On the "Using Random Forest to Learn Imbalanced Data"

by

Ramin Ziaei

Discussion 3

Under the Supervision of Dr. Cheng

Department of Mathematics

University of Nebraska - Omaha

Omaha, Nebraska

November 5^{th}, 2020

Contents

List of Figures

List of Tables

1 Introduction

In many practical classification problems, class imbalacny is an issue that needs to be addressed. An imbalanced classification problem is where the distribution of the known response classes are highly skewed. For example, a response might have 95% of one class and only 5% of the other class (if the problem is a binary classification problem). Here, even without a machine learning model and by simply predicting all the outcome as the first class, a high prediction accuracy will be obtained since almost all of them belong to the first class. However, the importance of this kind of problem is to predict the other class accurately since that is the one that happens rarely, and that is what makes correctly predicting this class important. Sometimes, not taking care of class imbalancy before making a model leads to a model that is not capable of predicting the minority class precisely. There are many different ways to tackle this problem some of which are down-sampling (that will remove rows of the set to make the occurrence of levels in a specific factor level equal), up-sampling (that will replicate rows of a data set to make the occurrence of levels in a specific factor level equal), SMOTE (that generates new examples of the minority class using nearest neighbors of these cases), Adaptive Synthetic Sampling Approach (that generates synthetic positive instances using ADASYN algorithm), to name but a few. In this paper, random forest is used to evaluate the performance of different methods used to address class imbalancy.

2 Datasets

The data is the one I will be using for my final project. It contains information of 2 different hotels, one being a resort hotel (H1) with 40,060 observations and the other one being a city hotel (H2) with 79,330 [2]. Both data sets share the same structure with 30 predictors, and the goal is to predict if a specific observation would cancel their booking or not. Data cleaning part has been already explained in my first draft, so it will not be covered here again, and I will just use the result of it. The cleaned data has 15 factor and 15 numeric columns. For this work, three different data sets with different class imbalancy are generated based on the original data set. 1%, 10% and 30% for the minority class were chosen to cover a good range of class imbalancy. The total number of observations used for this work was set to 50,000. Table 1 shows the number of observations in each case. 80% of the each data set was used for training, and the remaining 20% was used as the test set. Also, the response was stratified so the same ratio of both classes will be present in train and test data sets.

Table 1: Class imbalancy observations

	No. of Observations	
Imbalancy	Not canceled	Canceled
1%	49500	500
10%	45000	5000
30%	35000	15000

3 Balanced Random Forest

As we know, Random Forest (RF) is a machine learning technique used for both regression and classification problems. In this approach, many trees are built each of which is from a randomly drawn bootstrap sample from the training set, and the performance (the majority vote for classification problems) is calculated over the out-of-bag observations. In learning extremely imbalanced data, there is a significant probability that a bootstrap sample contains few or even none of the minority class, resulting in a tree with poor performance for predicting the minority class [1]. In the RF algorithm, there is an argument called $cutoff$ which is a vector of length equal to the number of classes. The 'winning' class for an observation is the one with the maximum ratio of proportion of votes to cutoff. The default is 1/k where k is the number of classes (i.e., majority vote wins). In a Balanced Random Forest (BRF), this cutoff point is changed in order to make the observations more balanced. In imbalanced data sets, not only is accuracy important, there are two more criteria to consider namely *Precision* and *Recall*. They are defined as follows:

$$Precision = \frac{\#\ of\ correctly\ predicted\ as\ positive}{\#\ of\ examples\ predicted\ as\ positive}$$

$$Recall = \frac{\#\ of\ correctly\ predicted\ as\ positive}{\#\ of\ positive\ examples\ in\ test\ set}$$

Figure 1 displays accuracy, precision and recall for the three data sets considered.

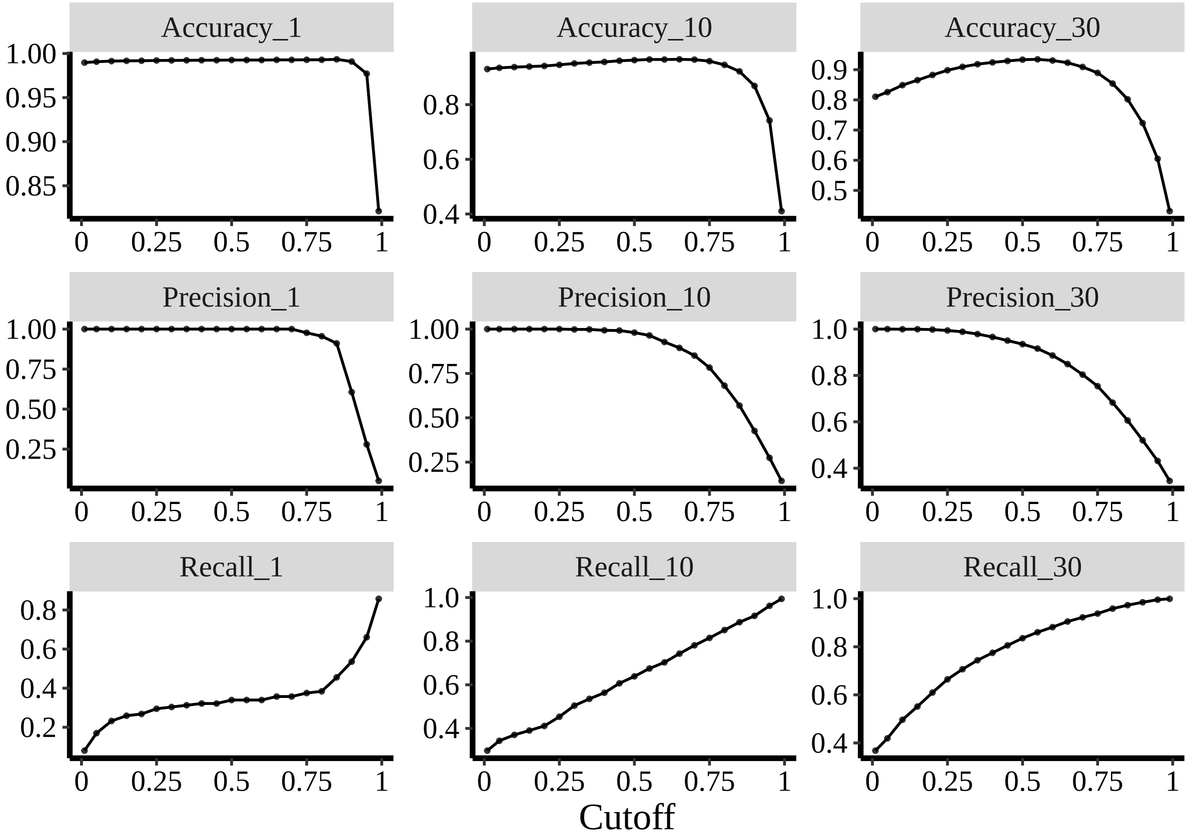

Figure 1: BRF accuracy, precision and recall for 1, 10 and 30 percent class imbalancy

Table 2 shows the cutoff points that result in the best performance in terms of accuracy.

Table 2: Cutoff points for best accuracy performance

Imbalancy	Cutoff
1%	0.85
10%	0.65
30%	0.55

It should also be noted that Regular RF (RRF) is when the cutoff point is 0.5 which is one of the cases considered in this BRF section. As table 2 shows, for all three imbalancy cases, BRF has a higher accuracy than a regular RF. While the highest accuracy occurs somewhere between 0.5 and 1 for these three cases, precision and recall change monotonically with the cutoff point. As the cutoff point increases, precision decreases while recall increases.

4 Weighted Random Forest

Another approach to make random forest more suitable for learning from extremely imbalanced data follows the idea of cost sensitive learning. Since the RF classifier tends to be biased towards the majority class, a heavier penalty shall be placed on mis-classifying the minority class. A weight is assigned to each class, with the minority class given larger weight (i.e., higher mis-classification cost). The class weights are incorporated into the RF algorithm in two places. In the tree induction procedure, class weights are used to weight the "Gini" criterion for finding splits. In the terminal nodes of each tree, class weights are again taken into consideration. The class prediction of each terminal node is determined by "weighted majority vote"; i.e., the weighted vote of a class is the weight for that class times the number of cases for that class at the terminal node. The final class prediction for RF is then determined by aggregating the weighted vote from each individual tree, where the weights are average weights in the terminal nodes.

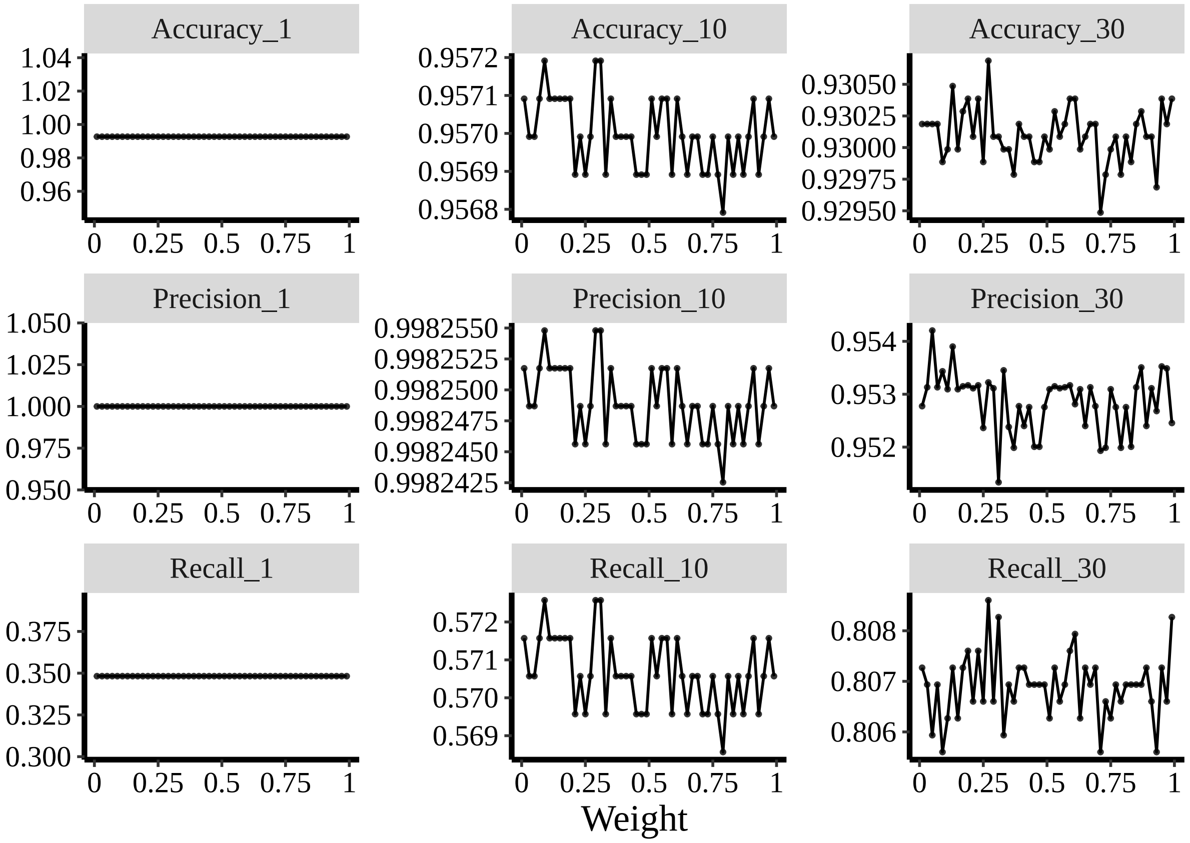

Figure 2: WRF accuracy, precision and recall for 1, 10 and 30 percent class imbalancy

For weighted RF, if the minority class imbalance is 1%, changing the class weight will not change any of the accuracy, precision or recall. For 10% and 30% cases, unlike BRF, changes are not monotonic. In this case, weight is the ratio of the majority class weight to the minority class weight ranges from 1:100 to 99:100. Table 3 shows the performance of up-sampling and down-sampling on the three data sets considered in this discussion.

Table 3: WRF best weights performance

Imbalancy	Measure	Best.weight	Value
1%	accuracy	All	0.9926993
	precision	All	1.0000000
	recall	All	0.3482143
10%	accuracy	0.09, 0.29, 0.31	0.9571914
	precision	0.09, 0.29, 0.31	0.9982548
	recall	0.09, 0.29, 0.31	0.5725726
30%	accuracy	0.27	0.9306861
	precision	0.05	0.9542045
	recall	0.27	0.8086029

5 Up- and Down-sampling

Finally, Up-sampling and Down-sapling were done to see how they would affect the performance measures. Down-sampling will remove rows of the set to make the occurrence of levels in a specific factor level equal while up-sampling will replicate rows of a data set to make the occurrence of levels in a specific factor level equal. Table 4 displays the results of these two methods.

Table 4: Up- and down-sampling performance on 1, 10 and 30 percent class imbalancy

Imbalancy	Measure	Up.sample	Down.sample
1%	accuracy	0.9923992	0.8644864
	precision	0.9500000	0.0613973
	recall	0.3392857	0.7767857
10%	accuracy	0.9650930	0.9174835
	precision	0.9188144	0.5584677
	recall	0.7137137	0.8318318
30%	accuracy	0.9337868	0.9238848
	precision	0.9294377	0.8635478
	recall	0.8432811	0.8862954

6 Comparison

Table 5 is the summary all four methods and their performance for different measures. According to this table, for all cases (except for the precision of 1%), BRF outperforms other methods. For the 1% precision case, it is a tie among RRF, BRF and WRF.

Table 5: Comparison among all models with their best performance for each measure

Imbalancy	Measure	Method	Value
1%	accuracy	RRF	0.9925993
		BRF	0.9933993
		WRF	0.9926993
		Up-sampling	0.9923992
		Down-sampling	0.8644864
	precision	RRF	1.0000000
		BRF	1.0000000
		WRF	1.0000000
		Up-sampling	0.9500000
		Down-sampling	0.0613973
	recall	RRF	0.3392857
		BRF	0.8571429
		WRF	0.3482143
		Up-sampling	0.3392857
		Down-sampling	0.7767857
10%	accuracy	RRF	0.9625925
		BRF	0.9654931
		WRF	0.9571914
		Up-sampling	0.9650930
		Down-sampling	0.9174835
	precision	RRF	0.9800307
		BRF	1.0000000
		WRF	0.9982548
		Up-sampling	0.9188144
		Down-sampling	0.5584677
	recall	RRF	0.6386386
		BRF	0.9939940
		WRF	0.5725726
		Up-sampling	0.7137137
		Down-sampling	0.8318318
30%	accuracy	RRF	0.9332867
		BRF	0.9343869
		WRF	0.9306861
		Up-sampling	0.9337868
		Down-sampling	0.9238848
	precision	RRF	0.9350746
		BRF	1.0000000
		WRF	0.9542045
		Up-sampling	0.9294377
		Down-sampling	0.8635478
	recall	RRF	0.8356119
		BRF	0.9993331
		WRF	0.8086029
		Up-sampling	0.8432811
		Down-sampling	0.8862954

7 Summary

The paper proposes a few methods for dealing with imbalanced data classification problems using the random forest (RF) algorithm. In BRF, in each tree, a bootstrap sample from the minority class is drawn, then the same number is drawn from the majority class. The difference between this approach and down-sampling is that here, this down-sampling occurs in each tree, but in down-sampling, the original data is shrunk to have comparable size of classes. The disadvantage of down-sampling is that some information will be lost. This is almost the same for BRF, but since this happens in each tree, and every iteration will pick different minority and majority class observations, information lost is less problematic compared to down-sampling. WRF focuses on putting more penalty on misclassifying the minority class. Usually when dealing with imbalanced data sets, the minority class is of more interest than the majority class, and using heavier penalties for misclassifying them is a way to make sure that they will be predicted will reasonable accuracy. All in all, it seems that for the data set considered in this discussion, BRF performs best. When choosing a model and an approch to address class imbalancy, it is crutially important to know what aspect of the response is more important for that specific work. Most of the time, people care most about the accuracy, but in significantly imbalanced data sets, accuracy is not as interesting as precision/recall can be. So, based on the type of data we deal with and the purpose of that work a suitable measure needs to be chosen.

References

1. Chen, C., Liaw, A., & Breiman, L. (2004). Using random forest to learn imbalanced data. University of California, Berkeley, 110(1-12), 24.

2. Antonio, N., de Almeida, A., & Nunes, L. (2019). Hotel booking demand datasets. Data in brief, 22, 41-49.