**Data level approach to handle Imbalanced Dataset**

1. Random – Under Sampling: The data is balanced by reducing the number of instances of the majority class. For ex:

Total Observation = 1000

Class 1 = 20

Class 2 = 980

Event rate of class 1: 20/1000 = 2%

In this case if we take 10% samples without replacement from class 2 instances and then all the data of class1:

Total Observation = 20 + 98 = 118

Event rate of class 1: 20/118 = 17%

Advantages:

* Improve runtime and storage problem by reducing the training data when the dataset is huge

Disadvantages:

* Data loss
* The sample chosen by random under sampling may be biased sample. And it will not be an accurate representative of the population. Thereby, resulting in inaccurate results with the actual data set.

1. Random-Over Sampling: Randomly replicate the instances of the minority class.

Total Observation = 1000

Class 1 = 20

Class 2 = 980

Event rate of class 1: 20/1000 = 2%

Incase we are replicating the data 20 times.

Class 1 = 400

Class 2 = 980

Total Observation = 1380

Event rate: 400/1380 = 29%

Advantages:

* No data loss
* Better than under sampling

Disadvantages:

* Increase the likelihood of the overfitting since it replicates the minority class events.

1. Cluster-Based Over Sampling: K-Means clustering is applied to the minority and majority class independently to identify the clusters in the dataset. Subsequently each class is oversampled such that all clusters of the same class have an equal number of instances and all the classes have the same size.

Total Observation = 1000

Class 1 = 20

Class 2 = 980

Event rate of class 1: 20/1000 = 2%

Majority class clusters:

Cluster 1: 150

Cluster 2: 120

Cluster 3: 230

Cluster 4: 200

Cluster 5: 150

Cluster 6: 130

Minority class cluster:

Cluster 1: 8

Cluster 2: 12

After oversampling each cluster, all cluster of same class contain same number of observations.

Majority class clusters:

Cluster 1: 170

Cluster 2: 170

Cluster 3: 170

Cluster 4: 170

Cluster 5: 170

Cluster 6: 170

Minority class cluster:

Cluster 1: 250

Cluster 2: 250

Event rate: 500 /(500+1020) = 33%

Advantages:

* If the minority class contains imbalanced instances in the sub clusters, the over sampling performed results in balanced data within the minority cluster

Disadvantages:

* Possibility of overfitting.

1. Informed Over sampling: SMOTE(Synthetic Minority Over-sampling Technique): To avoid over fitting with creation of exact replicas. A subset of the data is taken from the minority class as an example and then a new synthetic similar instances are created. These are then added to the original dataset.

Total Observation = 1000

Class 1 = 20

Class 2 = 980

Event rate of class 1: 20/1000 = 2%

15 samples are taken from minority class and similar synthetic instances are generated 20 times.

Class 1 = 300

Class 2 = 980

Event rate of class 1 = 300/1280 = 23.4%

Advantages:

* Mitigates problem of overfitting caused by random sampling as synthetic examples are generated rather than replication of instances.
* No loss of information.

Disadvantages:

* While synthesizing instances, SMOTE does not take into consideration neighbouring examples from other classes. This can result in increase in overlapping of classes and can introduce noise.
* Not effective for high dimensional data.

1. Modified SMOTE(MSMOTE): SMOTE does not consider the underlying distribution of the minority class and the latent noises in the dataset. To improve the performance of SMOTE MSMOTE is used.

The algorithm classifies the samples of minority class into 3 distinct groups: Security/Safe samples, Border samples and latent noise samples. This is done by calculating the distances among samples of the minority class and the samples of the training data.

Security samples are those data points which can improve the performance of a classifier. Whereas the noise samples are those datapoints which can reduce the performance. The ones which are difficult to categorize are classified as border samples.

In MSMOTE, the strategy of selecting nearest neighbours is different from SMOTE. The algorithm randomly selects a data point from the k nearest neighbours for the security sample, selects the nearest neighbour from the border samples and does nothing for latent noise.

**Algorithmic Ensemble Techniques:**

1. Bagging based: Bootstrap aggregation involves generating ‘n’ different bootstrap training samples with replacement. And training the algorithm on each bootstrapped algorithm separately and then aggregating the predictions at the end.

Bagging is used to reduce the Overfitting in order to create strong learners for generating accurate predictions. Unlike Boosting, bagging allows replacement in the bootstrapped samples.

Advantages:

* Improves accuracy and stability of machine learning algorithm
* Reduces variance
* Overcomes overfitting
* Improved misclassification rate of the bagged classifier.
* In noisy data environments, bagging outperforms boosting.

Disadvantages:

* Bagging works only if the base classifiers are not bad to begin with. Bagging bad classifiers can further degrade performance.

1. Boosting based:

Adaboost:

Advantages:

* Simple implementation
* Good generalisation suited for any kind of classification problem
* Not prone to overfitting

Disadvantages:

* Sensitive to noisy data and outliers

Gradient Tree Boosting:

Disadvantages:

* Gradient boosting trees are harder to fit than random forest.
* Have 3 hyper parameters: Shrinkage parameter, depth of tree, number of trees. Proper tuning is required, if not tuned properly will result in overfitting.

XGBoost:

Advantages:

* 10 times faster than normal gradient boosting as it implements parallel processing.
* Handles missing values
* Unlke gradient boosting which stops splitting a node as soon as it encounters a negative loss, XGBoost splits up to the maximum depth specified and prunes the trees backwards and removes splits beyond which there is an only negative loss.