	Creates Synthetic Data	Expands bondary minority	Categorical variables	Distance based	Number of KNN	Template for new samples	Over-sampling criteria	Observations
Random Oversampling	No	No	yes	No	-	All minority samples	Extracts samples from the minority class or classes, at random, and adds them to the final dataset	Duplicates Data
SMOTE	Yes	No	No	Euclidean	1	All minority samples	• Train a KNN on minority class observations - find each observation's 5 closest neighbours **To create the new synthetic data:** • Select examples from the minority class at random (to be used as templates) • Select a neighbour of each example at random (for the interpolation) • Extract a random number between 0 and 1 • Calculate the new examples as = original sample - factor * (original sample - neighbour) **• The final dataset consists of the original dataset + the newly created examples**	Both the template and the neighbour used in the interpolation belong to the minority class. Typically looks at the 5 closest neighbours.
SMOTE-NC	Yes	No	Yes	• For numerical features: Euclidean. • For categorical features: the squared median of the standard deviation of the continuous features in the minority class (if the values of the categories are different, otherwise 0)	1	All minority samples	Procedure identical to SMOTE with 2 considerations: **On distance calculation:** • For categorical features, the distance is calculated computing the median of the std of all continous features in the minority class • If the 2 observations show the same categorical value, the distance is 0, otherwise it is the square of the median as above **On new example creation:** • The value for continuous features is determined as in SMOTE • The value of the categorical features is that shown by the majority of the neighbours of the observation used as template **• The final dataset consists of the orignal data + the newly created examples**	Dataset must contain **both** continuous and categorical variables.
SMOTE-N	Yes	No	Yes	Value Difference Metric The VDM relies on conditional probabilities per class, so it needs to be calculated on the entire data	1	All minority samples	• Determine the distance between all observations (majority and minority) using the VDM • Train a KNN on minority class samples only, using the pre-computed distances • Take examples at random from the minority class (templates) • Create the new examples: the values of the categorical features are those shown by the majority of the template's neighbours **• The final dataset consists of the original data + the newly created examples**	• Dataset must contain **only** categorical variables. • Template and neighbours belong to minority class
Borderline SMOTE	Yes	Yes	No	Euclidean	2	Minority samples for which the majority of the neighbours belong to another class	• Train a KNN on entire dataset • Find the M closest neighbours to each observation from the minority group • If most, but not all, neighours belong to a different class, add the observation to a DANGER group **Variant 1:** • Train another KNN only on minority group, find each DANGER group observation's closest K neighbours • Interpolate as in SMOTE from templates in DANGER group to minority neighbours **Variant 2:** • Train another KNN only on minority group, find each DANGER group observation's closest K neighbours • Create some examples by interpolation as in SMOTE from templates in DANGER group to minority neighbours • Create other examples by interpolate as in SMOTE from templates in DANGER group to majority observations, but the factor f, used to create the new observation varies at random between 0 and 0.5 **• The final dataset consists of the original dataset + all the newly created examples**	Not clear in original article, how to find the neighbours from the majority in variant 2, and which proportion should be from minority and majority
SVM SMOTE	Yes	Yes	No	Euclidean	2 (plus 1 SVM)	Miority examples that are the support vectors of the SVM	• Train a SVM on entire dataset • Find the support vectors of the minority class, these will be the templates • Train a KNN on entire dataset, find the 10 closest neighbours of the support vectors. • Decide between inter and extrapolation: if most of the neigbhours are from majority class, interpolate, otherwise, extrapolate • Train another KNN, this time only on minority group, find the 5 closest neighbours to the support vectors • Create the synthetic examples by inter or extrapilation between templates and their neighbours • Note that the neighbours are not chosen at random, but instead from the closer to the furthest to create the synthetic data • The final dataset consists of the original dataset + all the newly created examples	• Template and neighbours belong to minority class • Majority class observations used to decide between inter and extrapolation
K-Means SMOTE	Yes	Yes	No	Euclidean	1	All minority samples	• With k-means, find the naturally occurring clusters in the dataset • Select the clusters to over-sample: those where the imbalance ratio > 1 (have at least 50% of observations from the minority). • Determine how many samples to over-sample from each selected cluster • Over-sample as per SMOTE within each cluster **• The final dataset consist of the original dataset + the newly created examples**	• We need to know a priori the number of naturally occurring clusters and or set up this as a hyperparameter. • To select clusters we can also optimize the IR to use as filter. • If the clusters contain few observations, we may need to reduce the number of neighbours for the interpolation.
ADASYN	Yes	Yes	No	Euclidean	1	• Minority samples for which some of their neighbours belong to another class. • More samples created from those with more neighbours from a different class.	1- Determine the balancing ratio = X(minority)/X(majority) 2- Determine the number of new examples to create: G = (Xmaj - Xmin) * factor (the factor is 1 to attain a balancing ratio of 1) 3- Train a KNN on entire dataset 4- Find the K closest neighbours to each example from the minority class 5- Determine a weighting factor for each observation of Xmin: ri = D / K, where K is the number of neighbours and D is the number of neighbours that do not belong to Xmin 6- Normalise ri: rnorm = ri / sum(r) 7- Determine how many observations should be created from each observation of Xmin: Gi = ri * G (G was determined in step2, ri om step 6) 8- Select a neighbour of each example at random (for the interpolation, can be Xmin or Xmaj) 9- Extract a random number between 0 and 1 10- Calculate the new examples as = original sample - factor * (original sample - neighbour) **• The final dataset consists of the original dataset + all the newly created examples**	• More synthetic data is generated from samples that are harder to classify • Template is from minority class • Neighbour for interpolation can be from any class • The idea is to create more examples from those observations that are harder to classify, aka, those at the boundary between classes