Handling imbalanced data with Random OverSampling Example: a Python implementation

Table of contents

1. Introduction

> *"It is the time you have wasted for your rose that makes your rose so important."* - Antoine de Saint-Exupery

Imbalanced learning refers to a classification or regression problems where the dataset classes are not represented equally. Common examples of such problems are churn, fraud and anomaly detection and clinical data, when one of the classes is rare because problematic, costly, unethical, dangerous to produce, or unexpected. Class unbalancing, specified as the proportion in the number of samples in different classes, can reach values in the orders of $10^2 \div 10^4 : 1$ and up to $10^5 : 1$ [1]

Fig. _ Examples of balanced and imbalanced datasets.

Most real datasets does not have exactly equal number of instances in each class, and while a small difference seldom matters, a heavy imbalance can quickly become a bottleneck in model training. Most learning methods have been conceived to identify the classification rules that better fits the data by some global accuracy criteria. Their target is to minimize the global error, that may be influenced enough from the minority class. Some methods, like the broadly used logistic regression, is more vulnerable to this effect, but even non-parametric methods, like trees, and association rules, are immune from this effect. For example tree generated from imbalanced datasets will have an high accuracy on the prevalent class and a very low precision in identifying the rare event. It appears evident how things become worse when the minority class is the event of interest, like a positive diagnosis of cancer in a patient.

A brief description of the sections of this work:

More data, less data. The most heard sentence in machine learning community is "You need more data!". Still, a large dataset might indeed expose a different, and perhaps more balanced perspective on the classes: more minority examples can indeed be useful. Other strategies may include considering more than once one or more minority samples. Chapter 1 will review the bibliography about solution for this problem, giving a view over cost-sensitive learning and different oversampling and undersampling methods, their advantages and disadvantage.

Random Over Sampling Examples In chapter 2 we will focus on one of these techniques, henceforth named just by its acronym ROSE, that propose a smart, albeit simple, way to generate new data from existing ones.

The Accuracy Paradox. To assess the performance of a solution a metric is needed. When a class represent almost the totality of a dataset, a learning algorithm can achieve a good accuracy by classifying every test sample as belonging to the majority class. To avoid this problem, different metrics has been developed to assess the real model utility and assessing capabilities. Chapter 3 will review available metrics that can be used to effectively evaluate performance of resampling methods.

A method is as good as it is available. To make ROSE easy to use, the main activity of this work involved incorporating it in the most used Python machine learning package, `scikit-learn` , and in particular in its sub-project `imbalanced-learn`. We will overview the development methods, CI/CD, software testing and documentation, in chapter 4.

But is it good? In chapter 5 will set up a wide testing framework for evaluating performance of resampling methods over 27 different famous standard datasets commonly used for classification problems. Different supervised models has been trained and tested on imbalanced and balanced data, and their performance reported. But toy datasets are usually easier to balance. In chapter 6 we used ROSE to dramatically improve classification capabilities of different models in a real-world dataset, with the aim of forecasting the economic outcome of small firms.

How can I use it? In appendix 1 we will show code snippets, use cases and links to repositories that will facilitate user's experience with ROSE, and guarantee repeatability of all experiments included in the present world.

2. Imbalanced learning

2.1. Imbalanced dataset problem

Despite the fact that in literature most imbalanced learning problems are traditionally referred to binary datasets, real world datasets can often be multiclass, as in microarray research[2], protein classification [3], medical diagnostics[4], activity recognition[5], target detection[6], and video mining[7]. Extending imbalanced classification to multi-class scenarios is a natural path, then. As the number of classes increases, so does the challenge of representing the whole problem space accurately, and the need for taking into account the presence of multi-minority and multi-majority classes [8].

In many problem, imbalancing is intrinsically tied to the nature of the data, and not due to lack of sampling, bias, or other sampling errors. In other cases no enough samples of a specific class exists at all.

Most learning methods' loss functions are supposed to be minimized globally, under the assumptions that all class have the same weight. When data are imbalanced, the learning process often achieves this objective by focusing on majority class, leading to bad performance [9], with higher errors on minority classes.

The lack of model effectiveness in prediction of rare classes has been deeply discussed in literature. Both parametric and non parametric methods appear to be sensitive to imbalancing. As an example in logistic regression, one of the most used for binary classification, this effect depends from an underestimation of conditional probability of the rare class [10],[11].

Even the more flexible non-parametric methods, like classification trees and association rules are immune from the effect of asymmetric class distribution. Trees, for example, are being grown finding the recursive divisions of the parameter space that maximize the impurity reduction. The imbalance found is the dataset will be often mirrored in the imbalance of the accuracy over different classes [12], [13]. Even association rules, being selected by their supports, tend to underperform[14],[15].

2.2. Treating imbalanced datasets

Many solution has been advanced to treat imbalanced data problems. Most fall in one of the following two approaches: using cost-sensitive learning models, and resampling the data.

2.2.1. Cost-sensitive learning

Cost sensitive learning is an umbrella term for algorithms in whose objective function it is possible to assign a different cost to misclassification of different classes. An intuitive example of this approach can be imagined when talking about a binary clinical cancer test: a false positive will lead to some extra exam, while a false negative will probably cost a life. The most logical decision is to estimate the relationship between these costs, and assign a larger (*hopefully, much larger*) cost to a false negative.

For multiclass data, a cost matrix $\mathbf{C}$ is computed, where $\mathbf{C}_{i,j}$ will be the cost of misclassifying a sample belonging to the class j as it were belonging to the class i [16], [17].

2.2.2. Resampling

A different, alternative approach against imbalancing can be tried by preprocessing the data, instead of modifying the learning rules, using sampling methods. This approach has consistently proven effective, with different degrees [18],[19]. Different resampling methods has been proposed, falling in two categories:

- undersampling methods, where majority class samples are being randomly discarded to remove imbalancing, at the price of sample size, in a non-heuristic way;
- oversampling methods, where different techniques can be used to generate new minority samples from the existing ones. The following sub-chapters (1.1.3 and 1.1.4) gives an overview of these methods.

Oversampling and undersampling presents different pro and cons, leading to the need of an empirical comparison between different methodologies.

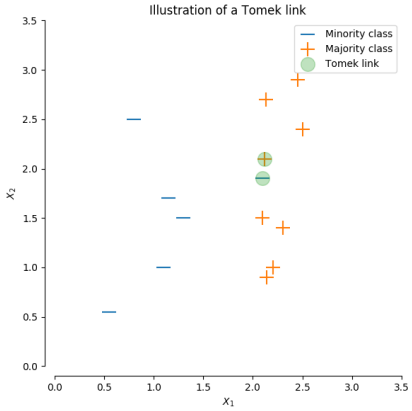
Methods	**Pros**	**Cons**
undersampling	faster learning	loss of sample size
oversampling	slower learning higher computational costs	introduction of artifacts possible overfitting

Despite those problems, resampling is more commonly used than cost-sensitive learning, that is not supported for all learning methods.

2.2.3. Undersampling strategies

Undersampling reduces the size of majority class to avoid imbalancing. In this paragraph we will provide an overview over commonly available undersampling strategies.

- **Random Undersampler**: it works by simply choosing randomly samples from the majority class.
- **Condensed NN**: [20] uses a 1-nearest neighbor rule to iteratively decide if a sample should be removed or not. It is sensitive to noise and will generate noisy samples.
- **One Sided Selection** [21]and **Tomek Links** instead tends to remove noisy samples.



- **Edited NN** and **Repeated Edited NN** [22] apply (respectively one of more times) a nearest-neighbors algorithm and "edit" the dataset by removing samples which do not agree "enough" with their neighborhood. For each sample in the class to be under-sampled, the nearest-neighbors are computed and if the selection criterion is not fulfilled, the sample is removed. The criterium can be based on majority, or totality of nearest neighbors belonging to the same class of the inspected sample to be kept in the dataset.
- **All KNN** is another iterative process that does the same of the latter, but incrementing at each iteration the number of considered neighbors.
- **Near Miss** [23] is a collection of three different algorithms that respectively:
    - selects the majority samples for which the average distance to the k *nearest* neighbors of the minority class is the *smallest*, or
    - selects the majority samples for which the average distance to the k *farthest* neighbors of the minority class is the *smallest*, or
    - first keel the M-nearest neighbors are kept, then, the majority samples selected are the one for which the average distance from the k nearest neighbors is the *largest*.
- **Neighborhood Cleaning Rule**[^Laurikkala, 2001] focuses on cleaning the data without condensing them.
- **Instance Hardness Threshold** [24] trains any classifier on the data, and the samples with lower probabilities are removed from the dataset. It is not guaranteed to output a balanced dataset, though.

2.2.4. Synthetic data generations

In this section we present the most commonly used oversampling techniques and their variants:

- Synthetic Minority Oversampling TEchnique (SMOTE) based
- ADAptative SYNthetic sampling (ADASYN).

2.2.4.1. SMOTE based methods

SMOTE[25] is a class of resampling algorithms that use the following approach:

- a random sample from the minority class is chosen
- his k-neighbors are found (default k=5)
- lines are drawn from the original sample to the neighbors
- new examples are drawn randomly along these lines, with $x_{new} = x_i + \lambda * (x_{nn} - x_i)$, where λ is drawn from $Uniform(0, 1)$, or other distributions.

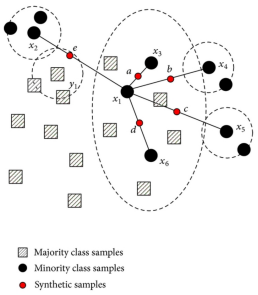


Fig. _ : SMOTE resampling concept.

There are many variants of SMOTE that has been developed to improve its performance.

Borderline SMOTE [26] will classify each sample x_i to be:

1. noise, when all k-neighbors are of a different class from x_i
2. in danger, when at least half of the neighbors belongs to the same class
3. safe, when all neighbors belongs to the same class.

The algorithm will then use "in danger" samples to generate new samples, with the same procedure of SMOTE.

K-Means SMOTE [27] uses a K-Means clustering method before to apply SMOTE. The clustering will group samples together and generate new samples depending of the cluster density.

SMOTENC [25] slightly change the way a new sample is generated by performing something specific for the categorical features. In fact, the categories of a new generated sample are decided by picking the most frequent category of the nearest neighbors present during the generation.

SVMSMOTE [28] uses a Support Vector Classifier to find support vectors and generate samples considering them. Tuning the C parameter of the SVM classifier allows to select more or less support vectors.

2.2.4.2. ADASYN

ADASYN [29] works similarly to the regular SMOTE. However, the number of samples generated for each x_i is proportional to the number of samples which are not from the same class than x_i in a given neighborhood. Therefore, more samples will be generated in the area where the nearest neighbor rule is not respected.

2.2.4.3. Combination and Ensemble methods

Combinations of different methods can be used efficiently. SMOTE based methods can generate noise when generating point between marginal outliers and inliers. After the resampling this issue can be solved by cleaning the space resulting from oversampling.

Two methods used for this purpose are:

- **Tomek's links**: [30] an undersampling technique used to remove unwanted overlaps between classes, where majority class links are removed until minimally-distanced neighbors pairs belong to the same class. Two instances form a Tomek's link if:
    - one of them is noise (*see Borderline SMOTE definition of noise)*, or
    - both are near a border

  In other words, if they are each other's closest neighbor, and of different classes.
- **Edited nearest-neighbors** [31] uses asymptotic convergence properties of nearest neighbor rules that use an editing procedure to reduce the number of preclassified samples and to improve performance [32]

Ensemble methods can be used generate undersampled subsets of many different oversampled datasets, or by bagging different undersamplers. Additionally, pipelines can be assembled, to chain different methods.

3. Random over-sampling examples (ROSE)

ROSE provides a different methodology to deal with imbalanced samples. As its alternatives do, it alters the distribution of the classes, using the following solution, based on the generation of new artificial data from the classes, according to a smoothed bootstrap approach [33]. It focuses on $\mathcal{X}$ domains included in $\mathbb{R}^d$, that is $P(\boldsymbol{x}) = f(\boldsymbol{x})$, a probability density function on $\mathcal{X}$. We consider that $n_j < n$ is the size of $\mathcal{Y}_j, j = 0, 1$. The ROSE procedure to generate a single new artificial sample consists in drawing a sample from $K_{\mathbf{H}_j}(\bullet, \mathbf{x}_i)$, with $K_{\mathbf{H}_j}$ a probability distribution centered at $\mathbf{x}_i$, and $\mathbf{H}_j$ a matrix of scale parameters, determining the width of the extracted sample neighborhood.

Usually $\mathbf{H}_j$ is chosen in the set of unimodal symmetric distributions. Once a class has been selected,

$$\begin{aligned}\hat{f}(\mathbf{x}|y = \mathcal{Y}_j) &= \sum_{i=1}^{n_j} p_i Pr(\mathbf{x}|\mathbf{x}_i) \\ &= \sum_{i=1}^{n_j} \frac{1}{n_j} Pr(\mathbf{x}|\mathbf{x}_i) \\ &= \sum_{i=1}^{n_j} \frac{1}{n_j} K_{\mathbf{H}_j}(\mathbf{x} - \mathbf{x}_i).\end{aligned}$$

such as, in this framework, the generation of new examples from the class $\mathcal{Y}_j$ will correspond to the generation of data from the kernel density estimate of $f(\mathbf{X}|\mathcal{Y}_j)$, to generate a new synthetic balanced training set $\mathbf{T}_m^*$. Usually m is set to the size of majority class, but can be set lower to perform under-sampling. The choice of K and $\mathbf{H}_j$ was addressed by a large specialized literature on kernel density estimation [34]. By choosing $\mathbf{H}_j \to 0$, ROSE collapses to a standard combination of over- and under-sampling.

Apart from enhancing learning, the generation of synthetic examples from an estimate of conditional densities of the classes may aid the estimation of learner accuracy and overcome the limits of both resubstitution and holdout methods. Resampled datasets can be efficiently employed in leave-K-out or bootstrap estimation.

3.1. Assumptions

ROSE requires that the resampled variables are of numeric type, being impossible to fit a multivariate kernel on unordered categorical variables. This can include variables with limited numeric support (e.g. $\{0, 1\}$, or percentage values.).

Variables belonging to $\mathbb{N}$ could generate non-integer samples. This problem can be contained by rounding.

Variables belonging to $\mathbb{N}^+$ or $\mathbb{R}^+$ domains poses another problem, since samples drawn from the kernel function are not guaranteed to be positive. This particular problem can be contained by a log-transform of the original dataset parameters.

Relatively to our work described in Chapter 4, future development of ROSE will consider the option to extend the class by including type inference or by collecting `numpy.array` and `pandas.DataFrame` dtypes data to dynamically change the random sampling function.

3.2. Kernel methods

Since 90s estimation and learning methods using positive definite kernels have become popular, particularly in machine learning [35]. Real world analysis problems ofter require nonlinear methods to detect the kind of dependencies that allows successful prediction of properties of interests.

The operational use of ROSE requires a prior specification of the $\mathbf{H}_j$ matrices. In principle this leads to a criticality, since different choices of the smoothing matrices leads to larger or smaller $K_{\mathbf{H}_j}$, namely larger or smaller neighborhoogs of the observations from which the synthetic samples are generated. There is a large body of literature on methods of choice of the smoothing parameters [36] , [34]. The idea beyond these methods is to minimize an optimality criterion, as the asymptotic mean integrated squared error (AMISE).

$$AMISE(h;r) = \frac{R(K^{(r)})}{nh^{2r+1}} + \frac{1}{4}h^4\mu_2^2(K)r(f^{(r+2)})$$

Among all possible alternatives, Menardi and Torelli's proposal is to use Gaussian Kernels with diagonal smoothing matrices $\mathbf{H}_j = diag(h_1^{(j)}, \ldots, h_d^{(d)})$, and minimize AMISE.

This leads to:

$$h_q^{(j)} = \left(\frac{4}{(d+2)n}\right)^{1/(d+4)} \hat{\sigma}_q^{(j)} (q = 1, \ldots, d; j = class)$$

where $\hat{\sigma}_q^{(j)}$ is the sample estimate of the standard deviation of the qth dimension of the observation belonging to the class $\mathcal{Y}_j$. Despite the naivety of this approach, authors reports good results, since the only interest is producing a reasonable neighborhood where to sample the new data from, and happens to perform well even if $f(\mathbf{x}|y = \mathcal{Y}_j)$ is not *Normal* , just unimodal.

Choice of $\mathbf{H}_j$ smoothing matrix gives control on data generation:

In the following image we generated three blobs of examples from multivariate normal distributions. For the three classes, n will be respectively 33, 50 and 170.

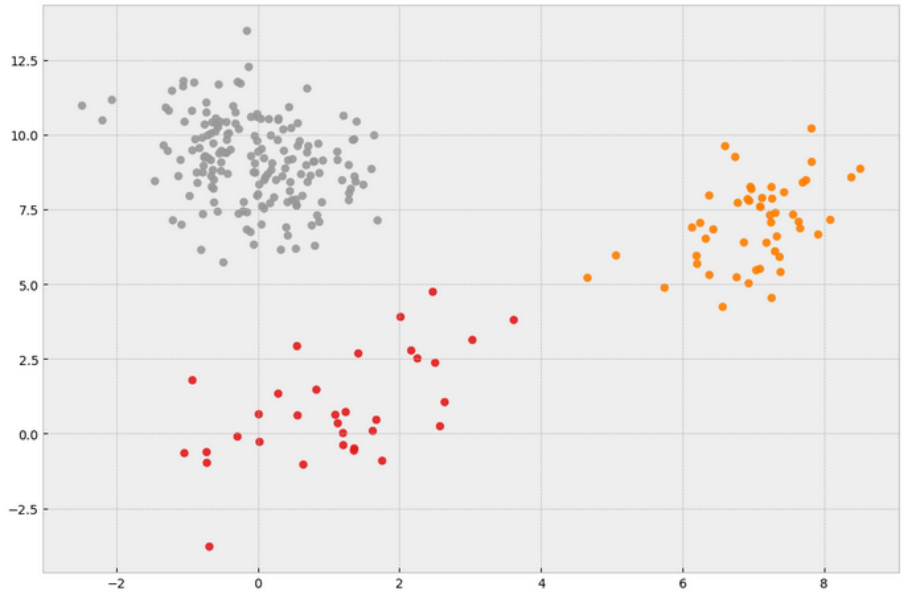


Fig: _ unbalanced classes.

In the next figure, we used ROSE to rebalance the datasets, and bring n to a total of 300 examples per class.

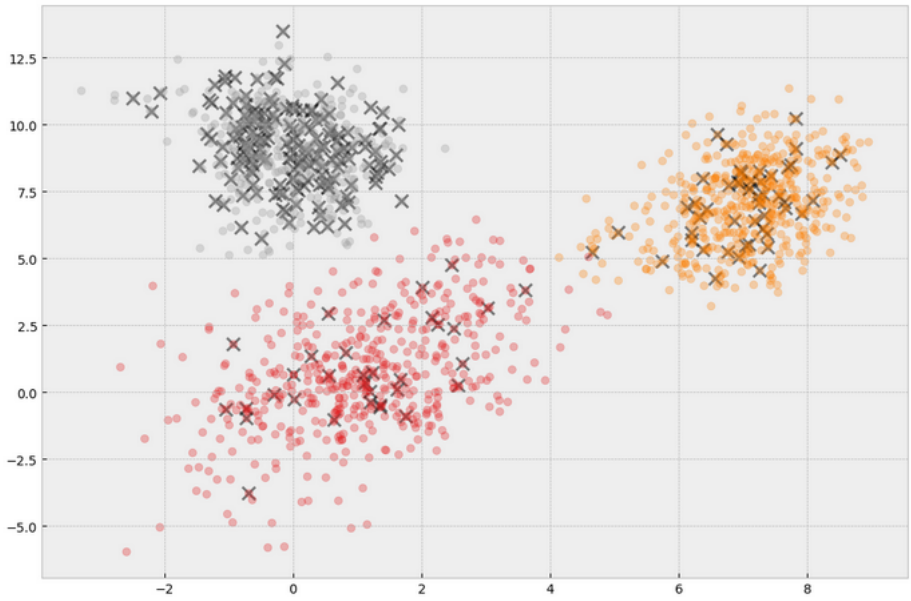


Fig _ : rebalanced datasets. Original data points are marked with grey"X".

Rose can use a **shrink factor** vector, to shrink kernels independently for each class. The following figure show how, decreasing the shrink factors, new data will be more and more closely clustered around original data points.

Fig _ Using shrink factors: $grey = 0.2, orange = 0.5, red = 1$.

4. Metrics

Evaluating performance is a critical part of building a machine learning model. In this chapter we will describe some of these tools, and how to choose the best one for our purposes in imbalanced data problems.

4.1. Confusion matrix

Confusion Matrices (henceforth CM) are tables that can be used to describe the performance of a classifier on a test set of data for which true values are known. They are detailed and simple to understand, but does not summarize well the performance.

n = 165	**Predicted: NO**	**Predicted: YES**
Actual: No	50	10
Actual: Yes	5	100

Table _ : An example of a confusion matrix for a binary classifier.

On the diagonal we find correctly predicted samples (true negatives, or TN, and true positives, or TP), leaving misclassified data on other cells (false positives, or FP, and false negatives, or FN). Confusion matrices can be extended to multiclass classifiers, their size becoming $j \times j$, for classes in $\mathcal{Y}_j$. Sums over rows and column will describe the total of actual vs predicted predictions. We have seen how secondary indexes can be computed form these values and their ratios.

When describing a model's performance, the most common classification metric is its *Accuracy* , defined as

$$Accuracy = (TP + TN)/(TP + TN + FP + FN)$$

This can be misleading, when the problem uses imbalanced data. Consider a sample with a 100:1 imbalance ratio. Classifying all values as the majority class will gives a $\sim 99\% Accuracy$ score. [37]. Different solutions has been proposed to solve this issue. For example, *BalancedAccuracy* score, defined as

$$Balanced\ Accuracy = \frac{\frac{TP}{P} + \frac{TN}{TN+FP}}{2}$$

can help. Another metric is *Predicted positive condition rate*, defined as

$$Predicted\ positive\ condition\ rate = \frac{TP + FP}{TP + FP + TN + FN}$$

which identifies the proportion of the total population correctly identified. Two other commonly used index is $F1$ score and Matthews correlation coefficient.

More informative visualizations of model performances can be given not by indices, but by plots, like Receiver Operating Characteristics and *Precision* vs *Recall* plots, that deserves a dedicated description in the following sub-chapters.

Additional metrics that can be extracted from CM are

- Cohen's Kappa, that is a measure of how well the classifier performed as compared to how well it would have performed simply by chance,
- Null Error Rate, that is how often you would have been wrong if you always predicted the majority class. This can be used as a useful baseline metric to compare a classifier against. Still, the Accuracy Paradox tells us that sometimes the best classifier will still have an high error rate than the null error rate.
- F_1 score. Since we will use it in our test suite later, we will dedicate next paragraph to its description.

4.2. F_1 Score

Called also F-score or F-measure, is an accuracy metric, calculated from the precision and recall of the test.

$$
\begin{aligned}
Precision &= \frac{TP}{TP + FP} \\
Recall &= \frac{TP}{TP + FN} \\
F_1 &= 2 * \frac{precision * recall}{precision + recall} \\
&= \frac{2 * TP}{2 * TP + FP + FN}
\end{aligned}
$$

It is a particular case of the more general F_β score, defined as

$$
\begin{aligned}
F_\beta &= (1 + \beta^2) \frac{precision * recall}{(\beta^2 * precision) + recall} \\
&= \frac{(1 + \beta^2) * TP}{(1 + \beta^2) * TP + \beta^2 * FN + FP}
\end{aligned}
$$

where recall is considered β times as important as precision. a $\beta > 1$ will increase recall importance, while $0 < \beta < 1$ will weight recall lower than precision [38]. It has recently been criticized as less informative and truthful than Mattees Correlation Coefficient (see below), especially for imbalanced classes.[39], and the adoption of new metrics is being suggested, like Informedness (Youden's J statistic)[40] and Markedness[41], in fields like biology and linguistics. When using geometric mean instead of harmonic mean of recall and precision it is known as Fowlkes-Mallows index [42].

4.3. Receiver Operating Characteristic (ROC) and AUC

A Receiver Operating Characteristic (ROC) curve is a plot that summarizes the performance of a binary classification model on the positive class. The x-axis indicates the False Positive Rate and the y-axis indicates the True Positive Rate.

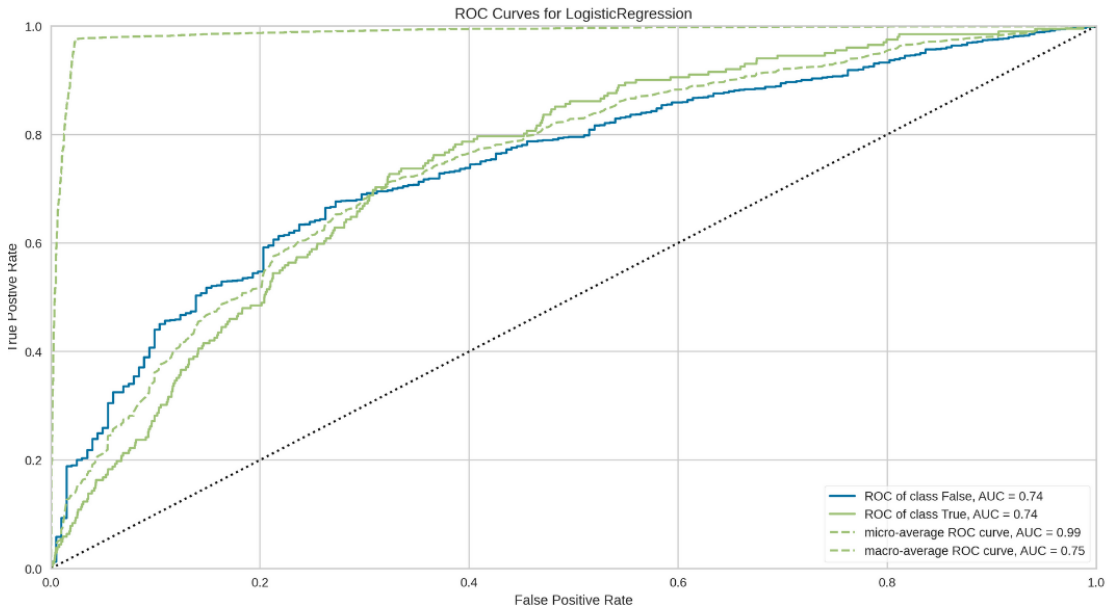


Fig _ : Example of ROC curve. AUC (Area under the curve) are shown in the bottom-right legend.

A ROC gives an intuitive visualization of a classifier performance: the dotted diagonal represent a classifier with no discriminative power, and the more the curve tends to the upper-left corner, the better the classifier is. The area under the curve (AUC) gives a commonly used single-valued index of performance. The threshold is applied to the cut-off point in probability between the positive and negative classes, which by default for any classifier would be set at 0.5, halfway between each outcome (0 and 1). A trade-off exists between the TP rate an FP rate, such that changing the threshold of classification will change the balance of predictions towards improving the TP rate at the expense of FP rate, or the reverse case.

By evaluating the true positive and false positives for different threshold values, the ROC curve is drawn. An interesting property is that the ROC is unbiased towards model that performs well on the minority class at the expense of the majority class, or vice versa, making it an interesting choice when dealing with imbalanced data.

4.4. Precision-recall plots

Precision-recall plots are a powerful visualization tool to evaluate binary classifiers, closely related to the Receiver Operating Characteristic described in the precedent sub-chapter. It shows the relation between these indexes, at the variation of a threshold

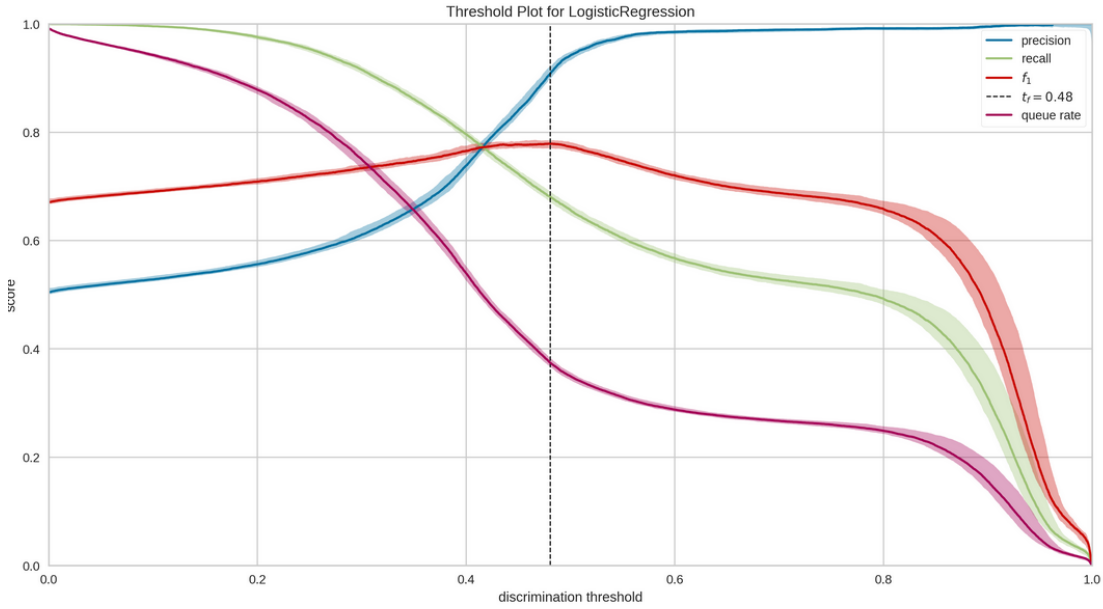


Fig _ Precision-recall plot of a logistic regression model. Bands are confidence interval around values. Queue rate can be seen as the "spam folder" or the inbox of the fraud investigation desk. This metric describes the percentage of instances that must be reviewed. If review has a high cost (e.g. fraud prevention) then this must be minimized with respect to business requirements; if it doesn't (e.g. spam filter), this could be optimized to ensure the inbox stays clean.

4.5. Matthews correlation coefficient (MCC)

Accuracy and F_1 score computed on confusion matrices have been (and still are) among the most popular adopted metrics in binary classification task [39]. However these measures can show overoptimistic inflated results, especially on imbalanced datasets. The Mattews correlation coefficient (henceforth, MCC) is instead a more reliable statistical rate which encompass all for confusion matrix categories (TP, FP, TN, FN), proportionally both to the size of positive and negative elements in the dataset. It derives from Guilford's ϕ coefficient[43].

$$\begin{aligned} MCC &= \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \\ &= \sqrt{\frac{\chi^2}{n}} \end{aligned}$$

5. Implementation of ROSE in the `imbalanced-learn` Python package

Practical steps taken to develop the package.

Computationally speaking, ROSE resampling is obtained with the following algorithm (pseudocode):

```
define make_samples (X,y,n,h_shrink):
     n = number of samples to be created
     p = number of features
     S = subset of samples randomly selected from X
     minAMISE = (4/((p+2)*n))**(1/(p+4))
     vars = variance/convariance matrix of all classes
     hOPT=h_shrink*minAMISE*vars
     randoms = multivariate_normal(size=(n,p))
     rose = randoms*hOPT + S
     return rose
```

5.1. `scikit-learn` context

Description of `scikit-learn` history and status.

5.2. Test driven development

Description of `pytest` and general overview about required test compliance.

5.3. GitHub and Kubernetes CI/CD

Description of `imblearn` repository continuous integration / continuous deployment process, and peer-review process. Linting.

5.4. Documentation

doctest, readthedocs, sphinx

6. Empirical analysis

Overview of the test framework

6.1. Materials & methods

Explanation of model estimation under different resampling strategies.

6.1.1. Datasets

Description of the 27 datasets used for benchmarking.

6.1.2. Models

Overview of the models used for benchmarking.

6.1.3. Chosen metrics

Description of the chosen metrics

6.2. Results

Tables with relative performance.

6.3. Chapter summary and discussion

Synthesis of the results, and discussion.

7. OSIRIS Dataset: a real-world application

7.1. Dataset description

7.2. Data source

7.3. HGF metrics

8. Appendix 1: how to use the package

Examples of package use for binary and multiclass balancing

9. Bibliography

1. Provost, Foster & Fawcett, Tom. (2001). Robust Classification for Imprecise Environments. Machine Learning. 42. 203-231. 10.1023/A:1007601015854. ↩

2. Yu, H., Hong, S., Yang, X., Ni, J., Dan, Y., Qin, B.: Recognition of Multiple imbalanced cancer types based on DNA microarray data using ensemble classifiers. BioMed Res. Int. 2013, 1–13 (2013) ↩

3. Zhao, X.M., Li, X., Chen, L., Aihara, K.: Protein classification with imbalanced data. Proteins Struct. Funct. Bioinf. 70(4), 1125-1132(2008) ↩

4. Cerf, L., Gay, D., Selmaoui-Folcher, N., Crémilleux, B., Boulicaut, J.F.: Parameter-free classification in multi-class imbalanced data sets. Data Knowl. Eng. 87, 109–129 (2013) ↩

5. Gao, X., Chen, Z., Tang, S., Zhang, Y., Li, J.: Adaptive weighted imbalance learning with application to abnormal activity recognition. Neurocomputing 173, 1927–1935 (2016) ↩

6. Razakarivony, S., Jurie, F.: Vehicle detection in aerial imagery: a small target detection benchmark. J. Vis. Commun. Image Represent. 34, 187–203 (2016) ↩

7. Gao, Z., Zhang, L., Chen, M.-yu., Hauptmann, A.G., Zhang, H., Cai, A.N.: Enhanced and hierarchical structure algorithm for data imbalance problem in semantic extraction under massive video dataset. Multimed. Tools Appl. 68(3), 641–657 (2014) ↩

8. Wang, S., Chen, H., Yao, X.: Negative correlation learning for classification ensembles. In: 2010 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2010) ↩

9. Ganganwar, Vaishali. (2012). An overview of classification algorithms for imbalanced datasets. International Journal of Emerging Technology and Advanced Engineering. 2. 42-47. ↩

10. King, Gary, and Langche Zeng. "Logistic regression in rare events data." *Political analysis* 9.2 (2001): 137-163. ↩

11. Menardi, G., Torelli, N. (2009) Some issues in building and assessing classification rules with extremely skewed datasets. Proceedings of the 7th Meeting of the Classification and data Analysis Group of the Italian Statistical Society (Invited Papers), Catania, ISBN 978-88-6129-406-6, Cleup (Padova). ↩

12. Menardi, G., and N. Torelli. "Training and assessing classification rules with unbalanced data) Working Paper Series." *N* 2 (2010): 2010. ↩

13. Chawla, Nitesh V., et al. "SMOTEBoost: Improving prediction of the minority class in boosting." *European conference on principles of data mining and knowledge discovery*. Springer, Berlin, Heidelberg, 2003. ↩

14. Gue, Kevin R. "A dynamic distribution model for combat logistics." *Computers & Operations Research* 30.3 (2003): 367-381. ↩

15. Ndour, Cheikh, Aliou Diop, and Simplice Dossou-Gbété. "Classification approach based on association rules mining for unbalanced data." *arXiv preprint arXiv:1202.5514* (2012). ↩

16. Liu, Xu-Ying, and Zhi-Hua Zhou. "The influence of class imbalance on cost-sensitive learning: An empirical study." *Sixth International Conference on Data Mining (ICDM'06)*. IEEE, 2006. ↩

17. Zhou, Zhi-Hua, and Xu-Ying Liu. "On multi-class cost-sensitive learning." *Computational Intelligence* 26.3 (2010): 232-257. ↩

18. He, Haibo, and Edwardo A. Garcia. "Learning from imbalanced data." *IEEE Transactions on knowledge and data engineering* 21.9 (2009): 1263-1284. ↩

19. Weiss, Roger D., et al. "Long-term outcomes from the national drug abuse treatment clinical trials network prescription opioid addiction treatment study." *Drug and alcohol dependence* 150 (2015): 112-119. ↩

20. P. Hart, "The condensed nearest neighbor rule," In Information Theory, IEEE Transactions on, vol. 14(3), pp. 515-516, 1968. ↩

21. M. Kubat, S. Matwin, "Addressing the curse of imbalanced training sets: one-sided selection," In ICML, vol. 97, pp. 179-186, 1997. ↩

22. Wilson, Dennis L. "Asymptotic properties of nearest neighbor rules using edited data." *IEEE Transactions on Systems, Man, and Cybernetics* 3 (1972): 408-421. ↩

23. I. Mani, I. Zhang. "kNN approach to unbalanced data distributions: a case study involving information extraction," In Proceedings of workshop on learning from imbalanced datasets, 2003. ↩

24. D. Smith, Michael R., Tony Martinez, and Christophe Giraud-Carrier. "An instance level analysis of data complexity." Machine learning 95.2 (2014): 225-256. ↩

25. N. V. Chawla, K. W. Bowyer, L. O.Hall, W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," Journal of artificial intelligence research, 321-357, 2002. ↩ ↩

26. Han, Hui, Wen-Yuan Wang, and Bing-Huan Mao. "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning." *International conference on intelligent computing*. Springer, Berlin, Heidelberg, 2005. ↩

27. Felix Last, Georgios Douzas, Fernando Bacao, "Oversampling for Imbalanced Learning Based on K-Means and SMOTE" https://arxiv.org/abs/1711.00837 ↩

28. H. M. Nguyen, E. W. Cooper, K. Kamei, "Borderline over-sampling for imbalanced data classification," International Journal of Knowledge Engineering and Soft Data Paradigms, 3(1), pp.4-21, 2009. ↩

29. He, Haibo, Yang Bai, Edwardo A. Garcia, and Shutao Li. "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," In IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 1322-1328, 2008. ↩

30. G. Batista, R. C. Prati, M. C. Monard. "A study of the behavior of several methods for balancing machine learning training data," ACM Sigkdd Explorations Newsletter 6 (1), 20-29, 2004. ↩

31. G. Batista, B. Bazzan, M. Monard, "Balancing Training Data for Automated Annotation of Keywords: a Case Study," In WOB, 10-18, 2003. ↩

32. Wilson, Dennis L. "Asymptotic properties of nearest neighbor rules using edited data." *IEEE Transactions on Systems, Man, and Cybernetics* 3 (1972): 408-421. ↩

33. Tibshirani, Robert J.; Efron, Bradley. An introduction to the bootstrap. *Monographs on statistics and applied probability*, 1993, 57: 1-436. ↩

34. Bowman, Adrian W., and Adelchi Azzalini. *Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations*. Vol. 18. OUP Oxford, 1997 ↩ ↩

35. Hofmann, Thomas, Bernhard Schölkopf, and Alexander J. Smola. "Kernel methods in machine learning." *The annals of statistics* (2008): 1171-1220. ↩

36. Silverman, Bernard W. *Density estimation for statistics and data analysis*. Vol. 26. CRC press, 1986. ↩

37. Mower, Jeffrey P. "PREP-Mt: predictive RNA editor for plant mitochondrial genes." *BMC bioinformatics* 6.1 (2005): 96. ↩

38. Van Rijsbergen, Cornelis J. "A new theoretical framework for information retrieval." *Acm Sigir Forum*. Vol. 21. No. 1-2. New York, NY, USA: ACM, 1986. ↩

39. Chicco, Davide, and Giuseppe Jurman. "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation." *BMC genomics* 21.1 (2020): 6. ↩ ↩

40. Youden, William J. "Index for rating diagnostic tests." *Cancer* 3.1 (1950): 32-35. ↩

41. Henning, Andersen. "Markedness: The First 150 Years." *Markedness in Synchrony and Diachrony, Olga M. Tomic (ed.), Mouton de Gruyter, Berlin–Germany* (1989): 11-46. ↩

42. Fowlkes, Edward B., and Colin L. Mallows. "A method for comparing two hierarchical clusterings." *Journal of the American statistical association* 78.383 (1983): 553-569. ↩

43. Guilford, Joy Paul. "Psychometric methods." (1954). ↩