

main

February 12, 2024

1 Atelier 2 - Systèmes de recommandation

1.0.1 Question 5 (b)

En utilisant les approches I-I et U-U, quels films sont recommandés pour un utilisateur qui a voté 5 pour Clockwork Orange (no. 179) et pour Full Metal Jacket (no. 188)? (10 pts.)

```
[42]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import tqdm.autonotebook as tqdm

votes = pd.read_csv("data/votes.csv")
items = pd.read_csv("data/items.csv")
users = pd.read_csv("data/u.csv")

MUI = votes.pivot(index="user.id", columns="item.id", values="rating")
MUI_numpy = MUI.to_numpy()

def cosinus_matrices(A, B):
    A = A.copy()
    B = B.copy()
    A[np.isnan(A)] = 0
    B[np.isnan(B)] = 0
    axis = 1
    col_norm_A = np.linalg.norm(A, axis=axis, keepdims=True)
    col_norm_B = np.linalg.norm(B, axis=axis, keepdims=True).T
    W = A @ B.T / (np.abs(col_norm_A * col_norm_B) + 1e-10)
    W = W * (np.ones_like(W) - np.identity(W.shape[0]))
    return W

MUI_Query = np.zeros_like(MUI_numpy)
MUI_Query[0, 179] = 1
MUI_Query[0, 188] = 1
```

Approche Utilisateur-Utilisateur

```
[46]: similariter = cosinus_matrices(MUI_numpy, MUI_Query)[: , 0]
      np.argsort(similariter)[::-1]
      items[" movie title "].iloc[np.argsort(similariter)[::-1][:10]]
```

```
[46]: 28          Batman Forever (1995)
      821          Faces (1968)
      414      Apple Dumpling Gang, The (1975)
      337          Bean (1997)
      639      Cook the Thief His Wife & Her Lover, The (1989)
      155          Reservoir Dogs (1992)
      521          Down by Law (1986)
      887          One Night Stand (1997)
      391      Man Without a Face, The (1993)
      660          High Noon (1952)
      Name: movie title , dtype: object
```

Approche Item-Item

```
[44]: similariter = cosinus_matrices(MUI_numpy.T, MUI_Query.T)[: , 0]
      np.argsort(similariter)[::-1]
      items[" movie title "].iloc[np.argsort(similariter)[::-1][:10]]
```

```
[44]: 1681      Scream of Stone (Schrei aus Stein) (1991)
      577          Demolition Man (1993)
      553          Waterworld (1995)
      554      White Man's Burden (1995)
      555          Wild Bill (1995)
      556      Farinelli: il castrato (1994)
      557      Heavenly Creatures (1994)
      558      Interview with the Vampire (1994)
      559      Kid in King Arthur's Court, A (1995)
      560      Mary Shelley's Frankenstein (1994)
      Name: movie title , dtype: object
```

1.0.2 Question 6 (b)

Je suis une femme ingénieure de plus de 52 ans. Quelle est la probabilité que j'aime Toy Story? Utilisez une approche Bayésienne dichotomique où le seuil pour aimer est un vote de 4 et plus et où celui de l'âge est fixé à 52 ans. Présumez de l'indépendance des facteurs. (10 pts.)

```
[56]: import pandas as pd
      import numpy as np
      import matplotlib.pyplot as plt
      import tqdm.autonotebook as tqdm

      votes = pd.read_csv("data/votes.csv")
      items = pd.read_csv("data/items.csv")
      users = pd.read_csv("data/u.csv")
```

```
MUI = votes.pivot(index="user.id", columns="item.id", values="rating")
MUI_numpy = MUI.to_numpy()
```

$$P(\text{aime} | \text{age} > 52, \text{job} = \text{ingenieur}) = \frac{P(\text{age} > 52, \text{job} = \text{ingenieur} | \text{aime}) * P(\text{aime})}{P(\text{age} > 52, \text{job} = \text{ingenieur})}$$

```
[88]: aime_toyStory = (MUI_numpy >= 4)[: , 0]
users_above_52_ingenieur = (users[" age "] > 52) & (users[" job "] ==
↪ "engineer").to_numpy()

Prob_aime_toyStory = np.mean(aime_toyStory)
Prob_users_above_52_ingenieur = np.mean(users_above_52_ingenieur)
Prob_users_above_52_ingenieur_sachant_aime_toyStory = np.
↪ mean(users_above_52_ingenieur[aime_toyStory])
Prob_aime_toyStory_sachant_users_above_52_ingenieur = (
    Prob_users_above_52_ingenieur_sachant_aime_toyStory * Prob_aime_toyStory
) / Prob_users_above_52_ingenieur
Prob_aime_toyStory_sachant_users_above_52_ingenieur
```

```
[88]: 0.4285714285714286
```

La probabilité qu'une personne aime Toy Story sachant que je suis une femme ingénieure de plus de 52 ans est de **0.43**.