Harnessing Mixed Features for Imbalance Data Oversampling: Application to Bank Customers Scoring

Abdoulaye Sakho[A,B], Emmanuel Malherbe[A], Carl-Erik Gauthier[C] and Erwan Scornet[B]
[A] Artefact Research Center, Paris, France
[B]Laboratoire de Probabilités, Statistique et Modélisation Sorbonne Université and Université Paris Cité, CNRS, F-75005, Paris
[C]Société Générale, Paris, France

This study investigates rare event detection on tabular data within binary classification. Standard techniques to handle class imbalance include SMOTE, which generates synthetic samples from the minority class. However, SMOTE is intrinsically designed for continuous input variables. In fact, despite SMOTE-NC—its default extension to handle mixed features (continuous and categorical variables)—very few works propose procedures to synthesize mixed features. On the other hand, many real-world classification tasks, such as in banking sector, deal with mixed features, which have a significant impact on predictive performances. To this purpose, we introduce MGS-GRF, an oversampling strategy designed for mixed features. This method uses a kernel density estimator with locally estimated full-rank covariances to generate continuous features, while categorical ones are drawn from the original samples through a generalized random forest. Empirically, contrary to SMOTE-NC, we show that MGS-GRF exhibits two important properties: (i) the coherence i.e. the ability to only generate combinations of categorical features that are already present in the original dataset and (ii) association, i.e. the ability to preserve the dependence between continuous and categorical features. We also evaluate the predictive performances of LightGBM classifiers trained on data sets, augmented with synthetic samples from various strategies. Our comparison is performed on simulated and public real-world data sets, as well as on a private data set from a leading financial institution. We observe that synthetic procedures that have the properties of coherence and association display better predictive performances in terms of various predictive metrics (PR and ROC AUC...), with MGS-GRF being the best one. Furthermore, our method exhibits promising results for the private banking application, with development pipeline being compliant with regulatory constraints.

1. Introduction

Addressing class imbalance in binary classification presents a significant challenge across various machine learning applications [30, 27], such as medical diagnosis, customer churn prediction or anomaly detection [14, 23, 30]. In particular, detecting fraud is a prime issue in banking [11, 19]: the vast majority of customers make legitimate transactions, while the fraudulent ones represent only a minority but have a significant operational, regulatory, and reputational impact.

Several seminal works have introduced rebalancing strategies in order to improve the predictive performances of generic classifiers [5, 22]. These strategies can be divided into two categories [25]: model-level strategies, that aim at adapting an existing algorithm, for example by weighting classes or minimizing a specific loss function [4, 20]; and data-level strategies, that act on the original data set by oversampling or undersampling the observations, and are thus model agnostic. A subclass of data-level strategies, named synthetic procedures, generates new samples in the minority class, with many variants introduced in the literature [5, 12, 10]. Data-level synthetic procedures handle class imbalance by generating new informative samples. This contrasts with model-level approaches

Published at Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD) 2025
Artefact Research Center, Paris, France

(which do not add or lose information), data-level undersampling (which entails information loss), and data-level oversampling via duplication (which does not introduce novel information). One key characteristic of synthetic procedure is that most of them are primarily designed to handle numerical features and thus do not handle categorical features [7, 27]. In practice, categorical features are very common in tabular data (e.g. job category, country, gender), and can represent a relevant signal for improving the predictive performances of the learning tasks, such as those described in [9, 8]. This highlights the importance of handling mixed features when generating samples for imbalance data.

One practical benefit of data-level procedures is to enable practitioners to change the final classifier of the learning task at no cost, i.e. without adapting its loss function or other hyperparameter. This flexibility makes them valuable and sometimes necessary in industrial modular pipelines with regular classification model substitution. To satisfy this practical constraint, in this paper, we focus on data-level strategies and, more specifically, synthetic oversampling strategies for tabular data with mixed features. When handling both continuous and categorical features, we emphasize that a major aspect is to ensure their intrinsic coherence of categorical combinations and their association with continuous features. The notion of *coherence* aims at expressing that a combination of categorical variables can be judged as plausible by some business owner, such as a bank analyst. Indeed, generating samples that do not seem credible may lead to a reluctance to apply subsequent machine learning analyses, without mentioning the potential negative impact on the predictive performances. On the other hand, the oversampling strategy needs to preserve the dependence between continuous and categorical features. In other terms, preserving such level of *association* ensures that the distribution of categorical variables conditional on continuous variables remains similar between original and synthetic data. We thus propose an oversampling strategy that generates synthetic samples that are coherent and preserve the association level compared to the original data. Our main contributions are:

- We introduce MGS-GRF, a strategy for mixed features, with a kernel density estimation for continuous ones and a generalized forest for categorical ones.
- We propose a formal definition for the notions of coherence and association of an oversampling strategy, and show that they are positively correlated with predictive performances. In particular, creating implausible samples not only makes the models less trustworthy, but also reduces the performances.
- On simulated data, we prove that SMOTE-NC, probably the most widely used synthetic rebalancing strategy, is not coherent and does not preserve association, thus creating unplausible samples. On the contrary, our method satisfies these properties, thus creating more realistic samples.
- We compare our proposed method with other rebalancing strategies on two banking public data sets and one private data set from a major financial institution. We show that our proposed strategy, MGS-GRF, which is both coherent and preserves association, has the best predictive performances.

2. Related work

Notations We consider a data-set $\{(X^i, Y^i)\}_{i=1}^N$ constituted of N independent pairs, each one distributed as the random variables (X, Y). X takes values in $\mathbb{R}^d \times \mathcal{X}^p$ and Y in $\{0, 1\}$, where $\mathcal{X}$ is the space of categorical features, of finite size. Here, without loss of generality, we assume that the first d features of X, denoted $X_{1:d}$, are continuous, while the p others, denoted $X_{d:}$, are categorical. Similarly, we suppose that the n first samples are labeled $Y = 1$, denoted $\{X^i\}_{i=1}^n$, verifying $n << N - n$ since we work in an imbalanced data setting.

Algorithm 1 All rebalancing strategies in the literature are divided into two parts, the first one handling the continuous features and the second one handling the categorical ones. Accordingly, we encompass all oversampling strategies in Algorithm 1, which describes the generation of a single synthetic example. Algorithm 1 may be run as many times as necessary to obtain the desired number of minority samples. The procedure starts by selecting uniformly at random $c \in \{1, \dots, n\}$ and the corresponding minority sample X^c. Let $\mathrm{NN}_{K,L}(X^c)$ be the set of the $K \in \mathbb{N}^*$ nearest neighbors of X^c among minority samples w.r.t. to a given norm L. Finally, `ContinuousSampler` and `CategoricalSampler` functions are applied on these neighbors, when necessary. We present below the state-of-the art procedures using Algorithm 1.

SMOTE [5] is the most common synthetic procedure for generating continuous new samples in the minority class. Thus, SMOTE does not have a `CategoricalSampler` in Algorithm 1. To generate a new synthetic sample, an observation X^k is drawn uniformly at random among the K nearest neighbors of X^c w.r.t. the L_2 norm. The synthetic sample generation is the following

$$\texttt{ContinuousSampler}\left(X^c_{1:d}, \mathrm{NN}_{K,L_2}(X^c)\right) = X^c_{1:d} + wX^k_{1:d},$$

where $w \sim \mathcal{U}([0,1])$ and with $\mathcal{U}$ the uniform distribution. Note that SMOTE has several variants [10, 12], but, to the best of our knowledge, these variants are also originally designed for continuous input features only.

SMOTE-N is presented in the original paper introducing SMOTE [5]. This methodology is designed only for categorical input. SMOTE-N uses a version of the Value Difference Metric [28], denoted L_{VDM}, as norm. More precisely, for two categorical vectors $u, v \in \mathcal{X}^p$ we have,

$$L_{\mathrm{VDM}}(u_j, v_j) = \sum_{j=1}^{p} \delta(u_j, v_j),$$

where $\delta(u_j, v_j) = 2|p_n(Y = 0|X_j = u_j) - p_n(Y = 0|X_j = v_j)|$. The value $p_n(Y = 0|u_j)$ is the empirical conditional probability that the output class is $Y = 0$ given that the feature j has the value u_j. Note that, in order to compute L_{VDM}, the majority class samples are necessary. To generate a new observation, a sample is drawn uniformly among the minority samples. Then, its nearest neighbors according to L_{VDM} are computed. Finally, the new minority sample is generated by a vote among the previous nearest neighbors along each variable. With Algorithm 1 notations :

$$\texttt{CategoricalSampler}\left(\mathrm{NN}_{K,L_{\mathrm{VDM}}}(X^c)\right) = \mathrm{Vote}\left(\mathrm{NN}_{K,L_{\mathrm{VDM}}}(X^c)\right),$$

where $\mathrm{Vote}\left(\mathrm{NN}_{K,L_{\mathrm{VDM}}}(X^c)\right)_j$, is a vote among the nearest-neighbors for the categorical feature $j \in \{1, \dots, p\}$.

SMOTE-NC is designed to handle data sets containing both continuous and categorical features, and is also presented in the original SMOTE paper [5]. The main idea is to define a distance metric, denoted L_{NC}, that takes into account the categorical features. To this aim, the median $C \in \mathbb{R}$ of standard deviations of all continuous features for the minority class is computed. The L_{NC} takes the form

$$L_{\mathrm{NC}}(X, X') = \sqrt{\sum_{j=1}^{d} \left(X'_j - X_j\right)^2 + C^2 \sum_{j=d+1}^{d+p} \mathbb{1}_{X_j \neq X'_j}}.$$

Then, continuous features are generated using SMOTE interpolation while categorical one are based on a nearest neighbors vote. For SMOTE-NC we have:

$$\texttt{ContinuousSampler}\left(X^c_{1:d}, \mathrm{NN}_{K,L_{\mathrm{NC}}}(X^c)\right) = X^c_{1:d} + wX^{k_c}_{1:d},$$
$$\texttt{CategoricalSampler}\left(\mathrm{NN}_{K,L_{\mathrm{NC}}}(X^c)\right) = \mathrm{Vote}\left(\mathrm{NN}_{K,L_{\mathrm{NC}}}(X^c)\right),$$

Algorithm 1 OverSampler: One iteration for generating a new sample.

Require: $X^1, \ldots, X^n$, `ContinuousSampler` and `CategoricalSampler`, d, p.
  Select uniformly X^c among $X^1, \ldots, X^n$.
  Derive $\mathrm{NN}_{K,L}(X^c)$ the set composed of the K nearest-neighbors of X^c.
  if $d > 0$: **then**
    $Z_{1:d} \leftarrow$ `ContinuousSampler` $\left(X^c_{1:d}, \mathrm{NN}_{K,L}(X^c)\right)$.
  end if
  if $p > 0$: **then**
    $Z_{d:} \leftarrow$ `CategoricalSampler` $\left(\mathrm{NN}_{K,L}(X^c)\right)$.
  end if
  return $Z = [Z_{1:d}, Z_{d:}]$, new minority class synthetic sample.

SMOTE-ENC [22] applies the same procedure as SMOTE-NC except that L_{NC} is replaced by

$$L_{\mathrm{ENC}}(X, X') = \sqrt{\sum_{j=1}^{d} \left(X'_j - X_j\right)^2 + \sum_{j=d+1}^{d+p} C_j^2 \mathbb{1}_{X_j \neq X'_j}}.$$

However, to the best of our knowledge, SMOTE-ENC has no implementation agnostic to the data-set, and in the original repository the computation of C_j differs for each data-set.

While we mainly focused above on the description of `CategoricalSampler`, please note that they have been a large variety of `ContinuousSampler` introduced in the literature [5, 12, 10], whose description is beyond the scope of this paper. Recently, there has been some theoretical studies of `ContinuousSampler`. [25] demonstrated that when used with default parameters on continuous features, SMOTE tends to copy the original minority samples, and is unable to regenerate the original distribution near the density support boundary. [29] showed that several `ContinuousSampler` can be seen as a kernel density estimator, opening thus to a wide range of methods. Similarly, some generative neural network methods have been proposed as `ContinuousSampler`, such as CTGAN [32], a GAN-based procedure, and TabDDPM [15], a diffusion-based approach. One practical aspect of such oversampling strategies is that they are computationally very expensive, especially in real-world high-dimensional applications with millions of rows. Besides, [15, 25] showed that, empirically, SMOTE remains competitive even compared to these recent models [15, see Table 4 and Table 5].

3. Our proposed algorithm: MGS-GRF

In this section, we describe our new algorithm to handle mixed data. It is organized similarly to Algorithm 1, so that we first describe our procedure to generate continuous features before detailing the categorical variables methodology.

3.1. Handling continuous features

Numerous studies have proposed Kernel Density Estimator (KDE) for generating synthetic samples within the minority class for continuous input features. Actually, SMOTE itself can be seen as a KDE with uniform kernel piece by piece [29]. For instance, [17, 18] introduce an oversampling strategy that, based on original samples, adds centered Gaussian noise with a unique diagonal scale matrix to original samples to generate new observations. [21] develops ROSE, a KDE based oversampling strategy which is associated to a unique scale matrix for the whole minority class. Later, [31] proposes a weighted sample KDE oversampling strategy with fixed diagonal scale matrix, thus isotropic, of the

form $h \times I$ with $h \in \mathbb{R}$ and I the identity matrix. h is as in Adasyn [12]: higher weights are given to original minority samples surrounded mostly by majority class samples.

Multivariate Gaussian SMOTE (MGS) is a synthetic procedure for continuous features introduced by [25]. MGS is presented as a variant of SMOTE that generate new samples from multivariate Gaussian distributions and no longer with a linear interpolation. We analyzed the MGS procedure and reformulate it as a Gaussian KDE from the sample smoothing estimator family [26], i.e. with several different full-rank local scale matrices. Furthermore, MGS does not assume the covariance matrix to be diagonal, thus not isotropic, which allows a better adaptivity to the unknown minority class distribution. We choose to generate the continuous features of synthetic samples, $Z_{1:d} \in \mathbb{R}^d$, according to the following density $\hat{f}_{MGS}(Z_{1:d})$ fitted on the original minority samples $\{X^i\}_{i=1,\dots,n}$:

$$\hat{f}_{MGS}(Z_{1:d}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{(2\pi)^{d/2}|\hat{\Sigma}^i|} \exp\left(-\frac{1}{2}(Z_{1:d} - \hat{\mu}^i)^T(\hat{\Sigma}^i)^{-1}(Z_{1:d} - \hat{\mu}^i)\right), \tag{1}$$

where $|\hat{\Sigma}^i|$ denotes the determinant of $\hat{\Sigma}^i$ and

$$\hat{\mu}^i = \frac{1}{K}\sum_{X \in \mathrm{NN}_{K,L_2}(X^i)} X_{1:d}, \quad \hat{\Sigma}^i = \frac{1}{K}\sum_{X \in \mathrm{NN}_{K,L_2}(X^i)} \left(X_{1:d} - \hat{\mu}^i\right)\left(X_{1:d} - \hat{\mu}^i\right)^T$$

are estimated for each minority class sample using its K nearest-neighbors from the minority class, w.r.t. to L_2 norm. We choose a value of $K = d + 1$ in order to possibly obtain full-rank covariance matrices. Besides, Σ^i can be estimated using shrinkage [16, 6] or simply the sample covariance matrix estimate [25], but empirically we got better results with the latter one.

One notes that the underlying distribution of this sampling is a n Gaussian mixture with equal weights. We also remark that ROSE [21] corresponds to the special case where all Σ^i are equal.

3.2. Handling categorical data via Generalized Random Forests

Now, we introduce our selected procedure for generating synthetic categorical features. A first remark when looking at Algorithm 1 is that multi-output classifiers, such as nearest neighbors, can be used to generate the categorical features. Indeed, such models can be trained using only minority samples, aiming at predicting the categorical features $\{X^i_{d:}\}_{i=1}^n$ based on the continuous features $\{X^i_{1:d}\}_{i=1}^n$. Denoting by $\hat{g}$ such trained classifier, based on Algorithm 1, one can repeatedly generate categorical samples as

$$\texttt{CategoricalSampler}(Z_{1:d}) = \hat{g}(Z_{1:d}).$$

Our selected methodology to generate categorical variables relies on Generalized Random Forests (GRF) [1]. The main difference between a random forest [3] and a GRF, is that, given the new point, GRF assigns a probability to each training sample. These probabilities are derive from the frequency of the training samples to fall in the same leaf as the predicted sample. Finally, GRF can be used to estimate any quantity identified via local moment conditions.

We implemented our own version of GRF from the *RandomForestClassifier* class of scikit-learn [24]. In our algorithm, the derivate probabilities are used to draw predicted target from training target vectors (Y^i). The predict procedure of our GRF is detailed in Algorithm 2. Besides, we try several default hyperparameters for our GRF and finally we keep the default values from *RandomForestClassifier* class of scikit-learn for the tree building. Furthermore, we do not apply the principle of honesty [2], and neither scale the target variables.

Algorithm 2 Prediction procedure of GRF

Require: Forest composed of T trees $\mathcal{T}_1, \ldots, \mathcal{T}_T$. A new unlabeled sample $Z_{1:d}$.
$\forall k = 1..T,\ \mathcal{L}_k(Z_{1:d}) \leftarrow$ set of training samples which end up in the same leaf as $Z_{1:d}$ in the tree $\mathcal{T}_k$
for $i \in [1, \ldots, n]$ **do**
$w_{(Z_{1:d})}(X^i) \leftarrow \frac{1}{T} \sum_{k=1}^{T} \frac{\mathbb{1}_{\{X^i \in \mathcal{L}_k(Z_{1:d})\}}}{|\mathcal{L}_k(Z_{1:d})|}$
end for
$Z_{d:} \leftarrow$ Sample $\{X^1_{d:}, \ldots, X^n_{d:}\}$ based on $\{w_{(Z_{1:d})}(X^1_{1:d}), \ldots, w_{(Z_{1:d})}(X^n_{1:d})\}$.
return $Z_{d:}$

3.3. MGS-GRF

Now we detail MGS-GRF, our new procedure that combines MGS and GRF which are described in the previous sections. MGS-GRF follows the three following steps. First, MGS is applied to generate the continuous features of the new synthetic samples. Then, a Generalized Random Forest (GRF) denoted by $\hat{g}_{GRF}$ is trained on all the original minority samples with the continuous features $\{X^i_{1:d}\}_{i=1}^n$ as inputs and the categorical features $\{X^i_{d:}\}_{i=1}^n$, as outputs. Finally, the trained GRF is used to build the categorical features based on the continuous ones generated in the first step. Using Algorithm 1 notations we have,

$$\texttt{ContinuousSampler}(\{X^i_{1:d}\}_{i=1}^n) = Z_{1:d} \sim \hat{f}_{\text{MGS}}$$
$$\texttt{CategoricalSampler}(Z_{d:}) = \hat{g}_{\text{GRF}}\,(Z_{1:d})\,.$$

Our proposed method enjoys the following properties: (i) GRF generates combinations of categorical features that are all from the original minority class. (ii) Due to tree building procedure, GRF may be able to use only the few continuous variables that are relevant to generate the categorical variables, thus ensuring a better correlation between continuous and categorical variables. (iii) The categorical features are generated directly from the continuous ones of the new sample. Thus, they are no longer based on the neighborhood of the central point.

4. Illustrations on simulated data

In the following, we describe our baselines before defining both coherence and associations. We illustrate these notions through numerical simulations. [1]

4.1. Baselines

Now, we introduce different strategies to preprocess the original imbalanced data set. We denote by None strategy the procedure where no rebalancing strategy is applied. CW is the class-weighting strategy while Random Oversampling strategy (ROS) and Random Undersampling Strategy (RUS) are data-level approaches. We also include the synthetic procedure SMOTE-NC, with the default number of nearest neighbors equal to 5. There is no generic implementation of SMOTE-ENC, thus we do not include this strategy (see Section 2).

Besides, we introduce 3 synthetic baselines for our comparison. MGS-NC selects a central point X^c uniformly over minority samples. MGS distribution is used (see Equation 1) with $\hat{\Sigma}^i$ and $\hat{\mu}^i$ computed on the K nearest neighbors $NN_{L_{\text{NC}},K}(X^c)$ of X^c w.r.t. the L_{NC} norm. Then, each categorical variable is generated separately via a vote among the same neighbors. The second baseline, MGS-5NN, applies

[1] All our experiments are available at https://github.com/artefactory/mgs-grf.

MGS on the continuous features, and builds the categorical ones using a $k = 5$ nearest neighbors w.r.t. L_2 norm as multi-output classifier $\hat{g}$ (see Section 3.2). Similarly, MGS-1NN is the same procedure with $k = 1$.

All strategies (except None) resample or generate observations so that each of the two classes contains the same number of observations (balanced data set).

4.2. Numerical illustrations of non-coherence notion

In our first experimental protocol, we want to analyze the distribution of categorical variables via the notion of coherence defined below.

Definition 4.1. We denote by C the set of combinations of categorical features in the original data set. We denote by $C_{Y=1}$ the combination present in the original minority class $Y = 1$. We say that a synthetic oversampling strategy is coherent, with respect to the minority class, if all combinations of generated categorical features belong to $C_{Y=1}$. Accordingly, we say that a minority sample is coherent if its categorical vector belongs to $C_{Y=1}$.

Our main objectives are to detect non-coherent synthetic procedures and assess whether incoherent samples harm predictive performances. We define the coherence value, denoted Coh, by the proportion of coherent synthetic observations generated by a strategy, over all the synthetic data. If we denote by n_g the number of generated samples $Z^{\ell}_{d:}$, we have

$$Coh = \frac{1}{n_g} \sum_{\ell=1}^{n_g} \mathbb{1}_{\{Z^{\ell}_{d:} \in C_{Y=1}\}}.$$

We note that strategies that generate categorical features one by one with a vote are not coherent, as they can mix original combinations. This applies to SMOTE-NC, MGS-NC and MGS-5NN. However, MGS-1NN copies the features of the nearest neighbor from the minority class, thus leading the combination to be originally present in the minority class. Similarly, GRF is coherent because it draws randomly a combination of categorical features from the minority class.

Protocol We simulate a binary classification task data set such that class 0 is overrepresented, with $d = 9$ continuous features and $p = 2$ categorical ones. We denote by $\mathcal{C} = \mathcal{D} \times \mathcal{E}$ the set of combinations of categorical features where $\mathcal{D}$ (resp. $\mathcal{E}$) is the set of possible modalities for the first (resp. second) categorical features. Each categorical feature is composed of m modalities, i.e. $|\mathcal{D}| = |\mathcal{E}| = m$ and $|C| = m^2$, and only m (out of m^2) combinations of categorical features are present in the minority class, written $C_{Y=1} = m$. Only the 3 informative continuous features and the categorical features are used for generating the target Y. Our procedure consists of the following steps :

1. Draw 5000 samples composed of d continuous features as follows $X_{1:d} = (X_1, \ldots, X_d) \sim \mathcal{N}(0, I_d)$.
2. Draw $Z \in C$ such that

$$\mathbb{P}[Z = c | X_{1:d}] = \frac{\exp(-\theta_c^\top X_{1:3})}{\sum_{\ell \in C} \exp(-\theta_\ell^\top X_{1:3})},$$

with c $\in C$. The set of parameters $\Theta = \{\theta_c, c \in C\}$, verify, for all $c \in C$, $\theta_c \in \mathbb{R}^3$. $X_{1:3}$ are the 3 informative components of X, while other X_j values ($j > 3$) do not impact Z value.

3. Draw the target variable Y such that

$$Y|X_{1:d}, Z=c \sim \mathcal{B}(\sigma(\alpha^\top X_{1:3} + \gamma_c)),$$

where $\mathcal{B}$ is the Bernoulli distribution, σ is the logistic function and $\alpha \in \mathbb{R}^3$. The set of parameters $\Gamma = \{\gamma_c, c \in C\}$ verify, for all $c \in C$, $\gamma_c \in \mathbb{R}$. In order to limit the number of coherent combinations present in the minority class, we set high γ_c values only for m different combinations $c \in C$. Besides, we choose α and all γ_c values such that the class $Y = 1$ is underrepresented.
4. Return $[X_1, \ldots, X_d, Z_1, Z_2, Y]$, where $Z_1 \in \mathcal{D}, Z_2 \in \mathcal{E}$ satisfy $Z = (Z_1, Z_2)$.

One combination of Θ, α, Γ We fix the values of Θ, α and Γ once and for all and run the above protocol 50 times. Thus, we produce 50 data sets (with different random seeds), that we split into train and test set. We preprocess the train set with the different rebalancing strategies and train a LightGBM classifier on it. Predictive performances on the test set are displayed in Table 1.

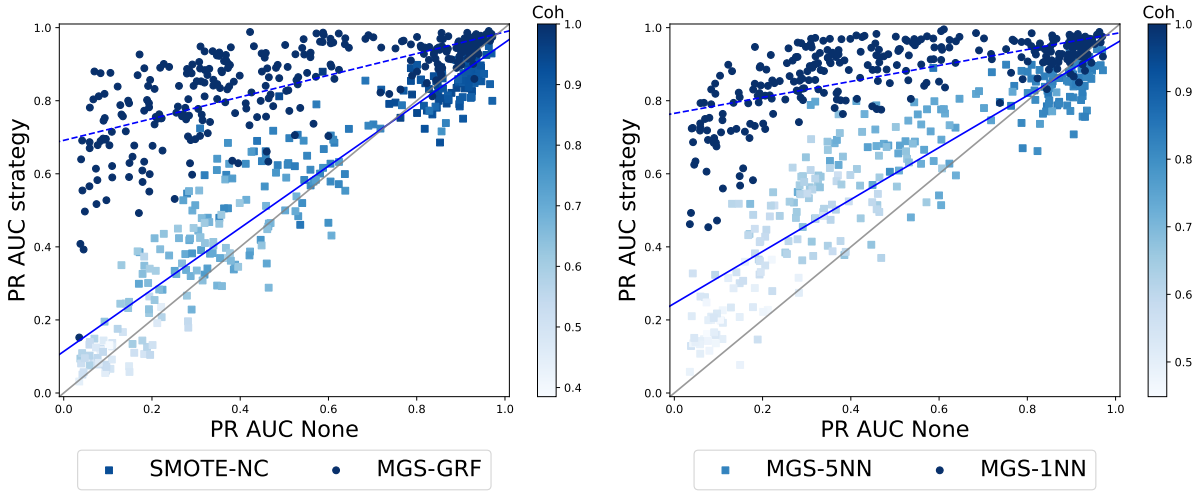
In Table 1, we see that MGS-GRF and MGS-1NN have a coherence value $Coh = 100\%$, which is expected for these two coherent strategies. On the contrary, the non-coherent strategies (SMOTE-NC, MGS-NC and MGS-5NN) have coherence values lower than 100%, as they can generate minority samples whose categorical vectors are not found in the original data set. We observe that strategies with low Coh, i.e. creating non-coherent combinations of categorical features deteriorate the predictive performances of the final classifier. This is particularly visible for MGS-5NN and MGS-1NN, while being very similar models. In contrast, MGS-GRF achieves the best predictive performances in terms of both PR AUC and ROC AUC, with a computation time (for oversampling and LightGBM training) only 50% longer than SMOTE-NC. Finally, we remark that MGS-NC leads to better predictive performances than SMOTE-NC, indicating that MGS seems to better regenerate the distributions of the minority class than SMOTE.

Different combination of Θ, α and Γ We run our protocol with 6 configuration values for Θ, α, Γ. For each configuration, we apply the protocol above, so that we obtain in total 300 datasets. The PR AUC of the LightGBM classifier for each rebalancing strategy and for each data set is displayed in Figures 1a and 1b, where each point corresponds to one of the 300 data sets. We display the PR AUC of a given rebalancing strategy in y-axis and the PR AUC of the None strategy in x-axis. Circles points are all associated to coherent strategies (MGS-1NN and our proposed strategy MGS-GRF), while the squares ones are associated to non-coherent ones (SMOTE-NC, MGS-5NN). We plot linear fitting curves and also add the first bisector in gray (line $y = x$).

In both figures, we remark that the points (both squares and circles) are above the first bisector, thus the rebalancing strategies lead to improvement of PR AUC. However, the average coherent strategies achieve higher PR AUC than the non-coherent ones. This difference is the highest when

Table 1 | LightGBM trained on simulated data from experimental protocol.

Strategy	None	CW	ROS	RUS	SMOTE -NC	MGS -NC	MGS -5NN	MGS -1NN	MGS -GRF
PR AUC	0.903	0.903	0.893	0.699	0.860	0.922	0.870	0.952	**0.954**
ROC AUC	0.975	0.977	0.975	0.935	0.962	0.984	0.970	**0.993**	**0.993**
Coh	**100%**	**100%**	**100%**	**100%**	90%	90%	83%	**100%**	**100%**
Time (s)	0.55	0.56	0.74	0.27	1.01	1.23	1.04	1.00	1.43



(a) SMOTE-NC and MGS-GRF.

(b) MGS-5NN and MGS-1NN.

Figure 1 | *PR AUC* of coherence simulations. Points color reflect their *Coh* value.

the PR AUC of the None strategy is the lowest (left side of the figures), which corresponds to more complex classification settings. In such difficult scenarios, non-coherent strategies have low *Coh* values, which may in turn explain their low PR AUC, close to that of the None strategy. When the learning task is easier, all strategies have similar performance (right side of the figures). All in all, this experiment shows that coherent strategies should be preferred to non-coherent ones, especially in more difficult classification problems.

4.3. Numerical illustrations of association notion

In the following, we define and present our numerical experiments on association.

Definition 4.2. The association level of a multi-output classifier is its predictive performance when inferring categorical features with respect to continuous ones. This performance is measured as the empirical excess risk w.r.t. Bayes error.

We choose to measure the association level of a classifier via its accuracy on a leave-one-out validation on original minority samples. More precisely, if we write $\hat{Z}^{\ell}_{d:}$ the prediction for $X^{\ell}_{1:d}$ of a given classifier trained on $\{X^i\}_{i\neq\ell}$ (leave-one-out prediction), the association level *Asso* of this classifier is

$$Asso = 1 - \left(\frac{1}{n} \sum_{\ell=1}^{n} \mathbb{1}_{\{X^{\ell}_{d:} \neq \hat{Z}^{\ell}_{d:}\}} - \frac{1}{n} \sum_{\ell=1}^{n} \mathbb{1}_{\{X^{\ell}_{d:} \neq h^*(X^{i}_{1:d})\}} \right),$$

where h^* is the Bayes classifier whose predictions are defined by $h^*(X_{1:d}) = \arg\max_{c\in C} \mathbb{P}(X_{d:} = c|X_{1:d})$, with C the set of combination of categorical features. In practice, we do not have access to the Bayes classifier, and thus to the association level. In such situations, the association level can also be estimated without the last term, that is with the classifier accuracy. Another point is that we focus on original minority points for which we have a ground truth $X^i_{d:}$. While measuring on generated continuous features would be ideal given our oversampling objective (see Section 3.1), we do not have the ground truth for the categorical values of those points, and all reference data for measuring association are with the minority points.

In this second numerical experiments, we generate an imbalance binary classification data based on four input variables: 3 of them are continuous and the remaining one is categorical, with 3 modalities. We add $d-3$ continuous noise variables, which are independent of all previous variables. More precisely:

1. Draw 5000 samples as a mixture of 3 Gaussian in $\mathbb{R}^3$: $(X_1, X_2, X_3) \sim \sum_{w=1}^{3} \pi_w \mathcal{N}(\mu_w, \Sigma_w)$, with $\sum_{w=1}^{3} \pi_w = 1$ and $\pi_w \geq 0$. Let $W \in \{1, 2, 3\}$ be the latent variable of the mixture s.t. $(X_1, X_2, X_3)|W \sim \mathcal{N}(\mu_W, \Sigma_W)$.
2. Draw $d-3$ noise features: $(X_4, \ldots, X_d) \sim \mathcal{N}(\mu_2, \lambda I_{d-3})$ with $\lambda \in \mathbb{R}^*$.
3. Draw $Z \in \{\text{“}A\text{”}, \text{“}B\text{”}, \text{“}C\text{”}\}$ such that,

$$\mathbb{P}[Z = c|X_{1:d}, W = w] = \frac{\exp(-\zeta_c^\top X_{1:3} + \chi_{w,c})}{\sum_{\ell \in C} \exp(-\zeta_\ell^\top X_{1:3} + \chi_{w,\ell})},$$

   where $\zeta_c \in \mathbb{R}^3$ and $\chi_{w,c} \in \mathbb{R}$. For each Gaussian, that is for each $w \in \{1, 2, 3\}$, we choose $\chi_{w,c}$ such that one modality is associated to the minority class. We emphasize that the notion of *association* between categorical and continuous features occurs at this step, where Z depends only on the 3 informative values $X_{1:3}$ (W is a confounding variable) while others $X_j (j > 3)$ are pure noise.
4. Draw the target variable Y such that

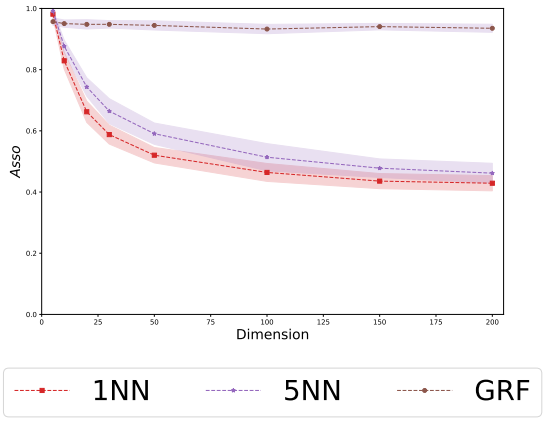
$$Y|X_{1:d}, W=w, Z=c \sim \mathcal{B}(\sigma(\beta^\top X_{1:3} + \eta_w + \phi_c)),$$

   where $\beta \in \mathbb{R}^3$, and $\eta_w, \phi_c \in \mathbb{R}$. One notes that Y depends on $X_{1:3}$, while other X_j $(j > 3)$ are non-informative.
5. Return $[X_1, X_2, X_3, X_4, \ldots, X_d, Z, Y]$.

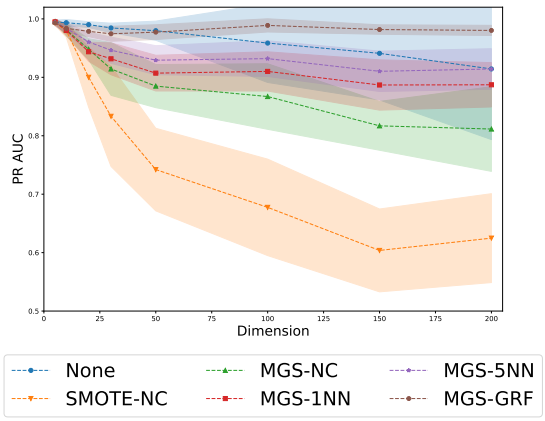
Following this protocol, we generate 8 data sets with increasing number of non-informative features $d-3$. On each of these 8 data sets, we estimate the association level of 3 multi-output classifiers: 1NN, 5NN and GRF. Since we are in a simulation setting, we have access to the Bayes predictor (see Section A for details), which allows us to compute the association level. We observe in Figure 2a, that the association level of nearest neighbors (1NN and 5NN) decreases with increasing dimension, contrary to that of GRF which remains constant, and close to 1, this more accurately generating categorical features. As it is a known behavior for supervised tasks with many noisy uninformative features, nearest neighbors do not predict well the categorical feature.

We now study how the initial prediction task (predicting $Y \in \{0, 1\}$ based on continuous and categorical features) is impacted by the categorical feature generation. On each data set, we apply rebalancing strategies followed by LightGBM (with default hyperparameters) and compute its PR AUC. Results are depicted in Figure 2b. We observe that all methods have the same performance for low dimensions. In this setting, the problem can be considered as easy (since the None strategy has good performances) and all rebalancing strategies are roughly equivalent, similarly to experiments implemented in Section 4.2.

As expected due to the curse of dimensionality, all performances degrade when the dimension increases, with the notable exception of our proposed method MGS-GRF, whose performances remain unaffected by the addition of noise variables. In fact, we see that the use of GRF compared to nearest neighbors (MGS-1NN or MGS-5NN) for generating categorical variables improves the final predictive performances. This finds explanation in the splitting procedure at work in GRF, which selects the variables that are the most predictive of the output (here the categorical input vector). On the contrary, nearest neighbors are unable to detect relevant variables for splitting, which explains their poor performances in high-dimensional settings.



(a) *Asso* w.r.t. to dimension.



(b) PR AUC w.r.t. to dimension.

Figure 2 | Association experiments in high dimensional setting with noisy features.

We also note that SMOTE-NC, probably the default synthetic rebalancing strategy, is the worst in high dimensions, both in terms of mean value and standard deviation. On the opposite, our proposed method MGS-GRF exhibits the best performances with a small standard deviation. This seems to indicate that a good generation of categorical features (via GRF) leads to good predictive performance on the initial binary classification task.

5. Experiments on real-world data sets

In this section, we describe all our numerical experiments on real-world data sets. We describe our protocol before commenting our results.

5.1. Data sets

We use two open source banking-related data sets, Bankmarketing [22] and Bankchurners [33], described in Table 2, both about bank customer behavior prediction. The first data set objective is to predict if a client subscribes to a banking offer after a phone marketing campaign. The second data set aims at predicting customer attrition from a financial institution. Both data set covariates contain historical records of the customers. To be closer to the challenge encountered in the private sector, we undersample the open source data set to have an imbalance ratio of 1%.

We also have a private data set, from Société Générale, that contains clients information from one country in Europe. The purpose is to predict if a customer meets some criterion from historical records. The target criterion is beyond the scope of this paper. Positive cases predicted by the model are pushed to analysts, with corresponding explainability results, and the analysts have to make a decision based on the model output. Furthermore, analysts give feedbacks on the pertinence of

Table 2 | Data sets.

	N samples	n/N imbalance	d continuous	p categorical
Private	$\simeq 10^7$	$<1\%$	>200	<10
BankMarketing	40325	1%	16	10
BankChurners	8585	1%	19	5

pushed cases to the data science team, who retrain the model several times per year. A version from the ML-based system has been deployed. The data set contains millions of *anonymized* customers, and we recall that all process is done in compliance with the country's regulatory requirements.

5.2. Evaluation

We evaluate the public data sets with the following protocol. For an iteration, the data set is evaluated through a 5-fold cross validation, with Z-score scaling of the train set. We stress on the fact that each strategy is applied on the same training set. We run this protocol 20 times and averaged the metrics from each run. The private data set is evaluated through a temporal train/test split, with the test set covering year 2023 and no overlap of clients between train and test sets. Tree-based models produce state-of-the-art performances on tabular data sets [9] and we choose LightGBM [13] as classifier due to its computational efficiency.

We introduce an evaluation metric, the precision at recall, denoted Pr-at-rec(x), which equals the precision associated to a recall of at least x, for any $x \in [0, 1]$. This metric aims at representing an industrial or operational trade-off between precision and recall. After discussions with the analyst, we choose a recall $x = 0.2$. We also use two usual aggregated metrics, the ROC AUC and the PR AUC. Results are displayed in Table 3.

5.3. Results

In Table 3, we first observe that oversampling strategies that preserve coherence (MGS-1NN and MGS-GRF) leads to better predictive performances than the non-coherent ones (SMOTE-NC, MGS-NC, MGS-5NN). Besides, we remark that SMOTE-NC induces the greatest deterioration of predictive performances, for example -28% of PR AUC on the private data set. Furthermore, MGS-NC strategy leads to better predictive performances than SMOTE-NC for all three data sets, reinforcing conclusions of [25]: the MGS KDE is better suited than SMOTE linear interpolation for minority class continuous features regeneration. We also see that our proposed method MGS-GRF has the best predictive performances for BankChurners and BankMarketing in Table 3 for all metrics, with a running time (for oversampling and LightGBM training) close to that of SMOTE-NC.

For the private data set, we observe that MGS-GRF and CW are the two best strategies (in italics). Those are promising results for our method and validate our findings of Section 4.2 and Section 4.3 on association and coherence. To take advantage of both strategies, we built an ensemble learning model CW×MGS-GRF combining the two LightGBM obtained after CW and MGS-GRF strategies. This methodology obtains the best results by far, highlighting the fact that samples generated via MGS-GRF procedure are informative for the learning task.

6. Conclusion and perspectives

In this paper, we propose an oversampling strategy, MGS-GRF which synthesizes continuous features with a kernel density estimator and categorical ones by a GRF. We show through our first experimental protocol with simulated data (Section 4.2), that coherent strategies (MGS-1NN and MGS-GRF) lead to better predictive performances in terms of PR AUC and ROC AUC. Then, in Section 4.3, we show that nearest-neighbor based oversampling strategies are not well suited to handle categorical variables, since they do not preserve the association of generated samples in the presence of noisy features.

We performed numerical experiments on two real-world open source data sets and an industrial private data set from a financial institution which is used in production. Our results show that

Table 3 | BankChurners, BankMarketing and Private data sets. For confidentiality motivations, private data set metrics are relative gains compared to None strategy and no running time is provided. Standard deviations are available in Table 4.

Metric	Data	Strategy									
		None	CW	ROS	RUS	SMOTE -NC	MGS -NC	MGS -5NN	MGS -1NN	MGS -GRF	CW×MGS-GRF
Pr-at -rec (0.2)	Churn	0.894	0.870	0.847	0.632	0.850	0.908	0.910	0.913	**0.930**	-
	Mark.	0.119	0.118	0.115	0.106	0.093	0.128	0.126	0.128	**0.129**	-
	Private	*Ref.*	*9*%	2%	*9*%	-34%	7%	*9*%	*9* %	*9*%	**13**%
PR AUC	Churn	0.622	0.608	0.576	0.394	0.595	0.655	0.653	0.663	**0.664**	-
	Mark.	0.092	0.090	0.090	0.082	0.076	0.099	0.099	0.098	**0.100**	-
	Private	*Ref.*	*11*%	7%	10%	-28%	8%	8%	10%	*11*%	**15**%
ROC AUC	Churn	0.977	0.971	0.963	0.941	0.975	0.983	0.983	**0.984**	**0.984**	-
	Mark.	0.890	0.882	0.878	0.881	0.861	**0.899**	**0.899**	**0.899**	0.898	-
	Private	*Ref.*	0%	0%	0%	-2%	0%	0%	0%	0%	0%
Time (s)	Churn.	0.30	0.33	0.49	0.06	0.85	2.15	1.24	1.22	0.89	-
	Mark	1.29	1.34	1.91	0.29	4.21	16.27	8.05	7.77	5.87	-

MGS-GRF is the most promising strategy for real-world applications, achieving the best predictive performances. Thus, we recommend designing strategies are coherent and preserve association and, among those, we recommend the use of our proposed method MGS-GRF, which achieves the best predictive performances on real-world data sets.

6.0.1.

We would like to express our gratitude to the following individuals for their valuable help and feedbacks : Vincent AURIAU, Mohamed CHTIBA, Jean-Baptiste JANVIER, Martin KLIEBER and Benjamin BOSCH.

References

[1] Athey, S., Tibshirani, J., Wager, S.: Generalized random forests (2019)

[2] Biau, G.: Analysis of a random forests model. The Journal of Machine Learning Research (2012)

[3] Breiman, L.: Random forests. Machine learning (2001)

[4] Cao, K., Wei, C., Gaidon, A., Arechiga, N., Ma, T.: Learning imbalanced datasets with label-distribution-aware margin loss. Advances in neural information processing systems (2019)

[5] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research (2002)

[6] Chen, Y., Wiesel, A., Eldar, Y.C., Hero, A.O.: Shrinkage algorithms for mmse covariance estimation. IEEE transactions on signal processing (2010)

[7] Fernández, A., García, S., Galar, M., Prati, R.C., Krawczyk, B., Herrera, F.: Learning from imbalanced data sets (2018)

[8] Garchery, M., Granitzer, M.: On the influence of categorical features in ranking anomalies using mixed data. Procedia Computer Science (2018)

[9] Grinsztajn, L., Oyallon, E., Varoquaux, G.: Why do tree-based models still outperform deep learning on typical tabular data? Advances in neural information processing systems (2022)

[10] Han, H., Wang, W.Y., Mao, B.H.: Borderline-smote: a new over-sampling method in imbalanced data sets learning. International conference on intelligent computing (2005)

[11] Hassan, A.K.I., Abraham, A.: Modeling insurance fraud detection using imbalanced data classification. Congress on Nature and Biologically Inspired Computing (2016)

[12] He, H., Bai, Y., Garcia, E.A., Li, S.: Adasyn: Adaptive synthetic sampling approach for imbalanced learning. International joint conference on neural networks. (2008)

[13] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems (2017)

[14] Khalilia, M., Chakraborty, S., Popescu, M.: Predicting disease risks from highly imbalanced data using random forest. BMC medical informatics and decision making (2011)

[15] Kotelnikov, A., Baranchuk, D., Rubachev, I., Babenko, A.: Tabddpm: Modelling tabular data with diffusion models. International Conference on Machine Learning (2023)

[16] Ledoit, O., Wolf, M.: A well-conditioned estimator for large-dimensional covariance matrices. Journal of multivariate analysis (2004)

[17] Lee, S.S.: Regularization in skewed binary classification. Computational Statistics (1999)

[18] Lee, S.S.: Noisy replication in skewed binary classification. Computational statistics & data analysis (2000)

[19] Li, K., Yang, T., Zhou, M., Meng, J., Wang, S., Wu, Y., Tan, B., Song, H., Pan, L., Yu, F., et al.: Sefraud: Graph-based self-explainable fraud detection via interpretative mask learning. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (2024)

[20] Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. Proceedings of the IEEE international conference on computer vision (2017)

[21] Menardi, G., Torelli, N.: Training and assessing classification rules with imbalanced data. Data mining and knowledge discovery (2014)

[22] Mukherjee, M., Khushi, M.: Smote-enc: A novel smote-based method to generate synthetic data for nominal and continuous features. Applied system innovation (2021)

[23] Nguyen, N.N., Duong, A.T.: Comparison of two main approaches for handling imbalanced data in churn prediction problem. Journal of advances in information technology (2021)

[24] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research (2011)

[25] Sakho, A., Malherbe, E., Scornet, E.: Do we need rebalancing strategies? a theoretical and empirical study around smote and its variants. arXiv:2402.03819 (2024)

[26] Scott, D.W.: Multivariate density estimation: theory, practice, and visualization. John Wiley & Sons (2015)

[27] Spelmen, V.S., Porkodi, R.: A review on handling imbalanced data. In: 2018 international conference on current trends towards converging technologies (2018)

[28] Stanfill, C., Waltz, D.: Toward memory-based reasoning. Communications of the ACM (1986)

[29] Stocksieker, S., Pommeret, D., Charpentier, A.: Generalized oversampling for learning from imbalanced datasets and associated theory: Application in regression

[30] Sun, Y., Wong, A.K., Kamel, M.S.: Classification of imbalanced data: A review. International journal of pattern recognition and artificial intelligence (2009)

[31] Tang, B., He, H.: Kerneladasyn: Kernel based adaptive synthetic data generation for imbalanced learning. In: IEEE congress on evolutionary computation (2015)

[32] Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K.: Modeling tabular data using conditional gan. Advances in neural information processing systems (2019)

[33] Zhyli: Prediction of churning credit card customers [data set] (2020)

A. Details on protocols

For each configuration Θ, α, Γ, from the protocol of Section 4.2, 50 different data sets (with different seeds) are generated. All in all, we obtain 300 datasets. Finally, each data set is composed of 5000 samples with an imbalance ratio less than 10%. For each data set, we apply different rebalancing strategies and apply a LightGBM classifier on the rebalanced data set.

The protocol from Section 4.3 is executed with the following dimensions values d : [5, 10, 20, 30, 50, 100, 150, 200 All the dimensions share the same parameter values. Each dimension simulation is executed 20 times in order to be able to compute standard deviations. Regarding the generation of the samples, let π_1, π_2, π_3 be the proportions of these three Gaussians, we have $\pi_1 = \pi_2 \gg \pi_3$.

Details on Figure 2a The Bayes Classifier for the *Asso* of Figure 2a is derived empirically from a LightGBM trained on continuous features to predict the categorical feature. This latter model is trained on a different simulated data sets with millions of samples. We make this choice because we predict only one categorical feature and because LightGBM is a consistent estimator.

B. Supplementary materials

Table 4 | Table 3 with standard deviations.

Metric	Data	Strategy									
		None	CW	ROS	RUS	SMOTE -NC	MGS -NC	MGS -5NN	MGS -1NN	MGS -GRF	CWxMGS-GRF
Pr-at -rec (0.2)	Churn	0.894	0.870	0.847	0.632	0.850	0.908	0.910	0.913	**0.930**	-
	std	±0.051	±0.056	±0.061	±0.123	±0.053	±0.057	±0.048	±0.050	±0.052	
	Mark.	0.119	0.118	0.115	0.106	0.093	0.128	0.126	0.128	**0.129**	-
	std	±0.009	±0.009	±0.011	±0.012	±0.007	±0.014	±0.008	±0.012	±0.008	
	Private	*Ref.*	*9%*	2%	*9%*	-34%	7%	*9%*	*9 %*	*9%*	**13%**
PR AUC	Churn	0.622	0.608	0.576	0.394	0.595	0.655	0.653	0.663	**0.664**	-
	std	±0.024	±0.026	±0.024	±0.043	±0.021	±0.026	±0.024	±0.022	±0.025	
	Mark.	0.092	0.090	0.090	0.082	0.076	0.099	0.099	0.098	**0.100**	-
	std	±0.008	±0.005	±0.006	±0.006	±0.004	±0.005	±0.005	±0.006	±0.006	
	Private	*Ref.*	*11%*	7%	10%	-28%	8%	8%	10%	*11%*	**15%**
ROC AUC	Churn	0.977	0.971	0.963	0.941	0.975	0.983	0.983	**0.984**	**0.984**	-
	std	±0.005	±0.005	±0.006	±0.008	±0.004	±0.002	±0.003	±0.003	±0.002	
	Mark.	0.890	0.882	0.878	0.881	0.861	**0.899**	**0.899**	**0.899**	0.898	-
	std	±0.003	±0.004	±0.003	±0.004	±0.004	±0.003	±0.002	±0.003	±0.003	
	Private	*Ref.*	0%	0%	0%	-2%	0%	0%	0%	0%	0%
Time (s)	Churn.	0.30	0.33	0.49	0.06	0.85	2.15	1.24	1.22	0.89	-
	std	±0.01	±0.02	±0.03	±0.00	±0.07	±0.05	±0.04	±0.06	±0.03	
	Mark	1.29	1.34	1.91	0.29	4.21	16.27	8.05	7.77	5.87	-
	std	±0.02	±0.03	±0.09	±0.01	±0.05	±0.56	±0.09	±0.20	±1.34	