Credit Card Fraud Detection

Data Resampling, Ensemble Learning and Cost-Sensitive Learning

Yuedi Wang & Ruiyu Hu

Abstract

This project aims to solve two issues in credit card fraud detection - skewness of the data and cost-sensitivity. In our analysis, we compare the effectiveness of ensemble learning (EL) and cost-sensitive learning (CSL) based on the recall score and savings. Undersampling and oversampling are used to deal with imbalanced data. We also take both fixed cost and real financial cost into account to evaluate the problem of cost-sensitivity. Our result shows that CSL and the combination of data resampling and EL can improve the performance (savings and recall score of fraud class on test set) by 35% and 25% separately.

1 Introduction

According to the Nilson Report, fraud losses reached 21.84 billion dollars in 2016, which lead to losses in the U.S. account for 38.7% of the total volume. 46% of Americans have had their card information compromised at some point in the last five years. When constructing a fraud detection system, we need to consider the following four issues: skewness of the data (imbalanced data), cost-sensitivity, short-time response of the system and feature preprocessing. The target class of credit card fraud detection is fraud transactions and the number of fraud transactions is much lower than the number of normal transactions. Many real-world classification problems are example-dependent cost-sensitive in nature, where the costs due to misclassification vary between examples. Fraud detection is a typical example of cost-sensitive classification. The third issue is unrelated to our class content. For the fourth one, the independent variables in our original dataset are the features after principal component analysis. Therefore, our analysis is mainly to solve the first two issues. And the goal of our project is to compare the effectiveness of ensemble learning and cost-sensitive learning on credit card fraud detection and to determine which method is superior.

2 Related Work

C. Chen, A. Liaw, and L. Breiman[1] proposed two methods of dealing with highly-skewed class distributions based on 2 random forest algorithms. All of the six datasets they used have imbalance class distributions. The methodology they used include Balanced Random Forest (BRF) based on undersampling and Weighted Random Forest (WRF) based on cost-sensitive learning. Both Weighted RF and Balanced RF have performance superior to most of the existing techniques that they studied before. Between WRF and BRF, however, there is no clear winner. Due to the limitation of WRF, they found that WRF is computationally less efficient with large imbalanced data, since each tree only uses a small portion of the training set to grow. And WRF assigns a weight to the minority class, possibly making it more vulnerable to noise (mis-labeled class) than BRF.

Gary M. Weiss, Kate McCarthy, and Bibi Zabar. [2] compared three methods for dealing with data that has a skewed class distribution and nonuniform misclassification costs. Twelve of

the data sets were obtained from the UCI Repository and two of the data sets came from AT&T. In their experiments, a false positive prediction, CFP is assigned a unit cost of 1. For the majority of experiments, CFN is evaluated for the values: 1, 2, 3, 4, 6, and 10, although for some experiments the costs were allowed to increase beyond this point. Oversampling and undersampling were also employed to implement the desired misclassification cost ratios, by altering the class distribution of the training data. They concluded that there is no clear winner among oversampling, undersampling and cost-sensitive learning when comparing the performance of algorithm. However, considering the cost, when a certain data set has more than 10000 examples, the cost-sensitive learning algorithm outperforms the sampling methods. However, they manually create the cost ratio of False Negative class instead of using actual cost.

Alejandro Correa Bahnsen, Djamila Aouada and Bj¨orn Ottersten.[3] used two dataset to propose a new example-dependent cost matrix for credit scoring. The first dataset is the 2011 Kaggle competition Give Me Some Credit and the second dataset is from the 2009 Pacific-Asia Knowledge Discovery and Data Mining conference (PAKDD) competition. They also proposed an algorithm that introduces the example-dependent costs into a logistic regression. The results highlighted the importance of using real financial costs. Moreover, by using the proposed cost-sensitive logistic regression, significant improvements were made in the sense of higher saving. But, they didn't adjust the model to include features that may also depend on the estimated probabilities.

Alejandro Correa Bahnsen, Djamila Aouada and Bjorn Ottersten.[4] again announced a new comparison measure that realistically represents the monetary gains and losses due to fraud detection. Due to the confidentiality of financial information, they didn't describe the detail of the dataset. Their methodology is using the proposed cost measure, a cost sensitive method based on Bayes minimum risk. Compared with state of the art algorithms, it showed improvements up to 23% measured by cost. They had pretty good results, however, an issue related to over-estimated fraud proportion might excited.

Andrea Dal Pozzolo, Olivier Caelen, Reid A. Johnson, Gianluca Bontempi. [5] studied how undersampling affects the posterior probability of a machine learning model. They ran experiments on several UCI datasets and a real-world fraud detection dataset made available to the public. They used Bayes Minimum Risk theory to find the correct classification threshold and show how to adjust it after undersampling. Experiments on several real-world unbalanced datasets validate our results. The bias due to the instance selection procedure in undersampling was essentially equivalent to the bias the occurs with a change in the priors when class-within distributions remain stable. And undersampling does not always improve the ranking or classification accuracy of an algorithm, but when it was the case we should use pˆ instead of pˆs because the first has always better calibration.

Alejandro Correa Bahnsen, Aleksandar Stojanovic, Djamila Aouada and Bjorn Ottersten. [6] announced two different methods for calibrating probabilities were evaluated and analyzed in the context of credit card fraud detection, with the objective of finding the model that minimizes

the real losses due to fraud. They still used the dataset in [4]. Calibration of probabilities due to a change in base rates and Calibrated probabilities using ROC convex hull. The best model selected by F1-Score was not the one that had the lower cost, and the reason might be the metric was not cost sensitive and assumed a constant false negative cost, which as explained before is not the case in the direct marketing problem. The experiments confirmed that using calibrated probabilities followed by Bayes minimum risk significantly outperform using just the raw probabilities with a fixed threshold or applying Bayes minimum risk with them, in terms of cost, false positive rate and F1-Score.

3 Methodology

3.1 Dataset

The dataset contains transactions made by credit cards in September 2013 by European cardholders, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, in which the proportion of the positive class (fraud transactions) is 0.172%.
It contains 30 independent variables. Features V1, V2, ... V28 are the principal components obtained with PCA, and the only features which have not been transformed with PCA are ‘Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset, and it is not included in our analysis because it is just a timestamp, which is unrelated to our analysis. The feature 'Amount' is the transaction Amount, and this feature can be used for example-dependent cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 for fraud transactions and 0 for normal ones. Finally, 29 independent variables and one dependent variable are included in our research.
What we can tell from Figure 1 is that the amounts of fraud class are much lower than the amounts of normal class. And Figure 2 shows the distributions of some sample features, where blue lines represent normal class and orange lines represent fraud class. Some of the features show that two classes are highly overlapped, while the others show the big differences between two classes.

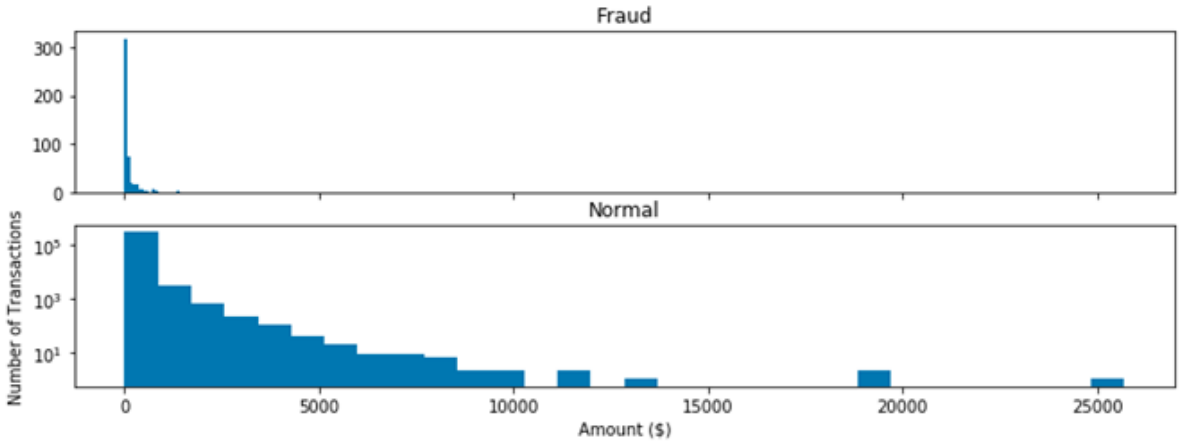

Fig. 1. Histograms of feature ‘Amount’ in class 1 (fraud) and class 0 (normal)

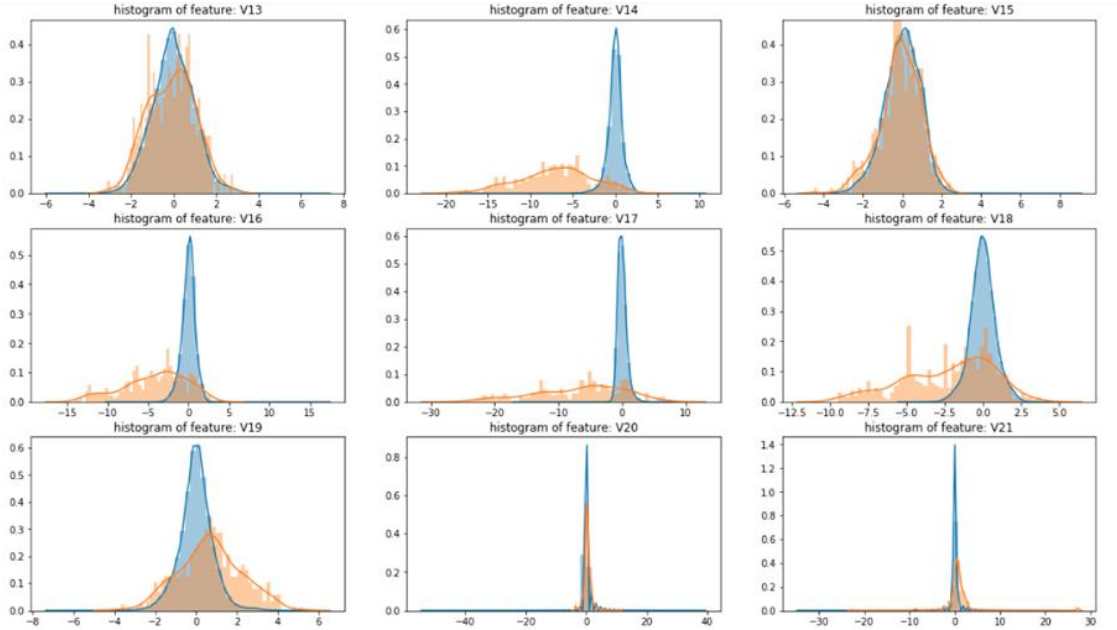

Fig. 2. Distributions of sample features

3.2 Methodology of Ensemble Learning

After splitting the whole dataset into training set and test set, we resample the data via undersampling and oversampling methods. RandomUnderSampler method is applied to undersample the data, while SMOTE is applied to oversample the dataset. Now we have three dataset candidates, undersampling data, oversampling data and the whole dataset. For each dataset, we run logistic regression classifier, decision tree classifier, and random forest classifier and tune the hyper-parameter simultaneously. The outcome scores from three classifiers are completely different. Besides, those classifiers are simple. Consequently, we choose bagging as our ensemble learning method.

To get access to the quality of models, we select recall score as the criterion among all scores from confusion matrix. It makes sense in two aspects. Firstly, recall calculates how many of the TP are recalled or found and it measures how many correct frauds are found in this case. The definition of recall makes more sense for our research goal. Secondly, the number of instances in class 1 is much lower than in class 0. FP and TN are values representing class 0 and too much bias will be taken into account if we include FP and TN. That is why we choose the criterion which takes only TP and FN into consideration and it turns out that recall score is the most suitable one.

TABLE I. Confusion matrix of a binary classification system

		True Class (y_i)	
		Fraud	Legitimate
Predicted Class(p)	Fraud	TP	FP
	Legitimate	FN	TN

3.3 Methodology of Cost Sensitive Learning

The performance of a classifier for a regular two-class classification can be described by the confusion matrix as well as a classifier for a fraud detection can be described by the cost matrix. In fraud detection field, misclassifying a fraudulent transaction as normal carries a higher cost than the inverse case. We would say that a credit card company has certain budget to implement the fraud detection system and each successful detection has a fixed cost. Based on this theory, to perform the cost sensitive learning, two cost matrices Table II and Table III are displayed.

TABLE II
COST MATRIX USING FIXED FN COST

		True Class (yi)	
		Fraud	Normal
Predicted	Fraud (1)	**Ca**	**Ca**
Class (pi)	Normal (0)	**100*Ca**	**0**

PS: Ca is the fix cost of dealing with an alert and we can change the ratio between FN and FP when necessary

TABLE III
COST MATRIX USING REAL FINANCIAL COSTS

		True Class (yi)	
		Fraud	Normal
Predicted	Fraud (1)	**Ca**	**Ca**
Class (pi)	Normal (0)	**Amti**	**0**

PS: Amti is amount of real transaction i

Both tables have same fixed cost -Ca assign for both true positive and false Positive classes. The true positive class represents both predicted and actual labels are "fraud", where the credit card company need a fixed number of cost to deal with this situation. The false positive means the company wrongly identify a normal transaction as fraud. We assume the cost is the same assigned to the true positive class since the company still need apply the standard procedure to evaluate this transaction by contacting the card holder.

Moreover, two approaches are performed to assign the false negative values. The credit card company should be able to calculate what is the exact profit from a typical consumer VS the loss due to fraud. In Table II, the cost of misclassifying a fraud is defined to be a hundred times Ca. The weight of false positive and false negative is vary on different companies and will be updated based on certain circumstance.

However, the cost due to fraudulent transactions range from few to thousands of dollars, which means that assuming constant cost based on Ca might be unrealistic. To optimize the weight, we defined the cost of a false negative to be the real amount Amt_i of the transaction. A false negative class reveals that a transaction is detect as normal but actually is a fraud, where the credit card company would lose the total number spent on the transaction.

Threshold Optimization

Threshold optimization is to modify the probability threshold of an algorithm to minimize the cost due to fraud. To make the model cost sensitive by threshold optimization, we make an optimization in the training dataset to find the new threshold then apply it to the testing set to obtain the results.

Bayes Minimum Risk (BMR)

The BMR is a decision model based on tradeoffs between decisions using probabilities and the relevant cost. In fraud detection, two decisions excited include predict a transaction as fraud p_f or as normal p_n. With real labels y_f and y_n, the risk of predicting a transaction as fraud with given x is defined as $R(p_f|x) = L(p_f|y_f)P(p_f|x) + L(p_f|y_n)P(p_n|x)$ and the risk associated with predicting a transaction as normal with given x is defined as $R(p_n|x) = L(p_n|y_n)P(p_n|x) + L(p_n|y_f)P(p_f|x)$. $P(p_f|x)$ is the probability of a transaction being fraud given x as well as the $P(p_n|x)$ means being normal. $L(p_f|y_f)$ is the loss function when a transaction is predicted as fraud and the real label is fraud too. A transaction is classified as fraud if $R(p_f|x) < R(p_n|x)$ which means if the cost of associated risk is lower than the risk associated with classifying it as normal

Along with two previous cost matrixes, in the case of constant cost (Table II), a transaction will be classified as fraud if:

$$C_a P(p_f|x) + C_a P(p_n|x) <= 100 * C_a(p_f|x)Ps$$

similarly, in the case of real financial cost (Table III), the formula will be

$$C_a P(p_f|x) + C_a P(p_n|x) <= Amt_i * P(p_f|x)$$

4 Result

4.1 Result of Ensemble Learning

It is important to note that the undersampling procedure can only applied to the training set since the test set must react to the real fraud distribution, so the test set for three dataset options are exactly same.

TABLE IV. The proportion of each class in different datasets

Number of Instances	Undersampling Data		Oversampling Data		Whole Dataset	
	Train	Test	Train	Test	Train	Test
Fraud	341	151	199023	151	341	151
Normal	341	85292	199023	85292	199023	85292
Total	682	85443	398046	85443	199364	85443

TABLE V. Optimized hyper-parameters for logistic regression classifier, decision tree classifier, random forest classifier given three dataset options

Recall on class 1	Undersampling Data	Oversampling Data	Whole Dataset
LR	C=1 penalty='l2'	C= 100 penalty='l2'	C= 10 penalty='l1'
DT	criterion='entropy' max_depth=3 min_samples_leaf=1 min_samples_split=4	criterion='gini' max_depth=4 min_samples_leaf=1 min_samples_split=2	criterion='gini' max_depth=4 min_samples_leaf=4 min_samples_split=2
RF	criterion='gini' max_depth=5 min_samples_leaf=9 min_samples_split=2	criterion='gini' max_depth=4 min_samples_leaf=1 min_samples_split=4	criterion='gini' max_depth=4 min_samples_leaf=1 min_samples_split=2

Based on the dataset options shown in Table IV, we tune the hyper parameters and generate the optimized model (Table V) for each option. Then we continue to compare the recall score on class 1 for each option.

The criterion we use to determine the quality of performance is the recall score of class 1 on test set. Table VI can be analyzed in two aspects. When comparing the performances given three dataset options, no matter what classifiers or methods we conduct, undersampling data keeps overperforming oversampling data and the whole dataset. Additionally, we compare the performances of different classifiers and bagging method given the same dataset. Logistic regression classifier performs better than decision tree and random forest on undersampling data and oversampling data. Decision tree performs best on the whole dataset among three classifiers. Also, the outcome scores from three classifiers are completely different and those classifiers are simple. Therefore, we choose bagging as our ensemble learning method.
Last but not least, bagging performs better than any individual classifier.

Table VI. The performances (recall score on class 1 - fraud) from logistic regression classifier, decision tree classifier, random forest classifier and bagging method given three dataset options

Recall on class 1	Undersampling Data		Oversampling Data		Whole Dataset	
	Train	Test	Train	Test	Train	Test
LR	0.93	0.89	0.95	0.88	0.67	0.62
DT	0.87	0.82	0.94	0.88	0.80	0.73
RF	0.91	0.87	0.92	0.85	0.70	0.60
Bagging	0.94	0.91	0.95	0.88	0.94	0.73

4.2 Result of Cost Sensitive Learning

	f1	pre	rec	acc	sav	cost
DT	0.823529	0.854962	0.794326	0.999438	0.775745	3162.0
RF	0.782258	0.906542	0.687943	0.999368	0.672766	4614.0
LR	0.730924	0.842593	0.645390	0.999216	0.630071	5216.0
Bag	0.869231	0.949580	0.801418	0.999602	0.784539	3038.0
DT-TO	0.715976	0.614213	0.858156	0.998876	0.830213	2394.0
RF-TO	0.628571	0.495902	0.858156	0.998326	0.823546	2488.0
LR-TO	0.781690	0.776224	0.787234	0.999274	0.766950	3286.0
Bag-TO	0.475822	0.327128	0.872340	0.996828	0.819007	2552.0
DT-BMR	0.715976	0.614213	0.858156	0.998876	0.830213	2394.0
RF-BMR	0.824742	0.800000	0.851064	0.999403	0.829787	2400.0
LR-BMR	0.734328	0.634021	0.872340	0.998958	0.844823	2188.0
Bag-BMR	0.728916	0.633508	0.858156	0.998947	0.831064	2382.0

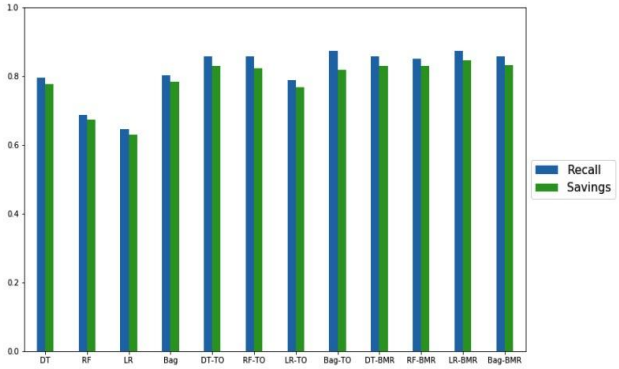

Fig. 3. Results by using fixed cost

	f1	pre	rec	acc	sav	cost
DT	0.819188	0.853846	0.787234	0.999427	0.758856	4855.20
RF	0.773663	0.921569	0.666667	0.999356	0.581940	8417.23
LR	0.730924	0.842593	0.645390	0.999216	0.560315	8852.63
Bag	0.862745	0.964912	0.780142	0.999590	0.749106	5051.50
DT-TO	0.715976	0.614213	0.858156	0.998876	0.821408	3595.78
RF-TO	0.847826	0.866667	0.829787	0.999508	0.827054	3482.11
LR-TO	0.611940	0.471264	0.872340	0.998174	0.805325	3919.59
Bag-TO	0.489066	0.339779	0.872340	0.996992	0.807166	3882.52
DT-BMR	0.627615	0.765306	0.531915	0.998958	0.829117	3440.56
RF-BMR	0.400000	0.323144	0.524823	0.997402	0.815728	3710.13
LR-BMR	0.420455	0.350711	0.524823	0.997612	0.807845	3868.86
Bag-BMR	0.494983	0.468354	0.524823	0.998233	0.822781	3568.13

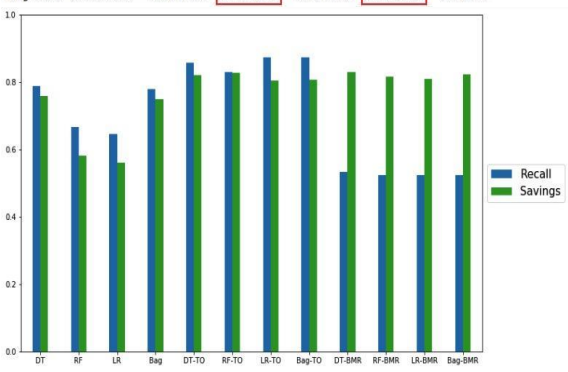

Fig. 4. Results by using real financial cost

First, we perform one ensemble learning bagging (BAG) and three basic models include decision tree (DT), logistic regression (LR) and random forest (RF). All four classifiers were trained using the parameters generated by previous gridsearch. For LR, we set penalty equal to L1 and C equal to 10. For DT and RF models, both have max depth equal to 4 and min_samples_split equal 2 while RF has min_samples_leaf equals to 1 and DT has same parameter which equals to 4.

As introduced before, the thresholding optimization (TO) attempts to make a classifier cost sensitive by changing the probability threshold. After performing TO approach, we got four new models DT-TO, LR-TO, RF-TO and BAG-TO. Similarly, we got four more models include DT-BMR, LR-BMR, RF-BMR and BAG-BMR after performing Bayes Minimum Risk.

Based on Figure 3 and Figure 4, either using constant cost or real financial cost on false negative class, we can observe the increasing saving score after performing TO and BMR. We can conclude that there is no clear winner among TO models and BMR models when evaluating the saving of algorithm. But LR carry a significant increased recall score (0.64 to 0.87) and saving score (0.63 to 0.83) after applying TO with either constant cost or real financial on false negative class. Overall, considering the result, cost and query speed, we recommend using LR-BMR with constant FN cost and LR-TO with real financial cost since ensemble learning didn't overperform other classifiers regards on saving score

5 Discussion

5.1 Discussion of Ensemble Learning

The best performance on the whole dataset comes from decision tree classifier, which equals to 0.73. And after resampling the data, the recall score can reach 0.89 from logistic regression classifier on undersampling data, which improves 22%. If we combine ensemble learning with resampled data together, the performance can even increase by 25%.

5.2 Discussion of Cost Sensitive Learning

It is even more interesting that, with real financial cost, all models lost almost 25% recall after applying BRM. Since recall equals TP/TP+FN, we can conclude FN may increase a lot after performing BRM. Based on our hypothesis, cost will increase when wrongly predict fraud transaction as the normal. However, after applying BRM, all models keep a stable saving scores. Therefore, we believe all BRM models are detecting the most relevant frauds which have higher amount and misclassifying the fraud with lower values.

6 Conclusion and Future Work

In this project, our result shows that data resampling, ensemble learning and cost-sensitive learning improve the performance (recall score of fraud class on test set and savings) compared with doing nothing. Also, the combination of ensemble learning and data resampling or the combination of ensemble learning and cost-sensitive learning is better than using these methods separately.

In future, the following ideas can also be applied to further our research. Firstly, we can compare bagging method with boosting and stacking. Based on the output from three individual classifiers, we use bagging as our ensemble method. But we can also try using the other two methods – boosting and stacking in our future analysis. Secondly, it is necessary to check the sensibility of the fixed cost Ca. In our analysis, we assume that Ca equals to 1, which means 1 dollar. But in the real business world, the fixed costs vary among different companies and different industries. And we can assign other values to Ca based on different situations. Thirdly, conducting under-sampling based on different proportions of fraud class can be useful. The proportion of the fraud class in our dataset is 0.17%. We can choose other percentages, like 1%, 5%, 10%, and then evaluate how different cost-sensitive learning algorithms perform our criterion.

7 Reference

[1] C. Chen, A. Liaw, and L. Breiman. Using random forest to learn unbalanced data. Technical Report 666, Department of Statistics, University of California at Berkeley, 2004. <http://statistics.berkeley.edu/sites/default/files/tech-reports/666.pdf>
[2] Gary M. Weiss, Kate McCarthy, and Bibi Zabar. Cost-Sensitive Learning vs. Sampling: Which is Best for Handling Unbalanced Classes with Unequal Error Costs? Department of

Computer and Information Science, Fordham University.
<https://pdfs.semanticscholar.org/9908/404807bf6b63e05e5345f02bcb23cc739ebd.pdf>
[3] Alejandro Correa Bahnsen, Djamila Aouada and Bj¨orn Ottersten. Example-Dependent Cost-Sensitive Logistic Regression for Credit Scoring. 2014 13th International Conference on Machine Learning and Applications
<http://albahnsen.com/files/Example-Dependent%20Cost-Sensitive%20Logistic%20Regression%20for%20Credit%20Scoring_publish.pdf>
[4] Alejandro Correa Bahnsen, Djamila Aouada and Bj¨orn Ottersten. Cost Sensitive Credit Card Fraud Detection using Bayes Minimum Risk 2013 12th International Conference on Machine Learning and Applications
<http://albahnsen.com/files/Cost%20Sensitive%20Credit%20Card%20Fraud%20Detection%20using%20Bayes%20Minimum%20Risk%20-%20Publish.pdf>
[5] Andrea Dal Pozzolo, Olivier Caelen, Reid A. Johnson, Gianluca Bontempi. Calibrating Probability with Undersampling for Unbalanced Classification 2015 IEEE Symposium Series on Computational Intelligence
<https://www3.nd.edu/~dial/publications/dalpozzolo2015calibrating.pdf>
[6] Alejandro Correa Bahnsen, Aleksandar Stojanovic, Djamila Aouada and Bjorn Ottersten. Improving Credit Card Fraud Detection with Calibrated Probabilities
<http://albahnsen.com/files/%20Improving%20Credit%20Card%20Fraud%20Detection%20by%20using%20Calibrated%20Probabilities%20-%20Publish.pdf>