Paper 11140-2016

Predicting Rare Events Using Specialized Sampling Techniques in SAS®

Rhupesh Damodaran Ganesh Kumar and Kiren Raj Mohan Jagan Mohan, Oklahoma State University

ABSTRACT

In recent years, many companies are trying to understand the rare events that are very critical in the current business environment. But a data set with rare events is always unbalanced and the models developed using this data set cannot predict the rare events precisely. Therefore, to overcome this issue, a data set needs to be sampled using specialized sampling techniques like over-sampling, under-sampling, or the synthetic minority over-sampling technique (SMOTE). The over-sampling technique deals with randomly duplicating minority class observations, but this technique might bias the results. The under-sampling technique deals with randomly deleting majority class observations, but this technique might lose information. SMOTE sampling deals with creating new synthetic minority observations instead of duplicating minority class observations or deleting the majority class observations. Therefore, this technique can overcome the problems, like biased results and lost information, found in other sampling techniques.

In our research, we used an unbalanced data set containing results from a thyroid test with 3,163 observations, out of which only 4.7 percent of the observations had positive test results. Using SAS® procedures such as PROC SURVERYSELECT and PROC MODECLUS, we created over-sampled, under-sampled, and the SMOTE sampled data set in SAS® Enterprise Guide®. Then we built decision tree, gradient boosting, and rule induction models using four different data sets (non-sampled, majority under-sampled, minority over-sampled with majority under-sampled, and minority SMOTE sampled with majority under-sampled) in SAS® Enterprise Miner™. Finally, based on the receiver operating characteristic (ROC) index, Kolmogorov-Smirnov statistics, and the misclassification rate, we found that the models built using minority SMOTE sampled with the majority under-sampled data yields better output for this data set

INTRODUCTION

In data mining projects, one of the most common problems is unbalanced data. A dataset is unbalanced when the class distribution is not uniform among the classes. So, in an unbalanced dataset with a binary target variable, either the 0's or the 1's in the target variables occurs very less number of times (usually less than 5% of the data) compared to the other.

Figure 1 is an example of unbalanced data, where the red dots indicate minority class while the blue dots indicate majority class. The models built using these dataset will generally be biased towards predicting the majority class. For example, assume that a credit card company is trying to find fraud customers and in their database they have 100 customers. Out of these 100 customers, say only 2 customers are frauds. So, this dataset is unbalanced with 2% of minority observation and 98% of majority observations. When a model is built using this dataset, it can be 98% accurate by just predicting all the customers as non-frauds. So, the models built using unbalanced dataset is biased towards the majority class (non-frauds) while the most interesting part is the minority observations (frauds).

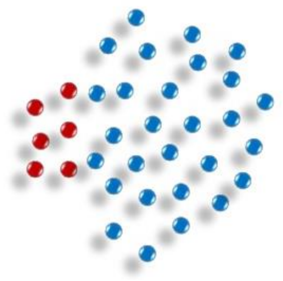


Figure 1, Unbalanced Dataset

So, to overcome this challenge, the dataset needs to be sampled using specialized sampling techniques such as over-sampling, under-sampling and synthetic minority over-sampling technique (SMOTE).

OVER-SAMPLING TECHNIQUE

This technique increases the apparent sampling frequency of the minority samples by randomly repeating each observation. Figure 2 is an example of over-sampling and in the graph, the green dots indicate majority class, the purple dots indicate minority class and the brown dots indicate randomly repeated minority class observations.

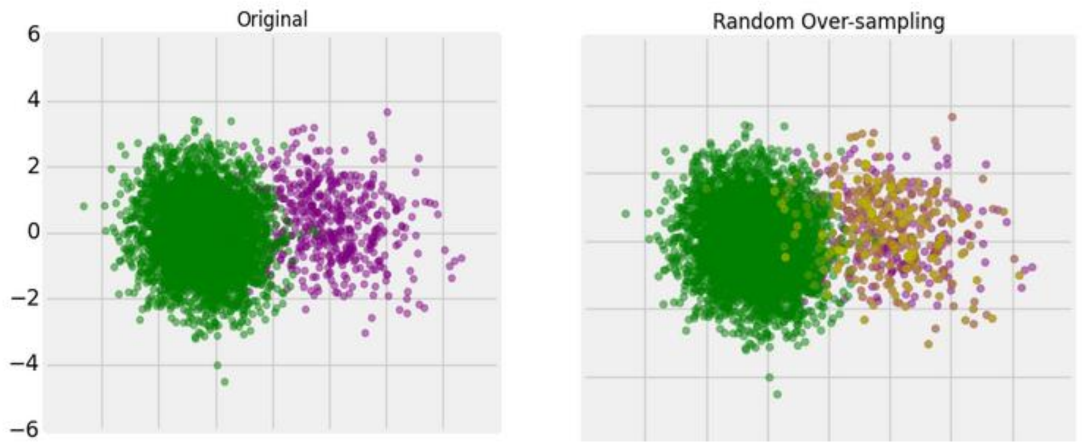


Figure 1, Over-Sampling Technique

UNDER-SAMPLING TECHNIQUE

This technique reduces the apparent sampling frequency of the majority samples by randomly removing observations. Figure 3 is an example of under-sampling and in the graph, the blue dots indicate majority class, and the red dots indicate minority class.

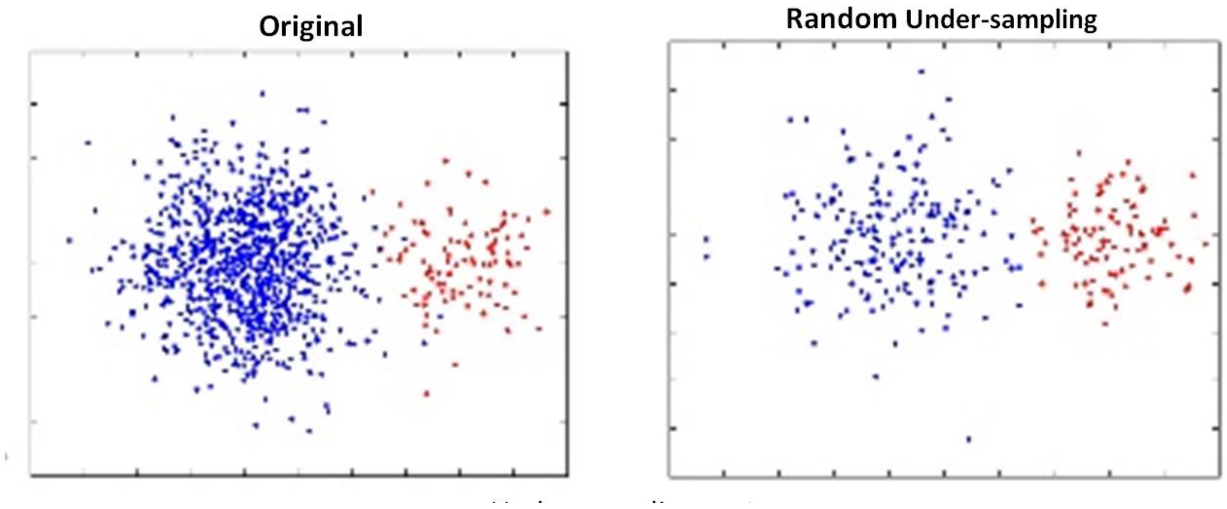


Figure 2, Under-Sampling Technique

SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE

This technique increases the apparent sampling frequency of the minority samples by creating new synthetic observations using a specific algorithm. Figure 4 is an example of SMOTE sampling and in the graph, blue squares indicate majority class, red circles indicate minority class and the yellow squares indicate new minority samples. The red circle shows the k-nearest neighbor algorithm applied to minority samples to generate new synthetic samples.

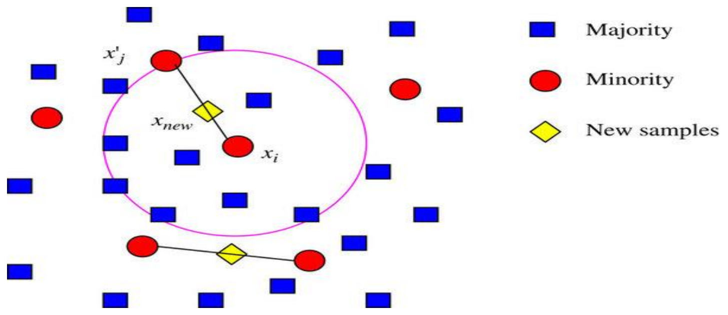


Figure 3, SMOTE Sampling Technique

The following are the steps used to create SMOTE sample in SAS® Enterprise Guide™:

1. Use the PROC SURVEYSELECT procedure, to create a subset of minority observation, equal to the frequency of new synthetic samples to be generated.

In the case of continuous variable, for each variable

2. Use the PROC MODECLUS procedure, to identify 5 nearest neighbors for each observation.
3. For each observation, generate a random number between 1 and 5 and then use this random number to select Kth nearest neighbor of this observation. The value of this observation is the new synthetic value.

In the case of class variable,

2. For each observation, generate a random number between 1 and 5.

For each variable,

3. Use the PROC MODECLUS procedure to identify 5 nearest neighbors for each observation.

For each observation,

4. Use the random number generated in step 2 to select Kth nearest neighbor of this observation and assign it to Y.
5. Assign the original value of this observation to X.
6. Compute the value for D, by subtracting X from Y and then assign a random number between 0 and 1 to G.
7. Compute new synthetic value using the formula N = X + G * D.

The following is the pseudocode for class variable

```
Algorithm SMOTE (T, N, k)
Input: Number of minority class samples T; Amount of SMOTE N%; Number of nearest neighbors k
Output: (N/100)* T synthetic minority class samples
1. (∗If N is less than 100%, randomize the minority class samples as only a random percent of them will
   be SMOTED.∗)
2. if N <100
3. then Randomize the T minority class samples
4. T = (N/100) *T
5. N = 100
```

```
6.  **endif**
7.  N = (int) (N/100) (*the amount of SMOTE is assumed to be in integral multiples of 100.*)
8.  k = Number of nearest neighbors
9.  numattrs = Number of attributes
10. Sample[ ][ ]: array for original minority class samples
11. newindex: keeps a count of number of synthetic samples generated, initialized to 0
12. Synthetic[ ][ ]: array for synthetic samples
(*Compute k nearest neighbors for each minority class sample only.*)
13. **for** i ←1 **to** T
14. Compute k nearest neighbors for i, and save the indices in the nnarray
15. Populate(N, i, nnarray)
16. **endfor**
Populate (N, i, nnarray) (*Function to generate the synthetic samples.*)
17. **while** N ≠0
18. Choose a random number between 1 and k, call it nn. This step chooses one of the k nearest
    neighbors of i.
19. **for** attr ←1 to numattrs
20. Compute: dif = Sample[nnarray[nn]][attr] −Sample[i][attr]
21. Compute: gap = random number between 0 and 1
22. Synthetic[newindex][attr] = Sample[i][attr] + gap ∗ dif
23. **endfor**
24. newindex++
25. N = N −1
26. **endwhile**
27. **Return** (*End of Populate.*)
End of Pseudo-Code.
```

In the case of continuous variable, from step 18 the following steps are replaced:

```
18. **for** attr ←1 **to** numattrs
19. Choose a random number between 1 and k, call it nn. This step chooses one of the k nearest
    neighbors of i.
20. Synthetic[newindex][attr] = Sample[nnarray[nn]][attr]
21. **endfor**
22. newindex++
23. N = N −1
24. **endwhile**
25. **Return** (*End of Populate.*)
End of Pseudo-Code.
```

DATASET DESCRIPTION

We used a real thyroid test data, which has 25 variables (target 1, continuous 6, and class 18) with 3,163 observations. The target variable used in this data is, whether the patient thyroid test is positive (1) or negative (0). Figure 5 shows the unbalanced thyroid data which has only 151 observations with thyroid test results as positive.

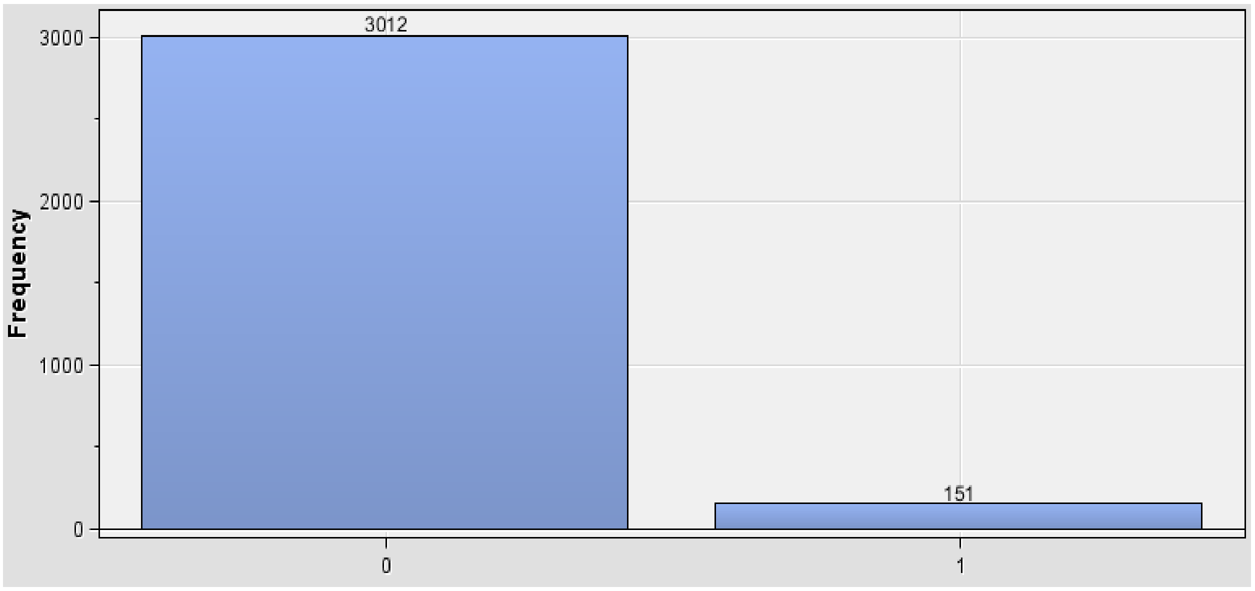


Figure 4, One-Way Frequency of Target

MODEL

We built decision tree, gradient boosting and rule induction models with four different dataset (non-sampled, majority under-sampled, minority over-sampled with majority under-sampled, minority SMOTE sampled with majority under-sampled) in SAS® Enterprise Miner™.

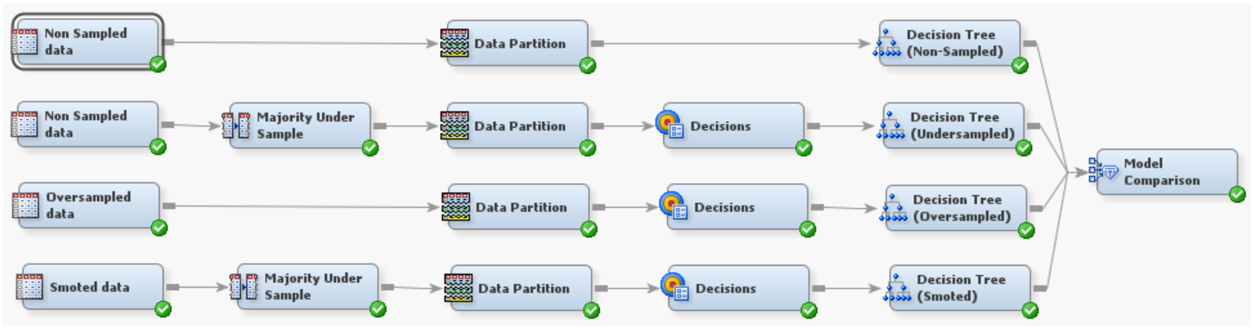


Figure 5, Process Flow

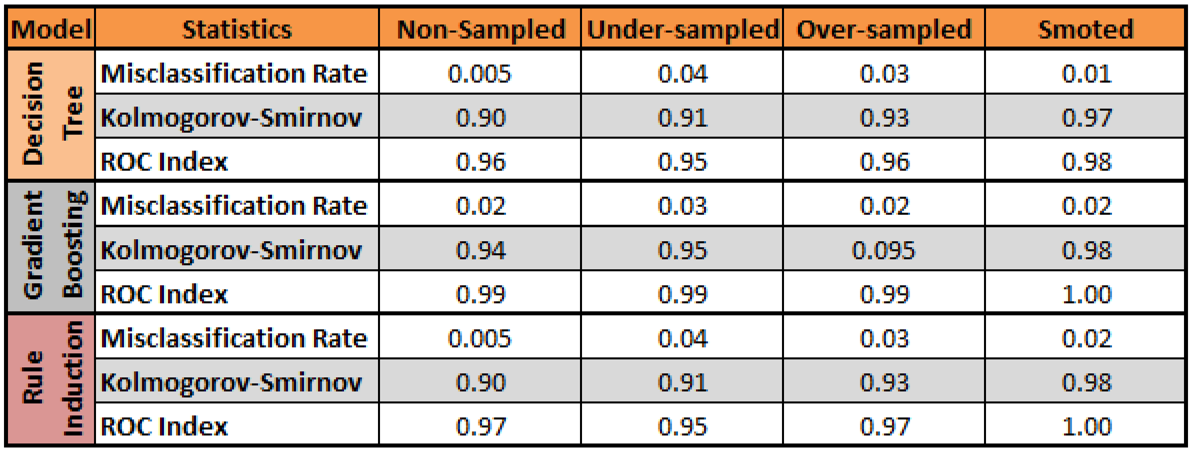
Figure 6 shows the process flow of models built in SAS® Enterprise Miner™. We used data partition node to split 70% of data for training and 30% of data for validation in the process flow in order to ensure honest assessment. For all sampled data models, we used decision node to set the prior probability as 0.048 for minority class observation (Target = 1) and 0.952 for majority class observation (Target = 0).

We built decision tree, gradient boosting and rule induction models using four different datasets. In the case of non-sampled scenario, we used all the 3,163 observations directly to build the models. In the case of majority under-sampled scenario, we used sample node in SAS® Enterprise Miner™ to under-sample the majority class observations (Target = 0) to match the number of majority class observations with number of minority class observations and finally built the models with 302 observations.

In the case of minority over-sampled with majority under-sampled scenario, we used PROC SURVEYSELECT procedure in SAS® Enterprise Guide™ to over-sample minority class observations (Target – 1) and also to under-sample majority class observations (Target = 0). Finally, we used a balanced data with 600 observations to build the models. In the case of minority SMOTE sampled with majority under-sampled scenario, we used PROC MODECLUS procedure in SAS® Enterprise Guide™ to generate new synthetic samples of minority class observations (Target = 1) and used sample node in SAS® Enterprise Miner™ to under-sample the majority class observation (Target = 0). Finally, we used a balanced data with 576 observations to build the models.

MODEL COMPARISON

We used the model comparison node in SAS® Enterprise Miner™ to compare twelve models, based on misclassification rate, Kolmogorov-Smirnov statistics and ROC Index. We can observe from the model comparison node results (figure 7) that all the models built using non-sampled data is biased toward majority samples with the least misclassification rate.



Model	Statistics	Non-Sampled	Under-sampled	Over-sampled	Smoted
Decision Tree	Misclassification Rate	0.005	0.04	0.03	0.01
	Kolmogorov-Smirnov	0.90	0.91	0.93	0.97
	ROC Index	0.96	0.95	0.96	0.98
Gradient Boosting	Misclassification Rate	0.02	0.03	0.02	0.02
	Kolmogorov-Smirnov	0.94	0.95	0.095	0.98
	ROC Index	0.99	0.99	0.99	1.00
Rule Induction	Misclassification Rate	0.005	0.04	0.03	0.02
	Kolmogorov-Smirnov	0.90	0.91	0.93	0.98
	ROC Index	0.97	0.95	0.97	1.00

Figure 6, Model Comparison Results

So, when comparing the models based on Kolmogorov-Smirnov statistics and ROC Index, all the models built using minority SMOTE sampled with majority under-sampled data is better. We can also observe the same in ROC curve shown in figure 8.

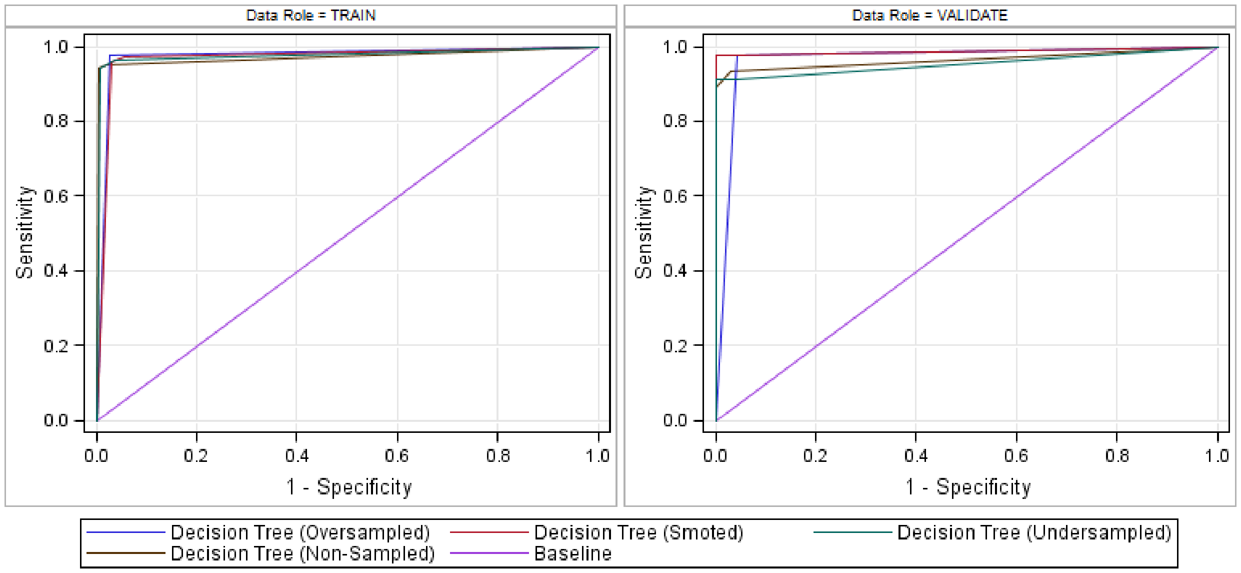


Figure 7, ROC Index

CONCLUSION

In this paper we discussed the specialized sampling techniques that can be used to predict the rare events. The results show that the models built using unbalanced data is always biased toward majority class. But when comparing models based Kolmogorov-Smirnov statistics and ROC index, the models built using the minority SMOTE sampled with majority under-sampled data yields better results. So, in the case of rare events, SMOTE sampling can improve the model performance and also overcome the problems, like biased results and lost information, found in other sampling techniques. The conclusion summarizes your paper and ties together any loose ends. You can use the conclusion to make any final points such as recommendations, predictions, or judgments.

REFERENCES

[1]Chawla Nitesh, Bowyer Kevin, Hall Lawrence and Kegelmeyer Philip. 2002. "SMOTE: Synthetic Minority Over-sampling Technique." *Journal of Artificial Intelligence and Research, 16:321-357.*
[2]Pears Russel, Finlay Jacqui, Connor Andy, Colab. "Synthetic Minority Over-sampling Technique (SMOTE) for Predicting Software Build Outcomes."
[3]Guzman Lina. 2015. "Data sampling improvement by developing SMOTE technique in SAS." *Proceedings of the SAS Global Forum 2015 Conference, 3483-2015.*
[4]Wang Ruizhe, Lee Novik, Wei Yun. 2015. "A Case Study: Improve Classification of Rare Events with SAS Enterprise Miner." *Proceedings of the SAS Global Forum 2015 Conference, 3282-2015.*

ACKNOWLEDGMENTS

We thank Dr. Goutam Chakraborty, Professor, Department of Marketing and founder of SAS and OSU Data Mining Certificate Program – Oklahoma State University for his support throughout this research.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Rhupesh Damodaran Ganesh Kumar
Phone: 405-612-5011
E-mail: rhupesh@okstate.edu

Kiren Raj Mohan Jagan Mohan
Phone: 405-385-3129
E-mail: kiren@ostatemail.okstate.edu

Dr. Goutam Chakraborty
Oklahoma State University
E-mail: goutam.chakraborty@okstate.edu