Package ‘fasi’

August 2, 2021

Title Fair Adjusted Selective Inference
Version 1.0.0
Description The Fair Adjusted Selective Inference (FASI) procedure as described in the paper "A Fairness-Adjusted Selective Inference Framework For Classification" By: Bradley Rava, Wen Sun, Gareth James and Xin Tong (2021). The package includes a function that preforms the FASI procedure and returns both the r-scores and classification for new observations.
License GPL-3
Encoding UTF-8
LazyData true
RoxygenNote 7.1.0
Depends R (>= 3.5.0), naivebayes, fastAdaboost, gam, stats
Suggests testthat, knitr, rmarkdown, dplyr
VignetteBuilder knitr
NeedsCompilation no
Author Bradley Rava [aut, cre],
Wenguang Sun [ctb],
Xin Tong [ctb],
Gareth James [ctb]
Maintainer Bradley Rava <brava@marshall.usc.edu>
Repository CRAN
Date/Publication 2021-08-02 08:40:02 UTC

R topics documented:

adult ................................................. 2
compas_data ........................................ 3
fasi .................................................. 4
predict.fasi ........................................ 5
rscore ................................................ 6

Index 8
2018 Census data from the UCI repository.

Description

Goal is to predict whether income exceeds $50K/yr based on census data. Also known as "Census Income" dataset

Usage

adult

Format

A data frame with 48842 rows and 14 variables:

- **age** continuous
- **fnlwgt** continuous
- **education** Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
- **education-num** continuous
- **relationship** Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
- **race** White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
- **sex** Female, Male.
- **capital-gain** continuous
- **capital-loss** continuous
- **hours-per-week** continuous
- **native-country** United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.
- **y** If an individuals salary is greater than $50K/yr

Source

https://archive.ics.uci.edu/ml/datasets/adult
### compas_data

**ProPublica’s Compas recidivism data**

**Description**

ProPublica’s Compas recidivism data

**Usage**

compas_data

**Format**

A data frame with 6172 rows and 17 variables:

- **age** numeric
- **c_charge_degree** c_charge_degree
- **race** race
- **age_cat** age category
- **score_text** compas score text
- **priors_count** number of priors
- **days_b_screening_arrest** numeric
- **decile_score** compas decile score
- **is_recid** is_recid
- **two_year_recid** two_year_recid
- **c_jail_in** c_jail_in
- **c_jail_out** c_jail_out
- **row_num** row number
- **score_prob** 1 divided by the decile score
- **is_recid_new** 2 if recidivated and 1 if they did not. This fits the notation of the FASI paper.
- **race_factor** race converted as a factor variable
- **sex** male or female

**Source**

https://github.com/propublica/compas-analysis
Create a ranking score model to implement the fasi classification algorithm.

Description

This function implements the Fair Adjusted Selective Inference method. It assumes that you have an observed data set that includes all variables needed for your ranking score model and class labels. The user is able to pick from a set of popular ML algorithms when estimating the ranking scores and is able to provide the algorithm their own model. If desired, the user can also directly provide their own ranking scores without using the functions pre-set algorithms. These will be directly used in the predict step when estimating the r-scores.

Usage

```r
fasi(
  observed_data,  # The observed data set that will be split into a testing and calibration data for you by proportion split_p - which is user-specified. If you are providing your own ranking scores, the observed data should just be the calibration data.
  model_formula,  # A formula that will be provided to a specified ML model used to produce ranking scores. Please be sure to follow the exact notation of each package and wrap your formula in the as.formula function.
  split_p = 0.5,   # The proportion of your observed data that should be used for the training data set.
  alg = "gam",    # A specified algorithm used to produce ranking scores. The options are "gam", "logit", "adaboost", "nonparametric_nb", and "user-provided".
  class_label = "y",  # The name of the class label variable in your data set. Defaults to "y".
  niter_adaboost = 10  # The number of weak learners you want to use for the adaboost algorithm. Defaults to 10. This parameter is useless if you did not select the adaboost algorithm.
)
```

Arguments

- **observed_data**: The observed data set that will be split into a testing and calibration data for you by proportion `split_p` - which is user-specified. If you are providing your own ranking scores, the observed data should just be the calibration data.
- **model_formula**: A formula that will be provided to a specified ML model used to produce ranking scores. Please be sure to follow the exact notation of each package and wrap your formula in the as.formula function.
- **split_p**: The proportion of your observed data that should be used for the training data set.
- **alg**: A specified algorithm used to produce ranking scores. The options are "gam", "logit", "adaboost", "nonparametric_nb", and "user-provided".
- **class_label**: The name of the class label variable in your data set. Defaults to "y".
- **niter_adaboost**: The number of weak learners you want to use for the adaboost algorithm. Defaults to 10. This parameter is useless if you did not select the adaboost algorithm.

Value

A list where the first element is the observed data with an extra variable denoting which observation was selected for the training and calibration data set, second is the model fit, third the training data, fourth the calibration data and lastly the chosen ranking score algorithm.
predict.fasi

Author(s)

Bradley Rava. PhD Candidate at the University of Southern California’s Marshall School of Business. Department of Data Sciences and Operations.

Examples

```r
fasi(observed_data, model_formula, split_p=0.5, alg="gam", class_label="y")
```

__predict.fasi  Prediction of a FASI Object__

Description

After a model is trained with the fasi function, predict estimates the r-scores and classification of all observations in the test data set.

Usage

```r
## S3 method for class 'fasi'
predict(
  object,
  test_data,
  alpha_1,
  alpha_2,
  rscore_plus = TRUE,
  ptd_group_var = "a",
  class_label = "y",
  ranking_score_calibrate,
  ranking_score_test,
  indecision_choice = "2",
  ...
)
```

Arguments

- `object`: An object of class fasi. It can be created from the fasi function.
- `test_data`: The test data set that contains new observations to be classified.
- `alpha_1`: User specified group and overall FSR control for class 1.
- `alpha_2`: User specified group and overall FSR control for class 2.
- `rscore_plus`: A logical variable that indicates if the r-score or r-score plus is calculated. By default the r-score plus is calculated.
- `ptd_group_var`: The name of the protected group variable in your data set. Defaults to "a".
- `class_label`: The name of the class label variable in your data set. Defaults to "y".
rscore

Calculate the r1 or r2 score for a new observation in ones test data.

Description

This function calculates an r-score for a given ranking score. It requires a calibration and testing dataset. Both the r-score+ and r-score can be implemented.

Usage

rscore(
  s_test_cur,
  y_class_cur,
  a_cur,
  z_cal,
)
rscore

z_test,
rscore_plus,
r2_indicator
)

Arguments

s_test_cur The current ranking score from the test data to be evaluated.
y_class_cur The class label you want to generate the r-scores for.
a_cur The current protected group from the test data to be evaluated.
z_cal The calibration data set.
z_test The test data set.
rscore_plus Logical variable, TRUE/FALSE, that determines if the r-score or r-score plus is calculated.
r2_indicator Logical variable, TRUE/FALSE, that determines if the r1 or r2 score is calculated.

Details

Do not call this function externally. It is only meant to be called from within the fasi function.

Value

The r-score corresponding to s_test_cur.

Author(s)

Bradley Rava. PhD Candidate at the University of Southern California’s Marshall School of Business. Department of Data Sciences and Operations.

Examples

rscore(s_test_cur, y_class_cur, a_cur, z_cal, z_test, rscore_plus, r2_indicator)
Index

* datasets
  adult, 2
  compas_data, 3

adult, 2
compas_data, 3
fasi, 4
predict.fasi, 5
rscore, 6