Application of Sampling Algorithms in the Problem of Classification of User Requests*

Sergey R. Maslikhov [1], Andrey S. Mokhov [1][0000-0002-1979-6411],
Viktoriya N. Taran [2][0000-0002-9124-0178] and Vladimir O. Tolcheev[1]

[1]National Research University "Moscow Power Engineering Institute", Krasnokazarmennaya 17, Moscow, 11250, Russian Federation
[2] V.I. Vernadsky Crimean Federal University, 295007, Simferopol, Russia
maslihov2000@mail.ru
asmokhov@mail.ru
victoriya_yalta@ukr.net
tolcheevvo@mail.ru

Abstract. The article discusses the application of methods for reducing the size of majority classes (undersampling) in unbalanced samples to the problem of processing and classifying user requests sent to the technical support of the portal of the Center for Industry Information and Analytical Systems. Sampling methods are investigated, indicators of the classification quality are given - F1-measure for macro and micro averaging. The paper considers a sample consisting of user requests to the technical support service, which has a highly unbalanced form - volume of some classes in the sample exceeds the volume of others by several times. On this sample, various undersampling algorithms are used, a conclusion is made about the advisability of using certain algorithms. A study of the influence of undersampling and oversampling methods on the accuracy of classification of user requests to the portal's technical support service when using gradient boosting on decision trees was carried out. The results obtained showed that an increase in F1-measure compared to the base model was achieved only with the use of the Tomek Links undersampling method; other methods for reducing the size of majority classes did not give an increase inaccuracy, and in most cases even worsened the result.

Keywords: Data Mining, Text Classification, Machine Learning, Data Sampling, Undersampling, Unbalanced Sampling.

1 Introduction

Nowadays, due to the rapid development of computer technology, a section of data science known as Machine Learning has gained popularity. One of the problems of machine learning is the classification problem, which often has to be solved in conditions of imbalanced classes. This situation arises in cases when the proportion of objects of one or several classes in the training sample is significantly greater than 1/n, where n is the number of classes. The consequence of such an imbalance in the data is both the worse quality of the model's operation on objects of minority classes (in comparison with the majority) and the bias of classification metrics, for example, accuracy (the proportion of correct answers).

To improve the quality of classification, there are sampling methods, which can be divided into two main groups: oversampling (generation of additional objects of a smaller (minority) class) and undersampling (removal of objects of a larger (majority) class). In this paper, we consider a sample consisting of user requests to the technical support service [8], which has a highly unbalanced form - the volume of some classes in the sample exceeds the volume of others by several times. In this sample, various under-sampling algorithms are used, a conclusion is made about the advisability of using certain algorithms.

The purpose of the article is to analyze the well-known sampling algorithms to increase the accuracy of the classification of user requests to the technical support service.

2 Description of samples

There are many requests from X users to the portal technical support. Each request, $t = \overline{1, 22000}$, belongs to one of 13 classes and is presented in text format with the following structure: a greeting phrase - a description of the problem or the reason for contacting - a signature.

The distribution of queries by class is shown in Table 1. It can be seen that the training set has a highly uneven distribution of objects

Table 1. Distribution of queries by classes.

#	Class	Share	Volume
0	State Tasks	5.427	1180
1	CRRR	46.130	10031
2	Miscellaneous	0.616	134
3	Performance reports	1.012	220
4	Standard costs	0.510	111
5	“Do the right thin” Portal	23.601	5132
6	FM rating	0.570	124
7	Subsidies calculations	1.734	377
8	Advanced trainings and seminars	3.527	767
9	SOI	9.501	2066
10	AC of FMCS	0.510	111
11	Salary monitoring	1.067	232
12	Budgeting and FEAP	5.794	1260

The solution to the problem is divided into 3 stages.

Data preprocessing Building a basic model (on the original sample) Building and analyzing models on modified samples Data processing is the extraction of Russian-language terms from the query text, their normalization (reduction to the initial form), and removal of stop words. In this task, the stop word dictionary included auxiliary parts of speech, names of people, and terms most frequently used in greeting phrases.

Figure 1 shows the visualization of the sample using the T-SNE method [1]. The imbalance of classes is most clearly expressed in the yellow, blue and red colors prevailing in the figure. Many classes, due to their small number, turned out to be hardly noticeable on visualization. Moreover, the data are rather strongly mixed, but, at the same time, a cluster structure can be distinguished.

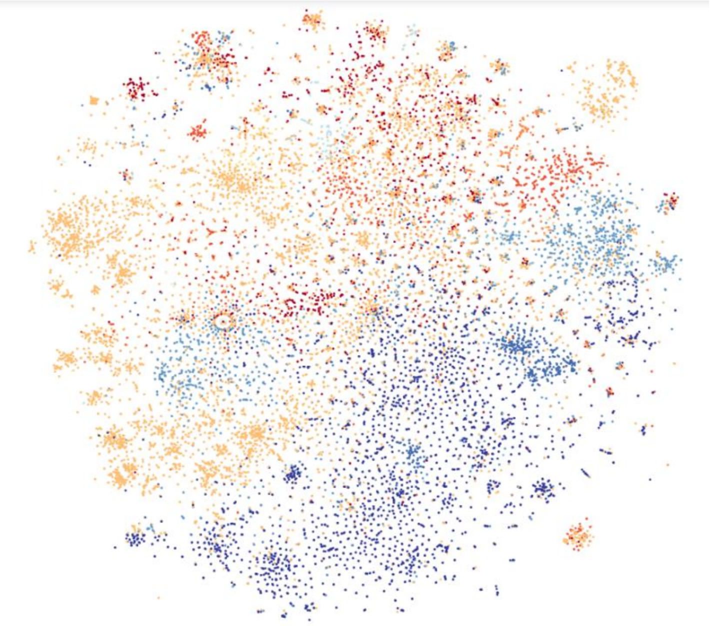

Fig. 1. Sample visualization.

As a basic model for classifying queries, we will consider a classifier built based on gradient boosting on decision trees (XGBClassifier) without using sampling methods. The choice of the XGBClassifier is based on the best classification results in comparison with other classifiers [2,6,7].

The f1-measure with micro and macro-class averaging will be used as an indicator of quality.

$$precision = \frac{TP}{TP+FP} \tag{1}$$

$$\text{recall} = \frac{TP}{TP+FN}, \tag{2}$$

where TP (True Positive), FP (False Positive), FN (False Negative) - error matrix indicators (confusion matrix) [3].

$$F1 = \frac{2*precision*recall}{precision+recall} \tag{3}$$

Micro-averaging does not take into account the belonging of an object to a class, i.e. each object contributes equally to the calculation of the measure. The total values TP, FP, FN, TN are calculated as the sum of the components for each class, and then the F1 measure itself is calculated from the total values

With macro averaging, the value of the F1-measure is calculated as the average F1 for all classes, each class equally participates in the calculation of the measure.

The choice of the preferred averaging is based on the task at the given moment. The calculation of the F1-measure for micro-averaging is carried out using absolute scales, while for macro-averaging - using relative scales. Having received the general indicators of the error matrix in the first case, a large contribution will be made by the objects of the majority classes (due to their numerical superiority). In the second case, we equalize the contribution of each of the classes, ignoring their numbers. Thus, if the value of identifying minority objects in the problem turns out to be higher than the value of majority objects, then it is worth focusing on macro-averaging (for example, identifying "unreliable" bank customers in the problem of credit scoring).

In our task, the value of each class turned out to be equal, therefore, more attention was paid to micro-homogenization. Next, we investigated several sampling methods based on reducing the size of majority classes. As is known, with an uneven distribution of objects across the sample classes, the classifier model is trained unevenly on different classes, and therefore there is a shift in training towards majority classes, and in the extreme case, it is more profitable for the classifier to assign all objects to the majority class than to try to allocate objects of the minority class.

In studies, we will reduce those classes whose share exceeds 1/n, where n is the number of classes in the sample. In the literature, there are quite a few sampling methods associated with reducing the size of classes [4]:

- Tomek Links method,
- Neighbourhood Cleaning Rule,
- Random UnderSampler,
- Near Miss,
- Condensed Nearest Neighbour.

From Table 2 it can be seen that sampling methods associated with reducing the size of majority classes in most cases do not increase the classification accuracy. A slight increase in accuracy was achieved only with the use of Tomek Links, the success of which can be justified by its algorithm, which removes objects of the majority class located near objects of the minority class. Other methods are likely to reduce the majority class too much, which leads to the underfitting of the model [5].

Table 2. Undersampling results.

Method	*F1-micro*	*F1-macro*
Basic model	0.825	0.669
Tomek Links	0.828	0.681
NeighbourhoodCleaningRule	0.815	0.674
EditedNearestNeighbours(k=3)	0.769	0.647
EditedNearestNeighbours(k=5)	0.730	0.609
EditedNearestNeighbours(k=1)	0.812	0.674
RandomUnderSampler	0.759	0.674
NearMiss(version=1)	0.721	0.654
NearMiss(version=2)	0.732	0.655
NearMiss(version=3)	0.583	0.474
CondensedNearestNeighbour	0.636	0.571

It is important to note that, despite the deterioration in micro-averaging, when using the Neighborhood Cleaning Rule, Edited Nearest Neighbors (k = 1), Random Under Sampler, there is an improvement in the macro-averaged F1-measure

3 OverSamping Method

The undersampling methods have been discussed above. At the same time, some methods make it possible to increase the number of objects of minority classes by synthesizing new objects based on existing

SMOTE

This strategy is based on the idea of generating some artificial examples that would be “similar” to those in the minority class, but at the same time would not duplicate them [9,10]. To create a new record, find the difference

$$d = Xb - Xa,$$

where Xa, Xb is the feature vectors of “neighboring” examples a and b from the minority class. They are found using the nearest neighbor (KNN) algorithm. In this case, it is necessary and sufficient for example b to obtain a set of k neighbors, from which the record b will be selected later. The rest of the steps of the KNN algorithm are not required.

Further, from d by multiplying each of its elements by a random number in the interval (0, 1), dˆ is obtained. The feature vector of the new example is calculated by adding Xa and dˆ. The SMOTE algorithm allows you to set the number of records that must be artificially generated. The degree of similarity between examples a and b can be adjusted by changing the value of k (the number of nearest neighbors).

ADASYN

Algorithm:

- Calculate the ratio of the number of minority objects to the number of majority ones: d = m_s / m_l, m_s is the number of minority objects, m_ (l) is the number of majority objects. If d is below a certain limit, go to the next steps of the algorithm.
- Calculate the total number of objects to generate: G = (m_l- m_s) β, where β = 1 means a fully balanced dataset.
- Find k nearest neighbors for each minority object and calculate

  r_i = 〚majority〛 _i / k,

  where majority i is the number of majority objects and k neighbors of the ith minority object.
- Normalize

  (r_i)ˆ= r_i / (∑r_i)
- For each neighborhood, calculate the number of generated objects

  G_i=G*(r_i)ˆ
- New Gi objects are generated according to the formula:

  s_i = x_i + (x_zi-x_i) α,

  x_i is an object relative to which new examples are generated.
  x_zi is a randomly selected example of a minority class in the x_i neighborhood.
  α is a random number in the range from 0 to 1.
  If there are no examples of a minority class out of k neighbors, then either no new objects are created for xi, or xi is duplicated Gi times.

RandomOverSampler – random selection of objects with the return.

SMOTE can link internal and external values, while ADASYN can focus exclusively on outliers, which in both cases can lead to suboptimal decision functions. In this regard, SMOTE offers two additional sample generation options (SVMSMOTE, BorderlineSMOTE). These methods focus on samples near the border of the optimal decision function. A visual illustration is shown in Table 3 and Fig. 2.

Table 3. Oversampling results.

Method	*F1_micro*	*F1_macro*
Basic quality	0.834	0.694
ADASYN	0.826	0.683
SVMSMOTE	0.827	0.681
BorderlineSMOTE	0.828	0.674
SMOTE	0.826	0.685
RandomOverSampler	0.826	0.692

The deterioration in the quality of the model can be explained by the low linear separation of the sample, since all methods, except for RandomOverSampler, use the algorithm of nearest neighbors, which can lead to the generation of new objects belonging to the majority class.

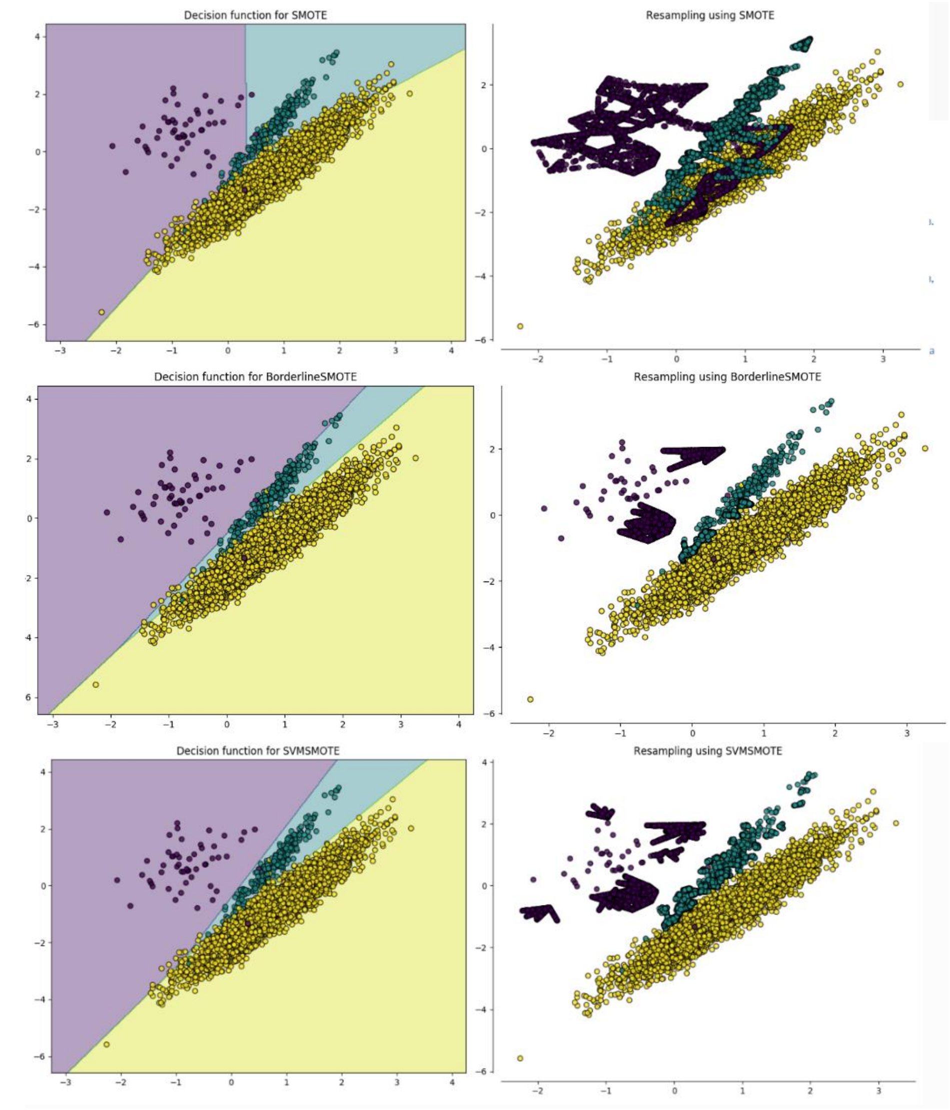

Fig. 2. Distribution of requests by classes.

4 Conclusions

A study was carried out of the influence of undersampling and oversampling methods on the accuracy of the classification of user requests to the portal technical support service when using gradient boosting on decision trees. The results obtained showed that an increase in accuracy compared to the baseline model was achieved only with the use of the Tomek Links undersampling method, other methods for reducing the size of majority classes did not give an increase inaccuracy, and in most cases even worsened the result. The use of oversampling methods did not give results on this sample, which is most likely due to the poor linear separability of the sample. A further direction of research is to study the possibility of improving the accuracy of classification by simultaneously increasing the minority classes and reducing the majoritarian ones.

This method is aimed at reducing the error and shows good results when applied in various applied calculations.

References

1. Van der Maaten, L.J.P., Hinton, G.E.: Visualizing Data Using t-SNE. Journal of Machine Learning Research. 2008. V. 9.
2. Tianqi, Chen, Carlos, Guestrin, Boost, X.G.: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 2016, pp. 785–794. DOI: 10.1145/2939672.2939785
3. 3. Manning, K., Raghavan, P., Schütze, H.: An Introduction to Information Retrieval. M. Williams, 2014.528 p.
4. Under-sampling. Imbalanced-learn user-guide. Электронный доступ: https://imbalanced-learn.readthedocs.io/en/stable/under_sampling.html
5. Tahir, M.A.U.H., Asghar, S., Manzoor, A., Noor, M.A.: A Classification Model for Class Imbalance Dataset Using Genetic Programming. IEEE Access. 2019. V. 7. C. 71013-71037. DOI: 10.1109/ACCESS.2019.2915611
6. Bobryakov, A., Kuryliov, V., Mokhov, A., Stefantsov, A.: Approaches to Automation Processing of User Requests in a Multi-Level Support Service Using Featured Models. Proceedings of the 30th DAAAM International Symposium, pp. 0936-0944, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-22-8, ISSN 1726-9679, Vienna, Austria DOI: 10.2507/30th.daaam.proceedings.130
7. Batura, T.V.: Metodi Avtomaticheskoy Klassifikacii Tekstov. Software & Systems. №1(30), pp. 85-89, DOI: 10.15827/0236-235X.117.085-099. (2017)
8. Alashkevich, M., Bobryakov, A., Klimenko, A.,Stefantsov, A.: Automation and InformationalSupport of Budgetary Institution Financing Processes, Chapter 27 in DAAAM International Scientific Book, 2015. – Pp. 319–328. B. Katalinic (Ed), Published by DAAAM International,ISBN: 978-3-902734-05-1, ISSN 1726-9687, Vienna, Austria. DOI: 10.2507/daaam.scibook.2015.27. (2015)
9. Chawla, Nitesh V., et al. SMOTE: Synthetic Minority Over-Sampling Technique. Journal of artificial intelligence research 16, pp. 321-357. (2002)
10. Fithria Siti Hanifah, Hari Wijayanto, Anang Kurnia: SMOTE Bagging Algorithm for Imbalanced Data Set in Logistic Regression Analysis. Applied Mathematical Sciences, Vol. 9, No. 138, pp. 6857-6865. (2015)