Ratio Mathematica Volume 53, 2024

Machine Learning Algorithms Analysis of Synthetic Minority Oversampling Technique (SMOTE): Application to Credit Default Prediction

Emmanuel de-Graft Johnson Owusu-Ansah*
Richard Doamekpor†
Richard Kodzo Avuglah‡
Yaa Adwubi Kyere§

Abstract

Financial risk management requires credit default prediction to measure borrower default risk. Class Imbalance in datasets can hinder model performance. This study examines how under-sampling, oversampling, and SMOTE affect classifier performance. XGBoost outperformed other classifiers in AUC across most datasets except SMOTE using a Ghanaian commercial bank dataset.It concludes that oversampling outperforms under-sampling in enhancing classifier performance and improving the prediction of minority class occurrences, demonstrating the value of data balancing in credit scoring models.

Keywords: SMOTE; Class Imbalanced; Machine Learning; Resampling; Credit Worthiness

2020 AMS subject classifications: 6208, 6211, 62D99 [1]

*Statistics and Actuarial Sci., Kwame Nkrumah University of Science & Technology, Ghana; edjowusu-ansah.cos@knust.edu.gh

†Statistics and Actuarial Sci., Kwame Nkrumah Univ. of Sci. & Tech.;mylordc2@gmail.com

‡Statistics and Actuarial Sci., Kwame Nkrumah Univ. of Sci. & Tech.; avuglah@gmail.com

§Statistics and Actuarial Sci., Kwame Nkrumah Univ. of Sci. & Tech.; avuglah@gmail.com

[1]Received on May 12, 2024. Accepted on December 5, 2024. Published on December 31, 2024. DOI:10.23755/rm.v53i0.1601. ISSN: 1592-7415. eISSN: 2282-8214.

1 Introduction

Imbalanced learning is one of the most challenging tasks in data mining. In imbalance learning, some classes have inadequate data while others have an abundance of data. Imbalanced data sets are often encountered in ”anomaly-type” classification problems. These issues are divided into two categories: ”normal” and ”anomaly,” with ”anomaly” including a broad variety of meanings. Examples include medical diagnosis Chan P [1999], machine failure detection Rao R, and fraud detection Kerdprasop K [2011].

When trying to foresee occurrences, it is critical to concentrate on the minority class since it varies from the majority class's typical trends. Standard classifiers are designed to decrease overall classification errors, regardless of class distribution. As a consequence, these classifiers emphasize majority-class samples above minority-class accuracy. There are three fundamental strategies for correcting data imbalance: cost-sensitive, algorithm-level, and data-level. The cost-sensitive approach calculates misclassification costs using class relevance and imbalance. Examples of cost-sensitive techniques are AdaCost [Fan W,] and the work in [Sun Y, 2007]. The data level approach entails changing the data to re-balance minority and majority categories. To do this, either under-sample the majority class or over-sample the minority class. The imbalance technique is extensively used due of its simplicity and universality, which make it usable regardless of classifier type. Behind the sample overlay, the classifier is intact. Under-sampling (Figure1) is the process of removing less significant patterns using random or heuristic criteria. Examples of under-sampling include the condensed nearest neighbor rule [Batista G E, 2004,] and one-sided selection [Fayed H, 2009].

Figure 1: Undersampling and Oversampling of an Imbalance data
Source: [Roweida M, 2020]

However, under-sampling is risky since critical information may be overlooked. Over-sampling [Fayed H, 2009] is most likely a more successful method. Minority class patterns may be repeated at random or created from scratch. Most sam-

pling algorithms are used in conjunction with ensemble learning. We generate several sample sets and build classifiers for each of them. Finally, we incorporate the classification of these classifiers. Examples of such models include SMOTEBoost [Chawla N, 2003] and RAMOBoost, which combine ensembles with oversampling [Chen S, 2010]. Other algorithms, such as RUSBoost [Liu X, 2009], EasyEnsemble, and BalanceCascade [Seiffert C, 2010]), generate datasets with under-sampled majority classes. Over-sampling (Figure 1) makes up for a lack of data in the minority class by producing additional data. The goal is to produce new data that is similar to the unknown underlying distribution.

In recent years, the growth of machine learning algorithms has transformed a wide range of industries, from healthcare to banking [Chawla N. V., 2002]. However, the performance of these algorithms is strongly dependent on the quality and balance of training data. In real-world circumstances, datasets often show class imbalance, with one class greatly outnumbering the other(s). This mismatch presents a significant difficulty for machine learning algorithms, since they tend to favour the majority class, resulting in inferior performance in forecasting the minority class(es) **?**. Imbalanced data occurs in a variety of scenarios, including fraud detection, illness diagnosis, and anomaly detection, when positive occurrences (for example, fraudulent transactions, uncommon diseases) are considerably less common than negative ones. Traditional machine learning methods, especially those relying on accuracy measures, fail to correctly categorize minority class cases owing to their low presence in the dataset. As a consequence, these algorithms often produce large false negative rates, missing important cases of relevance Fernndez A.) To address the issues provided by unbalanced data, academics and practitioners have devised a slew of strategies targeted at enhancing algorithm performance. One frequent strategy is to adjust the learning algorithms itself, for as by integrating class weights to punish minority misclassifications more harshly. Additionally, resampling approaches, such as oversampling the minority class and under-sampling the majority class, have been commonly used to re-balance the dataset. Furthermore, ensemble approaches, including as bagging and boosting, have shown useful in dealing with unbalanced data by merging many classifiers to reduce bias toward the majority class. Other sophisticated approaches like as anomaly identification and anomaly-based oversampling have shown potential in finding and creating synthetic instances of the minority class to supplement the training set [Sun Y.]. Despite these advances, determining the best strategy for a particular dataset remains a difficult issue, since the success of each method is largely dependent on the individual properties of the data and the learning algorithm used.

Statistical and machine learning techniques, such as logistic regression, discriminant analysis, naive bayes, decision tree, random forest, artificial neural networks, support vector machine, extreme gradient boosting, and adaboost, have

been widely used in developing scorecards and have successfully worked on real-life default data sets [Dastile et al., 2020]. The scoring methods are designed to divide credit applicants into two or more different categories based on demographic and socioeconomic factors such as loan amount, purpose, and age. Credit scoring model training has faced challenges such as explaining predictions to policymakers and customers, as well as the fact that almost all default data sets have uneven class distribution, making predictions biased [Dastile et al., 2020]. The issue of model transparency has been addressed in the literature by utilizing basic linear models such as logistic regression and linear discriminant analysis, as well as rule extraction approaches to explain certain black-box models. Again, strategies such as data resampling and algorithm modification have been devised to offset the effects of the imbalanced class distribution seen in default data sets. Imbalanced data sets have been shown to create models with many incorrect predictions because the algorithms are biased toward the majority class. Automated credit scoring algorithms can handle a large number of loan applications in a short period of time and minimize bias associated with subjective expert judgment Hand and Henley [1997]. Due to the influence of imbalance data on the classification model performance, this study seeks to employ the SMOTE technique in dealing with imbalance data and apply on the machine learning algorithms built on predicting credit worthiness. This study utilized an imbalanced datasets of 'a commercial bank in Ghana Customer Credit worthiness or loan default' to compare sampling g techniques using various evaluation metrics built from the credit worthiness scorecard.

2 Related Work

Researchers have increasingly focused their attention on the imbalance data dilemma. Wilson [Wilson, 1972] devised a well-known under-sampling strategy. It operates by removing data points when the target class does not match the majority of its KNNs. Monard [Monard M. C.], examined numerous challenges connected to learning with skewed class dispersal. For example, consider the relationship between class scatterings and price-sensitive information, as well as the mistake frequency and accuracy bounds for measuring model act. Visa [Vis] offered a study of the most generally used techniques of learning from unbalanced classes. They suggested that the poor performance of models generated using conventional machine learning techniques on unbalanced classes is mostly due to three issues: mistake costs, class dispersion, and accuracy. Cohen [Cohen G, 2006] offered resampling strategies to identify minority targets. They used a novel resampling strategy that included oversampling occasional positives and under-sampling the non-infected majority based on simulated conditions generated by class-specific

sub-clustering. the study found that the innovative resampling method outperformed standard random resampling. In the works of Duman [Duman E, 2012], writers utilized three distinct approaches on an advertising dataset. Logistic regression, Chi-squared, and neural networks are examples of automated interaction discovery. The three approaches' performance was measured using accuracy, AUC, and precision. They compared many alternative skewed datasets derived from the original dataset. They argued that accuracy is a useful metric for an unbalanced dataset. In Lemmatre's work [Lematre G, 2017], k-means grouping is used to balance unbalanced instances by minimizing the number of majority instances, and in some previous works, authors used an under-sampling strategy to delete information points from the majority of occurrences based on their spacing [Mani I, 2003].

Moreover, the handling of unbalanced data has been widely examined in the literature (Japkowicz N [2002], Kamalov F [2022], Wang et al. [2018], Haixiang G [2017], Wang L [2021], Kaur [2019]). However, just a few publications have provided theoretical or empirical assessments of data sampling strategies, namely SMOTE. This review focuses mostly on these works. Elreedy and Atiya [Elreedy D, 2019] calculate the expectation and covariance matrix of SMOTE-generated patterns. However, the research presented here is not limited to the moments, since we create a mathematical framework for the density function of the SMOTE-generated samples. This study differs from Elreedy and Atiya's work [Elreedy D, 2019] in that the former relies on approximations, while this work presents an accurate formula. Studying the density properties of SMOTE patterns might drive the development of density-oriented oversampling algorithms (Yan Y [2022], Mayabadi S [2022]). To the best of our knowledge, this is the first analytical formula for the density of SMOTE-derived materials. Elreedy and Atiya [Elreedy D, 2019] provide a complete examination of SMOTE characteristics. Moniz and Monteiro provide a theoretical examination of resampling methods [Moniz and Monteiro, 2021]. The authors specifically use no-free-lunch machine learning theorems to address unbalanced learning. Furthermore, they provide a comparative empirical examination of several resampling approaches, including random under-sampling, random over-sampling, importance sampling, SMOTE, and SMOTE coupled with random under-sampling. Given no previous information or data assumptions, the authors infer that any two resampling procedures would perform similarly in classification.

Extensive research has focused on using quantitative models to assess loan applicants' creditworthiness [Baesens et al., 2003, Chawla N. V., 2002, Li and Chen, 2020, Malekipirbazari and Aksakalli, 2015]. Such models speed up credit decision-making processes, leading in the development of loan markets. Researchers and practitioners have created effective models employing statistical, machine learning, and deep learning approaches on a variety of real-world datasets.

According to Baesens et al. [2003], sophisticated approaches like neural networks and LS-SVM outperform traditional methods for identifying borrowers as defaulters or non-defaulters. However, basic linear models such as linear discriminant analysis and logistic regression might be effective alternatives to complicated approaches. Similarly, Bellotti and Crook [2009] examined the classification performance of logistic regression (LR), support vector machine (SVM), and discriminant analysis (LDA) on a credit card dataset. They determined that the SVM classifier was equivalent to both LR and LDA. They said that SVM may also be utilized for feature selection. Brown and Mues [2012] did a comparative evaluation of several data mining algorithms on five default data sets and concluded that random forest and gradient boosting are more resistant to severe class imbalance data. Their results also revealed that classic scorecard-building approaches, such as logistic regression and linear discriminant analysis, performed similarly to newer methods. According to Blanco et al. [2013], credit rating methods have limited use in the microfinance business. So, in order to contribute to the industry, they trained a Multilayer Perceptron (MLP) neural network model and compared its performance to certain statistical models. The research, which included over 5,500 participants, found that the neural networks model (MLP) produced higher classification accuracy and lower misclassification costs than logistic regression, quadratic discriminant analysis, and linear discriminant analysis.

In addition, Guo [2019] examined the prediction power of logistic regression and back-propagation neural networks on a peer-to-peer lending dataset. The research found that back-propagation neural networks outperformed logistic regression in terms of prediction accuracy. Li and Chen [2020] compared five ensemble and five baseline methods on the Lending Club dataset. They discovered that, with the exception of Adaboost, the ensemble algorithms (random forest, XGBoost, lightGBM, and stacking) outperformed the individual methods (neural networks, logistic regression, decision tree, support vector machines, and naive bayes). Random forest outperformed XGBoost, LightGBM, stacking, and Adaboost when used in ensembles. However, XGBoost and LightGBM may be better alternatives than random forest. Li [2019] employed XGBoost to classify borrowers as defaulters or non-defaulters. The results were compared to logistic regression, and XGBoost performed the best. Moreover, Malekipirbazari and Aksakalli [2015] compared LR, RF, SVM, and k-NN using the Lending Club dataset. It was discovered that RF had the greatest performance in categorizing borrowers. Dastile et al. [2020] conducted a literature study on statistics and machine learning models. Ensemble approaches were shown to beat individual algorithms by a wide margin.

Collectively, these research emphasize the necessity of quantitative models in credit assessment and the advantages of utilizing ensemble algorithms to create credit scorecards.

3 Methodology

3.1 Class Imbalance Problem

From the data set for this study, the proportion of bad loan $(y = 1)$ is 77.8% and that of good loan $(y = 0)$ is 22.2%. This even class distribution show that the data set used for the analysis is imbalance. Imbalance data sets have been found in the literature to result in a classifier with bias predictions [López et al., 2013, Chawla N. V., 2002] . The minority class which is usually the class of interest is predicted poorly compared to the majority class. To address this challenge we used random under-sampling and oversampling technique (SMOTE) which are the most used balancing methods to balance the datasets. Under-sampling technique drops some training examples from the majority class while oversampling duplicates the training instances for the minority class to make the two classes more balanced [erbeke et al., 2012]. SMOTE [Chawla N. V., 2002], an advanced type of oversampling technique introduces synthetic training examples to the minority class, thus making the class distribution more balanced.

4 Theoretical analysis of SMOTE

The general case of SMOTE derivation underpinning the theoretical analysis for this application is derived and built upon the works of Dina and Kamalov [2023] and Dina and Atiya [2019] which gave the formulation basis and theoretical proofs laying the foundation for the practical application of the method.

4.1 The case of general distribution

To offer a mathematical foundation for the SMOTE technique, we will do a theoretical study. SMOTE's effectiveness as a sampling method depends on its ability to create patterns that closely match the underlying distribution. We will look at this matter here. The mean vector and covariance matrix are the key parameters that define every distribution. We derived approximation formulae for the mean and covariance matrix of SMOTE-generated patterns and compared them to genuine values.

$$\begin{aligned} Let \Delta = X - X_0, then: \\ Z = X_0 - w\Delta \end{aligned} \tag{1}$$

where w is a uniformly produced number in the range [0, w*]. The parameter w, normally higher than or equal to one, enables extrapolation and interpolation on the line joining the pattern X_0 and its randomly picked neighbor X. If $w = 1$, this

returns to the original SMOTE (interpolation only). If $w* > 1$, we can extrapolate beyond the point X_0. Equation 3 and Equation 4 provide final estimates for the mean and covariance matrix of the resulting pattern vector Z. Equations 3 and 4, apply to any distribution of minority class samples $p(X_0)$. In the next subsections, we shall simplify the formulae and derive closed-form solutions for common distributions such the multivariate Gaussian and Laplacian distributions.

$$E[z] \approx \mu_{X0} + \frac{Cw*}{2} \int_{X0} p(X_0)^{\frac{-2}{d}} \frac{\partial p(X_0)}{\partial X}, dX_0 \tag{2}$$

$where \left[\frac{\partial p(X)}{\partial X}\right]^T = \left(\frac{\partial p(X)}{\partial X_1}, ..., \frac{\partial p(X)}{\partial X_d}\right).$

$\sum_Z = \sum_{X0} + \frac{Cw^{*2}}{3} \int_{X0} p(X_0)^{(1-\frac{2}{d})} dX_0 I$

$- \frac{C^2 W^{*2}}{3} \int_{X_0} p(X_0) dX_0 \int_{X0} P(X_0)^{\frac{-2}{d}} \frac{\partial p(X_0)^T}{\partial X} dX_0$

$+ \frac{Cw*}{2} [\int_{X0} p(X_0)^{\frac{-2}{d}} \frac{\partial p(X_0)}{\partial X} [(X_0 - \mu_{X0})^T] dX_0$

$+ \int_{X0} p(X_0)^{\frac{-2}{d}} (X_0 - \mu_{X0}) \frac{\partial p(X_0)^T}{\partial X} dX0]$

where d represents the dimension of the pattern vector, N represent the number of original patterns of the considered class, μ_{X0} is the true mean vector of the class, X_0 is the true covariance function, $p(X_0)$ is the class-conditional density at point X_0, I is the identity matrix, and C is calculated as follows:

$$C = \frac{N! \Gamma(1 + \frac{2}{d})^{\frac{2}{d}} \Gamma(K + \frac{2}{d} + 1)}{\pi K! (d+2) \Gamma(N + \frac{2}{d} + 1)} \tag{3}$$

4.2 The case of multivariate Gaussian distribution

Simplifying the approximation of the true probability representing the multivariate Gaussian distribution

.

$$E[z] \approx \mu_{X0} \tag{4}$$

$$\sum_Z = \sum_{X0} + [(2\pi)^{\frac{1-d}{2}} \frac{Cw^*2}{3} det^{\frac{1-d}{2d}} (\sum_{X0}) (\frac{d}{2d-1})^{\frac{d}{2}} - 2\pi C w^* det^{\frac{1}{d}} (\sum_{X_0}) (\frac{d}{d-2})^{\frac{d+2}{2}}] I \tag{5}$$

where det refers to the determinant. The preceding Equation 6 is valid for dimensions $d > 2$ due to a convergence condition. According to Equation 6, the percentage $\frac{d}{d-2}$ is higher than one for every $d > 2$, resulting in a negative third

term in the resultant covariance matrix $\sum_Z$. Equation 6 shows that the second term is smaller than the

Except for big w, the third term in magnitude (for conventional SMOTE, $w = 1$) remains unchanged. SMOTE-generated patterns have a more constricted covariance matrix ($\sum_Z$) compared to the initial minority class instances ($\sum_{X_0}$), as expected. Subtracting the identity matrix times a positive value results in smaller diagonal elements.
From the preceding formulae, one can see the following: SMOTE-generated patterns have a mean vector that closely matches the genuine one. The covariance matrix shows some inconsistency. It is more contractive than the actual one. This supports the previous section's claim that the generation process shifts patterns inward.

While these findings concentrate on SMOTE distributional properties, they also have an indirect influence on classification performance. The Bayes rule states that the posterior probability of the class $p(Y|X)$ depends on the underlying class-conditional densities $p(X|Y)$. The distributional qualities of the data used in the design have an influence on the classification results of most classifiers, even if they are not explicitly employed.

4.3 Case of other distributions: Multivariate Laplace

The real probability density follows multivariate Laplace distribution is defined as follows;

$$p(X) = (\frac{\alpha}{2})^d e^{-\alpha \sum_{i=1}^{d} |X_i - \mu_i|} \tag{6}$$

where α is a distribution parameter, μ is the distribution's original mean, and d is the number of dimensions. The mean and covariance of SMOTE-generated patterns may be approximated as shown

$$E[z] \approx \mu_{X_0} \tag{7}$$

$$\sum_Z = \sum_{X_0} + \frac{4Cw^*}{\alpha^2}(\frac{d}{d-2})^d[\frac{w^*}{3} - (\frac{d}{d-2})]I \tag{8}$$

where C is computed as per Equation 4. The fraction $\frac{d}{d-2}$ is greater than one for any $d > 2$, so the third term of the generated examples' covariance matrix $\sum_Z$ would be negative. Since standard SMOTE, $w* = 1$, the covariance matrix $\sum_Z$ of the patterns generated by SMOTE would be more contracted than the original covariance matrix $\sum_{X_0}$.

4.4 Effect on the classification boundary

The distribution of SMOTE-generated instances follows Equations 2 and 3, which provide the mean and covariance matrix. The Fisher LDA equations for w and w_0 at the decision boundary $w^T x + w_0 = 0$ are defined as:

$$w = (\sum_{x0} + \sum_1)^{-1}(\mu_1 - \mu_{X0}) \tag{9}$$

$$W_0 = -\frac{1}{2}(\mu_1 - \mu_X 0)^T(\sum_{x0} + \sum_1)^{-1}(\mu_1 - \mu_{X0}) \tag{10}$$

The majority class distribution's mean and covariance are μ_1 and $\sum_1$, whereas the minority class distribution's mean and covariance are μ_{X0} and $\sum_{X0}$, respectively. The above equation for w_0 may be written in terms of w as follows:

$$W_0 = -\frac{1}{2}(\mu_{X0} + \mu_1)^T w \tag{11}$$

Using the SMOTE oversampling approach, we may replace Equations 2, and 3 in the Fisher linear discriminant (Equations 10, 11).

$$w_z = (\sum_Z + \sum_1)^{-1}(\mu_1 - \mu_Z) \tag{12}$$

$$W_0 = -\frac{1}{2}(\mu_Z + \mu_1)^T(\sum_Z + \sum_1)^{-1}(\mu_1 - \mu_z) \tag{13}$$

To quantify the difference in decision boundary imposed by SMOTE, we analyze two terms. $\Delta w = ||w_z w||$ and $\Delta w_0 = ||w_{0z} - w_0$.

$$\Delta_w = ||(\sum_Z + \sum_1)^{-1}(\mu_1 - \mu_Z) - (\sum_{x0} + \sum_1)^{-1}(\mu_1 - \mu_{X0})|| \tag{14}$$

Let$A = \sum_{X0} + \sum_1 andb = \mu 1 \mu_{X0}$. Also, let $T = \sum_Z \sum_{X0}$ The second, third, and fourth components in Equation 3 describe the covariance matrix discrepancy caused by SMOTE. Consider $h = \mu_Z - \mu_X 0$, the second term in Equation 2. Then, Δ_w is evaluated as

$$\Delta_w = ||(A + T)^{-1}(b - h) - A^{-1}b|| \tag{15}$$

Applying the following matrix inversion identity:

$$(A + BCD)^{-1} = A^{-1}BCDA^{-1}(I + BCDA^{-1})^{-1} \tag{16}$$

Let B = D = I, where I is the identity matrix. Since $A = \sum_{X0} + \sum_1$, we assume that the covariance matrices are positive definite and so invertible. We get:

$$(A+T)^{-1} = A^{-1} - A^{-1}TA^{-1}(I+TA^{-1})^{-1} \tag{17}$$

Then Δ_w can be evaluated as:

$$\Delta_w = ||(A^{-1} - A^{-1}TA^{-1}(I+TA^{-1})^{-1})(b-h) - A^{-1}b|| \tag{18}$$

which is simplified as

$$\begin{aligned} \Delta_w &= ||(-A^{-1}h - A^{-1}TA^{-1}(I+TA^{-1})^{-1})(b-h)|| \\ &= ||(A^{-1}h + A^{-1}TA^{-1}(I+TA^{-1})^{-1})(b-h)|| \end{aligned} \tag{19}$$

The theoretical study for multivariate Gaussian and multivariate Laplace distributions shows that the mean of SMOTE-generated patterns μ_Z is almost identical to the mean of the original distribution μ_{X0}, implying that $h \approx 0$. Thus, Equation 20 would be as follows:

$$\begin{aligned} \Delta_w &= ||A^{-1}TA^{-1}(I+TA^{-1})^{-1})b|| \\ &= ||A^{-1}(TA^{-1})(I+TA^{-1})^{-1})b|| \\ &= ||A^{-1}[(I+TA^{-1})AT^{-1})]^{-1}b|| \end{aligned} \tag{20}$$

Hence, Δ_w is estimated following:

$$\Delta_w = ||A^{-1}(AT^{-1}+I)^{-1}b|| \tag{21}$$

Similar to Δ_{w_0}, since w_0 is a scalar, we describe the difference in it using absolute difference:

$$\Delta_{w_0} = \frac{1}{2}|{w_z}^T(\mu_Z + \mu_1) - w^T(\mu_{X_0} + \mu_1)| \tag{22}$$

Let$\Delta = w_z - w$, then:

$$\Delta_{w_0} = \frac{1}{2}|(w+\Delta)^T(\mu_{X_0} + h + \mu_1) - w^T(\mu_{X_0} + \mu_1)| \tag{23}$$

simplifying Equation 24:

$$\Delta_{w_0} = \frac{1}{2}|(w+\Delta)^T(\mu_{X_0} + h + \mu_1) - w^T(\mu_{X_0} + \mu_1)| \tag{24}$$

$$\Delta_{w_0} = \frac{1}{2}|w^Th + \Delta^T(\mu_{X_0} + h + \mu_1)| = \frac{1}{2}[w^Th + \Delta^T(\mu_Z + \mu_1)| \tag{25}$$

Similar to the Δ_w derivation, assuming $h \approx 0$ yields the following:

$$\Delta_{w_0} = \frac{1}{2}|\Delta^T(\mu_{X_0} + \mu_1)| \tag{26}$$

5 Classification Techniques Algorithms

Considering $\vec{x} = \left[x_1, x_2, ..., x_n\right] \in \mathbb{R}^n$ where **x** represent n-dimensional feature vector (i.e loan applicants' characteristics). Assuming we have a binary response variable y with values 0 and 1 where y $\in \{0, 1\}$ and $y = 1$ implies defaulted borrowers and $y = 0$ represents non-defaulted borrowers. According to Bayes rule, the probability that an applicant i will default on a loan can be estimated as $p(y = 1 \mid \vec{x})$ while the probability of a non-default event occurring is just the complement of the default event $(i.e\ p(y = 0 \mid \vec{x}) = 1 - p(y = 1 \mid \vec{x})$. The probability that a classifier will predict a borrower as bad can be given as $p(y = 1 \mid \vec{x}) > \delta$. On the other hand, a loan will be approved for a customer i when the default probability of i is $p(y = 1 \mid \vec{x}) \leq \delta$ where δ represent a specified cut-off point. This is a classification problem.

Generally, the rule for classification can be defined as a function $h : \mathbf{x} \rightarrow y$. If a new **x** is observed, y (i.e labels) is predicted to be h(**x**). A classifier seeks to learn how the function h can generalize well on a new data points. This means that if a new data **x** which labels y is unknown is fed into the classifier h, the labels should be predicted to right.

5.1 Linear Discriminant Analysis (LDA)

Linear discriminant and quadratic discriminant analysis are popular classification techniques because their results are transparent, have closed-from solution and they have no hyper-parameter to be tuned. They work better for multi-class problems compared to logistic regression. LDA has a linear decision surface and can be used for dimensionality reduction. A feature vector $\vec{x}$ which represent borrowers' characteristics can be assigned to one of y discrete classes, where $y \in \{0, 1\}$ such that $y = 1$ and $y = 0$ represent defaulted loans and non_defaulted loans respectively. According to Bayes rule, the posterior probability $p(Y \mid X)$ can be written as given in Equation 27. The procedure used to estimate the posterior can lead to discriminant analysis or logistic regression.

$$P(Y \mid X) = \frac{f_k(x_0)\pi_k}{\sum_r f_r(x_0)\pi_k} \tag{27}$$

In both LDA and QDA the class conditional densities $f_k(x_0)$ and the prior π_k are estimated. They both assume $f_k(x_0)$ to be a multivariate Gaussian distribution and π_k is modeled as Bernoulli. LDA assumes equal covariance matrix for the classes (i.e. $\sum_1 = \sum_0$) with means μ_1, μ_0. The decision boundary of defaulted loans $(y = 1)$ and non-defaulted loans is given by Equation 28

$$P(y = 1 \mid X = x) = P(y = 0 \mid X = x) \tag{28}$$

The log of the posterior probability of the LDA then becomes a line in the form as given in Equation 29.

$$\beta^\top x + \alpha = 0 \tag{29}$$

where:

$$\beta^\top = \sum^{-1} \mu_1 - \sum^{-1} \mu_0$$

$$\alpha = \frac{1}{2}\mu_0^\top \sum^{-1} \mu_0 - \mu_1^\top \sum^{-1} \mu_1 + \log \frac{\pi_1}{\pi_0}$$

5.2 Quadratic Discriminant Analysis (QDA)

QDA has a quadratic decision surface and makes no assumption about the covariance matrices of the classes. One major drawback of QDA is that there are more parameters to be estimated compared to LDA. In QDA the covariance matrix for each class needs to be computed. The quadratic discriminant function is therefore computed as follows:

$$\vec{x}^\top A\vec{x} + W^\top \vec{x} + b > 0 \tag{30}$$

where:

$$A = \frac{1}{2}(\sum_0^{-1} - \sum_0^{-1}) \tag{31}$$

$$W^T = (\mu_1^T \sum_1^{-1} - \mu_0^T \sum_0^{-1}) \tag{32}$$

$$b = -\frac{1}{2}\mu_1^T \sum_1^{-1} \mu_1 + \frac{1}{2}\mu_0^T \sum_0^{-1} \mu_0 + K_1 - K_0 + \log \pi_1 - \log \pi_0 > 0 \tag{33}$$

5.3 Logistic Regression (LR)

In logistic regression, instead of estimating $f_k(x_0)$ and π_k, the posterior probability is modeled directly through a logistic function. if $y = 1$,

$$p(y = 1 \mid \vec{x}) = \frac{\exp^{\beta^\top x_i}}{1 + \exp^{\beta^\top x_i}} \tag{34}$$

if $y = 0$,

$$(y = 0 \mid \vec{x}) = \frac{1}{1 + \exp^{\beta^\top x_i}} \tag{35}$$

The posterior is then written as a parametric model in the form

$$p(y \mid \vec{x}) = p(y = 1 \mid \vec{x})^{1-y} * p(y = 1 \mid \vec{x})^{y} \tag{36}$$

Maximum likelihood estimation (MLE) is used to optimized the unknown parameter β. Using MLE does not result in a close form solution so Newton-Raphson method can then be applied. The parameter β in the logistic regression can be computed as follows:

$$\beta^{r+1} = \beta^r + (X^\top W X)^{-1}(\vec{y} - \vec{P})^{-1} X \tag{37}$$

where:
y is $n \times 1$ vector of all response variables
X is $n \times (d + 1)$
$\vec{P}$ a vector of $n \times 1$ such that $p_i = P(x_i \mid \beta)$
W is a diagonal matrix $n \times n$ such that $w_{ii} = (1 - P(x_i \mid \beta))\, P(x_i \mid \beta)$
β^{r+1} is a new β and β^r is an old β

5.4 Decision Tree (DT)

C5.0 algorithm proposed by Quinlan [1986] is one of the best implementations of decision trees because it handles many of the decisions automatically thereby avoiding over-fitting. It is an improvement to the C4.5 algorithm. Decision tree is a well-known white-box techniques. The results of white-box models are easy to interpret. This has made the algorithm one of the best choice to consider when building a credit scoring model because in most countries the reason for rejecting an applicant a loan must be well explained. The algorithm uses entropy to decide which feature in the data to split upon. The feature with the most information is splitted first. The formulae for calculating entropy for sample (S) can be represented as:

$$Entropy(S) = -p_1 \log_2(p_1) - (1 - p_1) \log_2(1 - p_1) \tag{38}$$

where p_1 and $1 - p_1$ represent the proportions of values falling into the defaulted and non-defaulted class respectively.

5.5 Random Forest (RF)

Random forest is one of the best ensemble learning algorithms that are used to solve both classification and regression tasks. The algorithm creates several decision trees and average their results using majority votes [Breiman, 2001]. Even without hyper-parameter tuning, this algorithm normally produce superior results. It has an in-built feature importance plot that enable you to find the features that contribute most to the prediction thereby deciding on which feature to drop or include in the model.

5.6 Support vector machine (SVM)

The goal of SVM classifier is to find the maximum margin hyperplane that best classify future data points into distinct classes. The algorithm can be used for both classification and regression tasks. SVMs are more robust to outliers because the decision function only depends on the support vectors rather than the entire training data. The closest data points to the hyperplane is what are termed as the support vectors. Considering $y_i \in \{+1, -1\} \forall\ i = \{1, 2\}$. The dual optimization problem for SVM can be written as:

$$\max_{\alpha} \left(\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{j=1} \sum_{i=1} \alpha_i \alpha_j y_i y_j x_i^{\top} x_j \right) \tag{39}$$

subject to the constraints $C \geq \alpha_i \geq 0$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$. α_i represents a Lagrange multiplier.

5.7 Artificial Neural Networks (ANNs)

One of the well-known neural network architecture is the multilayer feed forward neural network. In this network there are d-dimensional data points with at least three layers of perceptrons and each perceptron is fed by an input having some random weights. The inputs are weighted and a linear function is applied to the weighted sum of this inputs and we are going to have an output and the output is going to another layer of perceptron known as hidden layer. The process continues until we reach the final output layer.

ANNs seeks to find the optimum weights that minimize the objective function in Equation 40.

$$E = |y - \hat{y}|^2 \tag{40}$$

$$\frac{\partial E}{\partial a_{ie}} = \delta_i * z_e \tag{41}$$

The first weight of the network can be computed by Equation 41, but

$$\delta_i = \sum_j \delta_j * u_{ji} * \sigma'(a_i) = \sigma'(a_i) \sum_j \delta_j * u_{ji} \tag{42}$$

Knowing δ_i, we can compute δ for the last layer k. Each δ depends on the δ of the next layer. The δ for the last layer is computed in Equation 43

$$\delta_k = \frac{\partial E}{\partial \hat{y}} = \frac{\partial |y - \hat{y}|}{\partial \hat{y}} = -2|y - \hat{y}| \tag{43}$$

The value δ_k, can be substituted it in Equation 42 in order to find δ for all the previous layers up to the first layer. We multiply all the computed δ to any z to find the corresponding weight. Having all the derivatives of all of the weights we can do gradient descent as shown Equation 44

$$u_{ie}^{new} \leftarrow u_{ie}^{old} - \rho \frac{\partial E}{\partial u_{ie}} \tag{44}$$

where ρ is some learning rate, u_{ie}^{new} is new weight, u_{ie}^{old} is old weight and $\frac{\partial E}{\partial u_{ie}}$ is the derivative of the error or objective function. Back-propagation using gradient descent find local mean of the objective function instead of the global minima. This is the reason why neural networks was out of favour for a while.

5.8 Extreme gradient boosting (XGBoost)

XGBoost is considered as one of the most accurate, fastest and efficient algorithms and has won several kaggle competitions [Ma et al., 2018, Chen and Guestrin, 2016a]. The technique was introduced in 2014 so its application in developing credit scoring models is limited as compared to logistic regression, ANN, SVMs and Random forest. The scalability of this algorithm is responsible for its high efficiency [Chen and Guestrin, 2016b]. It is an ensemble learning technique. Ensemble learning combines the results of different base learners into a single, but robust model. XGBoost seeks to minimize the objective function

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(\vec{x}_i)) + \Omega(f_t) \tag{45}$$

where $\Omega(f_t)$ and $l(y_i, \hat{y}_i^{(t-1)} + f_t(\vec{x}_i))$ represent a regularization term which penalizes complexity of the model and a loss function respectively [Chen and Guestrin, 2016b].

5.9 Metrics for Performance Measures

Accuracy is the percentage of total true predictions divided by the total number of examples. It can only be a good measure of performance when the data set used to train a classifier is somewhat balanced.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{46}$$

Sensitivity or recall finds how many of the true positives the classifier predicted as positives. It becomes a reliable measure of classifier's efficiency when faced with an imbalanced data set.

$$Sensitivity = \frac{TP}{TP + FN} \tag{47}$$

Specificity is the percentage of the actual negatives which are correctly predicted as negative.

$$Specificity = \frac{TN}{TN + FP} \tag{48}$$

Precision measures how often a classifier is right when it predicts the positive class.

$$Precision = \frac{TP}{TP + FP} \tag{49}$$

With the help of the harmonic mean, F-measure or F1-score combines recall and precision into a single number that can be used to judge a classifiers separability. It becomes appropriate when we want to strike a balance between precision and recall

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{50}$$

AUC measures the ability of a classifier to discriminate between classes.

$$AUC = 0.5 \left(1 + \frac{TP}{FN + TP} - \frac{FP}{TN + FP}\right) \tag{51}$$

5.10 Data Description and Data Pre-processing

The data set used for the analysis was collected from one of the renowned commercial banks in Ghana. All confidential information were dropped from the data. It contained 734 observations and 14 attributes. Out of the 14 variables, 8 were categorical and 6 numeric. The categorical data were encoded into dummy variables. The binary response variable Y was creditworthiness where $y = \{0, 1\}$. $y = 1$ represent defaulted loans and $y = 0$ implies non-defaulted loans. 571

(77.8%) of the observations represent non-defaulters and 163 (22.2%) were defaulters. This uneven class distribution clearly indicate that the dataset was imbalanced. The description of both the categorical and numeric attributes are shown in Table 1 and Table 2 respectively. The data was partitioned into 70% training and 30% test set. Three data balancing techniques: under-sampling, oversampling and the synthetic minority oversampling technique (SMOTE) were employed to balance the data.

Table 1: Characteristics of the categorical variables and their description attributes

Description	Attributes	Levels
Creditworthiness	Credit rating	1 = Default 0 = Non-default
Purpose	Purpose	1 = Buy a car 2 = Funeral 3 = Pay rent advance 4= Buy plot of land 5 = Buy domestic appliances 6 = Complete building 7 = Education 8 = For Business 9 = Others
Educational status	edu_stat	1 = Primary 2 = Secondary 3 = Graduate/tertiary 4 = Postgraduate
Residential status	res_stat	1 = Tenant 2 = Own house 3 = With relatives
Sex	Sex	1 = Male 2 = Female
Marital status	marital	1 = Married 2 = Single 3 = Divorced/ Separated/Widowed
Occupation	occupation	1 = Employed by Government 2 = Employed by Private Sector 3 = Self Employed 4 = Unemployed
Payment of past loan	previous credit stat	1 = Delay in paying of past loan 2 = Past loan paid on time 3 = No Credit at the bank

Table 2: Characteristics of the numeric variables and their description attributes

Description	Attributes	Variable type
years at job	years_at_job	numeric
years at current address	years_at_address	numeric
Number of dependents	dependents	numeric
Applicants age	Age	numeric
Loan amount	Amount	numeric
Loan duration	Duration	numeric

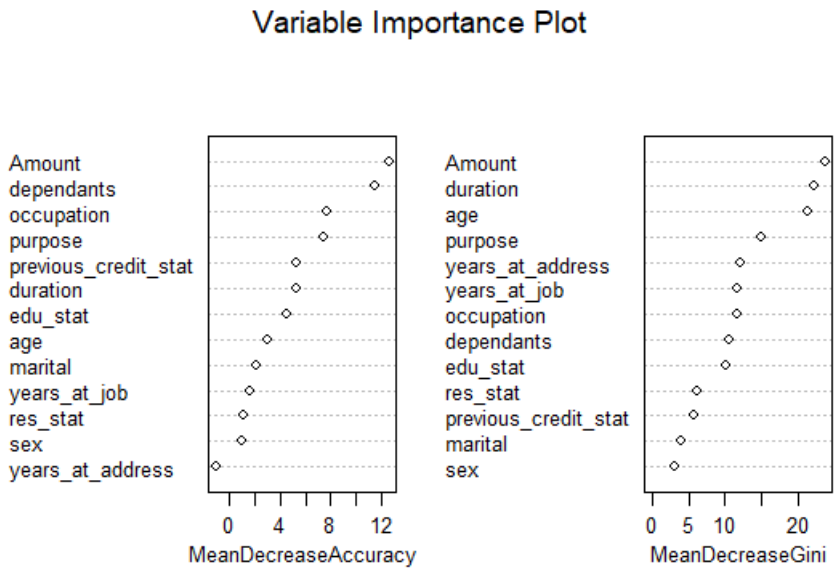

Figure 2: Determinants of credit default rates for variable of importance plot

6 Analysis and Results

The random forest algorithm developed a feature importance plot, as shown in Figure 2, to find the essential factors that drive credit default rates. The plot showed that the amount of loan had the greatest mean decrease accuracy score of roughly 14, followed by the number of dependents and finally the applicants' vocations. The mean decrease accuracy ratings reveal how much the random forest classifier's performance degrades when each feature is removed. As a result, indicators such as loan amount, number of dependents, employment, loan purpose, past credit status, loan term, and educational status might be regarded key factors in anticipating borrower default rates. Variables such as age, marital status, years at current work, residential status, gender, and year of current residence, on the other hand, had low mean decrease accuracy scores ranging from 0 to 4, allowing them to be eliminated from the model without substantially impacting its accuracy.

6.1 Data Resampling Techniques on Machine Learning Algorithms' Performance

This section presents the findings of the credit default prediction research on eight classification algorithms on four datasets, which include the original skewed dataset and balanced datasets utilizing under-sampling, SMOTE oversampling,

and SMOTE balanced approaches. The research used six main performance indicators to evaluate model performance, including accuracy (PCC), sensitivity, specificity, precision, F-measure, and area under the receiver operating characteristics curve (AUC). The major emphasis was on the AUC values.

6.1.1 Original Skewed Dataset

The AUC performance measure ranged from 0.583 to 0.834. Table 3 shows the classifiers' performance metrics after training and testing on the original imbalanced dataset. XGBoost had the greatest AUC score (0.834) and specificity (99.36%). The AUC value reflects the classifier's high ability to distinguish between defaulters and non-defaulters. However, the classifier's low sensitivity score (8.16%) indicates that it did badly in recognizing real affirmative cases (defaulters). Decision Tree (DT) outperformed XGBoost in terms of accuracy (PCC) at 80.58% and sensitivity (30.61%). Random forest has the second best AUC score (0.785). Artificial Neural Networks (ANNs) performed best in classifying the default group, with a sensitivity score of 40.82%. Despite excellent specificity scores (ranging from 73.25% to 99.36%), all classifiers trained with the skewed dataset showed low sensitivity (6.12% to 40.82%), indicating poor detection of real positive cases. Tables 3 through 5 demonstrate that sensitivity scores on balanced datasets improved compared to skewed datasets. This suggests that balancing strategies improve classifier performance.

Table 3: A Test Set Comparison of Classifiers using the Original Skewed Dataset

Classifier	PCC	Sensitivity	Specificity	Precision	F-Measure	AUC
RF	76.70%	6.12%	98.73%	60.00%	11.11%	0.785
DT	80.58%	30.61%	96.18%	71.43%	42.86%	0.707
LDA	77.67%	12.25%	98.09%	66.67%	20.70%	0.711
QDA	77.18%	14.29%	96.82%	58.33%	22.96%	0.635
LR	77.67%	20.41%	95.54%	58.82%	30.30%	0.750
SVMs	77.67%	8.16%	99.36%	80.00%	14.81%	0.707
ANNs	65.53%	40.82%	73.25%	32.26%	36.04%	0.583
XGBoost	77.67%	8.16%	99.36%	80.00%	14.81%	0.834

6.1.2 Balanced Dataset: Under-sampling Technique

Table 4 shows the performance of classifiers trained on the under-sampled dataset. The findings show that XGBoost outperformed other classifiers, with an accuracy (PCC) of 77.67%, F1-score of 58.69%, and AUC of 0.795. The perfor-

mance of classifiers trained on the under-sampled dataset was extremely competitive with that of the original skewed dataset, as indicated by their comparable PCC and AUC values. However, classifiers trained on under-sampled data had much higher sensitivity scores than their counterparts. Surprisingly, Artificial Neural Networks (ANNs) had the highest sensitivity score of 91.84%. These findings indicate that classifiers trained on undersampled data are often more successful at detecting defaulters than those trained on the other three datasets, while maintaining discriminating capacity. They also outperformed their rivals in terms of F-measure.

Table 4: A Test Set Comparison of classifiers using the Under-sampled Dataset

Classifier	PCC	Sensitivity	Specificity	Precision	F-Measure	AUC
RF	72.33%	69.39%	73.25%	44.70%	54.37%	0.776
DT	74.27%	59.18%	78.98%	46.77%	52.25%	0.761
LDA	73.79%	53.06%	80.25%	45.61%	49.05%	0.715
QDA	76.70%	30.61%	91.08%	51.72%	38.46%	0.627
LR	77.67%	57.14%	84.08%	52.83%	54.90%	0.763
SVMs	66.99%	48.98%	72.61%	35.82%	41.38%	0.688
ANNs	35.44%	91.84%	17.83%	25.86%	40.36%	0.594
XGBoost	77.67%	67.35%	80.89%	52.00%	58.69%	0.795

6.1.3 Balanced Dataset: Oversampling Technique

Finally, using the oversampling data as given in Table 5, XGBoost emerged as the best-performing classifier based on the AUC (0.801), followed by random forest with an AUC of 0.792. ANN had the best sensitivity score of 69.39%, although these classifiers only outperformed those trained on the skewed dataset.

Table 5: A Test Set Comparison of classifiers using the Oversampling Dataset

Classifier	PCC	Sensitivity	Specificity	Precision	F-Measure	AUC
RF	78.16%	26.53%	94.27%	59.09%	36.58%	0.792
DT	79.61%	34.69%	93.63%	62.96%	44.73%	0.779
LDA	73.30%	48.98%	80.89%	44.44%	46.56%	0.717
QDA	79.61%	32.65%	94.27%	64.00%	43.24%	0.634
LR	74.27%	59.18%	78.98%	46.77%	52.25%	0.734
SVMs	73.30%	40.87%	83.44%	43.48%	42.13%	0.658
ANNs	49.03%	69.39%	42.68%	27.42%	39.31%	0.572
XGBoost	76.70%	32.65%	90.45%	51.61%	40.00%	0.801

6.1.4 Balanced Dataset: SMOTE Oversampling Technique

Table 6 shows that Decision Trees (DT) performed best on the SMOTE dataset, with an AUC of 0.784 and a sensitivity of 53.06%. Random forest closely tracked the DT model, with an AUC of 0.780. Notably, sensitivity ratings on the SMOTE dataset improved relative to the original imbalanced data. Thus, the SMOTE oversampling approach produces much superior results. Oversampling outperforms under-sampling and classic oversampling for several classifiers, yielding greater scores in various assessment metrics.

Table 6: A Test Set Comparison of classifiers using the SMOTE Dataset

Classifier	PCC	Sensitivity	Specificity	Precision	F-Measure	AUC
RF	75.73%	40.82%	86.63%	48.78%	44.45%	0.780
DT	76.21%	53.06%	83.44%	50.00%	51.48%	0.784
LDA	78.16%	38.78%	90.45%	55.88%	45.79%	0.721
QDA	77.18%	26.53%	92.99%	54.17%	35.62%	0.654
LR	78.16%	51.02%	86.62%	54.35%	52.63%	0.739
SVMs	69.42%	34.69%	80.26%	47.50%	40.10%	0.632
ANNs	68.93%	38.78%	78.34%	35.85%	37.26%	0.574
XGBoost	77.67%	30.61%	92.36%	55.56%	39.47%	0.735

The study's results shed light on the relevance of data balancing strategies in model performance, as shown by the assessment of classifiers across various datasets. With the exception of the SMOTE oversampling dataset, XGBoost consistently beat the other classifiers on the other datasets in terms of AUC. This demonstrates XGBoost's superior discriminating performance when differentiating between defaulters and non-defaulters. This discovery is comparable to those of [Li and Chen, 2020, Dastile et al., 2020]. According to [Li and Chen, 2020, Dastile et al., 2020], ensemble algorithms such as random forest, XGBoost, and lightGBM outperformed individual algorithms including neural networks, logistic regression, decision tree, and support vector machines. In addition, random forest, decision tree, and logistic regression performed well and might be utilized as alternatives to XGBoost classifiers for developing credit scoring models. This finding is consistent with the research of Brown (2012). The authors of [Brown and Mues, 2012] found that random forest and gradient boosting are more robust to extreme class imbalance data sets. Their studies also revealed that standard scorecard-building approaches such as logistic regression and linear discriminant analysis performed comparably to sophisticated methods like as XGBoost.

Furthermore, Artificial Neural Networks (ANNs) shown promising results in effectively recognizing default cases and thereby minimizing type II mistakes,

as seen by high sensitivity ratings. Furthermore, classifiers trained on balanced datasets demonstrated higher sensitivity scores than those trained on the original skewed dataset, without sacrificing their capacity to distinguish between defaulters and non-defaulters. This demonstrates the value of data balancing strategies in increasing models' ability to anticipate minority class occurrences, namely defaulters. These results provide important insights for practitioners working with unbalanced data and contribute to the ongoing topic about employing machine learning methodologies to enhance data-driven decision-making processes.

More importantly, the improvements recorded in the balanced data techniques were found to be optimized when the SMOTE oversampling method is employed. The performance of the SMOTE techniques results in a higher performance metric values giving credence to the recommendation of SMOTE usage in both huge and relatively big imbalance dataset to help improve algorithms predictions. The decision tree algorithm performed creditably well in under this balancing method with other competitive algorithms such as the linear discriminant analysis and the linear regression.

7 Conclusion

This study presents three strategies to address class imbalance and apply them to various machine learning classification models. The study utilized the "A commercial bank Customer Transaction " dataset from the Ghana. The study utilized the unbalanced dataset to address and evaluate the issue. The study tested skewed original data, balanced the data by oversampling and under-sampling and SMOTE oversampling techniques on the dataset and evaluated classifiers using several criteria. SMOTE oversampling outperforms the skewed data, balanced under-sampling and oversampling for several classifiers, resulting in greater scores across assessment measures. In the future, we want to compare different gravitates of imbalance data on deep learning algorithms with resampling strategies.

References

B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen. Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the operational research society*, 54(6):627–635, 2003.

M. M. C. Batista G E, Prati R C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1), 2004.

T. Bellotti and J. Crook. Support vector machines for credit scoring and discovery of significant features. *Expert systems with applications*, 36(2):3302–3308, 2009.

A. Blanco, R. Pino-Mejías, J. Lara, and S. Rayo. Credit scoring models for the microfinance industry using neural networks: Evidence from peru. *Expert Systems with applications*, 40(1):356–364, 2013.

L. Breiman. Random forests. 2001.

I. Brown and C. Mues. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications*, 39 (3):3446–3453, 2012.

P. A. S. S. Chan P, Fan W. Distributed data mining in credit card fraud detection. *Intell. Syst. their Appl., IEEE*, 14(6):67–74, 1999.

H. L. B. K. Chawla N, Lazarevic A. Smoteboost: Improving prediction of the minority class in boosting. *Knowledge Discovery in Databases:PKDD 2003*, pages 107–119, 2003.

H. L. O. K. W. P. Chawla N. V., Bowyer K. W. Synthetic minority over-sampling technique(smote). *Journal of artificial intelligence research*, 16:321–357, 2002.

T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In editor, editor, *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016a.

T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016b.

G. E. Chen S, Haibo H. Ramoboost : Ranked minority oversampling in boosting. *IEEE Trans. Neural Network*, 21(10):1624–1642, 2010.

S. H. H. S. G. A. Cohen G, Hilario M. Learning from imbalanced data in surveillance of nosocomial infection. *Artificial intelligence in medicine*, 37(1):7–18, 2006.

X. Dastile, T. Celik, and M. Potsane. Statistical and machine learning models in credit scoring: A systematic literature survey. 2020.

Dina and Atiya. A comprehensive analysis of synthetic minority oversampling technique (smote) for handling class imbalance. *Information Science*, pages 32–64, 2019.

A. Dina and Kamalov. A theoretical distribution analysis of synthetic minority oversampling technique (smote) for imbalanced learning. *Machine Learning*, 2023.

T. A. Duman E, Ekinci Y. Comparing alternative classifiers for database marketing: The case of imbalanced datasets. *Expert Systems with Applications*, 39(1): 48–53, 2012.

A. A. F. Elreedy D. A comprehensive analysis of synthetic minority oversampling technique (smote) for handling class imbalance. *Information Sciences*, 505:32–64, 2019.

W. erbeke, K. Dejaeger, D. Martens, J. Hur, and B. Baesens. New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. 2012.

Z. J. C. P. Fan W, Stolfo S. Adacost: misclassification cost-sensitive boosting.

A. A. Fayed H. A novel template reduction approach for the-nearest neighbor method. *IEEE Trans. Neural Network*, 20(5):890, 2009.

H. F. C. N. V. Fernndez A., Garca S.

Y. Guo. Credit risk assessment of p2p lending platform towards big data based on bp neural network. 2019.

S. J. M. G. Y. H. B. G. Haixiang G, Yijing L. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220–239, 2017.

D. J. Hand and W. E. Henley. Statistical classification methods in consumer credit scoring: a review. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 160(3):523–541, 1997.

S. S. Japkowicz N. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449, 2002.

E. D. Kamalov F, Atiya A F. Partial resampling of imbalanced data. *arXiv preprint arXiv: 2207. 04631*, 2022.

M. Kaur, Pannu. A systematic review on imbalanced data challenges in machine learning; application and solutions. *ACM Computing Surveys*, 53(4): 1–36, 2019.

K. N. Kerdprasop K. Data preparation techniques for improving rare class prediction 2 intelligent methods for predicting. *Proceedings of the 13th WSEAS International Conference on mathematical methods*, pages 204–209, 2011.

A. C. K. Lematre G, Nogueira F. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *The Journal of Machine Learning Research*, 18(1):559–563, 2017.

Y. Li. Credit risk prediction based on machine learning methods. In editor, editor, *2019 14th International Conference on Computer Science Education (ICCSE)*, 2019.

Y. Li and W. Chen. A comparative performance assessment of ensemble learning for credit scoring. 2020.

Z. Z. Liu X, Wu J. Exploratory undersampling for class-imbalance learning. *IEEE Trans. Syst., Man, Cybernetics, Part B (Cybernetics)*, 39(2):539–550, 2009.

V. López, A. Fernández, S. García, V. Palade, and F. Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. 2013.

X. Ma, J. Sha, D. Wang, Y. Yu, Q. Yang, and X. Niu. Study on a prediction of p2p network loan default based on the machine learning lightgbm and xgboost algorithms according to different high dimensional data cleaning. *Electronic Commerce Research and Applications*, 31:24–39, 2018.

M. Malekipirbazari and V. Aksakalli. Risk assessment in social lending via random forests. 2015.

Z. I. Mani I. Knn approach to unbalanced data distributions: a case study involving information extraction. *Proceedings of workshop on learning from imbalanced datasets*, 126, 2003.

S. H. Mayabadi S. Two density-based sampling approaches for imbalanced and overlapping data. *Knowledge-Based Systems*, 241:108217, 2022.

B. G. E. Monard M. C. Learmng with skewed class distrihutions. *Advances in Logic, Artificial Intelligence, and Robotics: LAPTEC*.

M. Moniz and H. Monteiro. No free lunch in imbalanced learning. *Knowledge Based System*, 227:107222, 2021.

J. R. Quinlan. Induction of decision trees. 1986.

N. R. Rao R, Krishnan S. Data mining for improved cardiac care. *ACM SIGKDD Explorations Newsletter*, 8(1):3–10.

M. A. Roweida M, Jumanah R. Machine learning with oversampling and undersampling techniques: Overview study and experimental results. *11th International Conference on Information and Communication Systems*, 2020.

H. J. N. A. Seiffert C, Khoshgoftaar T. Rusboost: A hybrid approach to alleviating class imbalance. *IEEE Trans. Syst., Man, and Cybernatics*, 40(1):185–197, 2010.

K. M. S. Sun Y., Wong A. K. Classification of imbalanced data: A review.

W. A. W. Y. Sun Y, Kamel M. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40(12):3358–3378, 2007.

C. Wang, D. Han, Q. Liu, and S. Luo. A deep learning approach for credit scoring of peer-to-peer lending using attention mechanism lstm. 2018.

L. X. Z. N. C. H. Wang L, Han M. Review of classification methods on unbalanced data sets. *IEEE Access*, 9:64606–64628, 2021.

Wilson. Asymptotic properties of nearest neighbor rules using edited data. *Transactions on Systems, Man, and Cybernetics*, 3:408421, 1972.

Z. Z. Y. C. Z. Y. Z. Y. Yan Y, Jiang Y. Ldas: Local density-based adaptive sampling for imbalanced data classification. *Expert Systems with Applications*, 191: 116213, 2022.