Package ‘Metrics’

April 21, 2017

Title Evaluation Metrics for Machine Learning

Description Metrics is a set of evaluation metrics that is commonly used in supervised machine learning.

URL https://github.com/benhamner/Metrics/tree/master/R

Version 0.1.2

Suggests RUnit

Maintainer ORPHANED

Author Ben Hamner <ben@benhamner.com>

License BSD

Collate 'metrics.r'

Repository CRAN

Date/Publication 2017-04-21 14:55:21 UTC

NeedsCompilation no

X-CRAN-Original-Maintainer Ben Hamner <ben@benhamner.com>

X-CRAN-Comment Orphaned and corrected on 2017-04-21 as check errors were not corrected despite reminders.

R topics documented:

<table>
<thead>
<tr>
<th>Description</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>ae</td>
<td></td>
</tr>
<tr>
<td>apk</td>
<td></td>
</tr>
<tr>
<td>auc</td>
<td></td>
</tr>
<tr>
<td>ce</td>
<td></td>
</tr>
<tr>
<td>ll</td>
<td></td>
</tr>
<tr>
<td>logLoss</td>
<td></td>
</tr>
<tr>
<td>mae</td>
<td></td>
</tr>
<tr>
<td>mapk</td>
<td></td>
</tr>
<tr>
<td>MeanQuadraticWeightedKappa</td>
<td></td>
</tr>
<tr>
<td>mse</td>
<td></td>
</tr>
<tr>
<td>msle</td>
<td></td>
</tr>
<tr>
<td>rmse</td>
<td></td>
</tr>
</tbody>
</table>

1
ae

*Compute the absolute error* This function computes the elementwise absolute error for a number or a vector

**Description**

Compute the absolute error This function computes the elementwise absolute error for a number or a vector

**Usage**

`ae(actual, predicted)`

**Arguments**

- `actual` : ground truth number or vector
- `predicted` : predicted number or vector

apk

*Compute the average precision at k*

**Description**

This function computes the average precision at k between two sequences

**Usage**

`apk(k, actual, predicted)`

**Arguments**

- `k` : max length of predicted sequence
- `actual` : ground truth set (vector)
- `predicted` : predicted sequence (vector)
 auc  

Compute the area under the ROC (AUC)

**Description**
This function computes the area under the receiver-operator characteristic (AUC)

**Usage**
```r
auc(actual, predicted)
```

**Arguments**
- `actual`: binary vector
- `predicted`: real-valued vector that defines the ranking

---

 ce  

Compute the classification error

**Description**
This function computes the classification error between two vectors

**Usage**
```r
ce(actual, predicted)
```

**Arguments**
- `actual`: ground truth vector
- `predicted`: predicted vector

---

 ll  

Compute the log loss

**Description**
This function computes the elementwise log loss for a number or a vector

**Usage**
```r
ll(actual, predicted)
```

**Arguments**
- `actual`: binary ground truth number or vector
- `predicted`: predicted number or vector
logLoss

*Compute the mean log loss*

**Description**

This function computes the mean log loss between two vectors.

**Usage**

\[ \text{logLoss} \text{(actual, predicted)} \]

**Arguments**

- **actual**: binary ground truth vector
- **predicted**: predicted vector

mae

*Compute the mean absolute error*  
*This function computes the mean absolute error between two vectors*

**Description**

Compute the mean absolute error*  
*This function computes the mean absolute error between two vectors*

**Usage**

\[ \text{mae} \text{(actual, predicted)} \]

**Arguments**

- **actual**: ground truth vector
- **predicted**: vector
\textit{mapk} \hfill 5

\textbf{mapk} \hfill \textit{Compute the mean average precision at k}

\section*{Description}

This function computes the mean average precision at k of two lists of sequences.

\section*{Usage}

\begin{center}
\texttt{mapk(k, actual, predicted)}
\end{center}

\section*{Arguments}

\begin{itemize}
  \item \texttt{k} \hfill max length of predicted sequence
  \item \texttt{actual} \hfill list of ground truth sets (vectors)
  \item \texttt{predicted} \hfill list of predicted sequences (vectors)
\end{itemize}

\section*{MeanQuadraticWeightedKappa} \hfill \textit{Compute the mean quadratic weighted kappa}

\section*{Description}

This function computes the mean quadratic weighted kappa, which can optionally be weighted.

\section*{Usage}

\begin{center}
\texttt{MeanQuadraticWeightedKappa(kappas, weights)}
\end{center}

\section*{Arguments}

\begin{itemize}
  \item \texttt{kappas} \hfill is a vector of possible kappas
  \item \texttt{weights} \hfill is an optional vector of ratings
\end{itemize}
mse

Compute the mean squared error. This function computes the mean squared error between two vectors.

Usage

```
mse(actual, predicted)
```

Arguments

- `actual`: ground truth vector
- `predicted`: predicted vector

msle

Compute the mean squared log error.

Description

This function computes the mean squared log error between two vectors.

Usage

```
msle(actual, predicted)
```

Arguments

- `actual`: ground truth vector
- `predicted`: predicted vector
rmse

Compute the root mean squared error. This function computes the root mean squared error between two vectors.

Usage

rmse(actual, predicted)

Arguments

<table>
<thead>
<tr>
<th>actual</th>
<th>ground truth vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>predicted</td>
<td>predicted vector</td>
</tr>
</tbody>
</table>

rmsle

Compute the root mean squared log error.

Description

This function computes the root mean squared log error between two vectors.

Usage

rmsle(actual, predicted)

Arguments

<table>
<thead>
<tr>
<th>actual</th>
<th>ground truth vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>predicted</td>
<td>predicted vector</td>
</tr>
</tbody>
</table>
ScoreQuadraticWeightedKappa

*Compute the quadratic weighted kappa*

**Description**

This function computes the quadratic weighted kappa between two vectors of integers.

**Usage**

\[ \text{ScoreQuadraticWeightedKappa}(\text{rater.a, rater.b, min.rating, max.rating}) \]

**Arguments**

- `rater.a` is the first rater’s ratings
- `rater.b` is the second rater’s ratings
- `min.rating` is the minimum possible rating
- `max.rating` is the maximum possible rating

**se**

*Compute the squared error*

**Description**

This function computes the elementwise squared error for a number or a vector.

**Usage**

\[ \text{se}(\text{actual, predicted}) \]

**Arguments**

- `actual` ground truth number or vector
- `predicted` predicted number or vector
Compute the squared log error

**Description**

This function computes the elementwise squared log error for a number or a vector.

**Usage**

```r
sle(actual, predicted)
```

**Arguments**

- `actual`: ground truth number or vector
- `predicted`: predicted number or vector
Index

ae, 2
apk, 2
auc, 3
ce, 3
ll, 3
logLoss, 4
mae, 4
mapk, 5
MeanQuadraticWeightedKappa, 5
mse, 6
msle, 6
rmse, 7
rmsle, 7
ScoreQuadraticWeightedKappa, 8
se, 8
sle, 9