Package ‘modelmisc’

September 3, 2018

Type Package

Title Miscellaneous tools for modelling

Version 0.1.1

Date 2018-01-06

Author Tom Wilson <tpw2@aber.ac.uk>

Maintainer Tom Wilson <tpw2@aber.ac.uk>

Description Wrapper and helpers for modelling.

Depends ggplot2, ggpubr

Imports magrittr, dplyr, caret, randomForest, AUC, HandTill2001, reshape2, viridis

Suggests testthat, covr

License GPL (>=3)

LazyData TRUE

RoxygenNote 6.1.0

NeedsCompilation no

R topics documented:

`bland_altman_plot` *Bland Altman Plot*

Description

Bland Altman Plot

Usage

```
bland_altman_plot(x, y)
```

Arguments

x	a numeric vector (A)
y	a numeric vector (B)

Value

a ggplot2 object

`canberra_distance` *Canberra Distance*

Description

Calculate the Canberra Distance between two vectors of feature ranks. Input vectors must both be numeric and have equal cardinality.

Usage

```
canberra_distance(x, y, scale = TRUE)
```

Arguments

x a numeric vector

y a numeric vector

scale logical; if TRUE then the canberra distance is scaled by the 1 - (maximum possible distance) to give a value between 0 and 1. 1 = vectors are identical, 0 = no similiarity

Value

a numeric value for the canberra distance

References

Jurman, G., Merler, S., Barla, A., Paoli, S., Galea, A., Furlanello, C., 2008. *Algebraic stability indicators for ranked lists in molecular profiling*. Bioinformatics 24 (2):258-264

conf_int *Confidence Interval*

Description

Calculate a confidence interval for a vector of values

Usage

```
conf_int(x, ci = 0.975)
```

Arguments

x a numeric vector

ci a numeric value (0 - 1) for the required confidence interval

Value

a numeric value for the lower and upper bounds of the confidence interval

`dice_sorensen` *Dice Sorensen Index*

Description

Calculate the Dice-Sorenson Index between two feature vectors

Usage

```
dice_sorensen(x, y)
```

Arguments

`x`	a character vector
`y`	a character vector

Value

a numeric value for the Dice-Sorensen Index

References

Zucknick, M., Richardson, S., Stronach, E.A., 2008. *Comparing the characteristics of gene expression profiles derived by univariate and multivariate classification methods*. Statistical Applications in Genetics and Molecular Biology 7 (1):7

Loscalzo, S., Yu, L., Ding, C., 2009. *Consensus group stable feature selection*. In: Proceeding of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09),pp.567-575.

`forest_accuracy` *Forest Accuracy*

Description

Retrieve the forest accuracy (1 - OOB)

Usage

```
forest_accuracy(model)
```

Arguments

`model`	a randomForest model object

Value

a numeric value for the model accuracy (0 - 1)

`forest_auc` *Calculate the ROC-AUC from a randomForest model*

Description

Calculte the training ROC-AUC for a randomForest model. The votes cast during out-of-bag (OOB) predictions are used for the predicition of class. If classification is multinomial, then Hand and Till's (2001) method for multi-class AUC is used. If classification is binary, then the standard ROC-AUC apprach is used.

Usage

```
forest_auc(model)
```

Arguments

`model`	a randomForest classification model

Value

a numeric value for ROC-AUC

`forest_feat_ranks` *Random Forest feature rankings*

Description

Create rankings for random forest feature importance scores

Usage

```
forest_feat_ranks(model, meth = "gini")
```

Arguments

`model`	a randomForest model object
`meth`	string of either `gini` (for MeanDecreaseGini) or `perm` (for MeanDecreaseAcc)

Value

a `data.frame` of feature importance scores and rankings

`forest_kappa` *Calculate Cohen's Kappa from a randomForest object*

Description

Calculte the training inter-rate agreement (Kappa) for a randomForest model. The votes cast during out-of-bag (OOB) predictions are used for the predicition of class.

Usage

```
forest_kappa(model)
```

Arguments

`model`	a randomForest classification model

Value

a numeric value for the overall inter-rate agreement (Kappa)

`get_kappa` *Cohen's Kappa*

Description

Get the overall interate agreement rate (Cohen's Kappa)

Usage

```
get_kappa(train_model, test_data, test_class)
```

Arguments

`train_model`	a valid model object
`test_data`	a `data.frame` to be used for prediction
`test_class`	a vector of class lables for `test_data`

Value

a numeric value for Kappa

`get_test_auc` *Test ROC-AUC*

Description

Clculate the ROC-AUC using a training model and independant test-data. If classification is multinomial, then Hand and Till's (2001) method for multi-class AUC is used. If classification is binary, then the standard ROC-AUC apprach is used.

Usage

```
get_test_auc(train_model, test_data, test_cls)
```

Arguments

`train_model`	a training model
`test_data`	a `data.frame` of data for test predictions
`test_cls`	a vector of class labels for `test_data`

`get_test_kappa` *Calculate the test inter-rate agreement*

Description

Calculate Cohen's Kappa using a training model and independant test-data

Usage

```
get_test_kappa(train_model, test_data, test_cls)
```

Arguments

`train_model`	a training model
`test_data`	a `data.frame` of data for test predictions
`test_cls`	a vector of class labels for `test_data`

hammings_distance *Relative Hamming Distance*

Description

Calculate the Relative Hamming Distance between two feaure vectors

Usage

```
hammings_distance(x, y, m)
```

Arguments

`x`	a character vector
`y`	a character vector o
`m`	a numeric value for the total number of features in the dataset

Value

a numeric value for the Relative Hamming Distance

References

Dunne, K., Cunningham, P., Azuaje, F., 2002. *Solutions to instability problems with sequential wrapper-based approaches to feature selection*. Technical Report,Department of Computer Science, Trinity College, Dublin.

jaccards_index *Jaccard's Similiarity*

Description

Calculate the Jaccard's Similiarity (or Taminoto Distance) between feature vectors

Usage

```
jaccards_index(x, y)
```

Arguments

`x`	a character vector
`y`	a character vector

Value

a numeric value for the Jaccard's Similiarity Coefficent

`ochiais_index` *Ochiais Index*

Description

Calculate Ochiais Index between feature vectors

Usage

```
ochiais_index(x, y)
```

Arguments

`x`	a character vector
`y`	a character vector

Value

a numeric value for Ochiais Index

`percentage_overlap` *Percentage overlap*

Description

Calculate the percentage of overlap between two feature vectors

Usage

```
percentage_overlap(x, y)
```

Arguments

`x`	a character vector
`y`	a character vector

Value

two numeric values. Value one is the percentage of `x` which overlaps with `y`. Value two is the percentage of `y` which overlaps with `x`

`plot_rf_confusion` *Plot Confusion Matrix from Random Forest Classification*

Description

Plot Confusion Matrix from Random Forest Classification

Usage

```
plot_rf_confusion(rf_model)
```

Arguments

`rf_model`	a randomForest classification model

Value

a ggplot2 plot

`proximity_to_mds` *RF - MDS*

Description

Multi Dimensional Scaling (MDS) of randomForest proximities

Usage

```
proximity_to_mds(x)
```

Arguments

`x`	a `randomForest` object containing a valid proximity matrix

Value

a `data.frame` of `cmdscale (1 - proxmimity)` for Dimension 1 and 2

RPT *RPT*

Description

Calculates the Robustness Performance Trade-off

Usage

```
RPT(stability, performance, beta = 1)
```

Arguments

`stability`	a numeric value for model stability
`performance`	a numeric value for model performance
`beta`	a positive integer. Default is 1, which treats stability and performance equally.

Value

a numeric value for RPT between 0 and 1

strat_resamp *Stratified Resampling*

Description

Create a training and test set, stratfied by `class`

Usage

```
strat_resamp(x, cls, p)
```

Arguments

`x`	a `data.frame` of variables and observations
`cls`	a vector of class information for stratifying. It is assumed that `cls` is balanaced
`p`	a numeric value for the partitioning ratio (ie, 0.632)

Value

a list of four elements

- train_cls `cls` vector for training set
- train_x training data
- test_cls `cls` vector for test set
- test_x test data

`variance_exp` *Variance Explained*

Description

Variance Explained

Usage

```
variance_exp(x)
```

Arguments

`x`	a `prcomp` object

Value

a numeric vector of percentage variance explained for each principal componenet (PC)

Index