

DEFINIÇÃO DA INTERFACE DO MÓDULO DE PREDIÇÃO DE GLUCOSE E PRESSÃO ARTERIAL

Listagem de métodos

* predict\_bp
* predict\_glucose
* validate\_bp\_data
* validate\_glucose\_data

predict\_bp

**predict\_bp(patient\_input, pred\_horizon)**

Retorna um um array de tamanho 3. Quando os parâmetros são inválidos o array atribui todos os valores -1. Quando os parâmetros se encontram válidos o array é representado na primeira posição pelo valor da previsão correspondente a 1 dia após o último valor de input, na segunda posição o valor da previsão correspondente a 3 dias após o último valor de input e na última posição o valor da previsão correspondente a 7 dias após o último valor de input.

**Parameters**

**patient\_input : array\_like**

Array com dados de input. 1 valor de pressão arterial sistólica por dia. 3 a 10 valores, que devem ser inteiros ou float.

**pred\_horizon: None or int**

O argumento do horizonte de previsão pode estar vazio (None) ou pode ser um número inteiro: 1, 3 ou 7 (correspondente a 1 dia, 3 dias ou 7 dias).

**Returns**

***result: array\_like***

Array de tamanho 3, contendo as previsões de 1, 3 e 7 dias, respetivamente. Os valores vêm preenchidos a -1 sempre que não for pedido esse horizonte ou em caso de pedido inválido.

***error\_code : int***

Inteiro indicador da validade dos inputs (ver tabela da função *validate\_bp\_data*)

**Notes:**

Dados introduzidos inválidos (verificado através da função validate\_data.py): result = [-1, -1, -1]

Dados introduzidos válidos: modelo de JNN previamente treinado efetua a previsão: independentemente do tamanho do array de entrada, result é um array de tamanho 3, sendo o primeiro valor a previsão correspondente ao primeiro dia após o input, o segundo valor a previsão correspondente ao terceiro dia após o input e o último valor a previsão correspondente ao sétimo dia após o input.

**Examples:**

>>> predict\_bp**([151, 98, '40'], None)**

[-1, -1, -1], 202

>>> result, error\_code **=** predict\_bp**([151, 98, 40], 3)**

[0, 100.02143, 0], 0

predict\_glucose

**predict\_glucose(patient\_input, pred\_horizon)**

Retorna um um array de tamanho 3, correspondente à previsão dos horizontes 2h, 4h ou 12h, respetivamente.

Quando os parâmetros são inválidos o array atribui todos os valores -1. Quando os parâmetros se encontram válidos o array é representado na primeira posição pelo valor da previsão correspondente a 2h após o último valor de input, na segunda posição o valor da previsão correspondente a 4h após o último valor de input e na última posição o valor da previsão correspondente a 12h após o último valor de input.

**Parameters**

**patient\_input : array\_like**

Array com dados de input. Assume-se que os valores de glucose são adquiridos com uma frequência de 5min (isto é, 1h de medições corresponde a 12 valores de glucose).

**pred\_horizon: None or int**

O argumento do horizonte de previsão pode estar vazio (None) ou pode ser um número inteiro: 2, 4 ou 12 (correspondente a 2h, 4h ou 12h). Se None, as 3 previsões são devolvidas.

**Returns**

***result: array\_like***

Array de tamanho 3, correspondente à previsão dos horizontes 2h, 4h ou 12h, respetivamente

***error\_code : int***

Inteiro indicador da validade dos inputs (ver tabela da função *validate\_bp\_data*)

**Notes**

Se o pred\_horizon = 2, 4 ou 12:

|  |  |  |
| --- | --- | --- |
| Previsão desejada | Inputs possíveis com crescente nível de confiança | Input Ideal (nível de confiança mais elevado) |
| 2h | 2h (24 valores)  4h (48 valores)  6h (72 valores)  8h (96 valores) | 12h (144 valores) |
| 4h | 4h (48 valores)  8h (96 valores) | 12h (144 valores) |
| 12h | 12h(144 valores) | 24h (288 valores) |

Se o pred\_horizon = None:

|  |  |
| --- | --- |
| Input enviado | Output obtido |
| 2h (24 valores) <= len(patient\_data) < 4h (48 valores) | Previsão 2h |
| 4h (48 valores) <= len( patient\_data) <  12h (144 valores) | Previsão de 2h e 4h |
| len( patient\_data) >= 12h (144 valores) | Previsão de 2h, 4h e 12h |

Não há nenhum problema em serem enviados dados superiores ao input ideal.

**Examples:**

Dados introduzidos inválidos (verificado através da função validate\_data.py): result = [-1, -1, -1]

Dados introduzidos válidos: : independentemente do tamanho do array de entrada, result é um array de tamanho 3. Este é representado na primeira posição pelo valor da previsão correspondente a 2h após o último valor de input, na segunda posição o valor da previsão correspondente a 4h após o último valor de input e na última posição o valor da previsão correspondente a 12h após o último valor de input.

>>> result, error\_code **=** predict\_glucose**(patient\_input, None)**

[208.0, 169.0, 173.0], 0

>>> predict\_glucose**([170, 95, '55'], None)**

[-1, -1, -1], 202

>>> predict\_glucose**(patient\_input, 4)**

[0, 124.0, 0], 0

No primeiro exemplo válido assume-se que len(patient\_input) >= 24.

No segundo exemplo válido assume-se que len(patient\_input) >= 48.

validate\_bp\_data

**validate\_bp\_data(*patient\_input, pred\_horizon*)**

Avalia a validade dos inputs da função *predict\_bp*, tanto o array de dados como a indicação do horizonte de previsão são avaliados. Valores fora da gama admissível são balizados e missing values são resolvidos, quando possível.

Em casos de demasiados missing values, horizontes de predição inválidos, ou outros erros descritos na tabela, a função retorna o valor de ***error\_code*** consoante o código de erro que discrimina a situação**.**

**Parameters**

**patient\_input: *array\_like***

Array com dados de input. 1 valor de pressão arterial sistólica por dia. 3 a 10 valores, que devem ser inteiros ou float.

**pred\_horizon :** **int or None**

O argumento do horizonte de previsão pode estar vazio (None) ou pode ser um número inteiro: 1, 3 ou 7 (correspondente a 1 dia, 3 dias ou 7 dias).

**Returns**

***error\_code : int***

Inteiro indicador da validade dos inputs (ver tabela)

***patient\_input: array\_like***

Array de entrada com as alterações efetuadas (caso aplicável)

**Notes**

O input é válido se cumprir todas as seguintes condições (tabela):

* pred\_horizon -> é None, 1, 3 ou 7
* patient\_input -> todos os valores do input são inteiros ou float; não contém NaN; todos os valores são inteiros ou float entre 50 e 200 (se existirem valores inferiores a 50 ou superiores a 200 estes são igualados a 50 e 200, respetivamente); tem tamanho entre 3 e 10, inclusive.

|  |  |  |
| --- | --- | --- |
| Parâmetro | *error\_code* | Significado |
| - | 0 | Todos os dados introduzidos estão válidos. |
| *pred\_horizon* | *101* | *pred\_horizon* não corresponde a uma das opções válidas (*None*, 1, 3 ou 7). |
| *patient\_input* | *201* | Os dados de entrada contêm *NaN.* |
| 202 | Tipos de valor inválidos nos dados de entrada (válidos: int e float). |
| 203 | Tamanho do input inferior a 3 ou superior a 10 |

**Examples**

>>> validate\_bp\_data**([151, 512, '40'], None)**

202, [151, 512, '40']

>>> validate\_bp\_data**([151, 512, 40], None)**

0, [151, 200, 50]

validate\_glucose\_data

**validate\_glucose\_data(*patient\_input, pred\_horizon*)**

Avalia a validade dos inputs da função *predict\_glucose.* Tanto o array de dados como a indicação do horizonte de previsão são avaliados.

**Parameters**

**patient\_input: *array\_like***

Array com dados de input. Assume-se que os valores de glucose são adquiridos com uma frequência de 5min (isto é, 1h de medições corresponde a 12 valores de glucose).

**pred\_horizon :** **int or None**

O argumento do horizonte de previsão pode estar vazio (None) ou pode ser um número inteiro: 2, 4 ou 12 (correspondente a 2h, 4h ou 12h).

**Returns**

***error\_code : int***

Inteiro indicador da validade dos inputs (ver tabela)

***patient\_input : array\_like***

Array de entrada com as alterações efetuadas (caso aplicável)

**Notes**

O input é válido se cumprir todas as seguintes condições (tabela):

* pred\_horizon -> é None, 2, 4 ou 12
* patient\_input -> todos os valores do input são inteiros ou float; não contém NaN; todos os valores são inteiros ou float entre 20 e 800 (se existirem valores inferiores a 20 ou superiores a 800 estes são igualados a 20 e 800, respetivamente); para pred\_horizon = None ou 2 são necessários no mínimo 24 valores (2h); para pred\_horizon = 4 são necessários no mínimo 48 valores (4h); para pred\_horizon = 12 são necessários no mínimo

144 valores (12h);

|  |  |  |
| --- | --- | --- |
| Parâmetro | *error\_code* | Significado |
| - | 0 | Todos os dados introduzidos estão válidos. |
| *pred\_horizon* | *101* | *pred\_horizon* não corresponde a uma das opções válidas (*None*, 2, 4 ou 12). |
| *patient\_input* | *201* | Os dados de entrada contêm *NaN.* |
| 202 | Tipos de valor inválidos nos dados de entrada (válidos: int e float). |
| 203 | Para argumento de horizonte de previsão vazio (*None*) ou uma previsão de 2h, a entrada deve conter pelo menos 24 valores (2 horas). |
| 204 | Para uma previsão de 4h, a entrada deve conter pelo menos 48 valores (4 horas). |
| 205 | *﻿* *Para uma previsão de 12h, a entrada deve conter pelo menos 144 valores (12 horas).* |

**Examples**

>>> validate\_glucose\_data**([170, 95, '55'], None)**

202, [170, 95, '55']

>>> validate\_glucose\_data**(patient\_input, None)**

0, patient\_input

No segundo exemplo (válido) assume-se que len(patient\_input) >= 144.