**BALANCING THE DATASET (RESEARCH TECHNIQUES FOR BALANCING DATA SETS)**

**Introduction**

Imbalanced data is a common problem in machine learning, where one class has a significantly higher number of observations than the other. This can lead to biased models and poor performance on the minority class. In this blog, we will discuss techniques for handling imbalanced data and improving model performance.

**What is Imbalanced data?**

**Imbalanced data** refers to datasets where the distribution of class labels is not equal, with one class having a significantly higher number of observations than the other. This can be a problem for machine learning algorithms, as they can be biased towards the majority class and perform poorly on the minority class.

Example: Let’s assume that XYZ is a bank that issues a credit card to its customers. Now the bank is concerned that some fraudulent transactions are going on and when the bank checks their data they found that for each 2000 transactions there are only 30 Nos of fraud recorded. So, the number of fraud per 100 transactions is less than 2%, or we can say more than 98% of transaction is “No Fraud” in nature. Here, the class “No Fraud” is called the **majority class,** and the much smaller in size “Fraud” class is called the **minority class**.

More such example of imbalanced data is –

· Disease diagnosis

· Customer churn prediction

· Fraud detection

· Natural disaster

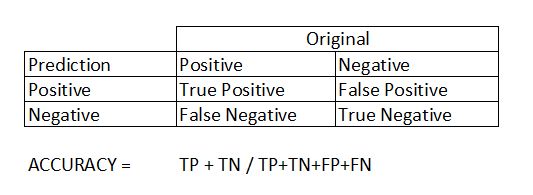
Class imbalanced is generally normal in classification problems. But, in some cases, this imbalance is quite acute where the majority class’s presence is much higher than the minority class.

**Problems with imbalanced data classification**

In very simple terms, the main problem with imbalanced dataset prediction is how accurately are we actually predicting both majority and minority class? Let’s assume we are going to predict disease from an existing dataset where for every 100 records only 5 patients are diagnosed with the disease. So, the majority class is 95% with no disease and the minority class is only 5% with the disease. Now, assume our model predicts that all 100 out of 100 patients have no disease.

Sometimes when the records of a certain class are much more than the other class, our classifier may get biased towards the prediction. In this case, the confusion matrix for the classification problem shows how well our model classifies the target classes and we arrive at the accuracy of the model from the confusion matrix. It is calculated based on the total no of correct predictions by the model divided by the total no of predictions. In the above case it is (0+95)/(0+95+0+5)=0.95 or 95%. It means that the model fails to identify the minority class yet the accuracy score of the model will be 95%.

Thus our traditional approach of classification and model accuracy calculation is not useful in the case of the imbalanced dataset.



**Approach to deal with the imbalanced dataset problem**

In rare cases like fraud detection or disease prediction, it is vital to identify the minority classes correctly. So model should not be biased to detect only the majority class but should give equal weight or importance towards the minority class too. Here I discuss some of the few techniques which can deal with this problem. There is no right method or wrong method in this, different techniques work well with different problems.

1. **Choose Proper Evaluation Metric**

**The accuracy** of a classifier is the total number of correct predictions by the classifier divided by the total number of predictions. This may be good enough for a well-balanced class but not ideal for the imbalanced class problem. The other metrics such as **precision** is the measure of how accurate the classifier’s prediction of a specific class and **recall** is the measure of the classifier’s ability to identify a class.

For an imbalanced class dataset F1 score is a more appropriate metric. It is the harmonic mean of precision and recall and the expression is –



So, if the classifier predicts the minority class but the prediction is erroneous and false-positive increases, the precision metric will be low and so as F1 score. Also, if the classifier identifies the minority class poorly, i.e. more of this class wrongfully predicted as the majority class then false negatives will increase, so recall and F1 score will low. F1 score only increases if both the number and quality of prediction improves.

1. **Resampling (Oversampling and Undersampling)**

This technique is used to upsample or downsample the minority or majority class. When we are using an imbalanced dataset, we can oversample the minority class using replacement. This technique is called oversampling. Similarly, we can randomly delete rows from the majority class to match them with the minority class which is called undersampling. After sampling the data we can get a balanced dataset for both majority and minority classes. So, when both classes have a similar number of records present in the dataset, we can assume that the classifier will give equal importance to both classes.

1. **SMOTE**

**Synthetic Minority Oversampling Technique** or **SMOTE** is another technique to oversample the minority class. Simply adding duplicate records of minority class often don’t add any new information to the model. In SMOTE new instances are synthesized from the existing data. If we explain it in simple words, SMOTE looks into minority class instances and use *k* nearest neighbor to select a random nearest neighbor, and a synthetic instance is created randomly in feature space.

1. **BalancedBaggingClassifier**

When we try to use a usual classifier to classify an imbalanced dataset, the model favors the majority class due to its larger volume presence. A BalancedBaggingClassifier is the same as a sklearn classifier but with additional balancing. It includes an additional step to balance the training set at the time of fit for a given sampler. This classifier takes two special parameters “sampling\_strategy” and “replacement”. The **sampling\_strategy** decides the type of resampling required (e.g. ‘majority’ – resample only the majority class, ‘all’ – resample all classes, etc) and **replacement** decides whether it is going to be a sample with replacement or not.

1. **Threshold moving**

In the case of our classifiers, many times classifiers actually predict the probability of class membership. We assign those prediction’s probabilities to a certain class based on a threshold which is usually 0.5, i.e. if the probabilities < 0.5 it belongs to a certain class, and if not it belongs to the other class.

For imbalanced class problems, this default threshold may not work properly. We need to change the threshold to the optimum value so that it can efficiently separate two classes. We can use ROC Curves and Precision-Recall Curves to find the optimal threshold for the classifier. We can also use a grid search method or search within a set of values to identify the optimal value.

1. **DATA AUGMENTATION**

Data augmentation involves creating additional data points by modifying existing data. This can be done by applying various transformations such as rotations, translations, and flips to the existing data.

1. **One-class classification**

One-class classification involves training a model on only one class and then using it to identify data points that do not belong to that class. This can be useful for identifying anomalies and outliers in the data.

1. **Cost-sensitive learning**

Cost-sensitive learning involves adjusting the cost of misclassifying data points to account for the class imbalance. This can be done by assigning a higher cost to misclassifying the minority class, which encourages the model to prioritize correctly classifying the minority class.

**Choosing the best technique for handling imbalanced data**

After discussing techniques for handling imbalanced data, we learned several approaches that can be used to address the issue. The most common techniques include undersampling, oversampling, and feature selection. Undersampling involves reducing the size of the majority class to match that of the minority class, while oversampling involves creating new instances of the minority class to balance the data. Feature selection is the process of selecting only the most relevant features to reduce the noise in the data.

In conclusion, it is recommended to use both undersampling and oversampling techniques to balance the data, with oversampling being the most effective. However, the choice of technique will ultimately depend on the specific characteristics of the dataset and the problem at hand.