Empirical Assessment of Performance Measures for Preprocessing Moments in Imbalanced Data Classification Problem

Pawel Szeszko, Magdalena Topczewska

Empirical assessment of performance measures for preprocessing moments in imbalanced data classification problem

Paweł Szeszko and Magdalena Topczewska

Bialystok University of Technology, Faculty of Computer Science
p.szeszko@gmail.com, m.topczewska@pb.edu.pl

Abstract. The article concerns the problem of imbalanced data classification, when classes, into which elements belong, are not equally represented. In the classification model building process cross-validation technique is one of the most popular to assess the efficacy of a classifier. While over-sampling methods are used to create new objects to obtain the balance between the number of objects in classes, inappropriate usage of the preprocessing moment has a direct impact on the achieved results. In most cases they are overestimated. To present and assess this phenomenon in this paper three preprocessing techniques (SMOTE, Safe-level SMOTE, SPIDER) and their modifications are used to make new elements of data sets to balance cardinalities of classes, and two classification methods (SVM, C4.5) are compared. k-folds cross-validation technique (k=10) considering two moments of preprocessing approaches is performed. The measures as precision, recall, F-measure and area under the ROC curve (AUC) are calculated and compared.

1 Introduction

The problem of imbalanced classes in datasets is deemed to be one out of ten the most important data exploration tasks nowadays. Therefore more and more scientists focus on developing methods that can provide the improvement of model performance for such difficult data. Many techniques to deal with the problem have been proposed [4, 8].

The issue appears when the cardinality of at least one class is lower than the cardinality of remaining classes. In the two classes task the *imbalance ratio* describes the proportion of the majority (*negative*) class objects to the minority (*positive*) class and can vary depending on a dataset.

Classification models created on the basis of such data have to cope with misclassification errors of the minority class. This is the consequence of the fact that many classifiers consider the misclassification errors equally for all objects, regardless of the class to which they belong. However, in such cases the costs of incorrectness vary markedly. Considering for instance cancer diseases, the number of ill patients can be significantly smaller than the healthy or the sick for other reasons. Another well-known example is fraud detection, when a fraud occurs relatively infrequently. In mentioned situations the great emphasis is placed

on the correct classification of minority class objects, definitely greater than in the majority class elements.
To reduce the impact of imbalance between the number of objects in particular classes, three groups of approaches have been proposed: data level, algorithm level and cost-sensitive level.
The algorithms belonging to the first category – data-level approach – operate on data objects solely and they are independent from the classification models. In undersampling methods superfluous objects are removed, thus the subset of original dataset is created to be used in model building process. In oversampling methods existing objects, from positive class in particular, can be replicated or new objects can be created. The third approach is constituted by hybrid methods – superfluous objects are removed and new necessary elements are created.
The second group describes algorithm-level approaches in which instead of data, existing model-building algorithms are adjusted to classify objects regardless of the disparity between cardinalities of classes.
Finally, the last category considers unequal misclassification costs, because false negative and false positive costs are not the same as in cancer patients example. An important issue here is to determine the cost ratio or the cost matrix. The aim of cost-sensitive level methods is to build models with minimum misclassification costs.
By creating new objects, replicating or removing existing ones, the number of objects as well as the distributions of classes in a dataset are changed. Then to proceed the classification process and assess the efficacy of classification methods, one has to be cautious. In certain articles, the authors do not apply data preprocessing in appropriate moment while using cross-validation techniques, and they test the performance of the model on the artificial examples as well. It can lead to the overestimated results of the classification.
The aim of this paper is to compare the results obtained for two different moments of imbalanced datasets preprocessing and their influence on the classification performance measures as well as to assess empirically the overestimation level. Three data-level approaches using k-nearest neighbour technique [5] are considered: SMOTE [3], Safe-level SMOTE [2] and SPIDER [10] and two classifiers are used: C4.5 and SVM using John Platt's sequential minimal optimization algorithm for training a support vector classifier.

2 Data preprocessing

In this paper three approaches to the construction of classifiers from imbalanced data sets are compared: SMOTE, Safe-level SMOTE and SPIDER.

2.1 SMOTE

The SMOTE (Synthetic Minority Over-sampling Technique) [3] is the most versatile method and enables to create new objects of the minority class. For each

p positive object, among its k nearest neighbours also belonging to the positive class, $N/100$ objects are sampled randomly with replacement, where N is a number of objects to generate. To compute the position of a new element, the difference between feature vector values of the considered object and a chosen neighbour should be calculated and multiplied by a *gap* – a random number between 0 and 1. The new object is then generated by adding that result to the feature vector of an examined p object.

2.2 Safe-level SMOTE

Safe-level SMOTE [2] is a modification of the original SMOTE algorithm. It draws particular attention to the classes of objects that encircle examined positive objects. For each examined positive p element the safe-level indicator (sl_p) is defined and calculated as the number of positive instances among its k nearest neighbours. Additionally, one randomly selected neighbour is chosen (n), and for this element safe-level indicator (sl_n) is also computed. In the next step *safe level ratio* ($sl_{ratio} = \frac{sl_p}{sl_n}$) is calculated and one of five cases is chosen:

- if $sl_{ratio} = \infty$ and $sl_p = 0$, no new object is generated, because p and n are treated as noisy, i.e. elements situated in the negative class area;
- if $sl_{ratio} = \infty$ and $sl_p \neq 0$, n is noisy; p object is replicated; so $gap = 0$;
- if $sl_p = sl_n$, then $sl_{ratio} = 1$ and gap is a random number between 0 and 1; new object will be situated between p and n;
- if $sl_{ratio} > 1$, p object is safer than n object, because in its surroundings there are more safer positive objects, hence gap is a random number between 0 and $\frac{1}{sl_{ratio}}$ to situate a new object closer to p;
- if $sl_{ratio} < 1$, n object is safer than p object, hence gap is a random number between $1 - sl_{ratio}$ and 1 to situate a new object closer to n.

The number of iterations is matched to balance the cardinalities of classes.

2.3 SPIDER

The selective preprocessing algorithm SPIDER [10] combines local over-sampling of the minority class with filtering difficult objects from the negative class. The method assigns to each object one of two labels: safe – if its classification result using k-nearest neighbour rule ($k = 3$) is correct, or noisy otherwise. The special D set is created and all noisy majority class objects are transferred into it to remove them from a dataset at the end of the algorithm. Three techniques of a dataset modification can be mentioned.
Weak amplification method increases the importance of the minority class objects labelled as noisy by their replications. The number of replications depends on the number of safe objects among three neighbours of each examined element. Additionally, all noisy majority class elements are removed from a dataset.
Weak amplification and relabelling adds an adjective step to the previous method. For elective noisy positive object, some among its three negative noisy neighbours

are relabelled as positive elements. Modified negative elements are removed from the D set.
The third method is called *strong amplification* and is focused on all positive class objects. The importance of each safe element is increased by its replication – the number of copied objects equals the number of negative objects among its nearest neighbours. The modification of noisy positive elements depends on the k-nearest neighbours classification results when $k = 5$. In the correct result case, the number of replications is the same as for safe objects, otherwise instead of three, five nearest neighbours for each examined element are taken into considerations.

3 Classification model performance

Predictive model creating is one of the main goals of machine learning process. Models built using existing elements from a dataset should give an answer which class a new unclassified object should be assigned to. To compare model building algorithms we need methods that can predict performance of a model for objects that were unused in a training process, because the model is fitted to the data on which it was constructed. Therefore training and test sets should be independent and it is very necessary issue if we try to assess performance of the classification models in real situations.

3.1 Estimating the predictive model performance

For the purpose described in a previous paragraph a cross-validation method was proposed. It divides the input set into independent training and test sets.
If the input dataset is large, it can be just randomly divided into two parts: training and test sets, for example in proportion 2:1. It is called a *hold-out method*. The main advantage of this approach is low cost of computing resources. A big disadvantage is uncertainty that the class distributions are represented in both parts properly, hence received results may not present a real efficacy of a model. This problem can be solved with repeated random division of the input dataset, but still there is no certitude that appropriate elements have been chosen to the test sets.
An approach that should provide proper representation of objects in a test set is k-fold cross-validation technique [7]. This method divides the input set into k parts. In each of k iterations k-th part is used as a test set and other objects are used for a model training. In advance of a division the order of the objects should be changed randomly. Additionally, the stratification process can be performed to preserve the distributions of classes in training and test tests.
In this paper 10-folds cross-validation process has been chosen for calculations.

In the class imbalance problem, when data need preprocessing, there is another issue to reflect – the moment of preprocessing. It might seem to be correct to prepare datasets first and then initiate classification process, as presented in the Figure 1. However, it is not appropriate mode of an action, since new

artificial objects will take part in testing process, and the results will be affected directly [11]. Due to that fact the appropriate approach is to run preprocessing algorithms for each training set internally during cross-validation process to not include synthetic elements into test sets, as shown in the Figure 2.

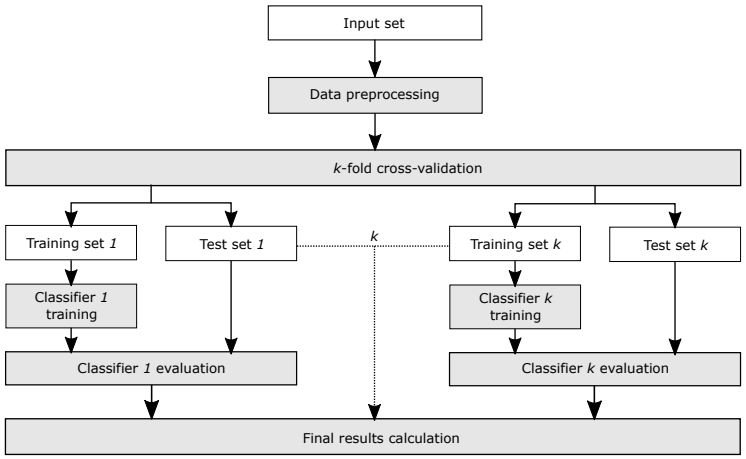

Fig. 1: Incorrect cross-validation process

3.2 Classification quality assessment methods

Comparison methods of various classification techniques constitute an important element of the machine learning process. In the case of two classes problem, the elements of the class that is the object of an interest are denoted as *positive*, while remaining objects are denoted as *negative*.
We use the following notation:

- TP (true positives) – a number of *positive* objects classified correctly
- TN (true negatives) – a number of *negative* objects classified correctly
- FN (false negatives) – a number of *positive* objects classified incorrectly as *negative*
- FP (false positives) – a number of *negative* objects classified incorrectly as *positive*

These four terms form the cells of the confusion matrix. Additionally, on the basis of them various measures are built.

3.3 Performance measures

The most popular and well-known measures describing the performance of a classifier are the *accuracy* Q (1) computed as the proportion of the objects

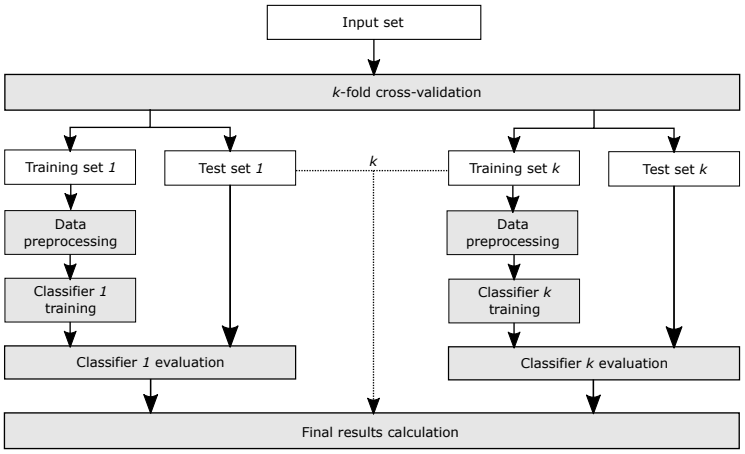

Fig. 2: Correct cross-validation process

classified correctly:

$$Q = \frac{TP + TN}{TP + FP + TN + FN} \ , \tag{1}$$

and the *error rate* Err_{rate} (2) respectively, as the proportion of misclassified objects:

$$Err_{rate} = \frac{FP + FN}{TP + FP + TN + FN} \ . \tag{2}$$

These measures are not appropriate for the case of the imbalanced class problem, when the error weights of the misclassified objects belonging to the minority class are not equal as for the majority class or the remaining classes. To comprehend of such a problem *precision* (3) and *recall* (4) have been introduced:

$$Precision = \frac{TP}{TP + FP} \ , \tag{3}$$

$$Recall = Sensitivity = TP_{rate} = \frac{TP}{TP + FN} \ . \tag{4}$$

Furthermore, there is *F-measure* that is defined as the harmonic mean of precision and recall

$$F = \frac{(1 + \beta^2) \cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall} \ , \tag{5}$$

and AUC is the the area under the ROC curve [9]

$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2} \ , \tag{6}$$

where $FP_{rate} = \frac{FP}{FP+TN}$.

4 Experiment

The experiment concerning the influence of the data preprocessing moment on the final results of the classification methods has been performed. 25 datasets from KEEL [1] repository were selected and their characteristics is presented in the Table 1. These datasets were created on the basis of 10 datasets from UCI repository [12].

Calculations have been performed using classes as parts of the Weka data min-

Table 1: Characteristics of the datasets

dataset	#objects	#attributes	#minority	IR
ecoli-0vs1	220	7	77	1.86
ecoli1	336	7	77	3.36
ecoli2	336	7	52	5.46
ecoli3	336	7	35	8.60
ecoli4	336	7	20	15.80
glass*	214	9	51	3.20
glass0	214	9	70	2.06
glass1	214	9	76	1.82
glass2	214	9	17	11.59
glass4	214	9	13	15.46
glass5	214	9	9	22.78
glass 6	214	9	29	6.38
haberman	306	3	81	2.78
iris0	150	4	50	2.00
new-thyroid1	215	5	35	5.14
new-thyroid2	215	5	35	5.14
pima	768	8	268	1.87
vehicle0	846	18	199	3.25
vehicle1	846	18	217	2.90
vehicle2	846	18	218	2.88
vehicle3	846	18	212	2.99
vowel0	988	13	90	9.98
wisconsin	683	9	239	1.86
yeast1	1484	8	429	2.46
yeast3	1484	8	163	8.10

ing software [6] and own application. Precision, recall, F-measure and AUC have been calculated for two moments of preprocessing and for two classification algorithms and ten preprocessing variants. To reduce the randomness of the results each 10-folds cross-validation process was repeated 10 times with different random number generator seed. Next, the differences between two points were calculated and examined. As modification of SMOTE algorithm also the number of synthetic objects was verified.

The numbers in the SMOTE$100, 200, \ldots, 500$ versions denote % of the minority

class elements that were generated. The SMOTEAuto indicates that after preprocessing process the number of objects in the classes was balanced. For the SVM method C=100 and linear kernel have been chosen to be presented in the paper.
To compare the differences between appropriate and inappropriate preprocessing moments during model evaluation, the t-test for dependent observations or the Wilcoxon signed-rank test were applied according to the normal or non-normal distributions of samples. The significance level was set at the level of 0.05. To unify the results, median values and the range as adequate statistics for all cases are presented in the Table 2.

For the whole range of datasets and methods, the average value of the precision difference between appropriate and inappropriate moment of preprocessing is at the level of 0.1792 ± 0.1259 (median 0.1691). The average value of the recall difference equals 0.0711 ± 0.0976 (median 0.0478); the average value of the F-measure difference: 0.1373 ± 0.1099 (median 0.1244) and the average value of the AUC difference: 0.0421 ± 0.0507 (median 0.0274).
For the C4.5 algorithm these results are as follows: precision: 0.1814 ± 0.1261 (median 0.1719); recall: 0.1154 ± 0.1011 (median 0.0920); F-measure: 0.1546 ± 0.1141 (median 0.1420) and AUC: 0.0659 ± 0.0546 (median 0.0532). For the SVM algorithm adequately: precision: 0.1768 ± 0.1259 (median 0.1667); recall: 0.0267 ± 0.0701 (median 0.0178); F-measure: 0.1200 ± 0.1028 (median 0.1063) and AUC: 0.0183 ± 0.0320 (median 0.0111). All these results are statistically significantly different than zero.
Apart from statistics, p-values for particular methods and algorithms are presented in the Table 2. In almost all cases there is a statistically significant difference between average results in two moments of preprocessing. Only for spider-type methods in few cases there is no statistical difference for SVM algorithm. The linear decision function not proper for the structure of data in classes may be the reason. Additionally two figures present the variability of F-measure differences (Fig. 3) and AUC differences (Fig. 4) for particular datasets and methods. Horizontal lines on the graphs show median value for each method.
The largest differences between mean values and two moments of preprocessing have been noticed for glass2 (precision: 0.42; recall: 0.26; F-measure: 0.36; AUC: 0.12), glass4 (precision: 0.30; recall: 0.21; F-measure: 0.27; AUC: 0.11) and haberman (precision: 0.33; F-measure: 0.23; AUC: 0.07). The quantities of the difference are astounding and further investigation will be performed. Inappropriate moment of data preprocessing while using cross-validation techniques, regardless of the classification method, may have large impact on the results giving too optimistic result of the classification process.

5 Conclusions

Class imbalance datasets is nowadays a common occurrence, mainly due to the possibility of collecting large amounts of information. This phenomenon may

Table 2: Results for precision, recall, F measure and AUC using three preprocessing methods and their modifications for two classification algorithms

Precision						
	C4.5			SVM		
method	*Me*	*Range*	p	*Me*	*Range*	*p*
slsmote	0.1826	0.0000; 0.6311	<0.0001	0.2439	0.0000; 0.5825	<0.0001
smote100	0.0685	0.0065; 0.1986	<0.0001	0.1050	0.0000; 0.2023	<0.0001
smote200	0.1637	0.0000; 0.3183	<0.0001	0.1826	0.0000; 0.3125	<0.0001
smote300	0.1866	0.0000; 0.3880	<0.0001	0.2345	0.0000; 0.3671	<0.0001
smote400	0.2184	0.0000; 0.4446	<0.0001	0.2631	0.0000; 0.4154	<0.0001
smote500	0.2297	0.0000; 0.4527	<0.0001	0.2796	0.0000; 0.4747	<0.0001
smoteAuto	0.2043	0.0000; 0.5665	<0.0001	0.2092	0.0000; 0.5835	<0.0001
spider-strong	0.1866	-0.0011; 0.5403	<0.0001	0.1367	0.0000 ;0.4031	<0.0001
spider-weak	0.1290	0.0000; 0.4388	<0.0001	0.1268	-0.0135; 0.3277	<0.0001
spider-weak-rel	0.1169	0.0000; 0.4445	<0.0001	0.1263	-0.0135; 0.3078	<0.0001
Recall						
method	*Me*	*Range*	p	*Me*	*Range*	*p*
slsmote	0.1271	0.0128; 0.6533	<0.0001	0.0452	0.0000; 0.2945	<0.0001
smote100	0.0709	-0.0277; 0.2647	<0.0001	0.0097	-0.0296; 0.0944	0.0001
smote200	0.0843	0.0064; 0.2294	<0.0001	0.0318	-0.0036; 0.2153	<0.0001
smote300	0.0858	0.0000; 0.2955	<0.0001	0.0293	0.0000; 0.2307	<0.0001
smote400	0.1000	-0.0022; 0.3458	<0.0001	0.0242	-0.0007; 0.2138	<0.0001
smote500	0.0971	0.0111; 0.3117	<0.0001	0.0273	0.0000; 0.1910	<0.0001
smoteAuto	0.0896	-0.0020; 0.4203	<0.0001	0.0236	-0.0050; 0.1330	<0.0001
spider-strong	0.1222	0.0009; 0.5724	<0.0001	0.0010	-0.2037; 0.2171	0.587
spider-weak	0.1002	0.0000; 0.5204	<0.0001	-0.0108	-0.1910; 0.2185	0.129
spider-weak-rel	0.0932	0.0000; 0.4998	<0.0001	-0.0229	-0.1802; 0.1800	0.048
F Measure						
method	*Me*	*Range*	p	*Me*	*Range*	*p*
slsmote	0.1618	0.0066; 0.6437	<0.0001	0.1561	0.0000; 0.5205	<0.0001
smote100	0.0852	0.0066; 0.2576	<0.0001	0.0761	0.0000; 0.1472	<0.0001
smote200	0.1254	0.0088; 0.2703	<0.0001	0.1070	0.0000; 0.2714	<0.0001
smote300	0.1496	0.0094; 0.3492	<0.0001	0.1508	0.0000; 0.2951	<0.0001
smote400	0.1641	0.0097; 0.4080	<0.0001	0.1773	0.0000; 0.3658	<0.0001
smote500	0.1654	0.0104; 0.3937	<0.0001	0.1907	0.0000; 0.3934	<0.0001
smoteAuto	0.1440	0.0059; 0.5113	<0.0001	0.1422	0.0000; 0.4893	<0.0001
spider-strong	0.1495	0.0004; 0.5578	<0.0001	0.1022	-0.0657; 0.2886	<0.0001
spider-weak	0.1169	0.0004; 0.4786	<0.0001	0.0587	-0.0455; 0.2317	0.0004
spider-weak-rel	0.1181	0.0004; 0.4731	<0.0001	0.0841	-0.0261; 0.2125	<0.0001
AUC						
method	*Me*	*Range*	p	*Me*	*Range*	*p*
slsmote	0.0684	0.0065; 0.1986	<0.0001	0.0215	0.0000; 0.1475	<0.0001
smote100	0.0341	0.0003; 0.0993	<0.0001	0.0053	-0.0081; 0.0462	<0.0001
smote200	0.0451	0.0006; 0.1152	<0.0001	0.0164	0.0000; 0.1054	<0.0001
smote300	0.0516	-0.0006; 0.1418	<0.0001	0.0126	0.0000; 0.1089	<0.0001
smote400	0.0516	-0.0001; 0.1461	<0.0001	0.0112	-0.0010; 0.1069	<0.0001
smote500	0.0457	0.0018; 0.1542	<0.0001	0.0109	-0.0004; 0.0972	<0.0001
smoteAuto	0.0450	-0.0011; 0.1994	<0.0001	0.0111	-0.0017; 0.0779	<0.0001
spider-strong	0.0757	0.0004; 0.2834	<0.0001	0.0076	-0.0921; 0.1122	0.063
spider-weak	0.0639	0.0004; 0.2552	<0.0001	0.0051	-0.0866; 0.1122	0.440
spider-weak-rel	0.0621	0.0004; 0.2613	<0.0001	0.0068	-0.0818; 0.0937	0.303

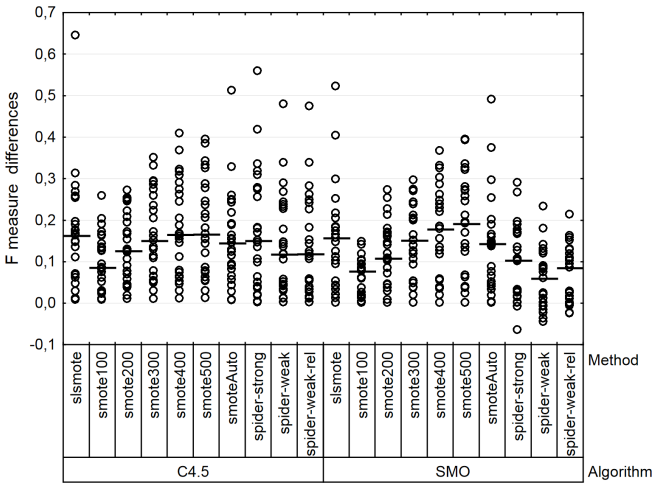

Fig. 3: Variability of F measures differences

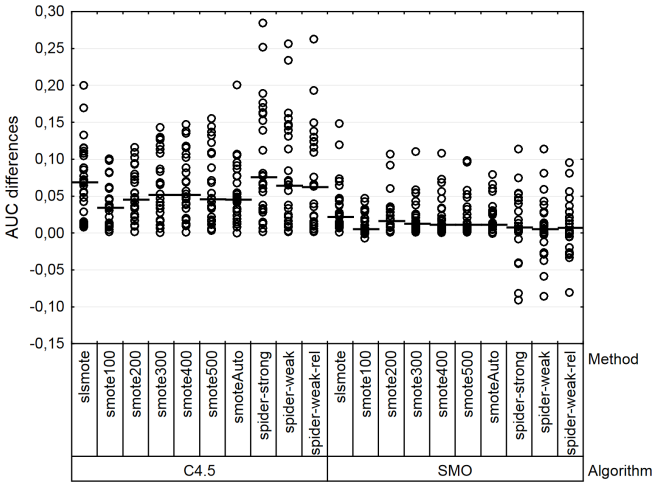

Fig. 4: Variability of differences between AUC values

have a direct impact on the performance of classification methods, because most classifiers are designed to maximize the accuracy or minimize the error rate. Assuming the equality of misclassification errors, the classifiers are usually overwhelmed by negative class objects, which leads to degradation of the performance of the classifier. Data preprocessing can be a good approach overcoming the disparity between classes cardinalities. However, the calculations should be performed in an attentive matter to not obtain excessively positive results. Even if the results are overestimated for about a few percent, each incorrect decision should be avoided.
There is a significant influence of the moment choice when preprocessing is performed. Results differences vary depending on a chosen classification algorithm and specific dataset. In most cases rates are inflated if preprocessing is performed before cross-validation. There are two reasons of this issue. Firstly, synthetic objects that are put into test sets are closely related to their prototypes in training sets, causing overfitting of a model. Secondly, proportions of classes in test sets in this method are not the same as real proportions that were given in the input sets and it affects the values of calculated rates.
The results occurred astounding and further investigation for various classification methods and datasets will be performed.

Acknowledgements

This work was performed in the framework of the grant S/WI/2/2013 (Bialystok University of Technology), founded by the Polish Ministry of Science and Higher Education.

References

1. Alcalá-Fdez J., Fernández A., Luengo J., Derrac J., and García S.: Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. Multiple-Valued Logic and Soft Computing, 17(2-3): 255–287 (2011)
2. Bunkhumpornpat C., Sinapiromsaran K., and Lursinsap C.: Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for Handling the Class Imbalanced Problem, In Proceedings of Advances in Knowledge Discovery and Data Mining: 13th Pacific-Asia Conference, PAKDD 35 2009 Bangkok, Thailand, April 27-30, 2009, 475–482. Springer, Berlin, Heidelberg (2009)
3. Chawla N. V., Bowyer K. W., Hall L. O., and Kegelmeyer W. P.: Smote: Synthetic minority over-sampling technique, Journal of Artificial Intelligence Research, 16:321–357 (2002)
4. Chawla N. V., Japkowicz N., and Kotcz A.: Editorial: special issue on learning from imbalanced data sets, SIGKDD Explorations, vol. 6, no. 1, 1–6 (2004)
5. Cover T., and Hart P.: Nearest neighbor pattern classification, IEEE Transactions on Information Theory, vol. 13, issue 1, 21–27 (1967)

6. Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., and Witten I. H.: The weka data mining software: An update. SIGKDD Explor. Newsl., 11(1): 10–18 (2009)
7. Hastie T., Tibshirani R., and Friedman J.: The elements of statistical learning: data mining, inference and prediction, Springer (2009)
8. He H. and Garcia E. A.: Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, 1263–1284 (2009)
9. Krzanowski W. J., and Hand D. J.: ROC Curves for Continuous Data, Chapman & Hall/CRC (2009)
10. Stefanowski J. and Wilk S.: Selective Pre-processing of Imbalanced Data for Improving Classification Performance. In Proceedings of Data Warehousing and Knowledge Discovery: 10th International Conference, DaWaK 2008 Turin, Italy, September 2-5, 283–292. Springer, Berlin, Heidelberg (2008)
11. Altini M.: Dealing with imbalanced data: undersampling, oversampling and proper cross-validation, http://www.marcoaltini.com/blog/dealing-with-imbalanced-data-undersampling-oversampling-and-proper-cross-validation (accessed 4.04.2016)
12. UC Irvine Machine Learning Repository, http://archive.ics.uci.edu/ml/, (accessed 15.02.2016)