Package 'plslda5'

December 5, 2022

Type Package

Title What the Package Does (Title Case)

Version 0.1.0

Author c('ATTAL Pauline','DUBRULLE Pierre', 'SLALMI Ibtissam')

Maintainer The package maintainer <yourself@somewhere.net>

Description More about what it does (maybe more than one line)
Use four spaces when indenting paragraphs within the Description.

License What license is it under?

Imports dplyr, ggplot2, corrplot

Encoding UTF-8

NeedsCompilation no

R topics documented:

cercle_correlation.PLSDA *Plots*

Description

Plots

Usage

```
cercle_correlation.PLSDA(object, PC1, PC2)
```

Arguments

object	
PC1,	first principal component
PC2,	second principal component

Value

returns a graph which shows the projection of the individuals on 2 principal axis

Examples

```
data(iris)
formule = Species~.
object = plslda.fit(formula=formule, data=iris)

cercle_correlation.PLSDA(object=object, "PC1", "PC2")
plan_factoriel.PLSDA(object=object, "PC1", "PC2")
correlationplot.PLSDA(object=object,"PC1")
propz.PLSDA(object)
```

plslda.cv *plslda.cv, Cross validation for PLSLDA*

Description

This function allows the user to apply a k-fold cross validation on the data in order to determine the optimal number of components `ncomp` to keep the `ncomp` that we get from this cv function can also be used in `plslda.fit` if the parameter of the function is set to "CV"

Usage

```
plslda.cv(formula, data)
```

Arguments

formula	: this parameter is an object of class formula
data	: the data to perform the cross validation on and from which to get the samples

Value

This function returns a list object containing : ncomp the number of principal components (or latent variables) that should be used in the fit function, if not given by the user.
PRESS PRESS is a vector of PRESS values for each component.
min.PRESSthe minimum value of the PRESS vector that has been calculated, giving the number of principal components to keep.

Examples

```
data(iris)
plslda.cv(Species~., data = iris)
plslda.cv(Species~.,data=iris, nfold = 50)
```

---

plslda.dummies *plslda.dummies, Converting factor to dummy*

Description

plslda.dummies, Converting factor to dummy

Usage

```
plslda.dummies(y, mod = NA)
```

Arguments

y	the dataframe to convert into dummies
mod	parameter is set by default to NA, if the mod parameter is given, it is used in cross validation to calculate the PREdicted Sum Square (PRESS) in a way that it allows the matrix of test data and the matrix of predicted values to have the same number and order of classes of the target variable so that calculating the PRESS becomes possible.

Value

It returns a dataframe, with factors converted to dummies (categorical variable converted to numeric (0,1))

Examples

```
data(iris)
formule = Species~.

y <- as.factor(model.response(model.frame(formula=formule, data = iris)))

dummy <- plslda.dummies(iris$Species, mod=NA)
dummy_mod <-plslda.dummies(iris$Species, mod= y)
```

`plslda.fit` *plslda.fit, Function used to fit the PLS-LDA Regression model*

Description

plslda.fit, Function used to fit the PLS-LDA Regression model

Usage

```
plslda.fit <- function(formula, data, ncomp = 2, max.iter = 100,tol = 1e-06)
```

Arguments

`formula`	: used to select variable to predict according to predictive variables that can be selected
`data`	; the dataset or dataframe to work on
`ncomp`	: can either be defined with a certain number of principal components or defined as "CV" in order to call the cross validation function that would determine the number of components
`max.iter,`	is the maximum number of iterations while (diff > tol & iter <= max.iter)
`tol`	is the value that determines the convergence of the current weight and old weight, and which we put as paramater value of the nipals function which is called in this fit function

Value

a plsda object which is a list of different variables : `comp_X` matrix of principal components of X
`poid_X` matrix of weights of components of X
`comp_Y` matrix of principal components of Y
`poid_Y` matrix of weights of components of Y
`quality` the quality is measured with the coefficient of determination R^2 which provides information about how good our fit is with our model
`intercept_`
`coef_` classification function
`coef_cte` the classification function matrix binded with constants
`X.init` is the initial matrix X
`y`

Examples

```
1st example : where all the variables are used
fit(Species ~ ., data = iris, ncomp = 2,max.iter = 100,tol = 1e-06)
2nd example : only 2 variables are selected
fit(Species ~ Sepal.Length + Petal.Length, data = iris, ncomp = 2, max.iter = 100,tol = 1e-0
3rd example : using "CV" in ncomp parameter
fit(Species ~ Sepal.Length + Petal.Length, data = iris, ncomp = "CV", max.iter = 100,tol = 1
```

---

plslda.metrics — *report.plslda, function to calculate the different metrics of a prediction, to define the quality of the prediction based on the confusion matrix components, and the accuracy score*

Description

report.plslda, function to calculate the different metrics of a prediction, to define the quality of the prediction based on the confusion matrix components, and the accuracy score

Usage

```
plslda.metrics(y, ypred)
```

Arguments

y	the values of test data
ypred	the predicted values

Value

It returns : a 'summary' dataframe which contains the information of the confusion matrix as in the true positive, true negative, false positive, false negative the precision, recall and f1-score an 'accuracy' variable which gives the accuracy of the prediction

Examples

```
data(iris)
formula = Species~.

data_split = plslda.split_sample(formula=formule, data=iris)

object = plslda.fit(formula=formula, data=data_split$train)

ypred = plslda.predict(object=object,newdata=data_split$Xtest)

metrics <- plslda.metrics(y=data_split$ytest, ypred=ypred)
print(metrics$summary)
print(metrics$accuracy)
```

`plslda.nipals` *plslda.nipals, Nipals*

Description

nipals is an internal function used in the fit function

Usage

```
plslda.nipals(X, y, ncomp, max.iter, tol)
```

Arguments

`X`	entry matrix containing observations and variables
`y`	the target variable
`ncomp,`	number of components
`max.iter,`	maximum number of iterations until convergence
`tol,`	tol is the value that determines the convergence of the current weight and old weight, and which we put as parameter value of the nipals function which is called in this fit function

Value

It returns a plsda object as a list which contains : `comp_X` matrix of latent variables of X
`poid_X` the weight matrix of the principal components of X
`comp_Y` matrix of latent variables of Y
#' `poid_Y` the weight matrix of the principal components of Y
#' `Y.iter` matrix containing the scores of the target variable
#' `quality` used for display in summary function

Examples

```
data(iris)
formula = Species~.
X <- as.matrix(model.matrix(formula=formule, data = iris)[,-1])
y <- as.factor(model.response(model.frame(formula=formule, data = iris)))
ydum <- plslda.dummies(y)

plslda.nipals(X=X, y=ydum, ncomp = 2, max.iter = 100, tol = 1e-06)
```

`plslda.predict` *plslda.predict, The function used to predict in our model*

Description

plslda.predict, The function used to predict in our model

Usage

```
plslda.predict(ObjectPLSDA,newdata)
```

Arguments

`newdata,`	optional parameter : if given, it is a dataframe used to select the predictive variables If not, the fitted values are used.
`plsda`	object, result of the fit method

Value

the function returns pred_, which is a vector containing, for each individual of the matrix X, the class name of the highest probability of belonging to a the target variable Y class (modality)

Examples

```
#' data(iris)
formula = Species~.

data_split = plslda.split_sample(formula=formule, data=iris)

object = plslda.fit(formula=formule, data=data_split$train)

ypred = plslda.predict(object=object,newdata=data_split$Xtest)
```

`plslda.scale` *plslda.scale, Function to standardize vectors and matrixes (center and reduce)*

Description

plslda.scale, Function to standardize vectors and matrixes (center and reduce)

Usage

```
plslda.scale(X, center = TRUE, scale = TRUE)
```

Arguments

X	the given matrix must contain numeric values
center	set to TRUE by default, in order to center the matrix data
scale	set to TRUE by default, in order to reduce the matrix data

Value

It returns a dataframe which is the result of the centered and reduced matrix, standardized data

Examples

```
data(iris)
scale.t1<-plslda.scale(iris[,-5],center=TRUE, scale=TRUE)
scale.t2<-plslda.scale(iris[,-5],center=TRUE, scale=FALSE)
```

---

plslda.split_sample

plslda.split_sample, This function is used to split data into a train and a test sets

---

Usage

```
plslda.split_sample(formula, data, train_size = 0.7)
```

Arguments

formula,	used to select the predictive variables
data,	the data to split into train and test sets
train_size,	is set to 0.7 by default (70

It returns train-test splits of inputs

plslda.split_sample, This function is used to split data into a train and a test sets

data(iris) formule = Species~. split_sample.t1<-plslda.split_sample(formula=formule, data=iris) split_sample.t2<-plslda.split_sample(formula=formule, data=iris, iris,0.5) train1 = split_sample.t1$train Xtrain1 = split_sample.t1$Xtrain ytrain1 = split_sample.t1$ytrain Xtest1 = split_sample.t1$Xtest ytest1 = split_sample.t1$ytest

`print.PLSDA` *print.PLSDA, Function used to print classification function and latent vectors*

Description

print.PLSDA, Function used to print classification function and latent vectors

Usage

```
## S3 method for class 'PLSDA'
print(object, ...)
```

Arguments

a	plslda object

Value

prints the classification function of matrix X which is obtained by the combination PLS-LDA method

prints the latent vectors of matrix x

Examples

```
data(iris)
formule = Species~.

PLSDA_object = fit(formula=formule, data=iris)
print(PLSDA_object)
```

`sel.forward` *sel.forward, The function used for variables selection, following the forward method*

Description

sel.forward, The function used for variables selection, following the forward method

Usage

```
sel.forward(formula, data, slentry = 0.01, verbose=FALSE)
```

Arguments

formula,	used to select the predictive variables
data,	dataset with many numeric variables
slentry,	if the p value of the F test stat is lower than slentry, we stop adding variables
verbose,	is set to FALSE by default. If set to TRUE, it prints the intermediate tables at each step

Details

This approach carries out a selection of variables before applying the model. It adds variables one by one, and uses the Wilks lambda which helps determine if a variable contributes significantly through its total dispersion. which means, in each forward step, the one variable that is added is the ones that gives the single best improvement to your model. The idea is to keep only the variables that contribute significantly to the distance between the barycentres (class centers).

Value

It returns a dataframe with individuals and only the selected variables

Examples

```
data(iris)
formule = Species~
sel <- sel.forward(formula=formule, data=iris, slentry = 0.01, verbose=FALSE)
```

---

summary.PLSDA *summary.PLSDA, This function prints the standardized projection coefficients of matrix X*

Description

summary.PLSDA, This function prints the standardized projection coefficients of matrix X

Usage

```
## S3 method for class 'PLSDA'
summary(object, ...)
```

Arguments

a	plsda object

Value

prints the standardized (centered and reduce) projection coefficients of matrix X

prints the restitution quality of the pls regression

Examples

```
data(iris)
formule = Species~.

PLSDA_object = fit(formula=formule, data=iris)
summary(PLSDA_object)
```

Index