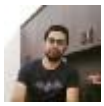


Satyam Kumar

May 3, 2021
·
5 min read

Stop using SMOTE to handle all your Imbalanced Data

Combination of Oversampling and Undersampling techniques

In classification tasks, one may encounter a situation where the target class label is not equally distributed. Such a dataset can be termed Imbalanced data. Imbalance in data can be a blocker to train a data science model. In case of imbalance class problems, the model is trained mainly on the majority class and the model becomes biased towards the majority class prediction.

Hence handling of imbalance class is essential before proceeding to the modeling pipeline. There are various class balancing techniques that solve the problem of class imbalance by either generating a new sampling of the minority class or removing some majority class samples. Handling class balancing techniques can be broadly classified into two categories:

- **Over-sampling techniques:** Oversampling techniques refer to create artificial minority class points. Some oversampling techniques are Random Over Sampling, ADASYN, SMOTE, etc.
- **Under-sampling techniques:** Undersampling techniques refer to remove majority class points. Some oversampling techniques are ENN, Random Under Sampling, TomekLinks, etc.

Read the below-mentioned article to know 7 oversampling techniques to handle the problem of class imbalance.

A disadvantage of using undersampling techniques is that we are losing out a lot of majority class data points in order to balance the class. Oversampling techniques cover this disadvantage but creating multiple samples within the minority class may result in overfitting of the model.

SMOTE is one of the popular and famous oversampling techniques among the data scientist community that create artificial minority data points within the cluster of minority class samples. The idea is to combine the oversampling and undersampling techniques and together it may be considered as another sampling technique to handle imbalanced class data.

Combination of Oversampling and Undersampling techniques:

SMOTE is one of the famous oversampling techniques and is very effective in handling class imbalance. The idea is to combine SMOTE with some undersampling techniques (ENN, Tomek) to increase the effectiveness of handling the imbalanced class.

Two examples of the combination of SMOTE and undersampling techniques are:

- **SMOTE with ENN**
- **SMOTE with Tomek**

Before proceeding to the combination of SMOTE with undersampling techniques, let's discuss what is SMOTE and how does it work under the hood.

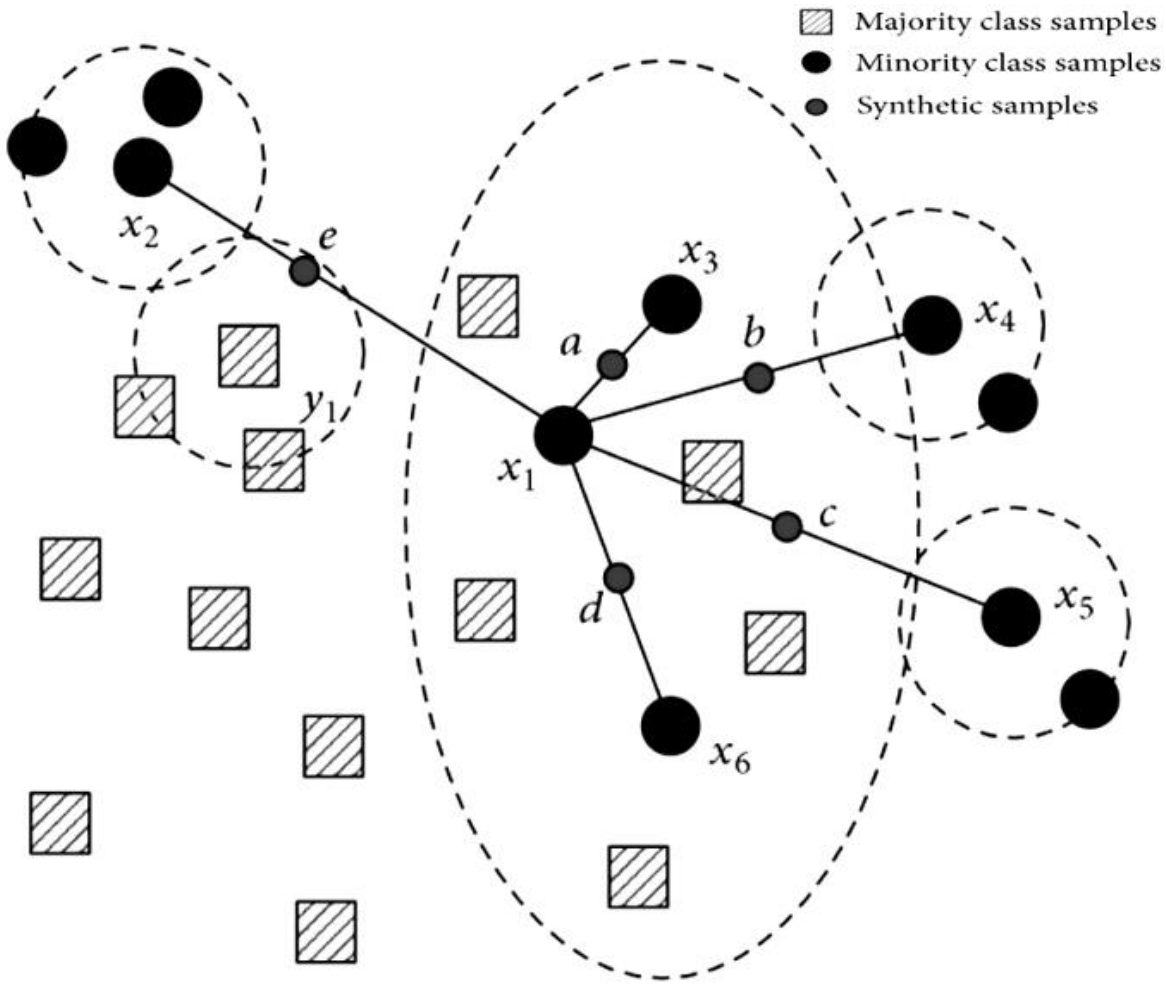
What is SMOTE?

SMOTE stands for Synthetic Minority Oversampling Technique, is an oversampling technique that creates synthetic minority class data points to balance the dataset.

SMOTE works using a k-nearest neighbour algorithm to create synthetic data points. The steps of SMOTE algorithm is:

1. Identify the minority class vector.
2. Decide the number of nearest numbers (k), to consider.
3. Compute a line between the minority data points and any of its neighbours and place a synthetic point.

4. Repeat step 3 for all minority data points and their k neighbours, till the data is balanced.



(Image by Author), SMOTE

The combination of SMOTE and some undersampling techniques are proven effective and together can be considered as a new sampling technique.

Combination of SMOTE with Tomek Links:

Tomek Links is an undersampling heuristic approach that identifies all the pairs of data points that are nearest to each other but belong to different classes, and these pairs (suppose a and b) are termed as Tomek links. Tomek Links follows these conditions:

- a and b are nearest neighbours of each other
- a and b belong to two different classes

Heuristically, these Tomek links points (a, b) are present on the boundary of separation of the two classes. So removing the majority class of Tomek links points increases the class separation, and also reduces the number of majority class samples along the boundary of the majority cluster.

Idea:

SMOTE is an oversampling technique and creates new minority class synthetic samples, and Tomek Links is an undersampling technique.

For an imbalanced dataset, first SMOTE is applied to create new synthetic minority samples to get a balanced distribution. Further, Tomek Links is used in removing the samples close to the boundary of the two classes, to increase the separation between the two classes.

Implementation:

The Imblearn package comes with the implementation of the combination of SMOTE and Tomek Links.

You can install the library from PyPl using **pip install imblearn**

```
from imblearn.combine import SMOTETomeksmt = SMOTETomek(random_state=42)
X, y = smt.fit_sample(X, y)
```

Combination of SMOTE with ENN:

ENN (Edited Nearest Neighbour) is an undersampling technique that removes the instances of majority class on the border or boundary whose predictions made by the KNN algorithm are different from the other majority class points.

Similar to SMOTETomek, first SMOTE is applied to create synthetic data points of minority class samples, then using ENN the data points on the border or boundary are removed to increase the separation of the two classes.

Implementation:

A combination of SMOTE with ENN algorithm also comes in the imblearn package.

```
from imblearn.combine import SMOTEENNsmt = SMOTEENN(random_state=42)
X, y = smt.fit_sample(X, y)
```

Conclusion:

The combination of SMOTE and undersampling techniques (ENN and Tomek Links) are proven effective. SMOTE is basically used to create synthetic class samples of minority class to balance the distribution then undersampling technique (ENN or Tomek Links) is used for cleaning irrelevant points in the boundary of the two classes to increase the separation between the two classes.

Imblean package comes with the implementation of SMOTETomek and SMOTEENN. There is a thumb rule about which works the best. One can write manual Python code to combine one oversampling technique proceeded by an undersampling technique to get the best results.

You can also read the below-mentioned article to know 7 oversampling techniques to handle class imbalance.

References:

[1] Imbalance Learn API documentation: http://glemaitre.github.io/imbalanced-learn/api.html