

## Projet IA : Recommandation de films

### Problématique du projet :

Le but de ce projet est, à partir d'un historique d'évaluations de films par des utilisateurs, de recommander un film/plusieurs films à un utilisateur.

### Description des sources utilisées :

Pour répondre à notre problématique, nous sommes parti d'un Dataset (pouvant être trouvé ici : <https://grouplens.org/datasets/movielens/>) contenant 3 fichiers :

- Un fichier « movies.dat » contenant une liste de 3952 films, chacun ayant un identifiant unique, un titre ainsi qu'une liste de genres de la forme : « IdDuFilm::Titre::Genres »

Exemple de film : « 1::Toy Story (1995)::Animation|Children's|Comedy »

- Un fichier « ratings.dat » contenant une liste d'évaluations des films par des utilisateurs, chacune ayant l'identifiant d'un utilisateur, l'identifiant d'un film, une note allant de 1 à 5 ainsi qu'un temps en seconde depuis l'évaluation de la forme :

« IdDeLUtilisateur::IdDuFilm::Note::Date ». Tous les utilisateurs ont au moins réalisé 20 évaluations. Exemple d'évaluation : « 1::1193::5::978300760 »

- Un fichier « users.dat » contenant une liste de 6040 utilisateurs, chacun ayant un identifiant unique, un genre, un âge, une occupation de la forme :

« IdDeLUtilisateur::Genre::Age::Occupation::ZipCode ». Exemple d'utilisateur : « 1::F::1::10::48067 »

### Choix du modèle :

Pour le choix de notre modèle, nous nous sommes aidés d'un article pour nous décider sur un modèle et nous aider en conséquence à comprendre ce dont nous aurions besoin parmi nos données. (L'article en question : <https://www.invivoo.com/blog/systeme-de-recommandation/>). Nous avons donc choisi d'utiliser 2 systèmes, tout d'abord la « Global Baseline » qui nous permet d'estimer les notes des films n'ayant pas été vu par un utilisateur en se basant sur les évaluations déjà données par l'utilisateur. Par exemple, on veut savoir la note que l'utilisateur 1 donnerait à un film A qu'il n'a jamais vu, on calcule alors une notre moyenne de l'utilisateur 1, par exemple 4.1, la moyenne des notes du film A, par exemple 2.6, la note moyenne de tous les utilisateurs, par exemple 3.6, et la note moyenne de tous les films, par exemple 3.2. Puis on applique la formule donnée dans l'article, c'est-à-dire :  $\text{Note de A estimé} = 4.1 + (4.1 - 3.6) + (2.6 - 3.2) = 4$ . Ensuite, la « Collaborative filtering » qui nous permet d'affiner les notes estimées précédemment grâce aux préférences des utilisateurs pour les genres de films. Par exemple, même exemple que précédemment, la note estimée de A par l'utilisateur 1 est 4. On sait que l'utilisateur a donné une note de 1/5 à un film B du même genre que A. Donc on met à jour la note de A :  $4 - 1 = 3$ , la note finale estimée du film A par l'utilisateur 1 est 3.

## **Modélisation :**

Tout d'abord, nous avons dû trouver comment utiliser les données des fichiers « .dat » dans notre programme. Pour cela, nous avons réalisé plusieurs fonctions permettant de transformer les données des fichiers « .dat » en liste, puis de les transformer en DataFrame car c'était plus simple pour nous d'utiliser les DataFrames avec des noms de colonnes clairs plutôt que des listes. Nous avons aussi décidé qu'il ne nous était pas nécessaire de transformer les données du fichier « users.dat » en DataFrame car nous n'utilisons aucun de ses attributs dans notre modèle, excepté l'identifiant de l'utilisateur que nous récupérons dans le fichier « ratings.dat ».

Ensuite, nous avons codé le système « Global Baseline » à partir des DataFrame créé précédemment. Nous avons tout d'abord récupéré la note moyenne d'un utilisateur dont on donne l'identifiant, puis nous avons obtenu la note moyenne de tous les utilisateurs, qui est 3.581564453029317, ensuite nous avons obtenu la moyenne des notes pour chacun des films de notre DataFrame dans une liste et la moyenne de note de tous les films, enfin grâce à tout ce que nous avons calculé précédemment, nous avons pu créer une liste contenant une note estimée pour chaque film par l'utilisateur donné.

Puis, après nous être rendu compte que le premier système ne suffisait pas puisque par exemple les films n'ayant été notés qu'une seule fois avec une note de 5 étaient tout le temps recommandés en premier, nous avons codé le système « Collaborative filtering ». Nous avons tout d'abord récupéré la liste de tous les genres possibles, puis la liste des genres de chaque film, ensuite nous avons consulté les genres des films vus par l'utilisateur et avons attribué une note à chaque genre selon les notes données par l'utilisateur, si un film est noté 1 ou 2, on diminue la note du genre de 1 ou 0.5 respectivement, et si la note est 4 ou 5, nous augmentons la note du genre de 0.5 ou 1 respectivement. Par exemple, si un film du genre « Animation » a été noté 2 et un autre film du même genre noté 5, nous avons donné une note au genre « Animation » de  $1 - 0.5 = 0.5$ . Enfin, nous avons mis à jour les notes estimées de chaque film en fonction des notes de genre. Par exemple, le genre « Animation » a la note 0.5 pour l'utilisateur donné, donc on ajoute 0.5 à la note de chaque film ayant le genre « Animation ».

Enfin, nous avons mis à 0 les notes estimées des films déjà vus par l'utilisateur et nous avons cherché les 5 films ayant les plus hautes notes estimées et nous les recommandons à l'utilisateur.

## **Résultats :**

Nous lançons le programme avec l'utilisateur 1 :

```
[INFO]Ouverture du fichier ../Data/ratings.dat
[INFO]Réussi
[INFO]Ouverture du fichier ../Data/movies.dat
[INFO]Réussi
Vous pouvez aussi passer l'utilisateur en paramètre
projet.py < id utilisateur >

Veuillez donner l'id de l'user (ex : 577) : 1

le score moyen de l'utilisateur 1 est 4.188679245283019

    Genres  Score
0  Animation  2.0
1  Children's  3.5
2  Comedy    9.0
3  Adventure  4.0
4  Fantasy   0.0
5  Romance   7.5
6  Drama    14.5
7  Action    4.5
8  Crime     2.5
9  Thriller   4.5
10 Horror    0.5
11 Sci-Fi    2.0
12 Documentary 1.0
13 War       1.0
14 Musical   0.5
15 Mystery   0.0
16 Film-Noir 0.0
17 Western   0.0

Voici certain film qui pourrait plaire à l'utilisateur :

3307 -- City Lights (1931) -- Comedy|Drama|Romance
1172 -- Cinema Paradiso (1988) -- Comedy|Drama|Romance
1211 -- Wings of Desire (Der Himmel über Berlin) (1987) -- Comedy|Drama|Romance
1244 -- Manhattan (1979) -- Comedy|Drama|Romance
1264 -- Diva (1981) -- Action|Drama|Mystery|Romance|Thriller
```

On observe que l'utilisateur 1 a un score moyen de 4.19/5 et que cet utilisateur aime beaucoup les genres Drama, Comédie et Romance. Ainsi, les films recommandés par notre programme sont les films les mieux notés parmi tous les utilisateurs et qui correspondent aux goûts de l'utilisateur, ils ont tous ou presque tous le genre Drama, Romance et Comédie.

## Conclusion :

Notre modèle nous permet bien de réaliser la recommandation de films à un utilisateur à partir de son historique, donc notre objectif est atteint. Cependant, nous remarquons que notre modèle recommande souvent les mêmes films selon les genres préféré des utilisateurs, par exemple les utilisateurs appréciant particulièrement les films du genre Drama auront dans leur recommandation les mêmes films que les autres utilisateurs, tel que City Lights ou Cinema Paradiso, tant qu'ils ne les auront pas vu. Sachant cela, nous aurions pu utiliser un autre modèle pour le comparer au notre car il existe plusieurs autres modèles différents avec d'autres points forts et points faibles.