Package ‘tools4uplift’

January 29, 2019

Version 0.1-1
Date 2019-01-28
Title Tools for Uplift Modeling
Author Mouloud Belbahri, Olivier Gandouet, Alejandro Murua, Vahid Partovi Nia
Maintainer Mouloud Belbahri <mouloud.belbahri@gmail.com>
Description Uplift modeling aims at predicting the causal effect of an action such as a medical treatment or a marketing campaign on a particular individual, by taking into consideration the response to a treatment. In order to simplify the task for practitioners in uplift modeling, we propose a combination of tools that can be separated into the following ingredients: i) quantization, ii) visualization, iii) feature engineering, iv) feature selection and, v) model validation. For more details, please read Belbahri et Al. (2019) <https://dms.umontreal.ca/~murua/research/UpliftRegression.pdf>.

Depends R (>= 3.1.2)
Imports dplyr, glmnet
License GPL-2 | GPL-3
Repository CRAN
Repository/R-Forge/Project tools4uplift
Repository/R-Forge/Revision 31
Repository/R-Forge/Date/TimeStamp 2019-01-29 02:47:23
Date/Publication 2019-01-29 15:30:03 UTC

R topics documented:

- tools4uplift-package ............................................................................. 2
- BestFeatures ......................................................................................... 3
- BinUplift ............................................................................................... 4
- BinUpliftEnhanced ............................................................................... 6
- DualPredict .......................................................................................... 7
- DualUplift ............................................................................................. 8
- InterPredict .......................................................................................... 9
tools4uplift-package  Tools for Uplift Modeling

Description

Uplift modeling aims at predicting the causal effect of an action such as a medical treatment or a marketing campaign on a particular individual, by taking into consideration the response to a treatment. In order to simplify the task for practitioners in uplift modeling, we propose a combination of tools that can be separated into the following ingredients: i) quantization, ii) visualization, iii) feature engineering, iv) feature selection and, v) model validation. For more details, please read Belbahri et Al. (2019) <https://dms.umontreal.ca/~murua/research/UpliftRegression.pdf>.

Details

The DESCRIPTION file:

Package: tools4uplift
Version: 0.1-1
Date: 2019-01-28
Title: Tools for Uplift Modeling
Author: Mouloud Belbahri, Olivier Gandouet, Alejandro Murua, Vahid Partovi Nia
Maintainer: Mouloud Belbahri <mouloud.belbahri@gmail.com>
Description: Uplift modeling aims at predicting the causal effect of an action such as a medical treatment and a marketing campaign on a particular individual, by taking into consideration the response to a treatment. In order to simplify the task for practitioners in uplift modeling, we propose a combination of tools that can be separated into the following ingredients: i) quantization, ii) visualization, iii) feature engineering, iv) feature selection and, v) model validation. For more details, please read Belbahri et Al. (2019) <https://dms.umontreal.ca/~murua/research/UpliftRegression.pdf>.
Depends: R (>= 3.1.2)
Imports: dplyr, glmnet
License: GPL-2 | GPL-3
Repository: R-Forge
Repository/R-Forge/Project: tools4uplift
Repository/R-Forge/Revision: 31
Repository/R-Forge/DateTimeStamp: 2019-01-29 02:47:23
Date/Publication: 2019-01-29 02:47:23

Index of help topics:

BestFeatures Feature selection for the interaction estimator
BestFeatures

BinUplift  Univariate quantization
BinUpliftEnhanced  Univariate quantization - augmented data
DualPredict  Predictions from a two-model estimator
DualUplift  Two-model estimator
InterPredict  Predictions from an interaction estimator
InterUplift  Interaction estimator
LassoPath  LASSO path for penalized logistic regression
QiniArea  Qini coefficient
QiniBarPlot  Uplift barplot
QiniCurve  Qini curve
QiniTable  Performance of an uplift estimator
SimUplift  Synthetic data for uplift modeling
SplitUplift  Split data with respect to uplift distribution
SquareUplift  Bivariate quantization
tools4uplift-package  Tools for Uplift Modeling

Author(s)
Mouloud Belbahri <mouloud.belbahri@gmail.com>, Olivier Gandouet, Alejandro Murua, Vahid Partovi Nia

BestFeatures  Feature selection for the interaction estimator

Description
Penalized logistic regression (LASSO) in order to select the features that maximize the Qini coefficient.

Usage
BestFeatures(data, treat, outcome, predictors, nb.lambda = 100, nb.group = 10, validation = FALSE, p = 0.3, value = FALSE)

Arguments

data  a data frame containing the treatment, the outcome and the predictors.
treat  name of a binary (numeric) vector representing the treatment assignment (coded as 0/1).
outcome  name of a binary response (numeric) vector (coded as 0/1).
predictors  a vector of names representing the predictors to consider in the model.
nb.lambda  the number of lambda values - Default is 100.
nb.group  the number of groups for computing the Qini coefficient - Default is 10.
validation  if TRUE, the best features are selected based on cross-validation - Default is FALSE.
if validation is TRUE, the desired proportion for the validation set. p is a value between 0 and 1 expressed as a decimal, it is set to be proportional to the number of observations per group - Default is 0.3.

value if TRUE, the values of the best lambda and Qini coefficient will be printed - Default is FALSE.

Details

The regularization parameter is chosen based on the interaction uplift model that maximizes the Qini coefficient. Using the LASSO penalty, some predictors have coefficients set to zero.

Value

a vector of names representing the selected best features from the penalized logistic regression.

Author(s)

Mouloud Belbahri

References


Examples

library(tools4uplift)
data("SimUplift")

features <- BestFeatures(data = SimUplift, treat = "treat", outcome = "y",
                          predictors = colnames(SimUplift[,3:7]))

features

 BinUplift  Univariate quantization

Description

Univariate optimal partitionning for Uplift Models. The algorithm quantizes a single variable into bins with significantly different observed uplift.

Usage

BinUplift(data, treat, outcome, x, n.split = 10, alpha = 0.05,
           n.min = 30, ylim = NULL, ylab = "Uplift",
           title = "Binning Results", color = NULL)
Arguments

data  a data frame containing the treatment, the outcome and the predictor to quantize.
treat name of a binary (numeric) vector representing the treatment assignment (coded as 0/1).
outcome name of a binary response (numeric) vector (coded as 0/1).
x  name of the explanatory variable to quantize.
n.split number of splits to test at each node. For continuous explanatory variables only (must be > 0).
alpha significance level of the statistical test (must be between 0 and 1).
n.min minimum number of observations per child node.
ylim a range for the y axis.
ylab a title for the y axis.
title an overall title for the plot.
color a color for the plot. If omitted, the color will be set by default to a custom light blue.

Value

out.tree Descriptive statistics for the different nodes of the tree
sas.code SAS code generated for variable quantization

Author(s)

Mouloud Belbahri

References


See Also

BinUpliftEnhanced

Examples

library(tools4uplift)
data("SimUplift")

binX1 <- BinUplift(data = SimUplift, treat = "treat", outcome = "y", x = "X1",
alpha = 0.10, n.min = 3, title = "Binning for X1")
**Description**

Univariate optimal partitionning for Uplift Models. The algorithm quantizes several variables into bins and creates an augmented dataset with the binned variables.

**Usage**

```r
BinUpliftEnhanced(data, treat, outcome, var.list, n.split = 10,
                    alpha = 0.05, n.min = 30, ylim = NULL,
                    ylab = "Uplift", title = "Binning Results",
                    color = NULL)
```

**Arguments**

- `data`: a data frame containing the treatment, the outcome and the predictor to quantize.
- `treat`: name of a binary (numeric) vector representing the treatment assignment (coded as 0/1).
- `outcome`: name of a binary response (numeric) vector (coded as 0/1).
- `var.list`: a vector of names representing the explanatory variables to quantize.
- `n.split`: number of splits to test at each node. For continuous explanatory variables only (must be > 0).
- `alpha`: significance level of the statistical test (must be between 0 and 1).
- `n.min`: minimum number of observations per child node.
- `ylim`: a range for the y axis.
- `ylab`: a title for the y axis.
- `title`: an overall title for the plot.
- `color`: a color for the plot. If omitted, the color will be set by default to a custom light blue.

**Value**

an augmented data frame with quantized variables. If a variable is enhanced, the function returns automatically a barplot.

**Author(s)**

Olivier Gandouet

**References**

**DualPredict**

**See Also**

BinUplift

**Examples**

```r
library(tools4uplift)
data("SimUplift")

train.enhanced <- BinUpliftEnhanced(data = SimUplift, treat = "treat", outcome = "y",
                                     var.list = colnames(SimUplift[,3:7]))
```

---

**Description**

Predictions from the two-model uplift model estimator with associated model performance.

**Usage**

```r
DualPredict(data, treat, outcome, model, nb.group = 10, plotit = FALSE)
```

**Arguments**

- `data`: a data frame containing the treatment, the outcome and the predictors.
- `treat`: name of a binary (numeric) vector representing the treatment assignment (coded as 0/1).
- `outcome`: name of a binary response (numeric) vector (coded as 0/1).
- `model`: a model that must be the output of DualUplift function.
- `nb.group`: number of groups of equal observations in which to partition the data in order to compute model performance.
- `plotit`: if TRUE, a QiniCurve and QiniBarPlot are returned.

**Value**

- `data`: a data frame augmented with the predicted uplift
- `qini`: a Qini Coefficient

**Author(s)**

Mouloud Belbahri
References


See Also

DualUplift

Examples

```r
library(tools4uplift)
data("SimUplift")
fit <- DualUplift(SimUplift, "treat", "y", predictors = colnames(SimUplift[, 3:12]))
pred <- DualPredict(SimUplift, "treat", "y", model = fit, nb.group = 5)[[1]]
```

---

**DualUplift**

*Two-model estimator*

**Description**

Fit the two-model uplift model estimator.

**Usage**

DualUplift(data, treat, outcome, predictors)

**Arguments**

- **data** a data frame containing the treatment, the outcome and the predictors.
- **treat** name of a binary (numeric) vector representing the treatment assignment (coded as 0/1).
- **outcome** name of a binary response (numeric) vector (coded as 0/1).
- **predictors** a vector of names representing the explanatory variables to include in the model.

**Value**

- **model0** Fitted model for control group
- **model1** Fitted model for treatment group
InterPredict

Author(s)

Mouloud Belbahri

References


See Also

DualPredict

Examples

library(tools4uplift)
data("SimUplift")

fit <- DualUplift(SimUplift, "treat", "y", predictors = colnames(SimUplift[, 3:12]))

InterPredict Predictions from an interaction estimator

Description

Predictions from the interaction uplift model estimator with associated model performance.

Usage

InterPredict(data, treat, outcome, model, nb.group = 10, plotit = FALSE)

Arguments

data a data frame containing the treatment, the outcome and the predictors.
treat name of a binary (numeric) vector representing the treatment assignment (coded as 0/1).
outcome name of a binary response (numeric) vector (coded as 0/1).
model a model that must be the output of InterUplift function.
nb.group number of groups of equal observations in which to partition the data in order to compute model performance.
plotit if TRUE, a QiniCurve and QiniBarPlot are returned.
Value

- data: a data frame augmented with the predicted uplift
- qini: a Qini Coefficient

Author(s)

Mouloud Belbahri

References


See Also

InterUplift

Examples

```r
library(tools4uplift)
data("SimUplift")
fit <- InterUplift(SimUplift, "treat", "y", colnames(SimUplift[, 3:12]))
pred <- InterPredict(SimUplift, "treat", "y", model = fit, nb.group = 5)[[1]]
```

---

**InterUplift**  
*Interaction estimator*

Description

Fit the interaction uplift model estimator.

Usage

```r
InterUplift(data, treat, outcome, predictors, input = "all")
```

Arguments

- **data**: a data frame containing the treatment, the outcome and the predictors.
- **treat**: name of a binary (numeric) vector representing the treatment assignment (coded as 0/1).
- **outcome**: name of a binary response (numeric) vector (coded as 0/1).
- **predictors**: a vector of names representing the explanatory variables to include in the model.
LassoPath

input an option for predictors argument. If "all" (default), the model assumes that the model has to create the interaction of all variables with treat. If "best", the model assumes that the predictors vector is the output of the BestFeatures function.

Value

an interaction model

Author(s)

Mouloud Belbahri

References


See Also

InterPredict

Examples

library(tools4uplift)
data("SimUplift")

fit <- InterUplift(SimUplift, "treat", "y", colnames(SimUplift[, 3:12]))

LassoPath

LASSO path for penalized logistic regression

Description

Fit an interaction uplift model via penalized maximum likelihood. The regularization path is computed for the lasso penalty at a grid of values for the regularization parameter lambda.

Usage

LassoPath(data, formula, nb.lambda = 100)
Arguments

data a data frame containing the treatment, the outcome and the predictors.
formula an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
nb.lambda the number of lambda values - Default is 100.

Value
a dataframe containing the coefficients values and the number of nonzeros coefficients for different values of lambda.

Author(s)
Mouloud Belbahri

References

See Also
bestfeatures, glmnet

Examples

#See glmnet() from library("glmnet") for more information

<table>
<thead>
<tr>
<th>QiniArea</th>
<th>Qini coefficient</th>
</tr>
</thead>
</table>

Description
Computes the area under the Qini curve.

Usage
QiniArea(x)

Arguments
x a table that must be the output of QiniTable function.

Value
the Qini coefficient
Author(s)
Mouloud Belbahri

References

See Also
QiniTable

Examples

```R
library(tools4uplift)
data("SimUplift")
square1 <- SquareUplift(SimUplift, "X1", "X2", "treat", "y")

# performance of the heat map uplift estimation on the training dataset
perf <- QiniTable(data = square1, treat = "treat",
                   outcome = "y", prediction = "Uplift_X1_X2", nb.group = 5)

QiniArea(perf)
```

---

### QiniBarPlot

**Uplift barplot**

**Description**
Barplot of observed uplift with respect to predicted uplift sorted from the highest to the lowest.

**Usage**

```R
QiniBarPlot(x, title = "Model Performance: Uplift by Group", color = NULL)
```

**Arguments**

- `x` a table that must be the output of `QiniTable` function.
- `title` an overall title for the plot.
- `color` color of the barplot.
Description

Curve of the function Qini, the incremental observed uplift with respect to predicted uplift sorted from the highest to the lowest.

Usage

QiniCurve(x, title = "Model Performance: Qini Curve", color = NULL)

Arguments

x a table that must be the output of QiniTable function.
title an overall title for the plot.
color color of the curve.
Value

a Qini curve

Author(s)

Mouloud Belbahri

References


See Also

QiniTable

Examples

```r
library(toolsTuplift)
data("SimUplift")
squareUpleft <- SquareUplift(SimUplift, "X1", "X2", "treat", "y")

# performance of the heat map uplift estimation on the training dataset
perf <- QiniTable(data = squareUpleft, treat = "treat",
                   outcome = "y", prediction = "Uplift_X1_X2", nb.group = 5)
QiniCurve(perf)
```

<table>
<thead>
<tr>
<th>QiniTable</th>
<th>Performance of an uplift estimator</th>
</tr>
</thead>
</table>

Description

Table of performance of an uplift model. This table is used in order to visualize the performance of an uplift model and to compute the qini coefficient.

Usage

QiniTable(data, treat, outcome, prediction, nb.group = 10)
**Arguments**

- **data**: a data frame containing the response, the treatment and predicted uplift.
- **treat**: a binary (numeric) vector representing the treatment assignment (coded as 0/1).
- **outcome**: a binary response (numeric) vector (coded as 0/1).
- **prediction**: a predicted uplift (numeric) vector to sort the observations from highest to lowest uplift.
- **nb.group**: number of groups of equal observations in which to partition the data set to show results.

**Value**

- a table with descriptive statistics related to an uplift model estimator.

**Author(s)**

- Mouloud Belbahri

**References**


**See Also**

- `QiniArea`, `QiniBarPlot` and `QiniCurve`

**Examples**

```r
library(tools4uplift)
data("SimUplift")
square1 <- SquareUplift(SimUplift, "X1", "X2", "treat", "y")

# performance of the heat map uplift estimation on the training dataset
perf <- QiniTable(data = square1, treat = "treat",
                   outcome = "y", prediction = "Uplift_X1_X2")

perf
```
Description

The synthetic data contains 20 predictors, a treatment allocation variable and an outcome binary variable. This dataset is used in the package examples.

Usage

data("SimUplift")

Format

A data frame with 1000 observations on the following 22 variables.

y a binary response vector
treat a binary treatment allocation vector
X1 a numeric vector
X2 a numeric vector
X3 a numeric vector
X4 a numeric vector
X5 a numeric vector
X6 a numeric vector
X7 a numeric vector
X8 a numeric vector
X9 a numeric vector
X10 a numeric vector
X11 a numeric vector
X12 a numeric vector
X13 a numeric vector
X14 a numeric vector
X15 a numeric vector
X16 a numeric vector
X17 a numeric vector
X18 a numeric vector
X19 a numeric vector
X20 a numeric vector

Examples

data("SimUplift")
SplitUplift

Split data with respect to uplift distribution

Description

Split a dataset into training and validation subsets with respect to the uplift sample distribution.

Usage

SplitUplift(data, p, group)

Arguments

data
   a data frame of interest that contains at least the response and the treatment variables.

p
   The desired sample size. p is a value between 0 and 1 expressed as a decimal, it is set to be proportional to the number of observations per group.

group
   Your grouping variables. Generally, for uplift modelling, this should be a vector of treatment and response variables names, e.g. c("treat", "y").

Value

train
   a training data frame of $p$ percent

valid
   a validation data frame of $1-p$ percent

Author(s)

Mouloud Belbahri

References


Examples

library(tools4uplift)
data("SimUplift")

split <- SplitUplift(SimUplift, 0.8, c("treat", "y"))
train <- split[[1]]
valid <- split[[2]]
SquareUplift

**Bivariate quantization**

**Description**

A non-parametric heat map representing the observed uplift in rectangles that explore a bivariate dimension space. The function also predicts the individual uplift based on the heatmap.

**Usage**

```r
SquareUplift(data, var1, var2, treat, outcome, n.split = 10,
              n.min = 1, categorize = TRUE, nb.group = 3,
              plotit = TRUE, nb.col = 20)
```

**Arguments**

- `data`: a data frame containing uplift models variables.
- `var1`: x-axis variable name. Represents the first dimension of interest.
- `var2`: y-axis variable name. Represents the second dimension of interest.
- `treat`: name of a binary (numeric) vector representing the treatment assignment (coded as 0/1).
- `outcome`: name of a binary response (numeric) vector (coded as 0/1).
- `n.split`: the number of intervals to consider per explanatory variable. Must be an integer > 1.
- `n.min`: minimum number of observations per group (treatment and control) within each rectangle. Must be an integer > 0.
- `categorize`: if TRUE, the algorithm will augment the data with the categorical variable `Cat_var1_var2` with nb.group categories sorted from the highest to the lowest predicted uplift.
- `nb.group`: number of categories of equal observations of the variable `Cat_var1_var2`. Must be an integer > 1.
- `plotit`: if TRUE, a heatmap of observed uplift per rectangle is plotted.
- `nb.col`: number of colors for the heatmap. From `royalblue` to `red`. Default is 20. Must be an integer and should greater than `n.split` for better visualization.

**Value**

returns an augmented dataset with `Uplift_var1_var2` variable representing a predicted uplift for each observation based on the rectangle it belongs to. By default, the function creates also a categorical variable `Cat_var1_var2` based on the predicted uplift and plots a heat map of observed uplift.

**Author(s)**

Mouloud Belbahri
References


Examples

```r
library(tools4uplift)
data("SimUplift")
square <- SquareUplift(SimUplift, "X1", "X2", "treat", "y")
```
# Index

*Topic **LASSO**
- BestFeatures, 3
- LassoPath, 11

*Topic **Qini**
- QiniArea, 12
- QiniBarPlot, 13
- QiniCurve, 14
- QiniTable, 15

*Topic **binning**
- BinUplift, 4
- BinUpliftEnhanced, 6

*Topic **datasets**
- SimUplift, 17

*Topic **heatmap**
- SquareUplift, 19

*Topic **logistic**
- BestFeatures, 3

*Topic **performance**
- QiniArea, 12
- QiniBarPlot, 13
- QiniCurve, 14
- QiniTable, 15

*Topic **prediction**
- DualPredict, 7
- InterPredict, 9

*Topic **sampling**
- SplitUplift, 18

*Topic **tree**
- BinUplift, 4
- BinUpliftEnhanced, 6

*Topic **uplift**
- DualPredict, 7
- DualUplift, 8
- InterPredict, 9
- InterUplift, 10
- SimUplift, 17
- SquareUplift, 19

**toolsUplift**
- tools4uplift (tools4uplift-package), 2
- tools4uplift-package, 2

BestFeatures, 3