OVERSAMPLING

THE ART OF CREATING BALANCE

OVERVIEW

- Class Imbalance

- Types
- Problems
- Methods to counter

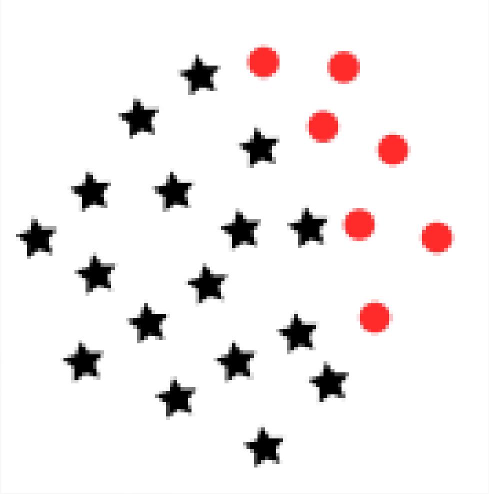
- Oversampling

- Types
- ROS
- SMOTE
- 7 SMOTE Variants
- Implementation

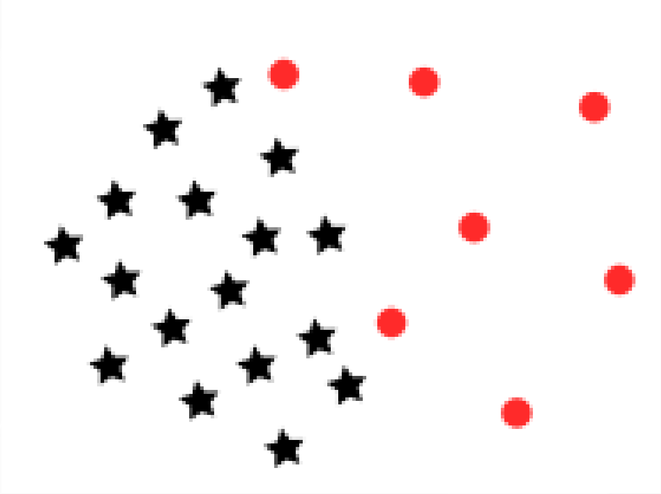
- Metrics

CLASS IMBALANCE

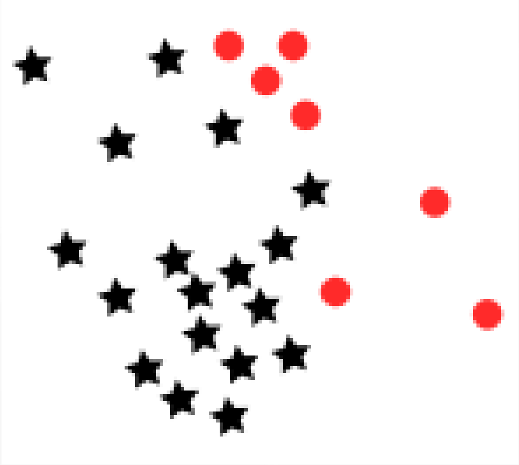
Class imbalance occurs when the distribution of classes in a dataset is skewed, with one class (the minority class) significantly outnumbered by another (the majority class). Example: Fraud Transitions among legitimate ones, rare diseases among common ailments, bots among genuine users.



Imbalance in the number of instances



Imbalance in the number and density of instances



Inter and Intra class Imbalance

PROBLEMS WITH CLASS IMBALANCE

- **Biased Models**: Machine learning algorithms tend to prioritize the majority class, leading to biased predictions.

- **Misleading Evaluation:** A model that predicts the majority class all the time may achieve high accuracy, but it is practically useless in most applications. Consequently we need to adopt new metrics like AUC-ROC, F1 score, etc for evaluation.

- **Poor Generalization**: Imbalanced datasets can result in models that do not generalize well to unseen data. Models may perform excellently on the majority class but fail to generalize to the minority class.

WAYS TO DEAL WITH IMBALANCED CLASSIFICATION

- **Data-Level Approach:**

- Resampling Techniques: Oversampling, Undersampling.

- Synthetic Data Generation: SMOTE Variants, GANs, Augmentation.

- **Algorithm-Level Approaches:**

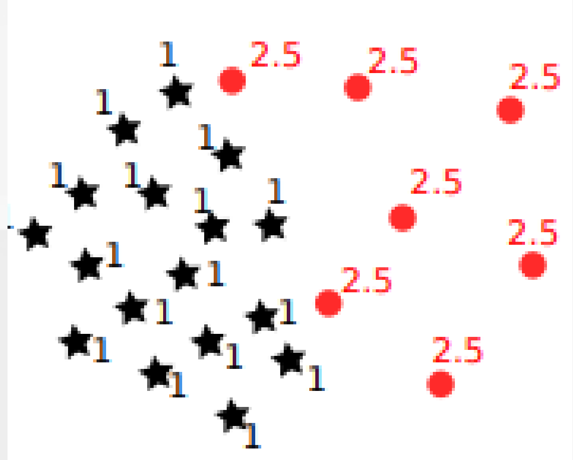
- Cost-Sensitive Learning: Assigning different misclassification weights to classes to make the algorithm more sensitive to the minority class.

- Ensemble Methods: Leveraging the power of ensemble techniques like Random Forest or Gradient Boosting to handle imbalance.

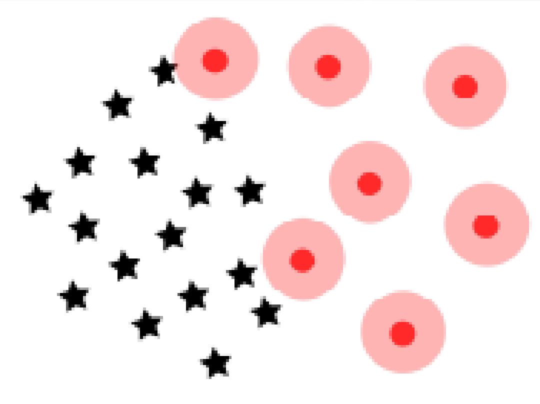
- Threshold Adjustment: Modifying the inference process by introducing a prior probability for each class.

- **One Class Learning:** Treat the minority sample as anomaly.

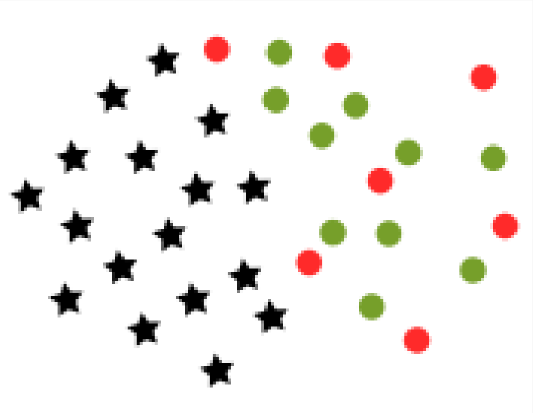
- **Hybrid Methods**



Reweighing



Minority points treated as objects with some volume to increase the minority region



Oversampling

OVERSAMPLING

Oversampling can be defined as the process of artificially increasing the number of instances in the minority class by creating synthetic data points. These new data points are generated based on the characteristics of the existing minority class instances.

- **Sampling by Cloning:** ROS, GNUS
- **Ordinary Synthetic Sampling:** SMOTE
- **With Filters:** SMOTE-ENN, CCR
- **Borderline:** BORDERLINE SMOTE
- **Using a Sampling Density:** ADASYN, PROWSYN
- **Using Kernel Density Estimation:** ROSE, KDE SMOTE

- **Using Dimensionality Reduction**
- **Use of Classifiers:** SMOTEBOOST
- **Use of Clustering:** KMEANS SMOTE
- **Component-Wise Sampling:** SMOTE-NC
- **Memetic Algorithms**
- **Application Specific:** GRAPHSMOTE

OVERSAMPLING TECHNIQUES

- Random Over Sampling
- SMOTE
- SMOTE TEMOK
- SMOTE ENN
- BorderLine Smote
- ADASYN
- KMeans Smote
- Smote-NC

RANDOM OVERSAMPLING (1997)

Random Oversampling is the simplest approach. It duplicates random instances of the minority class until balance is achieved.

Advantages:

- **Ease of Use:** Random Oversampling is simple to implement and a quick fix for imbalance.
- **No Complexity Added:** It does not introduce additional complexity to the model.

Limitations:

- **Risk of Overfitting:** Duplicating samples can lead to overfitting, where the model becomes too tailored to the training data.
- **Poor Generalization:** It might not provide a diverse representation of the minority.
- **Worse Evaluation:** It is prone to oversampling noisy data.

SMOTE (2002)

SMOTE stands for Synthetic Minority Oversampling Technique. It creates new synthetic samples based on the feature space similarity between existing instances of the minority class. It is particularly effective when we have sparse data.

SMOTE works by utilizing a k-nearest neighbor algorithm to create synthetic data:

- Identify the feature vector and its k nearest neighbor.
- Compute the distance between the two sample points.
- Multiply the distance with a random number between 0 and 1.
- Identify a new point on the line segment at the computed distance.
- Repeat the process for identified feature vectors.

$$n = x_i + r \cdot (x_j - x_i), \quad x_i, x_j \in \mathbb{R}^d, \quad r \in [0, 1]$$

Advantages of SMOTE

- **Data diversity:** By generating synthetic instances for the minority class, it increases the diversity of the data for the minority class.
- **Lower scope of overfitting:** Since the samples are diverse all the while preserving valuable information in the data, it has a lower scope of overfitting.
- **Versatility:** SMOTE can be applied in conjunction with different ML algorithms, depending on the nature of the embedding space.

Disadvantages of SMOTE

- **Class Disjunct Problem:** SMOTE doesn’t help in improving the intra class balance when the embedding space has sparse or irregularly distributed data.
- **Non-safe space Oversampling:** Like ROS, vanilla SMOTE can’t differentiate between safe samples, outliers and noisy samples, and oversamples all.
- **Impact on Decision Boundaries:** SMOTE can potentially lead the decision boundary to suboptimal positions.

SMOTE-TOMEK (2004)

SMOTE-TOMEK is a hybrid resampling technique that combines the oversampling using SMOTE and undersampling using TOMEK Links removal. It works on removing the noisy and borderline instances after oversampling.

- **SMOTE Oversampling:** First carry out traditional SMOTE on the minority samples
- **TOMEK Links:** TOMEK links are pairs of instances where one instance is a nearest neighbor of the other, but they belong to different classes. For two samples E_i and E_j to be Tomek links, they should be of different class and their distance d should fulfill the following criteria for any other sample E_l.

$$d(E_i, E_l) > d(E_i, E_j) \ or \ d(E_j, E_l) > d(E_i, E_j)$$

- **Filtering:** Such instances are considered ambiguous and potentially noisy. As an under-sampling method, only majority class samples are eliminated, and as a data cleaning method, examples of both classes are removed.

SMOTE-ENN (2004)

SMOTE-ENN is another hybrid resampling method that combines SMOTE for oversampling the minority class and the Edited Nearest Neighbors (ENN) algorithm for cleaning the dataset by removing noisy or borderline instances.

- **SMOTE Oversampling:** First carry out traditional SMOTE on the minority samples
- **Neighbor Classification:** Let for a minority sample E_i its three nearest neighbors are found. E_i is classified using its neighbors. E_i is given the same class as the most commonly occuring neighbor class (in a binary problem).
- **ENN Cleaning:** ENN removes any example whose class label differs from the class of at least two of its three nearest neighbors. If E_i belongs to the majority class, then it is removed. If E_i belongs to the minority class, then the nearest neighbors that belong to the majority class are removed.

.

ENN tends to remove more examples than the Tomek links does, and thus provides more in depth data cleaning.

BORDERLINE SMOTE (2005)

Borderline Smote performs SMOTE on the minority samples at or near the class boundary, which are more prone to be misclassified, and are thus more informative compared to the ones far from the boundary.

- **Data Separation:** Identify the minority class feature vector and find its k nearest neighbors.
- **Identify Noisy Points:** If all m neighbors belongs to majority class, mark the point as noise.
- **Find Borderline Points:** If the nearest neighbor from majority class is m' such that: $m/2 \leq m' < m$, mark them as borderline.
- **Generate Synthetic Samples** for the boundary points using SMOTE.

BORDERLINE SMOTE2: Also generate synthetic sample for each borderline minority sample with its nearest majority neighbors. But the interpolation factor is chosen between 0 and 0.5, thereby generating synthetic samples closer to the minority class.

ADASYN (2008)

ADASYN (Adaptive Synthetic Sampling Approach) adapts to the data distribution, focusing on the harder-to-learn examples by using density distribution as a criterion to decide the number of synthetic samples that need to be generated for each minority sample.

- Calculate the number of synthetic data examples needed (G) based on the desired balance level (β).
- For each minority class example (x_i), find the K nearest neighbors and calculate the normalized density distribution (r_i) such that:

$$r_i = \frac{\Delta_i}{K}, \quad i = 1, \ldots, m; \;\; r_i \in [0, 1]$$

$$\hat{r}_i = \frac{r_i}{\sum_{i=1}^{ms} r_i}$$

where Δ_i is the number of majority class samples in the K nearest neighbors of x_i.

- Calculate the number of synthetic data examples needed for xi (gi) based on the normalized ratio (r) and the total number of synthetic data examples (G).

$$g_i = G \times \hat{r_i}$$

- Generate new samples based on the weight g_i.

KMEANS SMOTE (2018)

K-Means SMOTE aids classification by generating minority class samples in safe and crucial areas of the input space. The method avoids the generation of noise and effectively overcomes imbalances between and within classes.

- **Clustering:** Cluster the entire data using the k-means clustering algorithm.
- **Filtering:** Select clusters that have a high proportions of minority class samples (>50%), less susceptible to noise generation..
- **Distribution:** Assign more synthetic samples to clusters where minority class samples are sparsely distributed.

$$\text{density}(f) = \frac{\text{minorityCount}(f)}{\text{averageMinorityDistance}(f)}$$

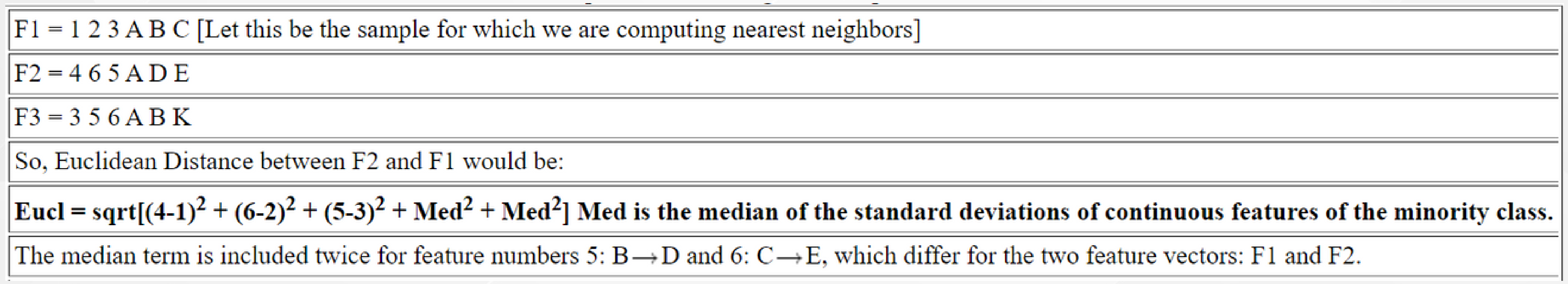
$$\text{sparsity}(f) = \frac{1}{\text{density}(f)}$$

- **SMOTE:** Generate synthetic samples

SMOTE-NC (2002)

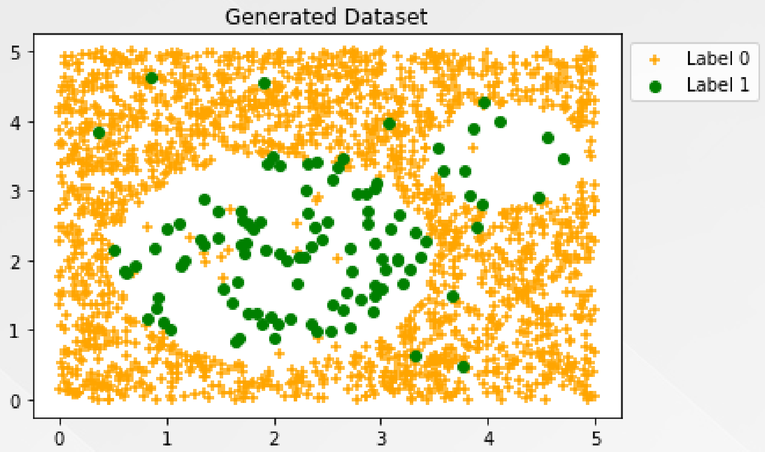
SMOTE-NC (Nominal and Continuous Features) is designed to handle datasets with a mix of both nominal (categorical) and continuous (numeric) features.

- **Feature Space Separation:** SMOTE-NC begins by separating the dataset's feature space into two parts: one for continuous features and the other for nominal features.
- **Calculating median:** Compute the median of standard deviations for continuous features in the minority class.
- **Euclidean distance Modification:** When identifying k-nearest neighbors for a minority sample, include the median in the Euclidean distance computation for instances with differing nominal features.
- **SMOTE Sampling:** Create synthetic minority class samples by interpolating continuous features using SMOTE.
- **Nominal Feature Sampling:** Assign the nominal feature of the synthetic sample the value that occurs most frequently among the k-nearest neighbors.

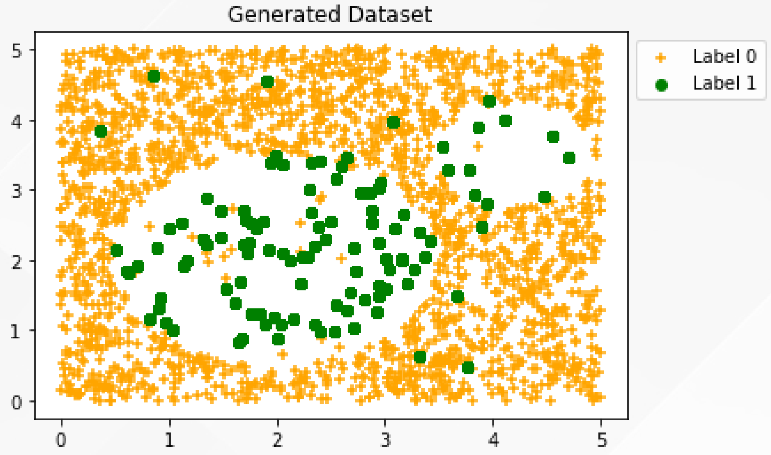


F1 = 1 2 3 A B C [Let this be the sample for which we are computing nearest neighbors]
F2 = 4 6 5 A D E
F3 = 3 5 6 A B K
So, Euclidean Distance between F2 and F1 would be:
Eucl = sqrt[$(4\text{-}1)^2 + (6\text{-}2)^2 + (5\text{-}3)^2 + \text{Med}^2 + \text{Med}^2$] Med is the median of the standard deviations of continuous features of the minority class.
The median term is included twice for feature numbers 5: B→D and 6: C→E, which differ for the two feature vectors: F1 and F2.

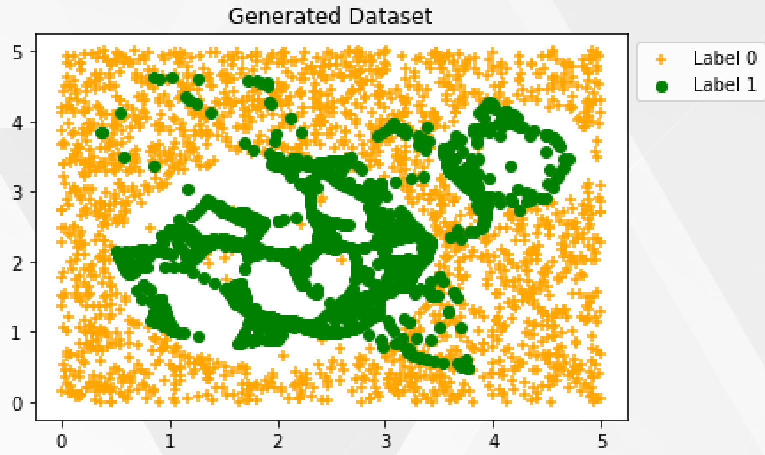
IMPLEMENTATION



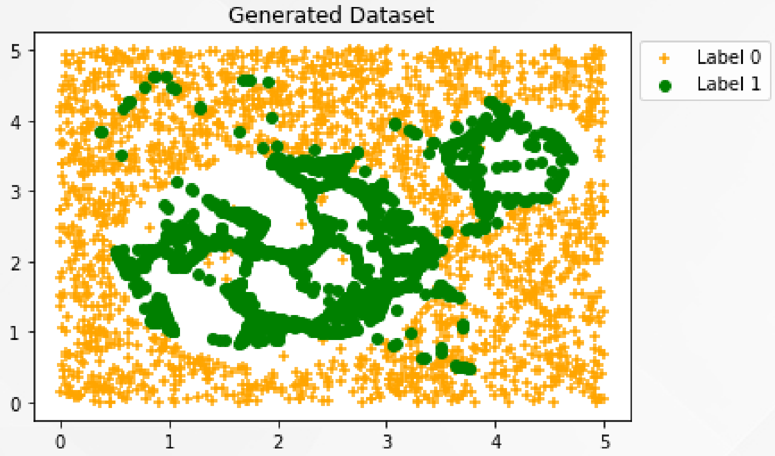
Original Data



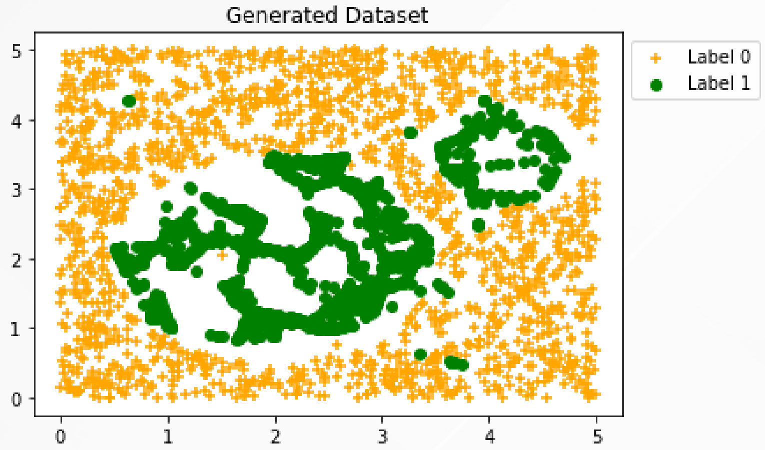
ROS



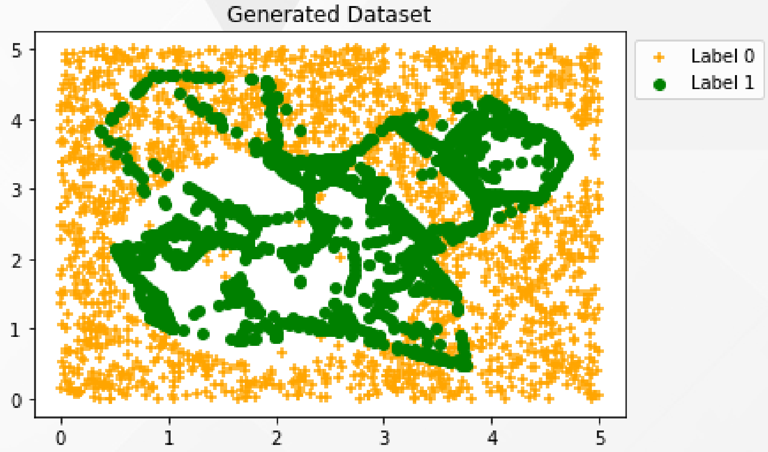
SMOTE



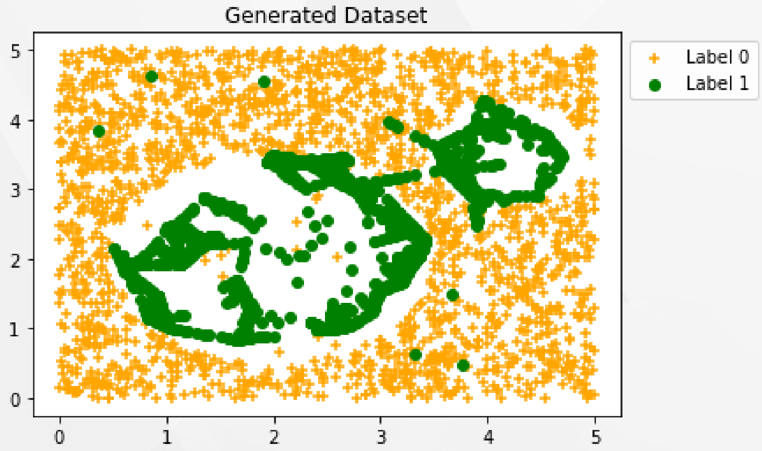
SMOTE-TOMEK



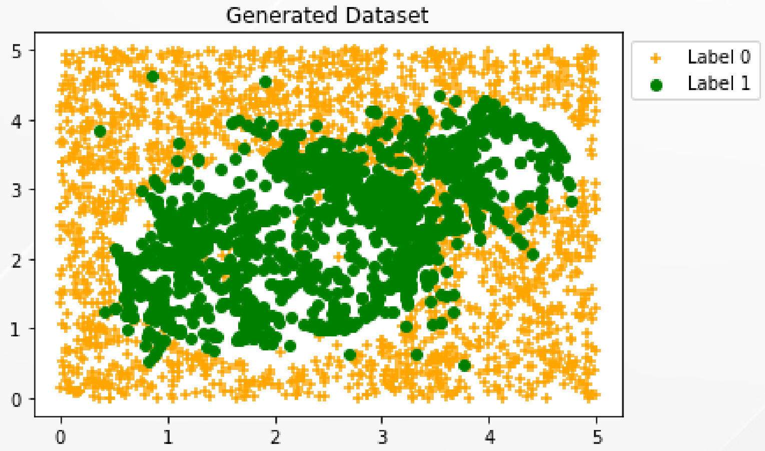
SMOTE-ENN



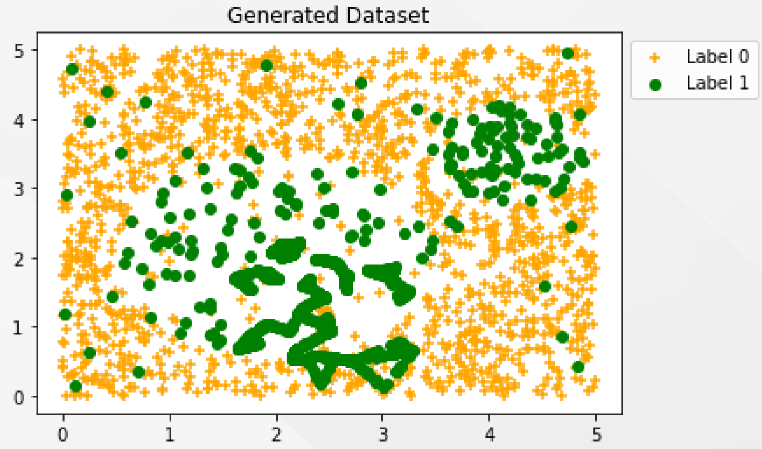
ADASYN



BLSM-1



BLSM-2



KMEANS SMOTE

OTHER VARIANTS

- **SMOTEBOOST (2003):** It combines SMOTE and boosting to improve the performance of classifiers. It generates synthetic samples for the minority samples before sending it to the next weak learner thereby further boosting the classifier's performance on top of weighting the misclassified samples.
- **Polynom-fit-SMOTE (2008):** it uses polynomial fitting methods to generate artificial data. It proposes 4 new approaches including star topology, bus topology, polynomial curve topology and mesh topology.
- **SMOBD (2011):** Ignores noise and oversamples on minority samples based on a density criteria decided by the radius of the hypersphere containing k nearest neighbor (e) and total samples (N) contained within it, as weighted sum between the two.
- **ProWSyn (2013):** ProWSyn generates effective weight values for the minority data samples based on sample’s proximity information, i.e., distance from boundary.
- **G-SMOTE (2014):** Find the difference vector between the chosen sample and the k neighbors. Find a new vector taken as the linear combination of all these tangent vectors with weights decided by random numbers. The synthetic sample is interpolated along this line.
- **KDE-based SMOTE (2014):** It uses a kernel density estimator to find the PDF for the minority samples to generate synthetic samples from. The generation might be lesser from near the boundary points where the density is less.
- **GAN-based oversampling:** It employs an encoder (generator) and a discriminator. The encoder learns to generate synthetic samples for the minority class while the discriminator distinguishes between real and synthetic samples, creating a competitive process.
- **GNUS (2016):** Basic Random Up-sampling with a small gaussian noise addition as augmentation.
- **Gaussian-SMOTE (2017):** To prevent synthetic samples from being generated on the same line between the frequently selected samples, a gaussian distribution is used to sample the random number r, having mean r and some SD. thereby expanding the region of synthetic data generation.
- **KNORR (2021):** Decides the safe minority sample based on their sorted order of increasing distance with the k-th nearest neighbor.
- **GRAPHSMOTE (2021):** Used in Homogenous Graphs.
- **HeteroGSMOTE (2024?)**

EVALUATION METRICS FOR IMBALANCED DATA

- **Precision (P)**: Measures the accuracy of positive predictions. Higher precision indicates fewer false positives.

$$P = \frac{TP}{TP + FP}$$

- **Recall (Sensitivity) (R)**: Measures the ability to capture all positive instances. Higher recall means fewer false negatives.

$$R = \frac{TP}{TP + FN}$$

- **F1-Score:** Harmonic mean of P and R. It provides a single metric that considers both false positives and false negatives.

$$F1 = \frac{2PR}{P + R}$$

- **ROC-AUC:** The Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC) measure the model's ability to distinguish between classes. A high AUC indicates better class separation. Y axis holds TPR and X axis holds FPR.

- **Confusion Matrix:** It is a tabular representation of the model's performance, showing true positives, true negatives, false positives, and false negatives.

REFERENCES

- Kovács, György. "An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets." Applied Soft Computing 83 (2019): 105662.
- Chawla, Nitesh V., Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research 16 (2002): 321-357.
- Batista, Gustavo EAPA, Ronaldo C. Prati, and Maria Carolina Monard. "A study of the behavior of several methods for balancing machine learning training data." ACM SIGKDD explorations newsletter 6, no. 1 (2004): 20-29.
- Han, Hui, Wen-Yuan Wang, and Bing-Huan Mao. "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning." In International conference on intelligent computing, pp. 878-887. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.
- He, Haibo, Yang Bai, Edwardo A. Garcia, and Shutao Li. "ADASYN: Adaptive synthetic sampling approach for imbalanced learning." In 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence), pp. 1322-1328. Ieee, 2008.
- Last, F., G. Douzas, and F. Bacao. "Oversampling for imbalanced learning based on k-means and smote. arXiv 2017." arXiv preprint arXiv:1711.00837 2.
- Mukherjee, Mimi, and Matloob Khushi. "SMOTE-ENC: A novel SMOTE-based method to generate synthetic data for nominal and continuous features." Applied System Innovation 4, no. 1 (2021): 18.
- Zhao, Tianxiang, Xiang Zhang, and Suhang Wang. "Graphsmote: Imbalanced node classification on graphs with graph neural networks." In Proceedings of the 14th ACM international conference on web search and data mining, pp. 833-841. 2021.
- 7 Over Sampling techniques to handle Imbalanced Data, Towards Data Science. https://towardsdatascience.com/7-over-sampling-techniques-to-handle-imbalanced-data-ec51c8db349f
- 7 SMOTE Variations for Oversampling, KD Nuggets. https://www.kdnuggets.com/2023/01/7-smote-variations-oversampling.html