Investigating the Performance of Smote for Class Imbalanced Learning: A Case Study of Credit Scoring Datasets

Maira Anis (PhD Scholar)
Mohsin Ali (Postdoc Fellow)
University of Electronic Science and Technology of China

Doi: 10.19044/esj.2017.v13n33p340 URL:http://dx.doi.org/10.19044/esj.2017.v13n33p340

Abstract

Classification of datasets is one of the major issues encountered by the data mining community. This problem heightens when the real world datasets is also imbalanced in nature. A dataset happens to be imbalanced when the numbers of observations belonging to rare class are greatly outnumbered by the observations of another class. Class with greater number of observation is called the majority or the negative class, while the other with rare observations is referred to as the minority or the positive class. Literature represents number of resampling techniques that address the problem of class imbalance. One of the most important strategies is to resample the datasets that aim to balance the number of minority or majority observations by over-sampling or under-sampling respectively. This paper aims to investigates and analyze the performance of most widely used oversampling procedure Synthetic Minority Oversampling Technique (SMOTE) for different thresholds of oversampling using four classifiers for three credit scoring datasets.

Keywords: Classification, Imbalanced Datasets, Oversampling, SMOTE, Credit Scoring

Introduction

Rapid advancements in technology have increased the number of its user's manifold that gave rise to larger datasets. Credit Scoring (CS) datasets are usually highly skewed with high number of NDF or credit worthy applicants that, in comparison under, represent the DF applicants. One of the major concerns of financial institutions (FI's) and banks is the classification of NDF's from the DF's. Correct prediction of such applicants can lead to saving huge revenue for the FI's and banks. In recent years, CS has gotten greater attention from the data mining community because of the enormous

implications towards generating revenues, reduction of financial risks, evaluation of credit risk, and maintaining the cash flow (Abrahams & Zhang, 2008; Baesens et al., 2009).

Traditionally, CS is categorized into two types on the basis of the data used and task assigned (Bijak & Thomas, 2012) i.e., application scoring and behavioral scoring. Application scoring will estimate the probability of applicants to default for some given time interval. Usually, the data used in training these models contains the demographic and financial information of the applicants alongside with their good or bad status recorded for any other time interval. Application scoring of applicants is performed before the loan is granted to the applicants. The second scoring is behavioral scoring and it is performed when the loan has been granted to the applicants. Behavioral scoring also estimates the likelihood of the applicant to default at a given interval of time. Data used for this scoring is based on the performance of loan repayment by the customers with their good or bad status. With this scoring, FI's are able to monitor the customer behavior that can further lead to making decisions about their status, either non defaulter (NDF)/good or defaulter (DF)/bad. For any FI to generate maximum profit, it is believed to predict the customers accurately for varied time of intervals (e.g., 2^{nd} month, 4^{th} month, 6^{th} month etc). This accurate prediction of customers, however, flags the DF customers with high risk and allows FI to take any necessary preemptive measure that can save them from huge losses.

Classification of scoring datasets is a vigorous task in making critical decisions for a customer in granting or refusing a loan. In 2008, the financial crunch has greatly emphasized the importance of customer lending (Benmelech & Dlugosz, 2010). CS is a fundamental problem faced by the data mining and operational research community (Basen et al., 2009). CS models are developed for the classification of customers to DF or ND customers. However, these datasets are highly imbalanced with more NDF (good) applicants in comparison to DF (bad) applicants. Conventionally, for imbalanced classification, algorithms are biased towards the class with more number of observations by predicting the overall accuracy. Thus in order to increase the true prediction i.e., True Positive Rate (TPR) of classifier, resampling techniques are implemented which include under-sampling technique or over-sampling technique of datasets. Under-sampling (Drummond & Holte, 2003) is a process of changing prior probabilities for the majority class, whereas oversampling increase the number of the minority class applicants (Drummond & Holte, 2003). For under-sampling technique and oversampling technique of datasets, a lot of techniques have been devised which aim at increasing the prediction rate for the minority class (Chawla et al., 2002; He et al., 2008; Jiang et al., 2016; Anis & Ali, 2017). Generally, oversampling techniques have been found to perform

much better than the under-sampling techniques because the original information is not lost as it does in under-sampling. Thus, this paper aims to investigate the performance of mostly used oversampling technique: Synthetic Minority Oversampling Technique (SMOTE) with four classifiers, Support Vector Machine (SVM), k-Nearest Neighbor (k-NN),Random Forest (RF), and J48 over three widely used scoring datasets: German Credit Approval (GCA), Australian Credit Approval (ACA), and Give Me Some Credit (GMSC).

Section 2 briefs with reference to comparison studies for SMOTE . Section 3 elaborates the general procedure of SMOTE and the classification algorithms adopted for this study. Section 4 gives the description of CS datasets and the evaluation metrics that have been used to assess the performance of the oversampling techniques for these datasets. Section 5 contains the experimentation results carried out for this research study. Lastly, section 6 concludes the whole study and the results obtained from the data analysis.

Related Work

CS is a fundamental issue for the banking industry as even percentage improvement in fraction in detecting the DF cases contributes to saving from huge amount of losses (Lee & Chen 2005; He et al., 2010) and building the reputation of FI's (Kennedy, 2013). This early detection can be very helpful to FI's in predicting the credit worthiness and in later granting or refusing a loan to the applicants (Chi & Hsu, 2012). Accurate prediction of CS applicants can maintain the viability of the customer lending process.

Evaluating the credit risk of applicants is an active and demanding research area in managing the financial risk. Coherently, a lot of research in developing CS models has been done using data mining and statistical techniques with remarkable contributions to the field (Kennedy et al., 2011). Therefore, these models predict the risk of the applicants by classifying them as ND or DF (Hand & Henley, 1997). This classification evaluates the credit risk associated with such applicants and the credit applications are either accepted or rejected on the basis of the classification performed (Han, et al., 2006). Thus, an accurate prediction of such applicants could possibly generate revenues along with building trust for the FI's towards their customers. Most of the work done for the CS was presented in the literature review studies. However, a very limited number among them present some novel strategies designed to model credit risk management. In this literature survey, we will cover some important studies that include both issues aforementioned.

A detailed literature review of classification techniques to CS was first given by hand and Henley in 1997. In their study, they discussed some

important classification issues related to the CS. There have also been some other comprehensive literature surveys that are focused on some of the classification methodologies. These studies are conducted by Xu et al. (2009) and Kennedy et al. (2010), Shi (2010), Lahsasna et al. (2010), Kennedy et al. (2011), Kennedy et al. (2013), and Nurlybayeva and Balakayeva (2013). However, some of the researchers have used semi supervised classification techniques for CS. On the other hand (Kennedy et al., 2011), Marqués et al. (2013) gave an efficient literature review by considering different imbalance ratios for CS datasets. Nevertheless, this study was limited when considering only some of the resampling techniques. Lessmann et al. (2015) performed a literature survey for about 50 papers that ranged from the year 2000 to 2014. Louzada et al. (2016) presented a systematic literature survey of all the binary classification methods inducted for CS studies from 1992 to 2015. A comparison study (Zakirov et al., 2015) for resampling techniques investigated the classification accuracy of certain algorithms. In this study, it was found that the classification algorithm Random Forest outruled other algorithms with under-sampling. Bennin et al. (2017) proposed a hybrid idea in building a credit score model using deep learning and genetic programming. Peng et al. (2017) proposed a random walk-based personal CS model to compute the trade reference rank for the CS applicants.

Methodology

This section explains the over sampling technique SMOTE and how this has been implemented on the datasets ACA, GCA, and GMSC along with the classification algorithms devised in this study.

Resampling Technique

SMOTE

SMOTE is an oversampling technique proposed by Chawla et al. (2002). This technique generates new minority class samples synthetically. This synthetic generation of minority class has created new samples in the vicinity of existing minority samples using k-NN (k Nearest Neighbor). More specifically, each minority samples is taken for the generation of new samples. However, the k-NN are chosen randomly along the line joining any of the k nearest minority samples for the creation of new balanced dataset.

Classification Techniques

In this study, we will perform supervised classification. For any model induced by supervised classification, a labelled set of examples is required to train and validate the model. On the other hand, the test dataset is comprised of formerly unseen examples which are later assigned by labels by the trained model. Every model is trained using the classification

algorithms. This study will be conducted by three algorithms which include: Random Forest, J48, Support Vector Machine, and kNN.

Decision Trees

Random Forest

Decision Tree gives the technique of classifying the samples by producing a tree like structure. Internal nodes of the tree represent the choice (binary) for each attribute. However, the branch of the tree signify the outcome for the desired choice (Zhang & Zhou, 2004). In recent years, many kinds of decision trees have been introduced by the researchers e.g., J48, CART. Among them, the most widely used classifier is Random Forest (RF) (Breiman, 2001). RF represents the collection of such trees that are produced to evade the risk of instability and minimalize the possibility of over training of samples (Bhattacharyya et al., 2011). These trees are also created to reduce over fitting using pruning techniques. It is a technique that progressively reduce the nodes without upsetting the overall performance of the classifier.

J48

C4.5 or J48 is a decision tree that creates pruned trees. This tree structure was established by Hunt et al in 1966 and developed by Quinlan in 1986. This is an extension to another decision tree ID3. C4.5 overcomes the shortening of ID3 algorithm. This classifier is also called a statistical classifier because of its recursive partitioning of the data. In building the tree, c4.5 take into consideration all possible tests to split the data. For discrete attributes, one test with outcome is considered with as many distinct values of that attribute. However, for continuous attribute, binary set with distinct values of attribute is considered. This process continues repeatedly for all of such attributes in the training data. For more detailed information on C4.5, the reader is referred to the study of Quinlan (1986).

K-NN

K Nearest Neighbor is also called IBK algorithm. An instance based selection technique is used by this algorithm. This implies that the classification of instances is done with the help of specific instances (Aha & Kibler, 1991). Each instance is described in an n-dimensional space, where n represents the number of attributes. However, it is not necessary for all the attributes to be defined in n-dimensional space. In that case, the missing values of those attributes are accepted (Aha, 1989) among all the n-attributes. In addition, target variable defines the class of the instances, whereas the other attributes are called predictor attributes.

Primarily, kNN algorithm is a function that defines how it maps instances to the given classes. For this purpose, it utilizes the past information of the instances that is learned in building a model during the training phse. Although after training, the set of instances is changed, but kNN classifies the instances according to how similar the new instances are to the past instances.

Support Vector Machine

Support Vector Machine (SVM) was established by Vapnik (1995). This classifier maps the linear functions to higher dimensional space. Mapping to higher dimensional space ultimately helps in solving the complex problem linearly with less complexity. Transformation of data to higher dimensional space is done using the kernel function. A kernel function is a linear mapping from original data to high dimensional space. The mathematical formation of such problem is as shown below:

$$k(x_1, x_2) = \langle \varphi(x_1), \varphi(x_2) \rangle$$

Here, $\varphi: X \rightarrow Y$ embodies a mapping from data $X(n-\text{features})$ to the higher dimensional space Y. This mapping generates a hyper plane that classifies the data samples to their relevant classes. Mathematically, this hyper plane is given by the following equation:

$$w.\emptyset(x) + b = 0$$

This hyper plane is ensured to have maximum separation between the data samples from both classes. Finally, the SVM classification is defined as:

$$\sum_{i=1}^{n} \propto_i y_i k(x_i, x) + b = 0$$

Experimental Analysis

Datasets

This study is implemented using three datasets. Two datasets, GCA and ACA, are taken from University of California (UCI) repository (Asuncion & Newman, 2010) and have been widely used for CS studies (Baesens et al., 2003; Somol et al., 2005; Huang et al., 2006; Tsai & Wu, 2008; Chang & Yeh, 2012; Liang et al., 2014). Whereas GMSC is taken from Kaggle repository*. These datasets have been found publicly and they contain different imbalance ratios with varied number of attributes and number of observations. For all the datasets, the target variable is either good/bad or negative/positive that will represent ND or DF applicants. Table 1 gives the original distribution of the data in terms of the number of their majority and minority instances, number of attributes, and their imbalance ratio (IR). An IR of any dataset can be defined as the ratio of majority instances to its minority instances.

Table 1. Dataset Characteristics

Dataset	Total Instances	Majority Instances	Minority Instances	Attributes	IR
ACA	690	383	307	14	1.24
GCA	1000	700	300	21	2.3
GMSC	1,50000	139974	10026	11	13.96

Note: * http://www.kaggle.com/c/GiveMeSomeCredit

Dataset Preparation

In data preparation, standard pre-processing steps are implemented. Missing values of the data were imputed by mean value of the attribute. After this, data partition was performed. For this purpose, each dataset is divided into its training and testing subsets. Following the benchmarking CS studies (baesens et al., 2009), training comprise two-third of the total dataset, while the testing contains one-third of the data.

Evaluation Metrics

In order to measure the predictive performance of classification algorithms, four evaluation metrics were used. These evaluation metrics are based on confusion matrix as shown in the following Table 2. It is a matrix of order 2x2 with four elements i.e., TP (True Positive), TN (True Negative), FP (False Positive), FN (False Negative).

Table 2. Confusion Matrix

Actual Class	Predicted Class	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

Where TP is the accuracy of the positive (bad) examples predicted as positive (bad). TN is the accuracy of negative (good) examples classified correctly as negative. FP is the accuracy of negative (good) examples classified incorrectly as positive (bad). FN is the accuracy of positive (bad) examples predicted incorrectly as negative (good).

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Where Recall is also called sensitivity or TPR (True Positive Rate).

Area under ROC (Receiver Operating Characteristic) curve or Area Under Curve (AUC) is another evalution metric used to measure the performance of imbalanced datasets. ROC is a two-dimensional curve representing a compromise between True Positive and False Positive rate. Whereas, the area under ROC curve or AUC is used in assessing the accuracy of the classifiers. Classifiers giving high values of AUC are considered to be best.

Synthetic Generation using SMOTE

For each dataset, the oversampling technique SMOTE generates the bad instances for the minority class for different thresholds i.e., 50%, 70%, and 100%. In literature, SMOTE has been carelessly applied with different thresholds that increase the number of bad instances in comparison to good instances. This results in oversampling the minority class that ends up with greater number of minority instances than the majority instances. This study concludes that bad instances may not exceed the good instances in number. For this course, a procedure that leads to bad sample generation for minority class, either equal in number to good instances or less than them, was developed. The procedure is as follows.

For each minority class sample $x_i, (i = 1,2, \ldots . n)$

- Find k-NN of x_i for $(i = 1,2, \ldots . n)$.
- Choose j neighbors of x_i randomly from the k neighbors. Value of j always depends on the number of neighbors chosen. For this study, $j = 5$.
- Randomly generate the synthetic instances along the lines of the j selected neighbors.
- Repeat these steps for all the oversampling thresholds i.e., 50%, 70%, 100%

The required thresholds are changed for those sets where minority instances exceed majority instances.

In this study, we will perform supervised classification. For any model induced by supervised classification, a labelled set of examples is required to train and validate the model, whereas the test dataset is comprised of formerly unseen examples which are later assigned by labels by the trained model. Every model is trained using the classification algorithms.

In order to achieve more robustness of any classification model, the training and validation sets have extreme importance. Firstly, the training process of any classification model was performed. Once the model has been learned, the validation set is used to endorse the performance of the predictive model (Bellazzi & Zupan, 2008). The procedure adopted for this process is as follows.

For validation of any model, some data is excluded from the training set. With the rest of the training data, the model is learned and is thought to be trained. A subset of examples that was excluded from the training data is now included, and this helps in validating the performance of the classifier. This process is called cross-validation. For this study, k-fold cross-validation is performed with k=10. SMOTE is performed using the WEKA toolkit (Holmes et al., 1994).

Table 3. Classification results for the classifier RF

CS Datasets	Oversampling Technique		Precision	Recall	F-measure	AUC
ACA	SMOTE	20%	0.864	0.873	0.868	0.913
		35%	0.880	0.863	0.871	0.922
GCA	SMOTE	50%	0.590	0.485	0.533	0.753
		70%	0.583	0.554	0.569	0.746
		100%	0.598	0.545	0.570	0.769
GMSC	SMOTE	50%	0.442	0.259	0.327	0.777
		70%	0.438	0.266	0.331	0.782
		100%	0.448	0.266	0.334	0.778

Table 4. Classification results for the classifier J48

CS Datasets	Oversampling Technique		Precision	Recall	F-measure	AUC
ACA	SMOTE	20%	0.843	0.892	0.867	0.895
		35%	0.843	0.892	0.867	0.895
GCA	SMOTE	50%	0.549	0.495	0.521	0.680
		70%	0.571	0.554	0.563	0.700
		100%	0.487	0.545	0.514	0.651
GMSC	SMOTE	50%	0.524	0.206	0.295	0.823
		70%	0.532	0.193	0.283	0.766
		100%	0.536	0.190	0.280	0.822

Results and Discussion

In this CS study, an oversampling technique SMOTE is implemented to increase the number of bad instances for the underrepresented minority class. Four classifiers, RF, J48, k-NN and SVM, were used over three CS datasets, ACA, GCA, and GMSC. All the datasets have different imbalance ratios. As the IR of ACA is much more balanced (44.5/55.5) than the other datasets of the study, it therefore requires less number of synthetic samples to be nearly equal or equal to the good instances. For other datasets, the thresholds followed are 50%, 70%, and 100%. But for ACA, more than 35% of oversampling alters the majority and minority class. For such reason, ACA has followed only 20% and 35% increase in the bad instances. Results of these datasets for all the classifiers are given for their respective evaluation measures from Table 3 to Table 6.

Table 5. Classification results for the classifier SVM

CS Datasets	Oversampling Technique		Precision	Recall	F-measure	AUC
ACA	SMOTE	20%	0.828	0.941	0.881	0.808
		35%	0.828	0.941	0.881	0.875
GCA	SMOTE	50%	0.636	0.673	0.654	0.755
		70%	0.627	0.683	0.654	0.756
		100%	0.624	0.673	0.648	0.751
GMSC	SMOTE	50%	0.586	0.013	0.026	0.506
		70%	0.586	0.013	0.026	0.506
		100%	0.586	0.013	0.026	0.506

Table 6. Classification results for the classifier KNN

CS Datasets	Oversampling Technique		Precision	Recall	F-measure	AUC
ACA	SMOTE	20%	0.806	0.735	0.769	0.782
		35%	0.806	0.735	0.769	0.782
GCA	SMOTE	50%	0.491	0.515	0.502	0.644
		70%	0.486	0.535	0.509	0.648
		100%	0.487	0.545	0.514	0.651
GMSC	SMOTE	50%	0.252	0.223	0.237	0.587
		70%	0.250	0.226	0.237	0.588
		100%	0.246	0.238	0.242	0.592

For the dataset ACA, it is clearly seen that as the dataset is almost balanced, there is no significant difference between the classification results of the classifier except for RF that presents a slight difference for different thresholds.

For GCA, it has been noticed that different classifiers are giving varied performance for the evaluation measures. Among all the classifiers, RF is performing well in terms of all the evaluation measures. However, the performance of other classifiers i.e. J48, SVM, and k-NN has achieved best results for 70% SMOTE. For 100% SMOTE results, the performance of the classifiers is lapsed.

Among all the datasets, GMSC is the only dataset that has not performed well for low threshold of SMOTE i.e., 50%. For other variants of SMOTE (SMOTE 70% and SMOTE 100%), classifier performance has reverted back. It is because the class imbalance GMSC contains, thus, makes it impossible to give better results even after 100% SMOTE is applied. As the IR of GMSC is still high after applying SMOTE, the synthetic generation of new minority samples is not helping the classifier in predicting the minority class.

Conclusion

This study focused on the most widely used oversampling technique SMOTE. For this purpose, three CS datasets were used to comprehend the findings. All the datasets represent varied number of imbalance ratio. It has been found that the imbalance ratio of any dataset can seriously affect the results and make them bias. It was found that the datasets with nearly balance ratio does not show significant performance even after synthetic generation of the instances. Similarly, the datasets which are extremely imbalanced also perform poorly as even after the generation of synthetic samples to 100% which does not provide satisfactory importance. Thus, there is a need of not only generating the samples for the minority class, but also eliminating those samples from the majority class which may not disturb the original distribution of the data. For future work, we foresee the use of some under-sampling techniques along with oversampling techniques in balancing the dataset.

References:

1. Abrahams, C. R., & Zhang, M. (2008). *Fair Lending Compliance: Intelligence and Implications for Credit Risk Management*. Wiley, Hoboken, NJ.
2. Aha, D.W., & Kibler, D. (1991). *Instance-based learning algorithms*. Machine Learning. 6:37-66
3. Aha, D.W. (1989). *Tolerating noise, irrelevant attributes, and novel attributes in instance-based learning algorithms*. Proceedings of the IJCAI-1989 Workshop on Symbolic Problem Solving in Noisy, Novel, and Uncertain Task Environments. Detroit, MI: Computing Research Laboratory, New Mexico State University.
4. Anis, M., & Ali, M. (2017). *A novel similarity nased under-sampling of imbalanced datasets*. International Journal of Computer Science and Information Security (IJCSIS), 15(1).
5. Asuncion, A., & Newman, D.J. (2010). *UCI Machine Learning Repository*. In: School of Information and Computer Science, University of California, Irvine, CA.
6. Baesens, B., Mues, C., Martens, D., & Vanthienen. J. (2009). *50 years of data mining and OR: upcoming trends and challenges*. Journal of the Operational Research Society 60(S1) 816–823.
7. Baesens, B., Gestel, T.V., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). *Benchmarking state-of-the-art classification algorithms for credit scoring*. Journal of the Operational Research Society 54(6), 627–635.

8. Bellazzi, R., & Zupan, B. (2008). *Predictive data mining in clinical medicine: Current issues and guidelines*," Int. J. Med. Informat., vol. 77(2), pp. 81-97.
9. Benmelech, E., & Dlugosz, J. (2010). *The Credit Rating Crisis , NBER Macroeconomics Annual 2009*, Volume 24 , Acemoglu, Rogoff, and Woodford.
10. Bennin, K.E., Keung, J., Phannachitta, P., Monden, A., & Mensah, S. (2017). *MAHAKIL: Diversity based Oversampling Approach to Alleviate the Class Imbalance Issue in Software Defect Prediction*, IEEE Transactions on Software Engineering. DOI: 10.1109/TSE.2017.2731766
11. Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J.C. (2011). *Data mining for credit card fraud: A comparative study*. Decision Support Systems. 50, 602-13.
12. Bijak, K. & Thomas, L. (2012). *Does segmentation always improve model performance in credit scoring?* Expert Systems with Applications, 39, 2433–2442. 4, 90, 111.
13. Breiman, L. (2001). *Random Forests. Machine Learning*. 45(1):5-32.
14. Chang, S.Y., & Yeh, T.Y. (2012). *An artificial immune classifier for credit scoring analysis*. Applied Soft Computing Journal 12 (2), 611–618.
15. Chawla, N.V., Bowyer, K.W., Hall, L.O., & Kegelmeyer, W.P. (2002). *Smote: synthetic minority over-sampling technique*," Journal of Artificial Intelligence Research, pp. 321–357.
16. Chi, B.W., & Hsu, C.C. (2012). A hybrid approach to integrate genetic algorithm into dual scoring model in enhancing the performance of credit scoring model. Expert Systems with Applications, Vol. 39, No. 3, 2650–2661, 2012.
17. Drummond, C., & Holte, R.C. (2003). *C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling*. In Workshop on Learning from Imbalanced Datasets II.
18. Han, J., Kamber, M., & Pei, J. (2006). *Data mining: concepts and techniques. Morgan kaufmann*.
19. Hand, D., & Henley, W. (1997). Statistical classification methods in consumer credit scoring: A review. Journal of the Royal Statistical Society. Series A: Statistics in Society 160 (3), 523–541.
20. He, J., Zhang, Y.C., Shi, Y., & Huang, G.Y. (2010). *Domain-driven classification based on multiple criteria and multiple constraint-level programming for intelligent credit scoring*. IEEE transactions on knowledge and data engineering, Vol. 22, No. 6, 826-838, 2010.
21. He, H., Bai, Y., Garcia, E., & Li, S. (2008). *Adasyn: Adaptive synthetic sampling approach for imbalanced learning*, in Proceedings

of the IEEE International Joint Conference on Neural Networks. pp. 1322– 1328.
22. Holmes, G., Donkin, A., & Witten, I.H. (1994). *WEKA: A machine learning workbench*, in Proc. 2nd Austral. New Zealand Conf. Intell. Inf. Syst., Nov./Dec. 1994, pp. 357_361
23. Hunt, E.B., Marin, J., & Stone, P.J. (1966). *Experiments in Induction*. New York: Academic Press.
24. Huang, Y.M., Hung, C. M., & Jiau, H., (2006). *Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem.* Nonlinear Analysis: Real World Applications 7(4), 720–747
25. Jiang, K., Lu, J., & Xia, K. (2016). A Novel Algorithm for Imbalance Data Classification Based on Genetic Algorithm Improved SMOTE, Arab J Sci Eng. 41: 3255. https://doi.org/10.1007/s13369-016-2179-2
26. Kennedy, K. (2013). *Credit scoring using machine learning*. Doctoral thesis. Dublin Institute of Technology. doi:10.21427/D7NC7J.
27. Kennedy, K., Mac Namee, B. & Delany, S.J. (2011). *Using Semi-Supervised Classifiers for Credit Scoring*. Journal of the Operational Research Society. doi:10.1057/jors.2011.30.
28. Kennedy, K., Mac Namee, B. & Delany, S.J. (2010). *Learning without default: A study of one-class classification and the low-default portfolio problem.* In: Proceedings of 20th Irish Conference on Artificial Intelligence and Cognitive Science. 174–187.
29. Kennedy, K., Delany, S.J. & Namee, B.M. (2011). *A Framework for Generating Data to Simulate Application Scoring. In: Credit Scoring and Credit Control XII,* Conference Proceedings, Credit Research Centre, Business School, University of Edinburgh, CRC. (2011).
30. Kennedy, K., Namee, B.M., & Delany, S.J. (2013). *A window of Opportunity: Assessing Behavioural Scoring.* Expert Systems with Applications, 40(4), 1372–1380.
31. Lahsasna, A., Ainon, R., & Wah, T., (2010). *Credit scoring models using soft computing methods: A survey.* International Arab Journal of Information Technology 7 (2), 115– 123.
32. Lee, T.S., & Chen, I.F. (2005). *A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines.* Expert Systems with Applications, Vol 28, 743–752.
33. Lessmann, S., Baesens, B., Seow, H.V., & Thomas, L. (2015). *Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research.* European Journal of Operational Research 247 (1), 124–136.

34. Liang, D., Tsai, C.F., & Wu, H.T. (2014). *The effect of feature selection on financial distress prediction.* Knowledge-Based Systems 73 (1), 289–297.
35. Louzada. F., Ara, A., Fernandes, G.B. (2016). *Classification methods applied to credit scoring: A systematic review and overall comparison,* Surveys in Operation Research and Management. https://doi.org/10.1016/j.sorms.2016.10.001
36. Marqués, A., García, V., & Sánchez, J. (2013). *On the suitability of resampling techniques for the class imbalance problem in credit scoring* . J Oper Res Soc. 64: 1060. DOI: https://doi.org/10.1057/jors.2012.120
37. Nurlybayeva, K., & Balakayeva, G., (2013). *Algorithmic scoring models*. Applied Mathematical Sciences 7 (9-12), 571–586.
38. Peng, Y., Xu, R., Zhao, H., Zhou, Z., Wu, N., & Yang, Y. (2017). *Random Walk Based Trade Reference Computation for Personal Credit Scoring,* IEEE 13th International Symposium on Autonomous Decentralized System (ISADS), pp. 122-127
39. Quinlan, J.R. (1986). *Induction of Decision Trees*, in Machine Learning, Volume 1, pages 81-106.
40. Shi, Y. (2010). *Multiple criteria optimization-based data mining methods and applications:* A systematic survey. Knowledge and Information Systems 24 (3), 369–391.
41. Somol, P., Baesens, B., Pudil, P., & Vanthienen, J., (2005). *Filter-versus wrapper-based feature selection for credit scoring*. International Journal of Intelligent Systems 20 (10), 985–999
42. Tsai, C.F., & Wu, J.W. (2008). *Using neural network ensembles for bankruptcy prediction and credit scoring*. Expert Systems with Applications 34 (4), 2639–2649.
43. Vapnik, V. (1995). *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA.
44. Xu, X., Zhou, C., & Wang, Z. (2009). *Credit scoring algorithm based on link analysis ranking with support vector machine.* Expert Systems with Applications 36 (2 PART 2), 2625– 2632
45. Zakirov, D., Bondarev, A., & Momtselidze, N. (2015). *A comparison of data mining techniques in evaluating retail credit scoring using R programming,* Twelve International Conference on Electronics Computer and Computation (ICECCO), pp 1-4
46. Zhang, D., & Zhou, L. (2004). *Discovering golden nuggets: data mining in financial application*. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 34, 513-22.