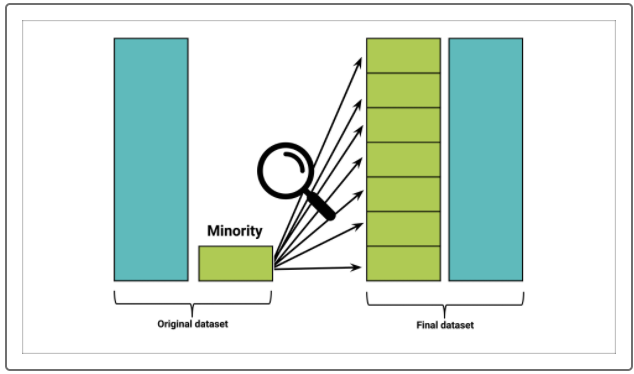
Class imbalance is a common problem in classification. It occurs when one class is much larger than the other class. For example, if you work for a credit card company and want to detect fraudulent transactions, you will deal with many more non-fraudulent transactions than fraudulent ones. In this case, the non-fraudulent class is much larger than the fraudulent class.

**Class imbalance** refers to a situation in which the existing classes in a dataset aren't equally represented. Earlier we discussed a fraud detection scenario in which a large number of credit card transactions are legitimate, and only a small number are fraudulent. For example, let's say that out of 100,000 transactions, 50 are fraudulent and the rest are legitimate. The pronounced imbalance between the two classes (fraudulent and non-fraudulent) can cause machine learning models to be biased toward the majority class. In such a case, the model will be much better at predicting non-fraudulent transactions than fraudulent ones. This is a problem if the goal is to detect fraudulent transactions!

In such a case, even a model that blindly classifies every transaction as non-fraudulent will achieve a very high degree of accuracy. As we saw previously, one strategy to deal with class imbalance is to use appropriate metrics to evaluate a model's performance, such as precision and recall.

Another strategy is to use **oversampling**. The idea is simple and intuitive: If one class has too few instances in the training set, we choose more instances from that class for training until it's larger.



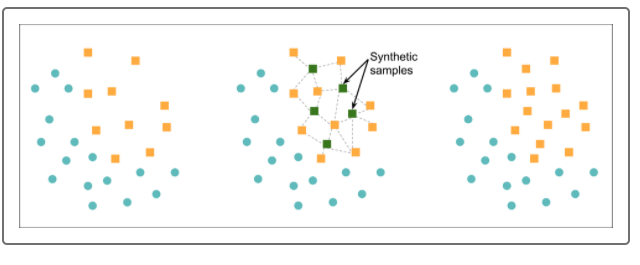
We'll discuss two oversampling techniques: random oversampling and synthetic minority oversampling technique.

**Random Oversampling**

In **random oversampling,** instances of the minority class are randomly selected and added to the training set until the majority and minority classes are balanced.

## Synthetic Minority Oversampling Technique

The **synthetic minority oversampling technique (SMOTE)** is another oversampling approach to deal with unbalanced datasets. In SMOTE, like random oversampling, the size of the minority is increased. The key difference between the two lies in how the minority class is increased in size. As we have seen, in random oversampling, instances from the minority class are randomly selected and added to the minority class. In SMOTE, by contrast, new instances are interpolated. That is, for an instance from the minority class, a number of its closest neighbors is chosen. Based on the values of these neighbors, new values are created.



It's important to note that although SMOTE reduces the risk of oversampling, it does not always outperform random oversampling. Another deficiency of SMOTE is its vulnerability to outliers. We said earlier that a minority class instance is selected, and new values are generated based on its distance from its neighbors. If the neighbors are extreme outliers, the new values will reflect this. Finally, keep in mind that sampling techniques cannot overcome the deficiencies of the original dataset!