



Deep Learning com Python para visão computacional

13/08/2018 das 14 às 17h

Tutorial Básico sobre Ferramentas Básicas

Jupyter Notebook

Fonte:

<http://jupyter-notebook.readthedocs.io/en/stable/notebook.html>

<http://webserver2.tecgraf.puc-rio.br/~mgattass/fcg/T1PythonInstall.pdf>

<https://www.dataquest.io/blog/jupyter-notebook-tutorial/>

<https://www.concrete.com.br/2017/07/21/introducao-ao-python-com-jupyter/>

<https://ipython.org/ipython-doc/3/notebook/nbformat.html>

O Jupyter Notebook é um aplicativo da Web de código aberto que permite criar e compartilhar documentos que contêm código ativo, equações, visualizações e texto narrativo. Os usos incluem: limpeza e transformação de dados, simulação numérica, modelagem estatística, visualização de dados, aprendizado de máquina e muito mais.

O Jupyter Notebook pode ser usado abrindo um terminal, acessando o Container criado no Docker e digitando Jupyter Notebook no mesmo. Outra opção é acessar o browser e digitar a URL.

Exemplo utilizado pelo terminal:

```
root@ivision-C2108-GP5-GE:/opt/Mask_RCNN# jupyter notebook --ip 192.168.212.201 --allow-root --port 6379
```

ou

<http://192.168.212.201:6380/tree#notebooks>

O Jupyter inicia um servidor Python local para servir esses aplicativos ao seu navegador, tornando-o essencialmente independente de plataforma e abrindo as portas para um compartilhamento mais fácil na web.

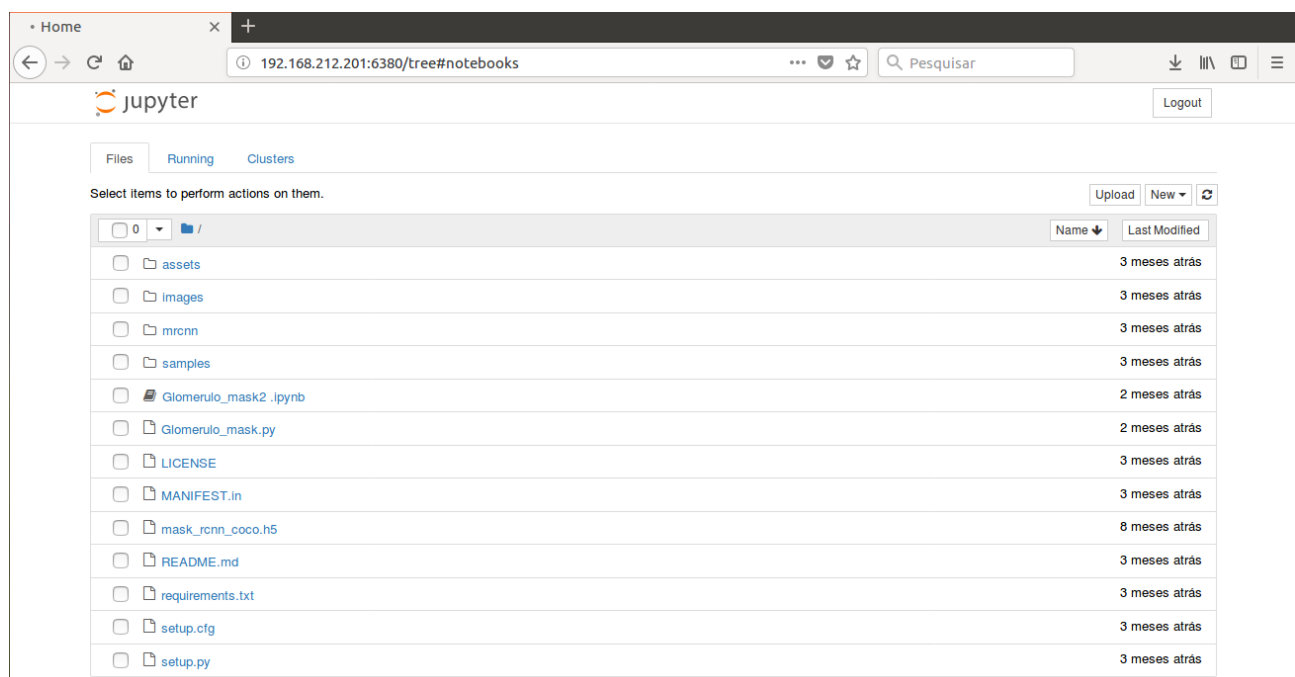
```
root@lvision-C2108-GP5-GE: /opt/Mask_RCNN
7b14d64e5ee5    ivision/mask-rcnn    "/bin/bash"    6 weeks ago    Up 6 weeks
5bc92b895c44    ivision/mask-rcnn    "/bin/bash"    2 months ago    Up 2 months
92a026e08ecb    ikaro-ssh            "/usr/sbin/sshd -D"    2 months ago    Up 2 months    0.0.0.0:
32776->22/tcp, 0.0.0.0:32775->6006/tcp, 0.0.0.0:32774->8888/tcp    ikaro_ssh_1
luizotavio@lvision-C2108-GP5-GE:~$ docker exec -it luiz_glomerulo
"docker exec" requires at least 2 arguments.
See 'docker exec --help'.

Usage: docker exec [OPTIONS] CONTAINER COMMAND [ARG...] [flags]

Run a command in a running container
luizotavio@lvision-C2108-GP5-GE:~$ docker attach luiz_glomerulo
^C^Z
[1]+  Stopped                  jupyter notebook --ip 0.0.0.0 --allow-root --port 6379
root@lvision-C2108-GP5-GE:/opt/Mask_RCNN# jupyter notebook --ip 0.0.0.0 --allow-root --port 6379
[I 14:06:30.628 NotebookApp] The port 6379 is already in use, trying another port.
[I 14:06:30.641 NotebookApp] Serving notebooks from local directory: /opt/Mask_RCNN
[I 14:06:30.642 NotebookApp] 0 active kernels
[I 14:06:30.642 NotebookApp] The Jupyter Notebook is running at:
[I 14:06:30.642 NotebookApp] http://0.0.0.0:6380/?token=1f4bf42e0d1f6c226b8fa1e7c3ada2e0dd9a3cdd63ac2c57
[I 14:06:30.642 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 14:06:30.642 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
    http://0.0.0.0:6380/?token=1f4bf42e0d1f6c226b8fa1e7c3ada2e0dd9a3cdd63ac2c57
[I 14:07:24.163 NotebookApp] 302 GET /?token=1f4bf42e0d1f6c226b8fa1e7c3ada2e0dd9a3cdd63ac2c57 (192.168.165.57) 0.59ms
```

O Jupyter é executado no terminal com exibição do token para acesso, nesse caso basta copiá-lo e inseri-lo no browser.



Running Jupyter

Existem 3 abas para interação: files, running e clusters.

Select items to perform actions on them.

Upload New ↺

0 ▾	/	Name ▾	Last Modified
<input type="checkbox"/>	assets		3 meses atrás
<input type="checkbox"/>	images		3 meses atrás
<input type="checkbox"/>	mrcnn		em 3 horas
<input type="checkbox"/>	samples		3 meses atrás
<input type="checkbox"/>	Glomerulo_mask2.ipynb		Running em 3 horas
<input type="checkbox"/>	Glomerulo_mask.py		2 meses atrás
<input type="checkbox"/>	LICENSE		3 meses atrás
<input type="checkbox"/>	MANIFEST.in		3 meses atrás
<input type="checkbox"/>	mask_rcnn_coco.h5		8 meses atrás
<input type="checkbox"/>	README.md		3 meses atrás
<input type="checkbox"/>	requirements.txt		3 meses atrás
<input type="checkbox"/>	setup.cfg		3 meses atrás
<input type="checkbox"/>	setup.py		3 meses atrás

Currently running Jupyter processes

↺

Terminals ▾

There are no terminals running.

Notebooks ▾

Glomerulo_mask2.ipynb

Python 3

Shutdown

segundos atrás

Clusters tab is now provided by IPython parallel. See ['IPython parallel'](#) for installation details.

O que é um arquivo ipynb?

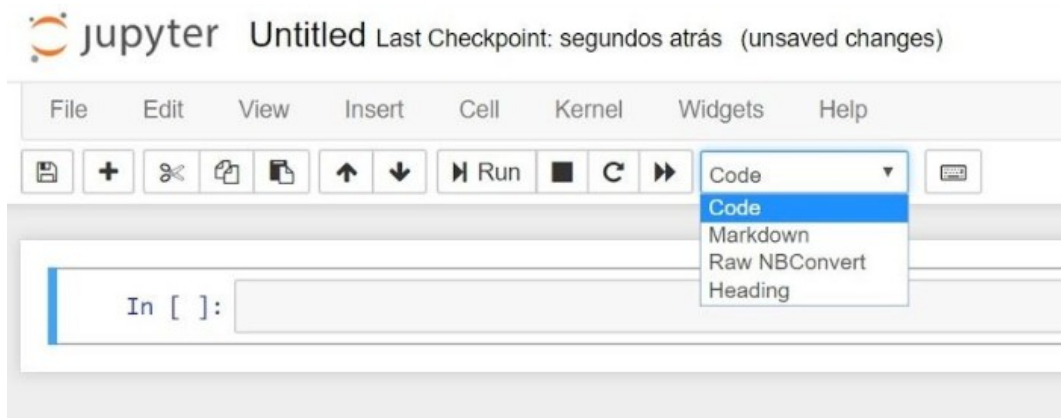
É um arquivo de texto que descreve o conteúdo do seu notebook em um formato chamado JavaScript Object Notation (JSON). Cada célula e seu conteúdo, incluindo anexos de imagem que foram convertidos em strings de texto, são listados junto com alguns metadados. Pode ser manipulado selecionando a opção de Editar.

The notebook interface

Existem dois termos usados no Jupyter: kernel e célula.

Um **kernel** é um "mecanismo computacional" que executa o código contido em um documento do notebook.

Uma **célula** é um recipiente para o texto a ser exibido no bloco de notas ou código a ser executado pelo kernel do notebook.



Code: vem por padrão quando é selecionado a opção de Python

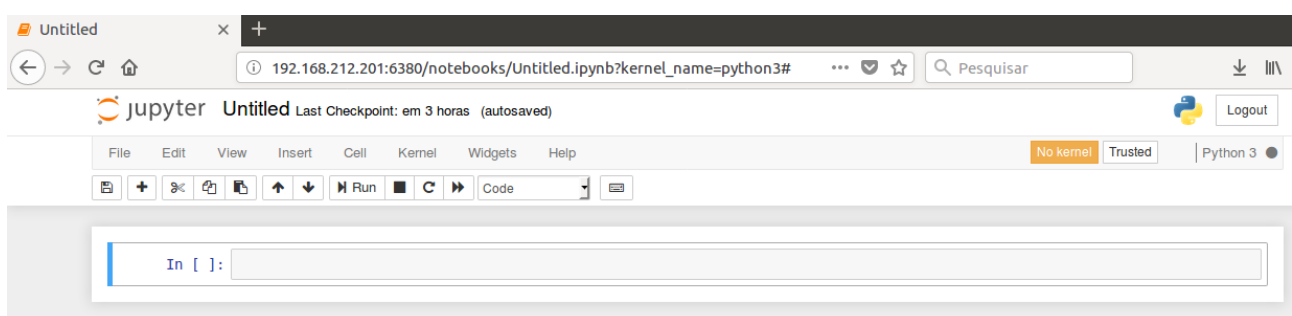
Markdown: é uma linguagem simples de marcação, que converte seu texto em HTML válido.

Raw NBConvert: é armazenada nos metadados do notebook. Por padrão (se nenhum formato de célula estiver selecionado), o conteúdo da célula é incluído (sem qualquer conversão) nas saídas HTML e LaTeX.

Heading:

Você pode adicionar 'headings' iniciando uma linha com um (ou vários) seguido de # por linha, como no exemplo a seguir:

```
# Heading 1
# Heading 2
## Heading 2.1
## Heading 2.2
```



Segue algumas bibliotecas na linguagem Python:

- **Python** versão ≥ 3.5 ;
- **Numpy**, o núcleo de extensões numéricas para álgebra linear e arrays multidimensionais;
- **Matplotlib**, destaque no traçado e bibliotecas de gráficos;
- **Pandas** é uma biblioteca de código aberto (BSD-licensed), fácil de usar na análise e estruturas de dados.

- **Scikit-learn** é uma ferramenta eficiente para aprendizado de máquina e análise de dados em Python.
- **Scikit-image** é uma biblioteca de processamento de imagens de código aberto para a linguagem de programação Python.

Python

Fonte:

https://pt.wikibooks.org/wiki/Python/Conceitos_b%C3%A1sicos/Tipos_e_operadores
<https://docs.python.org/3/tutorial/introduction.html#strings>
<https://www.concrete.com.br/2017/07/21/introducao-ao-python-com-jupyter/>
<https://www.w3schools.com/python/default.asp>
https://www.tutorialspoint.com/python/python_loops.html
<http://excript.com/python/indentacao-python.html>

Clicando em **New > Python** somos apresentados à tela de edição de scripts na qual podemos digitar nosso código, que é interpretado e exibido ao pressionar **CTRL + Enter**.

Existem 4 tipos numéricos:

- inteiro (int)
- ponto flutuante (float)
- booleano (bool)
- complexo (complex)

Suportam adição, subtração, multiplicação e divisão e também podem se relacionar.

Mesmo os tipos não sendo declarados explicitamente, eles sempre irão assumir um tipo de dado, abaixo, exemplos de retornos de tipos:

Tipo inteiro:

```
>>> a = 1
>>> type(a)
<type 'int'>
```

Tipo ponto flutuante:

```
>>> a = 1.0
>>> type(a)
<type 'float'>
```

Tipo booleano:

```
>>> a = True
>>> type(a)
```

```
<type 'bool'>
```

Tipo complexo:

```
>>> a = 4+3j
>>> type(a)
<type 'complex'>
```

OPERADORES

Operadores são símbolos que atuam sobre variáveis e valores. Existem:

- Operadores aritméticos (+, -, *, /, %, **, +=, -=, *=, /=, %=, **=):

```
>>> a = 1
>>> b = 2
>>> print a + b
3
```

ou

```
>>> a = 5
>>> print a**2
25
```

- Operadores de comparação (>, <, ==, >=, <=, !=):

```
>>> a = 1
>>> b = 2
>>> a == b
False
>>> print a > b
False
```

- Operadores lógicos (and, or, not):

```
>>> nome = "leonardo"
>>> idade = 22
>>> nome == "leonardo" and idade < 23
True
```

- Operadores de atribuição (=):

```
>>> a = 1
```

Indetação

Fonte:

<http://excript.com/python/indentacao-python.html>

Indentar é o recuo do texto em relação a sua margem, ou seja, se antes de escrevermos uma instrução, utilizamos 4 espaçamentos da margem esquerda até a instrução propriamente dita, podemos dizer que a indentação utilizada possui 4 espaços.

Em Python, a indentação possui função bastante especial, até porque, os blocos de instrução são delimitados pela profundidade da indentação, isto é, os códigos que estiverem rente à margem esquerda, farão parte do primeiro nível hierárquico. Já, os códigos que estiverem a 4 espaços da margem esquerda, estarão no segundo nível hierárquico e aqueles que estiverem a 8 espaços, estarão no terceiro nível e assim por diante.

Todos os blocos são delimitados pela profundidade da indentação e por isso, a sua importância, é vital para um programa em Python. O mau uso, isto é, utilizar 4 espaçamentos enquanto deveríamos estar utilizando 8, acarretará na não execução, ou então, no mal funcionamento em geral.

BLOCOS

Bloco de código é uma ou um conjunto de instrução que estejam numa mesma distância da margem esquerda.

A recomendação é que utilizemos, ou 1 tabulação, ou então, 4 espaços.

Ao invés de trabalharmos com quantidades de espaços, podemos utilizar uma determinada quantidade de tabulações. O primeiro nível hierárquico seria o nível 0, isto é, instruções que não possuem tabulações à sua esquerda. O segundo nível utilizaria uma única tabulação, o terceiro nível utilizaria 2 tabulações e assim sucessivamente.

O mais importante é sabermos que em Python, o espaçamento não é facultativo, isto é, não o colocamos se assim desejarmos, mas sim, somos obrigados a trabalharmos com algum sistema de espaçamento a fim de definirmos blocos isolados de códigos.

Estruturas de condição e repetição (IF Else For While)

Fonte:

<https://www.devmedia.com.br/estruturas-de-condicao-e-repeticao-em-python/37158>

<http://excript.com/python/while-else-python.html>

IF

```
1 idade = 18
2 if idade < 20:
3     print('Você é jovem!')
```

IF-ELSE

```
1 idade = 18
2 if idade >= 18:
3     print('maior de idade')
4 else:
5     print('menor de idade')
```

IF-ELIF-ELSE

```
1 idade = int(input('Digite sua idade: '))
2 if idade >= 10 and idade < 20:
3     print('Você é adoslecente')
4 elif idade >= 20 and idade < 30:
5     print('Você é jovem')
6 elif idade >= 30 and idade <= 100:
7     print('Você é adulto')
8 else:
9     print('Valor não encontrado!')
```

For

```
1 nomes = ['Pedro', 'João', 'Leticia']
2 for n in nomes:
3     print(n)
4 >>>
5 Pedro
6 João
7 Leticia
```

While

```
01 contador = 0
02 while contador < 5:
03     print(contador)
04     contador = contador + 1
```

Function

Fonte:

https://www.w3schools.com/python/python_functions.asp

Uma função é um bloco de código que só é executado quando é chamado. Pode passar dados, conhecidos como parâmetros, para uma função. Uma função pode retornar dados como resultado.

Em Python, uma função é definida usando a palavra-chave def:

```
def my_function(country = "Norway"):
    print("I am from " + country)

my_function("Sweden")
my_function("India")
my_function()
my_function("Brazil")
```

```
def my_function(x):  
    return 5 * x  
  
print(my_function(3))  
print(my_function(5))  
print(my_function(9))
```

Python Collections (Arrays)

Fonte:

https://www.w3schools.com/python/python_tuples.asp

Lista e Tupla

List é uma coleção ordenada e modificável.

Tuple é uma coleção que é ordenada e permanente.

List

```
thislist = ["apple", "banana", "cherry"]  
print(thislist)
```

```
thislist = ["apple", "banana", "cherry"]  
thislist[1] = "blackcurrant"  
print(thislist)
```

The list() Constructor

Também é possível usar o construtor list () para criar uma lista. Para **adicionar** um item à lista, use o método do objeto **append ()**. Para **remover** um item específico, use o método do objeto **remove ()**. A **função len ()** retorna o **tamanho da lista**.

Using the append() method to append an item:

```
thislist = list(("apple", "banana", "cherry"))  
thislist.append("damson")  
print(thislist)
```

Using the remove() method to remove an item:

```
thislist = list(("apple", "banana", "cherry"))  
thislist.remove("banana")  
print(thislist)
```

Run example »

Example

The len() method returns the number of items in a list:

```
thislist = list(("apple", "banana", "cherry"))  
print(len(thislist))
```

Tuplas

Example

Return the item in position 1:

```
thistuple = ("apple", "banana", "cherry")  
print(thistuple[1])
```

Run example »

Example

You cannot change values in a tuple:

```
thistuple = ("apple", "banana", "cherry")  
thistuple[1] = "blackcurrant" # test changeability  
print(thistuple)
```

Também é possível usar o construtor tuple () para fazer uma tupla. A função len () retorna o comprimento da tupla.

Example

The len() method returns the number of items in a tuple:

```
thistuple = tuple(("apple", "banana", "cherry"))  
print(len(thistuple))
```

Class/Object

Almost everything in Python is an object, with its properties and methods.

```
class MyClass:  
    x = 5
```

```
p1 = MyClass()  
print(p1.x)
```

Numpy

Fonte:

<http://cs231n.github.io/python-numpy-tutorial/#numpy>

<http://pyscience-brasil.wikidot.com/module:numpy>

O **NumPy** é o pacote básico da linguagem **Python** que permite trabalhar com arranjos, vetores e matrizes de N dimensões, de uma forma comparável e com uma sintaxe semelhante ao software proprietário **Matlab**, mas com muito mais eficiência, e com toda a expressividade da linguagem. Provê diversas funções e operações sofisticadas, incluindo (mas não se limitando a):

- Objeto array para a implementação de arranjos multidimensionais
- Objeto matrix para o cálculo com matrizes
- Ferramentas para álgebra linear
- Transformadas de Fourier básicas
- Ferramentas sofisticadas para geração de números aleatórios

Array

Int16 -> shortint: deve possuir tamanho de no mínimo 16 bits e não pode ser maior que int;

Int32 -> int: deve possuir tamanho mínimo de 32 bits;

Int64 -> longint: deve possuir tamanho mínimo de 64 bits.

Matplotlib**Fonte:**

<https://matplotlib.org/>

<https://matplotlib.org/tutorials/index.html>

Pandas**Fonte:**

http://minerandodados.com.br/index.php/2017/09/26/python-para-analise-de-dados/#Scripts_Dataset

<https://pythonprogramming.net/data-analysis-python-pandas-tutorial-introduction/>

<https://www.youtube.com/watch?v=-NR-ynQg0YM>

Pandas é uma biblioteca de código aberto (BSD-licensed), usado na análise e estruturas de dados

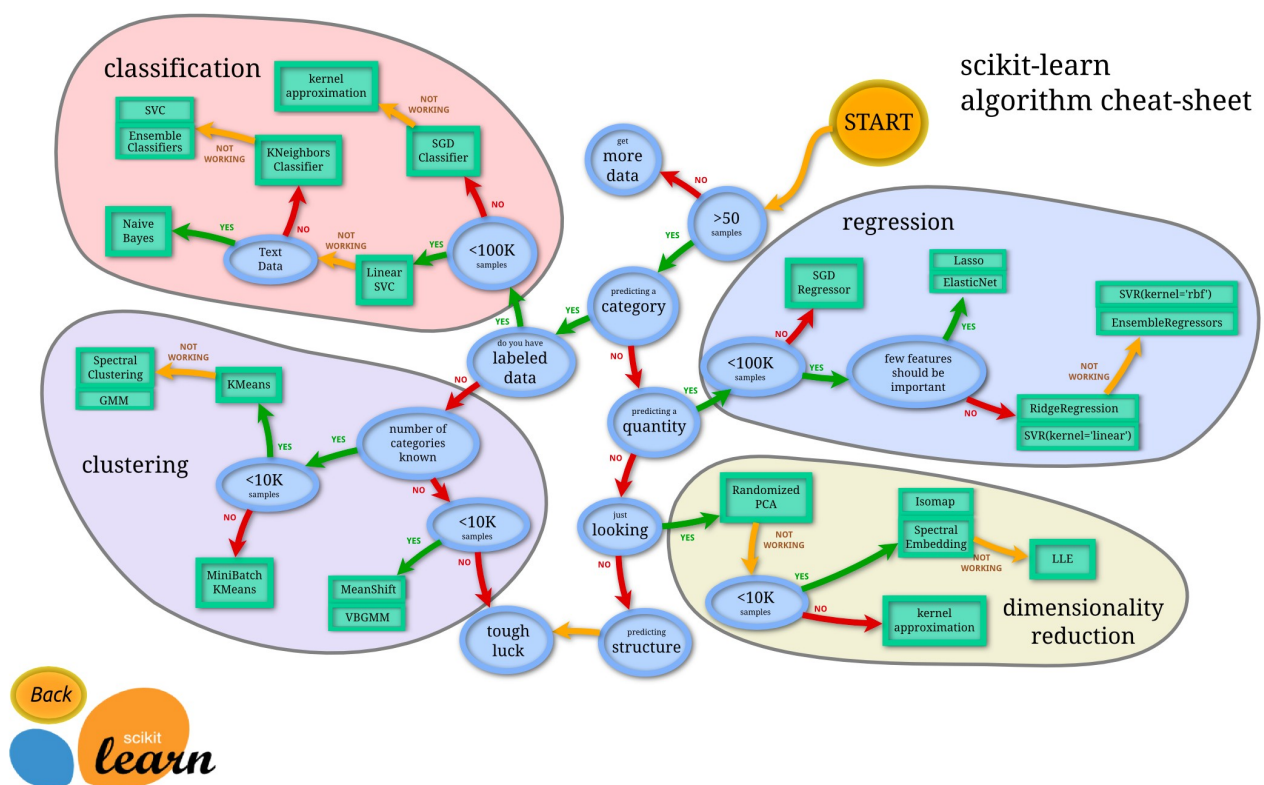
Scikit learn

Fonte:

http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html#sphx-glr-auto-examples-classification-plot-classifier-comparison-py

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Open source



Scikit – image

Fonte:

http://scikit-image.org/docs/dev/auto_examples/

<https://docplayer.com.br/58452239-Visao-computacional-em-python-utilizando-as-bibliotecas-scikit-image-e-scikit-learn.html>

<https://www.openhub.net/p/scikit-image/contributors>

Image processing in Python

Scikit-image is a collection of algorithms for image processing. It is available [free of charge and free of restriction](#). We pride ourselves on high-quality, peer-reviewed code, written by an active [community of volunteers](#).