

Package ‘viralmodels’

July 22, 2025

Title Viral Load and CD4 Lymphocytes Regression Models

Version 1.3.4

Description

Provides a comprehensive framework for building, evaluating, and visualizing regression models for analyzing viral load and CD4 (Cluster of Differentiation 4) lymphocytes data. It leverages the principles of the tidymodels ecosystem of Max Kuhn and Hadley Wickham (2020) <<https://www.tidymodels.org>> to offer a user-friendly experience in model development. This package includes functions for data preprocessing, feature engineering, model training, tuning, and evaluation, along with visualization tools to enhance the interpretation of model results. It is specifically designed for researchers in biostatistics, computational biology, and HIV research who aim to perform reproducible and rigorous analyses to gain insights into disease dynamics. The main focus is on improving the understanding of the relationships between viral load, CD4 lymphocytes, and other relevant covariates to contribute to HIV research and the visibility of vulnerable seropositive populations.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Suggests earth, nnet, rpart, testthat (>= 3.0.0)

Config/testthat/edition 3

Imports baguette, Cubist, dials, dplyr, glmnet, hardhat, kernlab, kkn, magrittr, parsnip, purrr, ranger, recipes, rsample, rules, stats, tidyselect, tune, viraldomain, workflows, workflowsets

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 09:10:02 UTC

Contents

viralmodel	2
viralpreds	3
viraltab	4
viralvis	6
Index	8

viralmodel	<i>Select best model</i>
------------	--------------------------

Description

Returns performance metrics for a selected model

Usage

```
viralmodel(output, modelo)
```

Arguments

- output A non-ranked viraltab output
- modelo A character value

Value

A table with a single model hyperparameters

Examples

```
library(dplyr)
library(magrittr)
library(baguette)
library(kernlab)
library(kknn)
library(ranger)
library(rules)
library(glmnet)
# Define the function to impute values in the undetectable range
set.seed(123)
impute_undetectable <- function(column) {
  ifelse(column <= 40,
    rexp(sum(column <= 40), rate = 1/13) + 1,
    column)
}
# Apply the function to all vl columns using purrr's map_dfc
library(viraldomain)
data("viral", package = "viraldomain")
viral_imputed <- viral %>%
```

```

mutate(across(starts_with("vl"), ~impute_undetectable(.x)))
traindata <- viral_imputed
semilla <- 1501
target <- "cd_2022"
viralvars <- c("vl_2019", "vl_2021", "vl_2022")
logbase <- 10
pliegues <- 2
repeticiones <- 1
rejilla <- 1
modelo <- "simple_rf"
set.seed(123)
viraltab(traindata, semilla, target, viralvars, logbase, pliegues,
repeticiones, rejilla, rank_output = FALSE) %>% viralmodel(modelo)

```

viralpreds

*Predict Viral Load or CD4 Count using Many Models***Description**

This function predicts viral load or CD4 count values based on multiple machine learning models using cross-validation. It allows users to specify two types of predictions: normal predictions on the full dataset or observation-by-observation (obs-by-obs) predictions.

Usage

```
viralpreds(output, semilla, data, prediction_type = "full")
```

Arguments

output	A non-ranked viraltab output
semilla	An integer specifying the seed for random number generation to ensure reproducibility.
data	A data frame containing the predictors and the target variable.
prediction_type	A character string specifying the type of predictions to perform. Use "full" (default) to perform predictions on the full dataset at once, or "batch" to perform predictions in a smaller size batches of data.

Value

A list containing two elements: predictions (a vector of predicted values for the target variable) and RMSE (the root mean square error of the best model).

Examples

```
library(dplyr)
library(magrittr)
library(bagette)
library(kernlab)
library(kknn)
library(ranger)
library(rules)
library(glmnet)
# Define the function to impute values in the undetectable range
set.seed(123)
impute_undetectable <- function(column) {
  ifelse(column <= 40,
    rexp(sum(column <= 40), rate = 1/13) + 1,
    column)
}
# Apply the function to all vl columns using purrr's map_dfc
library(viraldomain)
data("viral", package = "viraldomain")
viral_imputed <- viral %>%
mutate(across(starts_with("vl"), ~impute_undetectable(.x)))
traindata <- viral_imputed
target <- "cd_2022"
viralvars <- c("vl_2019", "vl_2021", "vl_2022")
logbase <- 10
pliegues <- 5
repeticiones <- 2
rejilla <- 2
semilla <- 123
viraltab(traindata, semilla, target, viralvars, logbase, pliegues,
repeticiones, rejilla, rank_output = FALSE) %>%
viralpreds(semilla, traindata, prediction_type = "full")
```

viraltab

Competing models table

Description

Trains and optimizes a series of regression models for viral load or CD4 counts

Usage

```
viraltab(
  traindata,
  semilla,
  target,
  viralvars,
  logbase,
```

```

    pliegues,
    repeticiones,
    rejilla,
    rank_output = TRUE
  )

```

Arguments

traindata	A data frame
semilla	A numeric value
target	A character value
viralvars	Vector of variable names related to viral data.
logbase	The base for logarithmic transformations.
pliegues	A numeric value
repeticiones	A numeric value
rejilla	A numeric value
rank_output	Logical value. If TRUE, returns ranked output; if FALSE, returns unranked output.

Value

A table of competing models

Examples

```

library(dplyr)
library(magrittr)
library(baguette)
library(kernlab)
library(kknn)
library(ranger)
library(rules)
library(glmnet)
# Define the function to impute values in the undetectable range
impute_undetectable <- function(column) {
  set.seed(123)
  ifelse(column <= 40,
    rexp(sum(column <= 40), rate = 1/13) + 1,
    column)
  }
library(viraldomain)
data("viral", package = "viraldomain")
viral_imputed <- viral %>%
mutate(across(starts_with("v1"), ~impute_undetectable(.x)))
traindata <- viral_imputed
semilla <- 1501
target <- "cd_2022"
viralvars <- c("v1_2019", "v1_2021", "v1_2022")
logbase <- 10

```

```

pliegues <- 2
repeticiones <- 1
rejilla <- 1
set.seed(123)
viraltab(traindata, semilla, target, viralvars, logbase, pliegues,
repeticiones, rejilla, rank_output = TRUE)

```

viralvis

Competing models plot

Description

Plots the rankings of a series of regression models for viral load or CD4 counts

Usage

```
viralvis(output)
```

Arguments

output A non-ranked viraltab output

Value

A plot of ranking models

Examples

```

library(dplyr)
library(magrittr)
library(baguette)
library(kernlab)
library(kknn)
library(ranger)
library(rules)
library(glmnet)
# Define the function to impute values in the undetectable range
set.seed(123)
impute_undetectable <- function(column) {
  ifelse(column <= 40,
    rexp(sum(column <= 40), rate = 1/13) + 1,
    column)
}
# Apply the function to all vl columns using purrr's map_dfc
library(viraldomain)
data("viral", package = "viraldomain")
viral_imputed <- viral %>%
mutate(across(starts_with("vl"), ~impute_undetectable(.x)))
traindata <- viral_imputed

```

```
semilla <- 1501
target <- "cd_2022"
viralvars <- c("vl_2019", "vl_2021", "vl_2022")
logbase <- 10
pliegues <- 2
repeticiones <- 1
rejilla <- 1
set.seed(123)
viraltab(traindata, semilla, target, viralvars, logbase, pliegues,
repeticiones, rejilla, rank_output = FALSE) %>% viralvis()
```

Index

viralmodel, 2
viralpreds, 3
viraltab, 4
viralvis, 6