

---

# **project**<sub>*n*</sub>*ameDocumentation*

***Release 0.1***

**notebooks-for-docs-and-tests-demo** Luiz Felipe Santiago rodrigu

**Jun 03, 2021**



# TUTORIALS

<b>1</b>	<b>Tutorial</b>	<b>3</b>
1.1	Initial steps . . . . .	3
1.2	Finding the answer . . . . .	3
1.3	Finding the question . . . . .	3
1.3.1	For those in a hurry . . . . .	4
1.3.2	For those with time . . . . .	4
<b>2</b>	<b>src package</b>	<b>5</b>
2.1	Subpackages . . . . .	5
2.1.1	src.data package . . . . .	5
2.1.1.1	Submodules . . . . .	5
2.1.1.2	src.data.make_dataset module . . . . .	5
2.1.2	src.features package . . . . .	5
2.1.2.1	Submodules . . . . .	5
2.1.2.2	src.features.build_features module . . . . .	5
2.1.2.3	src.features.computer module . . . . .	5
2.1.2.4	src.features.thingy module . . . . .	6
2.1.3	src.models package . . . . .	6
2.1.3.1	Submodules . . . . .	6
2.1.3.2	src.models.predict_model module . . . . .	6
2.1.3.3	src.models.train_model module . . . . .	6
2.1.4	src.visualization package . . . . .	6
2.1.4.1	Submodules . . . . .	6
2.1.4.2	src.visualization.visualize module . . . . .	6
<b>3</b>	<b>Indices and tables</b>	<b>7</b>
	<b>Python Module Index</b>	<b>9</b>
	<b>Index</b>	<b>11</b>



Welcome to the docs. I hope these are informative enough.



## TUTORIAL

This demonstrates how to use the `DeepThought` class to deal with your personal philosophical issues.

### 1.1 Initial steps

First, let us import the what we need.

```
[1]: import sys
     sys.path.append('../src')

     from features import computer
```

We will be using an instance of the special computer named `DeepThought`

```
[2]: thinker = computer.DeepThought()
```

### 1.2 Finding the answer

The answer for “Ultimate Question of Life, the Universe, and Everything” can be easily computed.

```
[3]: thinker.get_the_answer()
[3]: 42
```

### 1.3 Finding the question

The frustrating thing about finding the answer to “Ultimate Question of Life, the Universe, and Everything” is that “the Question” is actually a harder problem.

### 1.3.1 For those in a hurry

If you don't too much time in your hands, you can get away with "a question" (which may or may not be the right one). This can be done using the following method:

```
[4]: thinker.get_a_question()
[4]: 'What do you get if you multiply six by nine?'
```

Note that this may change if re-executed... *therefore, we added the tag: ``nbval-ignore-output`` to it*

```
[5]: thinker.get_a_question()
[5]: 'How many roads must a man walk down?'
```

### 1.3.2 For those with time

You may want to execute the following command to get a definite answer...

```
[6]: thinker.get_the_question()

This may take some time...
...

-----
KeyboardInterrupt                                Traceback (most recent call last)
<ipython-input-6-8fb74e69032b> in <module>
----> 1 thinker.get_the_question()

~/notebooks-for-docs-and-tests-demo/src/features/computer.py in get_the_question(self)
     52         print('This may take some time...')
     53         print('...')
--> 54         time.sleep(10)
     55
     56         return thingy.something

KeyboardInterrupt:
```

The above execution was interrupted due to the demo's time constraints.

*To to prevent it from entering the validation, we added the tag ``nbval-skip`` to it.*



## SRC PACKAGE

## 2.1 Subpackages

### 2.1.1 src.data package

#### 2.1.1.1 Submodules

#### 2.1.1.2 src.data.make\_dataset module

### 2.1.2 src.features package

#### 2.1.2.1 Submodules

#### 2.1.2.2 src.features.build\_features module

#### 2.1.2.3 src.features.computer module

**class** src.features.computer.DeepThought(*questions=None*)

Bases: object

Designed to allow better examining the “Ultimate Question of Life, the Universe, and Everything” [1]

**Parameters** *questions* (*List*) – Optional list of potential “Ultimate Questions..”. If absent, common sense will be used.

**get\_a\_question()**

Finds a (possible) “Ultimate Question of Life, the Universe, and Everything”

---

**Note:** This may or may not be satisfying... but it is a bit better than waiting.

---

**Returns** *question* – A... question!

**Return type** str

**get\_the\_answer()**

Finds the answer for the “Ultimate Question of Life, the Universe, and Everything”!

**Returns** *question* – The Answer!

**Return type** str

**get\_the\_question()**

Finds the “Ultimate Question of Life, the Universe, and Everything”!

**Returns** `question` – The Question!

**Return type** `str`

**2.1.2.4 src.features.thingy module**

Something may be happening here... but I will not tell you

**2.1.3 src.models package**

**2.1.3.1 Submodules**

**2.1.3.2 src.models.predict\_model module**

**2.1.3.3 src.models.train\_model module**

**2.1.4 src.visualization package**

**2.1.4.1 Submodules**

**2.1.4.2 src.visualization.visualize module**

## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## PYTHON MODULE INDEX

### S

- `src`, 5
- `src.data`, 5
- `src.data.make_dataset`, 5
- `src.features`, 5
- `src.features.build_features`, 5
- `src.features.computer`, 5
- `src.features.thingy`, 6
- `src.models`, 6
- `src.models.predict_model`, 6
- `src.models.train_model`, 6
- `src.visualization`, 6
- `src.visualization.visualize`, 6



## INDEX

### D

DeepThought (*class in src.features.computer*), 5

### G

get\_a\_question() (*src.features.computer.DeepThought*  
*method*), 5

get\_the\_answer() (*src.features.computer.DeepThought*  
*method*), 5

get\_the\_question() (*src.features.computer.DeepThought*  
*method*), 5

module, 6  
src.models.predict\_model  
module, 6  
src.models.train\_model  
module, 6  
src.visualization  
module, 6  
src.visualization.visualize  
module, 6

### M

module

src, 5  
src.data, 5  
src.data.make\_dataset, 5  
src.features, 5  
src.features.build\_features, 5  
src.features.computer, 5  
src.features.thingy, 6  
src.models, 6  
src.models.predict\_model, 6  
src.models.train\_model, 6  
src.visualization, 6  
src.visualization.visualize, 6

### S

src

module, 5

src.data

module, 5

src.data.make\_dataset

module, 5

src.features

module, 5

src.features.build\_features

module, 5

src.features.computer

module, 5

src.features.thingy

module, 6

src.models