Package: sae.projection (via r-universe)

December 20, 2024

Type Package

Title Small Area Estimation Using Projection Methods

Version 0.1.0

Author c(person(``Ridson", ``Al Farizal P", role = c(``aut", ``cre",``cph"), email = ``ridsonap@bps.go.id", comment = c(ORCID = ``0000-0003-0617-0214")), person(``Azka", ``Ubaidillah", role = ``aut", email = ``azka@stis.ac.id", comment = c(ORCID = ``0000-0002-3597-0459")))

Maintainer Ridson Al Farizal <ridsonap@bps.go.id>

Description The sae.projection package provides a robust tool for small area estimation using a projection-based approach. This method is particularly beneficial in scenarios involving two surveys, the first survey collects data solely on auxiliary variables, while the second, typically smaller survey, collects both the variables of interest and the auxiliary variables. The package constructs a working model to predict the variables of interest for each sample in the first survey. These predictions are then used to estimate relevant indicators for the desired domains. This condition overcomes the problem of estimation in a small area when only using the second survey data.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

URL https://github.com/Alfrzlp/sae.projection

BugReports https://github.com/Alfrzlp/sae.projection/issues

Imports cli, doParallel, dplyr, methods, parsnip, recipes, rlang, rsample, stats, survey, tune, workflows, yardstick

RoxygenNote 7.3.2

Depends R (>= 4.3.0), tidymodels

Config/pak/sysreqs make libicu-dev

Repository https://alfrzlp.r-universe.dev

RemoteUrl https://github.com/alfrzlp/sae.projection

RemoteRef HEAD

RemoteSha 7f376b393dfc4dd106b2e74e836b34d0de7c0bdc

Contents

`df_svy22` *df_svy22: The August 2022 National Labor Force Survey dataset in East Java, Indonesia.*

Description

A dataset containing several auxiliary variable in East Java, Indonesia in 2023.

Usage

```
df_svy22
```

Format

A data frame with 74.070 rows and 11 variables with 38 domains.

PSU Primary Sampling Unit

WEIGHT Weight from survey

PROV province code

REGENCY regency/municipality code

STRATA Strata

income Income

neet Not in education employment or training status

sex sex (1: male, 2: female)

age age

disability disability status (0: False, 1: True)

edu last completed education

Source

https://www.bps.go.id

`df_svy23` *df_svy23: The August 2023 National Labor Force Survey dataset in East Java, Indonesia.*

Description

A dataset containing several auxiliary variable in East Java, Indonesia in 2023.

Usage

```
df_svy23
```

Format

A data frame with 66.245 rows and 11 variables with 38 domains.

PSU Primary Sampling Unit

WEIGHT Weight from survey

PROV province code

REGENCY regency/municipality code

STRATA Strata

income Income

neet Not in education employment or training status

sex sex (1: male, 2: female)

age age

disability disability status (0: False, 1: True)

edu last completed education

Source

https://www.bps.go.id

`projection` *Projection Estimator*

Description

The function addresses the problem of combining information from two or more independent surveys, a common challenge in survey sampling. It focuses on cases where:

- **Survey 1:** A large sample collects only auxiliary information.
- **Survey 2:** A much smaller sample collects both the variables of interest and the auxiliary variables.

The function implements a model-assisted projection estimation method based on a working model. The working models that can be used include several machine learning models that can be seen in the details section

Usage

```
projection(
  formula,
  id,
  weight,
  strata = NULL,
  domain,
  model,
  data_model,
  data_proj,
  model_metric,
  kfold = 1,
  grid = 1,
  parallel_over = "resamples",
  seed = 1,
  ...
)
```

Arguments

`formula`	An object of class formula that contains a description of the model to be fitted. The variables included in the formula must be contained in the `data_model` dan `data_proj`.
`id`	Column name specifying cluster ids from the largest level to the smallest level, where ~0 or ~1 represents a formula indicating the absence of clusters.
`weight`	Column name in data_proj representing the survey weight.
`strata`	Column name specifying strata, use NULL for no strata

domain	Column names in data_model and data_proj representing specific domains for which disaggregated data needs to be produced.
model	The working model to be used in the projection estimator. Refer to the details for the available working models.
data_model	A data frame or a data frame extension (e.g., a tibble) representing the second survey, characterized by a much smaller sample, provides information on both the variable of interest and the auxiliary variables.
data_proj	A data frame or a data frame extension (e.g., a tibble) representing the first survey, characterized by a large sample that collects only auxiliary information or general-purpose variables.
model_metric	A yardstick::metric_set(), or NULL to compute a standard set of metrics (rmse for regression and f1-score for classification).
kfold	The number of partitions of the data set (k-fold cross validation).
grid	A data frame of tuning combinations or a positive integer. The data frame should have columns for each parameter being tuned and rows for tuning parameter candidates. An integer denotes the number of candidate parameter sets to be created automatically.
parallel_over	A single string containing either "resamples" or "everything" describing how to use parallel processing. Alternatively, NULL is allowed, which chooses between "resamples" and "everything" automatically. If "resamples", then tuning will be performed in parallel over resamples alone. Within each resample, the preprocessor (i.e. recipe or formula) is processed once, and is then reused across all models that need to be fit. If "everything", then tuning will be performed in parallel at two levels. An outer parallel loop will iterate over resamples. Additionally, an inner parallel loop will iterate over all unique combinations of preprocessor and model tuning parameters for that specific resample. This will result in the preprocessor being re-processed multiple times, but can be faster if that processing is extremely fast.
seed	A single value, interpreted as an integer
...	Further argument to the svydesign.

Details

The available working models include:

- Linear Regression `linear_reg()`
- Logistic Regression `logistic_reg()`
- Poisson Regression `poisson_reg()`
- Decision Tree `decision_tree()`
- KNN `nearest_neighbor()`
- Naive Bayes `naive_bayes()`
- Multi Layer Perceptron `mlp()`
- Random Forest `rand_forest()`

- Accelerated Oblique Random Forests (Jaeger et al. 2022, Jaeger et al. 2024) `rand_forest(engine = 'aorsf')`
- XGBoost `boost_tree(engine = 'xgboost')`
- LightGBM `boost_tree(engine = 'lightgbm')`

A complete list of models can be seen at the following link Tidy Modeling With R

Value

The function returns a list with the following objects (`model`, `prediction` and `df_result`): `model` The working model used in the projection. `prediction` A vector containing the prediction results from the working model. `df_result` A data frame with the following columns:

- `domain` The name of the domain.
- `ypr` The estimation results of the projection for each domain.
- `var_ypr` The sample variance of the projection estimator for each domain.
- `rse_ypr` The Relative Standard Error (RSE) in percentage (%).

References

1. Kim, J. K., & Rao, J. N. (2012). Combining data from two independent surveys: a model-assisted approach. Biometrika, 99(1), 85-100.

Examples

```
## Not run:
library(sae.projection)
library(dplyr)

df_svy22_income <- df_svy22 %>% filter(!is.na(income))
df_svy23_income <- df_svy23 %>% filter(!is.na(income))

# Linear regression
lm_proj <- projection(
  income ~ age + sex + edu + disability,
  id = 'PSU', weight = 'WEIGHT', strata = 'STRATA',
  domain = c('PROV', 'REGENCY'),
  model = linear_reg(),
  data_model = df_svy22_income,
  data_proj = df_svy23_income,
)

# Random forest regression with hyperparameter tunning
rf_proj <- projection(
  income ~ age + sex + edu + disability,
  id = 'PSU', weight = 'WEIGHT', strata = 'STRATA',
  domain = c('PROV', 'REGENCY'),
  model = rand_forest(mtry = tune(), trees = tune(), min_n = tune()),
  data_model = df_svy22_income,
  data_proj = df_svy23_income,
  kfold = 3,
```

```
  grid = 30
)

df_svy22_neet <- df_svy22 %>% filter(between(age, 15, 24))
df_svy23_neet <- df_svy23 %>% filter(between(age, 15, 24))

# Logistic regression
lr_proj <- projection(
  formula = neet ~ sex + edu + disability,
  id = 'PSU',
  weight = 'WEIGHT',
  strata = 'STRATA',
  domain = c('PROV', 'REGENCY'),
  model = logistic_reg(),
  data_model = df_svy22_neet,
  data_proj = df_svy23_neet
)

# LightGBM regression with hyperparameter tunning
library(bonsai)
show_engines('boost_tree')
lgbm_model <- boost_tree(
    mtry = tune(), trees = tune(), min_n = tune(),
    tree_depth = tune(), learn_rate = tune(),
   engine = 'lightgbm'
)

lgbm_proj <- projection(
  formula = neet ~ sex + edu + disability,
  id = 'PSU',
  weight = 'WEIGHT',
  strata = 'STRATA',
  domain = c('PROV', 'REGENCY'),
  model = lgbm_model,
  data_model = df_svy22_neet,
  data_proj = df_svy23_neet,
  kfold = 3,
  grid = 30
)

## End(Not run)
```

Index