, 12 datasets were created (the result of using 3 imputation methods on 4 datasets). Those datasets were classified using the Random Forest algorithm. During the cross-validation phase, the following ranges of parameters were tuned:

- number of trees: from 200 to 3000 by 200,
- number of features per node: from 2 to 20,
- minimum number of observations in node: form 1 to 10.

The specified range of parameters is based on recommendations (Breiman, 2001). The best classifier was tested on the validation dataset to produce an unbiased estimate of classification performance. Because it is difficult to visualize a 3-dimensional parameter set, results for best *nodesize* only are presented. Figure 6 presents results for datasets 1 and 2, which are also presented in Table 3 and Table 4 as full contingency tables for validation datasets and best mix of algorithms for those datasets. Figure 7 presents results for datasets 3 and 4, which are also presented in Table 5 and Table 6 as full contingency tables for validation datasets and best mix of algorithms for those datasets.

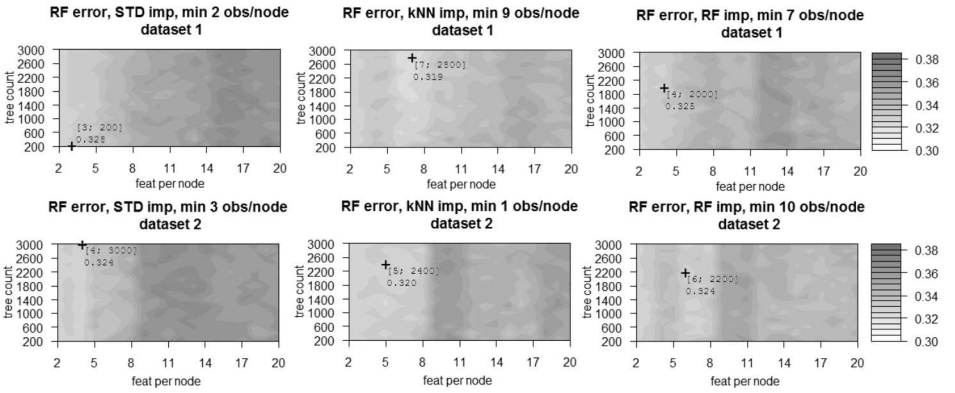

Figure 6. Results from cross-validation classification; datasets 1 and 2 imputed in 3 ways

Table 3. RF accuracy on kNN-imp validation dataset 1

Outcome prediction on kNN-imp validation observations dataset 1		Predicted outcome		Accuracy
		no	yes	
Observed outcome	no	274	7	0.975
	yes	145	8	0.052
Accuracy		0.654	0.533	0.650

Table 4. RF accuracy on kNN-imp validation dataset 2

Outcome prediction on kNN-imp validation observations dataset 2		Predicted outcome		Accuracy
		no	yes	
Observed outcome	no	275	6	0.979
	yes	142	11	0.072
Accuracy		0.658	0.647	0.659

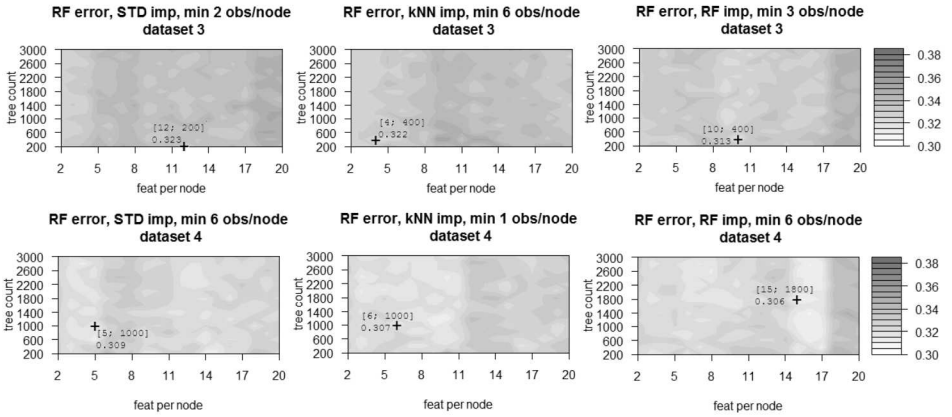

Figure 7. Results from cross-validation classification; datasets 3 and 4 imputed in 3 ways

Table 5. RF accuracy on RF-imp validation dataset 3

Outcome prediction on RF-imp validation observations dataset 3		Predicted outcome		Accuracy
		no	yes	
Observed outcome	no	267	14	0.950
	yes	137	16	0.104
Accuracy		0.661	0.533	0.652

Table 6. RF accuracy on RF-imp validation dataset 4

Outcome prediction on RF-imp validation observations dataset 4		Predicted outcome		Accuracy
		no	yes	
Observed outcome	no	268	13	0.954
	yes	127	26	0.170
Accuracy		0.678	0.667	0.677

Conclusions

Classification results in validation datasets differ from those obtained in k-fold cross-validation by 3–4 percent, which is a very good achievement. Since validation datasets were not used in the training of the classifier, this small difference guarantees error rate stability on future, unknown data. In the second selected phase of the treatment, accuracies for both possible outcomes are almost the same (about 2/3), with is also a very rare result. They are also comparable to results obtained from a full feature set. Treatment results can then be successfully predicted at the beginning of the actual treatment process, which is very important. Pregnancy prediction from medical history alone is far worse for the "yes" response, very near to 50% (the result ofa coin toss). Datasets 1 and 2 differ by only two features, so they should be very relevant to a successful outcome. The 3rd dataset contains a few new features compared to the 2nd, but they actually worse the "yes" response accuracy. The overall accuracies for the four datasets are almost the same, and worse by 12% than in the previous study (Milewski et al., 2012). This should be attributed to the previously mentioned change of imputation algorithms.

The analyzed database again proved to be resistive to successful classification even using state-of-the-art algorithms. However, this study shows that outcomes can be predicted in earlier phases of treatment and that predictions can be consistent in terms of success or failure. When comparing obtained results to previous, the focus should be on imputation algorithms. Since unsupervised and supervised methods are somewhat extreme paradigms, maybe some semi-supervised methods may improve classification results without compromising limited information availability at the beginning of treatment.

REFERENCES

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

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

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., & Leisch F. (2012). e1071: Misc Functions of the Department of Statistics (e1071), TU Wien. R package version 1.6-1. Retrieved from http://CRAN.R-project.org/package=e1071.

Milewski, R., Malinowski, P., Milewska, A. J., Ziniewicz, P., Czerniecki, J., Pierzyński, P., & Wołczyński S. (2012). Classification issue in the IVF ICSI/ET data analysis. *Studies in Logic, Grammar and Rhetoric*, 29(42), 75–85.

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*, 25(38), 49–57.

Milewski, R., Malinowski, P., Milewska, A. J., Ziniewicz, P., & Wołczyński, S. (2010). The usage of margin-based feature selection algorithm in IVF ICSI/ET data analysis. *Studies in Logic, Grammar and Rhetoric*, 21(34), 35–46.

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., Domitrz, J., & Wołczyński, S. (2008). In vitro fertilization ICSI/ET in women over 40. *Przegląd Menopauzalny*, 7(2), 85–90.

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: Termedia.

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

Stekhoven, D. J., & Bühlmann, P. (2012b). missForest: Nonparametric Missing Value Imputation using Random Forest R package version 1.3. Retrieved from http://CRAN.R-project.org/package=missForest.

Templ, M., Alfons, A., Kowarik, A. & Prantner, B. (2013). VIM: Visualization and Imputation of Missing Values. R package version 3.0.3.1. Retrieved from http://CRAN.R-project.org/package=VIM.