Imbalanced classification

Part 2: Cross validation and oversampling

By: Noureddin Sadawi, PhD

University of London

Cross validation

- The k-fold cross-validation procedure involves splitting the training dataset into *k* folds.
- The first *k-1* folds are used to train a model, and the holdout *k*th fold is used as the test set.
- This process is repeated and each of the folds is given an opportunity to be used as the holdout test set.
- A total of *k* models are fit and evaluated, **and the performance of the model is calculated as the mean of these runs.**

Cross validation

Iteration 1	Test	Train	Train	Train	Train
Iteration 2	Train	Test	Train	Train	Train
Iteration 3	Train	Train	Test	Train	Train
Iteration 4	Train	Train	Train	Test	Train
Iteration 5	Train	Train	Train	Train	Test

- Compute an evaluation metric for each iteration.
- In the end, compute the average and STD for all those metric values.

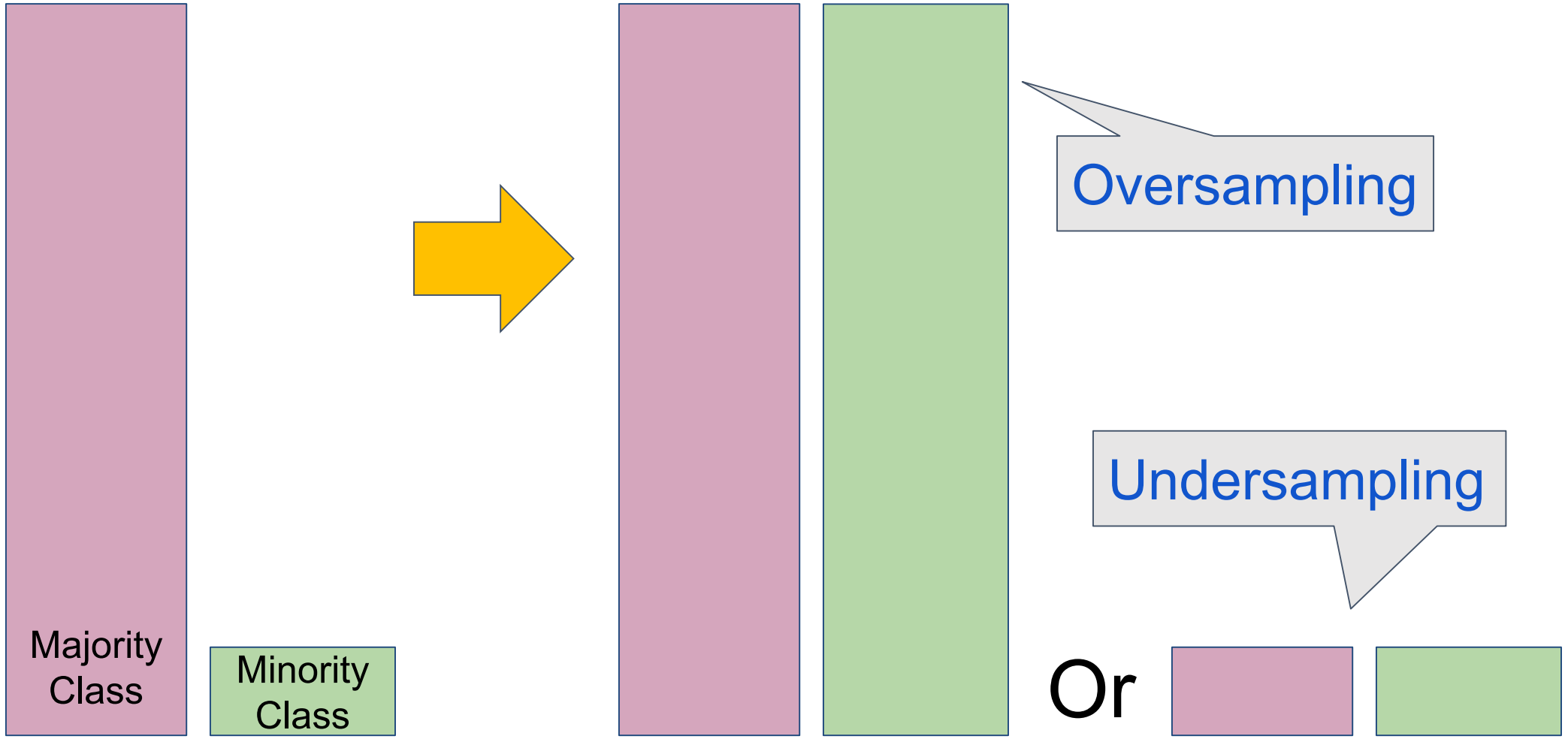
CV for imbalanced classification

- In CV the data is usually split into *k*-folds with a uniform probability distribution.
- This is not appropriate for evaluating imbalanced classifiers.
- It is likely that one or more folds will have few or no examples from the minority class.
- This means that some or perhaps many of the model evaluations will be misleading, as the model need only predict the majority class correctly.
- The solution: split a dataset randomly in a way that maintains the same class distribution in each subset.
- This is called **stratification** or **stratified sampling** and the target variable (*y*), the class, is used to control the sampling process.

Data sampling

- Using data sampling we change the composition of the training dataset (the most popular solution to an imbalanced classification problem).
- These methods are usually simple to understand and implement.
- Once applied to transform the training dataset, so many standard machine learning algorithms can then be used directly.
- Sampling is only performed on the training dataset.
- It is not performed on the holdout test or validation dataset.
- Evaluate the resulting model on data that is both real and representative of the target problem domain

Data sampling



Oversampling techniques

Oversampling methods increase the number of examples in the minority class by duplicating them or synthesising new examples from the original examples in the minority class.

Example methods:

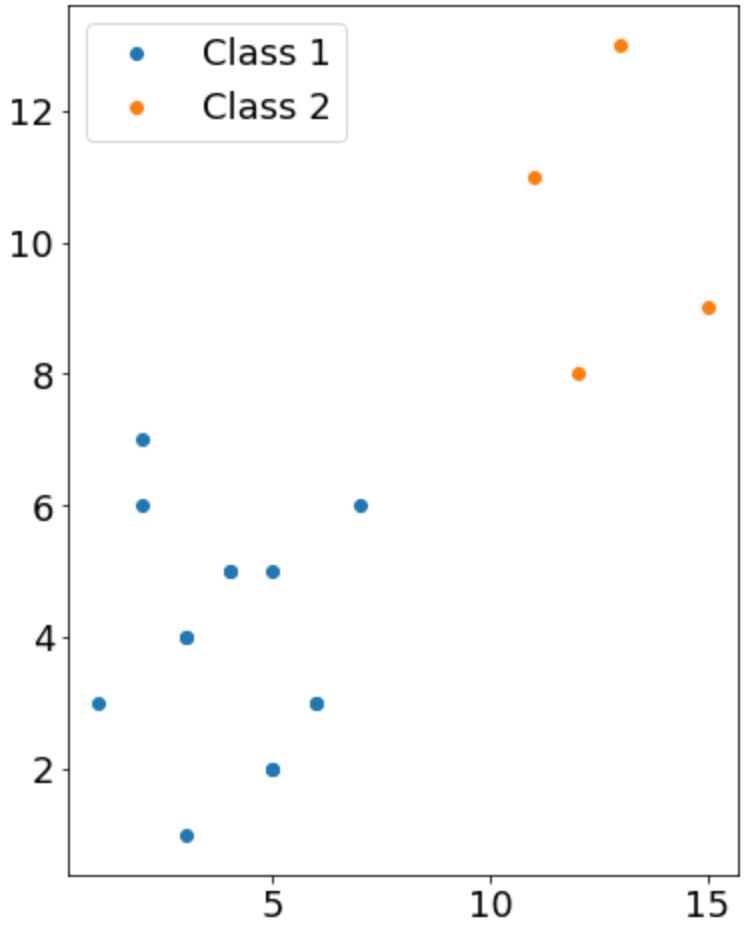
- random oversampling: randomly duplicating examples from the minority class in the training dataset (i.e. sampling with replacement)
- synthetic minority oversampling technique (SMOTE)
- borderline-SMOTE
- borderline oversampling with SVM
- adaptive synthetic sampling (ADASYN).

SMOTE

- Synthetic minority oversampling technique.
- Works by selecting examples that are close in the feature space, drawing a line between them generating a new sample as a point along that line.
  - The most popular and perhaps most successful oversampling method.
  - There are many extensions to the SMOTE method that aim to be more selective.

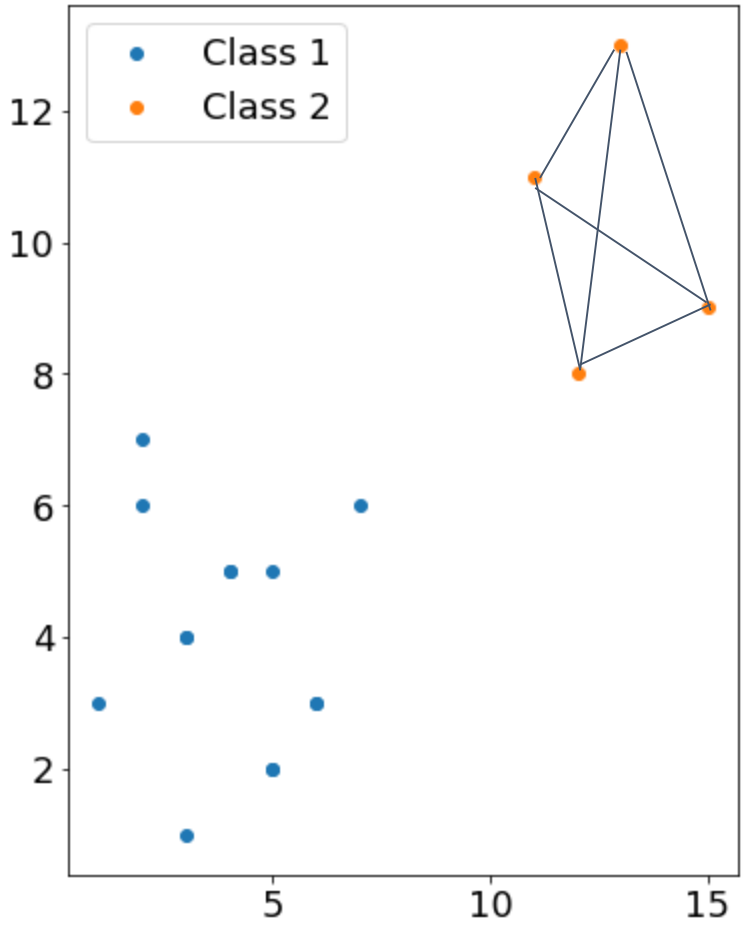
How SMOTE works 1/3

- Generates new instances (not real) based on existing real instances.
- Needs at least two instances of the class to work.



How SMOTE works 2/3

- Generates new instances (not real) based on existing real instances.
- Needs at least two instances of the class to work.



How SMOTE works 3/3

- Generates new instances (not real) based on existing real instances.
- Needs at least two instances of the class to work.