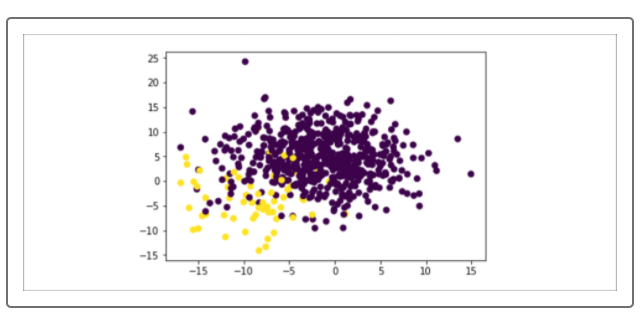
SMOTEENN, an approach to resampling that combines aspects of both oversampling and undersampling.

As previously discussed, a downside of oversampling with SMOTE is its reliance on the immediate neighbors of a data point. Because the algorithm doesn't see the overall distribution of data, the new data points it creates can be heavily influenced by outliers. This can lead to noisy data. With downsampling, the downsides are that it involves loss of data and is not an option when the dataset is small. One way to deal with these challenges is to use a sampling strategy that is a combination of oversampling and undersampling.

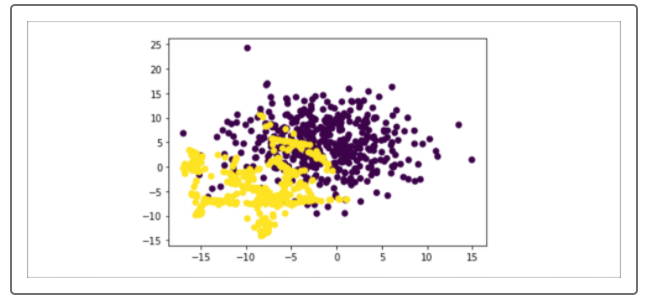
SMOTEENN combines the SMOTE and Edited Nearest Neighbors (ENN) algorithms. SMOTEENN is a two-step process:

1. Oversample the minority class with SMOTE.
2. Clean the resulting data with an undersampling strategy. If the two nearest neighbors of a data point belong to two different classes, that data point is dropped.

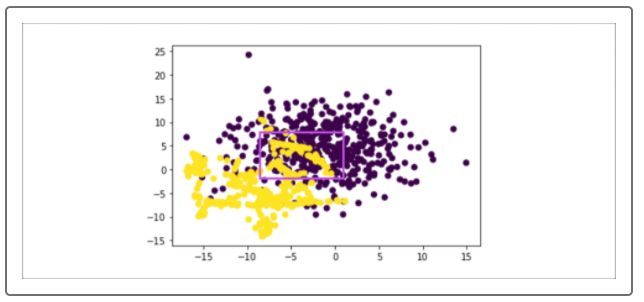
The series of images below help illustrate the SMOTEENN technique. The first image represents a synthetically generated dataset (using the make\_blobs module) and shows two classes: purple as the majority class and yellow as the minority class.



In the following image, the minority class is oversampled with SMOTE.



Note that the two classes significantly overlap, as the box indicates below. This overlap makes classification difficult.



In the next image, SMOTEENN is applied, instead of SMOTE. As with SMOTE, the minority class is oversampled; however, an undersampling step is added, removing some of each class's outliers from the dataset. The result is that the two classes are separated more cleanly.

