

Package: shinyML (via r-universe)

September 5, 2024

Type Package

Title Compare Supervised Machine Learning Models Using Shiny App

Version 1.0.1

Author Jean Bertin

Maintainer Jean Bertin <jean.bertin@mines-paris.org>

Description Implementation of a shiny app to easily compare supervised machine learning model performances. You provide the data and configure each model parameter directly on the shiny app. Different supervised learning algorithms can be tested either on Spark or H2O frameworks to suit your regression and classification tasks. Implementation of available machine learning models on R has been done by Lantz (2013, ISBN:9781782162148).

URL <https://jeanbertinr.github.io/shinyMLpackage/>

BugReports <https://github.com/JeanBertinR/shinyML/issues>

License GPL-3

Encoding UTF-8

Imports shiny (>= 1.0.3), argonDash, argonR, shinyjs, h2o, shinyWidgets, dygraphs, plotly, sparklyr, tidyr, DT, ggplot2, shinycssloaders, lubridate, graphics

Suggests knitr, rmarkdown, covr, testthat

Depends dplyr, data.table

LazyData True

RoxygenNote 7.1.1

Repository <https://jeanbertinr.r-universe.dev>

RemoteUrl <https://github.com/jeanbertinr/shinyml>

RemoteRef HEAD

RemoteSha 2aecbf230b6f63b847c8b4d48c8a20c6a0c4e6ff

Contents

shinyML_classification	2
shinyML_regression	3
Index	5

shinyML_classification

Implement a shiny web app to compare h2o and Spark supervised machine learning models for classification tasks

Description

This function creates in one line of code a shareable web app to compare supervised classification model performances

Usage

```
shinyML_classification(
  data = data,
  y,
  framework = "h2o",
  share_app = FALSE,
  port = NULL
)
```

Arguments

data	dataset containing one or more explanatory variables and one categorical variable to predict. The dataset must be a data.frame or a data.table and can contain time-based column on Date or POSIXct format
y	the categorical output variable to predict (must correspond to one data column)
framework	the machine learning framework chosen to train and test models (either h2o or Spark). h2o by default
share_app	a logical value indicating whether the app must be shared on local LAN
port	a four-digit number corresponding to the port the application should listen to. This parameter is necessary only if share_app option is set to TRUE

Author(s)

Jean Bertin, <jean.bertin@mines-paris.org>

Examples

```
## Not run:  
library(shinyML)  
shinyML_classification(data = iris,y = "Species",framework = "h2o")  
  
## End(Not run)
```

shinyML_regression	<i>Implement a shiny web app to compare h2o and Spark supervised machine learning models for regression tasks</i>
--------------------	---

Description

This function creates in one line of code a shareable web app to compare supervised regression model performances

Usage

```
shinyML_regression(  
  data = data,  
  y,  
  framework = "h2o",  
  share_app = FALSE,  
  port = NULL  
)
```

Arguments

data	dataset containing one or more explanatory variables and one numeric variable to forecast. The dataset must be a data.frame or a data.table and can contain time-based column on Date or POSIXct format
y	the numerical output variable to forecast (must correspond to one data column)
framework	the machine learning framework chosen to train and test models (either h2o or Spark). h2o by default
share_app	a logical value indicating whether the app must be shared on local LAN
port	a four-digit number corresponding to the port the application should listen to. This parameter is necessary only if share_app option is set to TRUE

Author(s)

Jean Bertin, <jean.bertin@mines-paris.org>

Examples

```
## Not run:
library(shinyML)
# Classical regression analysis
shinyML_regression(data = iris, y = "Petal.Width", framework = "h2o")

# Time series analysis
longley2 <- longley %>% mutate(Year = as.Date(as.character(Year), format = "%Y"))
shinyML_regression(data = longley2, y = "Population", framework = "h2o")

## End(Not run)
```

Index

shinyML_classification, 2
shinyML_regression, 3