Package: rcppmlpackexamples (via r-universe)

May 27, 2026

Type Package

Title Example Use of 'mlpack' from C++ via R

Version 0.0.1.1

Date 2025-10-28

Description A Minimal Example Package which demonstrates 'mlpack' use via C++ Code from R.

URL https://github.com/eddelbuettel/rcppmlpack-examples

BugReports https://github.com/eddelbuettel/rcppmlpack-examples/issues

License GPL (>= 2)

Suggests tinytest

Depends R (>= 3.5.0)

Imports Rcpp (>= 1.1.0)

LinkingTo Rcpp, RcppArmadillo (>= 15.0.2-1), RcppEnsmallen, mlpack (>= 4.6.3)

Encoding UTF-8

Config/roxygen2/version 8.0.0

RoxygenNote 6.0.1

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

Date/Publication 2026-05-27 13:11:19 UTC

RemoteUrl https://github.com/eddelbuettel/rcppmlpack-examples

RemoteRef HEAD

RemoteSha 792161db77f02c1c406d5b9bfd72267a67a04592

Contents

`rcppmlpackexamples-package` *Example Use of 'mlpack' from C++ via R*

Description

A Minimal Example Package which demonstrates 'mlpack' use via C++ Code from R.

Package Content

Index: This package was not yet installed at build time.

Maintainer

Dirk Eddelbuettel <edd@debian.org>

Author(s)

Dirk Eddelbuettel [aut, cre], Authors of mlpack [aut], Constantinos Giachalis [ctb]

`adaBoost` *An AdaBoost classification*

Description

Run AdaBoost using a simple Perceptron model as the weak learner

Usage

```
adaBoost(dataset, labels, iterations = 100L, tolerance = 2e-10,
  perceptronIter = 400L)
```

Arguments

`dataset`	A matrix of explanatory variables, i.e. "features"
`labels`	A vector of the dependent variable as integer values, i.e. "labels"
`iterations`	An integer value for the number of iterations
`tolerance`	A double with the desired tolerance
`perceptronIter`	An integer value for the number of a iterations for the weak learner

Value

A list object

Examples

```
data(iris)
X <- t(as.matrix(iris[,1:4]))
y <- as.integer(iris[,5]) - 1   # mlpack prefers {0, 1, 2}
adaBoost(X, y)
```

`covertype_small` *Covertype data subset used for classification*

Description

A subset of the UCI machine learning data set 'covertype' describing cloud coverage in seven different states of coverage. This smaller subset contains with 100,000 observations and 55 variables. The first 54 variables are explanatory (i.e. "features"), with the last providing the dependent variable ("labels". The data is in the 'wide' 55 x 100,000 format used by **mlpack**. The dependent variable has been transformed to the range zero to six by subtracting one from the values found in the data file.

Details

The original source of the data is the US Forest Service, and the complete file is part of the UC Irvince machine learning data repository.

Source

https://www.mlpack.org/datasets/covertype-small.csv.gz

References

https://archive.ics.uci.edu/dataset/31/covertype

datasetExample *Simple example of loading categorical data via 'mlpack'*

Description

Simple example of loading categorical data via 'mlpack'

Usage

```
datasetExample()
```

Value

Nothing is returned, the function is invoked for its side effect.

decisionTree *Run a decisionTree classification*

Description

Run decisionTree classifier

Usage

```
decisionTree(dataset, labels, pct = 0.3, min_leaf_size = 10L,
  minimum_gain_split = 1e-07, maximum_depth = 0L)
```

Arguments

dataset	A matrix of explanatory variables, i.e. "features"
labels	A vector of the dependent variable as integer values, i.e. "labels"
pct	A numeric value for the percentage of data to be retained for the test set
min_leaf_size	An integer value with the minimum number of elements per leaf
minimum_gain_split	A double with the gain needed to further split the tree
maximum_depth	An integer with the maximum tree depth, default zero means unlimited

Value

A list object

Examples

```
data(iris)
X <- t(as.matrix(iris[,1:4]))
y <- as.integer(iris[,5]) - 1   # mlpack prefers {0, 1, 2}
decisionTree(X, y)
```

kMeans *Run a k-means clustering analysis*

Description

Run a k-means clustering analysis, returning a list of cluster assignments

Usage

```
kMeans(data, clusters)
```

Arguments

data	A matrix of data values
clusters	An integer specifying the number of clusters

Details

This function performs a k-means clustering analysis on the given data set.

Value

A list with cluster assignments

Examples

```
x <- rbind(matrix(rnorm(100, sd = 0.3), ncol = 2),
           matrix(rnorm(100, mean = 1, sd = 0.3), ncol = 2))
colnames(x) <- c("x", "y")
cl <- kMeans(x, 2)

data(trees, package="datasets")
cl2 <- kMeans(t(trees),3)
```

linearRegression *Run a linear regression with optional ridge regression*

Description

Run a linear regression (with optional ridge regression)

Usage

```
linearRegression(matX, vecY, lambda = 0, intercept = TRUE)
```

Arguments

matX	A matrix of explanatory variables ('predictors') in standard R format (i.e. 'tall and skinny' to be transposed internally to MLPACK format (i.e. 'short and wide').
vecY	A vector of dependent variables ('responses')
lambda	An optional ridge parameter, defaults to zero
intercept	An optional boolean switch about an intercept, default is true.

Details

This function performs a linear regression, and serves as a simple test case for accessing an MLPACK function.

Value

A vector with fitted values

Examples

```
suppressMessages(library(utils))
data("trees", package="datasets")
X <- with(trees, cbind(log(Girth), log(Height)))
y <- with(trees, log(Volume))
lmfit <- lm(y ~ X)
# summary(fitted(lmfit))
mlfit <- linearRegression(X, y)
# summary(mlfit)
all.equal(unname(fitted(lmfit)),  as.vector(mlfit))
```

---

loanData *Loan data subset used for default prediction*

Description

A four column data set containing a binary variable 'Employed' (with zero denoting unemployment and one employment), a numeric variable 'Bank Balance', a numeric variable 'Annual Salary' and a binary target variable 'Defaulted?' (with zero denoting loan repayment and one denoting default).

Details

The original source of the data is not documented by mlpack.

Source

https://datasets.mlpack.org/LoanDefault.csv

References

https://archive.ics.uci.edu/dataset/31/covertype

loanDefaultPrediction *loanDefaultPrediction*

Description

Predict loan default using a decision tree model

Usage

```
loanDefaultPrediction(loanDataFeatures, loanDataTargets, pct = 0.25)
```

Arguments

loanDataFeatures	A matrix of dimension 3 by N, i.e. transposed relative to what R uses, with the three explanantory variables
loanDataTargets	A vector of (integer-valued) binary variables loan repayment or default
pct	A numeric variable with the percentage of data to be used for testing, defaults to 25%

Details

This functions performs a loan default prediction based on three variables on employment, bank balance and annual salary to predict loan repayment or default

Value

A list object with predictions, probabilities, accuracy and a report matrix

Examples

```
data(loanData)
res <- loanDefaultPrediction(t(as.matrix(loanData[,-4])),  # col 1 to 3, transposed
                             loanData[, 4],                # col 4 is the target
                             0.25)                         # retain 25% for testing
str(res)
res$report
```

`logisticRegression` *Run logistic regression*

Description

Run a logistic regression returning classification

Usage

```
logisticRegression(data, labels, lambda = 0)
```

Arguments

`data`	A matrix of data values
`labels`	A vector of class labels
`lambda`	An optional L2 regularization parameter, defaults to zero

Details

This function performs a logistic regression on the given data set. The data set is synthetic and follows an on-line example source 2025-10-28 which gave no direct source (as an example provided by Google / Gemini); it is now included in the `examples` directory of the package

Value

A list with predictions, probabilities and parmeters

Examples

```
data(logisticRegression)
X <- as.matrix(logisticRegressionData[, 1:2])
y <- as.matrix(logisticRegressionData[, 3])
res <- logisticRegression(X, y)
res$parameters
```

`logisticRegressionData`
Logistic regression example data set

Description

A three column (synthetic) data set to illustrate logistic regression.

randomForest *Run a Random Forest classificatio*

Description

Run a Random Forest Classifier

Usage

```
randomForest(dataset, labels, pct = 0.3, nclasses = 7L, ntrees = 10L)
```

Arguments

dataset	A matrix of explanatory variables, i.e. "features"
labels	A vector of the dependent variable as integer values, i.e. "labels"
pct	A numeric value for the percentage of data to be retained for the test set
nclasses	An integer value for the number of a distinct values in labels
ntrees	An integer value for the number of trees

Details

This function performs a Random Forest classification on a subset of the standard 'covertype' data set

Value

A list object

See Also

covertype_small

Examples

```
data(covertype_small)                       # see help(covertype_small)
res <- randomForest(covertype_small[-55,],  # features (already transposed)
                    covertype_small[55,],   # labels now in [0, 6] range
                    0.3)                    # percentage used for testing
str(res)  # accuracy varies as method is randomized but no seed set here
```

Index