Group-disjunct and stratified data partitioning for machine learning

Uwe Reichel, audEERING GmbH, Gilching, Berlin
ureichel@audeering.com

1 Introduction

A common task in partitioning a machine learning dataset is to obtain a split that at the same time

1. is **disjunct** with respect to a selected grouping variable like *speaker ID*, and
2. yields an optimal **stratification** for a target variable (e.g. *emotion class*) as well as for one or more further grouping variables (e.g. *gender*, *mother tongue*)

This article introduces the Python package `splituils` developed at *audEERING GmbH* that offers one general solution how to achieve these two goals for categorical variables. Furthermore, this package provides several possibilities to bin numeric variables so that they can be used for disjunct group-splitting and stratification. The package is available in the PyPI software repository:

```
$ pip install splitutils
```

To download the source code, please visit the GitHub page [2].

In the following two sections we will describe `splitutils`' approaches for split optimization and for binning of one or many numeric variables.

2 Split optimization

Finding the optimal split is simply done by brute force optimization. We create k group-disjunct splits and select the split among all candidates that minimizes the Jensen-Shannon divergence between underlying and obtained class distributions as well as the distance between intended and obtained partition sizes. For creating group-disjunct splits we use the `GroupShuffleSplit()` class from `scikit-learn` [1].

3 Numeric variable treatment

Numeric target or other stratification variables need to be binned in order to use them for stratification. With `splitutils` variables can be binned in isolation or in combination.

Binning of single variables For single variables binning can be done either intrinsically into n classes based on an equidistant percentile split, or extrinsically by means of a list of user defined lower boundary values.

Joint binning of multiple variables Multiple variables can be intrinsically binned into a single categorical variable with n levels by KMeans clustering for which we use the `KMeans` class of `scikit-learn` [1]. Before clustering all variables are centered and scaled with the `StandardScaler` [1].

Please find several minimal examples for partitioning and binning on the GiHub project page [2].

References

[1] Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay: *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research, 12:2825–2830, 2011.

[2] Reichel, U.: *splitutils.* GitHub project page, 2023. `https://github.com/reichelu/splitutils`.