Package 'accept'

June 6, 2025

Title The Acute COPD Exacerbation Prediction Tool (ACCEPT)

Version 1.0.2

Description Allows clinicians to predict the rate and severity of future acute exacerbation in Chronic Obstructive Pulmonary Disease (COPD) patients, based on the clinical prediction models published in Adibi et al. (2020) <doi:10.1016/S2213-2600(19)30397-2> and Safari et al. (2022) <doi:10.1016/j.eclinm.2022.101574>.

Depends R (>= 4.0.0)

License GPL-3

Encoding UTF-8

LazyData true

Imports stats, tidyselect(>= 1.2.0), dplyr, reldist, splines, tibble, hardhat, vctrs, vetiver (>= 0.2.1)

Suggests jsonlite, plotly, spelling, testthat (>= 3.0.0)

RoxygenNote 7.3.2

Language en-US

Config/testthat/edition 3

NeedsCompilation no

Author Amin Adibi [aut, cre],
Mohsen Sadatsafavi [aut, cph],
Abdollah Safari [aut],
Ainsleigh Hill [aut]

Maintainer Amin Adibi <`adibi@alumni.ubc.ca`>

Repository CRAN

Date/Publication 2025-06-05 22:20:22 UTC

Contents

`accept` *A flexible version of ACCEPT 2.0 model, which imputes predictors using MICE approach.*

Description

A flexible version of ACCEPT 2.0 model, which imputes predictors using MICE approach.

Usage

```
accept(
  newdata,
  format = "tibble",
  version = "accept2",
  prediction_interval = FALSE,
  return_predictors = FALSE,
  ...
)
```

Arguments

`newdata`	new patient data with missing values to be imputed before prediction with the same format as accept samplePatients.
`format`	default is "tibble". Can also be set to "json".
`version`	indicates which version of ACCEPT needs to be called. Options include "accept1", "accept2", and "flexccept"
`prediction_interval`	default is FALSE. If set to TRUE, returns prediction intervals of the predictions.
`return_predictors`	default is FALSE. IF set to TRUE, returns the predictors along with prediction results.
`...`	for other versions of accept.

Value

patientData with prediction.

Examples

```
results <- accept(newdata = samplePatients)
```

`accept1` *Predicts COPD exacerbation rate by severity level based on Acute COPD Exacerbation Tool (ACCEPT)*

Description

Predicts COPD exacerbation rate by severity level based on Acute COPD Exacerbation Tool (ACCEPT)

Usage

```
accept1(
  patientData,
  random_sampling_N = 100,
  lastYrExacCol = "LastYrExacCount",
  lastYrSevExacCol = "LastYrSevExacCount",
  ...
)
```

Arguments

`patientData`	patient data matrix. Can have one or many patients in it
`random_sampling_N`	number of random sampling. Default is 100.
`lastYrExacCol`	the column specifying last year all exacerbation count
`lastYrSevExacCol`	the column specifying last year severe exacerbation count
`...`	for backward compatibility

Value

patientData with prediction

Examples

```
results <- accept1(samplePatients)
```

`accept2` *Predicts COPD exacerbation rate by severity level based on the updated accept2 model, which improves accuracy in patients without an exacerbation history.*

Description

Predicts COPD exacerbation rate by severity level based on the updated accept2 model, which improves accuracy in patients without an exacerbation history.

Usage

```
accept2(
  patientData,
  random_sampling_N = 100,
  lastYrExacCol = "LastYrExacCount",
  lastYrSevExacCol = "LastYrSevExacCount",
  KeepSGRQ = TRUE,
  KeepMeds = TRUE,
  ...
)
```

Arguments

`patientData`	patient data matrix. Can have one or many patients in it
`random_sampling_N`	number of random sampling. Default is 100.
`lastYrExacCol`	the column specifying last year all exacerbation count
`lastYrSevExacCol`	the column specifying last year severe exacerbation count
`KeepSGRQ`	default is TRUE. If set to false, the reduced model without SGRQ will be used.
`KeepMeds`	default is TRUE. If set to false, the reduced model without medication predictors will be used.
`...`	for backward compatibility

Value

patientData with prediction

Examples

```
results <- accept2(samplePatients)
```

`handler_predict.accept_model` *Model handler functions for API endpoint*

Description

Useful for supporting accept_model class in vetiver

Usage

```
## S3 method for class 'accept_model'
handler_predict(vetiver_model, ...)
```

Arguments

`vetiver_model`	A deployable [vetiver_model()] object
`...`	any other inputs

Value

A ‘handler_startup‘ function should return invisibly, while a ‘handler_predict‘ function should return a function with the signature ‘function(req)‘. The request body (‘req$body‘) consists of the new data at prediction time; this function should return predictions either as a tibble or as a list coercable to a tibble via [tibble::as_tibble()].

`plotExacerbations` *Creates bar graph comparing no treatment with azithromycin treatment*

Description

Creates bar graph comparing no treatment with azithromycin treatment

Usage

```
plotExacerbations(
  patientResults,
  type = "rate",
  interval = "PI",
  colors = c("#007bff", "rgb(204,204,204)")
)
```

Arguments

`patientResults`	patient results vector, produced by accept.
`type`	string: either "probability" or "rate"
`interval`	string: either "CI" or "PI" PI = Predicted Interval CI = Confidence Interval
`colors`	vector: a vector of colors to be used in the graph must be length 2 can use hexadecimal, rgb, or R color codes

Value

a bar graph

Examples

```
## Not run:
results <- accept1(samplePatients[1,])
plotExacerbations(results)

## End(Not run)
```

`plotHeatMap` *Creates heatmap of number of exacerbations*

Description

Creates heatmap of number of exacerbations

Usage

```
plotHeatMap(patientResults, n = 10, shortened = TRUE)
```

Arguments

`patientResults`	patient results vector, produced by accept.
`n`	how many exacerbations to consider
`shortened`	boolean

Value

a heatmap

Examples

```
## Not run:
results <- accept1(samplePatients[1,])
plotHeatMap(results)

## End(Not run)
```

`predict.accept_model` *Model predictions*

Description

Model predictions

Usage

```
## S3 method for class 'accept_model'
predict(object, ...)
```

Arguments

`object`	a model object
`...`	new data for which the prediction is needed

Value

prediction results from ACCEPT model

`predictCountProb` *Predicts probability of observing n exacerbations in the next year*

Description

Predicts probability of observing n exacerbations in the next year

Usage

```
predictCountProb(patientResults, n = 10, shortened = TRUE)
```

Arguments

`patientResults`	patient results vector, produced by accept.
`n`	how many exacerbations
`shortened`	boolean: Shortened results groups into 0, 1, 2, and 3 or more exacerbations

Value

a matrix of probabilities with the number of exacerbations as rows and number of severe exacerbations as columns

Examples

```
results <- accept2(samplePatients[1,])
predictCountProb (results)
```

`samplePatients` *Sample Patient Characteristics Inputs*

Description

A dataset containing sample patient characteristics to run the prediction model variables are as follows:

Format

A data frame with 2 rows and 19 variables

Details

- ID. A unique character string identifying a patients
- male. whether the patient is male
- age. the age of the patient (40–90)
- smoker. whether the patient is currently a smoker
- oxygen. whether the patient has had supplemental oxygen therapy within the past year (0,1)
- FEV1. forced expiratory volume in 1 second in percent predicted
- BMI. body mass index (10–60)
- SGRQ. St. George's Respiratory Questionnaire score (0–100)
- statin. whether the patient is taking statins due to cardiovascular conditions
- LAMA. whether the patient is on long acting muscarinic antagonist
- LABA. whether the patient is on long acting beta agonist
- ICS. whether the patient is on inhaled corticosteroids
- LastYrExacCount. total number of exacerbations in the previous year
- LastYrSevExacCount. number of severe exacerbations in the previous year

---

`set_openai_api_key` *Set OpenAI API key*

Description

NLP functionalities require a paid OpenAI API key to communicate with Open AI models.

Usage

```
set_openai_api_key(key, overwrite = FALSE, temporary = FALSE)
```

Arguments

`key`	an OpenAI API key. For more information on keys see the OpenAI API blog
`overwrite`	Option to overwrite any existing OpenAI keys already stored locally.
`temporary`	Option to not store thew API key for use across sessions.

Examples

```
## Not run:
set_openai_api_key("YOUR_OPENAI_API_KEY")

# This will set the key permanently until overwritten again
set_openai_api_key("YOUR_OPENAI_API_KEY")

## End(Not run)
```

`show_openai_api_key` *View saved openAI API key*

Description

View saved API key'

Usage

```
show_openai_api_key()
```

Examples

```
show_openai_api_key()
```

`vetiver_create_description.accept_model` *Create a vetiver model object description*

Description

Create a vetiver model object description

Usage

```
## S3 method for class 'accept_model'
vetiver_create_description(model)
```

Arguments

`model`	model name

Value

the description of the model

`vetiver_create_meta.accept_model` *Metadata constructors for ‘vetiver_model()‘ object*

Description

Useful for supporting accept_model class in vetiver

Usage

```
## S3 method for class 'accept_model'
vetiver_create_meta(model, metadata)
```

Arguments

`model`	a trained model object
`metadata`	list containing additional metadata to store with the pin

Value

The vetiver_create_meta function returns a ‘vetiver_meta()‘ list.

`vetiver_ptype.accept_model` *Create a vetiver input data prototype*

Description

Create a vetiver input data prototype

Usage

```
## S3 method for class 'accept_model'
vetiver_ptype(model, ...)
```

Arguments

`model`	a vetiver model object
`...`	any other inputs

Value

A ‘vetiver_ptype‘ method returns a zero-row dataframe, and ‘vetiver_create_ptype()‘ returns either such a zero-row dataframe, ‘NULL‘, or the dataframe passed to ‘save_prototype‘.

Index