## subject 6: Predicting the cellular localization sites of proteins

The challenge of imbalanced classes

Imbalanced classification problems 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.

* Majority Class: More than half of the examples belong to this class, often the negative or normal case.
* Minority Class: Less than half of the examples belong to this class, often the positive or abnormal case.

Prediction accuracy is the most common metric for classification tasks, although it is inappropriate and potentially dangerously misleading when used on imbalanced classification tasks. The reason for this is because if 98 percent of the data belongs to the negative class, you can achieve 98 percent accuracy on average by simply predicting the negative class all the time, achieving a score that naively looks good, but in practice has no skill. Instead, alternate performance metrics must be adopted. Popular alternatives are the precision and recall scores that allow the performance of the model to be considered by focusing on the minority class, called the positive class.

A simple approach to using standard machine learning algorithms on an imbalanced dataset is to change the training dataset to have a more balanced class distribution.

* This can be achieved by deleting examples from the majority class, referred to as undersampling. A possible downside is that examples from the majority class that are helpful during modelling may be deleted.
* An alternative to deleting examples from the majority class is to add new examples from the minority class. This can be achieved by synthesizing new examples using existing ones in the training dataset. These new examples will be “close” to existing examples in the feature space, but different in small but random ways.
* Finally, cost-sensitive learning is a subfield of machine learning that takes the costs of prediction errors (and potentially other costs) into account when training a machine learning model. Many machine learning algorithms can be updated to be cost-sensitive, where the model is penalized for misclassification errors from one class more than the other, such as the minority class.

Project problem statement:

The computational prediction of the subcellular localization of bacterial proteins is an important step in genome annotation and in the search for novel vaccine or drug targets. In fact, in order to function properly, proteins must be transported to various localization sites within the cell, and the cellular localization site of a protein affects its potential functionality. Fortunately, the information needed for correct localization is generally found in the protein sequence itself. It can be framed as the problem of classifying proteins using their amino acid sequences in their cell localization sites.

E.Coli Dataset

In this project, we will use a standard imbalanced machine learning dataset referred to as the “E.coli” dataset, also referred to as the “protein localization sites” dataset. The dataset is credited to Kenta Nakai and was developed into its current form by Paul Horton and Kenta Nakai in their 1996 paper titled “A Probabilistic Classification System For Predicting The Cellular Localization Sites Of Proteins.” In it, they achieved a classification accuracy of 81 percent.

The dataset is comprised of 336 examples of E.coli proteins and each example is described using seven input variables calculated from the proteins amino acid sequence.

Ignoring the sequence name, the input features are described as follows:

* mcg: McGeoch’s method for signal sequence recognition.
* gvh: von Heijne’s method for signal sequence recognition.
* lip: von Heijne’s Signal Peptidase II consensus sequence score.
* chg: Presence of charge on N-terminus of predicted lipoproteins.
* aac: score of discriminant analysis of the amino acid content of outer membrane and periplasmic proteins.
* alm1: score of the ALOM membrane-spanning region prediction program.
* alm2: score of ALOM program after excluding putative cleavable signal regions from the sequence.

There are eight classes described as follows:

* cp: cytoplasm
* im: inner membrane without signal sequence
* pp: periplasm
* imU: inner membrane, non cleavable signal sequence
* om: outer membrane
* omL: outer membrane lipoprotein
* imL: inner membrane lipoprotein
* imS: inner membrane, cleavable signal sequence

The distribution of examples across the classes is not equal and, in some cases, severely imbalanced.

For example, the “cp” class has 143 examples, whereas the “imL” and “imS” classes have just two examples each.

Techniques to be used :

Explore the following classifiers:

* Support vector machines
* K- nearest neighbors

Don’t forget to tune the algorithms using grid search techniques.

Then work around the imbalance in the dataset using SMOTE technique.