2017 International Conference on Tools with Artificial Intelligence

Improving Credit Risk Prediction in Online Peer-to-Peer (P2P) Lending Using Imbalanced Learning Techniques

Luis Eduardo Boiko Ferreira,
Jean Paul Barddal, Fabrício Enembreck
Graduate Program in Informatics (PPGIa)
Pontifícia Universidade Católica do Paraná
Curitiba, Brazil
{luiseduardo.boiko, jean.barddal, fabricio}@ppgia.pucpr.br

Heitor Murilo Gomes
Télécom ParisTech
Université Paris-Saclay
LTCI
Paris, France
heitor.gomes@telecom-paristech.fr

***Abstract*—Peer-to-peer (P2P) lending is a global trend of financial markets that allow individuals to obtain and concede loans without having financial institutions as a strong proxy. As many real-world applications, P2P lending presents an imbalanced characteristic, where the number of creditworthy loan requests is much larger than the number of non-creditworthy ones. In this work, we wrangle a real-world P2P lending data set from Lending Club, containing a large amount of data gathered from 2007 up to 2016. We analyze how supervised classification models and techniques to handle class imbalance impact creditworthiness prediction rates. Ensembles, cost-sensitive and sampling methods are combined and evaluated along logistic regression, decision tree, and bayesian learning schemes. Results show that, in average, sampling techniques outperform ensembles and cost-sensitive approaches.**

I. Introduction

Peer-to-peer (P2P) lending is a global trend of financial markets. It allows individuals to obtain and concede loans without having a financial institution as a strong proxy. As a result, lenders usually earn higher return rates compared to conventional savings and investment products offered by banks, while borrowers can obtain credit with lower interest rates, even after the P2P company takes its fee for matching the two parts of the lend and also for providing a credit check of the borrower.

In America, one of the biggest P2P lending online matching companies is Lending Club[1]. As stated on its website, the consumer lending market has surpassed 3 trillion dollars, so there is room for investors to widen their range of investments. In addition to the results provided by P2P lending companies regarding the creditworthiness of potential borrowers, most of the data about loan requests is also provided. These data can be useful for many things, such as tracking how groups of users behave across loans and over time, but more importantly, to generalize their behavior and learn automatic credit scoring computational methods. In this context, machine learning techniques are widely applied to learn creditworthiness models for lending customers and they have shown reasonable success rates. Yet, one of the major limitations of these models is that the ratio between classes is imbalanced, i.e. the number of creditworthy and non-creditworthy customers greatly differs.

To deal with class imbalance, different strategies have been proposed in the machine learning literature, and not many have been applied in the context of P2P lending. These techniques are divided in algorithm adaptation (tuning), ensemble-based, cost-sensitive, and sampling techniques [1]. The goal of this paper is to evaluate representative methods from each of the aforementioned families, thus highlighting the best approaches for P2P lending and pointing out existing gaps on the area.

As contributions of this paper, we cite: (i) the preparation of a real-world P2P lending data set which is also made available for the public, and (ii) a benchmark of several approaches that cope with class imbalance applied to the context of P2P lending default prediction.

This paper is divided as follows. Section II introduces the P2P lending process and the data set used in this study. Section III reviews related work on applying machine learning to P2P lending and the approaches used. The benchmark takes place in Section IV, where a variety of class imbalance techniques are assessed across different learning algorithms. Finally, Section V concludes this paper and presents envisioned future works.

II. P2P Lending

Peer-to-peer (P2P) lending is a trend in the financial market that allows individuals to both obtain or concede loans. In contrast to the traditional lending process, P2P lending allows lenders to obtain higher return rates compared to savings and investments offered by banks, at the expense of having higher uncertainty whether a borrower will default or not.

To maximize the number of successful loans, investors use historical data to learn which characteristics of borrowers make them more susceptible to fully pay (in this work, we assume these to be **creditworthy**) or to be charged off (**non-creditworthy**). In addition to analytics, machine learning is

[1] https://www.lendingclub.com

2375-0197/17/31.00 ©2017 IEEE
DOI 10.1109/ICTAI.2017.00037

also often applied to determine whether a customer is creditworthy automatically, where algorithms are fed with historical data and are able to predict whether a borrower will fulfill its debts with reasonable accuracy.

In this study, we work with a specific data set provided by Lending Club. Lending Club is an online credit marketplace with the goal of facilitating P2P personal and business loans in the USA. By March 31st, 2017, Lending Club has participated of the funding of more than $26B, and the data behind all these loans are available for prospect investors[2]. Lending Club data is provided in two different formats: one on the website, where information about the outcome of loans is known (fully paid or charged off), or via API, which is only available for investors during the time of loan requests analyses. Our study is limited to *charged off* and *fully paid* loans, where the former represent loans that had payments delayed for more than 150 days and the latter are paid up loans. Finally, to avoid leakage, only the data that would be available upon API calls were used in the experiments, and details about the selection of each variable are available on the experiment repository. Below, the steps taken to pre-process the data set are reported and the entire script is made available at https://github.com/jaycwb/resampling-p2p-lending.

A. Data set preparation

To prepare the Lending Club data set for our benchmark, a script has been built to both format and preprocess it. All the data available between 2007 and 2016 was downloaded and merged into a single data set, where only data regarding fully paid and charged off loans were kept since the other possible loan outcomes are inconclusive and might change in the future. Next, several features were removed since they are continuously updated and their values are only available after the loan has been conceded, meaning that they could bias the classifier with privileged information (data leakage). String features were either removed or converted into numeric ones, when possible, and categorical ones were one-hot encoded. Details about the treatment provided to each feature can be found in the repository listed above. Finally, all features and instances with more than 50% of the values missing, and features with variability below 25% were removed. All remaining missing values were then imputed with the mean, for numeric features, or with the mode, for categorical data.

As a result of this preparation step, the final data set contains 578,331 instances (loan requests), such that 461,007 are “fully paid” (creditworthy) and 117,324 “charged off” (non-creditworthy), showing class proportion rates of 79.71% and 20.29% and 133 features plus the class label.

III. Related Work

In this section, we report related works that tackle the problem of applying machine learning in the context of P2P lending. This section also highlights the intersection (or lack thereof) of these works and the explicit handling of imbalanced data sets.

In [2] authors used a variation of Naive Bayes algorithm for multi relational data (MRNB) to classify good and bad borrowers using Prosper[3] data, which is a P2P lending site in U.S. The data used during experimentation are composed of core credit and social network data. In their experiments, authors managed to achieve 70.66% sensitivity and concluded that these good results were due to the MRNB model being able to reasonably combine card data with social network data.

In [3]–[5] authors presented experiments using Lending Club data set. Authors used 3 sets of data obtained from different periods in their analysis. The first set broadened 2007-2013 data, the second set 2012-2014 data, and the third set 2013-2015 data. In [3] they used Logistic Regression (LG), Naive Bayes, Support Vector Machine (SVM) and Random Forest (RF). The LG classifier outperformed the other tested classifiers if one takes into account the precision metric.

In [4] authors proposed an RF based classification method for predicting borrower status. Out of all features available, authors used only 15 of them in experiments and the procedure behind their selection is unclear. Also, it is important to mention that some of the features used, e.g. Revolving Utilization Rate and Revolving to Income Ratio, were deprecated and are no longer in use after December 2014. Using a 5-fold cross-validation procedure, classifiers' parameters were tuned and different metrics were reported. As a result, authors concluded that RF obtained superior results when compared to SVM, Linear Regression, and k-Nearest Neighbors.

Finally, authors in [5] used the Lending Club data set to verify which features were important and determined which individuals were more likely to repay their debts with interest and on time. They used precision and accuracy as measures of performance and concluded that RF is the most appropriate classifier to identify which borrowers would not pay their debts on time, while a single Decision Tree was the best for identifying creditworthy customers.

Despite the effort presented in above-mentioned works, the class imbalance trait of this data set has been nearly neglected. Imbalanced data are characterized by having many more instances belonging to a certain class (majority class) than others (minority classes). Hereafter, let us assume that a binary-labeled data set is denoted by S, the minority class is denoted by $S_{min} \in S$ and the major class as $S_{maj} \in S$, such that $S_{min} \cap S_{maj} = \emptyset$ and $S_{min} \cup S_{maj} = S$.

Since instances belonging to S_{min} rarely occur, the rules for classifying these classes tend to be rare, undiscovered, or ignored [1], [6]. Imbalanced learning problems can be either (i) intrinsic, or (ii) extrinsic. Intrinsic imbalance occurs when the problem is naturally imbalanced. For instance, rare diseases [7], credit card fraud [7], oil spills in the ocean [8], spectral classification of gamma rays [9], P2P lending [2]–[5], among others. In contrast to intrinsic imbalance, extrinsic imbalance occurs when the data is naturally balanced, however, the data acquisition or handling processes cause the imbalance.

[2] https://www.lendingclub.com/public/about-us.action

[3] https://www.prosper.com/

There are multiple alternatives to handle class imbalance. The most recurring approaches are often categorized as: ensembles learners [10], [11], cost-sensitive learning [12], [13] and sampling techniques [14]–[16]. In the following sections, we describe these approaches to imbalanced data sets, which are later used in our analysis in Section IV. It is also important to mention that, to the best of our knowledge, several of the listed techniques have not yet been used in the P2P lending context.

A. Ensemble

The rationale behind ensembles is to induce a set of diverse classification models and then combine their decisions while predicting unlabeled instances. The main motivation to combine classifiers is to improve the overall generalization ability since each classifier is expected to misclassify different instances since each is training with a different and limited amount of data. In this section, we focus on explaining ensemble approaches that are often applied to imbalanced learning. For the sake of brevity, none of these methods will be thoroughly discussed, and thus, readers interested on the topic are referred to the works of [17], [18].

In [19] and [4], authors compared the ensemble-based Random Forest (RF) [20] against some standard classifiers on imbalanced data sets. In the first work, authors used 10 generic imbalanced data sets for the experiments and assessed the impact of parameter tuning. Their recommendation is that the parameters for RF, i.e. number of trees and features, must be adjusted for each data set. In the second work, authors used the Lending Club data broadening January 2012 and September 2014 and their goal was to identify the creditworthiness of a potential loan borrower. Similarly to the study mentioned above, the RF classifier outperformed the others.

B. Sampling

Unlike ensemble and cost-sensitive approaches, sampling techniques are designed to run before learning, it is, during pre-processing. Sampling techniques change the data distribution so that standard algorithms focus on the cases that are more relevant to the user. One of the main advantages of sampling methods it that they can be used with any learning method, and they are divided in over-sampling, under-sampling and hybrid methods [21].

In [22], authors conclude that despite the fact that the RF is an interesting classifier to deal with class imbalance, sampling can further improve its results in the context of Bioinformatics data.

1) Oversampling: Oversampling methods increase the number of instances of S_{min} via the creation of synthetic instances until classes are balanced or nearly balanced. After the creation of new instances, the minority class S_{min}, which was originally underrepresented, may exert a greater influence on learning and on future classifications. Most of the oversampling techniques derive from the seminal *Synthetic Minority Over-sampling Technique* (SMOTE, hereafter denoted as SM for the sake of brevity) [14], and are often implemented as a unique package. In [23], authors observed that in the clinical data sets studied, the oversampling methods outperformed undersampling techniques.

2) Undersampling: The undersampling techniques, on the other hand, aims to reduce the number of instances from the majority class S_{maj} by removing instances from this class. Undersampling techniques often act in two ways, by removing noisy instances, or simply reducing instances from S_{maj} using heuristics or even randomly.

In [24], authors reported that for rare diseases data sets, the use of the RF combined with the Random Undersampling (RU) improves the base classifier area under the precision recall curve (AUPR) with a linear model by 40%.

Both undersampling and oversampling have drawbacks. Undersampling techniques may cause information loss during the sampling process once it is prone to delete relevant instances. On the other hand, oversampling techniques such as SMOTE (SM) may result in class overlaps since it has no guarantees that the synthetic samples will be generated in a region of the feature space that is not a borderline or even a region that is crowded with instances from S_{maj}.

C. Cost-sensitive

While sampling techniques attempt to balance the distribution between classes by considering the representative proportions of the class examples in the distribution, cost-based learning methods consider the costs associated with misclassification examples [25]. Instead of creating balanced distributions of data through different sampling strategies, cost-based learning targets the unbalanced learning problem by using different cost matrices describing the costs for misclassifying instances from S_{maj} and S_{min} [26].

In [27], authors tested different variants of AdaBoost[4] in 4 imbalanced data set from the medical field. In most of the scenarios reported, the cost-approach outperformed the base classifiers tested.

In [28], authors used 14 imbalanced data sets and compared different cost-sensitive approaches against sampling. They concluded that for larger data sets, cost-sensitive learning outperformed the sampling approaches.

IV. Analysis

In this paper, we test different techniques to cope with class label imbalance to improve classification rates in the P2P lending context, specifically the Lending Club data set. To have a comprehensive study, we present results using standard classifiers, various sampling techniques, cost-based methods and ensemble classifiers.

Following the framework proposed in [29], this analysis is organized in two steps: **(i)** an intra-family evaluation to determine the best performing approaches, and **(ii)** an inter-family comparison to find out the fittest approach for this P2P lending data set.

[4] AdaBoost is an ensemble-based method, however, it has been listed in the cost-sensitive learning section since it is built on the notion of misclassification costs.

A. Experimental Protocol

As previously discussed, we analyze mainly three different approaches to handle class imbalance: ensemble-based, cost-sensitive and sampling techniques. Also, we include results using the same classifiers without sampling or costs and these are assumed to be our *baseline* models.

Evaluating imbalanced data sets is not a simple task. The use of traditional metrics in imbalanced domains can lead to sub-optimal classification models and produce misleading conclusions [21]. This occurs since accuracy depends on label distribution. For instance, let us assume a problem where the distribution between classes in a data set is $S_{maj} = 0.8$ and $S_{min} = 0.2$ and we have a classifier that only guesses S_{maj}. In this scenario, we would obtain a reasonably high accuracy rate (80%), yet, the classifier would not correctly classify a single instance from S_{min}. To avoid this type of issue, we proceed with the Area Under the ROC curve (AUROC), specificity and sensibility as measures of classification quality, since they known to be more suitable for imbalanced data sets [21].

We split the data into two stratified data sets: a training set X_{train} and a test set X_{test}, with 70% and 30% of the data, respectively. Our validation process using X_{train} and X_{test} is detailed as follows:

- X_{train}: This data set is used to optimize the parameters of each of the previously mentioned methods. The tuning process performed adopts a 5-fold stratified cross-validation scheme. Tuning was performed to optimize both the parameters for classifiers and sampling techniques. The metric chosen for tuning classifiers is AUROC, since it accounts for the classification rates of both classes. At this point, it is important to highlight that for the sampling experiments, we were unable to tune and train classifiers using the entire X_{train} subset, given the SM complexity of $O(n^2m)$. Therefore, these experiments use 1% and 5% sub-samples of this subset to tune and train classifiers and sampling methods.
- X_{test}: Given the tuned versions of the classifiers and techniques obtained from the training set, these are then used in another 5-fold stratified cross-validation scheme over X_{test}. The results listed in the following sections are the averages obtained during this step.

All the methods used here follow the implementation provided by sklearn[5] and imbalanced-learn[6] Python packages. Friedmans statistic and Nemenyi post hoc tests were used to test for significance of AUROC differences between the classifiers.

B. Baseline classifiers

The classifiers used in this study are Decision Tree, Logistic Regression, and Gaussian Naive Bayes, aiming at using methods based on different assumptions. These algorithms are further discussed throughout the rest of this section and parameters are listed in Table I.

TABLE I
PARAMETERS FOR GRID SEARCH OF BASELINE CLASSIFIERS

Classifier	Parameter	Values
DT	criterion	gini, **entropy**
	splitter	best, **random**
	min_samples_split	**2**, 10, 20
	max_depth	None, 2, 5, **10**
	min_samples_leaf	1, 5, **10**
	max_leaf_nodes	**None**, 5, 10, 20
	class_weight	**balanced**, {0: 1.3}, {0: 1.5}, {0: 1.7}, {0: 2}, {0: 2.3}, {0: 2.5}, {0: 2.7}, {0: 3}, {0: 3.3}, {0: 3.5}, {0: 3.7}, {0: 4}
LG	C	0.001, 0.01, 0.1, 1, 10, 100, **1000**
	solver	**newton-cg**, lbfgs, liblinear, sag
	class_weight	**balanced**, {0: 1.3}, {0: 1.5}, {0: 1.7}, {0: 2}, {0: 2.3}, {0: 2.5}, {0: 2.7}, {0: 3}, {0: 3.3}, {0: 3.5}, {0: 3.7}, {0: 4}
GNB	priors	**None**, [0.1, 0.9], [0.2, 0.8], [0.3, 0.7], [0.4, 0.6], [0.5, 0.5], [0.6, 0.4], [0.7, 0.3], [0.8, 0.2], [0.9, 0.1]

Values listed in bold are the optimal ones. The *class_weight* parameter has been used and optimized only during cost-sensitive experiments.

Decision Tree. A decision tree (DT) uses a hierarchical representation for classification where nodes represent tests over the attributes while leaves represent the classes. It is built usually in two stages: tree building and tree pruning. In the first, it recursively splits the training data set based on a locally optimal criterion until all (or most) of the records belonging to each of the partitions bear the same class label [1]. As trees are prone to overfitting [30], the pruning stage removes branches that provide little classification power and usually allow the tree to generalize better. All the parameters were tuned for each experiment excluding *class_weight*, that was just used at the cost-sensitive experiments.

Logistic Regression. Logistic regression (LG) is a regression model where the dependent variable (class) is categorical [31]. There are variants for binary and multi-class tasks. For the binary task, it creates a linear model based on a sigmoid function that is used to estimate the probability of a binary response (in our case, fully paid or charged off) based on the input features.

Gaussian Naive Bayes. Naive Bayes is a probabilistic classifier that assumes independence between the input features. To cope with numeric features one commonly used approach is to use a Gaussian distribution for estimating the features distributions. The 'tunable' hyper-parameter of GNB is the *priors* probabilities of the classes, which was used only for the cost-sensitive experiments.

C. Ensemble classifiers

Ensemble based methods are often used to cope with imbalanced data sets. In our experiments we have used three ensemble methods: boosting, bagging and random forests. All ensemble-based classifiers adopted a decision tree as the base classifier. There are many ways to use these ensembles for imbalanced learning, one common approach is to combine them with other techniques such as cost-sensitive or sampling

[5] http://scikit-learn.org/stable/

[6] http://contrib.scikit-learn.org/imbalanced-learn/

TABLE II
Parameters for grid search for Ensembles and Sampling

Method	Parameter	Values
AdaBoost	n_estimators	10, 50, **100**
	learning_rate	0.1, **1**, 2
	algorithm	**SAMME**, SAMME.R
Bagging	n_estimators	10, 50, **100**
	max_samples	0.10, 0.25, **0.5**, 0.75, 1.0
	max_features	0.10, 0.25, 0.5, **0.75**, 1.0
RF	n_estimators	10, 50, **100**
	max_features	**None**, auto, sqrt, log2
	max_depth	None, 5, 10, **15**
RU	ratio	0.8, 09, 1.0
SM	kind	regular, borderline1, borderline2
	ratio	0.8, 09, 1.0

Values listed in bold are the optimal ones. For RU and SM the optimal parameters differ for each classifier, so they are shown only in the discussion

methods as in [32] or to rely on their original characteristics [19]. Table II presents the hyper-parameters used for the ensemble methods, which are further discussed as follows.

AdaBoost. The rationale behind boosting is that multiple learners can be combined into a strong one by decreasing both bias and variance. More specifically, learners are chained in a way that each one is trained assigning more importance to instances that were previously misclassified by other learners [1]. By far, the most popular implementation of Boosting is the Adaptative Boosting (AdaBoost) [33]. In imbalanced learning, the goal is to bias the learning towards those misclassified samples, which are often from the minority classes.

Bagging. Bagging (or bootstrap aggregating) [34] is an ensemble method that trains multiple base models on independent subsets of instances. These instances are drawn from the original training set uniformly with replacement. Bagging effectively reduces variance while decreasing the chances of overfitting the model. Predictions are obtained by combining the base models votes using a majority vote approach.

Random Forest. Random Forest (RF) classifier [35] is an ensemble of unpruned classification trees, induced from bootstraps of the training data. In contrast to conventional decision trees, during the branching process, only a randomly selected subset of features is evaluated [36]. As most classifiers, RF is constructed to minimize the overall error rate. It means this method is likely to be biased towards the majority class rather than to the minority one.

D. Sampling techniques

SMOTE. SMOTE (SM) technique generate synthetic samples based in the observations from S_{min}. SMOTE is designed to find the K-nearest neighbors ($K \in \mathbb{N}$) for each $x_i \in S_{min}$. Then, one of those neighbors is randomly selected ($\hat{x}_i$) and its distance from x_i is multiplied by a random number $\delta \in [0, 1]$, which results in a new vector (x_{new}), that is located between x_i and the selected neighbor.

In order to solve some problems introduced by regular SM the SMOTE Borderline1 (SMBL1) and SMOTE Borderline2 (SMBL2) methods were proposed in [15]. While the regular SM acts only S_{min}, these variants also take into account the S_{maj} samples. The SMBL1 act by finding samples from S_{min} that have more S_{maj} neighbors than S_{min} (risky samples) and then generate the new synthetic samples near this instances, in order to provide more neighbors bearing the same class label. The SMBL2 follows the same steps as SMBL1, but it does not generate instances based on risky samples from S_{min} while taking into account instances from S_{maj} to determine the borderline instead.

TABLE III
Results Baseline Classifiers

Classifier	Sensibility	Specificity	AUROC	Train Inst.	Test Inst.
DT	**0.98**	0.08	0.53	404830	173501
GNB	0.72	**0.45**	**0.59**	404830	173501
LG	**0.98**	0.07	0.53	404830	173501

Random Undersampling. One of the most straightforward techniques for removing instances is Random Undersampling (RU), which remove random instances from the S_{maj} to the desired level. This technique does not have parameters for tuning but the desired ratio of balance.

The values used in our experiments for sampling methods are available at Table II.

E. Discussion

Observing the results on the Table III, some conclusions can be drawn about the baseline classifiers. Despite the fact that GNB has no tuning parameters in these initial experiments, it scored the highest AUROC and specificity. Thus, it is important to highlight that despite the lower results obtained by GNB in sensitivity, this classifier should still be considered the best performing one since it accounts for both classes with reasonable accuracy, as depicted by AUROC.

One of the factors that may have contributed to DT's poor performance is that no pruning rule was used. One evidence that the DT has overfitted is that the optimized parameter encountered for *max_depth* was 10, *max_leaf_nodes* was *None* and the *min_samples_split* to 2, which means that the DT grew 10 levels at the vertical, there was no limit at the horizontal growing according to *max_leaf_nodes* and it created new nodes even if there are only two samples arriving at this node. It means it lost it generality capability. The LG also sheds evidence of overfitting, since the parameter *C* was tuned to 1000, the highest tested value, which means it has weaker regularization. In the end, no statistical difference between the tested classifiers at the baseline experiments.

As in the results of the baseline classifiers, in the cost-sensitive experiments, the GNB was the most stable classifier, as seen in Table IV. The optimized parameters for the *priors* parameter of the GNB was 0.6 for S_{min} and 0.4 for S_{maj}, while for DT and LG the optimized *class_weight* gives a weight 4 to S_{min} samples and 1 for S_{maj}. In the baseline LG, the optimized parameter *C* was 1000, in this experiment the optimized *C* was 100, which means that the classifier gained generality. All other parameters were the same than

TABLE IV
RESULTS COST-SENSITIVE CLASSIFIERS

Classifier	Sensibility	Specificity	AUROC	Train Inst.	Test Inst.
DT	**0.99**	0.01	0.50	404830	173501
GNB	0.49	**0.69**	**0.59**	404830	173501
LG	**0.99**	0.02	0.51	404830	173501

TABLE V
RESULTS FOR ENSEMBLES

Classifier	Sensibility	Specificity	AUROC	Train Inst.	Test Inst.
AdaBoost	0.98	0.08	0.53	404830	173501
Bagging	0.98	0.09	0.53	404830	173501
RF	0.98	**0.10**	**0.54**	404830	173501

TABLE VI
RESULTS FOR SAMPLING 1%

Classifier	Sensibility	Specificity	AUROC	Train Inst.	Test Inst.
DT+RU	0.65	0.63	0.64	4050	173501
GNB+RU	0.61	0.59	0.60	4050	173501
LG+RU	0.75	0.53	0.64	4050	173501
DT+SM-BL2	**0.94**	0.18	0.56	4050	173501
GNB+SM-BL2	0.36	**0.79**	0.57	4050	173501
LG+SM-BL2	0.64	0.66	**0.65**	4050	173501

TABLE VII
RESULTS FOR SAMPLING 5%

Classifier	Sensibility	Specificity	AUROC	Train Inst.	Test Inst.
DT+RU	0.62	0.66	0.64	20240	173501
GNB+RU	0.57	0.63	0.60	20240	173501
LG+RU	0.64	0.67	**0.66**	20240	173501
DT+SM	**0.97**	0.08	0.53	20240	173501
GNB+SM	0.37	**0.79**	0.58	20240	173501
LG+SM	0.69	0.62	0.65	20240	173501

the baseline experiment. In the other hand, the DT parameters changed most. The parameters *criterion*, *min_samples_leaf* and *min_samples_split* were tuned to *gini*, 1 and 20, while at the baseline they were *entropy*, 10 and 2, respectively.

Both LG and DT deteriorate its performance at S_{min} with this approach, while GNB decreases its performance at S_{maj}. Despite the loss of performance at S_{maj}, GNB increase its performance at S_{min}, keeping the same AUROC as the baseline. As as the baseline experiment, the GNB stills the most suitable classifier at cost-sensitive experiments and there is no statistical difference between the tested classifiers.

The results in Table V show that none of tested ensemble-based classifiers outperformed the baseline GNB for both AUROC and Specificity. Following the results of [4], [5], the best-performing method was RF. The optimal parameters for *n_estimators* for all tested ensembles was 100, which was the highest tested value. Despite the RF being the best ensemble, there is no statistic difference between the tested ensembles.

As mentioned earlier, due to the prohibitive computational requirements of sampling techniques, only a subset of our training set has been used. Two stratified variants have been tested, the first using 1% of the available training data, and another time with 5%.[7] For the sampling techniques, increasing the number of training instances from 1% to 5% did not show significant increases, as can be seen at Tables VI and VII. Furthermore, for DT classifier with oversampling, we observed unexpected results, where bigger training sets resulted in smaller AUROC rates.

Despite that, when comparing the results of classifiers with and without sampling techniques, our baselines were beaten in all scenarios in terms of specificity and AUROC. The best AUROC rate was obtained with LG+RU, using 5% of X_{train}, which is quite similar to the one obtained by LG+SMBL2 using 1% of the training set. Also, LG is statistically better than the other tested classifiers. The fact that the best result was obtained with RU evidentiate that information loss is not a problem with a data set of this size (20240 instances), however, for oversampling techniques, training with 1% of X_{train} result in many instances being generated at risky or borderline areas of the features space. This becomes evident when one observes that SMBL2 was the best oversampling technique for all tested classifiers when using 1% of X_{train}, which is mainly due to its cleaning steps to handle both risky and borderline instances. While SMBL2 is better with 1%, regular SM is better with 5% of X_{train}, giving evidence that the borderline samples are mostly observed because of lack of data.

With 1% of X_{train} the optimal ratios for RU were 0.8, 0.9 and 1.0 for LG, GNB and DT respectively. The optimal *kind* for SM was *borderline2*, while the ratios were 0.9 for GNB and DT and 1.0 for LG. When training with 5% of X_{train} the optimal RU ratio for every tested classifier was 1.0, while for SM, the optimal *kind* was *regular* and the ratios were 0.9 for DT and LG, while for GNB it was 0.8.

V. CONCLUSION

Identifying whether a borrower will default or not is vital for a P2P lending platform. This work presented a comprehensive and systematic experimental analysis of the most recurring approaches to handle the class imbalance problem. We illustrated our experiments using the Lending Club data set. The data preparation was thoroughly detailed and ensured that no leakage (e.g. future information) occurred, thus rendering a sustainable data set for future analyses. The classifiers and techniques to handle imbalanced were discussed and the models tuned using grid search, a process which was also clearly discussed and reported in our experiments section.

Despite the fact that RF has been reported as being suitable for predicting borrower‘s status with Lending‘s Club data, our experiments show that neither RF nor any other method is capable of overcoming GNB’s stability. GNB was also the classifier which performs best with the cost-sensitive approach, improving its correct classification of S_{min} samples, while at the expense of the classification rates of S_{maj} samples. The cost-sensitive approach was ineffective for DT and LG classifiers.

To the best of our knowledge, there has been no previous study that detailed the data set preparation and empirically tested ensembles, cost-sensitive approaches and sampling techniques using the Lending Club available data. With that said,

[7] It is important to highlight that bigger samples were tested, however, the experiments ended up taking too long or causing system crashes.

this work could serve as a guide for future researches. As future work, we intend to:

- Improve the pre-processing steps of our approach, so it encompasses more solid dimensionality reduction techniques, such as Principal Component Analysis;
- Test hybrid approaches to deal with class imbalance;
- Extend the gamut of tested classifiers to encompass deep learning approaches;
- Consider this data set as a data stream, due to the data set's temporal characteristics.

Acknowledgment

This research was financially supported by the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES) through the *Programa de Suporte à Pós-Graduação de Instituições de Ensino Particulares* (PROSUP) program.

References

[1] Y. Sun, A. K. C. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 4, pp. 687–719, 2009. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0218001409007326

[2] R. Vedala and B. R. Kumar, "An application of naive bayes classification for credit scoring in e-lending platform," in *Data Science & Engineering (ICDSE), 2012 International Conference on*. IEEE, 2012, pp. 81–84.

[3] K. Tsai, S. Ramiah, and S. Singh, "Peer lending risk predictor," 2014.

[4] M. Malekipirbazari and V. Aksakalli, "Risk assessment in social lending via random forests," *Expert Systems with Applications*, vol. 42, no. 10, pp. 4621–4631, 2015.

[5] V. Kumar, S. Natarajan, S. Keerthana, K. Chinmayi, and N. Lakshmi, "Credit risk analysis in peer-to-peer lending system," in *Knowledge Engineering and Applications (ICKEA), IEEE International Conference on*. IEEE, 2016, pp. 193–196.

[6] J. Gong and H. Kim, "Rhsboost: Improving classification performance in imbalance data," *Computational Statistics & Data Analysis*, vol. 111, pp. 1–13, 2017.

[7] N. H. Vo and Y. Won, "Classification of unbalanced medical data with weighted regularized least squares," in *Frontiers in the Convergence of Bioscience and Information Technologies, 2007. FBIT 2007*. IEEE, 2007, pp. 347–352.

[8] K. N. Topouzelis, "Oil spill detection by sar images: dark formation detection, feature extraction and classification algorithms," *Sensors*, vol. 8, no. 10, pp. 6642–6659, 2008.

[9] C. Bellinger, N. Japkowicz, and C. Drummond, "Synthetic oversampling for advanced radioactive threat detection," in *Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on*. IEEE, 2015, pp. 948–953.

[10] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463–484, 2012.

[11] S. Wang, L. L. Minku, and X. Yao, "A multi-objective ensemble method for online class imbalance learning," in *Neural Networks (IJCNN), 2014 International Joint Conference on*. IEEE, 2014, pp. 3311–3318.

[12] A. Ghazikhani, R. Monsefi, and H. S. Yazdi, "Online cost-sensitive neural network classifiers for non-stationary and imbalanced data streams," *Neural Computing and Applications*, vol. 23, no. 5, pp. 1283–1295, 2013.

[13] C. Zhang, K. C. Tan, and R. Ren, "Training cost-sensitive deep belief networks on imbalance data problems," in *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016, pp. 4362–4367.

[14] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[15] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: a new over-sampling method in imbalanced data sets learning," in *International Conference on Intelligent Computing*. Springer, 2005, pp. 878–887.

[16] J. M. Choi, "A selective sampling method for imbalanced data learning on support vector machines," 2010.

[17] L. Rokach, "Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography," *Computational Statistics & Data Analysis*, vol. 53, no. 12, pp. 4046–4072, 2009.

[18] H. M. Gomes, J. P. Barddal, F. Enembreck, and A. Bifet, "A survey on ensemble learning for data stream classification," *ACM Comput. Surv.*, vol. 50, no. 2, pp. 23:1–23:36, Mar. 2017. [Online]. Available: http://doi.acm.org/10.1145/3054925

[19] T. M. Khoshgoftaar, M. Golawala, and J. Van Hulse, "An empirical study of learning from imbalanced data using random forest," in *Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on*, vol. 2. IEEE, 2007, pp. 310–317.

[20] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[21] P. Branco, I. S. Torgo, and R. P. Ribeiro, "A Survey of Predictive Modeling on Imbalanced Domains," vol. 49, no. 2, pp. 1–50, 2016.

[22] D. J. Dittman, T. M. Khoshgoftaar, and A. Napolitano, "Is data sampling required when using random forest for classification on imbalanced bioinformatics data?" in *Theoretical Information Reuse and Integration*. Springer, 2016, pp. 157–171.

[23] A. Kasem, A. A. Ghaibeh, and H. Moriguchi, "Empirical study of sampling methods for classification in imbalanced clinical datasets," in *International Conference on Computational Intelligence in Information System*. Springer, 2016, pp. 152–162.

[24] D. Dai and S. Hua, "Random under-sampling ensemble methods for highly imbalanced rare disease classification," in *Proceedings of the International Conference on Data Mining (DMIN)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2016, p. 54.

[25] C. Elkan, "The foundations of cost-sensitive learning," in *International joint conference on artificial intelligence*, vol. 17, no. 1. Lawrence Erlbaum Associates Ltd, 2001, pp. 973–978.

[26] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[27] Y. Sun, M. S. Kamel, A. K. Wong, and Y. Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognition*, vol. 40, no. 12, pp. 3358–3378, 2007.

[28] K. McCarthy, B. Zabar, and G. Weiss, "Does cost-sensitive learning beat sampling for classifying rare classes?" in *Proceedings of the 1st International Workshop on Utility-based Data Mining*, ser. UBDM '05. New York, NY, USA: ACM, 2005, pp. 69–77. [Online]. Available: http://doi.acm.org/10.1145/1089827.1089836

[29] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 20–29, Jun. 2004. [Online]. Available: http://doi.acm.org/10.1145/1007730.1007735

[30] M. Bramer, "Avoiding overfitting of decision trees," in *Principles of Data Mining*. Springer, 2016, pp. 121–136.

[31] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 215–242, 1958.

[32] B. Yuan and X. Ma, "Sampling+ reweighting: boosting the performance of adaboost on imbalanced datasets," in *Neural Networks (IJCNN), The 2012 International Joint Conference on*. IEEE, 2012, pp. 1–6.

[33] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *European conference on computational learning theory*. Springer, 1995, pp. 23–37.

[34] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[35] B. L. Random, "Random forests," *Mach. Learn*, vol. 45, pp. 5–32, 2001.

[36] C. Chen, A. Liaw, and L. Breiman, "Using random forest to learn imbalanced data," *University of California, Berkeley*, vol. 110, 2004.