Package 'viralx'

March 29, 2024

Title Explainers for Regression Models in HIV Research

Version 1.3.0

Maintainer Juan Pablo Acuña González <22253567@uagro.mx>

Description A dedicated viral-explainer model tool designed to empower researchers in the field of HIV research, particularly in viral load and CD4 (Cluster of Differentiation 4) lymphocytes regression modeling. Drawing inspiration from the 'tidymodels' framework for rigorous model building of Max Kuhn and Hadley Wickham (2020) <https://www.tidymodels.org>, and the 'DALEXtra' tool for explainability by Przemyslaw Biecek (2020) <arXiv:2009.13248>. It aims to facilitate interpretable and reproducible research in biostatistics and computational biology for the benefit of understanding HIV dynamics.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.1

Imports DALEX, DALEXtra, dplyr, earth, Formula, kknn, parsnip, plotmo, plotrix, recipes, rsample, TeachingDemos, vdiffr, workflows

Suggests Cubist, rules, testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 2.10)

LazyData true

NeedsCompilation no

Author Juan Pablo Acuña González [aut, cre]
(<https://orcid.org/0009-0003-6029-6560>)

Repository CRAN

Date/Publication 2024-03-29 21:00:02 UTC

R topics documented:

---

glob_cr_vis   *Global Visualization of SHAP Values for Cubist Rules Model*

Description

This function generates a visualization for the global feature importance of a Cubist Rules (CR) model trained on HIV data with specified hyperparameters.

Usage

```
glob_cr_vis(vip_featured, hiv_data, cr_hyperparameters, vip_train, v_train)
```

Arguments

vip_featured	The name of the response variable to explain.
hiv_data	The training dataset containing predictor variables and the response variable.
cr_hyperparameters	

> A list of hyperparameters for the CR model, including:
>
> - committees: The number of committees to consider.
> - neighbors: The number of neighbors to consider.

vip_train	The dataset used for training the CR model.
v_train	The response variable used for training the CR model.

Value

A visualization of global feature importance for the CR model.

Examples

```
library(dplyr)
library(rules)
library(Cubist)
set.seed(123)
hiv_data <- train2
cr_hyperparameters <- list(neighbors = 5, committees = 58)
vip_featured <- c("cd_2022")
vip_features <- c("cd_2019", "vl_2019", "cd_2021", "vl_2021", "vl_2022")
vip_train <- train2 |>
dplyr::select(rsample::all_of(vip_features))
v_train <- train2 |>
dplyr::select(rsample::all_of(vip_featured))
glob_cr_vis(vip_featured, hiv_data, cr_hyperparameters, vip_train, v_train)
```

---

glob_knn_vis	*Global Visualization of SHAP Values for K-Nearest Neighbor Model*

Description

This function generates a visualization for the global feature importance of a K-Nearest Neighbors (KNN) model trained on HIV data with specified hyperparameters.

Usage

```
glob_knn_vis(vip_featured, hiv_data, knn_hyperparameters, vip_train, v_train)
```

Arguments

vip_featured   The name of the response variable to explain.

hiv_data   The training dataset containing predictor variables and the response variable.

knn_hyperparameters

A list of hyperparameters for the KNN model, including:

- neighbors: The number of neighbors to consider.
- weight_func: The weight function to use.
- dist_power: The distance power parameter.

vip_train   The dataset used for training the KNN model.

v_train   The response variable used for training the KNN model.

Value

A visualization of global feature importance for the KNN model.

Examples

```
library(dplyr)
set.seed(123)
hiv_data <- train2
knn_hyperparameters <- list(neighbors = 5, weight_func = "optimal", dist_power = 0.3304783)
vip_featured <- c("cd_2022")
vip_features <- c("cd_2019", "vl_2019", "cd_2021", "vl_2021", "vl_2022")
vip_train <- train2 |>
dplyr::select(rsample::all_of(vip_features))
v_train <- train2 |>
dplyr::select(rsample::all_of(vip_featured))
glob_knn_vis(vip_featured, hiv_data, knn_hyperparameters, vip_train, v_train)
```

---

glob_nn_vis | *Global Visualization of SHAP Values for Neural Network Model*

Description

The `glob_nn_vis` function generates a global visualization of SHAP (Shapley Additive Explanations) values for a neural network model. It utilizes the DALEXtra package to explain the model's predictions and then creates a global SHAP visualization.

Usage

```
glob_nn_vis(vip_featured, hiv_data, hu, plty, epo, vip_train, v_train)
```

Arguments

vip_featured	A character value specifying the featured variable of interest.
hiv_data	A data frame containing the HIV research data used for model training.
hu	A numeric value specifying the number of hidden units in the neural network model.
plty	A numeric value specifying the penalty parameter for the neural network model.
epo	A numeric value specifying the number of epochs (training iterations) for the neural network model.
vip_train	A data frame containing the training data used to fit the neural network model.
v_train	A numeric vector representing the response variable corresponding to the training data.

Value

A global visualization of SHAP values for the specified neural network model.

Examples

```
library(dplyr)
library(rsample)
cd_2019 <- c(824, 169, 342, 423, 441, 507, 559,
             173, 764, 780, 244, 527, 417, 800,
             602, 494, 345, 780, 780, 527, 556,
             559, 238, 288, 244, 353, 169, 556,
             824, 169, 342, 423, 441, 507, 559)
vl_2019 <- c(40, 11388, 38961, 40, 75, 4095, 103,
             11388, 46, 103, 11388, 40, 0, 11388,
             0,  4095,  40,  93,  49,  49,  49,
             4095,  6837, 38961, 38961, 0, 0, 93,
             40, 11388, 38961, 40, 75, 4095, 103)
cd_2021 <- c(992, 275, 331, 454, 479, 553,  496,
             230, 605, 432, 170, 670, 238,  238,
             634, 422, 429, 513, 327, 465,  479,
             661, 382, 364, 109, 398, 209, 1960,
             992, 275, 331, 454, 479, 553,  496)
vl_2021 <- c(80, 1690,  5113,  71, 289, 3063,  0,
             262, 0, 15089,  13016, 1513, 60, 60,
             49248, 159308, 56, 0, 516675, 49, 237,
             84,  292,  414, 26176,  62, 126,  93,
             80, 1690, 5113,    71, 289, 3063,   0)
cd_2022 <- c(700, 127, 127, 547, 547, 547, 777,
             149, 628, 614, 253, 918, 326, 326,
             574, 361, 253, 726, 659, 596, 427,
             447, 326, 253, 248, 326, 260, 918,
             700, 127, 127, 547, 547, 547, 777)
vl_2022 <- c(0,   0,  53250,   0,  40,  1901, 0,
             955,   0,   0,   0,   0,  40,   0,
             49248, 159308, 56, 0, 516675, 49, 237,
             0,   23601,  0,  40,   0,   0,   0,
             0,   0,   0,   0,   0,   0,  0)
x <- cbind(cd_2019, vl_2019, cd_2021, vl_2021, cd_2022, vl_2022) |>
as.data.frame()
set.seed(123)
hi_data <- rsample::initial_split(x)
set.seed(123)
hiv_data <- hi_data |>
rsample::training()
hu <- 5
plty <- 1.131656e-09
epo <- 176
vip_featured <- c("cd_2022")
vip_features <- c("cd_2019", "vl_2019", "cd_2021", "vl_2021", "vl_2022")
set.seed(123)
vi_train <- rsample::initial_split(x)
set.seed(123)
vip_train <- vi_train |>
rsample::training() |>
dplyr::select(rsample::all_of(vip_features))
v_train <- vi_train |>
```

```
rsample::training() |>
dplyr::select(rsample::all_of(vip_featured))
glob_nn_vis(vip_featured, hiv_data, hu, plty, epo, vip_train, v_train)
```

---

train2 *Training Data for Explainability of Models*

Description

This dataset contains training data for viral load explainer models. It includes CD4 and viral load measurements for different years.

Usage

```
data(train2)
```

Format

A tibble (data frame) with 25 rows and 6 columns.

Note

To explore more rows of this dataset, you can use the print(n = ...) function.

Author(s)

Juan Pablo Acuña González 22253567@uagro.mx

Examples

```
data(train2)
train2
```

---

viralx_knn *Explain K-Nearest Neighbors Model*

Description

Explains the predictions of a K-Nearest Neighbors (KNN) model for CD4 and viral load data using the DALEX and DALEXtra packages. It provides insights into the specified variable's impact on the KNN model's predictions.

Usage

```
viralx_knn(vip_featured, hiv_data, knn_hyperparameters, vip_train, vip_new)
```

Arguments

`vip_featured`	The name of the variable to be explained.
`hiv_data`	The data frame containing the CD4 and viral load data.
`knn_hyperparameters`	
	A list of hyperparameters to be tuned for the KNN model.
`vip_train`	The training data used for creating the explainer object.
`vip_new`	A new observation for which to generate explanations.

Value

A data frame containing explanations for the specified variable.

Examples

```
hiv_data <- train2
knn_hyperparameters <- list(neighbors = 5, weight_func = "optimal", dist_power = 0.3304783)
vip_featured <- c("cd_2022")
vip_train <- hiv_data
vip_new <- vip_train[1,]
viralx_knn(vip_featured, hiv_data, knn_hyperparameters, vip_train, vip_new)
```

`viralx_knn_glob`	*Global Explainers for K-Nearest Neighbor Models*

Description

This function calculates global feature importance for a K-Nearest Neighbors (KNN) model trained on HIV data with specified hyperparameters.

Usage

```
viralx_knn_glob(
  vip_featured,
  hiv_data,
  knn_hyperparameters,
  vip_train,
  v_train
)
```

Arguments

`vip_featured`	The name of the response variable to explain.
`hiv_data`	The training dataset containing predictor variables and the response variable.
`knn_hyperparameters`	
	A list of hyperparameters for the KNN model, including:

- `neighbors`: The number of neighbors to consider.

- weight_func: The weight function to use.
- dist_power: The distance power parameter.

vip_train   The dataset used for training the KNN model.

v_train   The response variable used for training the KNN model.

Value

A list of global feature importance measures for each predictor variable.

Examples

```
library(dplyr)
set.seed(123)
hiv_data <- train2
knn_hyperparameters <- list(neighbors = 5, weight_func = "optimal", dist_power = 0.3304783)
vip_featured <- c("cd_2022")
vip_features <- c("cd_2019", "vl_2019", "cd_2021", "vl_2021", "vl_2022")
vip_train <- train2 |>
dplyr::select(rsample::all_of(vip_features))
v_train <- train2 |>
dplyr::select(rsample::all_of(vip_featured))
viralx_knn_glob(vip_featured, hiv_data, knn_hyperparameters, vip_train, v_train)
```

---

viralx_knn_shap   *Explain K Nearest Neighbor Model using SHAP values*

Description

This function calculates SHAP (SHapley Additive exPlanations) values for a K-Nearest Neighbors (KNN) model trained on HIV data with specified hyperparameters.

Usage

```
viralx_knn_shap(
  vip_featured,
  hiv_data,
  knn_hyperparameters,
  vip_train,
  vip_new,
  orderings
)
```

Arguments

vip_featured   The name of the response variable to explain.

hiv_data   The training dataset containing predictor variables and the response variable.

knn_hyperparameters

                A list of hyperparameters for the KNN model, including:

- `neighbors`: The number of neighbors to consider.
- `weight_func`: The weight function to use.
- `dist_power`: The distance power parameter.

`vip_train`	The dataset used for training the KNN model.
`vip_new`	The dataset for which SHAP values are calculated.
`orderings`	The number of orderings for SHAP value calculations.

Value

A list of SHAP values for each observation in `vip_new`.

Examples

```
## Not run:
set.seed(123)
hiv_data <- train2
knn_hyperparameters <- list(neighbors = 5, weight_func = "optimal", dist_power = 0.3304783)
vip_featured <- c("cd_2022")
vip_train <- hiv_data
vip_new <- vip_train[1, ]
orderings <- 20
viralx_knn_shap(vip_featured, hiv_data, knn_hyperparameters, vip_train, vip_new, orderings)

## End(Not run)
```

---

viralx_knn_vis	*Visualize SHAP Values for K-Nearest Neighbor Model*

Description

Visualizes SHAP (Shapley Additive Explanations) values for a KNN (K-Nearest Neighbor) model by employing the DALEXtra and DALEX packages to provide visual insights into the impact of a specified variable on the model's predictions.

Usage

```
viralx_knn_vis(
  vip_featured,
  hiv_data,
  knn_hyperparameters,
  vip_train,
  vip_new,
  orderings
)
```

Arguments

`vip_featured`	The name of the response variable to explain.
`hiv_data`	The training dataset containing predictor variables and the response variable.
`knn_hyperparameters`	
	A list of hyperparameters for the KNN model, including:

- `neighbors`: The number of neighbors to consider.
- `weight_func`: The weight function to use.
- `dist_power`: The distance power parameter.

`vip_train`	The dataset used for training the KNN model.
`vip_new`	The dataset for which SHAP values are calculated.
`orderings`	The number of orderings for SHAP value calculations.

Value

A list of SHAP values for each observation in `vip_new`.

Examples

```
## Not run:
set.seed(123)
hiv_data <- train2
knn_hyperparameters <- list(neighbors = 5, weight_func = "optimal", dist_power = 0.3304783)
vip_featured <- c("cd_2022")
vip_train <- hiv_data
vip_new <- vip_train[1,]
orderings <- 20
viralx_knn_vis(vip_featured, hiv_data, knn_hyperparameters, vip_train, vip_new, orderings)

## End(Not run)
```

`viralx_mars`	*Explain Multivariate Adaptive Regression Splines Model*

Description

Explains the predictions of a Multivariate Adaptive Regression Splines (MARS) model for viral load or CD4 counts using the DALEX and DALEXtra tools.

Usage

```
viralx_mars(vip_featured, hiv_data, nt, pd, pru, vip_train, vip_new)
```

Arguments

`vip_featured`	A character value
`hiv_data`	A data frame
`nt`	A numeric value
`pd`	A numeric value
`pru`	A character value
`vip_train`	A data frame
`vip_new`	A numeric vector

Value

A data frame

Examples

```
library(dplyr)
library(rsample)
library(Formula)
library(plotmo)
library(plotrix)
library(TeachingDemos)
cd_2019 <- c(824, 169, 342, 423, 441, 507, 559,
             173, 764, 780, 244, 527, 417, 800,
             602, 494, 345, 780, 780, 527, 556,
             559, 238, 288, 244, 353, 169, 556,
             824, 169, 342, 423, 441, 507, 559)
vl_2019 <- c(40, 11388, 38961, 40, 75, 4095, 103,
             11388, 46, 103, 11388, 40, 0, 11388,
             0, 4095, 40, 93, 49, 49, 49,
             4095, 6837, 38961, 38961, 0, 0, 93,
             40, 11388, 38961, 40, 75, 4095, 103)
cd_2021 <- c(992, 275, 331, 454, 479, 553, 496,
             230, 605, 432, 170, 670, 238, 238,
             634, 422, 429, 513, 327, 465, 479,
             661, 382, 364, 109, 398, 209, 1960,
             992, 275, 331, 454, 479, 553, 496)
vl_2021 <- c(80, 1690, 5113, 71, 289, 3063, 0,
             262, 0, 15089, 13016, 1513, 60, 60,
             49248, 159308, 56, 0, 516675, 49, 237,
             84, 292, 414, 26176, 62, 126, 93,
             80, 1690, 5113, 71, 289, 3063, 0)
cd_2022 <- c(700, 127, 127, 547, 547, 547, 777,
             149, 628, 614, 253, 918, 326, 326,
             574, 361, 253, 726, 659, 596, 427,
             447, 326, 253, 248, 326, 260, 918,
             700, 127, 127, 547, 547, 547, 777)
vl_2022 <- c(0, 0, 53250, 0, 40, 1901, 0,
             955, 0, 0, 0, 0, 40, 0,
             49248, 159308, 56, 0, 516675, 49, 237,
```

```
              0,    23601,   0,   40,   0,   0,   0,
              0,    0,      0,    0,   0,   0,  0)
x <- cbind(cd_2019, vl_2019, cd_2021, vl_2021, cd_2022, vl_2022) |>
as.data.frame()
set.seed(123)
hi_data <- rsample::initial_split(x)
set.seed(123)
hiv_data <- hi_data |>
rsample::training()
nt <- 3
pd <- 1
pru <- "none"
vip_featured <- c("cd_2022")
vip_features <- c("cd_2019", "vl_2019", "cd_2021", "vl_2021", "vl_2022")
set.seed(123)
vi_train <- rsample::initial_split(x)
set.seed(123)
vip_train <- vi_train |>
rsample::training() |>
dplyr::select(rsample::all_of(vip_features))
vip_new <- vip_train[1,]
viralx_mars(vip_featured, hiv_data, nt, pd, pru, vip_train, vip_new)
```

| viralx_mars_shap | *Explain Multivariate Adaptive Regression Splines Using SHAP Values* |

Description

Explains the predictions of a MARS (Multivariate Adaptive Regression Splines) model using SHAP (Shapley Additive Explanations) values. It utilizes the DALEXtra and DALEX packages to provide SHAP-based explanations for the specified model.

Usage

```
viralx_mars_shap(
  vip_featured,
  hiv_data,
  nt,
  pd,
  pru,
  vip_train,
  vip_new,
  orderings
)
```

Arguments

vip_featured	A character value
hiv_data	A data frame

nt	A numeric value
pd	A numeric value
pru	A character value
vip_train	A data frame
vip_new	A numeric vector
orderings	A numeric value

Value

A data frame

Examples

```
library(dplyr)
library(rsample)
library(Formula)
library(plotmo)
library(plotrix)
library(TeachingDemos)
cd_2019 <- c(824, 169, 342, 423, 441, 507, 559,
             173, 764, 780, 244, 527, 417, 800,
             602, 494, 345, 780, 780, 527, 556,
             559, 238, 288, 244, 353, 169, 556,
             824, 169, 342, 423, 441, 507, 559)
vl_2019 <- c(40, 11388, 38961, 40, 75, 4095, 103,
             11388, 46, 103, 11388, 40, 0, 11388,
             0,  4095,  40, 93, 49,  49,  49,
             4095,  6837, 38961, 38961, 0, 0, 93,
             40, 11388, 38961, 40, 75, 4095, 103)
cd_2021 <- c(992, 275, 331, 454, 479, 553,  496,
             230, 605, 432, 170, 670, 238,  238,
             634, 422, 429, 513, 327, 465,  479,
             661, 382, 364, 109, 398, 209, 1960,
             992, 275, 331, 454, 479, 553,  496)
vl_2021 <- c(80, 1690, 5113, 71,  289, 3063, 0,
             262, 0, 15089,  13016, 1513, 60, 60,
             49248, 159308, 56, 0, 516675, 49, 237,
             84,  292,  414, 26176,  62,  126,  93,
             80, 1690, 5113,   71, 289, 3063,   0)
cd_2022 <- c(700, 127, 127, 547, 547, 547, 777,
             149, 628, 614, 253, 918, 326, 326,
             574, 361, 253, 726, 659, 596, 427,
             447, 326, 253, 248, 326, 260, 918,
             700, 127, 127, 547, 547, 547, 777)
vl_2022 <- c(0,   0,  53250,   0,   40,  1901, 0,
             955,   0,   0,   0,   0, 40,   0,
             49248, 159308, 56, 0, 516675, 49, 237,
             0,   23601,   0,  40,   0,   0,   0,
             0,   0,    0,    0,   0,   0,   0)
x <- cbind(cd_2019, vl_2019, cd_2021, vl_2021, cd_2022, vl_2022) |>
as.data.frame()
```

```
set.seed(123)
hi_data <- rsample::initial_split(x)
set.seed(123)
hiv_data <- hi_data |>
rsample::training()
nt <- 3
pd <- 1
pru <- "none"
vip_featured <- c("cd_2022")
vip_features <- c("cd_2019", "vl_2019", "cd_2021", "vl_2021", "vl_2022")
set.seed(123)
vi_train <- rsample::initial_split(x)
set.seed(123)
vip_train <- vi_train |>
rsample::training() |>
dplyr::select(rsample::all_of(vip_features))
vip_new <- vip_train[1,]
orderings <- 20
viralx_mars_shap(vip_featured, hiv_data, nt, pd, pru, vip_train, vip_new,orderings)
```

---

viralx_mars_vis   *Visualize SHAP Values for Multivariate Adaptive Regression Splines Model*

Description

Visualizes SHAP (Shapley Additive Explanations) values for a MARS (Multivariate Adaptive Regression Splines) model by employing the DALEXtra and DALEX packages to provide visual insights into the impact of a specified variable on the model's predictions.

Usage

```
viralx_mars_vis(
  vip_featured,
  hiv_data,
  nt,
  pd,
  pru,
  vip_train,
  vip_new,
  orderings
)
```

Arguments

vip_featured	A character value
hiv_data	A data frame
nt	A numeric value

pd	A numeric value
pru	A character value
vip_train	A data frame
vip_new	A numeric vector
orderings	A numeric value

Value

A ggplot object

Examples

```
library(dplyr)
library(rsample)
library(Formula)
library(plotmo)
library(plotrix)
library(TeachingDemos)
cd_2019 <- c(824, 169, 342, 423, 441, 507, 559,
             173, 764, 780, 244, 527, 417, 800,
             602, 494, 345, 780, 780, 527, 556,
             559, 238, 288, 244, 353, 169, 556,
             824, 169, 342, 423, 441, 507, 559)
vl_2019 <- c(40, 11388, 38961, 40, 75, 4095, 103,
             11388, 46, 103, 11388, 40, 0, 11388,
             0,  4095,  40, 93, 49,  49,  49,
             4095, 6837, 38961, 38961, 0, 0, 93,
             40, 11388, 38961, 40, 75, 4095, 103)
cd_2021 <- c(992, 275, 331, 454, 479, 553,  496,
             230, 605, 432, 170, 670, 238,  238,
             634, 422, 429, 513, 327, 465,  479,
             661, 382, 364, 109, 398, 209, 1960,
             992, 275, 331, 454, 479, 553,  496)
vl_2021 <- c(80, 1690,  5113,  71,  289, 3063, 0,
             262,  0, 15089,  13016, 1513, 60, 60,
             49248, 159308, 56, 0, 516675, 49, 237,
             84,  292,  414, 26176,  62,  126,  93,
             80, 1690, 5113,   71, 289, 3063,   0)
cd_2022 <- c(700, 127, 127, 547, 547, 547, 777,
             149, 628, 614, 253, 918, 326, 326,
             574, 361, 253, 726, 659, 596, 427,
             447, 326, 253, 248, 326, 260, 918,
             700, 127, 127, 547, 547, 547, 777)
vl_2022 <- c(0,   0,  53250,   0,   40,   1901, 0,
             955,   0,   0,   0,   0,  40,   0,
             49248, 159308, 56, 0, 516675, 49, 237,
             0,   23601,   0,   40,   0,   0,   0,
             0,   0,    0,    0,   0,    0,   0)
x <- cbind(cd_2019, vl_2019, cd_2021, vl_2021, cd_2022, vl_2022) |>
as.data.frame()
set.seed(123)
```

```
hi_data <- rsample::initial_split(x)
set.seed(123)
hiv_data <- hi_data |>
rsample::training()
nt <- 3
pd <- 1
pru <- "none"
vip_featured <- c("cd_2022")
vip_features <- c("cd_2019", "vl_2019", "cd_2021", "vl_2021", "vl_2022")
set.seed(123)
vi_train <- rsample::initial_split(x)
set.seed(123)
vip_train <- vi_train |>
rsample::training() |>
dplyr::select(rsample::all_of(vip_features))
vip_new <- vip_train[1,]
orderings <- 20
viralx_mars_vis(vip_featured, hiv_data, nt, pd, pru, vip_train, vip_new, orderings)
```

---

viralx_nn   *Explain Neural Network Regression Model*

Description

Explains the predictions of a neural network regression model for viral load or CD4 counts using the DALEX and DALEXtra tools

Usage

```
viralx_nn(vip_featured, hiv_data, hu, plty, epo, vip_train, vip_new)
```

Arguments

vip_featured	A character value
hiv_data	A data frame
hu	A numeric value
plty	A numeric value
epo	A numeric value
vip_train	A data frame
vip_new	A numeric vector

Value

A data frame

Examples

```
library(dplyr)
library(rsample)
cd_2019 <- c(824, 169, 342, 423, 441, 507, 559,
             173, 764, 780, 244, 527, 417, 800,
             602, 494, 345, 780, 780, 527, 556,
             559, 238, 288, 244, 353, 169, 556,
             824, 169, 342, 423, 441, 507, 559)
vl_2019 <- c(40, 11388, 38961, 40, 75, 4095, 103,
             11388, 46, 103, 11388, 40, 0, 11388,
             0,  4095,  40,  93,  49,  49,  49,
             4095,  6837, 38961, 38961, 0, 0, 93,
             40, 11388, 38961, 40, 75, 4095, 103)
cd_2021 <- c(992, 275, 331, 454, 479, 553,  496,
             230, 605, 432, 170, 670, 238,  238,
             634, 422, 429, 513, 327, 465,  479,
             661, 382, 364, 109, 398, 209, 1960,
             992, 275, 331, 454, 479, 553,  496)
vl_2021 <- c(80, 1690,  5113,  71, 289, 3063,  0,
             262,  0, 15089,  13016, 1513, 60, 60,
             49248, 159308, 56, 0, 516675, 49, 237,
             84,  292,  414, 26176,  62, 126,  93,
             80, 1690, 5113,    71, 289, 3063,   0)
cd_2022 <- c(700, 127, 127, 547, 547, 547, 777,
             149, 628, 614, 253, 918, 326, 326,
             574, 361, 253, 726, 659, 596, 427,
             447, 326, 253, 248, 326, 260, 918,
             700, 127, 127, 547, 547, 547, 777)
vl_2022 <- c(0,   0,  53250,   0,  40,  1901, 0,
             955,   0,   0,    0,   0,  40,   0,
             49248, 159308, 56, 0, 516675, 49, 237,
             0,    23601,  0,  40,   0,   0,   0,
             0,   0,    0,    0,   0,   0,   0)
x <- cbind(cd_2019, vl_2019, cd_2021, vl_2021, cd_2022, vl_2022) |>
as.data.frame()
set.seed(123)
hi_data <- rsample::initial_split(x)
set.seed(123)
hiv_data <- hi_data |>
rsample::training()
hu <- 5
plty <- 1.131656e-09
epo <- 176
vip_featured <- c("cd_2022")
vip_features <- c("cd_2019", "vl_2019", "cd_2021", "vl_2021", "vl_2022")
set.seed(123)
vi_train <- rsample::initial_split(x)
set.seed(123)
vip_train <- vi_train |>
rsample::training() |>
dplyr::select(rsample::all_of(vip_features))
vip_new <- vip_train[1,]
```

```
viralx_nn(vip_featured, hiv_data, hu, plty, epo, vip_train, vip_new)
```

---

viralx_nn_glob   *Global Explainers for Neural Network Models*

Description

The viralx_nn_glob function is designed to provide global explanations for the specified neural network model.

Usage

```
viralx_nn_glob(vip_featured, hiv_data, hu, plty, epo, vip_train, v_train)
```

Arguments

vip_featured	A character value specifying the variable of interest for which you want to explain predictions.
hiv_data	A data frame containing the dataset used for training the neural network model.
hu	A numeric value representing the number of hidden units in the neural network.
plty	A numeric value representing the penalty term for the neural network model.
epo	A numeric value specifying the number of epochs for training the neural network.
vip_train	A data frame containing the training data used for generating global explanations.
v_train	A numeric vector representing the target variable for the global explanations.

Value

A list containing global explanations for the specified neural network model.

Examples

```
library(dplyr)
library(rsample)
cd_2019 <- c(824, 169, 342, 423, 441, 507, 559,
             173, 764, 780, 244, 527, 417, 800,
             602, 494, 345, 780, 780, 527, 556,
             559, 238, 288, 244, 353, 169, 556,
             824, 169, 342, 423, 441, 507, 559)
vl_2019 <- c(40, 11388, 38961, 40, 75, 4095, 103,
             11388, 46, 103, 11388, 40, 0, 11388,
             0,  4095,  40, 93, 49,  49,  49,
             4095,  6837, 38961, 38961, 0, 0, 93,
             40, 11388, 38961, 40, 75, 4095, 103)
cd_2021 <- c(992, 275, 331, 454, 479, 553,  496,
             230, 605, 432, 170, 670, 238,  238,
```

```
                 634, 422, 429, 513, 327, 465,  479,
                 661, 382, 364, 109, 398, 209, 1960,
                 992, 275, 331, 454, 479, 553,  496)
   vl_2021 <- c(80, 1690,  5113,  71,  289,  3063,  0,
                 262,  0,  15089,  13016, 1513, 60, 60,
                 49248, 159308, 56, 0, 516675, 49, 237,
                 84,  292,  414, 26176,  62,  126,  93,
                 80, 1690, 5113,    71, 289, 3063,   0)
   cd_2022 <- c(700, 127, 127, 547, 547, 547, 777,
                 149, 628, 614, 253, 918, 326, 326,
                 574, 361, 253, 726, 659, 596, 427,
                 447, 326, 253, 248, 326, 260, 918,
                 700, 127, 127, 547, 547, 547, 777)
   vl_2022 <- c(0,   0,  53250,   0,   40,   1901, 0,
                 955,   0,    0,    0,   0,   40,   0,
                 49248, 159308, 56, 0, 516675, 49, 237,
                 0,    23601,  0,   40,   0,   0,   0,
                 0,   0,    0,    0,   0,    0,   0)
   x <- cbind(cd_2019, vl_2019, cd_2021, vl_2021, cd_2022, vl_2022) |>
   as.data.frame()
   set.seed(123)
   hi_data <- rsample::initial_split(x)
   set.seed(123)
   hiv_data <- hi_data |>
   rsample::training()
   hu <- 5
   plty <- 1.131656e-09
   epo <- 176
   vip_featured <- c("cd_2022")
   vip_features <- c("cd_2019", "vl_2019", "cd_2021", "vl_2021", "vl_2022")
   set.seed(123)
   vi_train <- rsample::initial_split(x)
   set.seed(123)
   vip_train <- vi_train |>
   rsample::training() |>
   dplyr::select(rsample::all_of(vip_features))
   v_train <- vi_train |>
   rsample::training() |>
   dplyr::select(rsample::all_of(vip_featured))
   viralx_nn_glob(vip_featured, hiv_data, hu, plty, epo, vip_train, v_train)
```

---

viralx_nn_shap   *Explain Neural Network Model Using SHAP Values*

Description

Explains the predictions of a neural network model using SHAP (Shapley Additive Explanations)
values. It utilizes the DALEXtra and DALEX packages to provide SHAP-based explanations for
the specified model.

Usage

```
viralx_nn_shap(
  vip_featured,
  hiv_data,
  hu,
  plty,
  epo,
  vip_train,
  vip_new,
  orderings
)
```

Arguments

vip_featured	A character value
hiv_data	A data frame
hu	A numeric value
plty	A numeric value
epo	A numeric value
vip_train	A data frame
vip_new	A numeric vector
orderings	A numeric value

Value

A data frame

Examples

```
library(dplyr)
library(rsample)
cd_2019 <- c(824, 169, 342, 423, 441, 507, 559,
             173, 764, 780, 244, 527, 417, 800,
             602, 494, 345, 780, 780, 527, 556,
             559, 238, 288, 244, 353, 169, 556,
             824, 169, 342, 423, 441, 507, 559)
vl_2019 <- c(40, 11388, 38961, 40, 75, 4095, 103,
             11388, 46, 103, 11388, 40, 0, 11388,
             0,  4095,  40, 93, 49, 49, 49,
             4095, 6837, 38961, 38961, 0, 0, 93,
             40, 11388, 38961, 40, 75, 4095, 103)
cd_2021 <- c(992, 275, 331, 454, 479, 553,  496,
             230, 605, 432, 170, 670, 238,  238,
             634, 422, 429, 513, 327, 465,  479,
             661, 382, 364, 109, 398, 209, 1960,
             992, 275, 331, 454, 479, 553,  496)
vl_2021 <- c(80, 1690,  5113,  71,  289,  3063,  0,
             262,  0,  15089,  13016, 1513, 60, 60,
```

```
               49248, 159308, 56, 0, 516675, 49, 237,
               84,  292,  414, 26176,  62,  126,  93,
               80, 1690, 5113,    71, 289, 3063,   0)
cd_2022 <- c(700, 127, 127, 547, 547, 547, 777,
               149, 628, 614, 253, 918, 326, 326,
               574, 361, 253, 726, 659, 596, 427,
               447, 326, 253, 248, 326, 260, 918,
               700, 127, 127, 547, 547, 547, 777)
vl_2022 <- c(0,   0,   53250,  0,   40,   1901, 0,
               955,   0,   0,     0,   0,    40,   0,
               49248, 159308, 56, 0, 516675, 49, 237,
               0,     23601,  0,  40,  0,    0,    0,
               0,     0,      0,  0,   0,    0,    0)
x <- cbind(cd_2019, vl_2019, cd_2021, vl_2021, cd_2022, vl_2022) |>
as.data.frame()
set.seed(123)
hi_data <- rsample::initial_split(x)
set.seed(123)
hiv_data <- hi_data |>
rsample::training()
hu <- 5
plty <- 1.131656e-09
epo <- 176
vip_featured <- c("cd_2022")
vip_features <- c("cd_2019", "vl_2019", "cd_2021", "vl_2021", "vl_2022")
set.seed(123)
vi_train <- rsample::initial_split(x)
set.seed(123)
vip_train <- vi_train |>
rsample::training() |>
dplyr::select(rsample::all_of(vip_features))
vip_new <- vip_train[1,]
orderings <- 20
viralx_nn_shap(vip_featured, hiv_data, hu, plty, epo, vip_train, vip_new, orderings)
```

---

viralx_nn_vis   *Visualize SHAP Values for Neural Network Model*

Description

Visualizes SHAP (Shapley Additive Explanations) values for a neural network model by employing the DALEXtra and DALEX packages to provide visual insights into the impact of a specified variable on the model's predictions.

Usage

```
viralx_nn_vis(
  vip_featured,
  hiv_data,
  hu,
```

```
  plty,
  epo,
  vip_train,
  vip_new,
  orderings
)
```

Arguments

`vip_featured`	A character value
`hiv_data`	A data frame
`hu`	A numeric value
`plty`	A numeric value
`epo`	A numeric value
`vip_train`	A data frame
`vip_new`	A numeric vector
`orderings`	A numeric value

Value

A ggplot object

Examples

```
library(dplyr)
library(rsample)
cd_2019 <- c(824, 169, 342, 423, 441, 507, 559,
             173, 764, 780, 244, 527, 417, 800,
             602, 494, 345, 780, 780, 527, 556,
             559, 238, 288, 244, 353, 169, 556,
             824, 169, 342, 423, 441, 507, 559)
vl_2019 <- c(40, 11388, 38961, 40, 75, 4095, 103,
             11388, 46, 103, 11388, 40, 0, 11388,
             0,  4095,  40,  93,  49,  49,  49,
             4095,  6837, 38961, 38961, 0, 0, 93,
             40, 11388, 38961, 40, 75, 4095, 103)
cd_2021 <- c(992, 275, 331, 454, 479, 553,  496,
             230, 605, 432, 170, 670, 238,  238,
             634, 422, 429, 513, 327, 465,  479,
             661, 382, 364, 109, 398, 209, 1960,
             992, 275, 331, 454, 479, 553,  496)
vl_2021 <- c(80, 1690,  5113,  71,  289, 3063,  0,
             262, 0,  15089,  13016, 1513, 60, 60,
             49248, 159308, 56, 0, 516675, 49, 237,
             84,  292,  414, 26176,  62,  126,  93,
             80, 1690, 5113,   71, 289, 3063,   0)
cd_2022 <- c(700, 127, 127, 547, 547, 547, 777,
             149, 628, 614, 253, 918, 326, 326,
             574, 361, 253, 726, 659, 596, 427,
```

```
                447, 326, 253, 248, 326, 260, 918,
                700, 127, 127, 547, 547, 547, 777)
vl_2022 <- c(0,   0,   53250,   0,   40,   1901, 0,
                955,   0,    0,    0,    0,    40,   0,
                49248, 159308, 56, 0, 516675, 49, 237,
                0,    23601,   0,   40,   0,    0,    0,
                0,    0,    0,    0,    0,    0,    0)
x <- cbind(cd_2019, vl_2019, cd_2021, vl_2021, cd_2022, vl_2022) |>
as.data.frame()
set.seed(123)
hi_data <- rsample::initial_split(x)
set.seed(123)
hiv_data <- hi_data |>
rsample::training()
hu <- 5
plty <- 1.131656e-09
epo <- 176
vip_featured <- c("cd_2022")
vip_features <- c("cd_2019", "vl_2019", "cd_2021", "vl_2021", "vl_2022")
set.seed(123)
vi_train <- rsample::initial_split(x)
set.seed(123)
vip_train <- vi_train |>
rsample::training() |>
dplyr::select(rsample::all_of(vip_features))
vip_new <- vip_train[1,]
orderings <- 20
viralx_nn_vis(vip_featured, hiv_data, hu, plty, epo, vip_train, vip_new, orderings)
```

Index