Paper 3282-2015

A Case Study: Improve Classification of Rare Events with SAS® Enterprise Miner™

Ruizhe Wang, GuideWell Connect; Novik Lee, GuideWell Connect;
Yun Wei, GuideWell Connect

ABSTRACT

Imbalanced data are frequently seen in fraud detection, direct marketing, disease prediction and many other areas. Rare events are sometimes of our primary interest and to classify them correctly are the challenges many predictive modelers face today.

In this paper, we use SAS® Enterprise Miner™ on a marketing data set to demonstrate and compare several approaches commonly used to handle imbalanced data problems in classification models. The approaches are based on cost-sensitive measures and sampling measures. A rather novel technique called SMOTE (Synthetic Minority Over-sampling TEchnique), which has achieved the best result in our comparison, is discussed.

INTRODUCTION

About 0.1% of all card transactions are fraudulent, so theoretically we can create a classification model with the accuracy rate as high as 99.9% by defaulting all the predicted outcomes to be non-fraudulent. However, there is no point of doing so, when the price for misclassifying the rare events is too high to be ignored. In this study, we also focus on other measurements like AUC (Area under the ROC curve) and K-S Statistics.

AUC, also known as the c-statistic, measures the predictive power of a binary classification model. It equals to the area under the Receiver Operating Characteristic (ROC) curve. A good model's ROC curve should sit close to the upper-left corner and the area under the ROC curve should be close to 1.

K-S Statistic measures the distance between two distribution functions, in this case, the two levels of target variable. The greater the value is, the better the model is in discriminating two classes.

The data we have is prepared based on a marketing dataset with a binary target (1=responder, 0=non-responder). The raw dataset has over 90,000 observations, and only about 1.25% of them are marked as responders. The goal is to build a classification model that can accurately distinguish the responders from the non-responders. Variable selection and dimension reduction has been performed prior to our comparison, so that each approach will have the same training data. In addition, we assume that the prior probability of the rare event is well represented by the raw dataset. A Decision Tree model and a Neural Network Model are used for each approach with the same settings.

MODEL PERFORMANCE WITH THE IMBALANCED DATA

Let's first take a look at how the models perform with the imbalanced data. To minimize over-fitting, we'll compare all the results generated from the validation dataset, which constitute 30% of the raw dataset.

The Model Comparison Node in SAS® Enterprise Miner™ provides the following results (see Table 1). As can be seen, The Neural Network model works better in this case.

The ROC curves of both models are close to the base line, thus indicating weak predictive powers (see Figure 1).

Model	ROC Index	K-S Statistics	Misclassification Rate
Neural Network	0.647	0.19	0.012
Decision Tree	0.566	0.08	0.012

Table 1 Model Performance

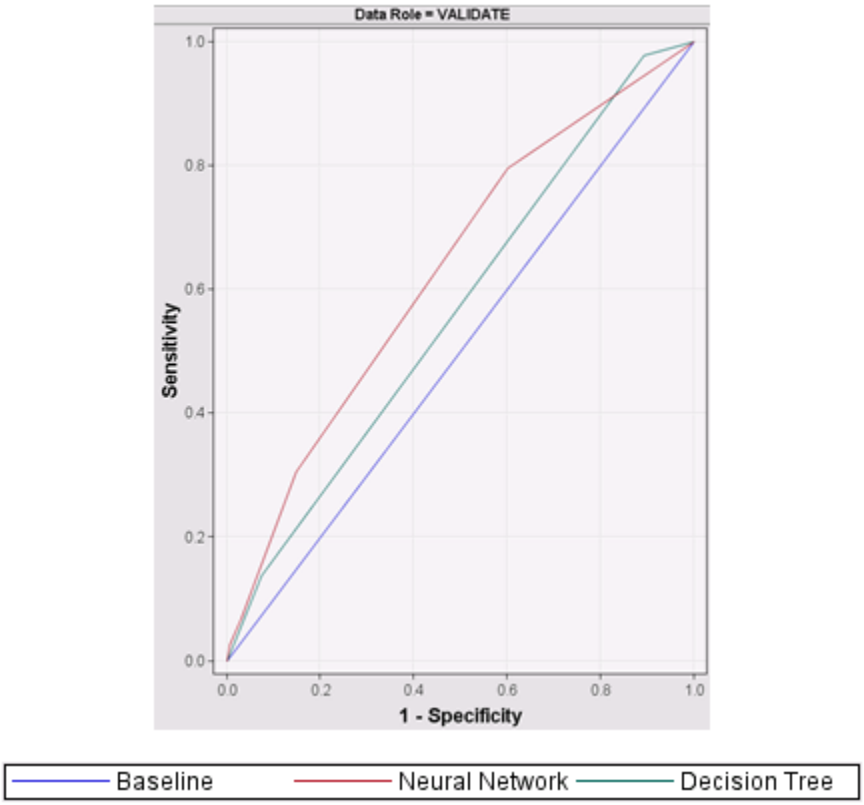


Figure 1: ROC Curve

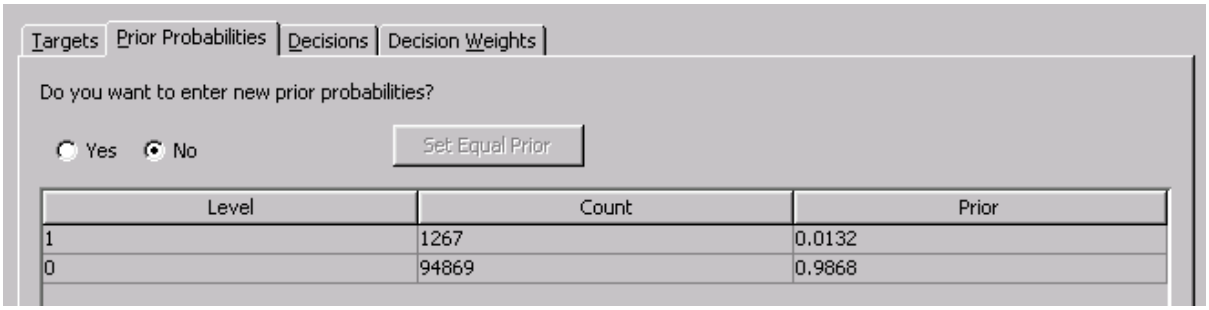
The classification table (see Output 1) shows more problems. Both models are reluctant to choose any positive class, because by default, if the 'decision' is not applied, both models try to minimize the misclassification rate. And for imbalanced data, choosing only the majority class is the 'correct' way to do it, provided that costs of misclassification error are equal.

Model Node	Model Description	Data Role	Target	False Negative	True Negative	False Positive	True Positive
Neural3	Neural Network	TRAIN	TARGET	792	60840	0	0
Neural3	Neural Network	VALIDATE	TARGET	341	26076	0	0
Tree4	Decision Tree	TRAIN	TARGET	792	60840	0	0
Tree4	Decision Tree	VALIDATE	TARGET	341	26076	0	0

Output 1: Event Classification Table

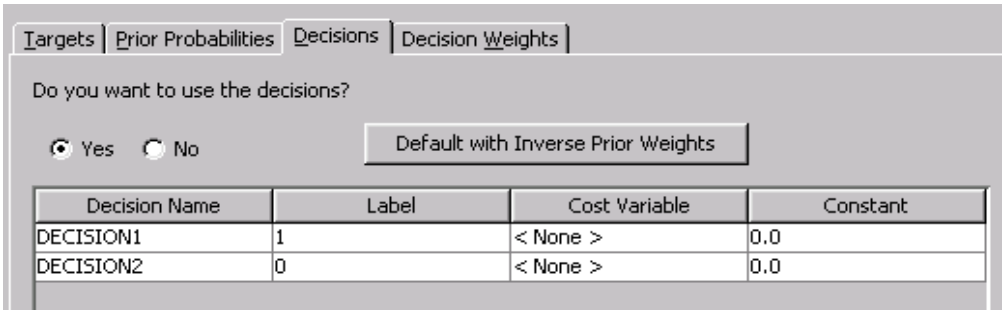
THE COST-SENSITIVE APPROACH

The Decision Node inside SAS® Enterprise Miner™ comes handy for users to adjust prior probability and decision costs. We keep the default prior probability for all the approaches to make a simple and fair comparison (see Display 1).



Display 1: Decision Node – Prior Probabilities

On the Decisions tab (see Display 2), we choose 'Yes' to apply decisions. We also have the option to assign cost for each decision. To simplify things, we'll not modify this part.



Display 2: Decision Node – Decisions

Next, we set up the decision matrix on the Decision Weights tab (see Display 3). In this section, we can assign different values (can be positive or negative) to each decision and its outcome. Users can specify whether to maximize the revenue or minimizing the cost. This is very useful when your goal is to maximizing the profit and you are familiar with the revenue or cost of each situation.

Display 3: Decision Node – Decision Weights

Again, to simplify the problem, we assume that there is no misclassification cost and the revenue of correctly identifying a rare event is much greater than that of identifying a majority event. How much greater? We can assume it's to the same extent of how rare the minority event is. On the Decision Tab

(see Display 2), there is an option for us to assign the decision weights based on target events' inverse prior probability.

Let's take a look at the result in Table 2:

Model	ROC Index	K-S Statistics	Misclassification Rate
Neural Network	0.647	0.19	0.0129
Decision Tree	0.566	0.08	0.0132

Table 2: Model Performance

The Neural Network Model has a higher ROC Index, but it shows the same value as before, so does its misclassification rate. If we look closer we can find that the model's sensitivity has increased slightly as can be seen in Figure 2, while the area under the curve remain the same.

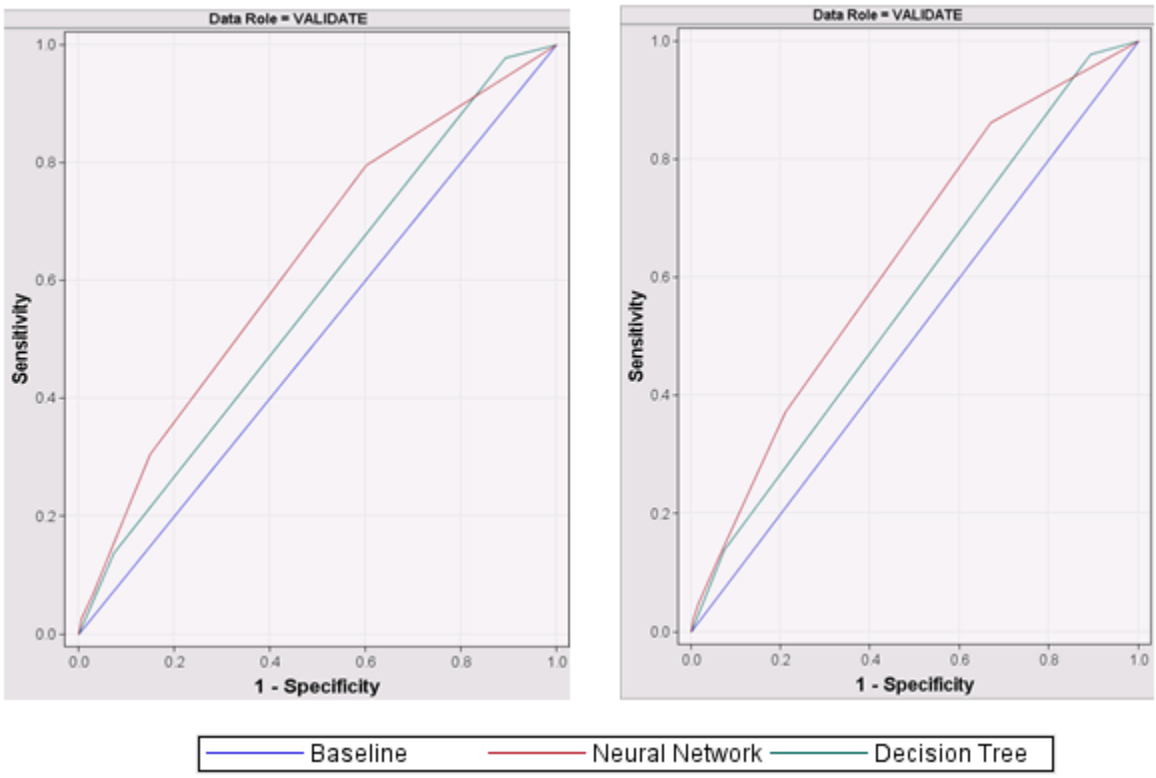


Figure 2: ROC Curve (left: before decision weight is applied; right: after decision weight is applied)

When decisions are applied, there will be an additional section called Decision Table (See Output 2) in the result output. This table provides classification with the profit matrix involved and tries to get the maximum expected profit.

Data Role=VALIDATE Target Variable=TARGET

Target	Outcome	Target Percentage	Outcome Percentage	Frequency Count	Total Percentage
0	0	99.2186	51.6183	13460	50.9520
1	0	0.7814	31.0850	106	0.4013
0	1	98.1713	48.3817	12616	47.7571
1	1	1.8287	68.9150	235	0.8896

Output 2: Decision Table

THE UNDER-SAMPLING APPROACH

This approach is to balance the data by randomly dropping records of the majority event. It's probably the most straightforward solution, but it may easily lose useful information.

In SAS® Enterprise Miner™, we use four Sample Nodes with different random seeds, to generate four datasets with the rare event proportion of 12.5%. After sampling, I set the prior probabilities to be the same as the raw dataset.

From the result below (see Table 3), we can see the models' performances rise and fall, indicating instability.

Model	Sample	ROC Index	K-S Statistics	Misclassification Rate
Neural Network	Sample 1	0.625	0.17	0.125
Neural Network	Sample 2	0.626	0.15	0.125
Neural Network	Sample 3	0.621	0.17	0.125
Neural Network	Sample 4	0.653	0.21	0.125
Decision Tree	Sample 1	0.589	0.09	0.125
Decision Tree	Sample 2	0.602	0.11	0.125
Decision Tree	Sample 3	0.614	0.17	0.125
Decision Tree	Sample 4	0.616	0.10	0.125

Table 3: Model Performance

From their ROC curves (see Figure 3), we also can see that using random under-sampling does not bring much improvement to the overall model performance.

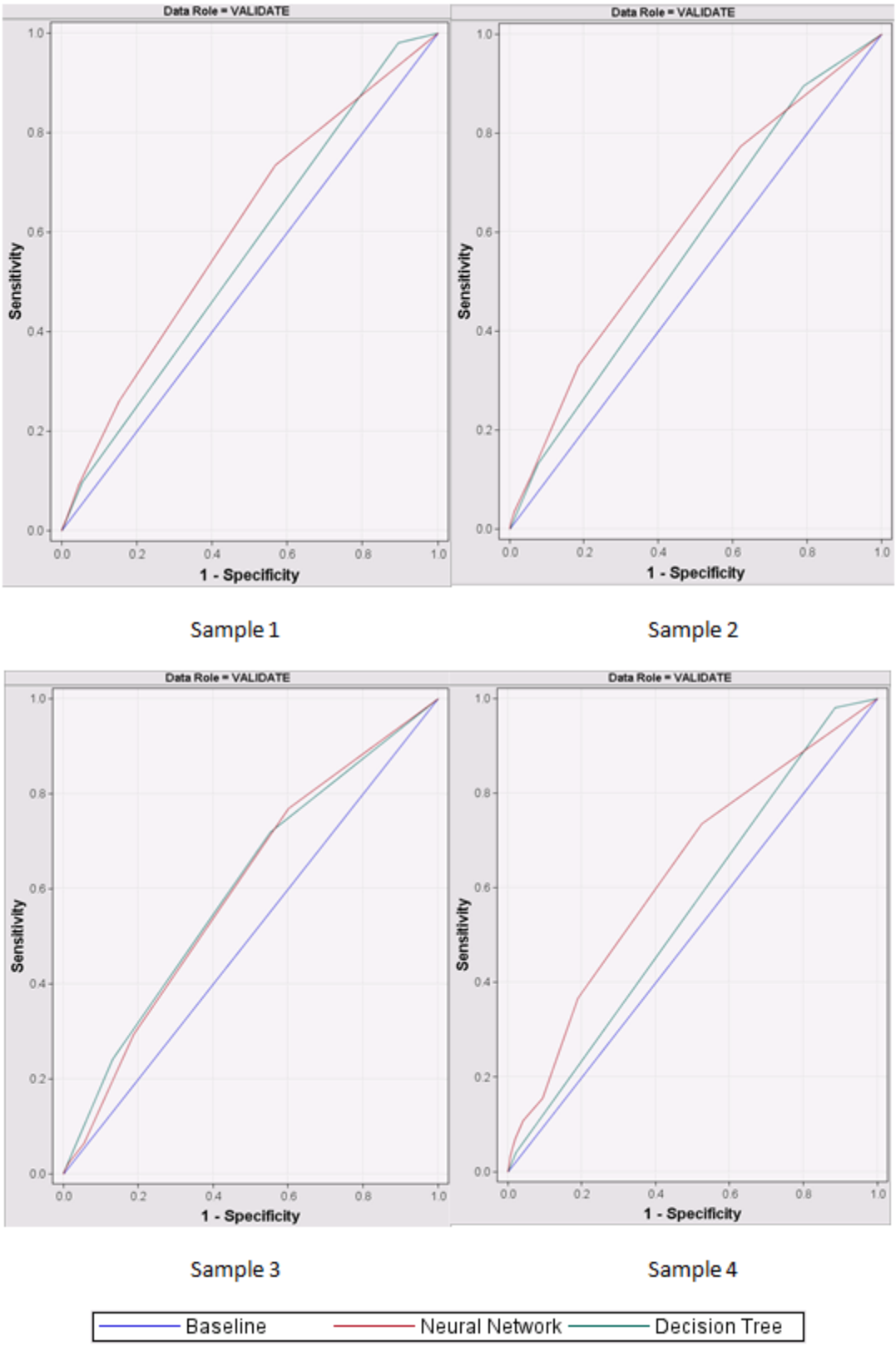


Figure 3: ROC Curves based on different data sample

THE OVER-SAMPLING APPROACH

Another way to balance the data is to randomly replicate the records that have rare events, in other words, to resample the rare cases with replacement.

To perform the over-sampling, we use the SURVEYSELECT procedure in a SAS Code node. Again, we set the proportion of the rare cases to 12.5% just like in the under-sampling approach, and we keep the original prior probability.

Based on the results in Table 4 and Figure 4, we can see both models' ROC Index and K-S Statistics have been improved with a small price of drop in accuracy.

Model	ROC Index	K-S Statistics	Misclassification Rate
Neural Network	0.754	0.37	0.125
Decision Tree	0.68	0.23	0.125

Table 4: Model Performance

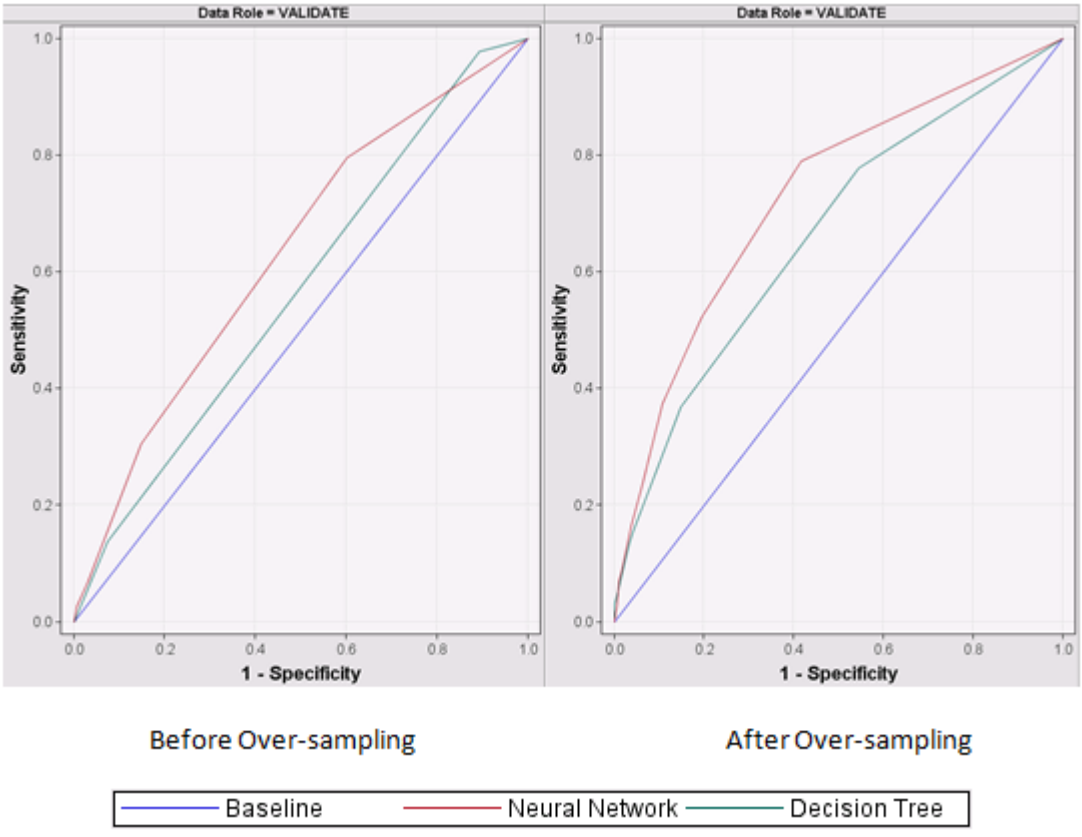


Figure 4: ROC Curve

THE SMOTE APPROACH

Synthetic Minority Over-sampling TEchnique is a special form of over-sampling. Instead of randomly selecting from the same rare cases, it synthetically generates rare cases based on the rare cases and their nearest neighbors in an effort to enlarge the model's decision boundary.

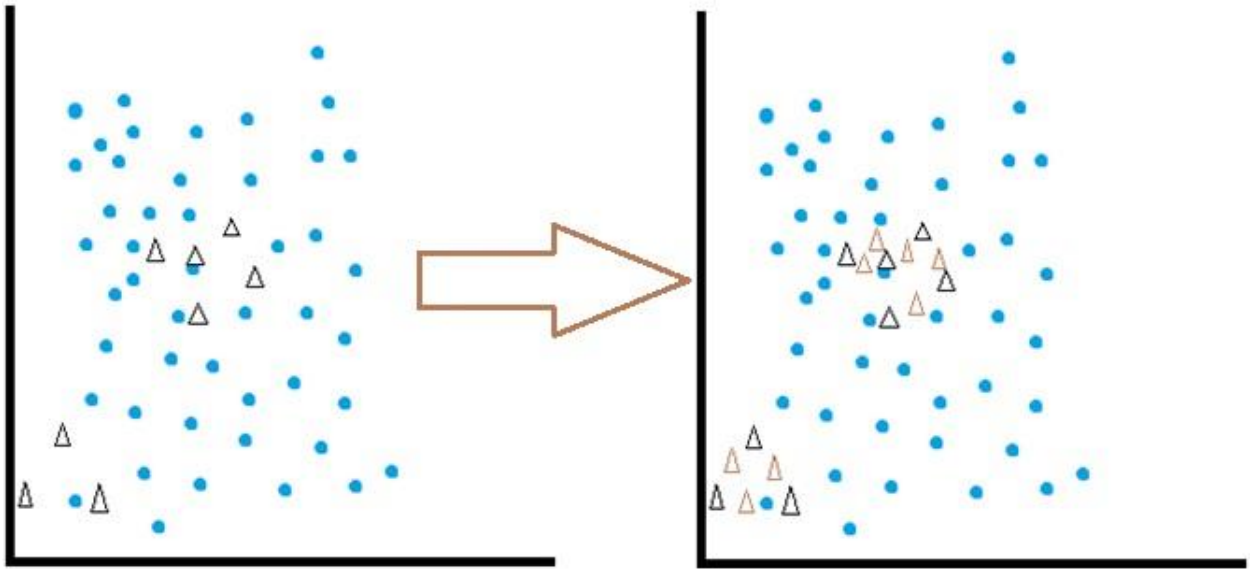


Figure 5

As an example in Figure 5, the triangles are rare cases and the blue dots are the majority cases. New rare cases are created between the existing rare cases and their nearest neighbors.

The first step is to identify a rare case's K nearest neighbors. As illustrated in Figure 6, for rare case M, we find k=5 other rare cases close to it based on their distances. Next, randomly select a point between two points and generate a new case. For example, we create point m1(c1,c2) between M(a1,a2) and M1(b1,b2), where c1=a1+(b1-a1)*rand('UNIFORM') and c2 = a2 + (b2-a2)* rand('UNIFORM').

For multi-dimensional data, the algorithm works in the same way. For observations with interval variables, we can directly calculate their Euclidean distances, but for the ones with categorical variables, we need to first apply dummy variable transformation to have them in a numeric form and then calculate the distances. If the interval variables are in different scales, it's best to standardize the values first before calculating the distances.

In our case study, we use a Principle Component node as an intermediate step for dimension reduction, so that we'll be computationally more efficient when calculating the distances. It also generates dummy variables for categorical variables and uses them as interval variables in the principle components analysis. As a result, we end up with 20 interval variables as our input variables. Next, we use the MODECLUS procedure to calculate 10 nearest neighbors for our rare cases and randomly generate 10 cases in between (See the SAS code in Appendix). Finally, the proportion of rare event becomes 12.5%.

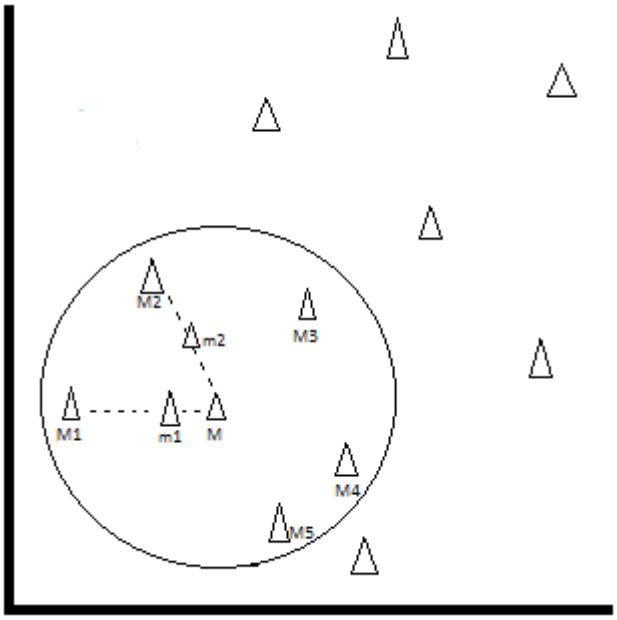


Figure 6

The ROC Index and K-S Statistics (see Table 5 and Figure 7) show great improvement compared with randomly over-sampling. The misclassification rate of the Neural Network model is also smaller than other sampling based approaches.

Model	ROC Index	K-S Statistics	Misclassification Rate
Neural Network	0.843	0.52	0.104
Decision Tree	0.714	0.32	0.125

Table 5: Model Performance

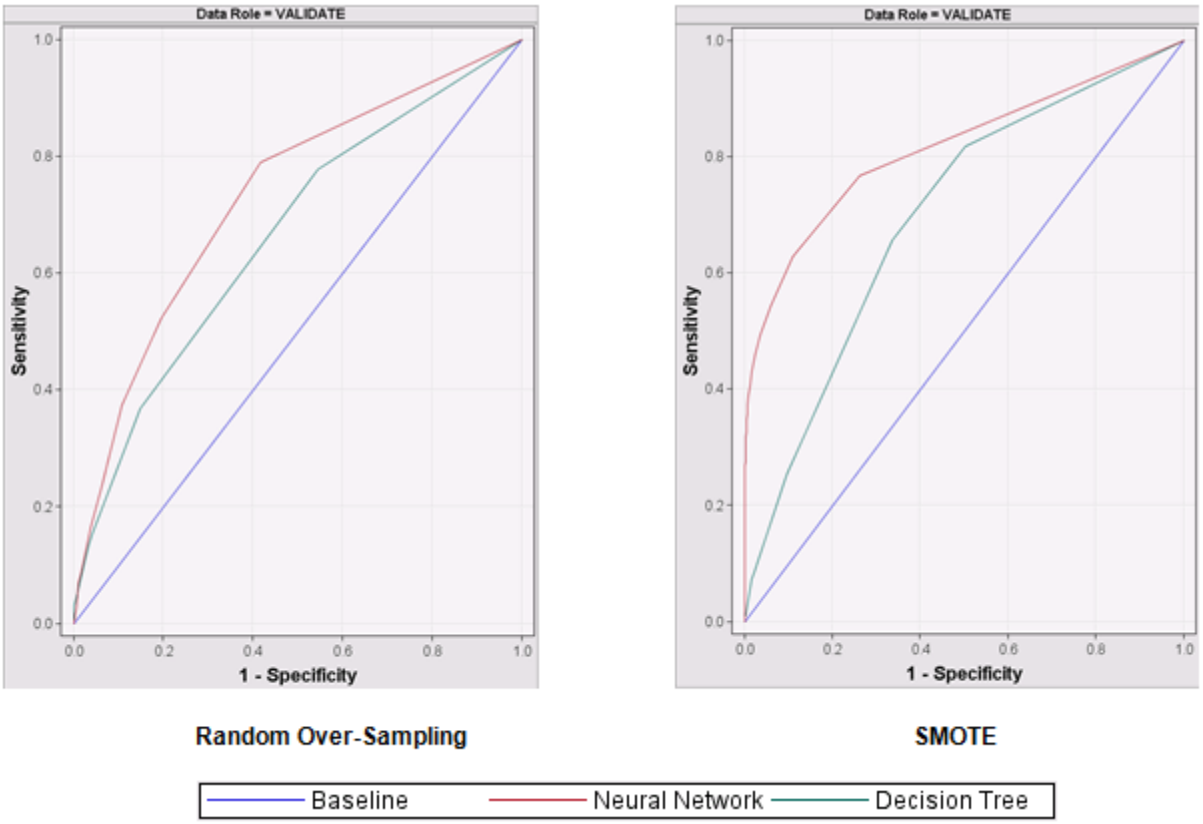


Figure 7: ROC Curve

Considering the fact that we use 10 nearest neighbors for each rare case, we may risk generating noise data that affect the model's accuracy. Therefore, we can either set a smaller number of neighbors or a distance threshold to generate new cases only when two existing cases are close enough. As a further experiment on the Neural Network model, we set 3 as the limit for the distance to see if it can make any difference in model performance. We can see a slight improvement from the following result in Table 6:

Sample	Model	Minority Case Proportion	ROC Index	K-S Statistics	Misclassification Rate
SMOTE	Neural Network	12.54%	0.843	0.52	0.104
SMOTE (adjusted)	Neural Network	12.35%	0.878	0.6	0.097

Table 6: Model Performance

However, the distance 3 we choose may not necessarily be the optimal distance in this case. Further experiment and calculation are needed to search for the optimal distance. Even so, with the constraint on the distance, the model performance has been improved.

CONCLUSION

This paper discusses some of the most used approaches to handle imbalanced data for classification model. With controlled dataset and fixed prior probability, we try to compare each approach's effectiveness on model performance.

Applying the profit matrix to a classification model can affect its decisions in order to maximize the expected profit. However, it won't improve a model's overall performance or discriminating power.

The Under-Sampling approach can slightly improve or hinder the classification, for it suffers from instability, given that large amount of data which might contain useful information or noise are excluded from analysis.

Over-sampling rare cases at random may prevent the model from overlooking additional rules to build the decision boundary, because classification rules are often ignored when lacking sufficient samples in the training data. However, this approach adds no additional information to the model and has the risk of duplicating noise samples.

SMOTE algorithm improves the balance of the data by adding new samples close to existing rare cases. Although it takes a few more steps to prepare the training data before this approach can be applied, the promising results may say it's worth the trouble.

All the discussed approaches can be applied using SAS® Enterprise Miner™ (some may require a little coding with the SAS Code node).

REFERENCES

Tan, Pang-Ning; Steinbach, Michael; Kumar, Vipin. 2006. An Introduction to Data Mining. Addison-Wesley.

Chawla, N. 2009. "Mining When Classes are Imbalanced, Rare Events Matter More, and Errors Have Costs Attached". The SIAM Data Mining Conference 2009. Sparks, NV. Available at https://www.siam.org/meetings/sdm09/chawla.pdf

Chawla, N.; Bowyer, K.; Hall, L.; Kegelmeyer, P. 2002. "SMOTE: Synthetic Minority Over-sampling Technique". Journal of Artificial Intelligence Research, 16:321-357

Lazarević, Aleksandar; Srivastava, Jaideep; Kumar, Vipin. 2004. "Data Mining for Analysis of Rare Events: A Case Study in Security, Financial and Medical Applications". Available at http://www-users.cs.umn.edu/~aleks/pakdd04_tutorial.pdf

Brennan, Peter. "A comprehensive Survey of Methods for Overcoming the Class Imbalance Problem in Fraud Detection". 2012. Available at http://www.dataminingmasters.com/uploads/studentProjects/thesis27v12P.pdf

SAS Institute Inc. 2012. SAS® Enterprise Miner™ 12.1 Extension Nodes: Developer's Guide . Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENTS

This work was made possible by the unconditional support of Mr. Arden Buchanan, Director of Database Marketing in GuideWell Connect. In addition, a thank you to Dr. Morgan Wang, professor of Data Mining program in University of Central Florida, for his valuable lessons and encouragement over the years.

CONTACT INFORMATION

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

Ruizhe Wang
GuideWell Connect
904.905.1950
Ruizhe.wang@guidewellconnect.com

Novik Lee
GuideWell Connect
904.436.4285
Novik.lee@guidewellconnect.com

Yun Wei
GuideWell Connect
904.436.4283
Yun.wei@guidewellconnect.com