Enn

def enn(data, y, samp_method = "balance", drop_na_col = True, drop_na_row = True,
  rel_thres = 0.5, rel_method = "auto", rel_xtrm_type = "both", rel_coef = 1.5, rel_ctrl_pts_rg = None,   k = 3, n_jobs = 1):

Function designed to help solve the problem of imbalanced data for regression; ENN under-samples the majority class.

Parameters:

main arguments / inputs:

data: pandas dataframe, the training set.

y: string, response variable y by name. It should be an header name found in the dataframe data.

samp_method: { 'balance' , 'extreme' }, default = 'balance' , specified method to determine over / under sampling percentage.

drop_na_col: bool, default = 'True' , if 'True' , auto drop columns with nan's.

drop_na_row: bool, default = 'True' , if 'True' , auto drop rows with nan's.

phi relevance function arguments / inputs:

rel_thres: positive real, default = 0.5, define the relevance threshold considered rare in phi relevance function.

rel_method: { 'auto' , 'manual' }, default = 'auto' , the relevance method in phi relevance function.

rel_xtrm_type: { 'low' , 'high' , 'both' }, default = 'both' , distribution focus on high, low or both.

rel_coef: positive real, default = 1.5, coefficient for box plot in phi relevance function to consider rare.

rel_ctrl_pts_rg: 2d array, default = None, when rel_method = 'manual' , it inputs for "manual" rel method.

KNeighborsClassifier attribute:

k: positive integer, default = 3, number of the neighbourhood to consider to compute the k-NN.

n_jobs: positive integer, default = 1, the number of parallel jobs to run for neighbors search.

RandomUnderSamplier

def random_under(data, y, samp_method = "balance", drop_na_col = True, drop_na_row = True, replacement = False, manual_perc = False, perc_o = -1,
  rel_thres = 0.5, rel_method = "auto", rel_xtrm_type = "both", rel_coef = 1.5, rel_ctrl_pts_rg = None):

Function designed to help solve the problem of imbalanced data for regression; RU under-samples the majority class.

Parameters:

main arguments / inputs:

data: pandas dataframe, the training set.

y: string, response variable y by name. It should be a header name found in the dataframe data.

samp_method: { ‘balance’ , ‘extreme’ }, default = ‘balance’ , specified method to determine over / under sampling percentage.

drop_na_col: bool, default = ‘True’ , if ‘True’ , auto drop columns with nan's.

drop_na_row: bool, default = ‘True’ , if ‘True’ , auto drop rows with nan's.

replacement: bool, default = ‘False’ , whether the sample is with or without replacement.

manual_perc: user defines percentage of under-sampling

perc_o: percentage of under-sampling that user defines

phi relevance function arguments / inputs:

rel_thres: positive real, default = 0.5, define the relevance threshold considered rare in phi relevance function.

rel_method: { ‘auto’ , ‘manual’ }, default = ‘auto’ , the relevance method in phi relevance function.

rel_xtrm_type: { ‘low’ , ‘high’ , ‘both’ }, default = ‘both’ , distribution focus on high, low or both.

rel_coef: positive real, default = 1.5, coefficient for box plot in phi relevance function to consider rare.

rel_ctrl_pts_rg: 2d array, default = None, when rel_method = ‘manual’ , it inputs for "manual" rel method.

TomekLinks

def tomeklinks(data, y, option = “majority” , drop_na_col = True, drop_na_row = True, rel_thres = 0.5, rel_method = "auto", rel_xtrm_type = "both", rel_coef = 1.5, rel_ctrl_pts_rg = None):

Function designed to help solve the problem of imbalanced data for regression. TomekLinks over-samples the minority class.

Parameters:

main arguments / inputs:

data: pandas dataframe, the training set.

y: string, response variable y by name. It should be a header name found in the dataframe data.

option: { ‘majority’ , ‘minority’ , ‘both’ }, default = ‘majority’ . Sampling information to sample the data set.

‘majority’ : resample only the majority class;

‘minority’ : resample only the minority class;

‘both’ : resample both majority and minority class.

drop_na_col: bool, default = ‘True’ , if ‘True’ , auto drop columns with nan's.

drop_na_row: bool, default = ‘True’ , if ‘True’ , auto drop rows with nan's.

phi relevance function arguments / inputs:

rel_thres: positive real, default = 0.5, define the relevance threshold considered rare in phi relevance function.

rel_method: { ‘auto’ , ‘manual’ }, default = ‘auto’ , the relevance method in phi relevance function.

rel_xtrm_type: { ‘low’ , ‘high’ , ‘both’ }, default = ‘both’ , distribution focus on high, low or both.

rel_coef: positive real, default = 1.5, coefficient for box plot in phi relevance function to consider rare.

rel_ctrl_pts_rg: 2d array, default = None, when rel_method = ‘manual’ , it inputs for "manual" rel method.