# Ensemble Learning with Resampling for Imbalanced Data

Firuz Kamalov[1(✉)], Ashraf Elnagar[2], and Ho Hon Leung[3]

[1] Faculty of Electrical Engineering, Canadian University Dubai, Dubai, UAE
firuz@cud.ac.ae
[2] Department of Computer Science, University of Sharjah, Sharjah, UAE
ashraf@sharjah.ac.ae
[3] Department of Mathematics, UAE University, Al Ain, UAE
hohon.leung@uaeu.ac.ae

**Abstract.** Imbalanced class distribution is an issue that appears in various applications. In this paper, we undertake a comprehensive study of the effects of sampling on the performance of bootstrap aggregating in the context of imbalanced data. Concretely, we carry out a comparison of sampling methods applied to single and ensemble classifiers. The experiments are conducted on simulated and real-life data using a range of sampling methods. The contributions of the paper are twofold: i) demonstrate the effectiveness of ensemble techniques based on resampled data over a single base classifier and ii) compare the effectiveness of different resampling techniques when used during the bagging stage for ensemble classifiers. The results reveal that ensemble methods overwhelmingly outperform single classifiers based on resampled data. In addition, we discover that NearMiss and random oversampling (ROS) are the optimal sampling algorithms for ensemble learning.

**Keywords:** Imbalanced data · Undersampling · Oversampling · Ensemble method · Data preprocessing sampling

## 1 Introduction

Imbalanced class distribution refers to a situation where one class considerably outnumbers another class. It appears in a variety of contexts including text classification, medical diagnostics, fraud detection and many others involving rare events. Skewed class distribution causes bias against the minority class in learning models. In particular, the prediction accuracy is often higher on the majority class relative to the minority class [23]. There exists a variety of approaches to deal with imbalanced data including feature selection, cost-sensitive learning, one-class learning, and others. One of the most popular such approaches is sampling the data to balance the class distributions. However, the use of sampling alone may result in a high variance classifier. To reduce the variance researchers have employed ensemble methods. In an ensemble method, the data is repeatedly sampled to obtain a collection of balanced datasets which are used to train base classifiers (weak learners). Then an ensemble rule is applied to aggregate the predictions of the base classifiers into a single response. In a


D.-S. Huang et al. (Eds.): ICIC 2021, LNCS 12837, pp. 564–578, 2021.
https://doi.org/10.1007/978-3-030-84529-2_48

basic ensemble method for imbalanced class distribution, the majority class is repeatedly undersampled to match the size of the minority data and a corresponding decision tree is constructed based on the balanced bootstrap sample. The process is carried multiple times depending on the user preference resulting in a collection of decision trees. Then the predictions of the resulting ensemble method are based on the majority or the mode of predictions of the constituent decision trees. The use of multiple tree to make predictions reduces the variance of the classifier. Since decision trees are very efficient algorithms ensemble methods do not experience any significant deterioration in execution time.

Ensemble learning with sampling for imbalanced data has been an active area of research with several authors proposing their own methods to combine ensemble and sampling techniques to improve classification performance. The goal of this paper is to carry out a comprehensive study of the effects of ensemble learning in regards to sampling for imbalanced data. Concretely, for each sampling technique, we compare classification performance between individual and ensemble tree classifiers. We consider a range of undersampling and oversampling techniques including random undersampling (RUS), NearMiss, random oversampling (ROS), synthetic minority oversampling technique (SMOTE), and ADASYN. To obtain broadly applicable results we carry out multiple numerical experiments using both simulated and real-life data. The real life-data covers a range of applications including astronomy, social science, medical diagnostics, and image recognition.

It is well known that bootstrap aggregating methods outperform single decision trees on balanced data. However, there does not exist an extensive study on bootstrap methods in the context of imbalanced data. The performance of a classifier on a balanced dataset is measured by its overall accuracy rate. By aggregating the predictions from a collection of classifiers an ensemble method reduces the variance of the predictions. As a result, it achieves improved accuracy on the testing set. On the other hand, the performance of a classifier on an imbalanced set is measured by AUC and F1-score. It is not immediately clear that the gains obtained by an ensemble classifier in accuracy rate will also materialize in AUC and F1-score. Therefore, a separate study is required to investigate the effects of bootstrap aggregating on the AUC and F1-score in the context of imbalanced data.

To fill the gap in the literature we test a range of sampling methods applied to a collection of simulated and real-life data. The results of the experiments show that ensemble methods consistently outperform single classifiers. In particular, we find that ensemble classifier outperforms single classifier on all tested datasets when using undersampling techniques. The ensemble classifier similarly produces better results when using oversampling techniques on the simulated data and 4 out 5 real-life datasets.

The paper is organized follows. In Sect. 2, we briefly review the current literature on ensemble methods and imbalanced data. In Sect. 3, we discuss the methodology and the results of the numerical experiments. Section 4 concludes the paper with a summary of our findings.

## 2 Literature Review

There exists a range of approaches in the literature to combat imbalance class distribution [17]. One of the approaches is based on selecting the optimal feature subset for identifying the minority class points [2, 3]. Another approach is based on balancing the class distribution in the dataset. Balancing the class distribution through resampling is arguably the most popular approach to dealing with imbalanced data. Resampling methods can be divided into two groups: undersampling and oversampling. Undersampling techniques consist of selecting a subset of the majority class that is of the same size as the minority class. The RUS algorithm is the simplest undersampling technique whereby a subset of the majority class is selected with uniform probability. In more intelligent approach called NearMiss the points of the majority class that are close to the border with the minority class are more likely to be selected [20]. Oversampling techniques consist of creating new minority points based on the existing minority points. A popular oversampling technique called SMOTE generates new points by linear interpolation between the existing minority points [4, 9]. An extension of the SMOTE algorithm called ADASYN generates new points in the same fashion as SMOTE but with greater emphasis on the minority points that lie in regions with high concentration of the majority class points [12]. In a more sophisticated approach, the authors in [14] employ kernel density estimation to estimate the underlying distribution of the existing minority points. The estimated distribution is used to generate the new minority points. Fusion approaches that combine multiple methods have also been used to deal with imbalanced data [25, 26]. As an alternative to resampling, other approaches such as weighted misclassification penalty and features selection can be employed to combat imbalanced data [16].

It has been widely accepted that ensemble approach reduces the variance of an estimator by introducing randomization into its construction procedure and then aggregating individual estimators. Ensemble methods such as bagging and random forest have been successfully modified to fit imbalanced data [19]. Each method applies a particular sampling technique during the bagging/boosting stage to balance the data. The most popular approach to constructing a bagging classifier is by random undersampling of the majority class to obtain a balanced subset which is used to train an estimator. An ensemble of estimators is created by repeating the sampling and training procedure. There exists several extensions of the random undersampling ensemble method. The authors in [10], propose a variation of the underbagging ensemble method by ap- plying evolutionary undersampling on the majority class. According to the proposed method new subsets of the majority class are sampled using evolutionary approach and base classifiers are constructed. The resulting ensemble method performs well on highly imbalanced data. Hido et al. [13] introduced Roughly Balanced Bagging (RBB), where the numbers of instances for both classes are determined in different ways. The number of minority points in each bootstrap set equals the size of the minority class while the number of majority points is determined according to the negative binomial distribution. RBB has been a popular ensemble learning tool that has been applied to various contexts [18]. Diez-Pastor et al. [6] proposed an ensemble method based on bootstrap sets with random class ratios. Each base classifier in the

ensemble is trained on a dataset with arbitrarily chosen class ratio. The proposed method aims to increase the resulting AUC. The authors in [5] propose an ensemble learning technique based on the threshold moving technique which applies a threshold to the continuous output of a model. The proposed method preserves the natural class distribution of the data resulting in well-calibrated posterior probabilities. The authors in [22] consider adjusting the existing ensemble rules to account for data imbalance. In [24], the authors explore the relationship between the diversity of the base classifiers in an ensemble and the performance of the final ensemble. Investigation of three ensemble approaches based on undersampling, oversampling and SMOTE reveal a positive relationship between the diversity and recall rates on the minority class.

In [27], the authors used one-vs-one (OVO) approach together with ensemble methods to tackle imbalanced distribution for multi-class data. An empirical study of various ensemble methods indicates the high effectiveness of ensemble learning with OVO scheme in dealing with the multi-class imbalance classification problems. Concretely, the authors find that decision tree-based ensemble classifier SMOTEBoost achieves average accuracy of 0.7676 while neural network-based classifier SMOTE + AdaBoost achieves average accuracy of 0.7915. The authors in [1], build on the previous work by considering different base classifiers on each subset of the OVO decomposition. Thus, a different base model is selected for each data subset in the ensemble classifier. In addition, the authors replace the OVO strategy with Error Correcting Output Codes. The test results show that the proposed method produces a higher F1-score compared to the other 17 benchmark algorithms. Concretely, the average F1-score for the proposed method is 0.7750 while the accuracy is 0.9323.

## 3 Sampling Techniques

In this section we present the sampling algorithms used to balance data with skewed class distribution. There are two types of sampling methods: undersampling and oversampling. In undersampling, a subset of the majority class, of the same size as the minority class, is selected (Fig. 1, top). There exists a number of undersampling techniques in the literature. The simplest undersampling technique - the RUS algorithm - consists of randomly selecting a subset of the majority class with or without replacement. The main advantage of RUS is its simplicity. However, it fails to take full advantage of the data by underutilizing the majority class subset. In a more sophisticated algorithm called NearMiss, the majority class instances that are closest to the minority class are more likely to be selected [20]. Concretely, pairwise distances between members of the majority and minority classes are calculated. Then for each point in the majority class $p^+$, the average distance to the closest $k$ points of the minority class $\overline{d}_{p+}$ is calculated. The points $p^+$ with the corresponding smallest values of $\overline{d}_{p+}$ are selected as the majority class sample. The motivation for the NearMiss algorithm is that given more points in the border region a classifier will be more likely to separate the two classes along the border. Despite its intelligent approach to undersampling it similar to RUS - also fails to take full advantage of the data.

In oversampling, the existing minority class points are used to generate the new minority points (Fig. 1, bottom). The simplest oversampling approach - the ROS algorithm - consists of randomly selecting (with replacement) from the existing minority points. The main advantage of ROS is its simplicity. However, it leads to overfitting by generating several minority points in the same location. In a more creative approach called SMOTE the new points are created synthetically by interpolating between the existing minority points [4]. Concretely, for each point in the minority class its k nearest neighbors in the minority class are determined. Given a minority point and one of its neighbors, the difference between the two is calculated and multiplied by a random number between 0 and 1. The new minority class instance is obtained by adding the preceding result to the minority point:

$$p_{new}^{-} = p^{-} + t\left(p^{-} - p_{k}^{-}\right), \tag{1}$$

where $p_{new}^{-}$ denotes the new minority class point, $p^{-}$ is the minority point under consideration, $p_{k}^{-}$ is a kth nearest neighbor, and t is a random number between 0 and 1. Although SMOTE generates minority points in new locations it is bound to only linear paths between neighboring minority points which restricts the range of points that can be generated. To combat the issue of restricted linear generation other methods employing KDE and Gamma distribution have been proposed recently [15]. An extension of SMOTE called Adaptive Synthetic (ADASYN) attempts to generate the new minority points around the minority points that are harder to learn [12]. Concretely, ADASYN employs the ratio of the majority to minority points in the neighborhood of an existing minority point to determine the number of new minority points to generate in that neighborhood. ADASYN is motivated by the logic that the minority points that lie in regions of high concentration of majority points are more likely to be ignored by a classifier.

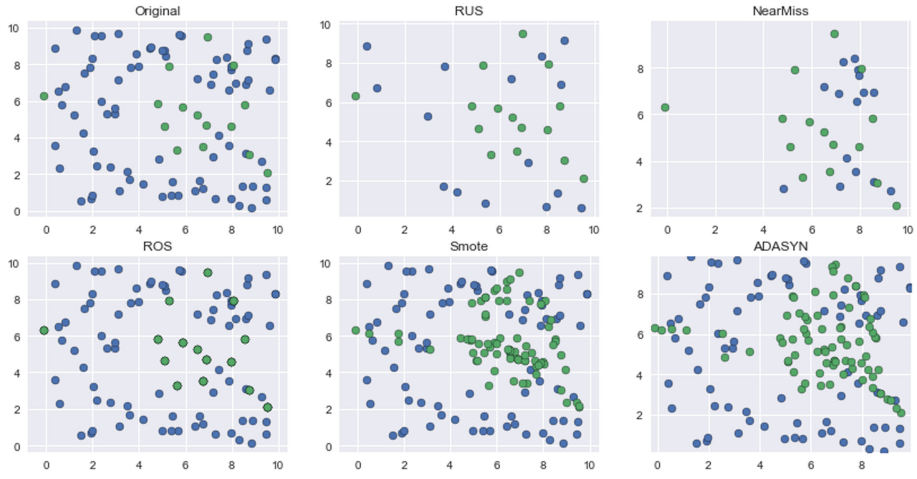


**Fig. 1.** Illustration of the sampling algorithms.

## 4 Numerical Experiments

In this section, we present the results of the numerical experiments that were carried out to compare single and ensemble classifiers in the context of imbalanced class distribution. In our experiments we use both simulated and real-life data. The simulated data contains 10,000 instances with 100 features while the real-life data is comparatively smaller.

### 4.1 Experimental Design

Constructing a single decision tree classifier for imbalanced data is straightforward. First, the original imbalanced data is resampled using an appropriate sampling technique in order to obtain a balanced dataset. Then a decision tree classifier is trained on the balanced data (Fig. 2).

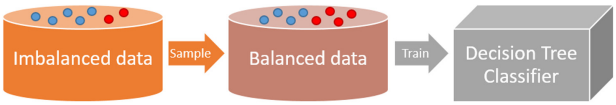


**Fig. 2.** Construction of a balanced decision tree classifier.

The ensemble classifier construction process for imbalanced data is illustrated in Fig. 3. To construct an ensemble classifier the original imbalanced data is resampled 50 times via an appropriate sampling technique. The resampling procedure produces a set of balanced datasets. We train a decision tree estimator on each balanced dataset. We obtain 50 individual decision tree estimators that are used to construct an ensemble classifier using the majority rule. Given new data, we make predictions using the individual estimators and select the most popular (majority) prediction as the final output of the ensemble classifier. During the experiments the data is divided into training and testing sets according to 75/25 ratio. The experiments are carried out in Python using machine learning libraries sklearn [21] and imblearn [19].

Traditional measures such as accuracy and error rate do not adequately capture the performance of a classifier on the minority class in the context of imbalanced data. Therefore, we use the F1-score to obtain a more unbiased measure of classifier performance. The F1-score is a balanced metric that combines precision and recall into a single value. Recall represents the fraction of positive instances that were correctly labeled as such. It is given by the equation

$$recall = \frac{tp}{tp + fn}, \tag{2}$$

where tp and fn denote the number of true positives and false negatives, respectively. Precision represents the fraction of truly positive instances from the total positively labeled instances,

$$precision = \frac{tp}{tp + fp}, \tag{3}$$

where fp denotes the number of false positives. The F1-score is the harmonic mean of precision and recall. It is given by the equation

$$F1 = 2\frac{precision.recall}{precision + recall}, \tag{4}$$

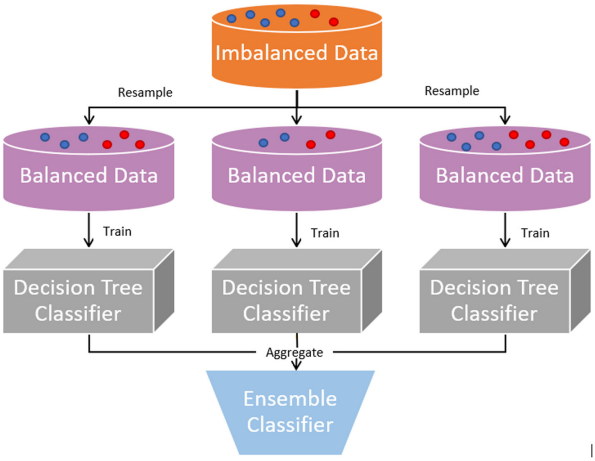


**Fig. 3.** Construction of a balanced ensemble decision tree classifier.

### 4.2 Simulated Data Experiments

We generate a random binary classification problem using the *make_classification* function from sklearn library. The dataset consists of a total of 100 features of which 15 are informative, 20 are redundant, 15 are repeated and the remaining are generated as random noise. The data is generated by placing clusters of points normally distributed (std = 1) about vertices of a 15-dimensional hypercube with sides of length 2 and assigning 2 clusters to each class. The redundant features are created as linear combinations of the informative features, while the repeated features are drawn randomly from the in- formative and redundant features [11]. The number of samples in the simulated dataset is 10,000. The ratio of minority to majority class points is 5/95.

We use the simulated dataset to examine the performance of ensemble and single decision tree classifiers in the context of various sampling algorithms. In particular, we investigate 2 undersampling and 2 oversampling algorithms: RUS, NearMiss, ROS, and SMOTE. The simulated data is balanced according to each sampling algorithm. We train ensemble and single decision tree classifiers on each balanced dataset. The performance of the classifiers is measured via the F1-score, recall, precision, and accuracy rates. As shown in Fig. 4, the ensemble classifier outperforms a single decision tree classifier in precision but underperforms it in recall. However, on balance

- as reflected by the F1-score - the ensemble classifier outperforms the single decision tree classifier with all 4 sampling algorithms. We also note that the ensemble classifier produces higher accuracy rate in almost all the tested sampling algorithms. Although accuracy rate is not the primary measure of performance for imbalanced data, it further supports the superiority of the ensemble approach. Observe that in the simulated data the average accuracy of ensemble classifier is 0.9619 which is higher than the benchmark score of 0.95 using the naive approach of classifying all instances as positive. Finally, we note that the ensemble approach produces better results with both undersampling (RUS, NearMiss) and oversampling (ROS, SMOTE) algorithms. The better performance in both the F1-score and accuracy indicates that the ensemble method is effective at identifying the minority as well as the majority class points. Identifying the minority (positive) instances is particularly important in many applications such as medical diagnostics. Thus, given an imbalanced data with a coherent underlying structure a sampling technique coupled with an ensemble method seems to be an effective solution.

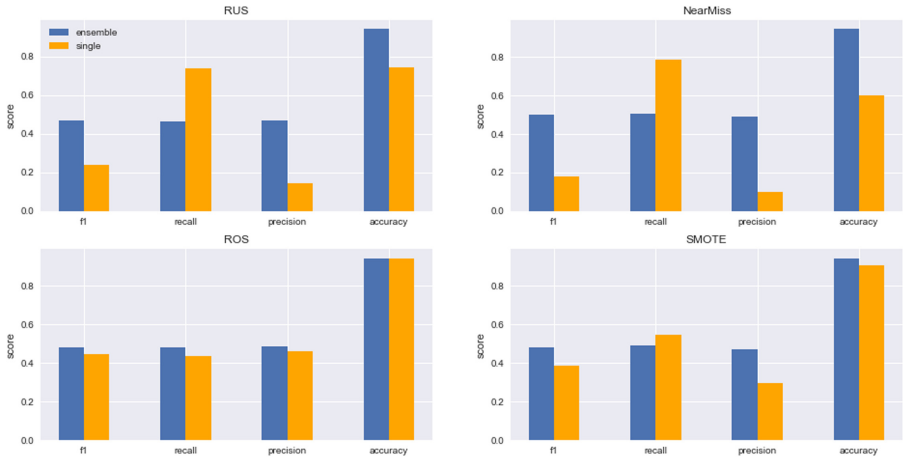


**Fig. 4.** Performance comparison of the ensemble and single decision tree classifier on simulated data consisting of 10,000 samples with 100 features.

**Table 1.** Experimental datasets.

| Name | Repository & Target | Ratio | #S | #F |
|---|---|---|---|---|
| 1 ecoli | UCI target: imU | 8.6:1 | 336 | 7 |
| 2 spectometer | UCI, target: > = 44 | 11:1 | 531 | 93 |
| 3 us crime | UCI, target: >0.65 | 12:1 | 1,994 | 100 |
| 4 libras move | UCI, target: 1 | 14:1 | 360 | 90 |
| 5 letter_img | UCI, target: Z | 26:1 | 20,000 | 16 |
| 6 mammography | UCI, target: minority | 42:1 | 11,183 | 6 |

### 4.3 Real-Life Data Experiments

We use a number of real-life datasets to compare the performance of ensemble and single tree classifiers in imbalanced class setting. The data is fetched through imblearn library where it is appropriately preprocessed. Alternatively, it can be obtained directly from the UCI repository [7]. The details of the datasets are presented in Table 1. We use a diverse set of datasets in order to obtain comprehensive results. The datasets are chosen from a range of fields including astronomy, image recognition, medical diagnostics, and social science. The class ratio of the datasets ranges from 8.6:1 to 42:1, the number of samples ranges from 336 to 20,000, and the number of features ranges from 6 to 100.

We begin our experiments with the RUS algorithm. Recall that RUS operates by randomly selecting a subset of the majority class to balance with the minority class. We apply RUS to balance each dataset given in Table 1. Then, we train an ensemble and single decision tree classifiers on the balanced sets as described in Sect. 4.1. The performance of the classifiers is measured by the F1-score, recall, precision, and accuracy rates. The results of the experiments are presented in Fig. 5. As can be observed from the figure, the ensemble classifier outperforms the single decision tree classifier in F1-score, precision, and accuracy rates for all the tested datasets. Note that the recall rate for single decision tree classifier is higher than the ensemble method. Recall is an important metric in imbalanced data classification - especially in applications such medical diagnostics, where effective discovery of positive instances is of great importance. Nevertheless, on balance - as reflected by the F1-score - the ensemble method produces superior results. The ensemble classifier achieves particularly impressive results in the *letter_img* dataset with the F1-score and accuracy rate near the perfect value of 1. Similarly, the ensemble method outperforms the single tree classifier on the *mammography* dataset by large margin. The strong performance on both the F1-score and accuracy suggest that the ensemble method does well on the minority and majority class data. Thus, if using the RUS algorithm on imbalanced data the ensemble method appears to be the optimal approach.

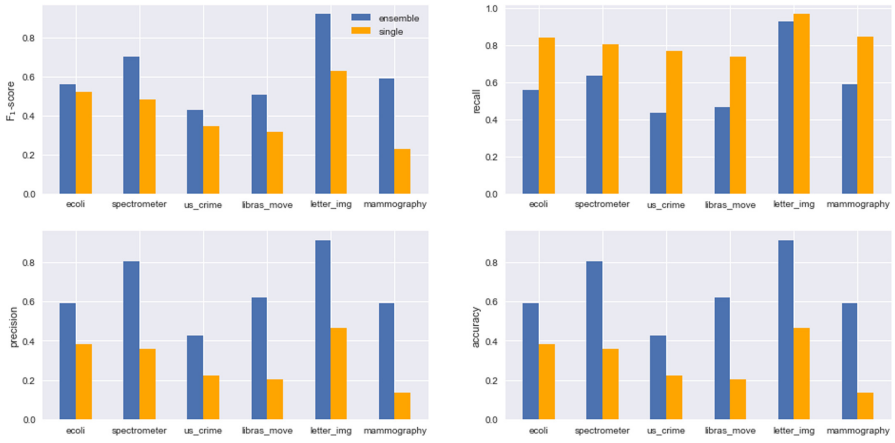


**Fig. 5.** Performance comparison of the ensemble and single decision tree classifier on simulated data consisting of 10,000 samples with 100 features.

Next, we compare the classification performance using the NearMiss algorithm. The NearMiss algorithm selects the samples of the majority class that are near the minority class points. As shown in Fig. 6, the ensemble classifier outperforms the single decision tree classifier on all the tested datasets. Concretely, the F1-score, precision, and accuracy of the ensemble classifier are substantially higher than a single tree classifier for all the tested datasets. The difference in the margin of the scores is particularly large in the case of spectrometer, letter_img, and mammography datasets. The results for the NearMiss algorithm are consistent with the RUS sampling algorithm.

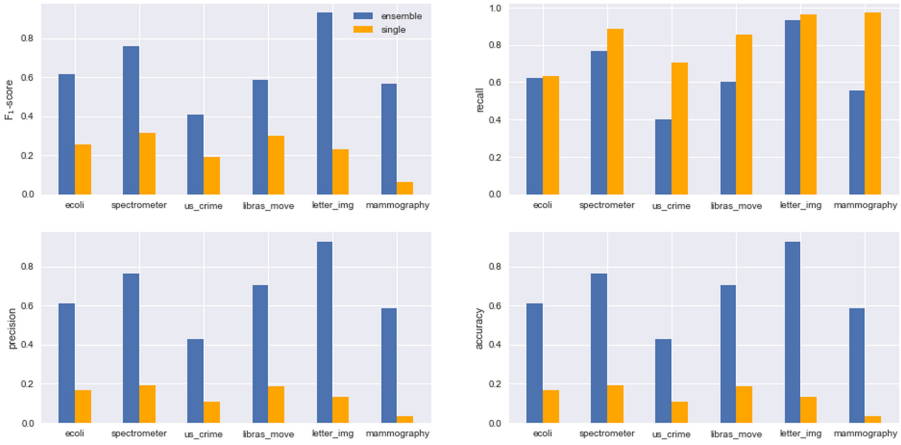


**Fig. 6.** Performance comparison of the NearMiss algorithm applied to ensemble and single decision tree classifiers.

We move on to investigate the performance of the ensemble and single tree classifiers using oversampling techniques. SMOTE is a popular oversampling algorithm that creates new minority samples through linear interpolation between existing minority points. As shown in Fig. 7, the ensemble classifier again outperforms the single decision tree classifier albeit in a slightly different manner than previously. In particular, the F1-score and accuracy of the ensemble method are higher on 5 out of 6 datasets. The results indicate that the ensemble method is effective in identifying the minority class instances while simultaneously producing strong outcomes on majority class data. Delving deeper into the results we see that the ensemble method produces better precision scores in all but one dataset. Unlike the case with the oversampling algorithms above, SMOTE-based ensemble classifier produces an even performance in recall rates. In particular, the ensemble classifier yields higher recall scores on 2 out 6 datasets. Given the overall results, as shown in Fig. 7, we conclude that the ensemble classifier is superior to the single decision tree classifier on SMOTE-sampled data.

Our final experiment is based on ADSYN sampled data. The ADASYN algorithm resembles SMOTE in the way it crates the new minority points through linear interpolation. The main difference is that ADASYN creates the new minority points around

the existing minority points with high concentration of the majority class points. In this way, ADASYN tries to account for the learning difficulty on each minority point. The performance of the ensemble and single decision tree classifier on ADASYN-sampled data is very similar to that of SMOTE-sampled data. In particular, as shown in Fig. 8, the ensemble classifier produces higher F1-score, precision, and accuracy rates on 5 out of 6 tested datasets. The recall performance of the ensemble classifier is even with the single decision tree classifier. Thus, on balance the ensemble classifier produces significantly better results.

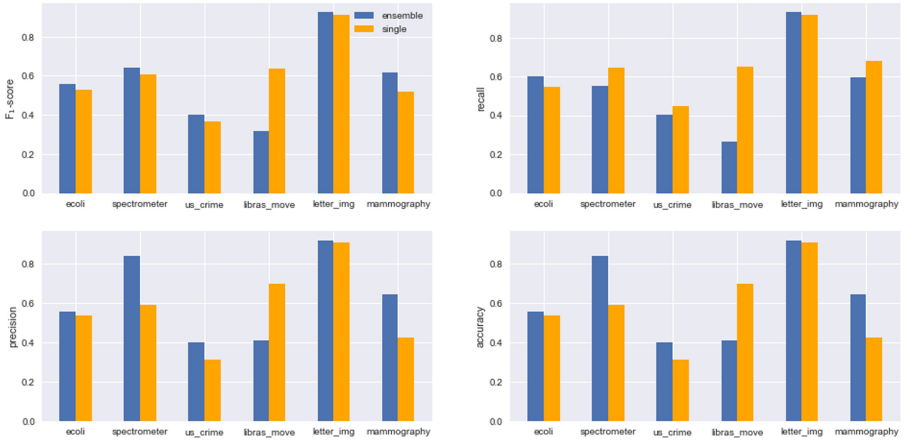


**Fig. 7.** Performance comparison of the SMOTE algorithm applied to ensemble and single decision tree classifiers.

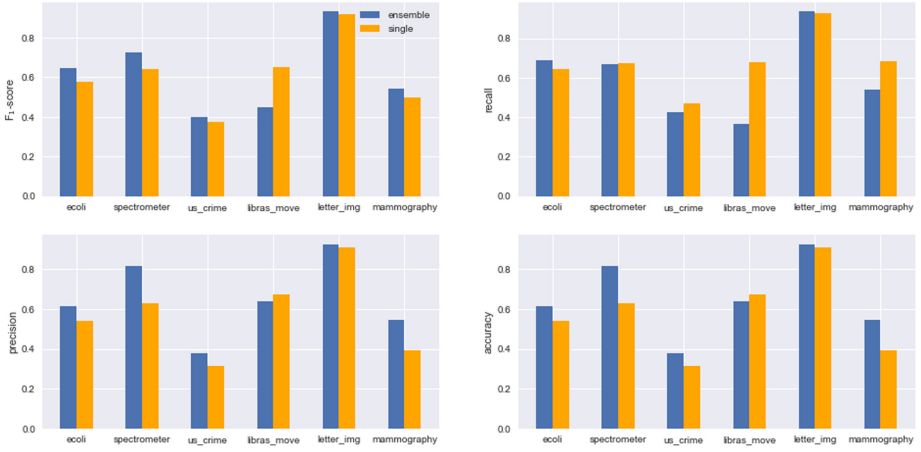


**Fig. 8.** Performance comparison of the ADASYN algorithm applied to ensemble and single decision tree classifiers.

Since they provide a way to reduce overfitting, bagging methods work best with strong and complex models such as fully developed decision trees. However, other base classifiers such as the k-nearest neighbors algorithm or logistic regression can also be employed as base classifiers. In Table 2, we present the average accuracy for ensemble and single classifier methods using 3 different base classifiers. The averages are calculated based on the results over the 6 datasets given in Table 1. It can be seen from the table that the ensemble classifiers outperform single classifiers for all 3 base classifiers. In addition, we observe that ROS and NearMiss achieve higher accuracy than other sampling methods in ensemble classification.

**Table 2.** Average accuracy using different base classifiers.

| Model | ROS | SMOTE | ADASYN | RUS | NearMiss |
|---|---|---|---|---|---|
| Ensemble DT | 0.9582 | 0.9532 | 0.9597 | 0.9577 | 0.9621 |
| Single DT | 0.9483 | 0.9367 | 0.9383 | 0.8397 | 0.5720 |
| Ensemble KNN | 0.9546 | 0.9483 | 0.9460 | 0.9474 | 0.9479 |
| Single KNN | 0.9436 | 0.9322 | 0.9342 | 0.8770 | 0.7534 |
| Ensemble LR | 0.9536 | 0.9473 | 0.9490 | 0.9494 | 0.9469 |
| Single LR | 0.9203 | 0.9235 | 0.9077 | 0.8852 | 0.7411 |

Similarly, in Table 3, we present the average F1-score for ensemble and single classifier methods. Although the results of the F1-scores are not exactly the same as the ac- curacy scores, they are generally consistent with our previous observations. Concretely, note that ensemble classifiers are generally better than single classifiers and that ROS and NearMiss are better sampling techniques in ensemble learning.

**Table 3.** Average F1-score using different base classifiers.

| Model | ROS | SMOTE | ADASYN | RUS | NearMiss |
|---|---|---|---|---|---|
| Ensemble DT | 0.6316 | 0.5770 | 0.6150 | 0.6181 | 0.6441 |
| Single DT | 0.6193 | 0.5953 | 0.6102 | 0.4204 | 0.2244 |
| Ensemble KNN | 0.6461 | 0.6303 | 0.6187 | 0.5944 | 0.6150 |
| Single KNN | 0.6991 | 0.6742 | 0.6635 | 0.4904 | 0.3608 |
| Ensemble LR | 0.6461 | 0.6303 | 0.6187 | 0.5944 | 0.6150 |
| Single LR | 0.5824 | 0.5951 | 0.5654 | 0.5041 | 0.3821 |

## 5 Conclusion

Imbalanced data is a widespread issue in a number of fields including medical diagnostics, text classification, fraud detection, and many others. Standard classifiers often struggle with imbalanced data by favoring the majority class data. However, the minority class data is often of more importance. For instance, it is more critical to identify fraudulent transactions than regular ones within credit card or insurance data.

One of the popular approaches to deal with imbalanced data is resampling whereby the original data is balanced prior to training a classifier. Resampling is naturally associated with ensemble classifiers as individual estimators of an ensemble can be trained on different iterations of resampled data. Therefore, we postulate that given a sampling procedure ensemble classifier would outperform single classifiers. To this end, we compared the performance of a single decision tree classifier to the performance an ensemble of decision trees in the context of applying a sampling procedure to imbalanced data. Concretely, investigated several undersampling and oversampling techniques and the performance of the classifiers on sampled data.

In order to obtain comprehensive results, we used simulated and real-life data. The simulated data (Sect. 4.2) consisted of 10,000 samples and 100 features of which only 15 were relevant. The results, as shown in Fig. 4, demonstrate the superiority of the ensemble classifier. The ensemble classifier outperformed the single classifier with respect to every sampling technique. In particular, the F1-score of the ensemble classifier exceeds that of the single decision tree classifier on every tested sampling technique. The results from the simulated data suggest that ensemble classifiers perform well on structured data in the context of resampled data.

We also investigated the performance single and ensemble classifiers on real-life resampled data. We used data from a range of applications to obtain a robust analysis. The experimental results, as presented in Sect. 4.3, show the superiority of the ensemble classifier. Using undersampling techniques the ensemble classifier yielded better F1- score on all 6 tested datasets. Using oversampling techniques, the ensemble classifier yielded better F1-score on 5 out of 6 tested datasets. Given the diversity of the tested datasets and sampling techniques, and the consistency of the results, we conclude an ensemble classifier is more suitable than a single classifier in the context of resampled data.

It is important to note that ensemble classifiers have a theoretically greater of algorithmic complexity. However, in practice the added training time is negligible because the underlying decision tree estimators are very efficient. Thus, in the light of the above discussion ensemble classifiers offer significantly better performance with little added cost.

As part of future work, our study can be extended to include multi-label imbalanced data. Using OVO or OVA approach multi-label data can be decomposed into a set of binary problems. Then an resampling-based ensemble approach can be applied to each binary problem. The effects of different resampling techniques on the performance of the corresponding ensemble methods would be interesting to study. In particular, applying these methods on high-dimensional big data can be a valuable addition to the existing literature.

## References

1. Bi, J., Zhang, C.: An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme. Knowl.-Based Syst. **158**, 81–93 (2018)

2. Bolon-Canedo, V., Sanchez-Marono, N., Alonso-Betanzos, A.: A review of feature selection methods on synthetic data. Knowl. Inf. Syst. **34**(3), 483–519 (2013)
3. Chandrashekar, G., Sahin, F.: A survey on feature selection methods. Comput. Electr. Eng. **40**(1), 16–28 (2014)
4. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
5. Collell, G., Prelec, D., Patil, K.R.: A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data. Neurocomputing **275**, 330–340 (2018)
6. Diez-Pastor, J.F., Rodrıguez, J.J., Garcıa-Osorio, C., Kuncheva, L.I.: Random balance: ensembles of variable priors classifiers for imbalanced data. Knowl. Based Syst. **85**, 96–111 (2015)
7. Dua, D., Graff, C.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA (2019). http://archive.ics.uci.edu/ml
8. Fawcett, T.: An introduction to ROC analysis. Pattern Recogn. Lett. **27**(8), 861–874 (2006)
9. Fernandez, A., Garcia, S., Herrera, F., Chawla, N.V.: Smote for learning from im- balanced data: progress and challenges, marking the 15-year anniversary. J. Artif. Intell. Res. **61**, 863–905 (2018)
10. Galar, M., Fernández, A., Barrenechea, E., Herrera, F.: EUSBoost: enhancing ensembles for highly imbalanced datasets by evolutionary undersampling. Pattern Recogn. **46**(12), 3460–3471 (2013)
11. Guyon, I., Gunn, S., Hur, A.B., Dror, G.: Design and analysis of the NIPS2003 challenge. In: Guyon, I., Nikravesh, M., Gunn, S., Zadeh, L.A. (eds.) Feature Extraction. Studies in Fuzziness and Soft Computing, vol. 207, pp. 237–263. Springer, Heidelberg (2006). https://doi.org/10.1007/978-3-540-35488-8_10
12. He, H., Bai, Y., Garcia, E.A., Li, S.: ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 1322–1328. IEEE (2008)
13. Hido, S., Kashima, H., Takahashi, Y.: Roughly balanced bagging for imbalanced data. Stat. Anal. Data Min. ASA Data Sci. J. **2**(5–6), 412–426 (2009)
14. Kamalov, F.: Kernel density estimation based sampling for imbalanced class distribution. Inf. Sci. **512**, 1192–1201 (2020)
15. Kamalov, F., Denisov, D.: Gamma distribution-based sampling for imbalanced data. Knowl. Based Syst. **207**, 106368 (2020)
16. Kamalov, F.: Sensitivity analysis for feature selection. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1466–1470. IEEE (2018)
17. Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. Progress Artif. Intell. **5**(4), 221–232 (2016). https://doi.org/10.1007/s13748-016-0094-0
18. Lango, M., Stefanowski, J.: Multi-class and feature selection extensions of roughly balanced bagging for imbalanced data. J. Intell. Inf. Syst. **50**(1), 97–127 (2018)
19. Lemaitre, G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning. J. Mach. Learn. Res. **18**(1), 559–563 (2017)
20. Mani, I., Zhang, I.: kNN approach to unbalanced data distributions: a case study involving information extraction. In: Proceedings of Workshop on Learning from Imbalanced Datasets, vol. 126 (2003)
21. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
22. Sun, Z., Song, Q., Zhu, X., Sun, H., Xu, B., Zhou, Y.: A novel ensemble method for classifying imbalanced data. Pattern Recogn. **48**(5), 1623–1637 (2015)

23. Thabtah, F., Hammoud, S., Kamalov, F., Gonsalves, A.: Data imbalance in classification: experimental evaluation. Inf. Sci. **513**, 429–441 (2020)
24. Wang, S., Yao, X.: Diversity analysis on imbalanced data sets by using ensemble models. In: 2009 IEEE Symposium on Computational Intelligence and Data Mining, pp. 324–331. IEEE (2009)
25. Yang, P., Liu, W., Zhou, B.B., Chawla, S., Zomaya, A.Y.: Ensemble-based wrapper methods for feature selection and class imbalance learning. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) PAKDD 2013. LNCS (LNAI), vol. 7818, pp. 544–555. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37453-1_45
26. Li, Y., Guo, H., Liu, X., Li, Y., Li, J.: Adapted ensemble classification algorithm based on multiple classifier system and feature selection for classifying multi-class imbalanced data. Knowl. Based Syst. **94**, 88–104 (2016)
27. Zhang, Z., Krawczyk, B., Garcia, S., Rosales-Perez, A., Herrera, F.: Empowering one-vs-one decomposition with ensemble learning for multi-class imbalanced data. Knowl. Based Syst. **106**, 251–263 (2016)