Package: Dforest (via r-universe)

November 2, 2024

Type Package

Title Decision Forest

Version 0.4.2

Date 2017-11-28

Author Leihong Wu <leihong.wu@fda.hhs.gov>, Weida Tong (Weida.tong@fda.hhs.gov)

Maintainer Leihong Wu <leihong.wu@fda.hhs.gov>

Depends R (>= 3.0)

Imports rpart, ggplot2, methods, stats

Description Provides R-implementation of Decision forest algorithm, which combines the predictions of multiple independent decision tree models for a consensus decision. In particular, Decision Forest is a novel pattern-recognition method which can be used to analyze: (1) DNA microarray data; (2) Surface-Enhanced Laser Desorption/Ionization Time-of-Flight Mass Spectrometry (SELDI-TOF-MS) data; and (3) Structure-Activity Relation (SAR) data. In this package, three fundamental functions are provided, as (1)DF_train, (2)DF_pred, and (3)DF_CV. run Dforest() to see more instructions. Weida Tong (2003) <doi:10.1021/ci020058s>.

License GPL-2

LazyLoad yes

LazyData yes

Encoding UTF-8

NeedsCompilation no

RoxygenNote 6.0.1

Date/Publication 2017-11-28 22:03:57 UTC

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

RemoteUrl https://github.com/cran/Dforest

RemoteRef HEAD

RemoteSha 729bce90307b5e85b9e97f46178052f1dd4391d5

Contents

`cal_MCC` *Performance evaluation from other modeling algorithm Result*

Description

Performance evaluation from other modeling algorithm Result

Usage

```
cal_MCC(pred, label)
```

Arguments

`pred`	Predictions
`label`	Known-endpoint

Value

result$ACC: Predicting Accuracy

result$MIS: MisClassfication Counts

result$MCC: Matthew's Correlation Coefficients

result$bACC: balanced Accuracy

Con_DT *Construct Decision Tree model with pruning*

Description

Construct Decision Tree model with pruning

Usage

```
Con_DT(X, Y, min_split = 10, cp = 0.01)
```

Arguments

X	dataset
Y	data_Labels
min_split	minimum number of node in each leaf
cp	pre-defined Complexity Parameter (CP) rpart program

Value

Decision Tree Model with pruning Implemented by rpart

See Also

rpart

data_dili *QSAR dataset with DILI endpoint for demo*

Description

This data set gives the DILI endpoint of various compounds (Most or No DILI-concern) with QSAR descriptors generated by MOLD2

Usage

```
rivers
```

Format

A List containing two vectors: X contains 958 observations and 777 variables. Y contains DILI endpoints of 958 observations

Source

In-house data

References

Minjun Chen (2011) *FDA-approved drug labeling for the study of drug-induced liver injury*. Drug discovery today

---

`Dforest` *Demo script to lean Decision Forest package Demo data are located in data/ folder*

Description

Demo script to lean Decision Forest package Demo data are located in data/ folder

Usage

```
Dforest()
```

Author(s)

Leihong.Wu

Examples

```
Dforest()
```

`DF_acc` *Performance evaluation from Decision Tree Predictions*

Description

Performance evaluation from Decision Tree Predictions

Usage

```
DF_acc(pred, label)
```

Arguments

`pred`	Predictions
`label`	Known-endpoint

Value

result$ACC: Predicting Accuracy

result$MIS: MisClassfication Counts

result$MCC: Matthew's Correlation Coefficients

result$bACC: balanced Accuracy

DF_calp *T-test for feature selection*

Description

T-test for feature selection

Usage

```
DF_calp(X, Y)
```

Arguments

X	X variable matrix
Y	Y label

DF_ConfPlot *Decision Forest algorithm: confidence level accumulated plot*

Description

Draw accuracy curve according to the confidence level of predictions

Usage

```
DF_ConfPlot(Pred_result, Label, bin = 20, plot = T, smooth = F)
```

Arguments

Pred_result	Predictions
Label	known label for Test Dataset
bin	How many bins occurred in Conf Plot (Default is 20)
plot	Draw Plot if True, otherwise output the datamatrix
smooth	if TRUE, Fit the performance curve with smooth function (by ggplot2)

Value

ACC_Conf: return data Matrix ("ConfidenceLevel", "Accuracy", "Matched Samples") for confidence plot (no plot)

ConfPlot: Draw Confidence Plot if True, need install ggplot2

DF_ConfPlot_accu — *Decision Forest algorithm: confidence level accumulated plot (accumulated version)*

Description

Draw accuracy curve according to the confidence level of predictions

Usage

```
DF_ConfPlot_accu(Pred_result, Label, bin = 20, plot = T, smooth = F)
```

Arguments

Pred_result	Predictions
Label	known label for Test Dataset
bin	How many bins occurred in Conf Plot (Default is 20)
plot	Draw Plot if True, otherwise output the datamatrix
smooth	if TRUE, Fit the performance curve with smooth function (by ggplot2)

Value

ACC_Conf: return data Matrix ("ConfidenceLevel", "Accuracy", "Matched Samples") for confidence plot (no plot)

ConfPlot: Draw Confidence Plot if True, need install ggplot2

DF_CV — *Decision Forest algorithm: Model training with Cross-validation*

Description

Decision Forest algorithm: Model training with Cross-validation Default is 5-fold cross-validation

Usage

```
DF_CV(X, Y, stop_step = 10, CV_fold = 5, Max_tree = 20, min_split = 10,
  cp = 0.1, Filter = F, p_val = 0.05, Method = "bACC", Quiet = T,
  Grace_val = 0.05, imp_accu_val = 0.01, imp_accu_criteria = F)
```

Arguments

X	Training Dataset
Y	Training data endpoint
stop_step	How many extra step would be processed when performance not improved, 1 means one extra step
CV_fold	Fold of cross-validation (Default = 5)
Max_tree	Maximum tree number in Forest
min_split	minimum leaves in tree nodes
cp	parameters to pruning decision tree, default is 0.1
Filter	doing feature selection before training
p_val	P-value threshold measured by t-test used in feature selection, default is 0.05
Method	Which is used for evaluating training process. MIS: Misclassification rate; ACC: accuracy
Quiet	if TRUE (default), don't show any message during the process
Grace_val	Grace Value in evaluation: the next model should have a performance (Accuracy, bACC, MCC) not bad than previous model with threshold
imp_accu_val	improvement in evaluation: adding new tree should improve the overall model performance (Accuracy, bACC, MCC) by threshold
imp_accu_criteria	if TRUE, model must have improvement in accumulated accuracy

Value

.$performance: Overall training accuracy (Cross-validation)

.$pred: Detailed training prediction (Cross-validation)

.$detail: Detailed usage of Decision tree Features/Models and their performances in all CVs

.$Method: pass evaluating Methods used in training

.$cp: pass cp value used in training decision trees

Examples

```
##data(iris)
X = iris[,1:4]
Y = iris[,5]
names(Y)=rownames(X)

random_seq=sample(nrow(X))
split_rate=3
split_sample = suppressWarnings(split(random_seq,1:split_rate))
Train_X = X[-random_seq[split_sample[[1]]],]
Train_Y = Y[-random_seq[split_sample[[1]]]]

CV_result = DF_CV(Train_X, Train_Y)
```

DF_CVsummary *output summary for Dforest Cross-validation results*

Description

Draw plot for Dforest Cross-validation results

Usage

```
DF_CVsummary(CV_result, plot = T)
```

Arguments

CV_result	Training Dataset
plot	if TRUE (default), draw plot

DF_dataFs *Decision Forest algorithm: Feature Selection in pre-processing*

Description

Decision Forest algorithm: feature selection for two-class predictions, kept statistical significant features pass the t-test

Usage

```
DF_dataFs(X, Y, p_val = 0.05)
```

Arguments

X	Training Dataset
Y	Training Labels
p_val	Correlation Coefficient threshold to filter out high correlated features; default is 0.95

Value

Keep_feat: qualified features in data matrix after filtering

Examples

```
##data(iris)
 X = iris[iris[,5]!="setosa",1:4]
 Y = iris[iris[,5]!="setosa",5]
 used_feat = DF_dataFs(X, Y)
```

DF_dataPre *Decision Forest algorithm: Data pre-processing*

Description

Decision Forest algorithm: Data pre-processing, remove All-Zero columns/features and high correlated features

Usage

```
DF_dataPre(X, thres = 0.95)
```

Arguments

X	Training Dataset
thres	Correlation Coefficient threshold to filter out high correlated features; default is 0.95

Value

Keep_feat: qualified features in data matrix after filtering

Examples

```
##data(iris)
 X = iris[,1:4]
 Keep_feat = DF_dataPre(X)
```

DF_easy *Simple pre-defined pipeline for Decision forest*

Description

This is a script of decision forest for easy use t

Usage

```
DF_easy(Train_X, Train_Y, Test_X, Test_Y, mode = "default")
```

Arguments

Train_X	Training Dataset
Train_Y	Training data endpoint
Test_X	Testing Dataset
Test_Y	Testing data endpoint
mode	pre-defined modeling

Value

data_matrix training and testing result

Examples

```
# data(demo_simple)
X = iris[,1:4]
Y = iris[,5]
names(Y)=rownames(X)

random_seq=sample(nrow(X))
split_rate=3
split_sample = suppressWarnings(split(random_seq,1:split_rate))
Train_X = X[-random_seq[split_sample[[1]]],]
Train_Y = Y[-random_seq[split_sample[[1]]]]
Test_X = X[random_seq[split_sample[[1]]],]
Test_Y = Y[random_seq[split_sample[[1]]]]

Result = DF_easy(Train_X, Train_Y, Test_X, Test_Y)
```

DF_perf *performance evaluation between two factors*

Description

performance evaluation between two factors

Usage

```
DF_perf(pred, label)
```

Arguments

pred	Predictions
label	Known-endpoint

Value

result$ACC: Predicting Accuracy

result$MIS: MisClassfication Counts

result$MCC: Matthew's Correlation Coefficients

result$bACC: balanced Accuracy

DF_pred *Decision Forest algorithm: Model prediction*

Description

Decision Forest algorithm: Model prediction with constructed DF models. DT_models is a list of Decision Tree models (rpart.objects) generated by DF_train() DT_train_CV() is only designed for Cross-validation and won't generate models

Usage

```
DF_pred(DT_models, X, Y = NULL)
```

Arguments

DT_models	Constructed DF models
X	Test Dataset
Y	Test data endpoint

Value

.$accuracy: Overall test accuracy

.$predictions: Detailed test prediction

Examples

```
# data(demo_simple)
X = data_dili$X
Y = data_dili$Y
names(Y)=rownames(X)

random_seq=sample(nrow(X))
split_rate=3
split_sample = suppressWarnings(split(random_seq,1:split_rate))
Train_X = X[-random_seq[split_sample[[1]]],]
Train_Y = Y[-random_seq[split_sample[[1]]]]
Test_X = X[random_seq[split_sample[[1]]],]
Test_Y = Y[random_seq[split_sample[[1]]]]

used_model = DF_train(Train_X, Train_Y)
Pred_result = DF_pred(used_model,Test_X,Test_Y)
```

DF_train *Decision Forest algorithm: Model training*

Description

Decision Forest algorithm: Model training

Usage

```
DF_train(X, Y, stop_step = 5, Max_tree = 20, min_split = 10, cp = 0.1,
  Filter = F, p_val = 0.05, Method = "bACC", Quiet = T,
  Grace_val = 0.05, imp_accu_val = 0.01, imp_accu_criteria = F)
```

Arguments

X	Training Dataset
Y	Training data endpoint
stop_step	How many extra step would be processed when performance not improved, 1 means one extra step
Max_tree	Maximum tree number in Forest
min_split	minimum leaves in tree nodes
cp	parameters to pruning decision tree, default is 0.1
Filter	doing feature selection before training
p_val	P-value threshold measured by t-test used in feature selection, default is 0.05
Method	Which is used for evaluating training process. MIS: Misclassification rate; ACC: accuracy
Quiet	if TRUE (default), don't show any message during the process
Grace_val	Grace Value in evaluation: the next model should have a performance (Accuracy, bACC, MCC) not bad than previous model with threshold
imp_accu_val	improvement in evaluation: adding new tree should improve the overall model performance (Accuracy, bACC, MCC) by threshold
imp_accu_criteria	if TRUE, model must have improvement in accumulated accuracy

Value

.$accuracy: Overall training accuracy

.$pred: Detailed training prediction (fitting)

.$detail: Detailed usage of Decision tree Features/Models and their performances

.$models: Constructed (list of) Decision tree models

.$Method: pass evaluating Methods used in training

.$cp: pass cp value used in training decision trees

Examples

```
##data(iris)
X = iris[,1:4]
Y = iris[,5]
names(Y)=rownames(X)
used_model = DF_train(X,factor(Y))
```

`DF_Trainsummary` *output summary for Dforest test results*

Description

Draw plot for Dforest test results

Usage

```
DF_Trainsummary(used_model, plot = T)
```

Arguments

`used_model`	Training result
`plot`	if TRUE (default), draw plot

`multiplot` *multiplot*

Description

Multiple plot function

If the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE), then plot 1 will go in the upper left, 2 will go in the upper right, and 3 will go all the way across the bottom.

Usage

```
multiplot(..., plotlist = NULL, cols = 1, layout = NULL)
```

Arguments

`...`	ggplot objects
`plotlist`	a list of ggplot objects
`cols`	Number of columns in layout
`layout`	A matrix specifying the layout. If present, 'cols' is ignored.

Pred_DT *Doing Prediction with Decision Tree model*

Description

Doing Prediction with Decision Tree model

Usage

```
Pred_DT(model, X)
```

Arguments

model	Decision Tree Model
X	dataset

Value

Decision Tree Predictions Different endpoints presented in multiple columns

Source

rpart

See Also

rpart

Index