SMOTE AND NEARMISS

What is an Imbalanced Dataset?

Imagine, you have two categories in your dataset to predict - A and B. When A is higher than B or vice versa, you have a problem of imbalanced dataset.

So how is this a problem?

Imagine in a dataset of 100 rows, Category-A is containing 90 records and Category-B is containing 10 records. You run a machine learning model and end up with 90% accuracy. You were excited until you checked the confusion matrix.

	Category-A	Category-B
Category-A	90	0
Category-B	10	0

Here, Category-B is completely classified as Category-A and the model got away with an accuracy of 90%.

Also such models fail to perform well on unseen data because they are biased models as training data didn’t had an equal distribution of the output classes.

Smote (Synthetic Minority Over-Sampling Technique)

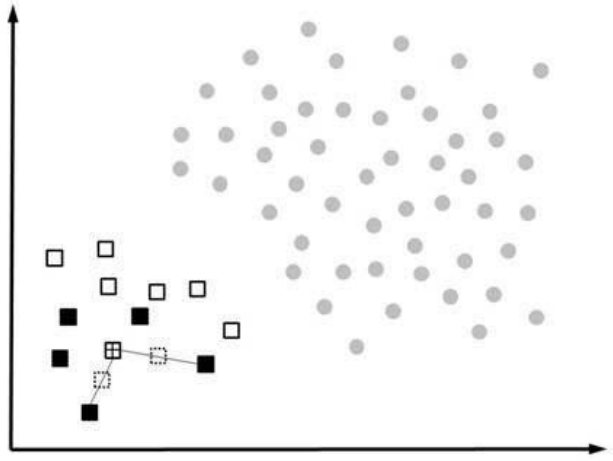
Smote is an over-sampling method. It works by creating synthetic observations based upon the existing minority observations.

SMOTE does this by selecting records and altering that record one column at a time by a random amount within the difference to the neighboring records.

How it works

- It first selects a minority class instance ‘a’ at random and finds its k nearest minority class neighbors.

- The synthetic instance is then created by choosing one of the k nearest neighbors ‘b’ at random and connecting ‘a’ and ‘b’ to form a line segment in the feature space. The synthetic instances are generated as a convex combination of the two chosen instances ‘a’ and ‘b’.



Near Miss

Near Miss is an under-sampling technique. Instead of resampling the minority class this will make the majority class equal to minority class. Near Miss aims to balance class distribution by randomly eliminating majority class examples. When two points belonging to different classes are very close to each other in the distribution, this algorithm eliminates the datapoint of the larger class thereby trying to balance the distribution.

How it works

- This method first finds the distances between all instances of the majority class and the instances of the minority class.

- Then, n instances of the majority class that have the smallest distances to those in the minority class are selected.

- If there are k instances in the minority class, the nearest method will result in k*n instances of the majority class.

For finding n closest instances in the majority class, there are several variations of Near Miss Algorithm.

Version 1: In the first version, the data is balanced by calculating the average minimum distance between the larger distribution and three closest smaller distributions.

Version 2: Here, the data is balanced by calculating the average minimum distance between the larger distribution and three furthest smaller distributions.

Version 3: Here, the smaller class instances are considered and m neighbors are stored. Then the distance between this and the larger distribution is taken and the largest distance is eliminated.

SMOTEENN

It is a combination of undersampling and oversampling. On majority side it would undersample and on minorty side it would oversample.