Class Imbalance Problem

Classification algorithms works best when the number of instances of each classes are roughly equal. When the number of instances of one class far exceeds the other, problems arise.
In imbalanced cases standard classifier algorithms have a bias towards classes which have large number of instances. They tend to only predict the majority class data. The features of the minority class are treated as noise and are often ignored. Thus, there is a high probability of misclassification of the minority class as compared to the majority class.
There are different ways of addressing class imbalance problems like:

- Synthesis of new minority class instances
- Over-sampling of minority class
- Under-sampling of majority class
- tweak the cost function to make misclassification of minority instances more important than misclassification of majority instances

We will be discussing the first three techniques mentioned above using different algorithms.

Algorithms to be Explored:

- SMOTE +Tomek Links
- Easy Ensemble
- Balanced Cascade
- SMOTE Boost
- SVM SMOTE
- ADASYN

SMOTE + Tomek Links

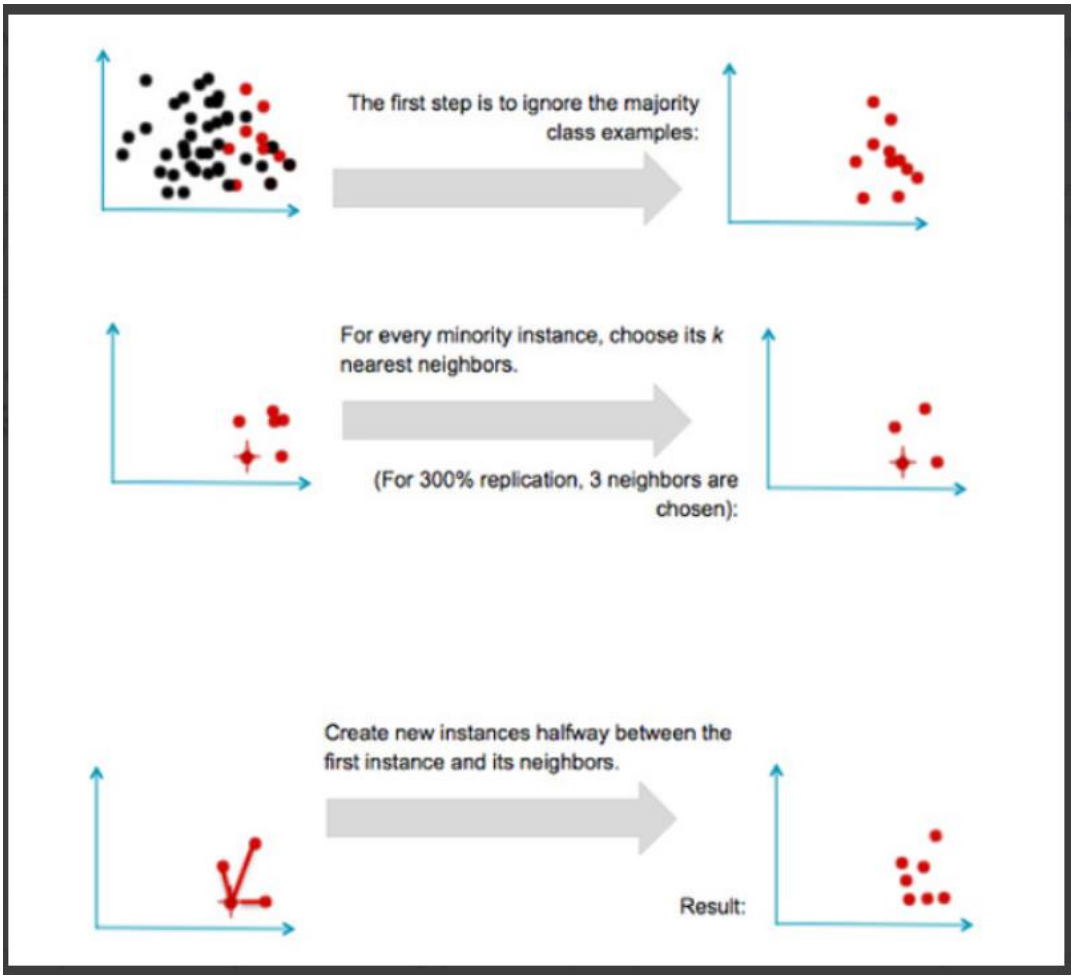
Synthetic Minority Over-sampling Technique(SMOTE) uses over-sampling approach in which the minority class is over-sampled by creating "synthetic" examples based upon the existing minority observations rather than by over-sampling with replacement.

The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen.

Synthetic samples are generated in the following way: Take the difference between the feature vector (sample) under consideration and its nearest neighbor. Multiply this difference by a random number between 0 and 1, and add it to the feature vector under consideration. This

causes the selection of a random point along the line segment between two specific features. This approach effectively forces the decision region of the minority class to become more general.

Following is a visual description for a dataset with two features with minority class in red and majority class in black:



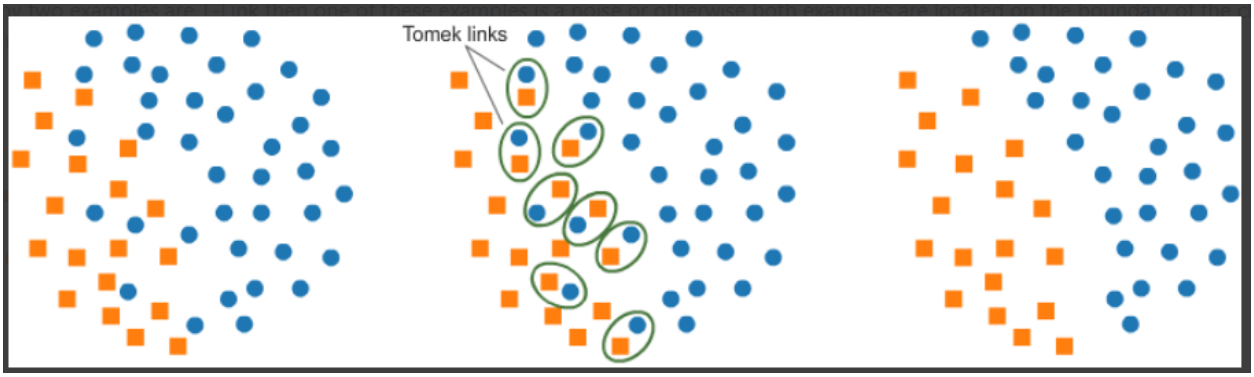
Since SMOTE only considers minority class while creating synthetic data points, problem arises because of overlap between classes.

Tomek Links

Removes unwanted overlap between classes where majority class links are removed until all minimally distanced nearest neighbor pairs are of the same class.

Let x be an instance of majority class(blue) and y an instance of minority class(orange).

Let d(x, y) be the distance between x and y.
(x, y) is a T-Link, if for any instance z, $d(x, y) < d(x, z)$ or $d(x, y) < d(y, z)$
If any two examples are T-Link then one of these examples is a noise or otherwise both examples are located on the boundary of the classes.
T-Link method can be used as a method of guided undersampling where the observations from the majority class are removed.



- Advantages
- Mitigates the problem of over fitting caused by random oversampling as synthetic examples are generated rather than replication of instances
- No loss of useful information

- Disadvantages
- While generating synthetic examples SMOTE does not take into consideration neighboring examples from other classes. This can result in increase in overlapping of classes and can introduce additional noise
- SMOTE is not very effective for high dimensional data

Easy Ensemble:

The idea behind EasyEnsemble is very simple. Several subsets of samples are created independently from the major class cases of the original dataset. Each subset of the major class cases should be about the **same size as the minor class.**

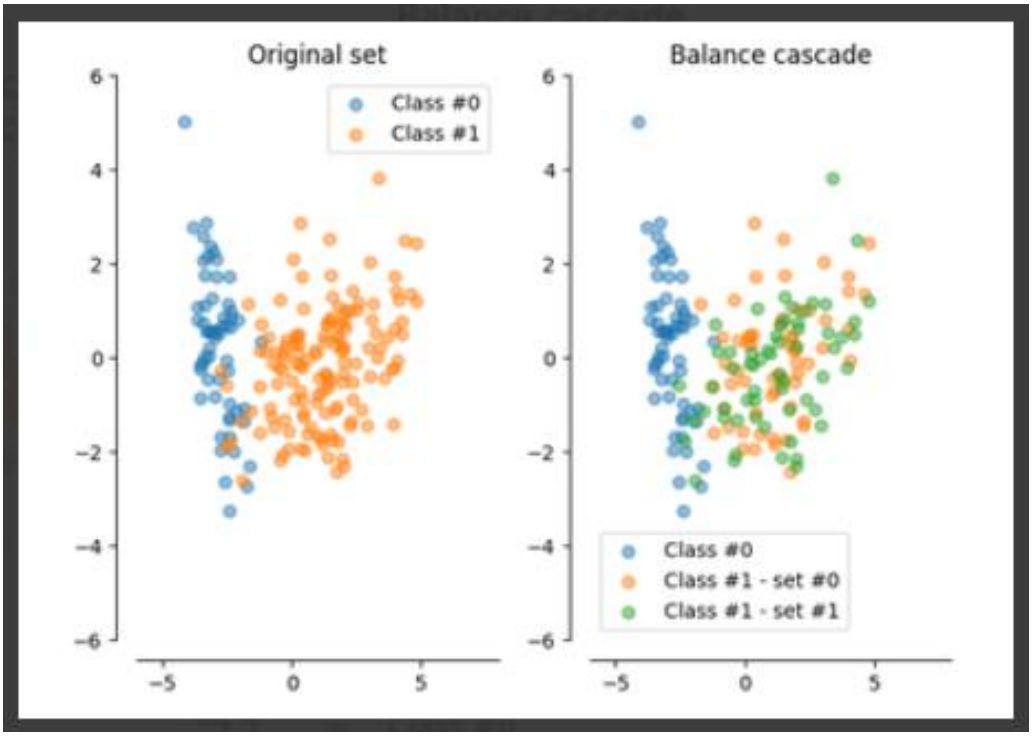
Each time, a subset of the majority class records gets selected and appended to all the minority class records. Then we train a classifier on this appended subset of data.

This process is reiterated multiple times until all the subsets of the majority class are modeled. In the end we combine all the classifiers created to produce final classification results.

Balanced Cascade:

BalanceCascade takes a more supervised approach to undersampling. It starts by constructing a subset made up of all the minority cases and a random sample of the majority class that is about the same size as the minority class.

After training on this subset, you take out the majority cases that can be classified correctly by the trained classifier and use the rest of the majority cases to go through the process again until the number of majority cases left is smaller than that of the minority cases.
In the end, you combine the classifiers of all these iterations in such a way that only the cases classified as positive by all the classifiers will be marked as positive



SMOTE Boost:

Data sampling methods combined with boosting works effectively in dealing with class imbalance problems.

AdaBoost iteratively builds an ensemble of weak learners(poor predictive models, but better than random guessing) by adjusting the weights of misclassified data during each iteration.

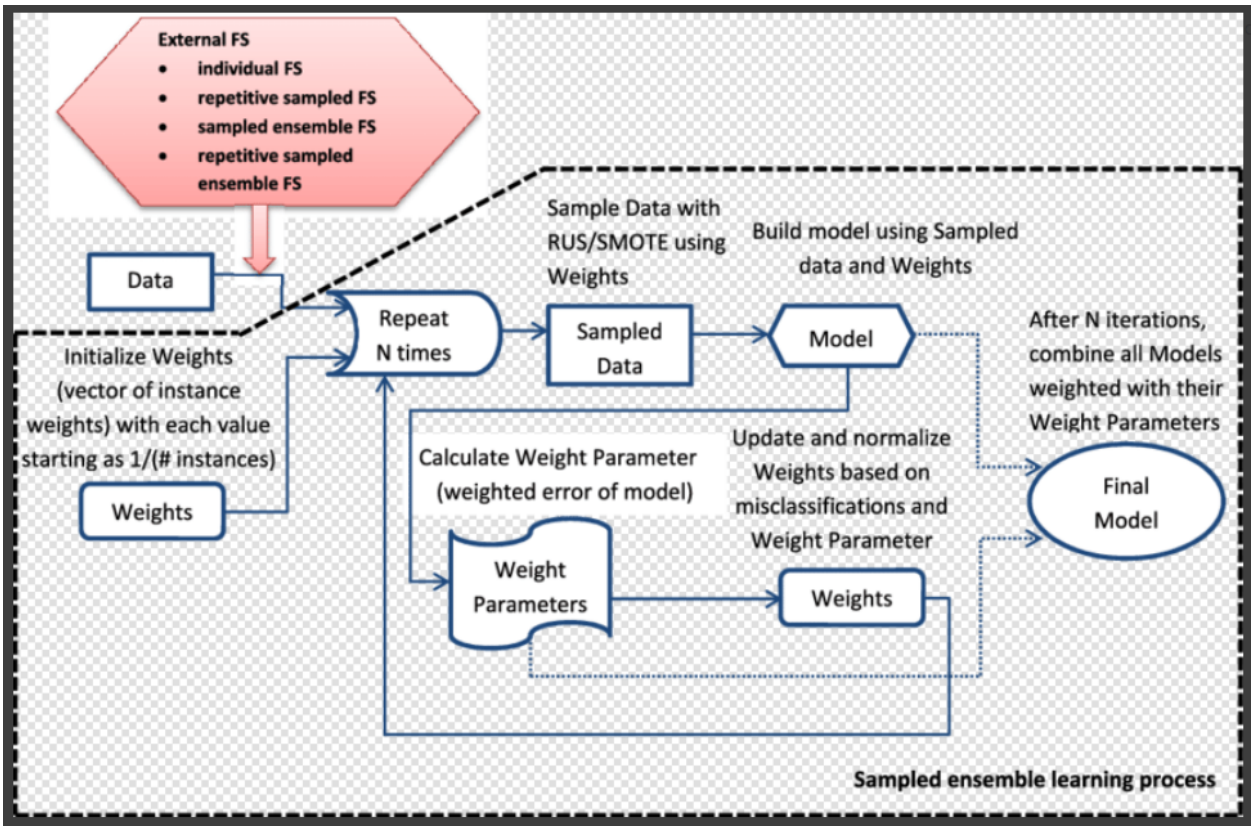
For the training of the first weak learner, AdaBoost assigns equal weight to each training set sample. For each subsequent weak learner, the weights are recalculated such that a higher weight is assigned to samples that the current weak learner misclassified.

This weight determines the probability that the sample will appear in the training of the next weak learner.and subsequently giving higher weight to the minority class at each successive iteration as data from this class is often misclassified.

SMOTEBoost injects the SMOTE method at each boosting iteration.

The advantage of this approach is that while standard boosting gives equal weights to all misclassified data, SMOTE gives more examples of the minority class at each boosting step.

Following is a diagram explaining the process of SMOTEBoost algorithm where Model used is AdaBoost.



SVM SMOTE:

SVM-SMOTE, an over-sampling technique, is used to handle the trade-off of either have less false positives or false negatives.

SMOTE balances class distribution by synthetically generating new minority class instances along directions from existing minority class instances towards their nearest neighbours.

SVM-SMOTE focuses on generating new minority class instances near borderlines with SVM so as to help establish boundary between classes.

ADASYN:

The essential idea of ADASYN is to use a weighted distribution for different minority class examples according to their level of difficulty in learning, where more synthetic data is generated for minority class examples that are harder to learn compared to those minority examples that are easier to learn.

The ADASYN approach improves learning with respect to the data distributions by reducing the bias introduced by the class imbalance and adaptively shifting the classification decision boundary toward the difficult examples.