**RAPPORT DU PROJET**

**Projet N°1 :** Amélioration d’un système de recommandation des films en utilisant l’intelligence artificielle sous python



Génie des systèmes embarqués et informatique industrielle – 2ème année

Réalisé par : BOUGRINE Imane Encadré par : Pr CHOUGRAD Hiba

EL-KANTRI Youssef

MANSOURI Anas

RIZKI Younes

2019-2020

Table des matières

[II. INTRODUCTION 5](#_Toc34769778)

[III. ANALYSE FONCTIONNELLE 6](#_Toc34769779)

[A. Item-based collaboratif filtering 6](#_Toc34769780)

[B. Similitude basée sur Pearson (corrélation) 6](#_Toc34769781)

[IV. LAB1 : 8](#_Toc34769782)

[1er exemple : 8](#_Toc34769783)

[2ème exemple 8](#_Toc34769784)

[3ème exemple 8](#_Toc34769785)

[4ème exemple : 8](#_Toc34769786)

[5ème exemple : 9](#_Toc34769787)

[6ème exemple : 10](#_Toc34769788)

[7ème exemple : 10](#_Toc34769789)

[8ème exemple : 11](#_Toc34769790)

[9ème et 10ème exemples : 11](#_Toc34769791)

[11ème exemple : 12](#_Toc34769792)

[V. Activité du LAB1 13](#_Toc34769793)

[VI. LAB2 14](#_Toc34769794)

[1er exemple : 14](#_Toc34769795)

[2ème exemple : 14](#_Toc34769796)

[3ème exemple : 14](#_Toc34769797)

[4ème exemple : 15](#_Toc34769798)

[5ème exemple 15](#_Toc34769799)

[6ème exemple 16](#_Toc34769800)

[7ème et le 8ème exemples 17](#_Toc34769801)

[9ème exemple 17](#_Toc34769802)

[VII. Activité du LAB2 18](#_Toc34769803)

[A. Modification de la min\_period : 18](#_Toc34769804)

[B. Modification de la méthode : 18](#_Toc34769805)

[C. Elimination des films similaires aux films auxquels il a donné un petit rating 19](#_Toc34769806)

[D. Elimination des valeurs aberrantes : 20](#_Toc34769807)

[VIII. Test du système : 21](#_Toc34769808)

[A. Test du programme sans modification 21](#_Toc34769809)

[B. Test du programme après modification de méthode : Spearman 21](#_Toc34769810)

[C. Test du programme après modification de min\_periods : 150 22](#_Toc34769811)

[D. Test du programme après élimination des outliers 22](#_Toc34769812)

[IX. RESUME 23](#_Toc34769813)

[X. CONCLUSION 24](#_Toc34769814)

# INTRODUCTION

Dans le monde numérique d'aujourd'hui, où il existe une variété infinie de contenus à consommer tels que des livres, des vidéos, des articles, films, etc., trouver le contenu de son goût est devenu une tâche fastidieuse. D'autre part, les fournisseurs veulent engager autant d'utilisateurs que possible sur leur service pendant la durée maximale. C'est ici que système de recommandation entre en scène où les fournisseurs de contenu recommandent aux utilisateurs le contenu selon le goût des utilisateurs. Dans ce rapport, nous avons amélioré un système de recommandation des films. L’objectif de système est de fournir des recommandations de film précises aux utilisateurs. Généralement, les systèmes de recommandation de base considèrent l'un des facteurs suivants pour générer des recommandations :

- la préférence de l'utilisateur (c'est-à-dire basé sur le contenu filtrage)

- la préférence d'utilisateurs similaires (filtrage collaboratif, par exemple). Pour construire un environnement stable et précis

Le système de recommandation utilisé dans ce projet est basé sur le filtrage collaboratif basé sur les éléments (dans ce cas les films).

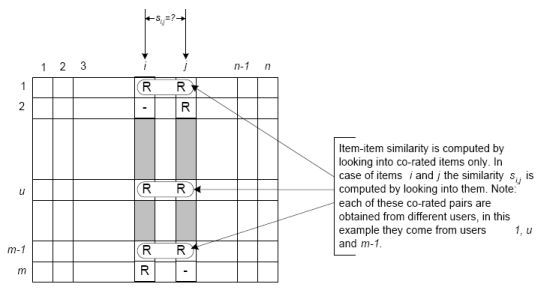
# ANALYSE FONCTIONNELLE

Le système développé dans ce projet utilise la technique du filtrage collaboratif basé sur les éléments (item-based collaboratif filtering), alors c’est quoi le filtrage collaboratif basé sur les éléments ?

## Item-based collaboratif filtering

Le filtrage collaboratif basé sur les éléments est un algorithme basé sur un modèle pour formuler des recommandations. Dans l'algorithme, les similitudes entre les différents éléments de l'ensemble de données sont calculées en utilisant l'une des nombreuses mesures de similitude, puis ces valeurs de similitude sont utilisées pour prédire les évaluations des paires utilisateur-élément non présentes dans l'ensemble de données.

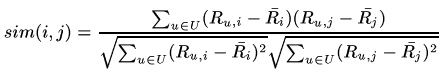
Les valeurs de similitude entre les éléments sont mesurées en observant tous les utilisateurs qui ont évalué les deux éléments. Comme le montre le diagramme ci-dessous, la similitude entre deux éléments dépend des notes attribuées aux éléments par les utilisateurs qui les ont évalués tous les deux:



Il existe un certain nombre de formulations mathématiques différentes qui peuvent être utilisées pour calculer la similitude entre deux éléments. Comme on peut le voir dans les formules ci-dessous, chaque formule comprend des termes additionnés sur l'ensemble des utilisateurs communs U.

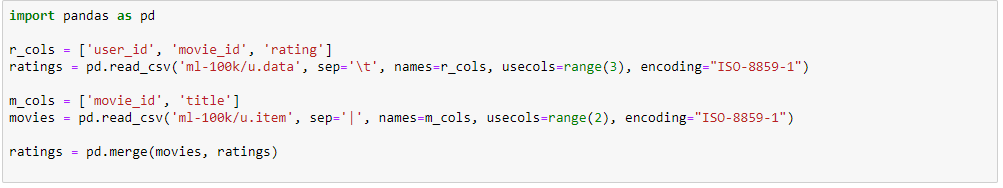
## Similitude basée sur Pearson (corrélation)

Cette mesure de similitude est basée sur la différence entre les notes attribuées par les utilisateurs ordinaires pour une paire d'articles par rapport aux notes moyennes de ces articles:



# LAB1 :

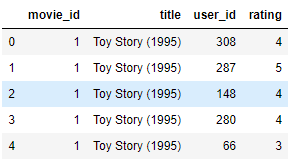
1er exemple : On a importé les données u.data et u.item et on les a combiné dans une dataframe nommée « ratings »



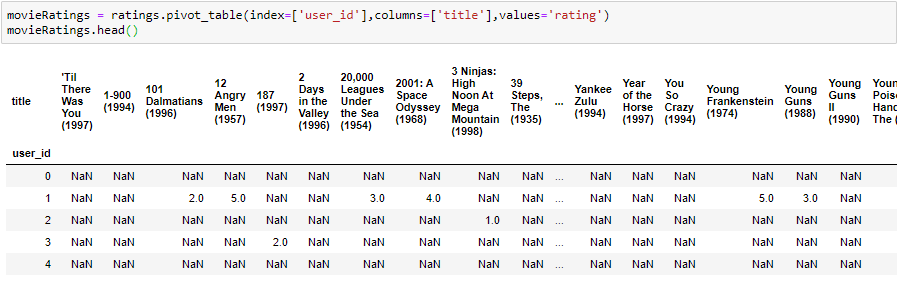
2ème exemple**:** dans cette exemple on a affiché les cinq premières lignes afin de visualiser la dataframe



Le résultat est le suivant :



3ème exemple**:** Dans cet exemple on a utilisé la fonction pivot\_table pour présenter l’évaluation de tous les films pour chaque user (ie les colonnes représentent les titres et les lignes représentent les user\_id et le point d’intersection représente le rating)



4ème exemple : on a affiché les 5 premieres ratings des users pour le film « Star Wars »

Le user ayant l’id 0 a evalué le film par 5 étoiles c’est à dire qu’il a aimé le film

Le user ayant l’id 1 a evalué le film par 5 étoiles et ainsi de suite



5ème exemple :on a d’abord récupérer la colonne qui contient ratings de tous les users sur le film Star Wars puis on a enlever les lignes des users qui n’ont pas évalué ce film ( ces valeurs sont représenté par NaN). Ensuite on a utilisé la fonction corrwidth pour mesurer la similarité de tous les films avec le film Star Wars et on a affiché les 10 premiers lignes obtenu .

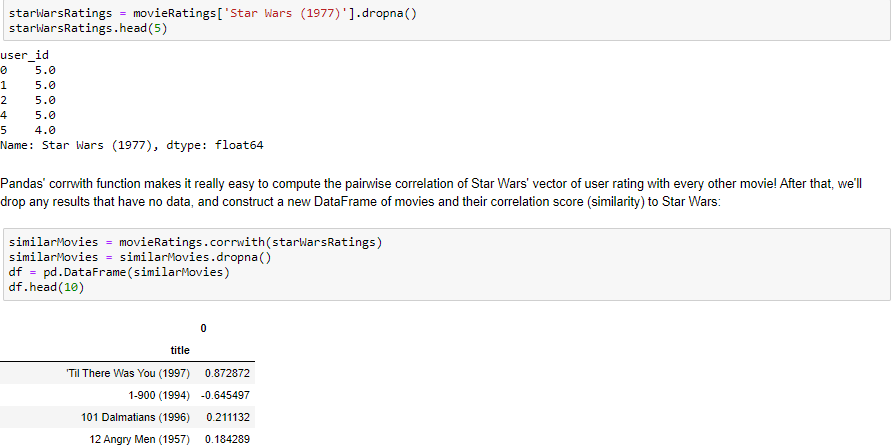
On remarque que les valeurs sont comprises entre -1 et 1

Zdfndsklfjeklfje

Les valeurs très proches de 1 sont dites de corrélation positive très forte c’est-à-dire qu’il sont très similaire à Star Wars et que les users qui ont aimé Star Wars ont aussi aimé ces films.

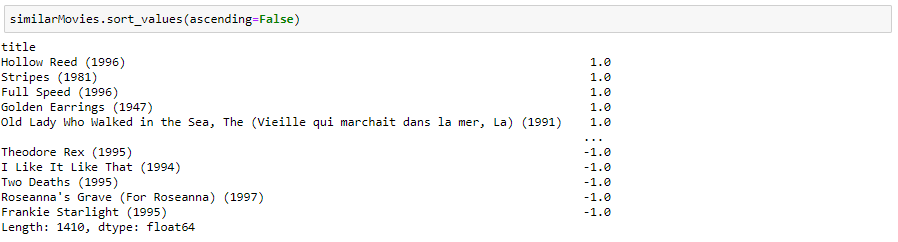
Les valeurs très proches de -1 sont dites de corrélation négative très forte c’est-à-dire que les users qui ont aimé Stars Wars n’ont pas aimé ces films.

Et les valeurs qui sont aux alentours de 0 n’ont pas de dépendance avec Stars Wars , c’est-à-dire que les users ont donné des ratings dispersés de telle sorte qu’on a pas pu établir une relation entre ces films et Star Wars

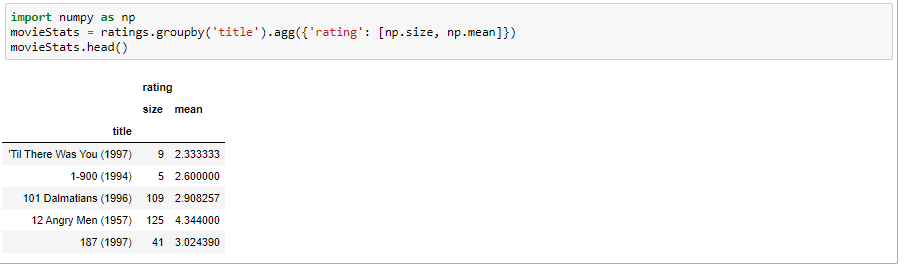


6ème exemple :dans cet exemple on a trié les titres des filmes selon la corrélation avec Star Wars en ordre décroissant**.**

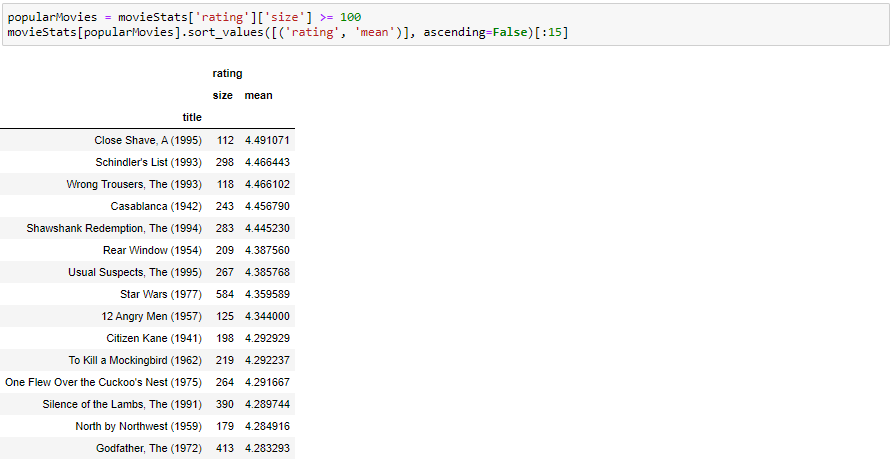
On ne peut pas juger si on a obtenu de bons résultats car il se peut qu’un soit évalué par quelques personnes seulement donc le résultat sera erroné.



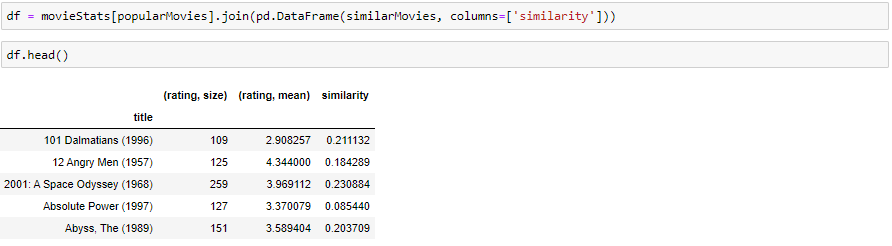
7ème exemple :dans cet exemple on a calculé pour chaque film le nombre total d’évaluations qui lui ont été accordé, ainsi que la moyenne de ces évaluations.



8ème exemple :On a stocké dans la variable popularMovies des valeurs booléennes dans les indices de la table telles que True représente les films ayant size >100 (c’est-à-dire le nombre de personnes qui ont évalué le film est supérieur à 100) et False sinon.

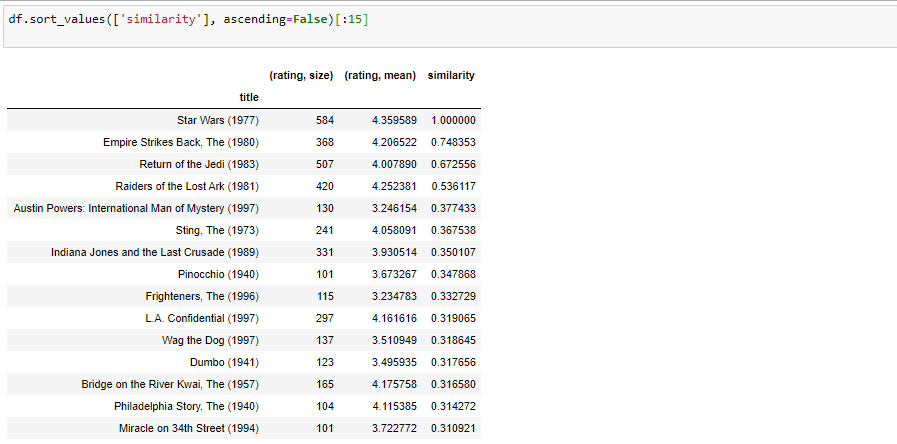
Puis on a affiché ces films en passant la variable popularMovies en indice de la table des évaluations moviesStats

9ème et 10ème exemples :dans cet exemple on a combiné les valeurs de l’exemple 8 avec les valeurs de la similarité de ces films avec Star Wars et on a affiché les 5 premières lignes



11ème exemple :dans cet exemple on a trié les valeurs obtenues dans l’exercice 9 et on a affiché les 15 premières valeurs.

On remarque que Star Wars a une corrélation de 1 et c’est logique c’est le film le plus similaire ç lui même

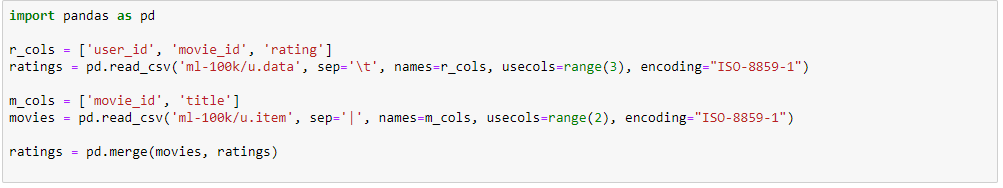


# Activité du LAB1

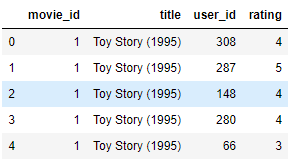
Lorsqu’on a modifié la seuil à 6 on a remarqué que le système nous recommande le film Hollow Reed même si la moyenne de rating de ce film est très lointe de celle de Star Wars . Alors que c’est faux car le système a pris que les 6 personnes qui ont voté sur Hollow Reed et sur Star Wars et il a trouvé qu’il avaient le même rating alors il a considéré que les deux films sont similaires mais ce n’est pas le cas car les autres personnes qui ont évalué Star Wars sont une majorité qui n’ont pas été pris en compte.

# LAB2

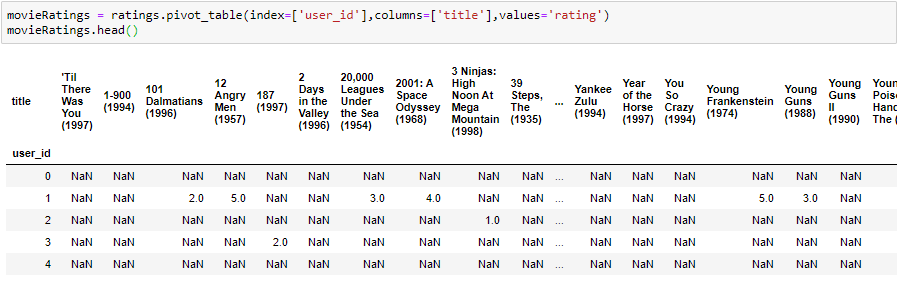
1er exemple : On a importé les données u.data et u.item et on les a combiné dans une dataframe nommée « ratings » et on a affiché les cinq premières lignes afin de visualiser la dataframe





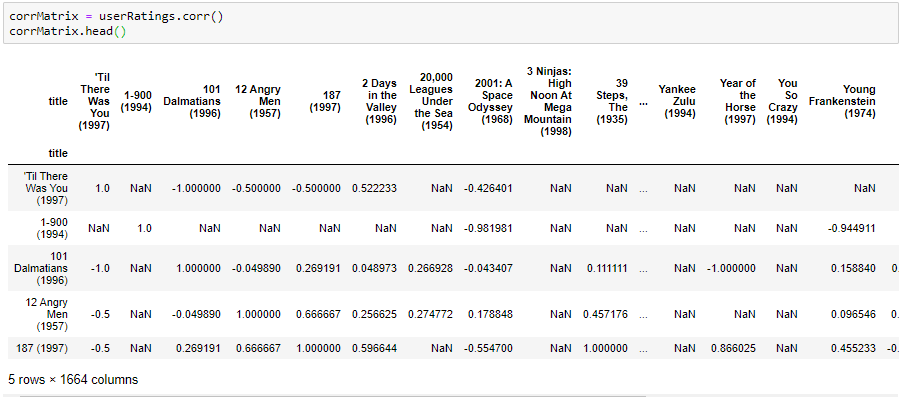


2ème exemple :Dans cet exemple on a utilisé la fonction pivot\_table pour présenter l’évaluation de tous les films pour chaque user (ie les colonnes représentent les titres et les lignes représentent les user\_id et le point d’intersection représente le rating)



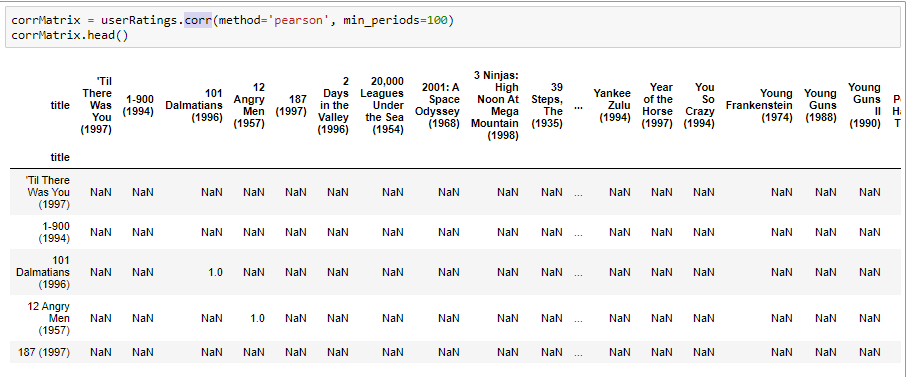
3ème exemple : Dans cet exemple on a utilisé la fonction corr( ) prédéfini dans Pandas pour calculer le taux de corrélation de chaque film avec les autres films de la dataset.

Interprétation: chaque intersection de deux films dans la table présente la valeur de correlation entre ces deux films avec une valeur de min\_period égale à 1 par défaut ( c’est à dire il suffit que les deux films soient évalués par au moins une personne )



4ème exemple :Dans cet exemple on a augmenté la valeur de min\_period à 100 afin de filtrer les films dont le nombre de personnes communes qui ont évalué ne dépasse pas 100 (c’est-à-dire la valeur d’intersection de ces films devient NaN).

On remarque que le nombre de NaN a augmenté et cela une preuve de prise en compte de min\_period



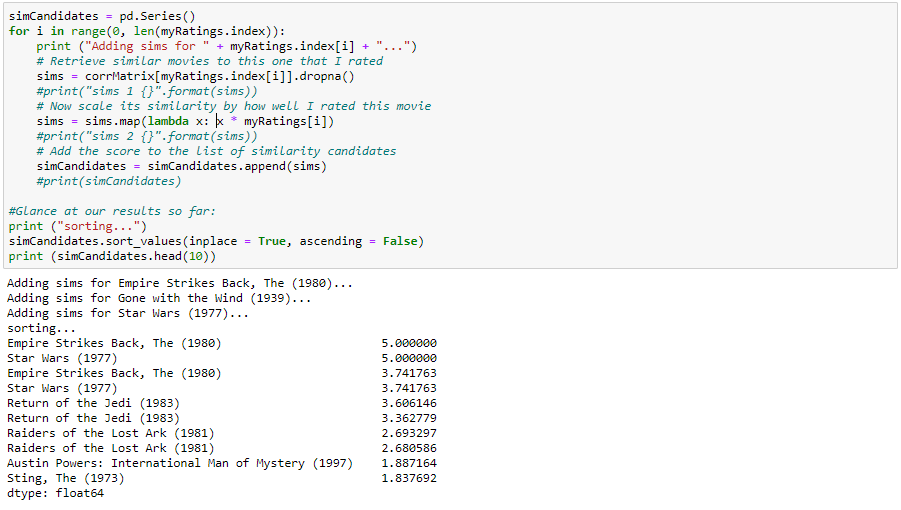
5ème exemple**:** Dans cet exemple on a pris la table « userRatings » qui représente l’évaluation de chaque utilisateur pour chaque film, puis on a récupéré la première ligne qui représente l’ID N° 0 , ensuite on a supprimé les valeurs NaN c’est-à-dire les films qui ne sont pas évalués par ce user et on a affiché le résultat .

On remarque que l’user d’ID 0 a évalué 3 films ‘Empire Strike’ , ‘Gone with the Wind’ et ‘Star Wars’

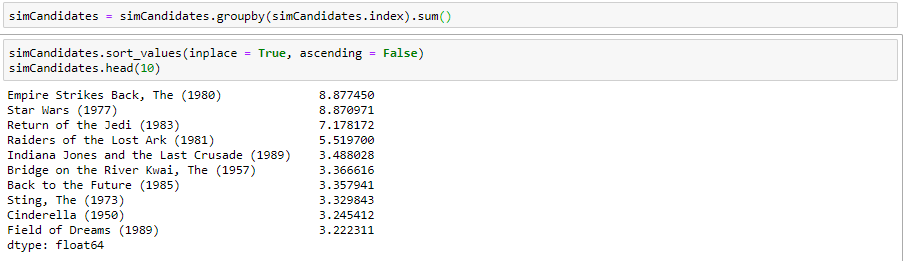


6ème exemple**:** Dans cet exemple on a rempli la Serie simCandidates par les films similaires aux 3 films évalués par l’user d’ID 0 et le coefficient correspondant à chaque film dit similaire est calculé en multipliant le coefficient de similarité par le rating du film similaire évalué par le user N°0.

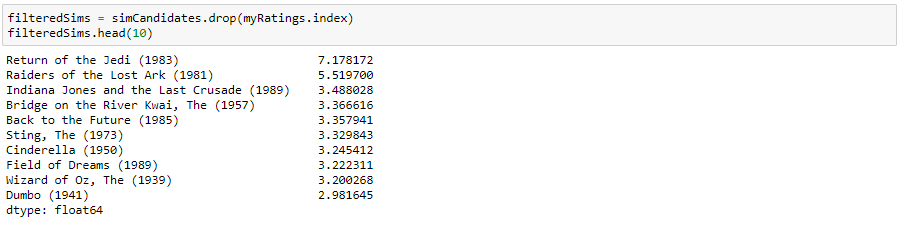
On remarque que plusieurs films apparaissent plus d’une fois , et cela du au fait qu’un film peut être simlilaire à plus d’un film évalué par l’utilisateur (c’est-à-dire si notre user a évalué le film X et Y , et le film Z est similaire aux deux films X et Y alors Z va apparaitre deux fois ).



7ème et le 8ème exemples **:** On a utilisé la fonction groupby pour grouper les films qui apparaissent plusieurs fois et on a défini une seule note d’évaluation pour chaque film qui se répète comme étant la somme des notes d’évaluation qui lui correspondent, puis on a affiché le résultat .



9ème exemple**:** Dans cet exemple on a enlevé les films dèjà évalués par l’utilisateur d’ID 0 pour ne pas

Les lui recommander encore une fois.

# Activité du LAB2

Pour l’instant on a construit un système de recommandation de films pour n’importe quel utilisateur dans la Dataset, Pour améliorer les résultats de la recommandation on va essayer d’appliquer les 4 méthodes suivantes :

## Modification de la min\_period :

Comme il est déjà montré, la variation de la min\_period affecte les résultats de recommandations, plus on augmente la min\_period plus le résultat s’améliore, mais à un certain niveau , le résultat reste inchangeable ou bien le système ne trouve même pas les films à recommander car ils ne sont pas évalués par le nombre min\_period de personne (le cas de 800 par exemple).

C’est pour cela il faut choisir la min\_period qui donne le bon résultat toujours, et pour ce faire on a pensé de remplacer la min\_period par la médiane des nombres d’évaluation de chaque deux films , car la médiane est un bon estimateur de moyenne et ne biaise pas le résultat.

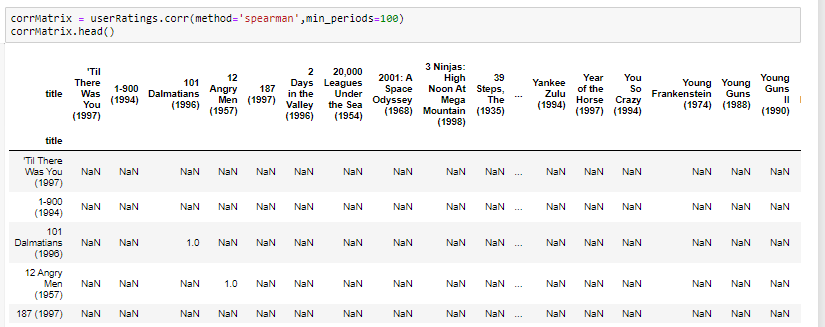
## Modification de la méthode :

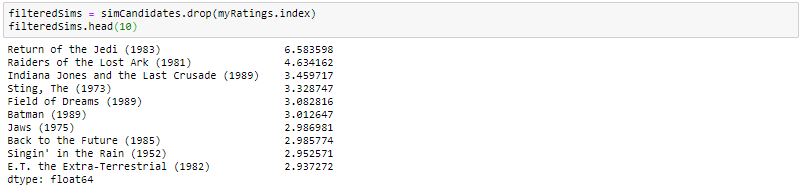
Il y’a 3 méthode de corrélation : ‘Pearson’, ‘kendall’ et ‘spearman’.

On a déjà utilisé la méthode Pearson

La méthode kendall n’utilise pas la min\_period et elle est trop complexe , et prend un temps d’exécution très grand

L’algorithme spearman :





On remarque que les coefficient ont été diminués et le quatrième film recommandé a été changé par un autre, mais on ne peut pas dire si c’est performant ou pas , le test N°5 va nous montré est ce que c’est mieux que pearson ou non

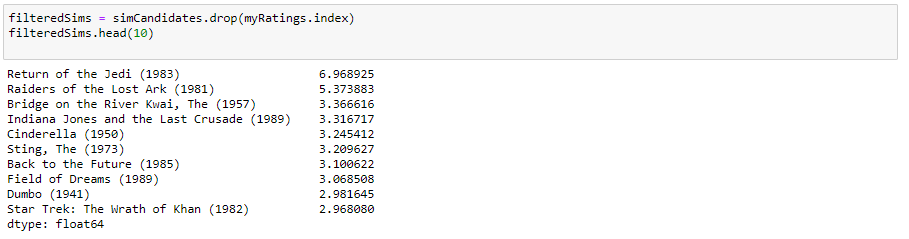
## Elimination des films similaires aux films auxquels il a donné un petit rating

Le système réalisé précédemment a affiché les films similaires à tous films évalués par l’utilisateur, donc même si l’utilisateur n’aime pas un film on va lui recommander un film similaire à celui-ci. Donc pour palier à ce problème on va désigner un seuil au-dessous duquel le film est considéré mal évalué par l’utilisateur c’est-à-dire l’utilisateur ne l’a pas aimé et donc les films similaires à celui-ci ne sera pas recommandé. Le seuil choisi dans ce cas est 3.

D’abord on filtre les films ayant un rating inférieur à 3

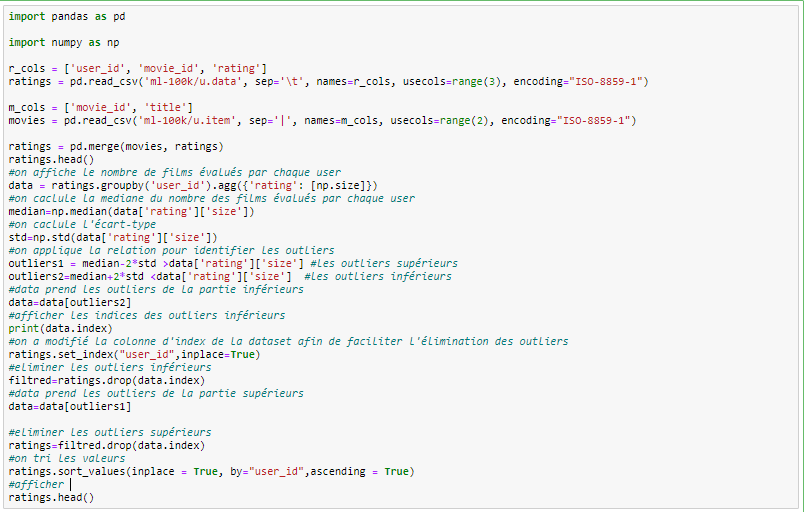


Puis on a suivi la même procédure. On remarque que les coefficients des films recommandés ont été changé ainsi l’ordre de recommandation de quelques films, et cela est du au fait que ces films étaient similaires au film mal évalué.

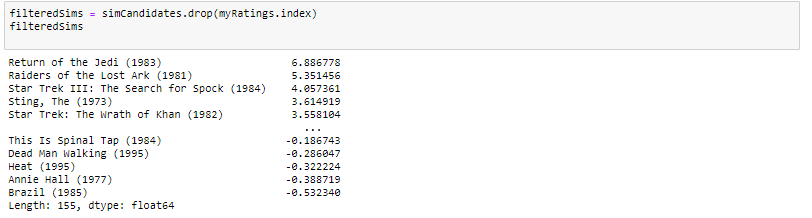


## Elimination des valeurs aberrantes :

Les valeurs aberrantes dans ce cas sont les utilisateurs qui ont évalués beaucoup de films , et on a utilisé la relation déjà montré dans le cours pour les identifier puis les éliminer :



Puis on exécute la suite du programme , le résultat est le suivant :

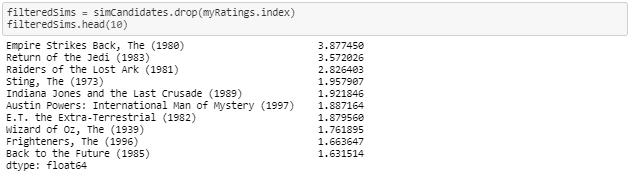


On remarque que les coefficients des films recommandés ont été diminué et c’est trivial car on s’est débarrassé des valeurs aberrantes

# Test du système :

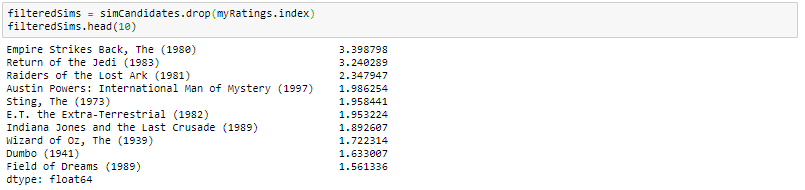
Afin de tester notre système avant et après les modifications alors on va prendre l’user d’ID 0 puis on va enlever le film « Empire Strikes Back The (1980) » ayant un rating 5, et on va voir si le système est capable de lui recommander ce film

## Test du programme sans modification



Le film est recommandé à l’utilisateur donc le système est performant

## Test du programme après modification de méthode : Spearman

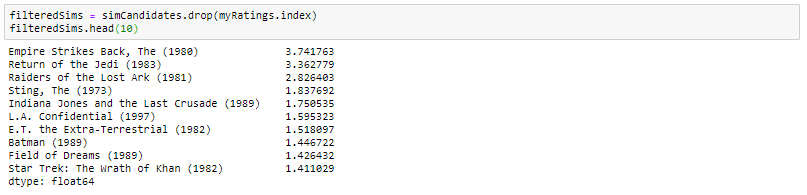


Le film est aussi recommandé mais avec un coefficient inférieur à celui de la méthode Pearson , mais quand même reste performant car il recommande le système.

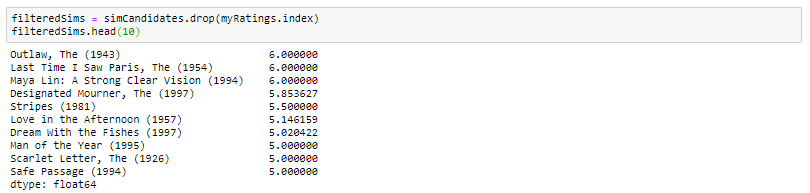
Donc un algorithme de corrélation complexe ne vau pas dire de bonne résultats

## Test du programme après modification de min\_periods : 150

Lorsuq’on augmente la min\_period de plus de 100 le resultat est bon mais avec 100 reste beaucoup mieux.

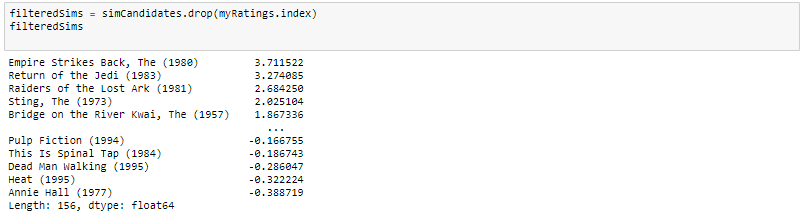


Lorsqu’on diminue la min\_period à 2 on remarque que le film n’est plus recommandé dans la min\_period est un facteur fondamental pour un bon système de recommandation



## Test du programme après élimination des outliers

On remarque que le film est recommandé donc le système est performant après élimination des outliers



# RESUME

Avec la quantité croissante des films produits chaque jour, il devient très difficile pour les utilisateurs de trouver les films les plus adaptées à leurs gouts. Les systèmes de recommandation sont utilisés dans les plateformes d’exposition des films pour résoudre le problème de surcharge des films. Ils sont conçus pour fournir des ressources pertinentes à un utilisateur en utilisant un certain nombre d’informations sur les utilisateurs et sur les ressources(films). Le présent travail s'inscrit dans le contexte des systèmes de recommandation des films aux utilisateurs. Nous avons amélioré une approche de recommandation des films en se basant sur les résultats de recherche dans le domaine des systèmes de recommandation. Nous nous appuyons sur les évaluations des films par les utilisateurs pour améliorer la précision des recommandations. Notre proposition est basée sur des modèles formels qui calculent la similarité entre les utilisateurs d'une plateforme d’exposition des films.

# CONCLUSION

Les systèmes de recommandations aident les utilisateurs à trouver et à sélectionner des articles (par exemple, des livres, des films, des restaurants) parmi l'immense numéro disponible sur le Web ou dans d'autres sources d'information électroniques. Compte tenu d'un grand nombre d'articles et d'une description des besoins de l’utilisateur, ils présentent à l’utilisateur un petit ensemble d’articles bien adaptés à la description. De même, un système de recommandation de films offre un niveau de confort et de personnalisation qui aide l'utilisateur à mieux interagir avec le système et à regarder des films qui répondent à ses besoins.

Fournir ce niveau du confort de l'utilisateur était notre principale motivation en optant pour l’amélioration de ce système de recommandation de films.

Le but principal de notre système est de recommander des films à ses utilisateurs en fonction de leur historique de visionnement et de leurs notes.

Ce petit projet nous a été une opportunité afin d’acquérir plus de notions en machine Learning au-delà de celle acquise en cours, et nous a permet aussi de se familiariser avec python et les librairies utilisés dans l’intelligence artificielle