Package ‘viraldomain’

May 28, 2025

Title Applicability Domain Methods of Viral Load and CD4 Lymphocytes

Version 0.0.7

Description Provides methods for assessing the applicability domain of models that predict viral load and CD4 (Cluster of Differentiation 4) lymphocyte counts. These methods help determine the extent of extrapolation when making predictions.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 3.5)

LazyData true

Imports applicable, dplyr, earth, kknn, magrittr, parsnip, ranger, recipes, stats, tidyselect, workflows

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2025-05-27 23:30:02 UTC

Contents

knn_domain_score — *Calculate the K-Nearest Neighbor model domain applicability score*

Description

This function fits a K-Nearest Neighbor (KNN) model to the provided data and computes a domain applicability score based on PCA distances.

Usage

```
knn_domain_score(
  featured_col,
  train_data,
  knn_hyperparameters,
  test_data,
  threshold_value
)
```

Arguments

`featured_col`	The name of the response variable to predict.
`train_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.
`test_data`	The test dataset for making predictions.
`threshold_value`	The threshold value used for computing domain scores.

Value

A data frame containing the computed domain scores for each observation in the test dataset.

Examples

```
set.seed(123)
library(dplyr)
library(magrittr)
featured_col <- "cd_2022"
# Specifying features for training and testing procedures
train_data = viral %>%
  dplyr::select(cd_2022, vl_2022)
test_data = sero
knn_hyperparameters <- list(neighbors = 5, weight_func = "optimal", dist_power = 0.3304783)
threshold_value <- 0.99
knn_domain_score(featured_col, train_data, knn_hyperparameters, test_data, threshold_value)
```

`mars_domain_score` *Calculate the MARS model domain applicability score*

Description

This function fits a MARS (Multivariate Adaptive Regression Splines) model to the provided data and computes a domain applicability score based on PCA distances.

Usage

```
mars_domain_score(
  featured_col,
  train_data,
  mars_hyperparameters,
  test_data,
  threshold_value
)
```

Arguments

`featured_col`	The name of the featured column.
`train_data`	A data frame containing the training data.
`mars_hyperparameters`	A list of hyperparameters for the MARS model, including: • `num_terms`: The number of terms to include in the MARS model. • `prod_degree`: The degree of interaction terms to include. • `prune_method`: The method used for pruning the MARS model.
`test_data`	A data frame containing the test data.
`threshold_value`	The threshold value for the domain score.

Value

A tibble with the domain applicability scores.

Examples

```
set.seed(123)
library(dplyr)
featured_col <- "cd_2022"
# Specifying features for training and testing procedures
train_data = viral %>%
  dplyr::select(cd_2022, vl_2022)
test_data = sero
mars_hyperparameters <- list(num_terms = 3, prod_degree = 1, prune_method = "none")
threshold_value <- 0.99
# Call the function
```

`nn_domain_score` *Calculate the Neural Network model domain applicability score*

Description

This function fits a Neural Network model to the provided data and computes a domain applicability score based on PCA distances.

Usage

```
nn_domain_score(
  featured_col,
  train_data,
  nn_hyperparameters,
  test_data,
  threshold_value
)
```

Arguments

`featured_col`	The name of the featured column in the training data.
`train_data`	The training data used to fit the Neural Network model.
`nn_hyperparameters`	A list of Neural Network hyperparameters, including hidden_units, penalty, and epochs.
`test_data`	The testing domain data used to calculate the domain applicability score.
`threshold_value`	The threshold value for domain applicability scoring.

Value

A tibble with the domain applicability scores.

Examples

```
set.seed(123)
library(dplyr)
featured_col <- "cd_2022"
# Specifying features for training and testing procedures
train_data = viral %>%
  dplyr::select(cd_2022, vl_2022)
test_data = sero
nn_hyperparameters <- list(hidden_units = 1, penalty = 0.3746312,  epochs =  480)
threshold_value <- 0.99
nn_domain_score(featured_col, train_data, nn_hyperparameters, test_data, threshold_value)
```

`rf_domain_score` *Calculate the Random Forest Model Domain Applicability Score*

Description

This function fits a Random Forest model to the provided data and computes a domain applicability score based on PCA distances.

Usage

```
rf_domain_score(
  featured_col,
  train_data,
  rf_hyperparameters,
  test_data,
  threshold_value
)
```

Arguments

`featured_col`	A character string specifying the name of the response variable to predict.
`train_data`	A data frame containing predictor variables and the response variable for training the model.
`rf_hyperparameters`	A list of hyperparameters for the Random Forest model, including: • `mtry`: Number of predictors sampled at each split. • `min_n`: Minimum number of data points in a node for further splitting. • `trees`: Number of trees in the ensemble.
`test_data`	A data frame for making predictions.
`threshold_value`	A numeric threshold value used for computing domain applicability scores.

Details

Random Forest creates a large number of decision trees, each independent of the others. The final prediction combines the predictions from all individual trees. This function uses the `ranger` engine for fitting regression models.

Value

A data frame containing the computed domain applicability scores for each observation in the test dataset.

Examples

```
set.seed(123)
library(dplyr)
featured_col <- "cd_2022"
train_data <- viral %>%
  dplyr::select(cd_2022, vl_2022)
test_data <- sero
rf_hyperparameters <- list(mtry = 2, min_n = 5, trees = 500)
threshold_value <- 0.99
rf_domain_score(featured_col, train_data, rf_hyperparameters, test_data, threshold_value)
```

`sero` *Seropositive Data for Applicability Domain Testing*

Description

This dataset is designed for testing the applicability domain of methods related to HIV research. It provides a tibble with 53 rows and 2 columns containing numeric measurements of CD4 lymphocyte counts (cd_2022) and viral load (vl_2022) for seropositive individuals in 2022. These measurements are vital indicators of HIV disease status. This dataset is ideal for evaluating the performance and suitability of various HIV-predictive models and as an aid in developing diagnostic tools within a seropositive context.

Usage

```
data(sero)
```

Format

A tibble (data frame) with 53 rows and 2 columns.

Note

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

Author(s)

Juan Pablo Acuña González acua6307@gmail.com

Examples

```
data(sero)
sero
```

`viral` *Predictive Modeling Data for Viral Load and CD4 Lymphocyte Counts*

Description

This dataset serves as input for predictive modeling tasks related to HIV research. It contains numeric measurements of CD4 lymphocyte counts (cd) and viral load (vl) at three different time points: 2019, 2021, and 2022. These measurements are crucial indicators of HIV disease progression.

Usage

```
data(viral)
```

Format

A tibble (data frame) with 35 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 acua6307@gmail.com

Examples

```
data(viral)
viral
```

Index