24th International Conference on Business Information Systems (BIS 2021)

Applications

https://doi.org/10.52825/bis.v1i.61

Published: 23 June 2021

A Novel Example-Dependent Cost-Sensitive Stacking Classifier to Identify Tax Return Defaulters

Sanat Bhargava[1], M. Ravi Kumar[2], Priya Mehta[2], Jithin Mathews[2], Sandeep Kumar[2], and Ch. Sobhan Babu[2]

[1] Indian Institute of Technology Roorkee, Roorkee, India

[2] Indian Institute of Technology Hyderabad, Hyderabad, India

Abstract. Tax evasion refers to an entity indulging in illegal activities to avoid paying their actual tax liability. A tax return statement is a periodic report comprising information about income, expenditure, etc. One of the most basic tax evasion methods is failing to file tax returns or delay filing tax return statements. The taxpayers who do not file their returns, or fail to do so within the stipulated period are called tax return defaulters. As a result, the Government has to bear the financial losses due to a taxpayer defaulting, which varies for each taxpayer. Therefore, while designing any statistical model to predict potential return defaulters, we have to consider the real financial loss associated with the misclassification of each individual. This paper proposes a framework for an example-dependent cost-sensitive stacking classifier that uses cost-insensitive classifiers as base generalizers to make predictions on the input space. These predictions are used to train an example-dependent cost-sensitive meta generalizer. Based on the meta-generalizer choice, we propose four variant models used to predict potential return defaulters for the upcoming tax-filing period. These models have been developed for the Commercial Taxes Department, Government of Telangana, India. Applying our proposed variant models to GST data, we observe a significant increase in savings compared to conventional classifiers. Additionally, we develop an empirical study showing that our approach is more adept at identifying potential tax return defaulters than existing example-dependent cost-sensitive classification algorithms.

Keywords: goods and services tax, tax evasion, example-dependent cost-sensitive stacking classifier, example-dependent cost-sensitive ANNs,Benford's analysis, social network analysis, cosine similarity.

1 Introduction

Taxes can be classified into direct taxes, which are payable directly to the government (Eg. Income tax). These taxes cannot be transferred to any other third party, and indirect taxes, which can be shifted to a third party by the entity that is levied the tax (Eg. VAT, excise duty). The Goods and Services Tax (GST) system is an indirect taxation system introduced in India in July 2017. This paper proposes a methodology to predict potential tax return defaulters for the GST system [1].

1.1 Working of the GST system

For demonstration purposes, we take a fictitious ornament manufacturer as an example, and 10% as the GST rate levied at every step (See Figure 1). Note that throughout the paper, we

4.2.5 Variant D

Finally, we have implemented a Cost-Sensitive analog for an Artificial Neural Network [10]. To design our Cost-Sensitive ANN Classifier (CSANN), we have used the ReLU function as the activation function for the hidden layers and the logistic (sigmoid) function for the output layer. We have used equation (1) as the loss function for the neural network to incorporate the example-dependent cost-sensitive losses.

5 Experimental Results

5.1 Software Used

All the models in this work have been designed using **Python** as it is a high-level, open-source language with an extensive library ecosystem. Python can also handle large amounts of data very well.

5.2 Cost Matrix

Table 3 gives different miss-classification costs of a given taxpayer.

- **True-negative cost** ($\mathbf{C_{TN}}$) is zero. We would not incur any cost for classifying an in-time return filer (actual class zero) as an in-time return filer (predicted class zero).
- **True-positive cost** ($\mathbf{C_{TP}}$) is the expenditure towards sending SMS, calling the taxpayer and other preventive measures and the cost of associated manpower. This cost is the same for all taxpayers whose actual class is one and predicted class is one. This cost is Rs. 150.
- **False-positive cost** ($\mathbf{C_{FP}}$) is the expenditure towards sending SMS, calling the taxpayer and other preventive measures and the cost of associated manpower. This cost is the same for all taxpayers whose actual class is zero and predicted class is one. This cost is also Rs. 150.
- **False-negative cost** ($\mathbf{C_{FN}}$) depends on the $ATPM$ of each taxpayer and the expected number of days of delay in filing return by a taxpayer. This is given by $\frac{ATPM*\text{expected number of days of delay}*18}{36500} * 3 + 100$.

Here $\frac{ATPM*\text{expected number of days of delay}*18}{36500}$ is the loss incurred due to late filing of return, where interest rate is 18%. This cost is different for every taxpayer as the $ATPM$ and expected number of days of delay in filing return may vary for each individual taxpayer. We have multiplied this loss by three times and added 100 to it, in order to deter a defaulter from becoming a chronic defaulter.

5.3 Performance of Proposed Variants

In this section, we have compared the four proposed variants (variants A, B, C, and D) on tax return data vis-à-vis each other. The models have been compared on the following metrics:

5.3.1 Savings score

The savings score is defined as the relative improvement in cost using a classifier $f(S)$, compared to the cost of classifying all entries as class one or class zero, whichever is lesser [7].

$$Savings(f(S)) = \frac{Cost(f(S)) - Cost_l(S)}{Cost_l(S)}.$$

5.3.2 Balanced Accuracy Score

The balanced accuracy score is a metric for models trained on imbalanced data sets, which avoids inflated performance metrics due to the abundance of one class (in a binary classification problem). It is defined as follows:

$$\textit{Balanced accuracy} = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right).$$

5.3.3 Recall

Recall or Recall score refers to the fraction of relevant records correctly classified by the models. It is defined as follows:

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

In the context of this paper, the recall score is the fraction of tax return defaulters correctly identified by the model.

5.3.4 F1-Score

The F1-Score is defined as the harmonic mean of the precision and recall of a model. Thus,

$$\textit{F1-Score} = 2 * \left(\frac{precision * recall}{precision + recall}\right).$$

The comparative performance of the four variants is summarized in the Table 4. From the four variants, we propose Variant D (G_2=Cost-sensitive ANN) to be our proposed approach (PA) for this data set as it is the most adept at correctly predicting tax return defaulters, with the highest savings score and the highest AUC-ROC predicted on the train and test data set among the four variants.

Proposed Models	Savings Score		Balanced Accuracy		F1-Score		Recall		AUC-ROC	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Variant A	0.536	0.174	83.36%	82.23%	0.76	0.75	0.95	0.94	0.92	0.92
Variant B	0.535	0.530	83.90%	83.82%	0.77	0.77	0.91	0.90	0.93	0.94
Variant C	0.583	0.572	85.58%	85.94%	0.83	0.83	0.90	0.91	0.94	0.94
Variant D	0.520	0.582	85.14%	85.21%	0.79	0.79	0.94	0.95	0.94	0.93

Table 4. Performance of variants

Proposed Models	Savings Score		Balanced Accuracy		F1-Score		Recall		AUC-ROC	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
ANN	0.272	0.242	84.11%	83.40%	0.69	0.66	0.76	0.74	0.93	0.93
CSANN	0.393	0.440	84.31%	84.29%	0.84	0.84	0.84	0.84	0.93	0.93
CSDT	0.610	0.600	81.52%	79.00%	0.84	0.84	0.78	0.71	0.93	0.94
CSB	0.557	0.633	82.00%	78.50%	0.83	0.83	0.83	0.83	0.94	0.94
CSRF	0.495	0.517	79.15%	83.34%	0.52	0.62	0.93	0.91	0.91	0.92
Proposed Approach	0.520	0.582	85.14%	85.21%	0.79	0.79	0.94	0.95	0.94	0.93

Table 5. Performance of PA compared to existing algorithms

5.4 Performance of Proposed Approach (PA) with existing algorithms

In this section, we have compared our proposed approach's performance with some cost-sensitive algorithms mentioned in [8]. Additionally, we compare the performance of the PA with a cost-sensitive ANN [10]. We have also compared the performance of our PA with a cost-insensitive ANN. We have chosen a cost-insensitive ANN as it gave the most promising results among various cost-insensitive algorithms we experimented with, including, KNNs, Random Forests, XGBoost Classifier, AdaBoostClassifier, and Logistic Regression. The performance has been compared using the same metrics described in section 5.3. The results have been summarized in Table 5.

5.5 Model Validation for PA

5.5.1 Confusion and Cost Matrices

Tables 6 and 7 are the training and the testing confusion matrices for the PA. Tables 8 and 9 are the training matrix and the testing cost matrix for the PA. These give the true-positive cost, false-negative cost, true- negative cost, and false-positive cost of both the training and testing data sets.

	Predicted 0	Predicted 1
Actual 0	9842	2946
Actual 1	196	2742

Table 6. PA Train Confusion Matrix

	Predicted 0	Predicted 1
Actual 0	3320	1006
Actual 1	48	930

Table 7. PA Test Confusion Matrix

	Predicted 0	Predicted 1
Actual 0	0	441900
Actual 1	276796	411300

Table 8. PA Train Cost Matrix

	Predicted 0	Predicted 1
Actual 0	0	150900
Actual 1	68217	130200

Table 9. PA Test Cost Matrix

5.5.2 Training and Testing ROC Curves

Training and testing ROC curves for the PA are given in Figure 2 and 3. The AUC value of training ROC curve is 0.94 and AUC value of testing ROC curves is also 0.93. From these values, one can conclude that the model is neither under-fitting nor over-fitting.

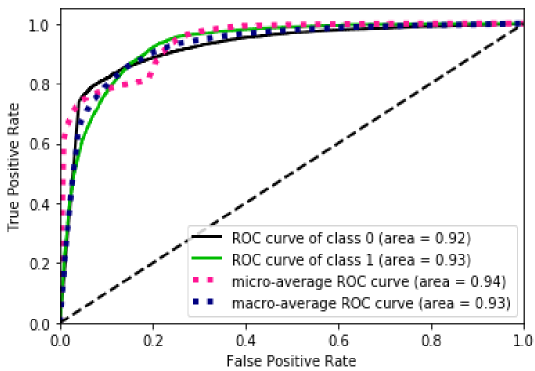

Figure 2. PA ROC on Train.

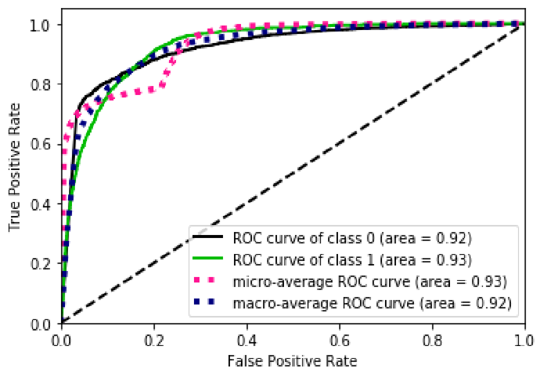

Figure 3. PA ROC on Test.

5.5.3 Savings score

To measure an example-dependent cost-sensitive algorithm's performance, we use the savings score (Section 4.3). As observed in Table 5, the savings score for the PA is 0.520 and 0.582 for the training and test sets, respectively. Since the values of the savings score for the training and testing set are reasonably high and almost similar, we can conclude that our PA is performing well.

6 Conclusion

In this paper, We propose a framework for example-dependent cost-sensitive stacked generalization comprising four variant models. We show that our Proposed Approach (PA) outperforms commonly used example-dependent cost-sensitive classifiers. We use our PA to predict whether a given taxpayer is a potential tax return defaulter or not for the upcoming month. While this framework was designed on the GST returns data for Telangana, it can be generalized to predict potential tax return defaulters using any of the four proposed variants depending on their performance, for any indirect taxation system around the world.

References

[1] S. Dani, "A research paper on an impact of goods and service tax (gst) on indian economy," *Business and Economics Journal*, vol. 07, Jan. 2016. DOI: 10.4172/2151-6219.1000264.

[2] J. Lismont, E. Cardinaels, L. Bruynseels, S. D. Groote, W. Lemahieu, and J. Vanthienen, "Predicting tax avoidance by means of social network analytics," *Decision Support Systems*, vol. 108, pp. 13–24, 2018.

[3] P. A. Bianchi and M. Minutti-Meza, "Professional networks and client tax avoidance: Evidence from the italian statutory audit regime," *SSRN Electronic Journal*, Jan. 2016. DOI: 10.2139/ssrn.2601570.

[4] V. V. Vlasselaer, L. Akoglu, T. Eliassi-Rad, M. Snoeck, and B. Baesens, "Guilt-by-constellation: Fraud detection by suspicious clique memberships," in *2015 48th Hawaii International Conference on System Sciences*, 2015, pp. 918–927.

[5] Y. Sahin and E. Duman, "Detecting credit card fraud by ann and logistic regression," in *2011 International Symposium on Innovations in Intelligent Systems and Applications*, 2011, pp. 315–319.

[6] C. Ling and V. Sheng, "Cost-sensitive learning and the class imbalance problem," *Encyclopedia of Machine Learning*, Jan. 2010.

[7] A. C. Bahnsen, D. Aouada, and B. Ottersten, "Example-dependent cost-sensitive logistic regression for credit scoring," in *2014 13th International Conference on Machine Learning and Applications*, 2014, pp. 263–269.

[8] A. C. Bahnsen, D. Aouada, and B. Ottersten, *Ensemble of example-dependent cost-sensitive decision trees*, 2015. arXiv: 1505.04637 [cs.LG].

[9] D. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, pp. 241–259, Dec. 1992. DOI: 10.1016/S0893-6080(05)80023-1.

[10] M. Kukar and I. Kononenko, "Cost-sensitive learning with neural networks," in *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98*, John Wiley & Sons, 1998, pp. 445–449.

[11] M. L. J. Nigrini Mark J., "The Use of Benford's Law as an Aid in Analytical Procedures," *Auditing: A journal of practice & theory*, vol. 41, p. 52, 1997.

[12] A. Asllani and M. Naco, "Using Benford¡¯s Law for Fraud Detection in Accounting Practices," *Journal of Social Science Studies*, vol. 2, no. 1, pp. 129–143, Jan. 2015. [Online]. Available: https://ideas.repec.org/a/mth/jsss88/v2y2015i1p129-143.html.

[13] C. Durtschi, W. Hillison, and C. Pacini, "The effective use of benford's law to assist in detecting fraud in accounting data," *J. Forensic Account*, vol. 5, Jan. 2004.