

Package ‘viralx’

September 13, 2023

Title Explainers for Regression Models in HIV Research

Version 1.1.0

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](https://arxiv.org/abs/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 7.2.3

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

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

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

Repository CRAN

Date/Publication 2023-09-13 07:10:02 UTC

R topics documented:

glob_nn_vis	2
viralx_mars	3
viralx_mars_shap	5
viralx_mars_vis	7
viralx_nn	9

viralx_nn_glob	11
viralx_nn_shap	13
viralx_nn_vis	15

Index	18
--------------	-----------

glob_nn_vis	<i>Global Visualization of SHAP Values for Neural Network Model</i>
-------------	---

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

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

```

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

<code>vip_featured</code>	A character value
<code>hiv_data</code>	A data frame
<code>nt</code>	A numeric value
<code>pd</code>	A numeric value
<code>pru</code>	A character value
<code>vip_train</code>	A data frame
<code>vip_new</code>	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

<code>vip_featured</code>	A character value
<code>hiv_data</code>	A data frame
<code>nt</code>	A numeric value
<code>pd</code>	A numeric value
<code>pru</code>	A character value
<code>vip_train</code>	A data frame
<code>vip_new</code>	A numeric vector
<code>orderings</code>	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)
v1_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, v1_2019, cd_2021, v1_2021, cd_2022, v1_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", "v1_2019", "cd_2021", "v1_2021", "v1_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	<i>Visualize SHAP Values for Multivariate Adaptive Regression Splines Model</i>
-----------------	---

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

<code>vip_featured</code>	A character value
<code>hiv_data</code>	A data frame
<code>nt</code>	A numeric value
<code>pd</code>	A numeric value
<code>pru</code>	A character value
<code>vip_train</code>	A data frame
<code>vip_new</code>	A numeric vector
<code>orderings</code>	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

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

<code>vip_featured</code>	A character value
<code>hiv_data</code>	A data frame
<code>hu</code>	A numeric value
<code>plty</code>	A numeric value
<code>epo</code>	A numeric value
<code>vip_train</code>	A data frame
<code>vip_new</code>	A numeric vector
<code>orderings</code>	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

glob_nn_vis, [2](#)

viralx_mars, [3](#)

viralx_mars_shap, [5](#)

viralx_mars_vis, [7](#)

viralx_nn, [9](#)

viralx_nn_glob, [11](#)

viralx_nn_shap, [13](#)

viralx_nn_vis, [15](#)