Package 'tgmMultiClass'

October 27, 2014

Type Package

Title Wrap functions for multi-class classification

Version 0.0.1

Author Thiago Guerrera Martins

Maintainer Thiago Guerrera Martins <thigm85@gmail.com>

Description Wrap functions for multi-class classification

License Whatever

Depends VGAM

Suggests testthat

Imports methods

LazyLoad yes

R topics documented:

`generateTestIndexes` *NA*

Description

Generate índexes necessary to evaluate models

Usage

```
generateTestIndexes(dataset, target_names, type = "3way", options, ...)
```

mcGet *Generic function to extract object elements*

Description

Standard function to be used when extracting information from objects.

Usage

```
mcGet(x, attr, ...)
```

mcGet.multiClass *Extract info from* multiClass *objects*

Description

Extract info from multiClass objects

Usage

```
## S3 method for class multiClass
mcGet(x, attr, i = NULL)
```

predict_r_vgam *multi-class prediction probability with 'vgam'*

Description

predict_r_vgam apply vgam function to different training and test sets.

Usage

```
predict_r_vgam(data, training, validation, test, ...)
```

Arguments

data	A data.frame containing the response variable and the features used for fitting and predictions.
training	A matrix where each column contain integers that indicate which lines of 'data' will be used for training.
validation	Similar to training but contain the lines used for validation, in case there are tuning parameters.
test	Similar to training but contain the lines used for test.
...	Arguments to be used in the vgam function.

Details

Different datasets are formed from `data` according to the indexes contained in `training`, `validation` and `test`. `predict_r_vgam` will then train with the training and validation part of the data using `vgam` and output the predicted probabilities on the test sets.

Value

It returns an S3 object of class c("multiClass_vgam", "multiClass"). This object contains the predicted probabilities for each test set. See the examples to get more info about how to extract the desired info from this object.

Examples

```
data(soccer_game)
indexes <- generateTestIndexes(dataset = soccer_game,
                               target_names = c("home.win", "home.draw", "home.lose"),
                               type = "3way",
                               options = list(prop_v = 0.2,
                                              prop_test = 0.2,
                                              number_replicates = 4))
pred_obj <- predict_r_vgam(data = soccer_game,
                           training = indexes$training,
                           validation = indexes$validation,
                           test = indexes$test,
                           formula = cbind(home.win, home.draw, home.lose) ~ 1 + fair.odd.home + fair.odd.
                           family = "multinomial")

all_predictions <- mcGet(pred_obj, "prob")
second_replication_only <- mcGet(pred_obj, "prob", 2)
```

Index