Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 9s (2025)

Advanced Re-Sampling Techniques for Multi-Class Imbalanced Classification

K.V.Chandra Sekhar[1], Balaka Ramjee[2], Landa Naresh[3], Pasala Mahesh[4], Kinthada Jayaramu[5]

[1]Assistant. Professor, Department of CSE(AI&ML), Aditya Institute of Technology and Management, Tekkali-532201, India.

[2,3,4,5]UG Students, Department of Computer Science and Engineering (AI&ML), Aditya Institute of Technology and Management, Tekkali-532201, India.

Article History:

Received: 12-01-2025

Revised: 15-02-2025

Accepted: 01-03-2025

Abstract:

Imbalanced classification is a common problem in machine learning, where one class significantly outnumbers the others. This imbalance leads to biased model performance, where the classifier favors the majority class, resulting in poor detection of the minority class. Traditional machine learning algorithms assume a balanced distribution, making them ineffective in such scenarios. Various techniques, including resampling methods (such as oversampling and undersampling), cost-sensitive learning, and synthetic data generation, have been proposed to address this challenge. Effective handling of imbalanced data is crucial in applications like fraud detection, medical diagnosis, and anomaly detection, where minority class predictions hold high significance. This study explores different approaches to mitigate class imbalance and improve classification performance, ensuring better generalization and robustness in real-world scenarios.

Introduction: Model predictions are skewed by class imbalances, rendering accuracy metrics less meaningful as is often the case in healthcare and fraud detection. SMOTE and its derivatives are also an example of resampling techniques which creates synthetic data to balance the classes for better learning. Such as Borderline-SMOTE, ADASYN, SMOTEENN or SMOTETomek helped to improve the decision boundaries and noise reduction. These techniques facilitate creation of better models by addressing the issue where minority class isn't represented in feature space fairly.

Methodology: The GMM-SMOTE method addresses imbalanced datasets by utilizing Gaussian Mixture Model (GMM) for clustering and applying SMOTE to oversample minority data in high-density areas. This approach involves clustering data, selecting clusters with significant minority presence, and generating synthetic samples to ensure better balance. GMM enhances clustering by assigning probabilities to data points, while SMOTE focuses on producing samples in less populated regions, effectively reducing noise and improving class representation and model performance in imbalanced situations.

Results: The study evaluates GMM-SMOTE against various oversampling techniques, including KMeans-SMOTE, KMeans-ADASYN, and GMM-ADASYN, using datasets such as Breast Cancer, Crx, and Churn BigML. Performance metrics include accuracy, AUC-ROC score, and computational efficiency across classifiers like Random Forest, SVM, Logistic Regression, and Neural Networks. Results demonstrate that GMM-SMOTE enhances classification through balanced decision boundaries and shows efficiency in training time, making it advantageous for

managing imbalanced datasets.

Conclusions: The study assesses the effectiveness of GMM-SMOTE in enhancing minority

class representation and maintaining balanced decision boundaries compared to traditional oversampling methods like SMOTE and ADASYN. GMM-SMOTE generates more

meaningful synthetic samples and mitigates overfitting. Future research will focus on adaptive parameter tuning, integration with deep learning, and real-time applications, with additional exploration into its effects on multi-class imbalance and computational efficiency. Overall, GMM-SMOTE stands out as a valuable resampling method for improving classification performance in imbalanced datasets.

Keywords: KMeansSMOTE, KMeansSMOTE with GMM.

1. Introduction

The dataset suffers from the problem of class imbalance when one class in a dataset has significantly more samples than others, leading to biased model predictions and favoring the majority class in general, mostly neglecting the minority. For example, while using accuracy metrics for evaluation can sometimes be very misleading, accuracy measures may tell the story differently when the minority is in consideration. Real-world datasets suffer from the problem of class imbalance in almost every domain, such as healthcare and fraud detection. There, the minority class, such as fraud cases or disease-positive cases, has fewer samples than the majority class.

The imbalance in this case results in biased models, where the majority class is privileged, leading to low performance in most cases. This problem is more complicated in high-dimensional and complex datasets, where learning meaningful patterns for the minority class is much tougher, given the overlapping data points, and standard performance metrics such as accuracy cannot reflect the true effectiveness of the model.

Resampling solutions is a set of techniques to address the class imbalance problem that occurs in many datasets. Class imbalance occurs when one class has many more instances than the others, making the model biased toward predicting the majority class. Resampling solutions try to modify the dataset through oversampling to increase the instances of the minority class or undersampling to decrease the number of instances of the majority class to achieve better class distribution. Such modifications are likely to benefit the model since it would be trained without bias, thereby enhancing the capability of giving better predictions about both classes. Examples of resampling solutions include SMOTE, random oversampling, random undersampling, and cluster-based undersampling. These methods can result in good model performance but have to be used with caution because of problems such as overfitting or loss of important data.

SMOTE works better than Random Undersampling (RUS) and Random Oversampling (ROS) because it does not merely replicate the minority class data but creates new synthetic instances by interpolating between existing samples. This avoids overfitting, which may occur in ROS due to repeated instances, and retains more information than RUS, which discards data from the majority class.

SMOTE also has various variants, among which are: Basic SMOTE, it selects one or more of its nearest neighbors and generates synthetic samples. This gives new data points as similar to the original samples. Borderline-SMOTE focuses on synthesizing the minority class examples close to the decision boundary of majority versus minority classes. It produces more informative samples, and that improves the discrimination capability of the model. ADASYN is an extension of SMOTE that adapts to the difficulty level of classifying samples.It focuses on generating challenging examples, which can improve the model's ability to handle complex decision boundaries In this case, there is emphasis on generating difficult samples. That may enhance its ability to classify well in complex decision boundaries.

SMOTEENN is basically the combination of both techniques: that of SMOTE with ENN cleaning technique. The method employs ENN right after it produced synthetic samples, to eliminate the noisy instances possibly

coming from a minority or even majority class. This results in an enhancement of the dataset quality as misclassified or borderline cases are eliminated. In SMOTETomek, the techniques involve combining SMOTE with Tomek Links cleaning method. Apart from balancing the dataset, it also removes ambiguities because it removes those pairs that will probably be misclassified. Each variant increases the performance by tackling particular problems, and Borderline-SMOTE would perform better than Basic SMOTE when there are a lot of borderline instances in the dataset, whereas ADASYN could work better with large datasets where most samples are hard to classify. SMOTEENN and SMOTETomek are more applicable in noisy or ambiguous cases because they improve the generalization capability of the model.

In conclusion, SMOTE and its variants are mostly used to improvement of the performance of classifiers on imbalanced data, or better management of the minority class and more accurate overall models.

2. Literature Survey

Nitesh et al. presented SMOTE, which is an algorithm that tries to solve class imbalance in machine learning by synthesizing new samples for the minority class. It does not repeat minority samples, like traditional random over-sampling does, thus it is less likely to overfit. This makes it improve variability within the dataset, enhance generalization of the classifier, and substantially improve metrics like recall, precision, and F1-score for the minority class. The paper does report significant improvement in accuracy performance on several data sets compared with baseline methods, specially for decision tree classification scenarios. SMOTE focused mainly on handling imbalanced data sets in order to find minority classes better and applied in fraud detection, medical diagnostics, and fault detection domains, among others. The benefits are that the models improve in performance, decrease the chances of overfitting, and are flexible enough to support any machine learning algorithm. Its innovative aspect is generating diverse synthetic data, hence why the technique has gained popularity for being used on imbalanced datasets[1]. Haibo He et al. proposed an extension of SMOTE called ADASYN, which will adaptively generate synthetic samples for the minority class based on data distribution. Unlike SMOTE, ADASYN focuses on creating more synthetic samples for harder-to-learn instances, such as those near the class boundary while creating fewer synthetic samples for well-represented regions. This reduces bias in imbalanced datasets and shifts the decision boundary of the classifiers towards the minority

class dynamically. The advantage of ADASYN is mainly the adaptively refinement of the dataset where focus is made upon difficult regions. The overall performance on imbalanced datasets is improved because ADASYN improves the classifier's ability to classify the minority class instances and enhances generalization[2]. Yuchun Tang et al. worked in the direction to enhance Support Vector Machines (SVMs) in dealing with highly imbalanced datasets. Modifications to the traditional SVMs, cost-sensitive learning altering misclassification costs for the minority class, and the application of oversampling and undersampling to balance data also were proposed. An innovation is through the GSVM-RU, granularized version of an SVM, based on information loss occurring when data cleaned has lesser support vector count for increasing predictions speed to better model quality. They improve better performances over SVM approaches concerning the capacity of detection from a minority-class instances and exhibits high performance superiority compared to baselines on benchmark datasets and metrics using G-mean and AUC evaluation along with metrics precision/recall measures[3]. Hui Han et al. proposed Borderline-SMOTE, an advanced version of SMOTE to boost the performance of classifiers on imbalanced datasets. The key aim is to overcome the weakness of SMOTE, which is the generation of synthetic samples only for the minority class instances near the decision boundary, or the "borderline" points, which are harder to classify. This helps Borderline-SMOTE improve classifier accuracy by over-sampling selectively all borderline instances; the model gets a good capturing of the decision boundary between classes. They did a major alteration by border-instance identification within the minority class for generating synthetic data only, in contrast to the uniform sampling performed on all points of the minority. Borderline-SMOTE

advantages include improved model performance, especially in identifying minority class instances, increased recall and F-measure, and decreased overfitting, since it focuses the sampling process on the most critical instances near the decision boundary[4]. Felix Last et al. combined K-Means clustering with the SMOTE technique in order to actually improve over-sampling in imbalanced learning. The main focus here would be the proposal of a new oversampling method that incorporates clustering techniques for enhancing the generation of synthetic samples for the minority class. This will be helpful in dealing with the problem of overfitting as SMOTE and K-Means would decrease the possibility of producing redundant or too similar synthetic samples for the highly imbalanced dataset. The key modification is the integration of K-Means clustering, which clusters the minority class instances into clusters, followed by applying SMOTE within these clusters to produce synthetic data. This approach has better diversity in synthetic samples, reduces overfitting, and improves model performance, since oversampling is more representative of the underlying distribution of the minority class. It also could help to enhance the classification for the minority class instances without over-redundancy in synthetic data[5]. Mimi Mukherjee and Matloob Khushi proposed an extension of the traditional SMOTE technique, namely SMOTE-ENC, to deal with datasets having a combination of both nominal and continuous features. This paper is primarily on the limitations of SMOTE as the technique mainly works for continuous data, generating a method which could generate synthetic samples for datasets containing both types of attributes: nominal and continuous. The use of SMOTE-ENC aims to enhance classification performance in imbalanced datasets with categorical features by creating valid synthetic instances, which preserve the integrity of both types of features. The main alteration made is adapting SMOTE to nominal data by providing a

method of handling categorical features during the creation of synthetic samples. The advantages of SMOTE-ENC are that it can handle mixed datasets, keeps the distribution of nominal and continuous features, and improves classifier performance in terms of both minority class detection and overall accuracy without distorting the data structure.[6]. Amerah Alabrah also improved the performance of the CCF Detector on handling class imbalances and normalization of outliers via the Interquartile Range (IQR) method. The emphasis in this improvement is towards making the detector more efficient and effective for application on datasets which have classes of imbalance and existent outliers whose presence might significantly degrade the outcome. The IQR method helps eliminate outliers in the data and reduces their effect, making the data cleaner and more balanced in favor of the classifier. The major change in this paper was the integration of the IQR method to remove or normalize outliers before the applications of a CCF detector on the data, which improved the quality of the data used for training. The advantage of this approach is that it helps the classifier to work better with imbalanced data by reducing the impact of extreme values, and by improving overall accuracy, leading to a more reliable model in real-world applications where data is often messy and noisy.[7].

3. Methodology:

The approach outlined in this paper utilizes the Gaussian Mixture Model (GMM) algorithm together with SMOTE oversampling in order to rebalance highly imbalanced datasets. It does an effective job of preventing noise generation by oversampling only in safe regions, where the data is better structured. Special emphasis is laid on handling both between-class imbalance and within-class imbalance, tackling the small disjuncts problem by inflating sparse minority regions. This approach is readily applicable because of the ease and common availability of both SMOTE and GMM. It is particularly distinct from similar approaches not only because of its low complexity but also because of its efficient method of distributing synthetic samples according to the density of Gaussian components.

3.1 Algorithm

The GMM-SMOTE algorithm involves three steps: clustering, filtering, and oversampling. Clustering is performed in the first step where the input space is partitioned into k groups with the help of the Gaussian

Mixture Model (GMM). Filtering is the second step that chooses clusters to be oversampled, keeping the ones with a high percentage of minority class samples. It further assigns the number of synthetic samples to be created, giving a higher number of samples to those clusters where minority class samples are scattered. Lastly, during the oversampling process, SMOTE is invoked within each identified cluster to realize the desired minority and majority instances ratio.

Input:

Begin:

X (matrix of observations),

y (target vector),

n (number of samples to be generated),

k (number of Gaussian components found by GMM),

irt (imbalance ratio threshold),

knn (number of nearest neighbors considered by SMOTE),

de (exponent for density computation, defaults to the number of features in *X*)

// Step 1: Fit GMM to input space and filter components with more minority instances than majority instances. components ← GMM(X, k)

filteredComponents ← {} for c in components do

imbalanceRatio ← (majorityCount(c) + 1) / (minorityCount(c) + 1) if imbalanceRatio < irt then

filteredComponents ← filteredComponents U {c}

end

end

// Step 2: For each filtered component, compute the sampling weight based on its minority density. for f in filteredComponents do

averageMinorityDistance(f) ← mean (euclideanDistances(f)) densityFactor(f) ← minorityCount(f) / (averageMinorityDistance(f)^de + 1) sparsityFactor(f) ← 1 / densityFactor(f)

end

sparsitySum ← sum(sparsityFactor(f) for f in filteredComponents) for f in filteredComponents do

samplingWeight(f) ← sparsityFactor(f) / sparsitySum

end

// Step 3: Oversample each filtered component using SMOTE. The number of samples to be generated is computed using the sampling weight.

generatedSamples ← {}

for f in filteredComponents do

numberOfSamples ← n * samplingWeight(f)

generatedSamples ← generatedSamples U SMOTE(f, numberOfSamples, knn)

end

return generatedSamples endThe GMM is a probabilistic method of clustering data, under the condition that the data comes from a mixture of multiple Gaussian distributions. In contrast to k-means, which puts every data point into a single cluster, GMM puts probabilities on the data points reflecting how likely they are to belong to every Gaussian component.

Expectation-maximization (EM) algorithm is employed to iteratively maximize the likelihood of the observed data by estimating the parameters of the Gaussian distributions. This clustering technique is capable of flexible, elliptical cluster shapes and soft memberships, which are beneficial for datasets where clusters vary in density and shape.

All the hyperparameters of GMM, including the number of components (k) and covariance type, have an effect on the results of clustering. The choice of a proper value for k is very important, as it impacts the number of minority clusters that can be identified and oversampled in the filtering step.

Filtering Step: Choosing Clusters to Oversample Following clustering, the filtering process determines which clusters have a majority of minority class records and need to be oversampled. This process makes sure oversampling occurs in areas it is most beneficial to enhance class balance.

Clusters are analyzed by their imbalance ratio, which is calculated as:

Imbalance Ratio=minorityCount(c)+1/majorityCount(c)+1

To calculate how many synthetic samples to produce per cluster, a sampling weight is allocated to every filtered cluster. The weight depends on the minority class density in the cluster. Normalize sparsity values in order to find sampling weights in such a way that the sum of all weights equals 1.

Lower-density clusters (more sparsely populated minority points) get a greater sampling weight, so more synthetic samples are created within those clusters.

After the calculation of sampling weights, SMOTE (Synthetic Minority Over-sampling Technique) is used in each chosen cluster to create synthetic samples.

The number of synthetic samples for each cluster is calculated as: samplesToGenerate=samplingWeight(f)×n(2)

where,n is the number of new samples required.

The SMOTE algorithm generates synthetic data points as follows:

- Choose a random minority instance a in the cluster.
- Identify its k-nearest neighbors among other minority instances.
- Choose one neighbor b randomly.
- Generate a synthetic instance by interpolating between a and b: $x=a+\lambda(b-a)$

where λ is a random value between 0 and 1

- Repeat until the required number of synthetic samples are created.

4. Experimental setup :

The experiments are carried out on class-imbalanced datasets to compare the performance of GMMSMOTE in dealing with class imbalance and other methods for class imbalance handling. The dataset selection is on the basis of major factors including class distribution, number of attributes, size of the dataset, and applicability in real-life scenarios. Every dataset goes through a preprocessing pipeline for quality assurance prior to the use of GMMSMOTE. This encompasses dealing with missing values by applying methods such as mean/mode imputation or predictive imputation, based on the type of missing data. Numerical attributes are normalized by employing min-max scaling or standardization, based on the need of the classifier, in order to avoid feature

magnitude creating biases. Categorical features are converted through one-hot encoding or label encoding for machine learning model compatibility. Data with extreme imbalance is also subject to initial exploratory analysis where class distributions, feature correlations, and biases are explored to know the characteristics of the data prior to resampling.

5. Computation environment:

Experiments are conducted on a high-end computing environment using Python-based development. The prominent libraries are Scikit-learn, NumPy, Pandas, TensorFlow/PyTorch (in case deep models are involved), and Imbalanced-learn for resampling methods. The hardware configuration comprises a multi-core CPU, GPU support (in case deep models are utilized), and ample RAM for efficient execution.

6. Metrics:

In order to determine the performance of GMM-SMOTE, some evaluation measures are taken into consideration to quantify its effect on unbalanced datasets. One of the most important parameters is the value of k, which is the number of clusters created by the Gaussian Mixture Model (GMM). The value of k has a deep impact on the quality of generated synthetic samples. A smaller value of k could result in overly simplified clusters that do not pick up the intricacy of minority class distributions, while an increased value of k could produce overly localized clusters and thus restrict the model's generalization capability. Hence, an optimal value of k needs to be chosen so that the synthesized data is able to optimize model learning.

Another significant measure is accuracy, which indicates the overall accuracy of the classifier's predictions. In imbalanced datasets, however, accuracy is deceptive since a model may obtain high accuracy through simply predicting the majority class more frequently and ignoring the minority class. Because of this drawback, it is useful to examine other metrics that offer most balanced measure of the performance of classification.

The AUC-ROC score is especially helpful in measuring how well the model separates various classes. The higher the AUC-ROC, the better the trade-off between sensitivity (recall) and specificity, and thus it is a good indicator for measurement of the effect of GMM-SMOTE. By enhancing the minority class representation, GMM-SMOTE seeks to increase AUC-ROC scores so that the model not only prefers the majority class but learns to distinguish between both classes effectively. These metrics collectively offer a complete assessment of GMM-SMOTE's contribution towards improving classification performance on imbalanced datasets.

7. Results

The findings offer the performance of GMM-SMOTE in contrast to other oversampling methods, like KMeans- SMOTE, KMeans-ADASYN, and GMM-ADASYN, over various datasets. The comparison takes into account primary measures, like accuracy, AUC-ROC score, and K value, to gauge the performance of the methods in managing class imbalance.

The experiments are performed on various datasets like Breast Cancer, Crx, and Churn BigML with classifiers like Random Forest, SVM, Logistic Regression, and Neural Networks. The results are presented in tables as well as through AUC-ROC curves, emphasizing the effect of various resampling methods.

The comparison also involves a training time analysis to assess computational efficiency. The results show that GMM-SMOTE efficiently enhances classification performance with an even decision boundary, proving its practical benefit in imbalanced learning tasks.

Brest-Cancer Dataset

Algorithm	K Value	Accuracy	AUC-ROC	Geometric Mean	F1 Score	PR AUC Score
KMeans SMOTE	10	0.7241	0.6834	0.6807	0.6000	0.5983
GMM SMOTE	5	0.7414	0.6789	0.5695	0.4828	0.6519
KMeans ADASYN	10	0.6552	0.7471	0.6467	0.5652	0.6795
GMM ADASYN	5	0.7069	0.6088	0.4813	0.3704	0.5675

Crx Dataset

Algorithm	K Value	Accuracy	AUC-ROC	Geometric Mean	F1 Score	PR AUC Score
KMeans SMOTE	10	0.8551	0.9069	0.8552	0.8551	0.8457
GMM SMOTE	5	0.9270	0.8872	0.8005	0.7273	0.8037
KMeans ADASYN	10	0.8188	0.8925	0.8189	0.8201	0.8468
GMM ADASYN	5	0.8623	0.9074	0.8624	0.8613	0.8412

Churn Bigml Dataset

Algorithm	K Value	Accuracy	AUC-ROC	Geometric Mean	F1 Score	PR AUC Score
KMeans SMOTE	10	0.9045	0.8878	0.7829	0.6667	0.7829
GMM SMOTE	5	0.9270	0.8872	0.8005	0.7273	0.8037
KMeans ADASYN	10	0.9045	0.8878	0.7829	0.6667	0.7829
GMM ADASYN	5	0.9045	0.8878	0.7829	0.6667	0.7829

Smote Varients

Algorithm	Accuracy	AUC-ROC	Geometric Mean	F1 Score	PR AUC Score
KMeans SMOTE	0.82644	0.86732	0.8252	0.8142	0.8761
SMOTE	0.69421	0.75506	0.6946	0.6992	0.7338
SMOTEN	0.64462	0.74767	0.6418	0.6195	0.7084
SMOTENC	0.73553	0.82703	0.7361	0.7333	0.8063
SVMSMOTE	0.73553	0.81704	0.7350	0.7241	0.8123
borderlineSMOTE	0.71074	0.73727	0.7106	0.7009	0.7101

8. Discussion

The motivation behind this study is to address the limitations of existing oversampling techniques by leveraging Gaussian Mixture Models (GMM) for clustering-based synthetic data generation. Standard methods like SMOTE create synthetic samples by linear interpolation, which can sometimes introduce noisy or unrealistic samples, especially in complex feature spaces. GMM-SMOTE improves upon this by using probabilistic modeling to identify underrepresented clusters and generate synthetic data that aligns more naturally with the distribution of the minority class.

Through our experiments, we successfully demonstrated that:

- GMM-SMOTE enhances the classifier's ability to generalize by producing better synthetic samples in regions where the minority class is underrepresented.
- Unlike traditional SMOTE, which applies uniform sampling, GMM-SMOTE

adapts to the feature distribution, ensuring that oversampling is performed in relevant data regions.

- The performance improvement is particularly significant in highly imbalanced datasets, where simple resampling methods tend to be ineffective.

Potential Improvements:

- **Adaptive Selection of the Number of Clusters:** Instead of setting a fixed number of clusters (K in GMM), an automated selection method (such as the Bayesian Information Criterion (BIC) or the Akaike Information Criterion (AIC)) could improve the clustering quality.
- **Hybrid Approach with Deep Learning:** Combining GMM-SMOTE with GAN-based (Generative Adversarial Networks) synthetic data generation could further enhance the quality of synthetic samples.
- **Time Complexity Reduction:** GMM clustering can be computationally expensive for large datasets. Optimizations such as Mini-batch GMM or parallel computing could be explored.

9. Ablation Study:

The ablation study examines the effect of various hyperparameters on performance, focusing on how changes impact model effectiveness.

I. Number of Clusters (K in GMM) :

As the number of clusters increases, GMM-SMOTE captures finer variations in the data.

However, too many clusters may lead to over-segmentation, where the minority class is split too finely, reducing the effectiveness of synthetic sample generation.

Conversely, too few clusters result in poor data representation, leading to synthetic samples that do not generalize well.

II. Covariance Type in GMM :

Full covariance matrices allow clusters to take any shape but increase computational cost. Diagonal covariance matrices speed up training but may restrict cluster flexibility.

III. Effect of Oversampling Ratio :

Increasing the oversampling ratio beyond a certain point may cause overfitting, where the model memorizes synthetic samples rather than generalizing well.

Finding the optimal oversampling percentage is crucial to balance minority class representation and model generalization.

By systematically varying these hyperparameters and analyzing their impact, we gain deeper insights into the effectiveness of GMM-SMOTE and identify the best configuration for different types of datasets.

10. Conclusion:

The study evaluates GMM-SMOTE against other oversampling techniques, analyzing its impact

on multiple datasets and classifiers. Performance metrics such as accuracy, AUC-ROC, and K value demonstrate that GMM-SMOTE effectively enhances minority class representation while maintaining a balanced decision boundary. The results confirm its superiority over traditional SMOTE and ADASYN in generating meaningful synthetic samples and reducing overfitting. Future work can explore adaptive parameter tuning for GMM- SMOTE to optimize performance across diverse datasets. Additionally, integrating GMM-SMOTE with deep learning models and extending its application to real-time imbalanced data scenarios could further enhance its effectiveness. Evaluating its impact on multi-class imbalanced problems and refining computational efficiency are also promising directions. This paper presents GMM-SMOTE as a robust resampling technique that combines the benefits of Gaussian Mixture Models and SMOTE to improve classification performance on imbalanced datasets. Through extensive experiments, we demonstrate its effectiveness, highlighting its potential for future advancements in machine learning and imbalanced data handling.

Refrences

[1] Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

[2] Ipsum dolor sit amet consectetur adipiscing elit pellentesque. Orci eu lobortis elementum nibh. Faucibus a pellentesque sit amet porttitor.

[3] Egestas tellus rutrum tellus pellentesque eu tincidunt tortor. Sagittis orci a scelerisque purus semper eget. Vitae purus faucibus ornare suspendisse sed nisi lacus sed viverra.

[4] Augue interdum velit euismod in pellentesque massa placerat duis ultricies. Metus aliquam eleifend mi in nulla posuere sollicitudin aliquam ultrices.

[5] Velit laoreet id donec ultrices tincidunt arcu non sodales neque. Non curabitur gravida arcu ac tortor dignissim convallis aenean et.

[6] Euismod in pellentesque massa placerat. Morbi non arcu risus quis varius quam quisque.