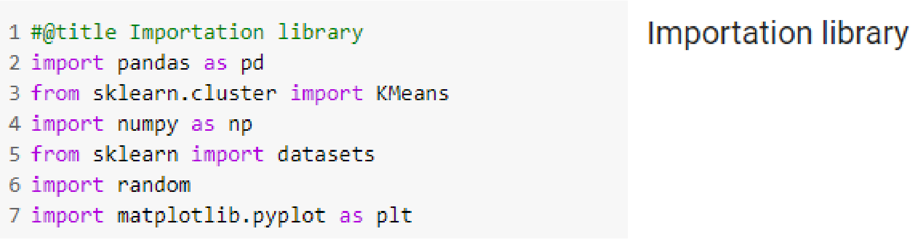
Master Informatique EID2

Deep Learning

TP 3 – Cartes Auto-Organisatrices

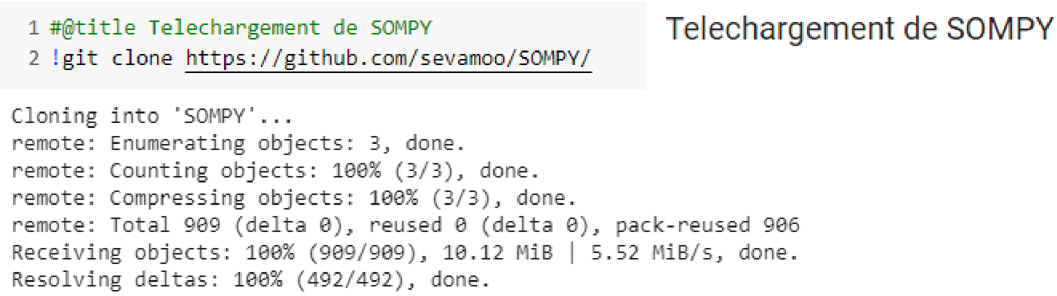
Mohamed Ben Saad
Elias ABDELLI



```python
#@title Importation library
import pandas as pd
from sklearn.cluster import KMeans
import numpy as np
from sklearn import datasets
import random
import matplotlib.pyplot as plt
```

Importation library

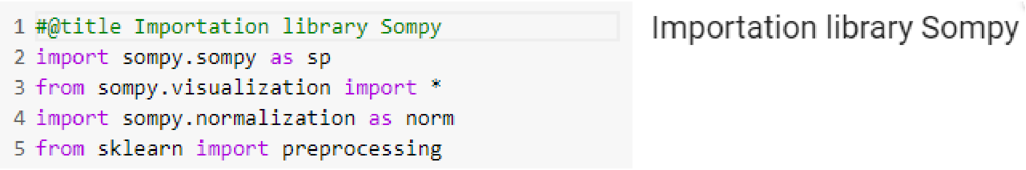
Som n’existe pas en python, il a fallu trouver une library proche de celle de Matlab



```
#@title Telechargement de SOMPY
!git clone https://github.com/sevamoo/SOMPY/
```

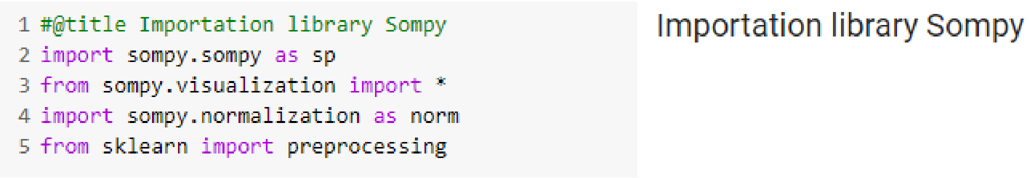
Telechargement de SOMPY

```
Cloning into 'SOMPY'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 909 (delta 0), reused 0 (delta 0), pack-reused 906
Receiving objects: 100% (909/909), 10.12 MiB | 5.52 MiB/s, done.
Resolving deltas: 100% (492/492), done.
```



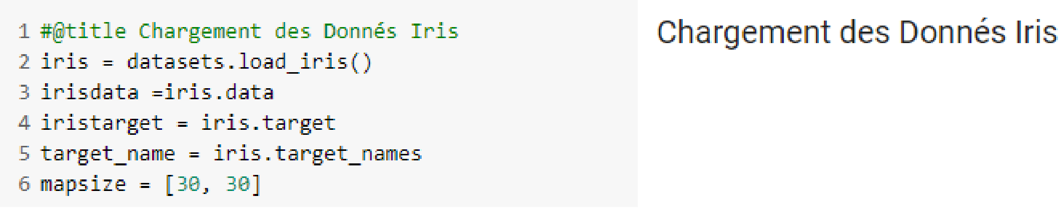
```python
#@title Importation library Sompy
import sompy.sompy as sp
from sompy.visualization import *
import sompy.normalization as norm
from sklearn import preprocessing
```

Importation library Sompy



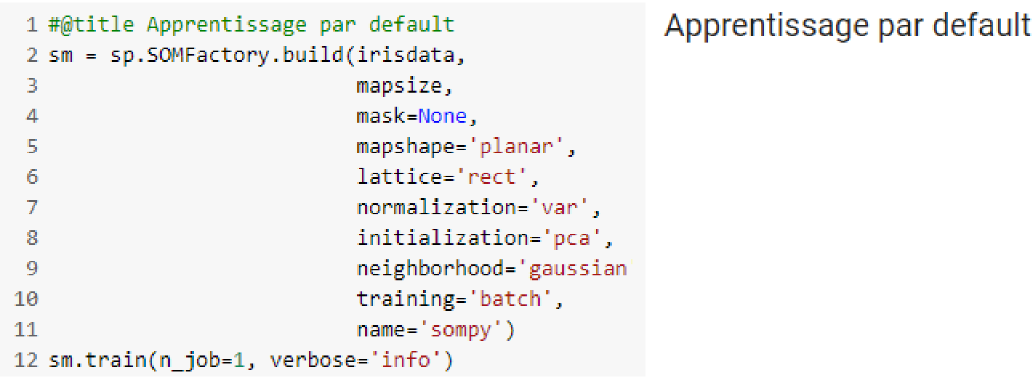
```python
#@title Importation library Sompy
import sompy.sompy as sp
from sompy.visualization import *
import sompy.normalization as norm
from sklearn import preprocessing
```

Importation library Sompy



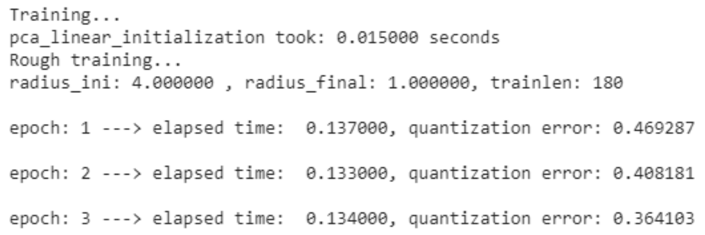
```python
#@title Chargement des Donnés Iris
iris = datasets.load_iris()
irisdata =iris.data
iristarget = iris.target
target_name = iris.target_names
mapsize = [30, 30]
```

Chargement des Donnés Iris



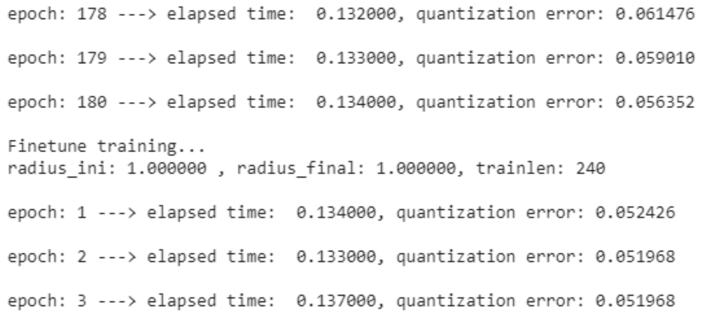
```python
#@title Apprentissage par default
sm = sp.SOMFactory.build(irisdata,
                         mapsize,
                         mask=None,
                         mapshape='planar',
                         lattice='rect',
                         normalization='var',
                         initialization='pca',
                         neighborhood='gaussian
                         training='batch',
                         name='sompy')
sm.train(n_job=1, verbose='info')
```

Apprentissage par default



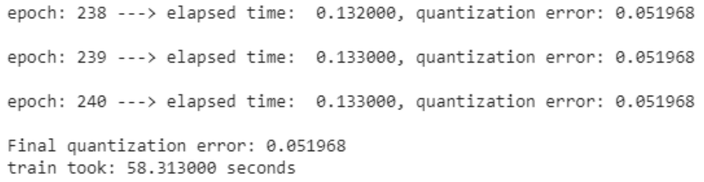
```
Training...
pca_linear_initialization took: 0.015000 seconds
Rough training...
radius_ini: 4.000000 , radius_final: 1.000000, trainlen: 180

epoch: 1 ---> elapsed time:  0.137000, quantization error: 0.469287

epoch: 2 ---> elapsed time:  0.133000, quantization error: 0.408181

epoch: 3 ---> elapsed time:  0.134000, quantization error: 0.364103
```

…

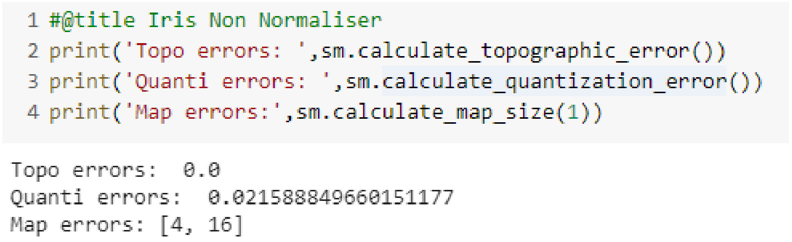


```
epoch: 178 ---> elapsed time:  0.132000, quantization error: 0.061476

epoch: 179 ---> elapsed time:  0.133000, quantization error: 0.059010

epoch: 180 ---> elapsed time:  0.134000, quantization error: 0.056352

Finetune training...
radius_ini: 1.000000 , radius_final: 1.000000, trainlen: 240

epoch: 1 ---> elapsed time:  0.134000, quantization error: 0.052426

epoch: 2 ---> elapsed time:  0.133000, quantization error: 0.051968

epoch: 3 ---> elapsed time:  0.137000, quantization error: 0.051968
```

....



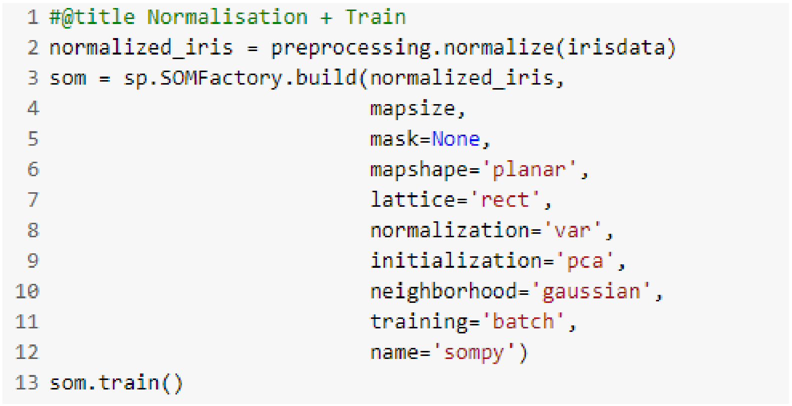
```
epoch: 238 ---> elapsed time:  0.132000, quantization error: 0.051968

epoch: 239 ---> elapsed time:  0.133000, quantization error: 0.051968

epoch: 240 ---> elapsed time:  0.133000, quantization error: 0.051968

Final quantization error: 0.051968
train took: 58.313000 seconds
```



```
#@title Iris Non Normaliser
print('Topo errors: ',sm.calculate_topographic_error())
print('Quanti errors: ',sm.calculate_quantization_error())
print('Map errors:',sm.calculate_map_size(1))
```

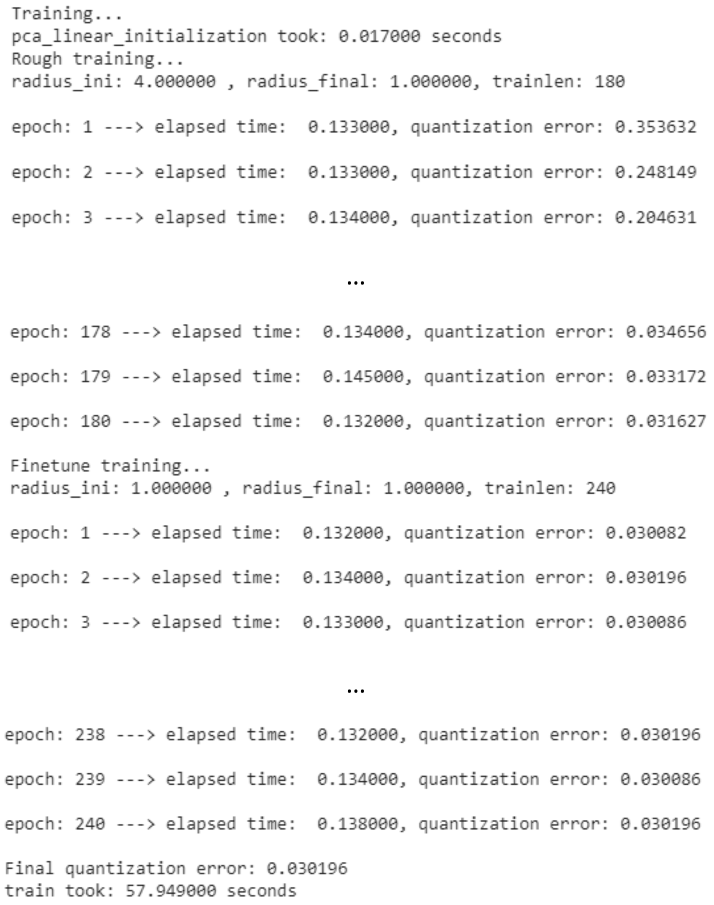
```
Topo errors:  0.0
Quanti errors:  0.021588849660151177
Map errors: [4, 16]
```

Iris Non Normaliser



```
#@title Normalisation + Train
normalized_iris = preprocessing.normalize(irisdata)
som = sp.SOMFactory.build(normalized_iris,
                          mapsize,
                          mask=None,
                          mapshape='planar',
                          lattice='rect',
                          normalization='var',
                          initialization='pca',
                          neighborhood='gaussian',
                          training='batch',
                          name='sompy')
som.train()
```

Normalisation + Train



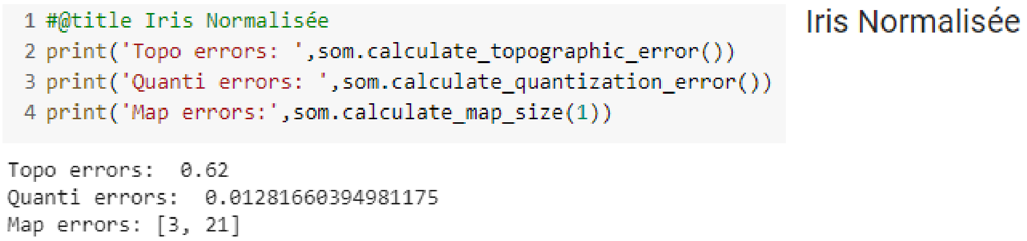
```
Training...
pca_linear_initialization took: 0.017000 seconds
Rough training...
radius_ini: 4.000000 , radius_final: 1.000000, trainlen: 180

epoch: 1 ---> elapsed time:  0.133000, quantization error: 0.353632

epoch: 2 ---> elapsed time:  0.133000, quantization error: 0.248149

epoch: 3 ---> elapsed time:  0.134000, quantization error: 0.204631
```

...

```
epoch: 178 ---> elapsed time:  0.134000, quantization error: 0.034656

epoch: 179 ---> elapsed time:  0.145000, quantization error: 0.033172

epoch: 180 ---> elapsed time:  0.132000, quantization error: 0.031627

Finetune training...
radius_ini: 1.000000 , radius_final: 1.000000, trainlen: 240

epoch: 1 ---> elapsed time:  0.132000, quantization error: 0.030082

epoch: 2 ---> elapsed time:  0.134000, quantization error: 0.030196

epoch: 3 ---> elapsed time:  0.133000, quantization error: 0.030086
```

...

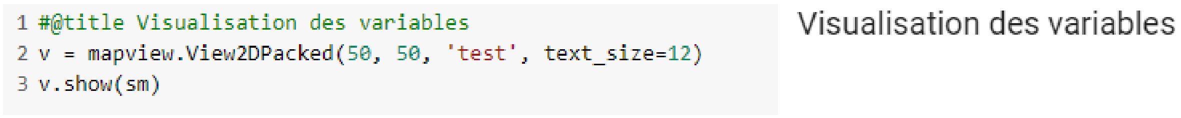
```
epoch: 238 ---> elapsed time:  0.132000, quantization error: 0.030196

epoch: 239 ---> elapsed time:  0.134000, quantization error: 0.030086

epoch: 240 ---> elapsed time:  0.138000, quantization error: 0.030196

Final quantization error: 0.030196
train took: 57.949000 seconds
```



```
#@title Iris Normalisée
print('Topo errors: ',som.calculate_topographic_error())
print('Quanti errors: ',som.calculate_quantization_error())
print('Map errors:',som.calculate_map_size(1))
```

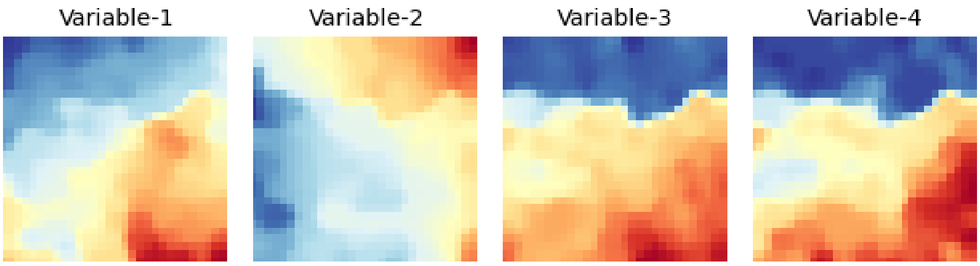
Iris Normalisée

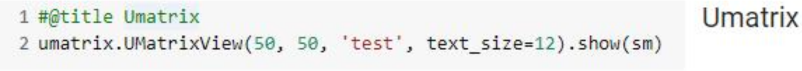
```
Topo errors:  0.62
Quanti errors:  0.012816603949811 75
Map errors: [3, 21]
```



```
#@title Visualisation des variables
v = mapview.View2DPacked(50, 50, 'test', text_size=12)
v.show(sm)
```

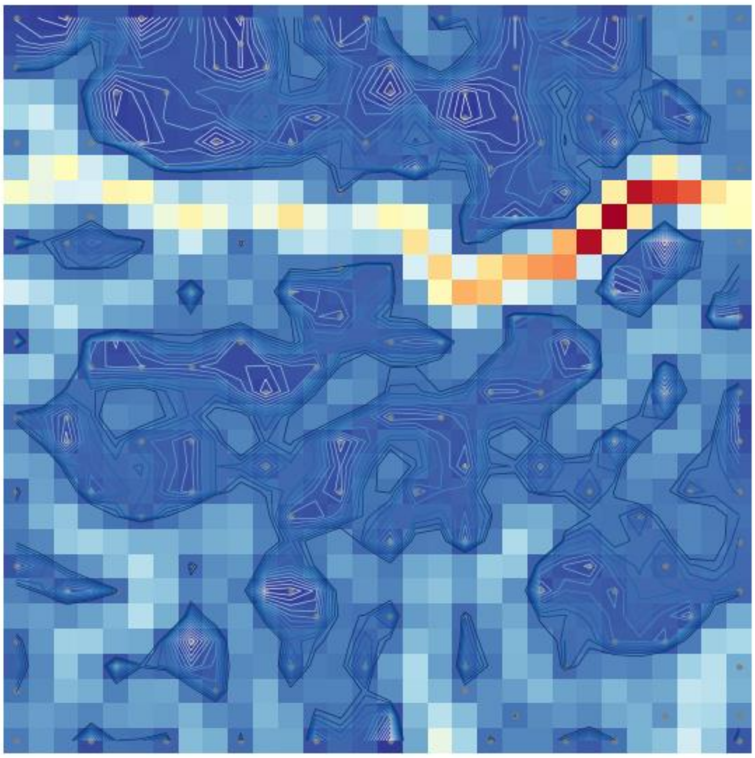
Visualisation des variables

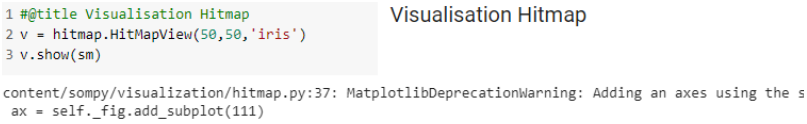




```
#@title Umatrix
umatrix.UMatrixView(50, 50, 'test', text_size=12).show(sm)
```

Umatrix

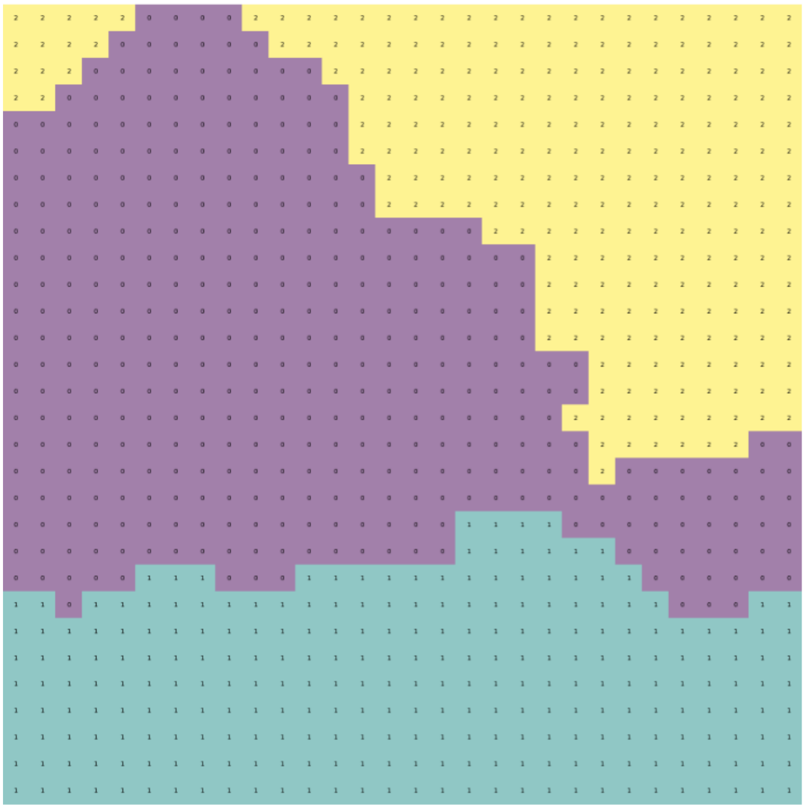


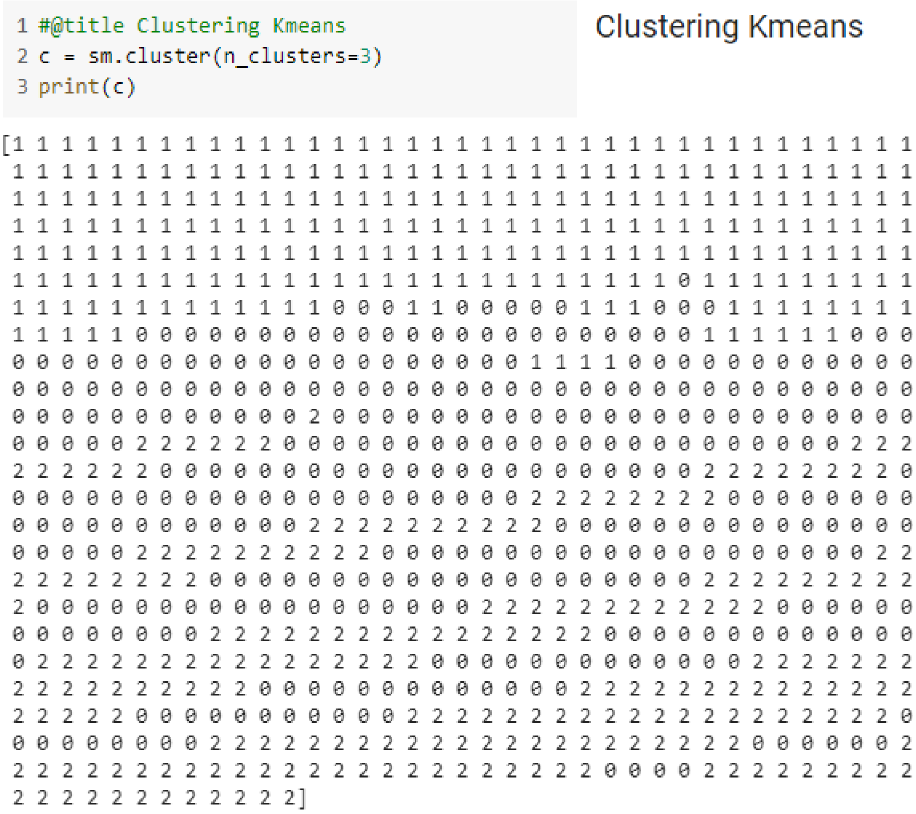


```
#@title Visualisation Hitmap
v = hitmap.HitMapView(50,50,'iris')
v.show(sm)
```

Visualisation Hitmap

```
content/sompy/visualization/hitmap.py:37: MatplotlibDeprecationWarning: Adding an axes using the s
 ax = self._fig.add_subplot(111)
```





```
#@title Clustering Kmeans
c = sm.cluster(n_clusters=3)
print(c)
```

Clustering Kmeans

```
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 0 0 1 1 1 0 0 0 1 1 1 1 1 1 1 1
 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2
 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 2 2 2 2 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2
 2 2 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 2
 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 2 2 2 2 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0
 0 0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 0 0 0 0 0 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 0 0 0 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2]
```