Package ‘marsruntime’

May 14, 2026

Title Portable Mars Runtime Replay

Version 0.0.0

Description Loads, validates, and replays portable 'mars' ModelSpec artifacts from R. The package provides helpers for constructing design matrices, generating predictions, and, when the companion runtime helper is available, fitting portable model specifications for cross-language replay.

License Apache License (>= 2)

Encoding UTF-8

URL https://github.com/edithatogo/mars

BugReports https://github.com/edithatogo/mars/issues

Imports jsonlite

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Dylan A Mordaunt [aut, cre]

Maintainer Dylan A Mordaunt <dylan.mordaunt@vuw.ac.nz>

Repository CRAN

Date/Publication 2026-05-14 09:10:02 UTC

Contents

marsruntime-package *marsruntime: Portable R Runtime Replay for Mars*

Description

The ‘marsruntime‘ package exposes a portable R runtime surface for validated Mars ‘ModelSpec‘ artifacts. It can load, validate, replay, and, when the Rust runtime helper is available, fit portable model specifications.

Details

The package is designed to work both from the source tree and from an installed package. Runtime replay helpers fall back to pure R logic when the Rust runtime binary is unavailable. Training via ‘fit_model()‘ requires the Rust runtime helper and returns a portable ‘ModelSpec‘ that can be replayed through the same helpers.

Quick Start

Load a validated ‘ModelSpec‘, inspect its design matrix, and generate predictions with the same portable runtime surface:

```
library(marsruntime)
spec <- load_model_spec("path/to/model_spec.json")
design_matrix(spec, matrix(c(0, 1, 2), ncol = 1))
predict_model(spec, matrix(c(0, 1, 2), ncol = 1))
```

Training

If the Rust runtime helper is present, ‘fit_model()‘ can fit a portable specification from feature and response data:

```
if (nzchar(Sys.getenv("MARS_RUNTIME_BIN", unset = ""))) {
  fitted <- fit_model(matrix(c(0, 1, 2), ncol = 1), c(1, 3, 5))
  predict_model(fitted, matrix(c(0, 1, 2), ncol = 1))
}
```

Validation

The package includes conformance tests that replay shared fixtures and verify the runtime-backed training path when the Rust binary is available.

Documentation

Per-function help is available for ‘load_model_spec()‘, ‘validate_model_spec()‘, ‘design_matrix()‘, ‘predict_model()‘, and ‘fit_model()‘. The release docs describe the R publication path, package-check expectations, and the manual build path.

Examples

```
spec <- list(
  spec_version = "1.0",
  basis_terms = list(),
  coefficients = list(),
  feature_schema = list(n_features = 0)
)
validate_model_spec(spec)
```

`design_matrix` *Compute a Design Matrix from a Model Specification*

Description

Compute the design matrix for a portable mars model specification.

Usage

```
design_matrix(spec, rows)
```

Arguments

`spec`	A validated model specification.
`rows`	A numeric matrix or data frame of input rows.

Value

A numeric matrix of basis-function values.

`fit_model` *Fit a Portable Mars Model*

Description

Fit a mars model using the shared runtime bridge.

Usage

```
fit_model(
  x,
  y,
  max_terms = 21,
  max_degree = 1,
  penalty = 3.0,
  minspan = 0.0,
  endspan = 0.0,
```

```
  threshold = 0.001,
  allow_linear = TRUE,
  allow_missing = FALSE,
  categorical_features = integer(),
  feature_names = NULL,
  sample_weight = NULL
)
```

Arguments

x	Training feature matrix.
y	Training response vector.
max_terms	Maximum number of basis terms to retain.
max_degree	Maximum interaction degree.
penalty	Complexity penalty used during pruning.
minspan	Minimum span between knots.
endspan	End-span constraint for knot placement.
threshold	Stopping threshold for forward selection.
allow_linear	Whether to include linear terms.
allow_missing	Whether to include missing-value terms.
categorical_features	One-based indexes of categorical columns.
feature_names	Optional feature names for the exported spec.
sample_weight	Optional sample weights for the training rows.

Value

A validated portable model specification.

---

load_model_spec *Load a Portable Model Specification*

Description

Load and validate a portable mars model specification.

Usage

```
load_model_spec(path_or_json)
```

Arguments

path_or_json	Path to a JSON specification file or a JSON string.

Value

A validated model specification list.

predict_model *Predict from a Portable Model Specification*

Description

Predict from a portable mars model specification.

Usage

```
predict_model(spec, rows)
```

Arguments

spec	A validated model specification.
rows	A numeric matrix or data frame of input rows.

Value

A numeric vector of predictions.

validate_model_spec *Validate a Portable Model Specification*

Description

Validate a portable mars model specification.

Usage

```
validate_model_spec(spec)
```

Arguments

spec	A model specification object.

Value

Invisible TRUE on success.

Index