Random Under Sampler

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

Random Under Sampler under-samples the majority class by randomly picking samples with or without replacement.

Function designed to help solve the problem of imbalanced data for regression.

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 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: bool, user defines percentage of under-sampling

perc_u: float, percentage of under-sampling that user defines

phi relevance function arguments / inputs:

rel_thres: float, positive real number, 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: float, positive real number, 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.

References:

Branco, P., Torgo, L., Ribeiro, R. (2017). SMOGN: A Pre-Processing Approach for Imbalanced Regression. Proceedings of Machine Learning Research, 74:36-50. http://proceedings.mlr.press/v74/branco17a/branco17a.pdf.

Branco, P., Torgo, L., & Ribeiro, R. P. (2019). Pre-processing approaches for imbalanced distributions in regression. Neurocomputing, 343, 76-99. https://web.cs.dal.ca/~branco/PDFfiles/j14.pdf

Kunz, N. (2019). SMOGN: Synthetic Minority Over-Sampling for Regression with Gaussian Noise (Version 0.1.0). Python Package Index. https://pypi.org/project/smogn.