Package 'PPtreeregViz'

December 23, 2022

Title Projection Pursuit Regression Tree Visualization

Version 2.0.5

Maintainer HyunSun Cho <sunsmiling@naver.com>

Description

It was developed as a tool for exploring 'PPTreereg' (Projection Pursuit TREE of REGression). It uses various projection pursuit indexes and 'XAI' (eXplainable Artificial Intelligence) methods to help
understand the model by finding connections between the input variables and prediction values of the model.
The 'KernelSHAP' (Aas, Jullum and Løland (2019) <arXiv:1903.10464>) algorithm was modified to fit 'PPTreereg',
and some codes were modified from the 'shapr' package (Sellereite, Nikolai, and Martin Jullum (2020) <doi:10.21105/joss.02027>).
The implemented methods help to explore the model at the single instance level as well as at the whole dataset level.
Users can compare with other machine learning models by applying it to the 'DALEX' package of 'R'.

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.1

Depends R (>= 4.0.0)

NeedsCompilation yes

Repository CRAN

Date 2022-12-19

Imports Rcpp, data.table, DALEX, shapr, ggplot2, dplyr, tidyr, tibble, PPtreeViz, reshape2, magrittr, utils

Suggests testthat (>= 3.0.0), gridExtra, grid, ggExtra, partykit, ggparty, progress, tidyselect, ggforce, waterfalls, forcats, RColorBrewer, gtable, knitr, rmarkdown, MASS, covr

LinkingTo Rcpp, RcppArmadillo

VignetteBuilder knitr

URL https://github.com/sunsmiling/PPtreeregViz

Author Eun-Kyung Lee [aut, ctb],
HyunSun Cho [aut, cre],
Nikolai Sellereite [ctb, cph] (Author of included shapr fragments),
Martin Jullum [ctb, cph] (Author of included shapr fragments),
Annabelle Redelmeier [ctb, cph] (Author of included shapr fragments),
Norsk Regnesentral [cph]

Date/Publication 2022-12-23 19:20:05 UTC

R topics documented:

`dataXY` *Simulated data*

Description

The `dataXY` dataset is simulated data for running Projection Pursuit Regression Tree Model.

Usage

```
data(dataXY)
```

Format

A data frame with 100 rows and 4 variables.

Details

It contains 100 rows and 4 variables.

References

doi:10.3390/app11219885

---

decisionplot *Decision plot*

Description

decision plot for PPKernelSHAP

Usage

```
decisionplot(
  PPTreeregOBJ,
  testObs,
  final.rule = 5,
  method = "simple",
  varImp = "shapImp",
  final.leaf = NULL,
  Yrange = FALSE
)
```

Arguments

PPTreeregOBJ	PPTreereg class object - a model to be explained
testObs	test data observation
final.rule	final rule to assign numerical values in the final nodes. 1: mean value in the final nodes 2: median value in the final nodes 3: using optimal projection 4: using all independent variables 5: using several significant independent variables
method	simple or empirical method to calculate PPKernelSHAP
varImp	shapImp or treeImp - Sorted by descending order of variance or the variable importance from coefficient values of the nodes inside the PPTreereg.
final.leaf	location of final leaf
Yrange	show the entire final prediction range of the dependent variable. Default value is FALSE.

Details

Decision plots are mainly used to explain individual predictions that how the model makes decision, by focusing more on how model's predictions reach to their expected y value with PPKernelSHAP values.

Value

An object of the class ggplot

Examples

```
data(dataXY)
testX <- dataXY[1,-1]
Model <- PPTreereg(Y~., data = dataXY, DEPTH = 2)
decisionplot(Model, testX, final.rule =5, method="simple")
```

---

explain_PP *Make explain of* PPTreeregObj *for* DALEX *package*

Description

Create Model Explainer for PPTreereg

Usage

```
explain_PP(PPTreeregOBJ, data, y, final.rule,...)
```

Arguments

PPTreeregOBJ	PPTreereg class object - a model to be explained
data	data.frame or matrix - data that was used for fitting. If not provided then will be extracted from the model. Data should be passed without target column (this shall be provided as the y argument).
y	numeric vector with outputs / scores. If provided then it shall have the same size as data
final.rule	rule to calculate the final node value
...	arguments to be passed to methods

Details

This function creates a unified representation explain of PPTreereg model for cooperate with DALEX package.

Value

An object of the class explainer.

References

Explanatory Model Analysis. Explore, Explain and Examine Predictive Models. https://ema.drwhy.ai/

Examples

```
library("DALEX")
library("dplyr")
data(dataXY)
Model <- PPTreereg(Y~., data = dataXY, DEPTH = 2)
new_explainer <- explain_PP(Model, data = dataXY[,-1],y = dataXY[,1],final.rule= 5)
DALEX::model_performance(new_explainer) %>% plot(geom = "ecdf")
```

`feature_exact` *feature_exact*

Description

The original source for much of this came from 'shapr' package code in github.com/NorskRegnesentral/shapr/blob/master/R/f

Usage

```
feature_exact(m, weight_zero_m = 10^6)
```

Arguments

`m`	List. Contains vector of integers indicating the feature numbers for the different groups.
`weight_zero_m`	weight_zero_m

Details

Below is the original license statement for 'shapr' package.

Value

A data.table with all feature group combinations, shapley weights etc.

Author(s)

Nikolai Sellereite

References

The shapr package developed by Nikolai Sellereite, Martin Jullum, Annabelle Redelmeier, Norsk Regnesentral. doi:10.1016/j.artint.2021.103502 and modified some codes at https://github.com/NorskRegnesentral/shapr

---

insurance *Insurance Data*

Description

Dataset insurance is a part of dataset imported from insurance.csv in Kaggle "Medical Cost Personal Dataset". This data source material comes from Machine Learning with R by Brett Lantz book. It is simply come cleaned up and, it contains 1338 rows and 7 variables. These are:

Usage

```
data(insurance)
```

Format

a data frame with 1338 rows and 7 columns.

Details

- charges - Individual medical costs billed by health insurance.
- age - age of primary beneficiary.
- sex - insurance contractor gender, female, male.
- bmi - Body mass index, providing an understanding of body, weights that are relatively high or low relative to height, objective index of body weight (kg / m ^ 2) using the ratio of height to weight, ideally 18.5 to 24.9.
- children - Number of children covered by health insurance / Number of dependents.
- smoker - Smoking.
- region - the beneficiary's residential area in the US, northeast, southeast, southwest, northwest.

Source: https://www.kaggle.com/mirichoi0218/insurance

Source

The insurance.csv dataset was downloaded from the Kaggle site. The dataset was obtained from https://www.kaggle.com/mirichoi0218/insurance on May 11, 2021.

plot.PPimportance *Variable importance plot of* PPTreereg

Description

Visualize importance measure of trained PPTreereg model.

Usage

```
## S3 method for class 'PPimportance'
plot(x, marginal = FALSE, num_var = 5, ...)
```

Arguments

x	an importance object of the class PPimpobj, created with PPimportance function
marginal	plot global importance. Default value is FALSE.
num_var	number of variables to show.
...	arguments to be passed to methods

Details

To visualize the variable importance values of PPTreereg model, two types of plots are provided - importance of variables for each final node and global variable importance.

Value

An object of the class ggplot

Examples

```
data(dataXY)
Model <- PPTreereg(Y~., data = dataXY, DEPTH = 2)
Tree.Imp <- PPimportance(Model)
plot(Tree.Imp)
plot(Tree.Imp, marginal = TRUE)
```

plot.PPTreereg *PPTreereg plot*

Description

projection pursuit regression tree plot

Usage

```
## S3 method for class 'PPTreereg'
plot(x, font.size = 17, width.size = 1, ...)
```

Arguments

x	PPTreereg class object
font.size	font size of plot
width.size	size of eclipse in each node.
...	arguments to be passed to methods

Details

Draw projection pursuit regression tree with tree structure. It is modified from a function in party library.

Value

plot object

Examples

```
data(dataXY)
Model <- PPTreereg(Y~., data = dataXY, DEPTH = 2)
plot(Model)
```

PPimportance *Calculate variable importance*

Description

Calculate the importance of variables in the PPTreereg model. For local importance, weighted sum of projection coefficients with the number of data corresponding to each node as the weighted value in each node is used. The global importance is absolute sum of local importance.

Usage

```
PPimportance(PPTreeregOBJ,...)
```

Arguments

PPTreeregOBJ	PPTreereg class object - a model to be explained
...	arguments to be passed to methods

Value

An object of the class PPimpobj

Examples

```
data(dataXY)
Model <- PPTreereg(Y~., data = dataXY, DEPTH = 2)
PPimportance(Model)
```

PPregNodeViz *Node visualization*

Description

Visualize node in projection pursuit regression tree.

Usage

```
PPregNodeViz(PPTreeregOBJ,node.id,Rule=5)
```

Arguments

PPTreeregOBJ	PPTreereg class object - a model to be explained
node.id	node ID of inner or final node
Rule	split rule 1: mean of two group means 2: weighted mean of two group means - weight with group size 3: weighted mean of two group means - weight with group sd 4: weighted mean of two group means - weight with group se 5: mean of two group medians 6: weighted mean of two group medians - weight with group size 7: weighted mean of two group median - weight with group IQR 8: weighted mean of two group median - weight with group IQR and group size

Details

This function is developed for the visualization of inner and final nodes. Visual representation of the projection coefficient value of each node and the result of projected data help understand growth process of the projection pursuit regression tree. For the inner node, two plots are provided - the bar chart style plot with projection pursuit coefficients of each variable, the histogram of the projected data. For the final node, scatter plot of observed Y vs. fitted Y according to the final rules.

Value

An object of the class ggplot

Examples

```
data(dataXY)
Model <- PPTreereg(Y~., data = dataXY, DEPTH = 2)
PPregNodeViz(Model,node.id=1)
PPregNodeViz(Model,node.id=4)
```

PPregVarViz *Visualize independent variable action in projection pursuit regression tree.*

Description

This function is developed to see the influence of independent variables on the range of dependent variable.

Usage

```
PPregVarViz(PPTreeregOBJ,var.id,indiv=FALSE,
                    DEPTH=NULL,smoothMethod="auto", var.factor=FALSE)
```

Arguments

PPTreeregOBJ	PPTreereg class object - a model to be explained
var.id	independent variable name
indiv	TRUE: individual group plot, FALSE: combined one plot
DEPTH	depth for exploration
smoothMethod	method in geom_smooth function
var.factor	TRUE when indepedent variable is a categorical variable (as factor)

Value

An object of the class ggplot

Examples

```
data(dataXY)
Model <- PPTreereg(Y~., data = dataXY, DEPTH = 2)
PPregVarViz(Model,"X1")
PPregVarViz(Model,"X1",indiv = TRUE)
```

PPshapdependence *Dependency plot*

Description

Dependency plot using PPKernelSHAP

Usage

```
PPshapdependence(data_long, x, y=NULL, color_feature=NULL, smooth=TRUE)
```

Arguments

data_long	ppshapr_prep class object.
x	the independent variable to see
y	the interaction effect by putting the values of the independent variables in different colors.
color_feature	display other variables with color. Default value is NULL.
smooth	geom_smooth option. Default value is TRUE.

Details

Dependency plots are designed to show the effect of one independent variable on the model's prediction. Each point corresponds to each row of the training data, and the y axis corresponds the PPKernelSHAP value of the variable, indicating how much knowing the value of the variable changes the output of the model for the prediction of the data.

Value

An object of the class ggplot

Examples

```
data(dataXY)
testX <- dataXY[1,-1]
Model <- PPTreereg(Y~., data = dataXY, DEPTH = 2)
shap_long <- ppshapr_prep(Model, final.rule =5, method="simple")
PPshapdependence(shap_long,x = "X1")
```

`ppshapr.empirical` *Calculate* `PPKernelSHAP` *values with empirical methods*

Description

This function should only be called internally, and not be used as a stand-alone function. The original source for much of this came from 'shapr' package code in github.com/NorskRegnesentral/shapr/blob/master/R/prediction

Usage

```
ppshapr.empirical(PPTreeregOBJ, testObs, final.rule, final.leaf = NULL)
```

Arguments

`PPTreeregOBJ`	PPTreereg class object - a model to be explained
`testObs`	test data observation
`final.rule`	final rule to assign numerical values in the final nodes. 1: mean value in the final nodes 2: median value in the final nodes 3: using optimal projection 4: using all independent variables 5: using several significant independent variables
`final.leaf`	location of final leaf

Details

Below is the original license statement for 'shapr' package.

Value

List of empirical methods and model values

ppshapr.simple *Calculate* PPKernelSHAP *values with simple methods*

Description

This function should only be called internally, and not be used as a stand-alone function. The original source for much of this came from 'shapr' package code in github.com/NorskRegnesentral/shapr/blob/master/R/prediction

Usage

```
ppshapr.simple(PPTreeregOBJ, testObs, final.rule, final.leaf = NULL)
```

Arguments

PPTreeregOBJ	PPTreereg class object - a model to be explained
testObs	test data observation
final.rule	final rule to assign numerical values in the final nodes. 1: mean value in the final nodes 2: median value in the final nodes 3: using optimal projection 4: using all independent variables 5: using several significant independent variables
final.leaf	location of final leaf

Details

Below is the original license statement for 'shapr' package.

MIT License Copyright (c) 2019 Norsk Regnesentral Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Value

List of simple methods and model values

ppshapr_prep *Calculate* PPKernelSHAP *for all train data set*

Description

All train data set to calculate PPKernelSHAP

Usage

```
ppshapr_prep(PPTreeregOBJ = NULL, final.rule = 5, method = "simple")
```

Arguments

PPTreeregOBJ	PPTreereg class object - a model to be explained
final.rule	final rule to assign numerical values in the final nodes. 1: mean value in the final nodes 2: median value in the final nodes 3: using optimal projection 4: using all independent variables 5: using several significant independent variables
method	simple or empirical method to calculate PPKernelSHAP

Value

ppshapr_prep class object

Examples

```
data(dataXY)
testX <- dataXY[1,-1]
Model <- PPTreereg(Y~., data = dataXY, DEPTH = 2)
shap_long <- ppshapr_prep(Model, final.rule =5, method="simple")
```

PPshapsummary *Summary plot*

Description

Summary plot using PPKernelSHAP

Usage

```
PPshapsummary(data_long,...)
```

Arguments

data_long	ppshapr_prep class object.
...	arguments to be passed to methods

Details

A summary plot is used to see the aspects of important variables for each final node. The summary plot summarizes information about the independent variables that contributed the most to the model's prediction in the training data in the form of a density plot.

Value

An object of the class ggplot

Examples

```
data(dataXY)
testX <- dataXY[1,-1]
Model <- PPTreereg(Y~., data = dataXY, DEPTH = 2)
shap_long <- ppshapr_prep(Model, final.rule =5, method="simple")
PPshapsummary(shap_long)
```

PPTreereg *Construct the projection pursuit regression tree*

Description

Find regression tree structure using various projection pursuit indices in each split.

Usage

```
PPTreereg(formula,data,DEPTH=NULL,Rr=1,PPmethod="LDA",
                  weight=TRUE,lambda=0.1,r=1,TOL.CV=0.1,selP=NULL,
                  energy=0,maxiter=500,
                  standardized=TRUE,even=TRUE,space=0,
                  maxFinalNode=20,maxNodeN=10,...)
```

Arguments

formula	an object of class "formula"
data	data frame
DEPTH	depth of the projection pursuit regression tree
Rr	cutoff rule in each node
PPmethod	method for projection pursuit; "LDA", "PDA", "Lr", "GINI", and "ENTROPY".
weight	weight flag in LDA, PDA and Lr index
lambda	lambda in PDA index
r	r in Lr index
TOL.CV	CV limit for the final node

selP	number of variables for the final node in Method 5
energy	energy parameter
maxiter	number of maximum iteration
standardized	standardize each X variable before fitting the tree structure. Default value is TRUE
even	divide evenly at each node. Default value is TRUE
space	space between two groups of dependent variable
maxFinalNode	maximum number of final node
maxNodeN	maximum number of observations in the final node
...	arguments to be passed to methods

Value

Tree.result projection pursuit regression tree result with PPtreeclass object format

MSE mean squared error of the final tree

mean.G means of the observations in the final node

sd.G standard deviations of the observations in the final node.

coef.G regression coefficients for Method 3, 4 and 5

origY original dependent variable vector

origX.mean mean of original X

origX.sd standard deviation of original X

class.origX.mean means of the each independent variables in the final node

References

...

Examples

```
data(mtcars)
Tree.result <- PPTreereg(mpg~.,mtcars,DEPTH=2,PPmethod="LDA")
Tree.result
```

pp_ggparty *PPTreereg plot with independent variable*

Description

projection pursuit regression tree plot with independent variable

Usage

```
pp_ggparty(PPTreeregOBJ,ind_variable,final.rule=5,Rule=1, ...)
```

Arguments

PPTreeregOBJ	PPTreereg class object
ind_variable	independent variable to show
final.rule	final rule to assign numerical values in the final nodes. 1: mean value in the final nodes 2: median value in the final nodes 3: using optimal projection 4: using all independent variables 5: using several significant independent variables
Rule	split rule 1: mean of two group means 2: weighted mean of two group means - weight with group size 3: weighted mean of two group means - weight with group sd 4: weighted mean of two group means - weight with group se 5: mean of two group medians 6: weighted mean of two group medians - weight with group size 7: weighted mean of two group median - weight with group IQR 8: weighted mean of two group median - weight with group IQR and group size
...	arguments to be passed to methods

Details

Draw projection pursuit regression tree with independent variable. It is modified from a function in partykit library.

Value

An object of the class ggplot

Examples

```
data(dataXY)
Model <- PPTreereg(Y~., data = dataXY, DEPTH = 2)
pp_ggparty(Model, "X1", final.rule=5)
```

`predict.PPTreereg` *predict* `PPTreereg`

Description

predict projection pursuit regression tree

Usage

```
## S3 method for class 'PPTreereg'
predict(
  object,
  newdata = NULL,
  Rule = 1,
  final.rule = 1,
  classinfo = FALSE,
  ...
)
```

Arguments

`object`	a fitted object of class inheriting from `PPTreereg`
`newdata`	the test data set
`Rule`	split rule 1: mean of two group means 2: weighted mean of two group means - weight with group size 3: weighted mean of two group means - weight with group sd 4: weighted mean of two group means - weight with group se 5: mean of two group medians 6: weighted mean of two group medians - weight with group size 7: weighted mean of two group median - weight with group IQR 8: weighted mean of two group median - weight with group IQR and group size 9: cutoff that minimize error rates in each node
`final.rule`	final rule to assign numerical values in the final nodes. 1: mean value in the final nodes 2: median value in the final nodes 3: using optimal projection 4: using all independent variables 5: using several significant independent variables
`classinfo`	return final node information. Default value is FALSE
`...`	arguments to be passed to methods

Details

Predict class for the test set with the fitted projection pursuit regression tree and calculate prediction error.

Value

Numeric

Examples

```
data(dataXY)
Model <- PPTreereg(Y~., data = dataXY, DEPTH = 2)
predict(Model)
```

`print.PPTreereg` *Print PPTreereg result*

Description

Print PP.Tree.reg result

Usage

```
## S3 method for class 'PPTreereg'
print(
  x,
  tree.print = TRUE,
  coef.print = FALSE,
  cutoff.print = FALSE,
  verbose = TRUE,
  final.rule = 1,
  ...
)
```

Arguments

`x`	PPTreereg object
`tree.print`	print the tree structure when TRUE
`coef.print`	print the projection coefficient in each node when TRUE
`cutoff.print`	print the cutoff values in each node when TRUE
`verbose`	print if TRUE, no output if FALSE
`final.rule`	rule to calculate the final node value
`...`	arguments to be passed to methods

Details

Print the projection pursuit regression tree result

Value

tree print

shapley_weights *shapley_weights*

Description

The original source for much of this came from 'shapr' package code in github.com/NorskRegnesentral/shapr/blob/master/R/s
Below is the original license statement for 'shapr' package.

Usage

```
shapley_weights(m, N, n_components, weight_zero_m = 10^6)
```

Arguments

m	m
N	N
n_components	n_components
weight_zero_m	weight_zero_m

Details

Value

Numeric

Author(s)

Nikolai Sellereite

References

The shapr package developed by Nikolai Sellereite, Martin Jullum, Annabelle Redelmeier, Norsk Regnesentral. doi:10.1016/j.artint.2021.103502 and modified some codes at https://github.com/NorskRegnesentral/shapr

subpick *projection pursuit* submodular *pick algorithm* PP SP-LIME

Description

Pick several data containing various information for each final node for PPTreereg submodular Pick (SP-LIME) was developed (Ribeiro et al., 2016) to selects representative data with important information to determine the reliability of model based on the LIME algorithm. In order to extract data for each final node in the PPTreereg model, PP SP-LIME was proposed based on SP-LIME.

Usage

```
subpick(data_long, final.leaf, obsnum = 5)
```

Arguments

data_long	ppshapr_prep class object.
final.leaf	location of final leaf
obsnum	The number of budgets (instance to be selected). Default value is 1.

Value

Observation names and their original values as data

References

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 2016. doi:10.1145/2939672.2939778 https://github.com/marcotcr/lime/blob/master/lime/submodular_pick.py

Examples

```
data("dataXY")
Model <- PPTreereg(Y~., data = dataXY, DEPTH = 2)
shap_long=ppshapr_prep(Model,final.rule =3,method="simple")
subpick(shap_long,final.leaf = 1, obsnum = 5)
```

summary.PPTreereg *Summary* PPTreereg *result*

Description

summary PPTreereg result

Usage

```
## S3 method for class 'PPTreereg'
summary(object, c = NA, ...)
```

Arguments

object	a fitted object of class inheriting from PPTreereg
c	choose node id to summary. Default value is FALSE.
...	arguments to be passed to methods

Details

summary the projection pursuit regression tree result

Value

coefficient results of tree

waterfallplot *Waterfall plot*

Description

waterfall plot for PPKernelSHAP

Usage

```
waterfallplot(
  PPTreeregOBJ,
  testObs,
  final.rule = 5,
  method = "simple",
  final.leaf = NULL
)
```

Arguments

PPTreeregOBJ	PPTreereg class object - a model to be explained
testObs	test data observation
final.rule	final rule to assign numerical values in the final nodes. 1: mean value in the final nodes 2: median value in the final nodes 3: using optimal projection 4: using all independent variables 5: using several significant independent variables
method	simple or empirical method to calculate PPKernelSHAP
final.leaf	location of final leaf

Details

Waterfall plot is mainly used to explain individual predictions, and is suitable for showing an explanation when a single piece of data is entered as an input using PPKernelSHAP values.

Value

An object of the class ggplot

Examples

```
data(dataXY)
testX <- dataXY[1,-1]
Model <- PPTreereg(Y~., data = dataXY, DEPTH = 2)
waterfallplot(Model, testX, final.rule =5, method="simple")
```

---

weight_matrix *weight_matrix*

Description

The original source for much of this came from 'shapr' package code in github.com/NorskRegnesentral/shapr/blob/master/R/s
Below is the original license statement for 'shapr' package.

Usage

```
weight_matrix(X, normalize_W_weights = TRUE)
```

Arguments

X	X
normalize_W_weights	default is TRUE

Details

Value

Numeric matrix

Author(s)

Nikolai Sellereite

References

The `shapr` package developed by Nikolai Sellereite, Martin Jullum, Annabelle Redelmeier, Norsk Regnesentral. doi:10.1016/j.artint.2021.103502 and modified some codes at https://github.com/NorskRegnesentral/shapr

Index