Matrix Factorization with Python and scikit learn

Exercise 1 : The Olivetti Faces Dataset

1. Import the following librairies

```
from time import time
from numpy.random import RandomState
import pylab as plt
import numpy as np
from sklearn.datasets import fetch_olivetti_faces
from sklearn import decomposition
```

2. We fix the following variables

```
n_row, n_col = 2, 5
n_components = n_row * n_col
image_shape = (64, 64)
rng = RandomState(0)
```

3. Using the function `fetch_olivetti_faces` import the faces from the dataset olivetti, and save the data in variable `face`

4. Center the faces

5. Nous allons maintenant afficher les visages de la bases de données. We define the following function to plot the dataset

```
def plot_gallery(title, images, n_col=n_col, n_row=n_row, cmap=plt.cm.gray):
    plt.figure(figsize=(2. * n_col, 2.26 * n_row))      plt.suptitle(title, size=16)
    for i, comp in enumerate(images):
        plt.subplot(n_row, n_col, i + 1)
        vmax = max(comp.max(), -comp.min())
        plt.imshow(comp.reshape(image_shape),cmap=cmap,interpolation='nearest',
        vmin=-vmax,vmax=vmax)
        plt.xticks(())
        plt.yticks(())
    plt.subplots_adjust(0.01, 0.05, 0.99, 0.93, 0.04, 0.)
```

Comment this function. Use it to plot a part of the database.

6. Use the function `PCA` de la librairie scikit-learn pour extraire `n_components` composantes. Do not forget that the observations are 64 × 64 images and that you have to reshape them to be able tu use PCA.

7. Utiliser la fonction `NMF` de la librairie scikit-learn pour extraire `n_components` composantes.

Exercise 2 : Introduction to recommender systems

We shall use real data that can be downloeaded here :
`https://grouplens.org/datasets/movielens/`.
The data comes from the website `MovieLens`. We shall use the `small version` of the `MovieLens Latest Datasets`. We get precisely 100 004 evaluations on 9125 movies given by 671 distinct users.

Description of the dataset

The dataset contains three tables :

- Movies : In this table are stored the informations related to the 9125 movies. At each line of the file is associated a film with three variables :
  - MovieId : allowing to identify the movies .
  - Title : title of the movie.
  - Genres : film genre.
- Ratings : This table contains the 100 004 score of movies of our dataset. Each line represents a score given by a user for a movie, and each movie can be rated only once by a user. The table contains 4 variables :
  - UserId : identifier of the user.
  - MovieId : identifier of the movie.
  - Rating : score awarded.
  - Timestamp : date of the score.
- Tags : This table contains the 1 296 tags given by the users on the movies. Each line represents a proposed tag by a user for a film at a given time. A fixed user can tag several time a film. The variables are
  - UserId : identifier of the user.
  - MovieId : identifier of the movie.
  - Tag : given tag.
  - Timestamp : date of teh tag.

1. Display the number of distincts users in `ratings` and thereafter the maximal number of marks per user and per movie
2. Define the matrix $id_{user} \times film$
3. Approximate this matrix by a low rank matrix
4. Can we use this approximation to predict the best movie for a given user?