# 8 Tactics to Combat Imbalanced Classes in Your Machine Learning Dataset

[[Link](https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/)]

1. **Can you collect more data?**
   1. Almost always overlooked
2. **Try changing your performance metric**
   1. Confusion Matrix
      1. Precision
      2. Recall
      3. F1 Score
   2. Kappa (or Cohen’s Kappa):   
      Classification accuracy normalized by the imbalance of the classes in the data
   3. ROC Curves: Like precision and recall, accuracy is divided into sensitivity and specificity and models can be chosen based on the balance thresholds of these values
3. **Try resampling your dataset**
   1. There are two main methods to even-up the classes:
      1. Add copies of instances from the under-represented class called over-sampling(or more formally sampling with replacement), or
      2. Delete instances from the over-represented class, called under-sampling
   2. These approaches are easy to implement and fast to run
   3. An excellent starting point
   4. Rules of thumb:
      1. Consider under-sampling when you have lots of observations (> tens of thousands)
      2. Consider over-sampling when you do not have lots of observations
      3. Consider testing random and non-random (e.g., stratified) sampling schemes
      4. Consider testing different resampled ratios (e.g., you do not have to target a 1:1 ratio in a binary classification problem, try other ratios)
4. **Try generating synthetic samples**
   1. Simple way to generate synthetic samples is to randomly sample the attributes from instances in the minority class.
   2. You could sample them empirically within your dataset or you could use a method like Naïve Bayes that can sample each attribute independently when run in reverse. You will have more and different data, but the non-linear relationships between the attributes may not be preserved.
   3. Systematic Algorithms:
      1. SMOTE (Synthetic Minority Over-sampling Technique)
         1. The algorithm selects two or more similar instances(using a distance measure) and perturbing an instance one attribute at a time by a random amount within the difference to the neighboring instances
5. Try different algorithms
   1. Decision trees often perform well on imbalanced datasets
      1. C4.5, C5.0, CART, and Random Forest
6. Try Penalized Models
   1. Penalized classification imposes an additional cost on the model for making classification mistakes on the minority class during training
   2. These penalties can bias the model to pay more attention to the minority class
   3. Often the handling of class penalties or weights are specialized to the learning algorithm.
   4. Penalized versions of algorithms:
      1. Penalized-SVM
      2. Penalized-LDA
   5. Generic frameworks for penalized models:
      1. Weka has a CostSensitive Classifier that can wrap any classifier and apply a custom penalty matrix for miss classification
   6. Setting up the penalty matrix is complex
   7. Penalization is desirable when:
      1. you are locked into a specific algorithm
      2. unable to resample
      3. you are getting poor results
7. Try a different perspective
   1. There are fields of study dedicated to imbalanced datasets. They have their own algorithms, measures, and terminology
   2. Looking and thinking about your problem from these perspectives can sometimes shame loose some ideas
   3. Two you might like to consider are
      1. **Anomaly detection**
      2. **Change detection**
   4. Anomaly detection:
      1. detection of rare events
         1. Machine malfunction indicated through its vibrations, or
         2. Malicious activity by program indicated by its sequence of system calls
      2. This shift in thinking considers the minor class as the outliers class which might help you think of new ways to separate and classify samples
   5. Change detection:
      1. Like anomaly detection, except rather than looking for anomaly it is looking for change or difference.
      2. This might be change in behavior of a user as observed by usage patterns or bank transactions
8. Try Getting Creative
   1. Really climb inside your problem and think about how to break it down into smaller problems that are more tractable.