Available online at http://scik.org
J. Math. Comput. Sci. 2022, 12:18
https://doi.org/10.28919/jmcs/6648
ISSN: 1927-5307

CLASSIFICATION OF MINORITY CLASS IN IMBALANCED DATA SETS

DANAIL SANDAKCHIEV*, IVAN IVANOV

Faculty of Economics and Business Administration, Sofia University “St. Kliment Ohridski”, Sofia 1000, Bulgaria

Abstract: This paper focuses on imbalanced data sets and uses different methodologies for the part of classification process related to building train and test subsets. Popular classification methods are then applied and evaluated based on the recall result for the minority class. On the basis of the results from our experiments we suggest that for imbalanced data sets when the minority class presents noticeably higher interest, we should use alternative methodology for building the train subset and not the standard random allocation of observations, in order to improve the predictive power of the classifier for the minority class.

Keywords: machine learning; classification problems; imbalanced data sets.

2010 AMS Subject Classification: 00A99.

1. INTRODUCTION

Machine learning has gained great popularity in recent years, as it has various application in the many areas. Making relationships between different features in data and extracting valuable information to solve problems can be very challenging. Machine learning has proven to be efficient in such cases. This is more true than ever in today’s world of information technologies with

*Corresponding author
E-mail address: sandukchie@uni-sofia.bg

Received October 15, 2021

constant inflow of data being recorded in huge number of systems and databases.

In a database, observations can form data sets with specific features. This way each observation in the data set is described by available features. The features can be continuous, categorical or binary. Observations may belong to a specific class characterized by the values of its features. If we know the correct class for given observations then we can apply supervised learning. Part of supervised learning are classification problems [9]. Some of the commonly used classification methods include - logistic regression, k nearest neighbors, decision trees, support vector machines, neural networks and more.

Solving classification problems can help society and organizations overcome different risks and challenges. Helping medical personnel process different tests of patients and direct patients to appropriate specialists for treatment is an example of a use case in health care. Targeting potential customers can significantly improve the efficiency of commercial companies. Categorizing different texts and images helps automate many processes in different organizations [1]. These are just few examples of the many possible application of classification problems.

There are a number of challenges that analysts face when trying to work on classification tasks. It is not unusual for a data set to have a dominant class - observations predominantly belonging to one class, and much less observations belonging to another class. It is often referred to such data sets as imbalanced. Often, the minority class constitutes the real interest for the analysts [3]. For example, a bank would have mostly borrowers who are able to make installments on their loans, otherwise the bank would fail. Hence, the bank would have to make prediction on customers with high risk using considerably less bad payers versus good payers available in its customer database.

Traditionally, when building a classifier the input data set is split into two subsets. One of the subsets is used to build the classifier, hence it is called train subset and the other subset evaluates the performance of the classifier by categorizing observations not “seen” by the classifier during the training phase. As standard, the bigger portion of the observations are allocated to the train subset. Both the train and test subsets must have the same features [7].

In our study, we examine different imbalanced data sets and apply popular classification methods on them. We use different methodologies to create the training and test subsets. In our results, we focus on improving the predictive power for the minority class. The rest of the paper is structured as follows. In Literature Review, we present briefly the different classification methods and methodologies for building the training subset which are used in the experiments. We also make an overview of the key indicators that are used to evaluate the performance of the classificators. In the Experiments section of the paper, we present the results from the study. Finally, the final remarks are given in the Conclusions section of the paper.

2. LITERATURE REVIEW

In our study, we will use several popular classification methods and apply them to the selected data sets. We will briefly present each method without going into details, as it is outside the scope of this paper. The methods in question are logistic regression, k nearest neighbors, decision trees, random forest, support vector classifier and naive bayes classifier.

Logistic regression models the posterior probabilities of the K classes using linear functions of the independent variables. The probabilities are constrained to sum to 1 specified by the K-1 log-odds or logit transformations [10]. Logistic regression models are widely used in medical fields, social sciences, scoring loan applicants, sports betting among others.

One of the simpler classification methods is k nearest neighbors. When a new test observation is classified, the distance (usually Euclidean or Manhattan) is computed with each training data point. The class of the k nearest train points (neighbors) is considered at next stage. The predominant class within the k nearest neighbors determines the class of the test observation at the end [2]. The method finds applications in bankruptcy predictions, customer relationship management, fraud detection, image recognition and many more fields.

The concept of decision trees is relatively easy to grasp. The method partitions the feature space into a set of rectangles. Following that in each one of the rectangles a simple model is fit. The same partition can be illustrated also as a tree or a flowchart-like structure. In this way each node tests an attribute, each branch is an outcome when applying the model and leaf nodes represent a class label. The road from the root to the leaf is determined by classification rules. Random forest uses similar concept, but creates multiple decision trees and determines the class based on results from the individual trees. For example, the mode or mean of the individual trees can be used as decision rule for the class [10].

Support vector classifier belongs to the support vector machine family for classification. The technique handles primarily binary classification problems, but it can be modified to work with data sets containing more than two classes. The method can use different kernels which can be linear or non-linear. In principal, these methods (support vector machines) are defined as optimization problem which has a single solution [6]. The method is widely used in many areas - finance, credit risk, time series forecasting, image recognition and many more.

The final classification method which we use in our experiments is the Naive Bayes classifier. The classifier strives to create a model (from an existing and available data) that splits the data into different classes. Based on the build model, it tries to predict the class of a new observation. When building the model, a subset of the input data set is created and it is determined the elements of which class are most common. It is considered that any new observation in this subset would belong to the same class. [10] The method derives from the Bayes definition for conditional probability and the assumption for independence among the variables (hence the “Naive” part in the name).

As standard practice in classification problems, a model is created by training it on part of the data set and then the remaining of the observations are used to test the predictive power of the model. A very intuitive way to separate the input data set into train and test is to randomly assign each observation to one of the two subsets using a predefined proportion of observations for each subset. For example, we have a data set of 100 observations. We decide that 80 percent of the observations are reserved for training and the remaining 20 percent for testing. We proceed by randomly selecting 80 observations for the training subset and 20 observations for the test subset.

There are alternative methodologies available for the train/test subset creation. In our study, we use Random Oversampling and Random Undersampling. These methodologies can be of particular help when working with imbalanced data sets or data sets that have predominant class (classes are not equally represented). In the following paragraph, we are briefly describing the concept of these two methodologies.

Random Oversampling and Random Undersampling are two sides of the same coin. Random Oversampling tries to balance class distribution by replicating random observations from the minority class. On the other hand, Random Undersampling aims to balance class distribution via random elimination of observations that belong to the majority class. A drawback of Random Oversampling is that could potentially lead to overfitting due to the extensive copying of minority class examples. As for the Random Oversampling, this methodology could possibly harm the training by eliminating useful members of the majority class [4].

To evaluate the performance of a classifier, analysts often rely on the confusion matrix. Below we visualize the dimensions of a confusion matrix.

Table 1. Confusion Matrix

	Actual	
Predicted	True Positives (TP)	False Positives (FP)
	False Negatives (FN)	True Negatives (TN)

Positives refer to correctly classified observations, whereas negatives refer to misclassified observations. More specifically, True Positives (TP) gives us the observations which are observed positive and are actually positive, True Negatives (TN) are observed negative and actual negative, False Positives (FP) are observed negatives and actual positives, and finally, False Negatives are observed positives and actual negatives [11].

Using the confusion matrix, we can define key measures for the performance of a given classifier. Accuracy as a performance measure looks at the sum of all correctly classified observations divided by the total number of observations in the subset [11]

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Sensitivity, also known as recall, provides information on the accuracy of only positive predictions [11].

$$\text{Sensitivity} \quad = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Specificity, also known as precision, represents the accuracy only of negative predictions [11].

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

F1 score is a statistical test that takes into consideration both sensitivity and specificity. In other words, it is the weighted average of the sensitivity and specificity [11]:

$$\text{F1} = 2 * \frac{\text{Sensitivity} * \text{Specificity}}{\text{Sensitivity} + \text{Specificity}}$$

In our study, we are focusing on the recall (sensitivity) measure for the minority class within a data set. This measure is particularly important when we work with imbalanced data set where the minority class has high value even though in reality it is considerably less observable than the majority class. For example, when banks give out loans, their primary interest is to avoid giving loans to entities unable to pay the loan back. These cases (this class) are usually far fewer than the ones who pay their principal and interest on their loans. Therefore, banks are much more interested in correctly predicting who will not be able to pay back the loan than the cases which will pay back on time. Recall focuses exactly on giving us indication what proportion of the entities not able to pay the loan are correctly predicted. Similar examples can be given for health care cases, customer retention cases and many more fields.

3. Main Results

The experiments for the study are performed using Python 3.7 and Spyder 4.1.1 as development environment.

Six different classification methods are used in the study - k-NN, Logistic Regression, Decision Tree, Random Forest, Support Vector Classifier and Naive Bayes Classifier. The following libraries and methods in Python are applied. For k-NN classifier, the neighbors method from the sklearn library is imported. As number of neighbors we use the square root of the number of observations in the respective data set. For the Logistic Regression, we load the LogisticRegression method from the sklearn.linear_model library. For the Decision Tree, we import the DecisionTreeClassifier from sklearn.tree library. For the Random Forest, we use the RandomForestClassifier from sklearn.ensemble. For Support Vector Classifier, we load the SVC method from sklearn.svm library. And finally, for the Naive Bayes classifier, we import the GaussianNB method from sklearn.naive_bayes library. All classifiers are used with default values for their parameters.

We are also applying three different methodologies when building the train and test subsets. We refer to the random allocation of observations from the input data set into train and test subsets as the standard methodology. For this purpose, we take advantage of the train_test_split method available from the sklearn.model_selection library. In all instances (data sets), 80 percent of the observations from the input data sets are reserved for the train subset and respectively 20 percent of the observations go to the test subset. The other methodologies experimented with are Random Undersampling and Random Oversampling. In order to use them the following commands are imported in Python. For Random Undersampling, we load RandomUnderSampler from imblearn.under_sampling. For Random Oversampling, we import RandomOverSampler from imblern.over_sampling. All methodologies are used with their default values for the their respective parameters.

In Table 2, we present the data sets that are used for the experiments. For each data set, we note the number of observations, features and number of observations representing each class. Each data set has two classes, where class 0 is always the majority class and class 1 is always the minority class in the respective data set.

The data sets have a range of number of observations between 2 201 and 7 178. All of the data sets can be considered large in terms of number of observations. The number of features vary from 3 to 19 in the selected data sets. Based on the number of observations representing each class, we can say that the data sets are imbalanced, as the majority class has twice as much (or close to twice as much) observations than the minority class.

Table 2. Data sets

Dataset	Number of observations	Number of features	Class 0 Observations	Class 1 Observations
Marital Satisfaction	7178	11	4903	2275
Titanic	2201	3	1490	711
Phoneme	5404	5	3818	1586
Churn	7043	19	5174	1869

Tables 3, 4, 5 and 6 depict the recall results achieved for the minority class for each of the 4 data sets when using the standard method for allocating observations to train and test data set, and also the recall result achieved when using Undersampling and Oversampling as methodology. The leftmost column in the tables informs of the classification method applied.

Table 3.

Classificator	Marital Satisfaction		
	Standard	Undersampling	Oversampling
k-NN	0.52	0.69	0.70
Logistic Regression	0.58	0.73	0.73
Decision Tree	0.59	0.68	0.61
Random Forest	0.61	0.80	0.69
SVC	0.56	0.71	0.71
Naive Bayes	0.59	0.67	0.66

Table 4.

Classificator	**Phoneme**		
	Standard	**Undersampling**	**Oversampling**
k-NN	0.65	0.91	0.88
Logistic Regression	0.45	0.81	0.76
Decision Tree	0.77	0.85	0.75
Random Forest	0.88	0.92	0.87
SVC	0.76	0.94	0.90
Naive Bayes	0.73	0.84	0.80

Table 5.

Classificator	**Titanic**		
	Standard	**Undersampling**	**Oversampling**
k-NN	0.40	0.45	0.39
Logistic Regression	0.52	0.56	0.47
Decision Tree	0.42	0.54	0.61
Random Forest	0.42	0.61	0.53
SVC	0.54	0.55	0.61
Naive Bayes	0.57	0.66	0.55

Table 6.

Classificator	Churn		
	Standard	Undersampling	Oversampling
k-NN	0.26	0.58	0.65
Logistic Regression	0.56	0.81	0.81
Decision Tree	0.52	0.65	0.51
Random Forest	0.50	0.77	0.54
SVC	0.26	0.45	0.38
Naive Bayes	0.75	0.83	0.83

We can clearly see that recall results for the minority class are almost always higher when we use Random Undersampling or Random Oversampling as methodology for train/test. In many instances the improvement in the recall is quite noticeable and significant. For example, the Marital Satisfaction data set using Random Forest as classifier we get 0.61 recall for the minority class when the standard methodology for building train/test subsets is used, and with Random Undersampling we achieve 0.80 recall with the same classifier. Similar, the Churn data set has recall result 0.56 using the standard train/test split methodology when Logitic Regression is applied and much better recall when either Random Undersampling or Random Oversampling is applied - 0.81. Another example is Titanic data set, where we get recall of 0.42 for the minority class with Decision Tree classifier and impressive 0.61 recall applying Random Oversampling methodology for building the train and test subsets.

In general, we can note that using Random Oversampling or Random Undersamping manages to give us a recall improvement of at least 0.10 using different classification methods.

4. CONCLUSIONS

In this paper, we selected 8 data sets with imbalanced distribution of the observations, according to their classes. We examined building train subsets with standard partition of the data and alternative methodologies - Random Oversampling and Random Undersampling. We applied a number of popular methods for classification and evaluated their performance using the recall measure on the minority class.

Based on the results, we can conclude that no matter which classification method is being applied, in order to achieve a better recall result for the minority class in a given imbalanced data set, we need to use methodology for spitting the data et into train/test subsets different than the standard random allocation of observations to train/test subsets, like the examined Random Oversampling and Random Undersampling methodologies. Looking at the results presented in the paper, we could expect a significant increase of 10% in recall of minority class when applying an alternative methodology for splitting the input data set in train and test.

In future work, we can analyze characteristics of imbalanced data sets which may suggests specifically what methodology for train/test split would be most appropriate for the given data set.

CONFLICT OF INTERESTS

The author(s) declare that there is no conflict of interests.

REFERENCES

[1] A. Khan, B. Baharudin, L.H. Lee, K. Khan, A review of machine learning algorithms for text-documents classification, J. Adv. Inform. Technol. 1 (2010), 4-20.

[2] D. Sandakchiev, Modified k-NN algorithm with improved efficiency, Sofia University, Master’s Thesis (2019).

[3] G. Menardi, N. Torelli, Training and assessing classification rules with imbalanced data, Data Min. Knowl. Disc. 28 (2014), 92–122.

[4] G.E.A.P.A. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data, SIGKDD Explor. Newsl. 6 (2004), 20–29.

[5] H. He, Y. Bai, E.A. Garcia, S. Li, ADASYN: Adaptive synthetic sampling approach for imbalanced learning, in: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational

Intelligence), IEEE, Hong Kong, China, 2008: pp. 1322–1328.

[6] I. Ivanov, V. Tanov, Big data analytics algorithms and applications. Machine Learnings, Sofia (in Bulgarian) (2018).

[7] K. Weiss, T.M. Khoshgoftaar, D. Wang, A survey of transfer learning, J. Big Data. 3 (2016) 9.

[8] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002), 321-357.

[9] S.B. Kotsiantis, Supervised machine learning: a review of classification techniques, emerging artificial intelligence applications in computer engineering, IOS Press (2007).

[10] T. Hastie, R. Tibshirani, J. Friedman, The elements of statistical learning (data mining, inference, and prediction), Springer, Second Edition (2008).

[11] D. Tripathi, D.R. Edla, V. Kuppili, A. Bablani, R. Dharavath, Credit scoring model based on weighted voting and cluster based feature selection, Procedia Computer Sci. 132 (2018), 22–31.