	import from imblearn.under_sampling.	Brief Definition / Steps	Method of data Exclusion	Final Dataset Size	Fixed vs Cleaning	Under-sampling
Random	RandomUnderSampler	extracting at random samples from the majority class, until they reach a certain ratio (typically 1)	Random	2 x minority class	Fixed	Random
Condensed	CondensedNearestNeighbour	1) Put all minority class observations in a group, say group O 2) Add 1 sample (at random) from the majority class to group O 3) Train a KNN with group O 4) Take a sample of the majority class that is not in group O yet 5) Predict its class with the KNN from point 3 6) If the prediction was correct, go to 4 and repeat 7) If the prediction was incorrect, add that sample to group O, go to 3 and repeat 8) Continue until all samples of the majority class were either assigned to O or left out 9) Final version of Group O is our undersampled dataset	Samples outside the boundary between the classes	Varies	Cleaning	Keep boundary observations
Tomek Links	TomekLinks	Removing Tomek links. (if 2 observations are nearest neighbours, and from a different class, they are Tomek Links)	Samples are Tomek Links	Varies	Cleaning	Remove noisy observations
One Sided	OneSidedSelection	CNN + Tomek Links	Combined	Varies	Cleaning	Both keep and remove boundary
Edited Nearest	EditedNearestNeighbours	1) Train a KNN algorithm on the data, user defines number of neighbours, typically 3 2) Find the nearest neighbour to each observation (each observation should have the number of neighbours defined in step 1, 3 is the default) 3) Find the label of each of the neighbours (we know it, is the target in the dataset) Two undersampling strategies: 4) mode: if the majority of the neighbours show the same label as the observation, then we keep the observation 5) all: if all the neighbours show the same label as the observation, then we keep the observation 6) Alternatively, we remove the observation of the dataset - undersample 7) The undersampled dataset, is the one left after removing observations	Observation's class is different from that of its nearest neighbours	Varies	Cleaning	Remove noisy observations

Repeated ENN	RepeatedEditedNearestNeighbours	1) Train a KNN algorithm on the entire dataset, typically a 3 KNN 2) Check all observations from majority class and remove observations if its class is different from that of its neighbour 3) Train a new 3 KNN over the remaining observations 4) Go to 2 and repeat.	Repeats ENN multiple times	Varies	Cleaning	Remove noisy observations
All KNN	AllKNN	1) Train a 3 KNN algorithm on the entire dataset 2) Check all observations from majority class and remove observations if its class is different from that of its (3) neighbours 3) Train a 4 KNN algorithm on the remaining samples. 4) Go to 2 and repeat. Every time you reach 3, add a neighbour to the KNN.	Repeats ENN, plus 1 neighbour in each KNN iteration	Varies	Cleaning	Remove noisy observations
Neighbourhood Cleaning Rule	NeighbourhoodCleaningRule	1) Train a 3 KNN on entire dataset 2) Remove observations which class is not that of its 3 neighbours 3) Train a 1 KNN on the entire dataset 4) Remove observations from the majority class that are misclassified by the 1 KNN	Combines ENN with a 1 KNN data exclusion criteria	Varies	Cleaning	Remove noisy observations
Near Miss (version 1)	NearMiss(version=1,)	Select and retain majority class observations closer to the closest minority class	distance	2 x minority class	Fixed	Keep boundary observations
Near Miss (version 2)	NearMiss(version=2,)	Select and retain majority class observations closer to the farthest minority class	same as above	same as above	same as above	same as above
Near Miss (version 3)	NearMiss(version=3,)	Select and retain majority class observations furthest from their nearest neighbours *NearMisses are good for working with text datasets	same as above	same as above	same as above	same as above
Instance	InstanceHardnessThreshold	1) train a classifier 2) Determine the P of each observation of the majority class 3) Find the threshold above which to retain the samples: (1 - n(minor)/n(major)) * 100 4) based on the classifier prediction, chose major observationa that are above the threshold 5) select at least as many observations from the major class as those from minor class	Probability by a certain classifier > a threshold	Varies. Minimum 2 x minority class	Fixed	Remove noisy observations