Improve Class Prediction By Balancing Class Distribution For Diabetes Dataset

Mohammad Al Khaldy, Mohammad Alauthman, Majed S. Al-Sanea and Ghassan Samara

Abstract: When using machine-learning algorithms to analyses clinical data, some challenges are facing this kind of data. One of the limitations of data is class imbalance because class imbalance could create a suboptimal performance of the classifier. The purpose of this article is to evaluate the influence of imbalance class on classification efficiency for multiple classification methods. In addition, we resample data by random replacement technique with replacement and without replacement to see how balancing data can improve the performance of classification techniques. The experiments show that resampling with imbalanced replacement class obtains a considerable boost in classification effectiveness for most of the learning algorithms used, but after resampling class, the Naive Bayes algorithm has not been improved.

Index Terms: Imbalance class; Resampling; Random Forest, Naive Bayes, Bagging.

1. INTRODUCTION

Data mining is a multi-procedures that analyses a large amount of data to do classification, clustering, and regression [1-3]. This is done through different analytical tools, such as statistical and machine learning tools. Machine learning algorithms can learn the model on a large dataset to gain information and then knowledge. The learning model by machine learning algorithms can be unsupervised or supervised learning. In supervised learning, the input and output are presented for training data, (Classification is a technique of supervised learning, that leaned by training on a given label and then testing the model by predicting the class label.). (Whereas the other technique is unsupervised learning, which means that there are unlabeled examples, and the model will learn by clustering the instances in such similarity groups of patterns). One of the common challenges of investigating clinical datasets is class imbalance. The data mining models must manipulate any challenges that can affect the prediction accuracy. (The data is an imbalance when it has an unequal number of instances for each class). The minority class is the class with fewer samples, whereas the majority class refers to the class with larger samples. The problem of class imbalance is that the minority class has little chance to be trained in the learning algorithm, (therefore affecting the outcome of the results.)thus the result of performance will be affected [4]. Resampling is a method that can manage the data by raising the minority sample or lowering the majority sample, known as over-sampling and under-sampling, respectively, so that both class labels can be viewed equally [5].

2. BALANCING DATA TECHNIQUES

The sampling procedure is used when processing a big amount of data, selecting random samples with lower in size) [6]. By random sampling of the training data set, each particular data has the same chance of being involved; replacement or without replacement. Sampling without replacement; principally rejects the second copy for each example chosen Sampling with replacement; choosing more than one instance, this is used for the bootstrap algorithm)[7, 8]. The imbalance ratio (IR) can be defined by dividing the number of minority class samples into major class samples.). Class imbalance can be solved either by data level or algorithm level:

A. Data level: by balancing training dataset using resampling techniques [9]. Which are:

1. Resampling (external): its efficient and convenient way is not to modify the learning algorithm but only to alter the initial training set) [10]. This technique can be done by oversampling or under-sampling. The oversampling technique will increase the frequency of the minority class (Even as the frequency of the majority class is reduced under the sampling, see figure -1). (The drawbacks of the oversampling in more copies of the minority class being added to the data, which cause the overfitting), and the time to build a classifier will be increased because it increased the size of the training dataset [11]. (The disadvantage of the under-sampling is that many interesting and insightful examples may be removed; thus these can be important when creating the classifier) [12]. Active learning: the technique selects the more active sample to use in training algorithm; this can improve learner performance [13].

2. Weighting the data space: The distribution of the training set is manipulated using information regarding error classified costs.[14].

B. Algorithm level (Cost-sensitive learning): The approach makes the minority class's classifier algorithm more accurate. Attempt to understand more about minority examples to minimize higher cost failures by taking into account the more significant cost of misclassifying beneficial samples [10-14].

In this study, we use two technique to manipulate imbalance class:

1. Resample, this approach using either sampling with no replacement or with replacement. The original

- *Mohammad Al Khaldy, Al Khawarizmi University Technical College, Amman, Jordan, E-mail: m.khaldy@khwawarizmi.edu.jo*
- *Mohammad Alauthman, Department of Computer Science, Faculty of Information Technology, Zarqa University, Zarqa, Jordan, , E-mail: malauthman@zu.edu.jo*
- *Majed S. Al-Sanea, Computer and Information Science, Arab East Colleges, Riyadh, KSA , E-mail: msalsanea@arabeast.edu.sa*
- *Ghassan Samara, Department of Computer Science, Faculty of Information technology, Zarqa University, Zarqa, Jordan, , E-mail: gsamara@zu.edu.jo*

dataset needs to be entirely stored and the number of instances can be specified in the generated data set. The dataset must have a nominal class attribute using supervised learning. The filter can be produced to preserve the class distribution in the subsample or to favor a uniform class distribution.

2. Spread Sub-sample: Produces random dataset sub-samples. It must be fitfully in the initial dataset. The maximum "spread" between the fewest and most popular classes can be specified in this filter. For example, the difference in class frequencies must be at most 2:1.

3. EXPERIMENTAL DESIGN

3.1. EXPERIMENTAL SETUP

The statistical language environments WEKA and R were used for the experiments and statistical processing. In the following experiments, we performed 10-fold cross-validation so that the training set and test sets maintain the sa me class distribution as the original sets. Then compare the prediction performance by calculated metrics such as Accuracy, F-score, G-mean, ROC area, Sensitivity, and Specificity.

3.2. EXPERIMENTAL DATASET

This dataset includes attributes obtained from a picture of Measidor to predict whether or not a picture has indications of diabetic retinopathy; the dataset name is Diabetic Retinopathy Debrecen Data [15]. The number of features is 20 with 1151 patients' records, see table 1. The last attribute is a class variable that classifies the instances to "contain signs of DR" or "no signs of DR'.

***TABLE 1**. THE RESEARCH DATASET (DIABETIC RETINOPATHY DEBRECEN DATASET)*

Number of features	20	
Number of samples	1151	
Target output	Class	
Class	signs of DR	no signs of DR
Frequency	611	540
Imbalance Ratio (IR)	12%	

3.3. EXPERIMENTAL PARAMETERS

The confusion matrix is a particular table where it quantifies the number of prediction results after running the model comp ared to actual class outcomes. The matrix shows the number of positive instances that classified correctly called true positive (TP); the number of negative instances that classified correctly called (TN); the number of positive class the incorrectly classified called false positive (FP); and the number of negative class that incorrectly classified(FN). (Based on the confusion matrix, a series of key performance metrics are acquired, including precision, G-mean, F-measure, accuracy, sensitivity, and specificity. These measurements show the performance of the classification and allow the comparison of the multiple sets of features chosen when selecting the features. Accuracy measures the algorithm's capacity to predict the class accurately by finding the percentage of predictions that where correct [16, 17]. The precision relates to the proximity between two or more me asurements. The sensitivity is a test's capacity to correctly clas sify an individual for a desirable class, while the specificity is a test's capacity to classify an individual for a negative class correctly. G-mean is the prediction precision product for both classes calculated. Low G-mean value will be achieved with lower results in the prediction of positive examples. The G-mean is therefore essential to assess the degree to which the positive class is ignored, in order to avoid an over-fit to the negative class. The F-Meter combines accuracy with a reminder of the positive class prediction. A greater F-method shows that FP and FN are better balanced by the positive class [18]. The formulas are:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Sensitivity\ (Recall) = \frac{TP}{TP+FN} \quad (3)$$

$$Specificity = \frac{TN}{TN+FP} \quad (4)$$

$$F-Score = \frac{2 \times Precision \times Recall}{Precision+Recall} \quad (5)$$

$$G-mean = \sqrt{\frac{TN}{TN+FP} \times \frac{TP}{TP+FN}} \quad (6)$$

3.4. EXPERIMENTAL MEASURES (PREDICTION PERFORMANCE)

Numerous of classification methods can be used to measure the performance of learning algorithms. In this paper, we try to use different classifier techniques, such as:

1. Naïve Bayes: Probabilistic classifier that uses theorem Bayes with the premise of independence between features. Clearly stated, a classifier for Naive Bayes assumes that the presence of a particular function in a class does not seem to do with the other attributes. Naive Bayes model for large input sets is simple to design and notably convenient. Naive Bayes provides even highly advanced classification tools in addition to simplicity.
2. Bagging: Bagging: it is a random process that produces the ensemble and then utilizes the combination method to decrease the final output variance it is a [19]. Giving the original training set D we create the raining set $D_m, \in (1,2,\dots,m)$ by randomly sampling D with replacement. Whereas each training set D_m, only contains two-thirds of the original samples [20]. Therefore, the algorithm uses separate datasets for practice. The bootstrapping and averaging scheme provides and shares multiple predictor variations. Bootstrap measurements are individual base classifiers that were trained in various training sets individually. The Bootstrap tests are conducted by randomly selecting some examples with the substitution of the initial samples[21].
3. J48, it improves the ID3 algorithm, the improvement of ID3 includes characteristics such as velocity, size, memory, and a rule-specific performance [22, 23]. The steps to the algorithm [24, 25] are:
   a. For instance, in the same class, the tree represents a leaf, and the leaf is marked with the same class.
   b. The data for each variable is calculated, the data chosen by an attributes test, and then the obtained data chosen by the variable test is calculated.
   c. Apply the current selection principle in order to find the highest gain of data; the variable for branching is chosen.
4. Random Forest (RF): RF comprises of numerous

random data and attributes. The n-separate variabilities may be used to make a decision tree, the variables are randomly chosen into sets, and such random decision-making collections generate a forest[26]. The advantage of the great number of trees is that most trees can predict the class correctly. Another point is that no error is made by all trees in the same location[27]. As it is taken as a conjunction of more than one classifier, the final classification obtains precise outcomes [28].

5. REPTree: a quick training decision-tree. It makes an information gain decision tree and uses reduced error cutting to cut it. The values of numeric features are sorted once at the start of model preparation [29, 30]. In many iterations, the above technique produces many trees and then chooses the best tree representative. For prediction, pruning utilizes a square error [31].

3.5. Results and Discussion

Resampling imbalance class were implemented in this research; they are random resampling without replacement and random resampling with replacement, to find the influence of balance data on improving the class prediction. The next step is to find the performance of the classification using different classification techniques, such as Naïve Bayes, Bagging, J48, Random Forest, and REPTree. Tables 2 illustrates the results of performance measures such as accuracy, specificity, sensitivity, precision f-score, and G-mean for some classification methods applying on imbalanced data. Table 3 shows the outcomes when applying on balance data after resampling with replacement technique. Table 4 shows the results of accuracy, specificity, sensitivity, and PPV for the classification methods applying on balance data after resampling without replacement technique. From the tables, we notice that resampling imbalance data without replacement has little effect on the performance outcome for all the classification methods used. Whereas the output of classification methods implemented on balance data that resampling with replacement has been improved for most of the classification methods used. In the imbalanced data, the best performance results come from using RF and Bagging methods for learning classification, while Naïve Bayes has the lowest outcome results. As we can see, all outcomes have approximately the same poor performance results. In contrast, the outcomes of classification performance for the same data after resampling have noticeable variations. The highest improvement for using RF and RT were the performance has jumped from 67% to 87.4% and from 62% to 85.4%, respectively. Because RF suffers from an imbalanced data learning curse. Since this is designed to minimize the total error rate, the predicted accuracy of the majority class tends to be more crucial, which often leads to reduced accuracy of the minority class. While using J48 and Bagging to classify balancing data has improved the prediction but less than RF and RT due to the incorrect classification of negative examples. In contrast, Naïve Bayes didn't improve the prediction performance for the balance class due to the increase in the negative, because the resampling decrease the number of predicted positive classes which then effect on the accuracy and false-positive rate was other measures keeps the same.

Table 2. *The Prediction Performance Results for Different Classification Algorithms Implemented on the research Dataset.*

	Accuracy	Sensitivity	Specificity	Precision	f-score	G-mean
Naïve Bayes	56.82%	95.37%	84.76%	52.18%	67.45%	70.54%
Bagging	66.12%	66.30%	68.89%	63.25%	64.74%	64.76%
J48	64.38%	67.96%	68.37%	60.76%	64.16%	64.26%
RF	69.24%	72.78%	73.32%	65.50%	68.95%	69.04%
RT	60.38%	56.30%	62.36%	58.02%	57.14%	57.15%
REPTree	64.47%	69.26%	68.91%	60.62%	64.65%	64.79%

Table 3*. The Prediction Performance Results for Different Classification Algorithms Implemented on the research Dataset after Resampling Data using Replacement Technique.*

With replacement	Accuracy	Sensitivity	Specificity	Precision	f-score	G-mean
Naïve Bayes	56.99%	96.39%	88.62%	51.63%	67.24%	70.54%
Bagging	80.36%	81.02%	83.28%	77.22%	79.07%	79.10%
J48	78.97%	76.85%	80.51%	77.14%	77.00%	77.00%
RF	86.88%	88.24%	89.61%	83.94%	86.03%	86.06%
RT	83.49%	82.16%	84.89%	81.85%	82.01%	82.01%
REPTree	74.28%	75.14%	77.80%	70.59%	72.79%	72.83%

Table 4*. The Prediction Performance Results for Different Classification Algorithms Implemented on the research Dataset after Resampling Data without Replacement Technique.*

Without replacement	Accuracy	Sensitivity	Specificity	Precision	f-score	G-mean
Naïve Bayes	56.82%	95.37%	84.76%	52.18%	67.45%	70.54%
Bagging	64.38%	67.96%	68.37%	60.76%	64.16%	64.26%
J48	69.24%	72.78%	73.32%	65.50%	68.95%	69.04%
RF	60.38%	56.30%	62.36%	58.02%	57.14%	57.15%
RT	64.47%	69.26%	68.91%	60.62%	64.65%	64.79%
REPTree	56.82%	95.37%	84.76%	52.18%	67.45%	70.54%

4. CONCLUSIONS

In practical terms, we have discussed the imbalanced class issue using Diabetes data. The imbalanced class was manipulated by replacement and no substitution and Naïve Bayes and Bagging, J48, RF, and RT were used as a training sample. Our results from this paper show that the handling of an imbalanced replacement dataset is vital for most classification methods Naïve Bayes expects. It seems that imbalanced classes give samples that are unequal to overview groups of examples in clinical fields of data classification. Likewise, the findings show that resampling of the imbalanced class results in factors like accuracy, specificity, sensitivity, f-score, and g-measurement have a decent performance improvement. In contrast, using resampling without replacement to balance the class distributions, the results were no deference compared with resampling with

replacement. The g-measure and f-score show that the performance in the prediction of the positive prediction and negative.

REFERENCES

[1] S. Batra, H. J. Parashar, S. Sachdeva, and P. Mehndiratta, "Applying data mining techniques to standardized electronic health records for decision support," in 2013 Sixth International Conference on Contemporary Computing (IC3), 2013: IEEE, pp. 510-515.

[2] D. J. Hand, H. Mannila, and P. Smyth, Principles of data mining (adaptive computation and machine learning). MIT Press, 2001.

[3] G. Potamias and V. Moustakis, "Knowledge discovery from distributed clinical data sources: the era for internet-based epidemiology," in 2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2001, vol. 4: IEEE, pp. 3638-3641.

[4] G. Menardi and N. Torelli, "Training and assessing classification rules with imbalanced data," Data Mining and Knowledge Discovery, vol. 28, no. 1, pp. 92-122, 2014.

[5] P. Cao, X. Liu, J. Zhang, D. Zhao, M. Huang, and O. Zaiane, "ℓ 2, 1 norm regularized multi-kernel based joint nonlinear feature selection and over-sampling for imbalanced data classification," Neurocomputing, vol. 234, no. 19 April 2017, pp. 38-57, 2016.

[6] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2016.

[7] D. G. Horvitz and D. J. Thompson, "A generalization of sampling without replacement from a finite universe," Journal of the American statistical Association, vol. 47, no. 260, pp. 663-685, 1952.

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," Journal of artificial intelligence research, vol. 16, pp. 321-357, 2002.

[9] O. Loyola-González, M. A. Medina-Pérez, J. F. Martínez-Trinidad, J. A. Carrasco-Ochoa, R. Monroy, and M. García-Borroto, "PBC4cip: A new contrast pattern-based classifier for class imbalance problems," Knowledge-Based Systems, vol. 115, pp. 100-109, 2017.

[10] P. Cao, X. Liu, J. Zhang, D. Zhao, M. Huang, and O. Zaiane, "ℓ2, 1 norm regularized multi-kernel based joint nonlinear feature selection and over-sampling for imbalanced data classification," Neurocomputing, vol. 234, pp. 38-57, 2017.

[11] A. Al-Shahib, R. Breitling, and D. Gilbert, "Feature selection and the class imbalance problem in predicting protein function from sequence," Applied Bioinformatics, vol. 4, no. 3, pp. 195-203, 2005.

[12] R. Batuwita and V. Palade, "Efficient resampling methods for training support vector machines with imbalanced datasets," in The 2010 International Joint Conference on Neural Networks (IJCNN), 2010: IEEE, pp. 1-8.

[13] P. B. andLuis Torgo and R. Ribeiro, "A survey of predictive modeling under imbal-anced distributions," ACM Comput. Surv, vol. 49, no. 2, pp. 1-31, 2016.

[14] V. López, A. Fernández, and F. Herrera, "On the importance of the validation technique for classification with imbalanced datasets: Addressing covariate shift when data is skewed," Information Sciences, vol. 257, pp. 1-13, 2014.

[15] B. Antal and A. Hajdu, "An ensemble-based system for microaneurysm detection and diabetic retinopathy grading," IEEE transactions on biomedical engineering, vol. 59, no. 6, pp. 1720-1726, 2012.

[16] H. Chauhan, V. Kumar, S. Pundir, and E. S. Pilli, "A comparative study of classification techniques for intrusion detection," in 2013 International Symposium on Computational and Business Intelligence, 2013: IEEE, pp. 40-43.

[17] M. Al Khaldy and C. Kambhampati, "Performance Analysis of Various Missing Value Imputation Methods on Heart Failure Dataset," in Proceedings of SAI Intelligent Systems Conference, 2016: Springer, pp. 415-425.

[18] S. Hido, H. Kashima, and Y. Takahashi, "Roughly balanced bagging for imbalanced data," Statistical Analysis and Data Mining: The ASA Data Science Journal, vol. 2, no. 5-6, pp. 412-426, 2009.

[19] C. C. Aggarwal and S. Sathe, Outlier ensembles: An introduction. Springer, 2017.

[20] U. R. Salunkhe and S. N. Mali, "Classifier ensemble design for imbalanced data classification: a hybrid approach," Procedia Computer Science, vol. 85, pp. 725-732, 2016.

[21] G. L. Agrawal and H. Gupta, "Optimization of C4. 5 decision tree algorithm for data mining application," International Journal of Emerging Technology and Advanced Engineering, vol. 3, no. 3, pp. 341-345, 2013.

[22] P. Sharma, D. Singh, and A. Singh, "Classification algorithms on a large continuous random dataset using rapid miner tool," in 2015 2nd International Conference on Electronics and Communication Systems (ICECS), 2015: IEEE, pp. 704-709.

[23] G. Kaur and A. Chhabra, "Improved J48 classification algorithm for the prediction of diabetes," International Journal of Computer Applications, vol. 98, no. 22, 2014.

[24] A. Almutairi and D. Parish, "Using classification techniques for creation of predictive intrusion detection model," in The 9th International Conference for Internet Technology and Secured Transactions (ICITST-2014), 2014: IEEE, pp. 223-228.

[25] A. Galathiya, A. Ganatra, and C. Bhensdadia, "Classification with an improved decision tree algorithm," International Journal of Computer Applications, vol. 46, no. 23, pp. 1-6, 2012.

[26] J. Xu, J. Chen, and B. Li, "Random forest for relational classification with application to terrorist profiling," in 2009 IEEE International Conference on Granular Computing, 2009: IEEE, pp. 630-633.

[27] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston, "Random forest: a classification and regression tool for compound classification and QSAR modeling," Journal of chemical information and computer sciences, vol. 43, no. 6, pp. 1947-1958, 2003.

[28] A. Cuzzocrea, S. L. Francis, and M. M. Gaber, "An information-theoretic approach for setting the optimal number of decision trees in random forests," in 2013 IEEE International Conference on Systems, Man, and Cybernetics, 2013: IEEE, pp. 1013-1019.

[29] W. N. H. W. Mohamed, M. N. M. Salleh, and A. H. Omar, "A comparative study of reduced error pruning method in

decision tree algorithms," in 2012 IEEE International conference on control system, computing and engineering, 2012: IEEE, pp. 392-397.

[30] A. Balasundaram and P. Bhuvaneswari, "Comparative study on decision tree based data mining algorithm to assess risk of epidemic," 2013.

[31] J. Park, H.-R. Tyan, and C.-C. J. Kuo, "Ga-based internet traffic classification technique for qos provisioning," in 2006 International Conference on Intelligent Information Hiding and Multimedia, 2006: IEEE, pp. 251-254.