Classification issue in the IVF ICSI/ET data analysis

Robert Milewski[1], Paweł Malinowski[1], Anna Justyna Milewska[1], Piotr Ziniewicz[1], Jan Czerniecki[2,3], Piotr Pierzyński[4], Sławomir Wołczynski[4]

[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 in Olsztyn, Poland

[3] Department of Cytobiochemistry, Institute of Biology, University of Bialystok, Poland

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

Abstract. The effectiveness of infertility treatment using IVF ICSI/ET method depends on many different factors. Their identification and classification of individual cases remains a difficult task, despite the use of advanced statistical methods. This article presents the application of Random Forest and SVM classifiers, to analyze the data of patients undergoing the infertility treatment process.

Introduction

There are many methods referring to the general term “data mining methods” which can be used for the analysis of medical data, for example in [5] basket analysis was used on data of patients hospitalized in the gynecological ward. These methods are especially useful in the treatment of outcome prediction, like in [6, 11], were artificial neural networks have been used to predict the success of IVF ICSI/ET treatment. This article focuses on algorithms for classification in order to generate decision rules. Generated rules allow predicting the target class observations, also for new data. Since medical data are analyzed, those rules should have high efficiency and resistance to accidental errors and over-fitting. Therefore, only state-of-the-art classifiers are used: SVM and Random Forest, using their R language implementations. In contrast to [9], no feature selection algorithm is used. To counter over-fitting, cross-validation meta-algorithm was extensively used. Finally, three different imputation methods were used to fill missing data and make classifier work easier.

ISBN 978–83–7431–336–0 ISSN 0860-150X

The medical problem

Infertility is a social disease, which despite the intensive development of medical knowledge and advanced treatment techniques, still affects a significant percentage of couples. One of the contributing factors is postponing parenthood [12]. The chance for getting pregnant decreases with woman's age, mainly due to the decrease in the number and quality of oocytes. The effectiveness of infertility treatments, including the most advanced called In Vitro Fertilization with Intracytoplasmic Sperm Injection and Embryo Transfer (IVF ICSI/ET), is also correlated to woman's age, with success rates averaging at 10–15% pregnancies per treatment cycle in women of 40 and more years of age [10]. Then predictive methods allowing individual prognosis are needed. They could allow to select the best possible treatment approach and reduce the risk of complications.

Material and methods

Data for analysis were collected using the system of electronic registration of information about patients treated for infertility [7], with the statistical module based on artificial neural networks [11]. The system was designed to collect the specialist data, which significantly increased the accuracy and precision of the collected data, and led to increasing the number of recorded features. More recently, such systems have become more popular – they are dedicated to the specificity of the chosen medical unit, such as for instance the system to support clinical-research-teaching unit [14–15].

IVF ICSI/ET data were analyzed using methods which are accessible from the R software (http://www.R-project.org), an open source implementation of the computer language S – either by a native implementation or an interface to existing libraries. Some additional methods were implemented manually. In [Tab. 1] the corresponding R packages along with their version numbers are listed. R version 2.14.2 was used for the analysis.

Tab. 1. Used R packages and their versions

Package	Version	URL
e1071	1.6	cran.r-project.org/web/packages/e1071
randomForest	4.6–6	cran.r-project.org/web/packages/randomForest
VIM	3.0.1	cran.r-project.org/web/packages/VIM

The most frequently used meta-algorithm was cross-validation. The whole dataset was divided randomly to learning and validation part at 7:3 ratio. In order to learn a specified algorithm, a k-fold cross-validation ($k = 10$) was performed on learning data. Learning data was randomly partitioned into k subsamples. Of the k subsamples, a single subsample was retained as the test data for the model, and the remaining $k - 1$ subsamples were used as training data. The cross-validation process was then repeated k (folds) times, with each of the k subsamples used exactly once as the test data. The k results from the folds were then averaged to produce a single estimation.

Two classification algorithms were used in order to predict the outcome:

- Support Vector Machine (further referred as SVM)
- Random Forest (further referred as RF)

The SVM method [1] tries to build a hyperplane in parameter space that separates observations that belong to different classes. It is achieved by maximizing the margin, i.e. distance of hyperplane to nearest training observation of any class. Sometimes such hyperplanedoes not exist. SVM algorithm allows some violation of linear separation by using additional C (cost) parameter. SVM can be also modified to create a non-linear classifier via kernel trick, transforming original parameter space to other. This transformation may be nonlinear and resulted transformed space high dimensional; thus though the classifier is a hyperplane in the high-dimensional (even infinite-dimension) feature space, it may be nonlinear in the original input space. For analysis the Gaussian kernel was chosen with one parameter γ. Those two parameters – C and γ – were selected using grid search over wide range of values and 10-folf cross-validation to find the best-ones. SVM in R language is implemented in package "e1071" [3], and it is an interface to libsvm (version 2.6), and this implementation was used.

Random Forest algorithm [2], proposed by Leo Breiman and Adele Cutler, builds a set of decision trees based on learning data. Prediction of each tree is used as a sort of vote. Whole forest chooses class with majority of votes. Let N be the number of observation in training set, and M – the number of features. Each tree is grown as follows:

- pick up a sample of N observations at random with replacement – selected tree will be trained on this sample only
- at each node pick up *mtry* features (a number much smaller than M – a square root by default for classification) at random, and find the best split using those *mtry* features only
- grow each tree to full extent (this is also recommended for classification, but can be changed).

When full forest is grown, a version of distance matrix, called proximities, can be computed based on it. All the data are put down in each tree. If two observations are in the same terminal node, then their proximity is increased by one. The final result is normalized by dividing them by the number of trees.

Among many algorithm parameters which can be set for further tuning, the following were selected:

- number of trees,
- number of features at each node split,
- minimum number of observations per final node.

Those parameter were tuned using again grid search and 10-fold cross-validation. Random Forest in R language is implemented in package "randomForest" [4], based on original Fortran 77 implementation by Breiman, and this implementation was used.

Three algorithms were used for data imputation:

- a "standard" one
- kNN-based
- proximity-based

A "standard" algorithm imputes missing value based on mean (for numerical features), median (ordinal) or mode (categorical). It was partially implemented manually due to lack of cross-validation friendly version of this procedure in R language. The kNN-based algorithm tries to fill data in similar way to the standard algorithm, but utilizing only the part of observations. In given observation missing data are filled with values based on values from its k nearest neighbors only. Those neighbors are found by using version of Gover distance. This algorithm in R language is implemented in "VIM" package [13]. Proximity-based algorithm is also similar to the "standard" one. Algorithm, implemented in package "randomForest", runs as follows:

- fill missing data using "standard" method,
- run random forest on such data to find proximities,
- correct previously filled data based on calculated proximities. Use weighted mean (for numerical feature; weights are proximities) or the category with the largest average proximity (for categorical or ordinal data),
- calculate step 2 and 3 chosen (20) number of times.

Although standard imputation procedure is fairly simple, the next two have free parameters and required further tuning. For this purpose, 10-fold cross-validation procedure was applied on for each set of parameters. At each step additional 5% of filled test data were randomly marked as missing, and imputation algorithm with given set of parameters tried to fill it, based on train data observations. The objective was to minimize mean (relative) error of missing values prediction, based on 10 folds. After the best set of parameters was found (for second and third algorithm), the whole dataset was imputed based on train and test observations only, using those three methods.

Data preparation

The dataset has contained 1445 observations and 150 features. About 22% of original data was missing. Features containing more than 80% missing data were removed from dataset. Further investigation revealed features with only 1 level, which were also removed. Resulted dataset contained 108 features and only 5% of the data were missing. This dataset is symbolically depicted on [Fig. 1]. In the main part of [Fig. 1] black color means missing data and colors from white to gray means different levels of given feature. The first feature is the dependent one – treatment outcome. On the right side there is a barcode-like indicator of learning and validation division of the dataset. Black color indicates observation which was taken to the learning set, while white color is reserved for validation data.

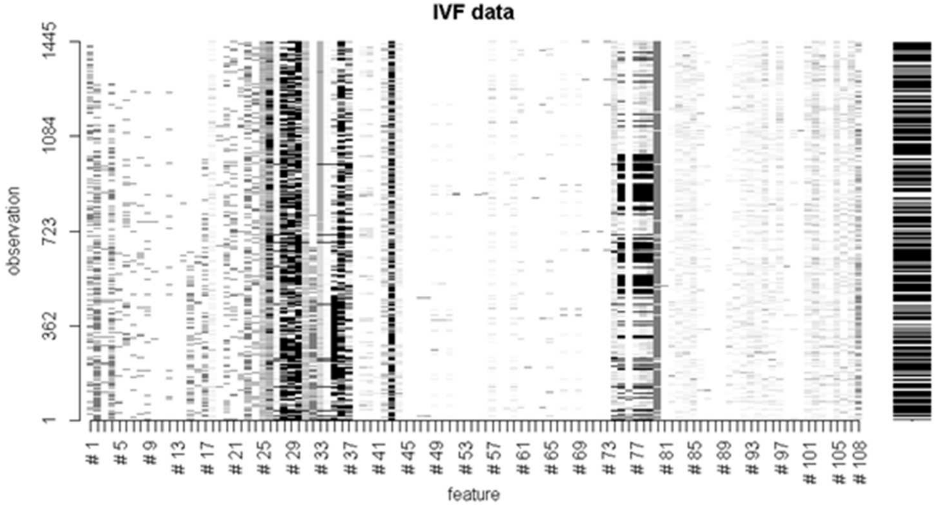

Fig. 1. Analyzed dataset with validation division

The next step was imputation of missing data. Three algorithms were chosen:

- “standard”one, further referred as “STD imp”
- kNN-based, further referred as “kNN imp”
  - k in 20–65 range
- proximity-based, further referred as “RF imp”
  - tree count in 1000–3000 by 200 range
  - *mtry* in 5–24 range

[Fig. 2] presents 10-fold cross-validation mean (relative) prediction error for kNN imp and RF imp procedures. Best found k equals to 42 at mean error slightly less than 30%. For proximity-based imputation procedure, the best set of parameters includes 2000 trees and 11 features at each split node at mean error around 25%.

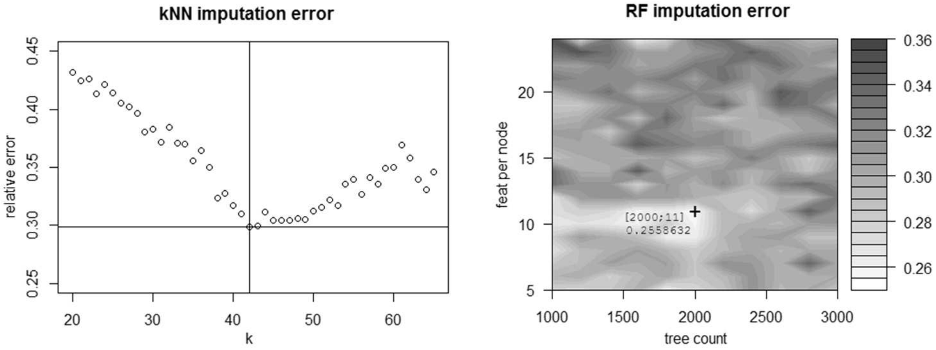

Fig. 2. kNN imputation cross-validation error

kNN-based imputation procedure had almost one and a half greater error range than proximity-based. After the best set of parameters was found, whole dataset was imputed based on learningdata only. This created 6 datasets, which were used for further analysis.

Data Classification

Two parameters of chosen SVM classifier – C and γ were tuned using grid search. Range of search was the same for both parameters:

$$range = \{2^{-20}, 2^{-19.8}, 2^{-19.6}, \ldots, 2^{0} = 1, \ldots, 2^{19.6}, 2^{19.8}, 2^{20}\}$$

SVM classifier was trained with each combination (out of 40201) of parameters using 10-fold cross-validation on learning datasets generated by 3 different imputation methods. In [Fig. 3] ean classification error is presented for those datasets in specified range of parameters with best model marked with (+).

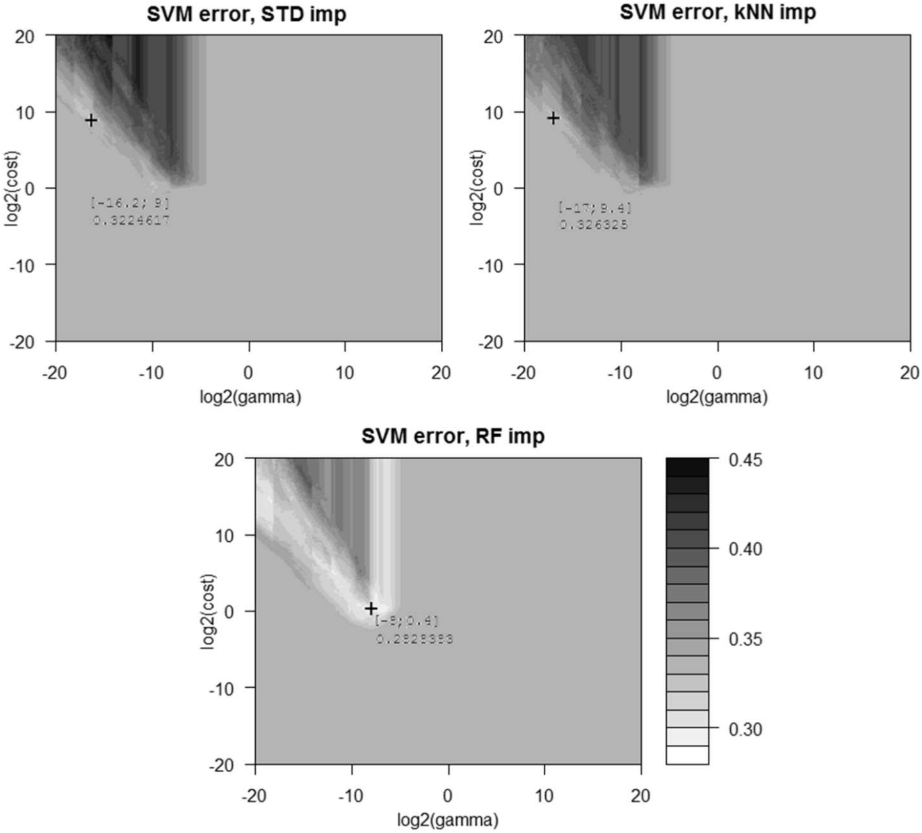

Fig. 3. SVM classifier cross-validation error

In wide range of parameters classification error was around 33%. This corresponds to outcome ratio in the whole dataset. In fact, for most of the parameters in the studied range, trained SVM classifier has predicted lack of pregnancy for all cases. This includes the “best” results on STD and kNN imputed datasets. Results on validation observations confirmed that behavior. Only on RF imputed dataset SVM yield different and superior result, which is shown in [Tab. 2].

Random Forest algorithm was trained using grid search on following parameters range

- number of trees
  $ntree = \{1000, 1200, 1400, \ldots, 2600, 2800, 3000\}$
- number of used features at each split node
  $mtry = \{5, 6, 7, \ldots, 22, 23, 24\}$
- minimal number of cases in tree terminal node
  $nodesize = \{1, 2, 3, 4, 5, 6, 7\}$.

Tab. 2. SVM accuracy on RF-imp validation dataset

Outcome prediction on RF-imp validation observations		Predicted outcome		Accuracy
		no	yes	
Observed outcome	no	261	27	90.6%
	yes	97	49	33.6%
Accuracy		72.9%	64.5%	71.4%

Again, 10-fold cross-validation procedure was used which each combination (out of 1540) of parameters on learning datasets generated by 3 different imputation methods, to find the best one. Because it is difficult to visualize 3-dimension parameter space, results for only two *nodesize* values will be presented per each imputation method, which gave minimum mean error. [Fig. 4] presents result for STD-imp learning dataset, which are also presented in [Tab. 3] as full contingency table for validation dataset.

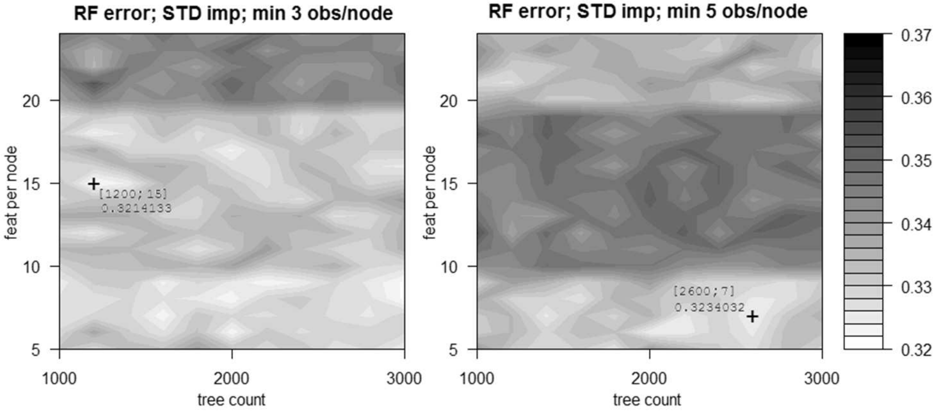

Fig. 4. RF cross-validation error on STD-imputed dataset

Tab. 3. RF accuracy on STD-imp validation dataset

Outcome prediction on STD-imp validation observations		Predicted outcome		Accuracy
		no	yes	
Observed outcome	no	266	22	92.3%
	yes	124	22	15.1%
Accuracy		68.2%	50%	66.4%

[Fig. 5] presents results for kNN-imp learning dataset, which are also presented in [Tab. 4] as full contingency table for validation dataset.

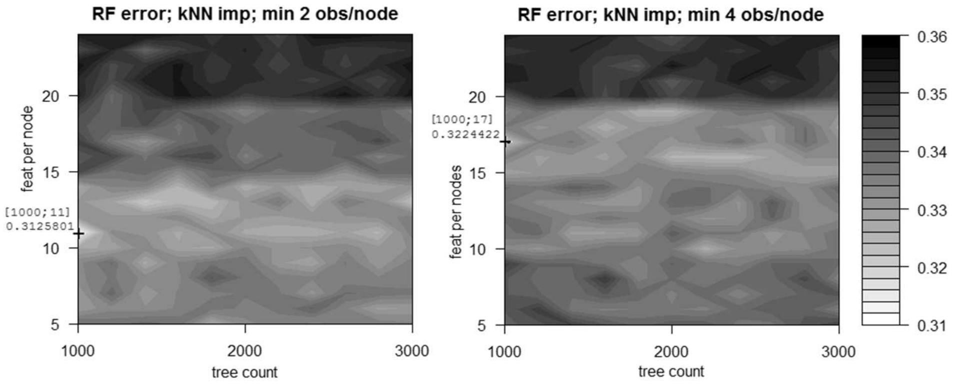

Fig. 5. Random forest mean error on kNN-imputed dataset

Tab. 4. RF accuracy on kNN-imp validation dataset

Outcome prediction on kNN-imp validation observations		Predicted outcome		Accuracy
		no	yes	
Observed outcome	no	265	23	92.0%
	yes	128	18	12.3%
Accuracy		67.4%	43.9%	65.2%

[Fig. 6] presents results for RF-imp learning dataset, which are also presented in [Tab. 5] as full contingency table for validation dataset.

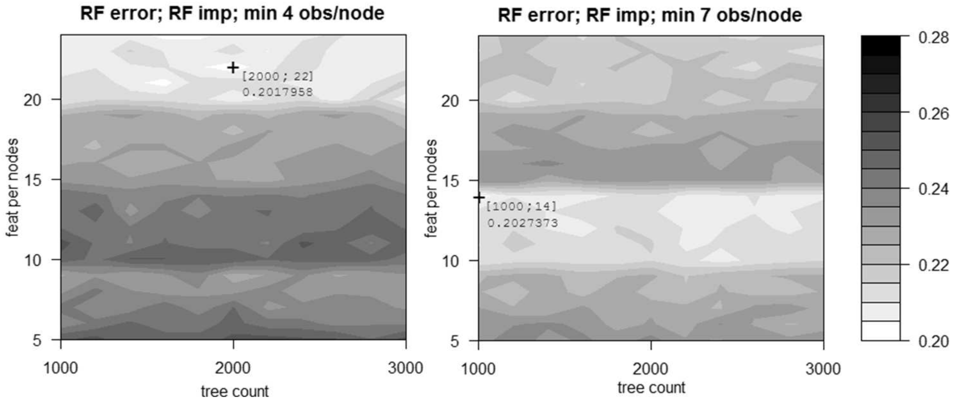

Fig. 6. Random forest mean error on proximity-imputed dataset

Tab. 5. RF accuracy on RF-imp validation dataset

Outcome prediction on RF-imp validation observations		Predicted outcome		Accuracy
		no	yes	
Observed outcome	no	227	61	78.8%
	yes	30	116	79.5%
Accuracy		88.3%	65.5%	79.0%

Conclusions

Proximity based imputation algorithm clearly outperforms other methods, despite relative error of prediction only 5% less than kNN-based one. It was only the imputation method, which yield sensible result with SVM classifier. SVM classifier performance was disappointing. For wide range of parameters this method predicted only lack of pregnancy. Although error rates for RF classifier on STD and kNN imputed datasets was similar to those obtained by SVM, the first algorithm actually tries to distinguish outcomes of observations. Cross-validation means errors for RF classifier was almost the same over the whole range of checked parameters. Relative differences reached 5 or 8 percent only. Change of *nodesize* parameter only slightly changed this error. It is worth noting that the RF classifier preferred higher *mtry* value on RF-imp dataset. Finally, the use of the RF classifier and RF-based imputation procedure leads to superior result: almost 80% accuracy on learning (note that this is mean accuracy based on 10 folds) and 79% on validation dataset. This error is unequally distributed among negative and positive outcome on the validation dataset. When the algorithm predicts lack of pregnancy, there is ∼88% probability, that this answer is a correct one, but for success this probability is only 65.5%. This behavior is consistent with previous studies [6, 8] on this dataset.

Further studies are required to find better algorithms for classification and imputation on IVF data. This article did not include feature selection algorithms; including them in analysis may also yield better results.

REFERENCES

[1] Boser B. E., Guyon I. M., Vapnik V. N., A training algorithm for optimal margin classifiers, In Haussler D. (editor); 5th Annual ACM Workshop on COLT, Pittsburgh, PA, ACM Press, pp. 144–152, 1992.

[2] Breiman L., Random Forests,Machine Learning, 45(1), 2001.

[3] Dimitriadou E., Hornik K., Leisch F., et al., Misc Functions of the Department of Statistics, TU Wien, R package version 1.6., 2011. http://CRAN.R-project.org/package=e1071

[4] Liaw A. and Wiener M., Classification and Regression by randomForest, R News, 2 (3), pp. 18–22, 2002.

[5] Milewska A.J., Górska U., Jankowska D., et al., The use of the basket analysis in a research of the process of hospitalization in the gynecological ward, Studies in Logic, Grammar and Rhetoric, 25 (38), pp. 83–98, 2011.

[6] Milewski R., Jamiołkowski J., Milewska A. J., et al., 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), pp. 900–906, 2009.

[7] Milewski R., Jamiołkowski J., Milewska A. J., et al., The system of electronic registration of information about patients treated for infertility with the IVF ICSI/ET method, Studies in Logic, Grammar and Rhetoric, 17 (30), pp. 225–239, 2009.

[8] Milewski R., Malinowski P., Milewska A.J., et al., Nearest neighbor concept in the study of IVF ICSI/ET treatment effectiveness, Studies in Logic, Grammar and Rhetoric, 25 (38), pp. 49–57, 2011.

[9] Milewski R., Malinowski P., Milewska A.J., et al., The usage of margin-based feature selection algorithm in IVF ICSI/ET data analysis, Studies in Logic, Grammar and Rhetoric, 21 (34), pp. 35–46, 2010.

[10] Milewski R., Milewska A.J., Domitrz J., et al., In vitro fertilization ICSI/ET in women over 40, Przegląd Menopauzalny, 2(36), pp. 85–90, 2008.

[11] Milewski R., Milewska A.J., Jamiołkowski J., et al., The statistical module for the system of electronic registration of information about patients treated for infertility using the IVF ICSI/ET method, Studies in Logic, Grammar and Rhetoric, 21(34), pp. 119–127, 2010.

[12] te Velde E.R., Pearson P.L., The variability of female reproductive ageing, Human Reproduction Update, 8 (2), pp. 141–154, 2002.

[13] Templ M., Alfons A., Kowarik A. and Prantner B., VIM: Visualization and Imputation of Missing Values. R package version 3.0.1. http://CRAN.R-project.org/package=VIM, 2012.

[14] Ziniewicz P., Malinowski P., Milewski R., et al., Clinical department information system's internal structure, Studies in Logic, Grammar and Rhetoric, 25 (38), pp. 191–200, 2011.

[15] Ziniewicz P., Malinowski P., Mnich S. Z., et al., Clinical department information system development, Studies in Logic, Grammar and Rhetoric, 2 (34), pp. 129–142, 2010.