Package: mlspatial (via r-universe)

June 3, 2026

Title Machine Learning and Mapping for Spatial Epidemiology

Version 0.1.1

Description Provides tools for the integration, visualisation, and modelling of spatial epidemiological data using the method described in Azeez, A., & Noel, C. (2025). 'Predictive Modelling and Spatial Distribution of Pancreatic Cancer in Africa Using Machine Learning-Based Spatial Model' <doi:10.5281/zenodo.16529986> and <doi:10.5281/zenodo.16529016>. It facilitates the analysis of geographic health data by combining modern spatial mapping tools with advanced machine learning (ML) algorithms. 'mlspatial' enables users to import and pre-process shapefile and associated demographic or disease incidence data, generate richly annotated thematic maps, and apply predictive models, including Random Forest, 'XGBoost', and Support Vector Regression, to identify spatial patterns and risk factors. It is suited for spatial epidemiologists, public health researchers, and GIS analysts aiming to uncover hidden geographic patterns in health-related outcomes and inform evidence-based interventions.

RoxygenNote 7.3.3

Suggests knitr, rmarkdown, tidyr, kernlab, writexl, testthat (>= 3.0.0)

VignetteBuilder knitr

Depends R (>= 4.1)

Imports sf, readxl, dplyr, ggplot2, randomForest, xgboost, e1071, caret, tmap, spdep, ggpubr, stats, methods

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Config/testthat/edition 3

NeedsCompilation no

Author Adeboye Azeez [aut, cre], Colin Noel [aut]

Maintainer Adeboye Azeez <`azizadeboye@gmail.com`>

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev make libicu-dev libpng-dev libuv1-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev zlib1g-dev

Repository https://azizadeboye.r-universe.dev

Date/Publication 2026-03-30 06:33:46 UTC

RemoteUrl https://github.com/azizadeboye/mlspatial

RemoteRef HEAD

RemoteSha 28b84a17ce6170ae7360bc8e757d944aafcb0446

Contents

`africa_shp` *Africa shapefile data*

Description

A dataset containing spatial polygons of Africa.

Usage

```
africa_shp
```

Format

An `sf` object with spatial features.

Source

Your data source

`africa_shps` *Africa shapefile data 2*

Description

A dataset containing spatial polygons of Africa.

Usage

```
africa_shps
```

Format

An `sf` object with spatial features.

Source

Your data source

`compute_spatial_autocorr` *Compute Moran's I & LISA, classify clusters*

Description

Computes global and local Moran's I to assess spatial autocorrelation and classifies observations into spatial cluster types (e.g., High-High).

Usage

```
compute_spatial_autocorr(sf_data, values, signif = 0.05)
```

Arguments

`sf_data`	An `sf` object containing spatial features.
`values`	A numeric vector or column name with the variable to test.
`signif`	Numeric significance level threshold for clusters (default 0.05).

Value

A named list with elements:

- `data`: An `sf` object with added columns for standardized values, spatial lag, local Moran's I values, z-scores, p-values, and cluster classification.
- `moran`: An object of class `htest` with global Moran's I test results.

Examples

```
library(sf)
library(spdep)
library(dplyr)

#Load and prepare spatial data
mapdata <- st_read(system.file("shape/nc.shp", package="sf"), quiet = TRUE)
mapdata <- st_make_valid(mapdata)

#Variable to analyze
values <- rnorm(nrow(mapdata))

#Run function
result <- compute_spatial_autocorr(mapdata, values, signif = 0.05)

#Inspect results
head(result$data)
result$moran
```

`eval_model` *Get RMSE/MAE/R² metrics on training data*

Description

Evaluate Model Performance by calculating RMSE, MAE, and R² metrics.

Usage

```
eval_model(model, data, formula, model_type = c("rf", "xgb", "svr"))
```

Arguments

`model`	A trained model
`data`	A data frame
`formula`	A formula object
`model_type`	Character string: one of "rf", "xgb", or "svr"

Value

A numeric value representing the model's accuracy

`global_variables_eval` *Declare known global variables to suppress R CMD check NOTE Global variables used in evaluation functions*

Description

This is to suppress R CMD check notes about undefined global variables.

`join_data` *Join spatial and incidence datasets*

Description

Join spatial and incidence datasets

Usage

```
join_data(sf_data, tbl_data, by)
```

Arguments

`sf_data`	sf object
`tbl_data`	tibble of incidence
`by`	Column name to join on

Value

sf object with joined attributes

`load_incidence_data` *Load incidence data from Excel*

Description

Load incidence data from Excel

Usage

```
load_incidence_data(xlsx_path)
```

Arguments

`xlsx_path`	Path to Excel file

Value

tibble of data

`load_shapefile` *Load shapefile as sf + optionally convert to sp*

Description

Load shapefile as sf + optionally convert to sp

Usage

```
load_shapefile(shp_path, to_sp = FALSE)
```

Arguments

`shp_path`	Path to shapefile (.shp)
`to_sp`	logical: also return Spatial object?

Value

list with sf and optionally sp object

`model_evaluation_examples` *Examples for model evaluation functions*

Description

Examples for model evaluation functions

Examples

```
library(randomForest)
library(caret)
data(panc_incidence)
mapdata <- join_data(africa_shp, panc_incidence, by = "NAME")
rf_model <- randomForest(incidence ~ female + male + agea + ageb + agec + fagea + fageb + fagec +
magea + mageb + magec + yrb + yrc + yrd + yre, data = mapdata, ntree = 500,
importance = TRUE)

rf_preds <- predict(rf_model, newdata = mapdata)
rf_metrics <- postResample(pred = rf_preds, obs = mapdata$incidence)
print(rf_metrics)
```

`panc_incidence` *Pancreatic Cancer Incidence Data*

Description

This dataset contains pancreatic cancer incidence rates across African countries.

Usage

```
data(panc_incidence)
```

Format

A data frame with the following variables:

NAME Character. Name of the country.

incidence Double. Incidence rate per 100,000 population.

female Double. Female pancreatic cancer patients.

male Double. Male pancreatic cancer patients.

ageb Double. Patients age between 20-54 years.

agec Double. Patients age above 55 years.

agea Double. Patients age below 20 years.

fageb Double. Female patients age between 20-54 years.

fagec Double. Female patients age above 55 years.

fagea Double. Female patients age below 20 years.

mageb Double. Male patients age between 20-54 years.

magec Double. Male patients age above 55 years.

magea Double. Male patients age below 20 years.

yra Double. Incidence rate in year 2017.

yrb Double. Incidence rate in year 2018.

yrc Double. Incidence rate in year 2019.

yrd Double. Incidence rate in year 2020.

yre Double. Incidence rate in year 2021.

Source

Global Burden of Disease (GBD) 2021 estimates, Seattle, United States https://vizhub.healthdata.org/gbd-results/

`panc_prevalence` *Pancreatic Cancer Prevalence Data*

Description

This dataset contains pancreatic cancer incidence rates across African countries.

Usage

```
data(panc_prevalence)
```

Format

A data frame with the following variables:

NAME Character. Name of the country.

prevalence Numeric. Prevalence rate per 100,000 population.

female Numeric. Female pancreatic cancer patients.

male Numeric. Male pancreatic cancer patients.

ageb Numeric. Patients age between 20-54 years.

agec Numeric. Patients age above 55 years.

agea Numeric. Patients age below 20 years.

fageb Numeric. Female patients age between 20-54 years.

fagec Numeric. Female patients age above 55 years.

fagea Numeric. Female patients age below 20 years.

mageb Numeric. Male patients age between 20-54 years.

magec Numeric. Male patients age above 55 years.

magea Numeric. Male patients age below 20 years.

yra Numeric. Incidence rate in year 2017.

yrb Numeric. Incidence rate in year 2018.

yrc Numeric. Incidence rate in year 2019.

yrd Numeric. Incidence rate in year 2020.

yre Numeric. Incidence rate in year 2021.

Source

Global Burden of Disease (GBD) 2021 estimates, Seattle, United States https://vizhub.healthdata.org/gbd-results/

`pancre_mort` *Pancreatic Cancer Mortality Data*

Description

This dataset contains pancreatic cancer incidence rates across African countries.

Usage

`data(pancre_mort)`

Format

A data frame with the following variables:

NAME Character. Name of the country.

mortality Numeric. Mortality rate per 100,000 population.

female Numeric. Female pancreatic cancer patients.

male Numeric. Male pancreatic cancer patients.

ageb Numeric. Patients age between 20-54 years.

agec Numeric. Patients age above 55 years.

agea Numeric. Patients age below 20 years.

fageb Numeric. Female patients age between 20-54 years.

fagec Numeric. Female patients age above 55 years.

fagea Numeric. Female patients age below 20 years.

mageb Numeric. Male patients age between 20-54 years.

magec Numeric. Male patients age above 55 years.

magea Numeric. Male patients age below 20 years.

- **yra** Numeric. Incidence rate in year 2017.
- **yrb** Numeric. Incidence rate in year 2018.
- **yrc** Numeric. Incidence rate in year 2019.
- **yrd** Numeric. Incidence rate in year 2020.
- **yre** Numeric. Incidence rate in year 2021.

Source

Global Burden of Disease (GBD) 2021 estimates, https://vizhub.healthdata.org/gbd-results/

`plot_map_grid` *Arrange Multiple tmap Plots in a Grid*

Description

Arrange a list of tmap objects into a grid layout.

Usage

```
plot_map_grid(maps, ncol = 2)
```

Arguments

`maps`	A list of tmap objects.
`ncol`	Number of columns in the grid (default is 2).

Value

A tmap object representing arranged maps.

Examples

```
library(sf)
library(tmap)

# Load sample spatial data
nc <- st_read(system.file("shape/nc.shp", package = "sf"), quiet = TRUE)

# Add mock variables to map
nc$var1 <- runif(nrow(nc), 0, 100)
nc$var2 <- runif(nrow(nc), 10, 200)

# Create individual maps
map1 <- tm_shape(nc) + tm_fill("var1", title = "Variable 1")
map2 <- tm_shape(nc) + tm_fill("var2", title = "Variable 2")

# Arrange the maps in a grid using your function
plot_map_grid(list(map1, map2), ncol = 2)
```

`plot_obs_vs_pred` *Plot observed vs predicted values with correlation*

Description

Creates a scatterplot of observed vs predicted values, with a 1:1 reference line and Pearson's R^2.

Usage

```
plot_obs_vs_pred(observed, predicted, title = "")
```

Arguments

`observed`	Numeric vector of observed values.
`predicted`	Numeric vector of predicted values.
`title`	String for the plot title (default: "").

Value

No return value; called for side effect of displaying a plot.

Examples

```
observed <- c(10, 20, 30, 40)
predicted <- c(12, 18, 33, 39)
plot_obs_vs_pred(observed, predicted, title = "Observed vs Predicted")
```

`plot_single_map` *Build a tmap for a single variable*

Description

Creates a thematic map using the `tmap` package for a single variable in an sf object.

Usage

```
plot_single_map(sf_data, var, title, palette = "reds")
```

Arguments

`sf_data`	An sf object containing spatial data.
`var`	Variable name as a string to map.
`title`	Legend title for the fill legend.
`palette`	Color palette for the map (default is "reds").

Value

A tmap object representing the thematic map.

Examples

```
library(sf)
# Create example sf object
nc <- st_read(system.file("shape/nc.shp", package = "sf"), quiet = TRUE)
nc$incidence <- runif(nrow(nc), 0, 100)

# Plot
p1 <- plot_single_map(nc, "incidence", "Incidence")
```

`train_rf` *Train Random Forest model*

Description

Trains a Random Forest regression model.

Usage

```
train_rf(data, formula, ntree = 500, seed = 123)
```

Arguments

`data`	A data frame containing the training data.
`formula`	A formula describing the model structure.
`ntree`	Number of trees to grow (default 500).
`seed`	Random seed for reproducibility (default 123).

Value

A trained randomForest model object.

Examples

```
library(randomForest)
data(mtcars)
rf_model <- train_rf(mtcars, mpg ~ cyl + hp + wt, ntree = 100)
print(rf_model)
```

`train_svr` *Train Support Vector Regression (SVR) model*

Description

Train Support

Usage

```
train_svr(data, formula)
```

Arguments

`data`	A data frame containing the training data.
`formula`	A formula specifying the model.

Details

Trains an SVR model using the radial kernel.

Value

A trained `svm` model object from the **e1071** package.

Examples

```
# Load required package
library(e1071)

# Use built-in dataset
data(mtcars)

# Define regression formula
svr_formula <- mpg ~ cyl + disp + hp + wt

# Train SVR model
svr_model <- train_svr(data = mtcars, formula = svr_formula)

# Print model summary
print(svr_model)

# Predict on the same data (for illustration)
preds <- predict(svr_model, newdata = mtcars)
head(preds)
```

`train_xgb` *Train XGBoost model*

Description

Train XGBoost model

Usage

```
train_xgb(data, formula, nrounds = 100, max_depth = 4, learning_rate = 0.1)
```

Arguments

`data`	A data frame with the training data.
`formula`	A formula defining the model structure.
`nrounds`	Number of boosting iterations.
`max_depth`	Maximum tree depth.
`learning_rate`	Learning rate for boosting.

Details

Trains an XGBoost regression model.

Value

A trained xgboost model object.

Examples

```
# Load required package
library(xgboost)

# Use built-in dataset
data(mtcars)

# Define regression formula
xgb_formula <- mpg ~ cyl + disp + hp + wt

# Train XGBoost model
xgb_model <- train_xgb(data = mtcars, formula = xgb_formula, nrounds = 50)

# Print model summary
print(xgb_model)
```

Index