# Assignment –Imbalanced classes in label - effects, remedies - data transformation, choice of evaluation metrics, robust algorithms

**Classification** predictive modelling involves predicting a class label for a given observation.

An imbalanced classification problem is an example of a classification problem where the distribution of examples across the known classes is biased or skewed. The distribution can vary from a slight bias to a severe imbalance where there is one example in the minority class for hundreds, thousands, or millions of examples in the majority class or classes.

Imbalanced classifications pose a challenge for predictive modelling as most of the machine learning algorithms used for classification were designed around the assumption of an equal number of examples for each class. This results in models that have poor predictive performance, specifically for the minority class. This is a problem because typically, the minority class is more important and therefore the problem is more sensitive to classification errors for the minority class than the majority class.

**Effect of imbalanced class in Label: -**

Most machine learning algorithms assume data equally distributed. So, when we have a class imbalance, the machine learning classifier tends to be more biased towards the majority class, causing bad classification of the minority class.

Remedies to solve Imbalanced classes in label

# Use the right evaluation metrics

# Applying inappropriate evaluation metrics for model generated using imbalanced data can be misleading.

# Example: - Training data is the one illustrated in graph below. If accuracy is used to measure the goodness of a model, a model which classifies all testing samples into “0” will have an excellent accuracy (99.8%), but obviously, this model won’t provide any valuable information for us.

# 

* Precision/Specificity: how many selected instances are relevant.
* Recall/Sensitivity: how many relevant instances are selected.
* F1 score: harmonic mean of precision and recall.
* MCC: correlation coefficient between the observed and predicted binary classifications.
* AUC: relation between true-positive rate and false positive rate.

# *Resample the training set*

## **2.1. Under-sampling**-Under-sampling balances the dataset by reducing the size of the abundant class. This method is used when quantity of data is sufficient.

## **2.2. Over-sampling**-On the contrary, oversampling is used when the quantity of data is insufficient. It tries to balance dataset by increasing the size of rare samples.

# Use K-fold Cross-Validation in the Right Way:-

Cross-validation should be applied properly while using over-sampling method to address imbalance problems.

Keep in mind that over-sampling takes observed rare samples and applies bootstrapping to generate new random data based on a distribution function. If cross-validation is applied after over-sampling, basically what we are doing is overfitting our model to a specific artificial bootstrapping result. That is why cross-validation should always be done before over-sampling the data, just as how feature selection should be implemented.

1. **Ensemble Different Resampled Datasets: -**

The easiest way to successfully generalize a model is by using more data. The problem is that out-of-the-box classifiers like logistic regression or random forest tend to generalize by discarding the rare class. One easy best practice is building n models that use all the samples of the rare class and n-differing samples of the abundant class.

# Resample with Different Ratios: -

# Instead of training all models with the same ratio in the ensemble, it is worth trying to ensemble different ratios.  So, if 10 models are trained, it might make sense to have a model that has a ratio of 1:1 (rare: abundant) and another one with 1:3, or even 2:1. Depending on the model used this can influence the weight that one class gets.

# Imbalanced data image

# Cluster the abundant class: -

# An elegant approach was proposed by Sergey on Quora [2]. Instead of relying on random samples to cover the variety of the training samples, he suggests clustering the abundant class in r groups, with r being the number of cases in r. For each group, only the medoid (centre of cluster) is kept. The model is then trained with the rare class and the medoids only.