Resampling and Ensemble Strategies for Churn Prediction

Araştırma Makalesi/Research Article

Serra Çelik[1]*, Seda Tolun Tayalı[2]

[1]Department of Informatics, Istanbul University, Istanbul, Türkiye
[2]Department of Quantitative Methods, School of Business, Istanbul University, Istanbul, Türkiye

serra.celik@istanbul.edu.tr, stolun@istanbul.edu.tr

(Geliş/Received:16.06.2023; Kabul/Accepted:28.08.2023)
DOI: 10.17671/gazibtd.1314870

Abstract— Churn analysis is a customer relationship management analytics that companies implement to predict the customers who are likely to terminate doing business with them. The success of marketing efforts to retain the existing customers is possible only if probable churners are correctly specified beforehand. Therefore, having powerful models with high prediction capabilities that lead to a profit growth is crucial. The imbalanced nature of churn datasets negatively effects the classification performance of machine learning methods. This study examines resampling –over- and under-sampling- and ensemble learning –bagging, boosting, and stacking– strategies integrated with the cross-validation procedure on imbalanced churn prediction. The experimental results, which are compared to the results of Support Vector Machines taken as the benchmark, show that ensemble methods improve the prediction performances. Also, applying over-sampling achieves a noticeable performance in comparison with the under-sampling approach.

Keywords— churn prediction; class imbalance; resampling; support vector machines; evaluation metrics; ensemble learning

Müşteri Kaybı Tahmini için Yeniden Örnekleme ve Topluluk Yöntemleri

Özet— Müşteri kayıp analizi; şirketlerin, kendileriyle çalışmayı sonlandırması muhtemel müşterileri tahmin etmek için kullandığı bir müşteri ilişkileri yönetimi analitiğidir. Mevcut müşterileri elde tutmaya yönelik pazarlama çalışmalarının başarısı, ancak olası müşteri kayıplarının önceden doğru bir şekilde belirlenmesiyle mümkündür. Bu nedenle kâr artışına yol açacak, yüksek tahmin kabiliyetli, güçlü modellere sahip olmak çok önemlidir. Kayıp analizi için kullanılan veri kümelerinin dengesiz doğası, makine öğrenimi yöntemlerinin sınıflandırma performansını olumsuz etkilemektedir. Bu çalışma, dengesiz kayıp tahmini üzerinde çapraz doğrulamanın prosedürüyle entegre edilmiş yeniden örnekleme - aşırı ve az örnekleme - ve topluluk öğrenme - bagging, boosting, ve stacking - stratejilerini incelemektedir. Referans noktası olarak alınan Destek Vektör Makinelerinin sonuçlarıyla karşılaştırılan deneysel sonuçlar, topluluk yöntemlerinin tahmin performanslarını iyileştirdiğini göstermektedir. Ayrıca aşırı örneklemenin uygulanması, az örnekleme yaklaşımına kıyasla fark edilebilir bir performans artışı sağlamıştır.

Anahtar Kelimeler— kayıp tahmini, sınıf dengesizliği, yeniden örnekleme, destek vektör makineleri, değerlendirme metrikleri, topluluk öğrenme

1. INTRODUCTION

Building and maintaining successful relationships with customers is an inevitable necessity to survive in today's competitive and demanding markets. The purpose of customer relationship management (CRM) is to understand customer behaviors and develop a sustainable communication with them through a personalized approach to prevent customer churn [1]. Therefore, the analytical aspect of CRM, which focuses on analyzing customer and market data, hugely benefits from data analytics by deploying the obtained analysis results to have an increased effectiveness of marketing efforts.

Customer churn, also known as customer attrition, is defined as the likelihood that a customer terminates doing business with a company [2]. It is a financially expensive problem for businesses since gaining new customers is known to be much more expensive than keeping the current ones with regards to the cost of marketing efforts. Also, the existing customers are more open to communication and spend more than the new ones. As a result, "increasing customer retention rates by 5% increases profits by 25% to 95%" [3].

Telecommunication sector is a prevalent domain for customer churn analysis. Customers switching between operators, who are by definition churners, are quite common in the sector. The annual churn rate is around 30% in average and acquiring new customers is at least 5 times more expensive than keeping the existing ones [4]. Therefore, losing a high number of customers results in high losses for the telecom companies because of the lost acquisitions as well as of certain CRM efforts such as reducing the prices to keep the highly potential churners in the company portfolio. This makes the minimization of the churn rate crucial for the telecom companies, and consequently successful churn analysis becomes an important tool.

Churn analysis is handled by data analytics approaches, especially by using predictive analytics via machine learning algorithms. However, the problem –as in other CRM cases such as fraud detection, response modeling, and credit evaluation– inherits imbalance data classes which is a factor that negatively effects the classification performance of models and turns customer classification into a more challenging task.

This study tackles the class imbalance problem both with a data-level approach of resampling and an algorithm-level approach of ensemble learning. The prediction performances of the applied methods are evaluated on an imbalanced churn dataset in the telecom sector. The study differentiates from the existing literature by providing a more comprehensive comparison of techniques as well as their hybrids. To the authors' knowledge, there is no prior research on churn prediction that provides the hybridization of the stacking ensemble with both under- and over-sampling techniques. The other point that is missing in the churn prediction literature is how to apply resampling techniques with cross-validation. Studies in the domain either do not mention about this at all or state that they are applying cross-validation after resampling, which may cause achieving overly optimistic results. This study implements the correct integration of resampling and cross-validation to its empirical design. The prediction performances of the applied methods are compared with the benchmark results of Support Vector Machines (SVM) that is known to have high performance in binary classification problems.

Comparative analyses are carried out in order to answer the following questions within the scope of the research context:

Research Question 1 (RQ1): Do resampling methods affect the prediction performance of SVM?
Research Question 2 (RQ2): Do ensemble strategies increase the prediction performance for imbalanced telecom churn problem? Which ensemble yields the best results?
Research Question 3 (RQ3): Does the combination of resampling and ensemble strategies increase the prediction performance for the imbalanced telecom churn problem?
Research Question 4 (RQ4): Do the selected performance metrics give compatible results? Are they all appropriate measures for evaluating the imbalanced customer churn prediction?

The remainder of this paper is as follows: Section 2 gives a summary of prior research on imbalanced datasets in a churn setting. Section 3 explains the methods used and Section 4 explains the experimental framework of the study. Section 5 presents and discusses the empirical findings and Section 6 concludes with remarks and future research directions.

2. PREVIOUS WORK ON IMBALANCED CHURN PROBLEM

There is a substantial discrepancy in the sample size of each target class in a typical churn dataset, which is known as the class or data imbalance problem. Kwon and Sim [5] specify the class imbalance as one of the data set characteristics that has a negative effect on the performance of classification algorithms. There are two main approaches –data-level (external) and algorithm-level (internal) – to handle this problem. The algorithm-level approach involves modifications of the existing classifiers to favor the learning from the minority class, whereas the data-level approach is independent of classifiers and resizes the training data to decrease the imbalance ratio for to diminish the effects caused by the skewed class distribution [6].

Resampling is a pre-processing technique suggested as a data-level approach for obtaining more balanced classes. Qureshi et al. [7] use both random under- and over-sampling and keep the imbalance ratio to a certain level prior to applying classification algorithms. Amin et al. [8] focus on over-sampling techniques and compare their effects on the classification performance of algorithms based on the rough set theory. There are studies [9], [10] that implement more sophisticated sampling methods as well as studies [11], [12] that investigate the combination of over-sampling and under-sampling techniques to compensate the drawbacks of each technique. Li et al. [13] propose one-sided sampling to balance massive churn datasets once the dataset is split into clusters by the k-means algorithm. Two cluster-based under-sampling methods are applied in [14] with Support Vector Machines and their performance for a telecom churn dataset is found adequate. Verbeke et al. [15] examine the effect of over-sampling on the performance of a telecom customer churn prediction model and conclude that the dataset structure and the classification technique can completely change the results as Haixiang et al. [16] also emphasize.

The literature on churn prediction is more focused on the algorithm-level approach, which includes the modifications of traditional classifiers that are especially developed for learning from imbalanced datasets [17], [18]. Several studies on the domain try to understand the algorithm-level effect of methods such as one-class learning [19], and cost-sensitive learning [20].

Ensemble classifiers are solution methods that can be sub-categorized under the algorithm-level approach [21], [22]. Bagging and random forests are the most popular ensembles used in churn prediction. The literature on imbalanced churn prediction that use the ensembles favors random forests as well as its modified versions [23] and states that the ensemble improves the prediction accuracy. Boosting algorithms found applications in churn prediction [24], [25]. There are studies that propose new solution approaches combining ensemble methods with cost-sensitive learning [26] and transfer learning [27] for imbalanced churn prediction and declare the results as prominent.

With the ensemble methods showing their strength in improving the classification performances, researchers also investigated the combinations of ensemble learning methods –especially bagging and random forests– with resampling methods to tackle the imbalance churn problem. The findings in [28], [29] show that random over-sampling combined with random forests yields better results than resampling with random under-sampling and SMOTE. On the other hand, Zhu et al. [30] state that bagging and random forests achieve the most promising results with respect to the profit-based measure when there is no resampling involved. The authors in [31] claim that the combination of simple under-sampling and SMOTE with cost-sensitive version of random forests give better performance than random forests.

Burez and Van del Poel [32] compared the performance of under-sampling, gradient boosting machine, and weighted random forests, whereas Liu et al. [33] made a similar comparison between under-sampling, weighted random forests, and RUSBoost. Both studies concluded that under-sampling performs better in terms of accuracy. The literature on investigating the effects of stacking as an ensemble in churn prediction is scarce. Ahmed et al. [34] suggest to use hybrid models of boosted-stacked and bagged-stacked classification. Authors investigate the optimal number of base learners in a stacking ensemble through implementing all combinations of selected classifiers in stacking and comparing the performances. Amin et al. [35] follow a just-in-time perspective and apply stacking, using SVM as the base classifier, on the training set of a cross-company and test it on the company dataset.

This study examines the main strategies of ensemble learning –bagging, boosting, and stacking–, resampling methods and their combinations to see whether they affect the performance of imbalanced churn prediction. While testing the performances of these methods, the methodological framework of using resampling techniques with cross-validation is another focal point.

3.IMBALANCED CHURN DATA CLASSIFICATION

Classification is a supervised learning task, and customer churn analysis can be modeled as a binary classification problem. Let's define a training dataset as;

$$L = \{(x_i, y_i), i = 1,2, \dots, n\}. \tag{1}$$

where n refers to the number of customers and the vector $x_i \in \mathbb{R}^d$ represents the values the ith customer takes with respect to the attributes denoting the characteristics of customers. y_i is the target vector and $y_i \in \{-1,1\}$ for a binary classification case, where -1 refers to a non-churner and 1 refers to a churner. The general assumption is that the data are independent and identically distributed (iid) realizations of a sample randomly drawn from a population (X, Y). Therefore, the objective is to build a learning model, a function $\hat{f}: x \mapsto \hat{f}(x) \in \{-1,1\}$, and use this model to predict the class label for the previously unseen (test) data.

Class imbalance is a skewed distribution problem of classes inherent in churn datasets. Let p be the minority class referring to churners and q be the majority class referring to non-churners. $p = \{y_1, y_2, \dots, y_P\}$ and $q = \{y_1, y_2, \dots, y_Q\}$, where P and Q refer to the number of samples in the minority and majority class, respectively. The imbalance ratio (IR), $IR = Q/P$, is greater than 1 in telecom churn datasets since the number of non-churners are more than churners [13]. The imbalanced structure of a dataset either distorts the performance of classification or causes overfitting and gives fallacious high accuracies. Hence, the skewly distributed target values turn classification into a challenging task. This study focuses on resampling methods as a data-level strategy, and ensemble learning methods as an algorithm-level strategy to tackle the imbalance problem.

3.1. Resampling methods

A dataset with balanced classes has a better chance of being classified accurately and outputting a high prediction rate without facing an overfitting issue. This study pursues resampling as a data-level approach to tackle the class imbalance problem and examines the effects of the following techniques.

Random Over-Sampling (ROS): ROS balances the representation of classes in the training set by randomly duplicating the observations in p. The regenerations in ROS can cause overfitting. [36]

Synthetic Minority Over-Sampling Technique (SMOTE): SMOTE searches k nearest neighbors for each observation x_i in class p and generates synthetic samples based on the linear interpolations between each x_i in p and their selected nearest neighbors[37].

$$x_{synthetic} = x_i + (x' - x_i) * \delta \qquad (2)$$

where x' is the randomly selected neighbor, and $\delta \in [0,1]$ is a random number. The parameter k that is based on the over-sampling size, determines the number of samples to be generated for a minority sample.

Random under-sampling (RUS): RUS randomly eliminates the observations in q so that the representation of classes in the training set is balanced.

Clustering Based Under-Sampling (CLUSBUS): After splitting the dataset into training and test sets in accordance with the distribution of classes, the training set is divided into homogenous groups via clustering. Each cluster has data that belongs to both the majority and the minority class. The number of samples from the majority class ($SSize_{MA}^{i}$) are randomly selected from each cluster based on (1) and combined with the minority class units. Thus, a new training set is constructed [38].

$$SSize_{MA}^{i} = (m \times Size_{MI}) \times \frac{Size_{MA}^{i}/Size_{MI}^{i}}{\sum_{i=1}^{k} Size_{MA}^{i}/Size_{MI}^{i}} \qquad (3)$$

$Size_{MI}^{i}$ refers to the sample size of the minority class in the ith cluster. m stands for the ratio of majority class ($Size_{MA}$) over minority class ($Size_{MI}$) in the training set and $m \geq 1$. This study uses Partitioning Around Medoids (PAM) as the clustering algorithm because of the mixed structure of the dataset attributes.

3.2. Ensemble Learning

Ensemble learning is a combination of base or weak supervised learning algorithms to form a stronger classification. This study investigates the three main ensemble learning –bagging, boosting, stacking– in churn classification along with their combinations with resampling methods.

Let $\hat{\varphi}_n(x)$ be a predictor from a classification model fitted to the learning random sample defined in (1).

3.2.1. Bagging

Bagging starts with drawing bootstrap samples $\{L^{(b)}\}$, where $b = 1, 2, \dots, B$ –subsets of the same size (n) as $\mathcal{L}$ are randomly drawn with replacement– from L. The next step is to form a learning model $\varphi_n(x) = \{\varphi(x, L^{(b)})\}$ for each $\{L^{(b)}\}$ that predicts their class labels. The bagging predictor $\hat{\varphi}_{n,B}(x)$ is then formed by aggregating the results of all $\varphi_n(x)$'s as in (4) [39].

$$\hat{\varphi}_{n,B}(x) = \text{MajorityVote}\{\varphi_n(x)\}_{b=1}^{B} \qquad (4)$$

Bagging is a variance reduction technique for unstable machine learning procedures that are highly variant such as the tree-based algorithms due to their sensitivity to the training data. This study uses bagged CART and random forests that are both decision-tree based procedures.

Bagged CART: Bagged CART follows the bagging procedure by selecting bootstrap samples from L and creating a learning model using CART that uses a greedy algorithm to choose the feature to split on [40].

Random Forests (RF): A random forest extends the idea of bagging by realizing the splits through randomly selecting a subsample of features with replacement and choose the best split from among those features instead of choosing the best split among all predictors as is the case in bagging.

3.2.2. Boosting

The idea behind boosting is to use weak learners, whose performance is at least slightly better than random chance, several times to get a stronger learner [41]. As opposed to bagging, classifiers are created sequentially (iteratively) in boosting [42]. It starts by assigning equal weights to all examples $\alpha_i^1 = 1/n$. At each iteration b, with $b \in \{1,2, \dots, B\}$ the weak classifier $\hat{\varphi}_{n,b+1}(x)$ updates the weights $\{\alpha_{b+1}\}_{i=1}^{n}$ from $\{\alpha_b\}_{i=1}^{n}$ by giving more concentration to the examples that are unfitted by the previous classifier $\hat{\varphi}_{n,b}(x)$. It assigns more weights to the erroneous classifications while decreasing the weights of those that are correctly classified by $\hat{\varphi}_{n,b}(x)$.

The predictions of the binary classification are then a weighted linear combination of the selected binary classifiers, $\varphi_1(x), \dots, \varphi_B(x)$;

$$\hat{\varphi}_{n,B}(x) = sign(\textstyle\sum_{b=1}^{B} \alpha_b \varphi_b(x)) \qquad (5)$$

where $\alpha \in \mathbb{R}: (0, 1]$ refers to the assigned weights. The sign function is 1 when the argument is non-negative, and -1 otherwise.

C5.0: The algorithm is a tree-based classifier and the successor of C4.5 [43] with the boosting property. The C5.0 classifier splits on the feature that has the maximum information gain based on the entropy measure calculated as $-\sum_{i=1}^{n} p_i \log p_i$, where p_i denotes the probability of a given class as the outcome for each of the classes for y. The information gain of attribute A on L is defined as the difference between the empirical entropy of set L and the empirical conditional entropy of L under the given condition A: $IG(L, A) = \hat{\varphi}(L) - \hat{\varphi}(L|A)$. The algorithm assigns considerable weights to the misclassified instances at each iteration, while decreasing the assigned weights of correctly classified instances in a slower rate. The final prediction is a simple average of class probabilities generated from each tree.

Stochastic Gradient Boosting (SGB): SGB is a regularized boosting algorithm through the learning rate $v \in [0,1]$, in addition to preserving the advantages of bagging. At each iteration, a decision tree as the base learner is built using the random subset $\{x_{\pi(i)}, y_{\pi(i)}\}_i^{\tilde{n}}$, where $\{\pi(i)\}_i^{\tilde{n}}$ is a random permutation of the possible values of $i \in \mathbb{Z}$ and $\tilde{n} < n$. Classification trees are constructed sequentially from the gradient of the loss function, instead of misclassification rates, of the previous tree [44].

3.2.3. Stacking

Stacking involves a two-level learning structure. When cross-validation is incorporated to this structure, the training dataset L as defined in (1) is split into K equal folds $L_1, L_2, \dots, L_K$, where $k = 1,2, \dots, K$. L_k refers to the validation set and the remaining folds $(L - L_k)$ shown as $L^{(-k)}$ refer to the training set.

In the first level (Level-0), the selected classifiers –called as the base or weak classifiers– are individually trained using the training set and have predictions for the validation set. Given that $c = 1,2, \dots, C$ denoting the selected classifiers, the cth classifier is trained on $L^{(-k)}$ to build a model $M_c^{(-k)}$ that is used to predict the output value y_i for all x_i in L_k. This process runs K times so that each fold is considered as a validation set and a prediction value is obtained for all x_i in L. The predictions for the cth classifier compose a vector $(z_1, z_2, \dots, z_n)^T$. This procedure is applied to all base classifiers and we achieve a matrix $\mathbf{Z}$ of $n \times C$, where each column corresponds to a base classifier's prediction of Level-0. In the second-level (Level-1), a learning algorithm –called as the meta-classifier– builds a model $\widetilde{M}$ that imputes the predictions obtained from Level-0 $(z_{i1}, z_{i2}, \dots, z_{iC})$ as the input of the new training set of Level-1 and maps them to the original class label y_i. [45]

This study includes SVM, bagged CART, RF, C5.0, and SGB for the Level-0 trials of stacking and all algorithms act as a meta-classifier for Level-1 of different experiments

3.3. Support vector machines as a benchmark

Support vector machines (SVM) is a powerful nonparametric supervised learning method based on structural risk minimization [28] and is proven to show good performance especially for binary classification tasks. SVM searches for the optimal separating hyperplane, which is usually a nonlinear decision function $f(x) = (w^T \phi(x) + b)$, where $\phi(x)$ is a nonlinear transform function. The margin between two hyperplanes is calculated as $\left\| {}^{2}/_{w} \right\|$, therefore the solution to the minimization problem in (6) finds the optimal separating hyperplane with the maximum margin leading to a good generalization performance.

$$\hat{\varphi}_{n,B}(x) = sign(\textstyle\sum_{b=1}^{B} \alpha_b \varphi_b(x)) \qquad (6)$$

subject to $y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i$

where $C > 0$ is a regularization parameter called as the penalty term and $\xi_i \geq 0$ $(i = 1,2, \dots l)$ are the slack parameters.

To solve the minimization problem in (6), w can be written in terms of a linear combination of $\phi(x_i)$ such that $w = \sum_{i=1}^{l} \alpha_i y_i \phi(x_i)$. Kernel functions allow the calculation of dot products in a high dimensional feature space $K(x_i, x) = \phi(x_i) \cdot \phi(x)$ and $\phi: X \subset \mathbb{R}^d \to \mathbb{R}^r$ is a transformation that maps x_i to the attribute space, not explicitly to the input space. Hence, the solution is calculated by the decision function $f(x) = \sum_i \alpha_i K(x_j, x) + b$. The maximization problem in (7) solves for $\alpha_i \geq 0$.

$$\max \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \qquad (7)$$

subject to $0 \leq \alpha_i \leq C \quad and \ \sum_{i=1}^{l} \alpha_i y_i = 0$

The solution to this problem is $f(x) = \sum_j \alpha_j y_j (x_j \cdot x) + b$, where x_j's are the support vectors. Motivated by the findings in [46] this study applies SVM with the RBF kernel function (8) as the benchmark classifier, where x_i and x_j refer to n-dimensional inputs, and σ is the shape parameter.

(RBF) kernel:

$$K(x_i, x_j) = exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right) \qquad (8)$$

3.4. Evaluation metrics for imbalanced data classification

Evaluation metrics are used to compare different experimental results as well as to quantify the performance of a classifier in machine learning. The choice of the evaluation metric can completely change the results of the analyses and accordingly the conclusion driven, since each metric has a different assumption of what matters the most about the problem at hand. Accuracy is a commonly used metric that calculates the correctly classification rate of an algorithm but can be misleading for imbalanced datasets since it only considers the overall prediction rate. The metric can yield high values by favoring the majority class for highly skewed class distributions which is the case in the telecom churn practices.

Evaluation results of classifier performances can vary with regards to different metrics for imbalanced classes, yet there is no commonly held metric. The decision of selecting the appropriate model and of evaluating classifiers for imbalanced churn problems should not be based on one metric but a combination of them. This study uses the Receiver Operating Characteristic (ROC) curve, which depicts the true positive rate as a function of the false negative rate for all possible prediction thresholds, as the evaluation metric when training the classifiers. In addition, the metrics in (9) to (11) are used to evaluate the prediction results. The calculation of the selected metrics is based on the confusion matrix values in Table 1.

Table 1. Confusion matrix

	Classified as	
Actual	*Churner*	*Non-churner*
Churner	True Positive (TP)	False Negative (FN)
Non-churner	False Positive (FP)	True Negative (TN)

AUC (The area under curve) = $\frac{1}{2}\left(\frac{TP}{TP+FN}+\frac{TN}{TN+FP}\right)$ (9)

Recall (Sensitivity) = $\frac{TP}{(TP+FN)}$ (10)

Lift = $\frac{TP}{(TP+FN)} \times \frac{1}{P/(P+N)}$,

$P = TP + FN \; and \; N = TN + FP$ (11)

In churn prediction, the FNs referring to the misclassification of churners as non-churners is an important error that should be avoided. In such cases, the operational CRM processes do not show the necessary effort to retain these potential churners that in return has a high financial cost to the company. The AUC distinguishes churners from non-churners and measures the area under the ROC curve (9). Sensitivity (10), and Lift (11) eliminate the drawbacks of traditional metrics for imbalanced classes and expose the correctly classification of the churner class. The minimum (the worst) score and the maximum (the best) score for the metrics formulated in (9) and (10) are 0 and 1, respectively. The lift value is between the range [0, ∞] and we look for high values.

4. EMPIRICAL EVALUATION DESIGN

This study observes the effects of two different strategies, resampling and ensembles, as well as their hybridization on customer churn prediction performance to answer the research questions stated in Section 1.

After preparing the dataset for the experiments –the details are in Section 5-, the first step is dividing the dataset into training and test sets preserving the IR ratio. Training and test set proportions are specified as 70% and 30%, respectively. After the split, the test set does not interfere with any step of the model building and sets aside until we are ready to proceed with prediction in order to align the setup to practice. Preprocessing the attributes of the training set by the z standardization [$z = (x - \mu)/\sigma$, where μ is the mean value, and σ is the standard deviation of an attribute] rescales data for to reduce the impact of degree of attribute magnitudes on classifiers.

The other steps of the empirical design change regarding the research question and hence the method used. The study pursues two strategies, resampling and ensemble learning, to handle class imbalance. Under- and over-sampling methods are incorporated to the model construction, yet their setup is different when the cross-validation procedure is applied. For under-sampling, the preprocessed training set is first resampled implementing the method of concern and then cross-validation is performed with the learning algorithm. For over-sampling on the other hand, applying the same procedure as in under-sampling –apply CV after resampling– may cause over optimism as explained in [47], [48] which is a crucial point most studies ignore. There is a probability that the resampled training folds and the test fold contain the same samples, which may lead to a significantly biased prediction. The correct way to combine over-sampling with CV is through first dividing the training dataset into k folds and then resampling each training fold and not the validation fold. Once the learning model is constructed, the validation fold is used for prediction. Hence, it is guaranteed that the classifier is not exposed to any repetitions of the validation set examples in the learning phase. This problem does not occur in under-sampling for either of the designs since CV after under-sampling and the integration of under-sampling and CV yield the same performance. Fig.1 depicts the methodology of applying CV with the resampling methods and also contains the hybridization of stacking and CV (Note that L_{Rk} refers to the resampled learning data for the kth fold, and n' is the resampled training data size).

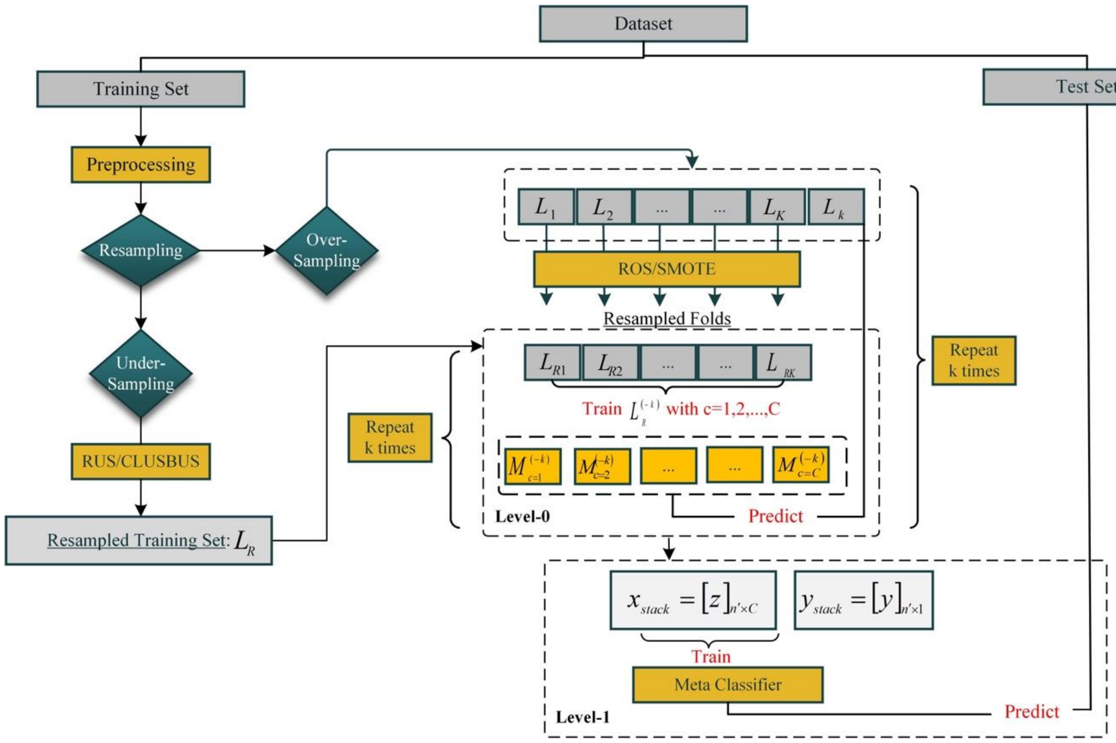
Dataset
Training Set
Test Set
Preprocessing
Resampling
Over-Sampling
Under-Sampling
RUS/CLUSBUS
Resampled Training Set: L_R
L_1 L_2 ... ... L_K L_k
ROS/SMOTE
Resampled Folds
L_{R1} L_{R2} ... ... L_{RK}
Train $L_R^{(-k)}$ with c=1,2,...,C
$M_{c=1}^{(-k)}$ $M_{c=2}^{(-k)}$ ... ... $M_{c=C}^{(-k)}$
Repeat k times
Predict
Level-0
Repeat k times
$x_{stack} = [z]_{n'\times C}$
$y_{stack} = [y]_{n'\times 1}$
Train
Meta Classifier
Predict
Level-1

Fig. 1. Flowchart for cross-validation with resampling and stacking hybridization

Another decision to make is the imbalance ratio to target with resampling. However, as emphasized in [32], [49] the best class distribution changes with respect to each training set size and algorithm. Therefore, this study uses the default IR parameters of resampling techniques which provide a more practical approach. Hence, the majority and the minority class sizes become equal (IR=1) with the application of RUS, ROS, and CLUSBUS, whereas the IR becomes 1.33 with the application of SMOTE. According to CLUSBUS technique the dataset is divided into 3 clusters.

The parameters of the algorithms do not invoke any fine tuning to prevent a drastic altering of decision boundaries and to sustain the generalization capability of the models that can resist to minor changes in the data.

After the model building phase on the training set (L) involving the CV procedure, the test set that has no prior involvement in the learning process and kept separate is used for the final model evaluation.

5. RESULTS AND DISCUSSION

This study applies its empirical setup on a well-known churn dataset from the telecom sector. Churn: This is a dataset from the UCI Machine Learning Repository[1]. There are 17 explanatory attributes in total after the elimination of the “State”, “Area.Code”, and “Phone” attributes prior to the analyses and a target feature referring to whether a customer is a churner or not. 15 attributes take numerical values and the remaining 2 attributes take discrete values. There were no missing values, the sample size is 3333 and the IR is 5.9.

The experimental results provide the evaluations of the model performances of different strategies based on the frequently preferred metrics for imbalanced churn datasets. The benchmark values are the SVM results obtained by implementing the binary classification task to the imbalanced datasets with original imbalance ratios. All phases of the model building are coded and executed in R programming language.

Table 2. SVM prediction results: The benchmark

Metric	Performance
AUC	0.738
Sens.	0.498
F1	0.637
Lift	3.455

The results in Table 2 indicate that SVM performs poorly in predicting churners since the value of the AUC ratio is satisfactory (over 0.7), but not supported by the sensitivity measure. Therefore, SVM is affected by the imbalanced structure of the dataset and this costly problem needs improved solutions. The analyses results are informative in terms of answering our research questions presented in Section 1.

[1] The dataset no longer exists in the UCI Repository, but can be retrieved from https://www.kaggle.com/spscientist/telecom-data/data?select=telecom_churn.csv

Table 3. Prediction results for resampling methods combined with SVM

Classifier	Resampling	AUC	Sens	F1	Lift
SVM	*RUS*	0.738	0.498	0.637	3.455
	ROS	0.794	0.616	0.713	4.275
	SMOTE	0.801	0.629	0.722	4.363
	CLUSBUS	0.699	0.437	0.565	3.034

Answer to RQ1: Under-sampling deteriorates the results of SVM even more, whereas over-sampling has a positive effect on the SVM performance. SMOTE yields the best prediction value.

Table 4. Prediction results of ensemble methods

Ensemble type	Classifier	AUC	Sens	F1	Lift
Bagging	*BaggedCART*	0.928	0.894	0.824	6.204
	RF	0.950	0.974	0.679	6.757
Boosting	*C5.0*	0.950	0.934	0.857	6.483
	SGB	0.944	0.925	0.841	6.417
Stacking	*SVM*	0.935	0.905	0.844	6.277
	BaggedCART	0.929	0.887	0.852	6.155
	RF	0.941	0.913	0.856	6.337
	C5.0	0.936	0.902	0.862	6.254
	SGB	0.945	0.917	0.877	6.359

Answer to RQ2: Ensemble strategies significantly improve the results for the churn dataset in comparison to the benchmark results. Among all ensemble techniques experimented in this study, RF performs the best with respect to the AUC, Sensitivity, and Lift measures. This result is in line with the literature findings. Boosted C5.0 also shows a good performance, yet is less successful in predicting the churners than RF.

Table 5. Prediction results of resampling techniques combined with bagging ensembles

Resampling type	Classifier	AUC	Sens	F1	Lift
RUS	*BaggedCART*	0.794	0.604	0.726	4.188
	RF	0.797	0.606	0.734	4.206
ROS	*BaggedCART*	0.895	0.819	0.819	5.685
	RF	0.934	0.917	0.791	6.365
SMOTE	*BaggedCART*	0.844	0.710	0.781	4.927
	RF	0.883	0.792	0.819	5.496
CLUSBUS	*BaggedCART*	0.720	0.477	0.599	3.312
	RF	0.705	0.448	0.576	3.107

Bagging and Boosting methods achieve better results when combined with over-sampling rather than under-sampling techniques (Table 5 and Table 6). For Stacking ensembles, the only combination that does not work well is the one with CLUSBUS. The over-sampling results are neither over-optimistic nor over-fitting since these techniques are applied within cross-validation as explained in Section 4. Under-sampling techniques, however, do not perform well in general. The possible reason for this could be the elimination of some distinctive examples in the process

Table 6. Prediction results of resampling with boosting ensembles

Resampling type	Classifier	AUC	Sens	F1	Lift
RUS	*C5.0*	0.752	0.520	0.663	3.605
	SGB	0.736	0.491	0.636	3.403
ROS	*C5.0*	0.935	0.896	0.867	6.218
	SGB	0.845	0.707	0.793	4.901
SMOTE	*C5.0*	0.864	0.744	0.819	5.164
	SGB	0.879	0.783	0.817	5.435
CLUSBUS	*C5.0*	0.730	0.496	0.614	3.439
	SGB	0.680	0.398	0.534	2.761

Answer to RQ3: The combination of under-sampling strategy with bagging and boosting techniques does not have any effect on churn prediction results. Although RUS technique combined with stacking ensemble performs well in general, CLUSBUS results are disappointing. Over-sampling techniques on the other hand, significantly increase the prediction performances compared to the benchmark results. However, there is a noticeable result overall; which is the performance of RF when there is no resampling involved.

Answer to RQ4: All selected metrics show compatible evaluations in general, except for the F1 ratio. Based on the results we can conclude that F1 is not a preferable metric for imbalanced churn prediction cases. Among all four metrics, when the values are close to each other, it is better to choose the model directed by sensitivity and lift measures. An example to this can be given for the results achieved by the combination of resampling and stacking ensembles (Table 7). The best combination according to the AUC and F1 ratios is when ROS is combined with stacking Bagged CART, while sensitivity and lift ratios indicate the best combination as SMOTE and stacked SVM. In such situations, the decision should be made in favor of the indication of the sensitivity and lift ratios. Both metrics give more importance to the minority class, which is what we are seeking in churn prediction.

Table 7. Prediction results of resampling with stacking ensembles

Re-sampling type	Meta-Classifier	AUC	Sens	F1	Lift
RUS	*SVM*	0.940	0.927	0.803	6.433
	BaggedCART	0.890	0.823	0.781	5.710
	RF	0.923	0.890	0.802	6.173
	C5.0	0.918	0.870	0.829	6.037
	SGB	0.938	0.911	0.843	6.322
ROS	*SVM*	0.950	0.929	0.871	6.446
	BaggedCART	0.953	0.931	0.887	6.461
	RF	0.949	0.924	0.880	6.408
	C5.0	0.952	0.931	0.883	6.457
	SGB	0.942	0.910	0.874	6.312
SMOTE	*SVM*	0.950	0.945	0.819	6.559
	BaggedCART	0.911	0.864	0.803	5.994
	RF	0.930	0.896	0.833	6.216
	C5.0	0.925	0.884	0.835	6.131
	SGB	0.924	0.887	0.821	6.154
CLUSBUS	*SVM*	0.730	0.496	0.614	3.439
	BaggedCART	0.709	0.455	0.584	3.158
	RF	0.719	0.475	0.598	3.298
	C5.0	0.731	0.498	0.617	3.454
	SGB	0.725	0.487	0.607	3.381

6. CONCLUSION

The binary classification of imbalanced datasets is a hot topic of data analytics in recent years and churn analysis is one of the application areas. Particularly churn in the telecommunication sector is a problem that needs a focused attention since the financial cost of misclassification is already known. The purpose of this study is to address the skewed class distribution problem in the domain, and investigate the effects of resampling and ensemble strategies on churn prediction performance.

The study handles the imbalanced dataset classification from a data-level and an algorithm-level approach. We can conclude that ensembles examined under the algorithm-level approach drastically improve the prediction performances. Form the data-level approach, over-sampling tends to yield better results than under-sampling approach in general. A further improvement to this study would be to apply feature selection in the preprocessing step and investigate the effects of feature selection on the performance of resampling and ensemble learning strategies, which could reveal the importance of the data characteristics.

REFERENCES

[1] J. F. Tanner, M. Ahearne, T. W. Leigh, C. H. Mason, & W. C. Moncrief, “CRM in sales-intensive organizations: A review and future directions?”, *Journal of Personal Selling and Sales Management*, 25(2), 169–180, 2005.

[2] H. Singh & H. V. Samalia, “A business intelligence perspective for churn management”, *Procedia - Soc. Behav. Sci.*, 109, 51–56, 2014.

[3] F. F. Reichheld & P. Schefter, “E-Loyalty: Your secret weapon on the web”, *Harvard Business Review*, 78, 105–113, 2000.

[4] J. Lu, “Predicting customer churn in the telecommunications industry — an application of survival analysis modeling using SAS”, Data Mining Techniques, Retrieved from http://www2.sas.com/proceedings/sugi27/p114-27.pdf, 114–127, 2002.

[5] O. Kwon & J. M. Sim, “Effects of data set features on the performances of classification algorithms”, *Expert Systems with Applications*, 40(5), 1847–1857, 2013.

[6] A. Ali, S. M. Shamsuddin, & A. L. Ralescu, “Classification with class imbalance problem: A review”, *International Journal of Advances in Soft Computing and Its Applications*, 7(3), 176–204, 2015.

[7] S. A. Qureshi, A. S. Rehman, A. M. Qamar, A. Kamal, & A. Rehman, “Telecommunication subscribers’ churn prediction model using machine learning”, **In 8th International Conference on Digital Information Management, ICDIM 2013**, 131–136, 2013.

[8] A. Amin, S. Anwar, A. Adnan, M. Nawaz, N. Howard, J. Qadir, & A. Hussain, “Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study”, *IEEE Access*, 4(Ml), 7940–7957, 2016.

[9] A. Aditsania, Adiwijaya, & A. L. Saonard, “Handling imbalanced data in churn prediction using ADASYN and backpropagation algorithm”**, In Proceeding - 2017 3rd International Conference on Science in Information Technology: Theory and Application of IT for Education, Industry and Society in Big Data Era, ICSITech 2017**, 2018-Janua, 533–536, 2017.

[10] H. Faris, “Neighborhood cleaning rules and particle swarm optimization for predicting customer churn behavior in telecom industry”, *International Journal of Advanced Science and Technology*, 68, 11–22, 2014.

[11] M. A. H. Farquad, V. Ravi, & S. B. Raju, “Churn prediction using comprehensible support vector machine: An analytical CRM application”, *Applied Soft Computing Journal*, 19, 31–40, 2014.

[12] U. R. Salunkhe, & S. N. Mali, “A hybrid approach for class imbalance problem in customer churn prediction: A novel extension to under-sampling”, *International Journal of Intelligent Systems and Applications*, 10(5), 71–81, 2018.

[13] H. Li, D. Yang, L. Yang, Y. Lu, & X. Lin, “Supervised massive data analysis for telecommunication customer churn prediction”, *Proceedings - 2016 IEEE International Conferences on Big Data and Cloud Computing, BDCloud 2016, Social Computing and Networking, SocialCom 2016 and Sustainable Computing and Communications, SustainCom 2016*, 163–169, 2016.

[14] P. Li, X. Yu, B. Sun, & J. Huang, "Telecom customer churn prediction based on imbalanced data re-sampling method", *Proceedings of 2013 2nd International Conference on Measurement, Information and Control*, ICMIC 2013, 1, 229–233, 2013.

[15] W. Verbeke, K. Dejaeger, D. Martens, J. Hur, & B. Baesens, "New insights into churn prediction in the telecommunication sector: A profit driven data mining approach", *European Journal of Operational Research*, 218(1), 211–229, 2012.

[16] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, & G. Bing, "Learning from class-imbalanced data: Review of methods and applications", *Expert Systems with Applications*, 73, 220–239, 2017.

[17] X. Yu, S. Guo, J. Guo, & X. Huang, "An extended support vector machine forecasting framework for customer churn in e-commerce", *Expert Systems with Applications*, 38(3), 1425–1430, 2011.

[18] Y. J. Dong, X. H. Wang, & J. Zhou, "CostBP algorithm and its application in customer churn prediction", **In 5th International Joint Conference on INC, IMS, and IDC - NCM 2009**, 794–797, 2009.

[19] Y. Xu, "Predicting customer churn with extended one-class support vector machine", **in Proceedings - International Conference on Natural Computation**, 97–100, 2012.

[20] C. Wang, R. Li, P. Wang, and Z. Chen, "Partition cost-sensitive CART based on customer value for Telecom customer churn prediction", **in 36th Chinese Control Conference (CCC)**, 5680–5684, 2017.

[21] K. W. De Bock, & D. Van Den Poel, "An empirical evaluation of rotation-based ensemble classifiers for customer churn prediction", *Expert Systems with Applications*, 38(10), 12293–12301, 2011.

[22] M. C. Mozer, R. Wolniewicz, D. B. Grimes, E. Johnson, & H. Kaushansky, "Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry", *IEEE Transactions on Neural Networks*, 11(3), 690–696, 2000.

[23] Y. Xie, X. Li, E. W. T. Ngai, & W. Ying, "Customer churn prediction using improved balanced random forests", *Expert Systems with Applications*, 36(3) PART 1, 5445–5449, 2009.

[24] Y. Xie, & X. Li, "Churn prediction with linear discriminant boosting algorithm", **In Proceedings of the 7th International Conference on Machine Learning and Cybernetics- ICMLC**, 1, 228–233, 2008.

[25] A. Idris, A. Iftikhar, & Z. ur Rehman, "Intelligent churn prediction for telecom using GP-AdaBoost learning and PSO undersampling", *Cluster Computing*, 22(s3), 7241–7255, 2019.

[26] J. Xiao, C. He, B. Zhu, & G. Teng, "One-step classifier ensemble model for customer churn prediction with imbalanced class", **In J. Xu, S. Nickel, V. C. Machado, & A. Hajiyev (Eds.), Proc. of the Eightth International Conference on Management Science and Engineering Management**, 281, 843–854, 2014.

[27] Y. Wang, & J. Xiao, "Transfer ensemble model for customer churn prediction with imbalanced class distribution", **In 2011 International Conference of Information Technology, Computer Engineering and Management Sciences- ICM 2011**, 3, 177–181, 2011.

[28] A. Hanif, & N. Azhar, "Resolving class imbalance and feature selection in customer churn dataset", **In 2017 International Conference on Frontiers of Information Technology - FIT 2017**, (2017-Janua), 82–86, 2018.

[29] C. Gui, "Analysis of imbalanced data set problem: The case of churn prediction for telecommunication", *Artificial Intelligence Research*, 6(2), 93–99, 2017.

[30] B. Zhu, B. Baesens, & S. K. L. M. vanden Broucke, "An empirical comparison of techniques for the class imbalance problem in churn prediction", *Information Sciences*, 408, 84–99, 2017.

[31] V. Effendy, K. Adiwijaya, & A. Baizal, "Handling imbalanced data in customer churn prediction using combined sampling and weighted random forest", **In 2nd International Conference on Information and Communication Technology - ICoICT 2014**, 325–330, 2014.

[32] J. Burez, & D. Van den Poel, "Handling class imbalance in customer churn prediction", *Expert Systems with Applications, 36(3 PART 1)*, 4626–4636, 2009.

[33] N. Liu, W. L. Woon, Z. Aung, & A. Afshari, "Handling class imbalance in customer behavior prediction", **In International Conference on Collaboration Technologies and Systems - CTS 2014**, 100–103, 2014.

[34] M. Ahmed, H. Afzal, I. Siddiqi, M. F. Amjad, & K. Khurshid, "Exploring nested ensemble learners using overproduction and choose approach for churn prediction in telecom industry", *Neural Computing and Applications*, 32(8), 3237–3251, 2020.

[35] A. Amin, F. Al-Obeidat, B. Shah, M. Al Tae, C. Khan, H. Ur Rehman Durrani, & S. Anwar, "Just-in-time customer churn prediction in the telecommunication sector", *Journal of Supercomputing*, 1–25, 2017.

[36] Y. P. Zhang, L. N. Zhang, & Y. C. Wang, "Cluster-based majority under-sampling approaches for class imbalance learning", **In IEEE International Conference on Information and Financial Engineering -ICIFE 2010**, 400–404, 2010.

[37] Z. Zheng, Y. Cai, & Y. Li, "Oversampling method for imbalanced classification", *Computing and Informatics*, 34(5), 1017–1037, 2015.

[38] S. J. Yen, & Y. S. Lee, "Cluster-based under-sampling approaches for imbalanced data distributions", *Expert Systems with Applications*, 36, 5718–5727, 2009.

[39] L. Breiman, "Bagging Predictors", *Machine Learning*, 24(421), 123–140, 1996.

[40] C. D. Sutton, "Classification and regression trees, bagging, and boosting", *Handbook of Statistics*, 24, 303–329, 2005.

[41] N. S. Yanofsky, "Probably approximately correct: nature's algorithms for learning and prospering in a complex world", *Common Knowledge*, 21(2), 340–340, 2015.

[42] Y. Freund, & R. E. Schapire, "Experiments with a new boosting algorithm", **In Proceedings of the 13th International Conference on Machine Learning**, 1–9, 1996.

[43] S. L. Salzberg, "C4.5: Programs for machine learning by J.Ross Quinlan. Morgan Kaufmann Publishers inc. 1993", *Machine Learning*, 16, 235–240, 1994.

[44] J. H. Friedman, “Stochastic gradient boosting”, *Computational Statistics and Data Analysis*, 38(4), 367–378, 2002.

[45] K. M. Ting, & I. H. Witten, “Issues in stacked generalization”, *Journal of Artificial Intelligence Research*, 10, 271–289, 1999.

[46] B. Zhu, B. Baesens, A. Backiel, & S. K. L. M. Vanden Broucke, “Benchmarking sampling techniques for imbalance learning in churn prediction”, *Journal of the Operational Research Society*, 69(1), 49–65, 2018.

[47] R. Blagus, & L. Lusa, “Joint use of over-and under-sampling techniques and cross-validation for the development and assessment of prediction models”, *BMC Bioinformatics*, 16(1), 1–10, 2015.

[48] M. S. Santos, J. P. Soares, P. H. Abreu, H. Araujo, & J. Santos, “Cross-validation for imbalanced datasets: Avoiding overoptimistic and overfitting approaches [Research Frontier]”, *IEEE Computational Intelligence Magazine*, 13(4), 59–76, 2018.

[49] G. M. Weiss, & F. Provost, “Learning when training data are costly: The effect of class distribution on tree induction”, *Journal of Artificial Intelligence Research*, 19, 315–354, 2003.