

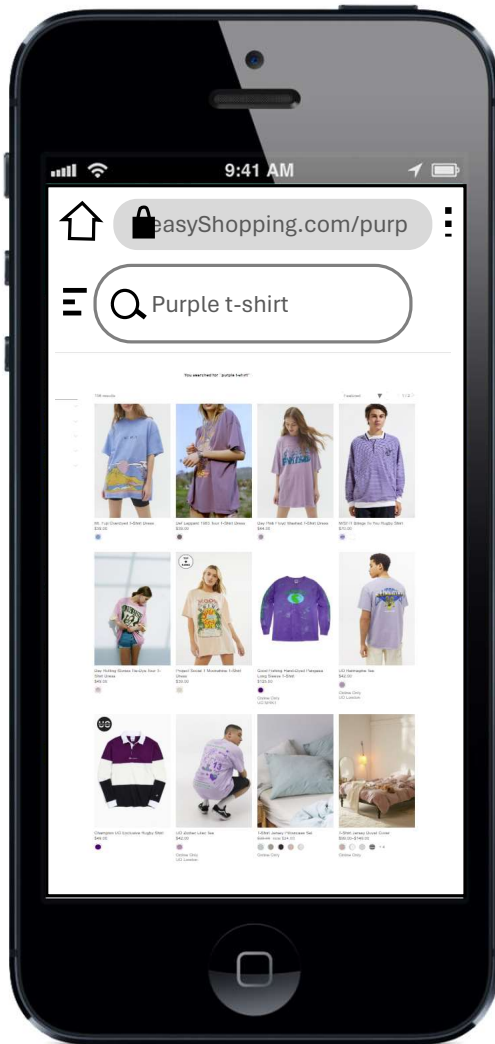
PROJECTE 2 **Etiquetador**

SS: Sessió de Suport
Project 2 - Part 1

Artificial Intelligence

2023-2024

Universitat Autònoma de Barcelona



1. Kmeans i NumPy

La implementació del k-means pot ser molt més eficient si feu servir la llibreria **numpy**, que us permet simplificar la programació amb les següents avantatges:

- Ofereix un **maneig eficient de les dades i optimitzacions** a més baix nivell que eviten la sobrecàrrega interpretativa de Python.
- Aproveiteu els arrays de NumPy per a realitzar càlculs i operacions més fàcilment.
- Proveeix d'un ampli rang de funcionalitats per a **creació i manipulació d'arrays, generació aleatòria de números, cerca d'elements, comparatives eficients** i d'altres operacions matemàtiques.

Exemple: Càlcul de la distància euclidiana, $d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2}$

```
def euclidean_distance(x,y):  
    return np.sqrt(np.sum(np.power(x-y,2)))
```

2. Carpeta images i fitxer utils.py

Etiquetarem els colors predominants de cada peça de roba



- Recordeu que a la carpeta images trobareu el ground-truth al fitxer **gt.json**
"12721": ["Sandals", ["Black", "Blue", "Grey", "White"]], "22250": ["Sandals", ["Brown", "White", "Yellow"]], ...
- Recordeu que trobareu funcions per a convertir les imatges en color en altres espais al fitxer **utils.py**.
Per exemple la funció que passa de RGB a noms de colors: `utils.get_color_prob()`

Exemple de crida:

```
utils.get_color_prob(A) (A: matriu  $K \times D$  [[254.79, 254.82, 254.70],[190.64 169.34 140.11]] )
```

Retorna:

```
['White','Yellow']
```

3. Número de colors predominants – k ideal

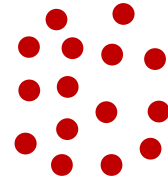
Problema del k-means: Quina **k** s'adapta millor a cada imatge?

La distància intra-class o within-class (WCD):

$$WCD = \frac{1}{N} \sum_{\mathbf{x} \in \mathbf{X}} \text{distancia}(\mathbf{x}, \mathbf{C}_x)^2 = \frac{1}{N} \sum_{\mathbf{x} \in \mathbf{X}} (\mathbf{x} - \mathbf{C}_x)(\mathbf{x} - \mathbf{C}_x)^T$$

Fa la mitjana de les distàncies al quadrat entre cada punt i el representant de la seva classe

- **X** representa el vector de punts
- **C** representa el vector de centroides on cada \mathbf{C}_x representa el centroide de la classe a la que pertany el punt \mathbf{x}
- **N** és el nombre total de punts que tenim a **X**



Classe compacta 
Bona classificació

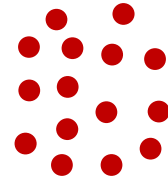
3. Número de colors predominants – k ideal

Problema del k-means: Quina **k** s'adapta millor a cada imatge?

Calcularem el percentatge de decrement de WCD per a cada K:

$$\%DEC_k = 100 \frac{WCD_k}{WCD_{k-1}}$$

En el moment que $100 - \%DEC < 20\%$ (*exemple*) s'estabilitzi a un valor baix, agafarem aquesta **K** com a **K** ideal

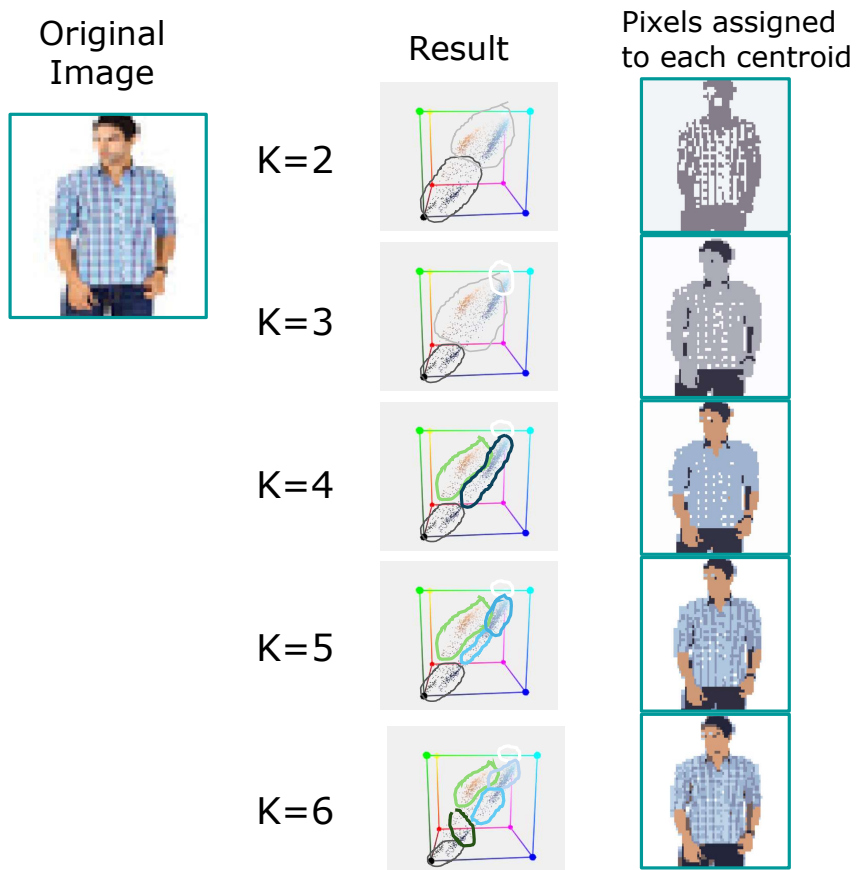


Classe compacta 
Bona classificació

4. See your results

You can visualize the results of your k-means with the function: **visualize_k_means()**

Example:




5. Proveu el vostre codi

Per a posar a provar el vostre codi hareu de descarregar el fitxer:

`<wooclap-testCases-kmeans.zip>`

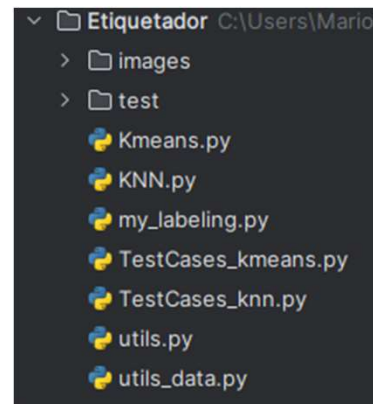
que trobareu a **cv.uab.cat**>>Pràctiques>>Projecte 2. Etiquetatge

1.  **testCases_img**: Carpeta que conté el set d'imatges que utilitzarem per a les proves.
2. `testCases-kmeans.py`: fitxer on escriureu les línies de codi per respondre a les preguntes del qüestionari

5. Proveu el vostre codi

El qüestionari que fareu a continuació us proposarà un seguit de proves per a les vostres funcions de **Kmeans.py**

Deseu el contingut del fitxer `'wooclap-testCases-kmeans.zip'` al mateix directori que la resta de fitxers de codi.



Per contestar el qüestionari haureu de provar cada un dels casos plantejats,

Si no contesteu bé a les preguntes, **haureu d'esbrinar què falla al vostre codi ...**

5. Proveu el vostre codi

```
from Kmeans import *  
from PIL import Image  
import numpy as np
```

*Aquí s'importen tots els fitxers
que contenen les funcions que necessita el vostre codi*

```
test_folder = 'testCases_img/'
```

```
def read_image(image_filename):  
    image_path = test_folder + image_filename  
    image = Image.open(image_path)  
    image = np.array(image)  
    return image
```

```
def test_kmeans_on_image(image_filename, k, option):
```

```
    image_data = read_image(image_filename)
```

```
    ###BELOW HERE YOU CAN CALL ANY FUNCTION THAT YOU HAVE PROGRAMED TO ANSWER THE QUESTIONS FOR THE TEST###
```

```
    kmeans = KMeans(image_data, k, {'km_init': option})  
    kmeans.fit()  
    return kmeans.centroids
```

```
centroids_test_1 = test_kmeans_on_image('4solid_colors.jpg', 1, 'first')  
print("Test 1 Centroids:", centroids_test_1)
```

5. Proveu el vostre codi

```
from Kmeans import *  
from PIL import Image  
import numpy as np
```

Aquí declarem i carreguem la matriu corresponent als pixels de les imatges que avaluarem

```
test_folder = 'testCases_img/'
```

```
def read_image(image_filename):  
    image_path = test_folder + image_filename  
    image = Image.open(image_path)  
    image = np.array(image)  
    return image
```

```
def test_kmeans_on_image(image_filename, k, option):
```

```
    image_data = read_image(image_filename)
```

```
    ###BELOW HERE YOU CAN CALL ANY FUNCTION THAT YOU HAVE PROGRAMED TO ANSWER THE QUESTIONS FOR THE TEST###
```

```
    kmeans = KMeans(image_data, k, {'km_init': option})  
    kmeans.fit()  
    return kmeans.centroids
```

```
centroids_test_1 = test_kmeans_on_image('4solid_colors.jpg', 1, 'first')  
print("Test 1 Centroids:", centroids_test_1)
```

5. Proveu el vostre codi

```
from Kmeans import *  
from PIL import Image  
import numpy as np
```

```
test_folder = 'testCases_img/'
```

```
def read_image(image_filename):  
    image_path = test_folder + image_filename  
    image = Image.open(image_path)  
    image = np.array(image)  
    return image
```

```
def test_kmeans_on_image(image_filename, k, option):
```

```
    image_data = read_image(image_filename)
```

```
    ###BELOW HERE YOU CAN CALL ANY FUNCTION THAT YOU HAVE PROGRAMED TO ANSWER THE QUESTIONS FOR THE TEST###
```

```
    kmeans = KMeans(image_data, k, {'km_init': option})  
    kmeans.fit()  
    return kmeans.centroids
```

```
centroids_test_1 = test_kmeans_on_image('4solid_colors.jpg', 1, 'first')  
print("Test 1 Centroids:", centroids_test_1)
```

Aquí haureu d'inserir les vostres línies de codi per a executar i respondre les preguntes del qüestionari

Entrega Part 1

Per a l'avaluació d'aquesta primer part de la pràctica haureu de pujar al Campus Virtual el vostre fitxer **Kmeans.py** que ha de contenir el **NIU** de tots els membres del grup a la variable authors (a l'inici de l'arxiu). Els NIUs s'hauran d'afegir encara que els grups sigui d'una sola persona (e.g., [1290010,10348822] o [23512434]).

L'entrega s'ha de fer abans del dia **14 d'Abril del 2024 a les 23:55.**

ATENCIÓ! és important que tingueu en compte els següents punts:

1. La **correcció** del codi es fa de manera **automàtica**, per tant, assegureu-vos de penjar els arxius amb la nomenclatura i format correctes. Si no ho poseu bé la nota serà un 0. (No canvieu el nom del fitxer o els imports al principi del fitxer)
2. El codi està sotmès a **detecció automàtica de plagis** durant la correcció.
3. Qualsevol part del codi que no estigui dins de les funcions de l'arxiu Kmeans.py **no** podrà ser **avaluada**, per tant, no modifiquen res fora d'aquest arxiu.
4. Per evitar que el codi entri en bucles infinits hi ha un **límit de temps** per a cada exercici, per tant si les vostres funcions triguen massa les considerarà incorrectes.

Recordeu el que es diu a la **guia docent** sobre **copiar o deixar copiar**

Sense perjudici d'altres mesures disciplinàries que s'estimin oportunes, i d'acord amb la normativa acadèmica vigent, les **irregularitats comeses per l'alumnat** que puguin conduir a una variació de la qualificació es qualificaran amb un zero (0). Les activitats d'avaluació qualificades d'aquesta forma i per aquest procediment no seran recuperables. Si és necessari superar qualsevol d'aquestes activitats d'avaluació per aprovar l'assignatura, aquesta assignatura quedarà suspesa directament, sense oportunitat de recuperar-la en el mateix curs. Aquestes irregularitats inclouen, entre d'altres:

- còpia total o parcial d'una pràctica, informe, o qualsevol altra activitat d'avaluació;
- deixar copiar;
- ús no autoritzat i/o no referenciat de la IA (p. ex, Copilot, ChatGPT o equivalents) per a resoldre exercicis, pràctiques i/o qualsevol altra activitat avaluable;
- presentar un treball de grup no fet íntegrament pels membres del grup;
- presentar com a propis materials elaborats per un tercer, encara que siguin traduccions o adaptacions, i en general treballs amb elements no originals i exclusius de l'alumnat.
- tenir dispositius de comunicació (com telèfons mòbils, smart watches, etc.) accessibles durant les proves d'avaluació teòric-pràctiques individuals (exàmens).

En resum: **copiar, deixar copiar o plagiar** en qualsevol de les activitats d'avaluació equival a un SUSPENS amb **nota inferior o igual a 3,0**.