**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.