Package 'APML'

June 27, 2021

Type Package

Title An Approach for Machine-Learning Modelling

Version 0.0.2

Description We include
1) data cleaning including variable scaling, missing values and unbalanced variables identification and removing, and strategies for variable balance improving;
2) modeling based on random forest and gradient boosted model including feature selection, model training, cross-validation and external testing.
For more information, please see Deng X (2021). <doi:10.1016/j.scitotenv.2020.144746>; H2O.ai (Oct. 2016). R Interface for H2O, R package version 3.10.0.8. <https://github.com/h2oai/h2o-3>; Zhang W (2016). <doi:10.1016/j.scitotenv.2016.02.023>.

License GPL-3

Encoding UTF-8

Imports tidyverse,h2o,performanceEstimation,dummies,dplyr,ggplot2,pROC,survival

NeedsCompilation no

Author Xinlei Deng [aut, cre, cph],
Wangjian Zhang [aut],
Shao Lin [aut]

Maintainer Xinlei Deng <xdeng3@albany.edu>

Repository CRAN

Date/Publication 2021-06-27 07:40:04 UTC

R topics documented:

APML 2
datatrans 3
expl_rr 5
outCI 6
splits_selection 7
uncertainty 8

Index **10**

APML *Develop models with grid search*

Description

Develop models with the optional parameters identified through the grid search and return model performance metrics. In order to run properly, the response column must be the first column and of a type of either numeric for "gaussian" or factor for "bernoulli" or "multinomial".

Usage

```
APML(model,AUC_stopping, xcol, traindata,testdata, hyper,
distribution,imbalance, sort_by, extra_data,stopping_metric)
```

Arguments

model	The model to be used. Currently, only allow "gbm" (default) for the gradient boosted tree, and "rf" for the random forest.
AUC_stopping	Logical. If TRUE, the combinations of the hyperparameters will be randomly searched with AUC metric-based early stopping. Default:FALSE.
xcol	A vector containing the names or indices of the predictors to be used.
traindata	The training dataset.
testdata	The testing dataset.
hyper	List of hyper parameters (i.e., list(ntrees=c(1,2), max_depth=c(5,7)))
distribution	Distribution of the outcome: "bernoulli" (default), "bernoulli", "quasibinomial", "multinomial", "gaussian", "poisson", "gamma", "tweedie", "laplace", "quantile", "huber" or "custom".
imbalance	Logical. If true, balancing the case numbers in the training data via over/under-sampling when developing the model. Default:FALSE
sort_by	Select the best model in the grid space by sorting with a metric. Choices are "logloss", "residual_deviance", "mse", "auc", "accuracy", "precision", "recall", "f1", etc
extra_data	Extra dataset for evaluating model performance.
stopping_metric	Metric to use for early stopping (AUTO: logloss for classification, deviance for regression and anonomaly_score for Isolation Forest). Must be one of: "AUTO", "deviance", "logloss", "MSE", "RMSE", "MAE", "RMSLE", "AUC", "AUCPR", "lift_top_group", "misclassification", "mean_per_class_error", "custom", "custom_increasing". Defaults to AUTO.

Details

This function uses the grid search technique to tune models' parameters and return the optimal model.

Value

bestmodel	Best H2o model via grid search
train_metrics	Model performance metrics based on the training data.
test_metrics	Model performance metrics based on the testing data.
summary	Summary of model performance.
extra_metrics	Model performance metrics based on extra data. Only available when "model_metric" is used.

Note

This function is based on h2o package. In order to run this function, we need to run h2o.init() before using this function. The response variable should be the first column.

References

LeDell E, Gill N, Aiello S, Fu A, Candel A, Click C, et al. 2019. h2o: R Interface for "H2O." Zhang W, Du Z, Zhang D, Yu S, Hao Y. 2016a. Boosted regression tree model-based assessment of the impacts of meteorological drivers of hand, foot and mouth disease in Guangdong, China. Sci Total Environ 553; doi:10.1016/j.scitotenv.2016.02.023.

Examples

```
library(h2o)
data(iris)
attach(iris)
h2o.init()
hyper <- list(ntrees=c(2,3,5))
iris <- iris[1:100,c(5,1:4)]
idx <- sample(100,50)
traindata <- iris[idx,]
testdata <- iris[-idx,]
xcol <- names(iris)[2:5]
results <- APML(xcol=xcol,hyper=hyper,
                traindata=traindata,testdata=testdata,
                sort_by ='auc',distribution = 'bernoulli')
h2o.shutdown(prompt=FALSE)
Sys.sleep(2)
```

---

datatrans *Data Cleaning*

Description

This function can help transform a bunch of numeric variables into factors or dummy variables, and change the reference for dichotomous variables. It can also drop constant colunms, variables with a great number of missing values and categorical variables with number of minority less than this ratio of number of target minority. For numeric variables, it can rescale the values.

Usage

```
datatrans(data,class_number,rescale,factor_dummy,ref, target,drop_ratio,missing_rate)
```

Arguments

data	A data.frame representing raw data needed for cleaning.
class_number	A integer representing numbers of unique categories for distinguishing categorical variables and continuous variables. Every variable with unique value greater than the number will be treated as continuous variable. Default:5
rescale	Logical. Whether or not to rescale continuous variables with Z-score scaling method. Default:False
factor_dummy	A character which could be "factor", "NULL" or "dummy". If "factor", categorical varaibles will be transformed into factors. If "dummy", it will create dummy variables for categorical variables. If "NULL", do nothing. Default: NULL
ref	Could be a number, "s" or "b". For dichotomous variables, this specifys the reference category. If a number, it will set the number as 0, the other as 1. If "s", it sets the smaller value as 0. If "b", it sets the bigger one as 0. Default:NULL, no changes.
target	A character representing the target variable. If give the target name, it will drop categorical variables with number of minority less than certian ratio of number of target minority.
drop_ratio	A number specifying the ratio for dropping categorical variables with number of minority less than this ratio of number of target minority. Only used if argument target is given. Default:0, not dropping.
missing_rate	A number specifying what ratio of missings in a variable, which should be dropped. Default:0.5.

Details

datatrans is only used for cleaning raw data. Raw data shouldn't contain any characters. Only numbers are permitted. Character information should be converted into numbers before use this function.

Value

A cleaned data.frame.

Note

After the data is cleaned, it is ready for modelling.

See Also

splits_selection, APML

Examples

```
library(survival)
data(lung)
attach(lung)
data = datatrans(lung,rescale=TRUE,factor_dummy = 'factor')
head(data)
str(data)
```

`expl_rr` *Explore the Risk Ratio with Cubic Spline*

Description

plot the changes of risk ratio of a risk factor in relation to the outcome using predictions from a general model, and identify the threshold.

Usage

```
expl_rr(data,formula,low=0.01,high=0.99,ref = 'min')
```

Arguments

`data`	A data.frame containing the risk factor and the outcome prediction probability from a model.
`formula`	Formula. Specify the outcome prediction and risk factor like pred~x
`low`	Set x scale limits
`high`	Set x scale limits
`ref`	Set reference. Using the smallest("min"), mean("mean"), median("median") or customized value of loess prediction

Details

For health data, if it is a cohor study, it will calculate the Risk Ratio. The risk ratio is calculated through following approaches. First, fit the risk factor and outcome prediction with loess regression, and get the smallest(mean, median, customized) value of loess prediction. Using the smallest(mean, median, customized) value from loess prediction as reference, the risk ratio = outcome prediction/the smallest value. Plot the risk factor and risk ratio with cubic spline.

Value

`p`	Cubic Spline plot.
`pred`	Loess model prediction.
`min_pred`	Reference value. The smallest value from loess prediction.
`threshold`	Threshold value for the plot. Identified by youden index

Note

For health data, if it is a cohor study, it will calculate the Risk Ratio.

Examples

```
library(h2o)
data(iris)
attach(iris)
h2o.init()
hyper <- list(ntrees=c(2,3,5))
iris <- iris[1:100,c(5,1:4)]
idx <- sample(100,50)
traindata <- iris[idx,]
testdata <- iris[-idx,]
xcol <- names(iris)[2:5]
results <- APML(xcol=xcol,hyper=hyper,
                traindata=traindata,testdata=testdata,
                sort_by ='auc',distribution = 'bernoulli')
data <- as.h2o(iris)
pred<- h2o.predict(results$bestmodel,newdata=data)
data <- h2o.cbind(data,pred)
data <- as.data.frame(data)
plots <- expl_rr(data,setosa~Sepal.Length,ref = 'mean')
plots$p
h2o.shutdown(prompt=FALSE)
Sys.sleep(2)
```

---

outCI *output CI in specific format.*

Description

Output CI in specific format.

Usage

```
outCI(x,l,h,n=2,type='OR')
```

Arguments

x	OR or RR.
l	lower or left side of confidence interval.
h	higher or right side of confidence interval.
n	number of digits.
type	output OR/RR or ER(excess risk).

Value

A character

Examples

```
outCI(8.601581,4.678212,12.524951)

##"8.60(4.68,12.52)"
```

`splits_selection` *Split dataset and select variables*

Description

Split dataset into training data and testing data and select variables based on relative importance.

Usage

```
splits_selection(data,split_ratio,split_seed,
feature_model,imbalance,nfolds,
RAN_type,RAN.seed,smote.seed,
xcol_enter,distribution)
```

Arguments

`data`	A data.frame used to build models
`split_ratio`	A numeric value indicating the ratio of total rows contained in each split. Must less than 1
`split_seed`	Random seed for splitting
`feature_model`	Name of model for feature selection. Currently, only allow "gbm" for gradient boosted tree, and "rf" for random forest
`imbalance`	Logical or "SMOTE"(for categorical response). True for balancing training data class counts via over/under-sampling when building the model. "SMOTE" for applying SMOTE and returning SMOTE training data.
`nfolds`	Number of folds for K-fold cross-validation. Default:5.
`RAN_type`	"both", "binominal" or "normal". "both" for generating both binominal and normal random terms for feature selection. "binominal" or "normal" only generate one specific type of random term. Categorical or continuous variables with relative importance greater than corresponding random term(s) will be selected.
`RAN.seed`	Random seed for random term(s)
`smote.seed`	Random seed for SMOTE. Only used if argument "imbalance"="SMOTE"
`xcol_enter`	A character vector of variables are required to enter the model, also called "forced entry". If xcol_enter contains all independent variables' names, it will not use random terms to select variables.
`distribution`	Distribution type. Must be one of: "AUTO", "bernoulli", "quasibinomial", "multinomial", "gaussian", "poisson", "gamma", "tweedie", "laplace", "quantile", "huber", "custom". Defaults to AUTO.

Details

This function applys a technique to use random term to select variables. We consider variables with relative importance greater than random term as truly important variables.

Value

`importance`	A data.frame containing the relative importance scores of selected variables.
`train_data`	Training dataset. If "imbalance"="SMOTE", it returns the SMOTE training set.
`test_data`	Testing dataset.
`raw_traindata`	Same training dataset. If "imbalance"="SMOTE", it returns the original training set before SMOTE.

Note

This function is based on h2o package. In order to run this function, we need to run h2o.init() before using this function. The response variable should be the first column.

Examples

```
library(survival)
library(h2o)
library(performanceEstimation)
data("lung")
attach(lung)
data <- datatrans(lung,factor_dummy = 'dummy',rescale = TRUE)
data <- data[,c(3,1,2,4:14)]
h2o.init()
selection <- splits_selection(data,imbalance = 'SMOTE')
h2o.shutdown(prompt=FALSE)
Sys.sleep(2)
```

---

uncertainty *Calculate the Uncertainty (95 percent confidence interval) of Risk Ratio Based on Threshold.*

Description

Calculate the uncertainty (95 percent confidence interval) of risk ratio with prediction from general models based on threshold.

Usage

```
uncertainty(x,y,th,ref=0)
```

Arguments

x	A vector representing the risk factor.
y	A vector representing the outcome prediction probability.
th	Threshold for dividing the risk facot into two groups.
ref	Reference indicator. If 0, set the group lower than threshold as reference. Default:0

Details

This is used to calculate the risk ratio, but not for odd ratio.

Value

Uncertainty (95 percent confidence interval) of risk ratio.

References

Díaz-Francés, E., Rubio, F.J. On the existence of a normal approximation to the distribution of the ratio of two independent normal random variables. Stat Papers 54, 309-C323 (2013).

Examples

```
library(h2o)
data(iris)
attach(iris)
h2o.init()
hyper <- list(ntrees=c(2,3,5))
iris <- iris[1:100,c(5,1:4)]
idx <- sample(100,50)
traindata <- iris[idx,]
testdata <- iris[-idx,]
xcol <- names(iris)[2:5]
results <- APML(xcol=xcol,hyper=hyper,
                traindata=traindata,testdata=testdata,
                sort_by ='auc',distribution = 'bernoulli')
data <- as.h2o(iris)
pred<- h2o.predict(results$bestmodel,newdata=data)
data <- h2o.cbind(data,pred)
data <- as.data.frame(data)
plots <- expl_rr(data,setosa~Sepal.Length,ref = 'mean')
uncertainty(data$Sepal.Length,data$setosa,plots$threshold)
h2o.shutdown(prompt=FALSE)
Sys.sleep(2)
```

Index

APML, 2

datatrans, 3

expl_rr, 5

outCI, 6

splits_selection, 7

uncertainty, 8