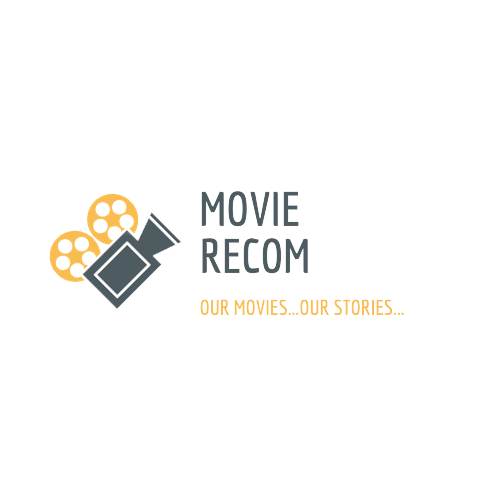
Réalisé par : EL MEHDI EL MOUFID

Numéro d’étudiant :23011001

Soumis le : 16/12/2023

******

***Technologie émergentes & Innovation.***

Encadré par : Mr. Youssef Mekouar

Table des matières

[Tables de figures : 3](#_Toc153636139)

[1- Introduction : 4](#_Toc153636140)

[2- Analyse des Exigences et Use Cases : 4](#_Toc153636141)

[1-Détails des exigences fonctionnelles et non fonctionnelles : 4](#_Toc153636142)

[Collecte de données : 4](#_Toc153636143)

[Algorithme de recommandation : 5](#_Toc153636144)

[Interface utilisateur : 5](#_Toc153636145)

[Performance et accessibilité : 5](#_Toc153636146)

[2- Diagrammes de Use Case et descriptions associées : 6](#_Toc153636147)

[Description des Acteurs : 6](#_Toc153636148)

[Description des Cas d’utilisation : 6](#_Toc153636149)

[3-Scénarios utilisateur détaillés illustrant l'utilisation de l'application : 7](#_Toc153636150)

[3-Démonstration de l'Application : 8](#_Toc153636151)

[Guide étape par étape de l'utilisation de l'application : 8](#_Toc153636152)

[4-Code et développement : 9](#_Toc153636153)

[1-La recherche : 9](#_Toc153636154)

[2-La recommandation : 10](#_Toc153636155)

[5-Conclusion et Perspectives : 11](#_Toc153636156)

[Stockage : 11](#_Toc153636157)

[Traitement : 11](#_Toc153636158)

[Visualisation : 11](#_Toc153636159)

[Références : 11](#_Toc153636160)

Tables de figures :

[Figure 1 : Use Case 6](#_Toc153636019)

[Figure 2 : Interface d'accueil 8](#_Toc153636020)

[Figure 3 : Interface de recommandations 8](#_Toc153636021)

[Figure 4 : Fonction de recherche 9](file:////Users/kesraoui/Documents/MR/Rapport.docx#_Toc153636022)

[Figure 5 : Fonction de recommandation 10](file:////Users/kesraoui/Documents/MR/Rapport.docx#_Toc153636023)

# Introduction :

Les cinéphiles du monde entier ont toujours recherché des moyens plus efficaces de découvrir de nouveaux films susceptibles de correspondre à leurs goûts. Avec la prolifération des services de streaming en ligne et l’abondance des films disponibles, il peut être difficile de choisir parmi toutes les options. C’est là qu’une application Web qui recommande des films s’avère utile.

L'application Web Movie\_Recom est un outil innovant qui aide les utilisateurs à trouver des films qui correspondent à leurs goûts, à découvrir des genres et des réalisateurs auxquels ils n'avaient jamais pensé et à se tenir au courant des dernières sorties de films.

Cette application utilise des algorithmes avancés pour analyser les préférences et les évaluations des utilisateurs et générer des recommandations personnalisées.

Dans le monde en constante évolution du streaming de films et de télévision, les applications Web de recommandation de films constituent un moyen pratique de naviguer dans la richesse du contenu disponible. Il transforme l'expérience visuelle en une aventure plus immersive et enrichissante en encourageant les cinéphiles à sortir de leur zone de confort tout en garantissant qu'ils trouvent toujours un film qui va leur plaire.

Dans cette introduction, nous avons compris les principales fonctionnalités d'une telle application, son importance dans l'industrie cinématographique actuelle et comment elle facilite la recherche de films pour les cinéphiles.

# Analyse des Exigences et Use Cases :

### 1-Détails des exigences fonctionnelles et non fonctionnelles :

### Collecte de données :

Le système doit collecter des informations sur les films disponibles, les genres leur date de sortie et l’évaluation des utilisateurs. Car tous ces derniers peuvent agir dans la recommandation.

### Algorithme de recommandation :

L’application utilise un algorithme de recommandation pour suggérer des films aux utilisateurs en fonction de leurs recherches en se basant sur les notes (ratings) des autres utilisateurs pré-enregistré pour chaque film dans la base de données.

### Interface utilisateur :

L’application fournit une interface conviviale, facile et simple à manipuler permettant aux utilisateurs de rechercher et consulter les recommandations des films avec une touche moderne et un bel affichage, alors le choix de la couleur rouge en référence à NETFLIX a été évident pour donner à l’utilisateur l’impression de réagir avec une interface qui lui ai familière.

Sans obligation de création de compte pour avoir plus de liberté et une plus rapide utilisation.

### Performance et accessibilité :

Les recommandations doivent être générées rapidement, idéalement et en quelques millisecondes. En addition que l’application doit être accessible sur différents appareils (ordinateurs, tablettes, smartphones) et aussi différents navigateurs web.

## 2- Diagrammes de Use Case et descriptions associées :

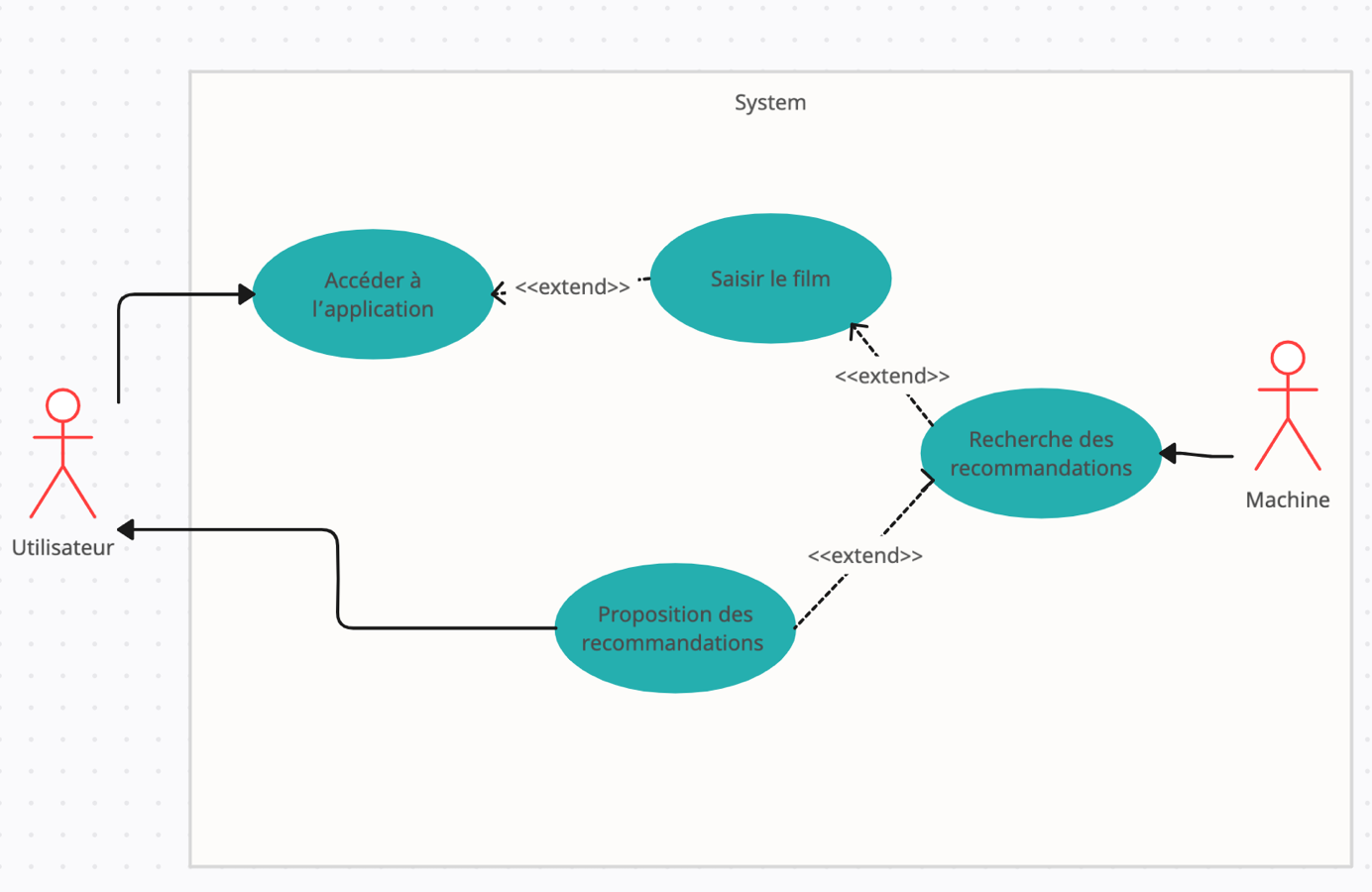


Figure : Use Case

### Description des Acteurs :

1-L’utilisateur : Un utilisateur qui veut utiliser l’application pour recevoir des recommandations des films.

2- La machine : Le système en charge de gérer les fonctionnalités de la recommandation et les données.

### Description des Cas d’utilisation :

Accéder à l’application : Permet à l’utilisateur d’accéder à l’interface d’accueil dans laquelle il pourra visualiser la barre de recherche.

Saisir le film : Remplir la barre de recherche avec le film de son choix pour pouvoir être traité.

Recherche des recommandations : Appliquer l’algorithme de recommandation sur le film donné par l’utilisateur.

Proposition des recommandations : Afficher une liste de 10 films recommandés à l’utilisateur en se basant sur sa saisie.

## 3-Scénarios utilisateur détaillés illustrant l'utilisation de l'application :

Après une journée acharnée de travail Jonathan rentre chez lui pour se détendre un peu et pourquoi pas regardé un film mais tous les films qu’il aime les a regardés minimum une fois, et en connaissant Jonathan il n’aime pas sortir de sa zone de confort, alors y a-t-il un moyen pour lui pour regarder un film qui ne connait pas mais qui va sûrement lui plaire ?

Movie\_Recom alors est la solution ultime, il pourra accéder à l’application faire sa recherche qui nécessite juste à saisir un film qui l’a déjà regardé, qui l’a plu et qui veut regarder un film similaire.

Une fois la saisie est faite l’application lui propose 10 films qui vont lui plaire et pourquoi 10 ? Non pas 5 ou 3 ou même juste un film ? La réponse est évidente car peut-être Jonathan en a déjà regardé alors 10 films est le nombre ultime.

# 3-Démonstration de l'Application :

## Guide étape par étape de l'utilisation de l'application :



Figure : Interface d'accueil

La figure ci-dessus représente l’interface d’accueil que l’utilisateur découvre à chaque utilisation de l’application, c’est une interface facile à comprendre avec une barre de recherche et un bouton de recherche, où l’utilisateur n’aura qu’à saisir son film et cliquer sur le bouton pour démarrer l’engine de recommandation.

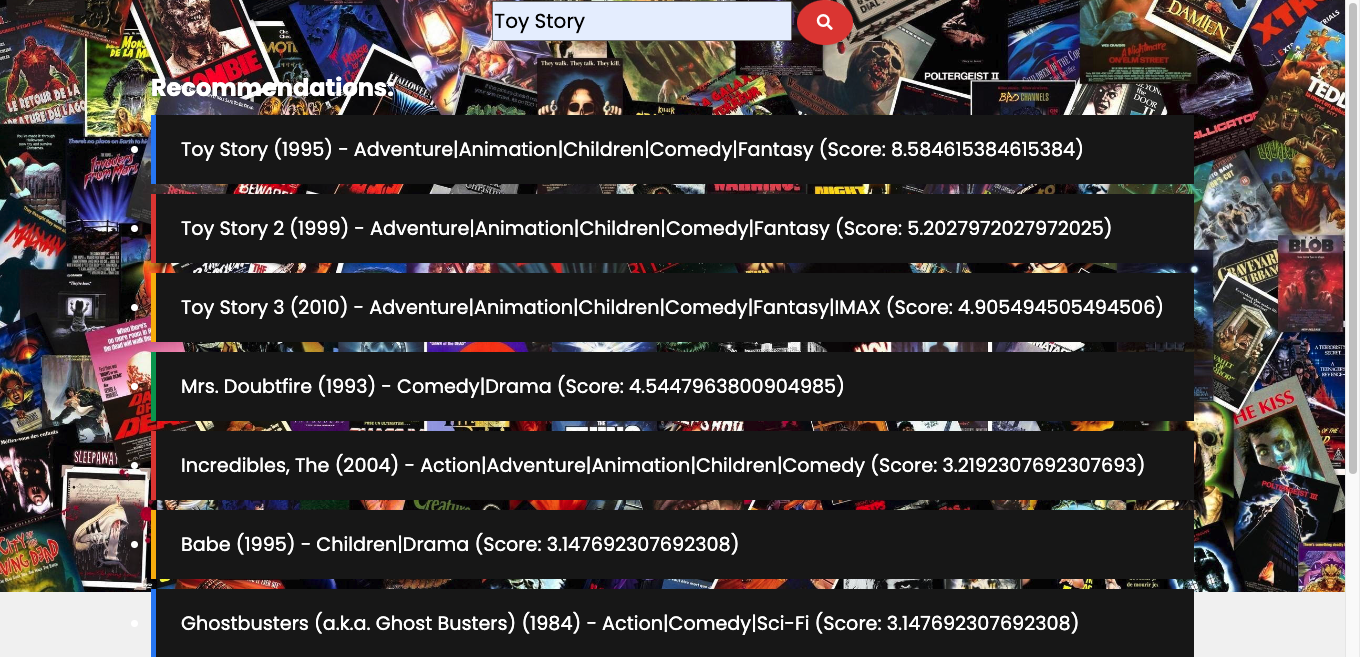


Figure 3 : Interface de recommandations

Une fois la saisie faite le système vas afficher une liste comme l’affiche le figure 3 avec une liste de 10 films qui sont censés plaire à l’utilisateur et va lui afficher aussi les détails dans la liste tels que le nom, la date de production et aussi le genre du film avec aussi un score qui est le score de recommandation ou autrement dis plus le score est haut plus les chances que ce dernier plaise à l’utilisateur.

4-Code et développement :

Le code de cette application a été réaliser avec Python, Flask comme Framework et HTML, CSS pour l’affichage.

La partie la plus intéressante sera celle de la recommandation, comment cette dernière est faite ?

La recommandation est une combinaison de deux fonctions majeures « La recherche » et « La recommandation »

## 1-La recherche :

La fonction de recherche consiste à prendre les titres de tous les films les nettoyer (Garder que les caractères d’a à z, de A à Z, et de 0 à 9), ensuite transformer ces titres en vecteur pour appliquer la similarité cosinus et avoir les 5 films qui ressemble le plus au titre que l’utilisateur a saisi.

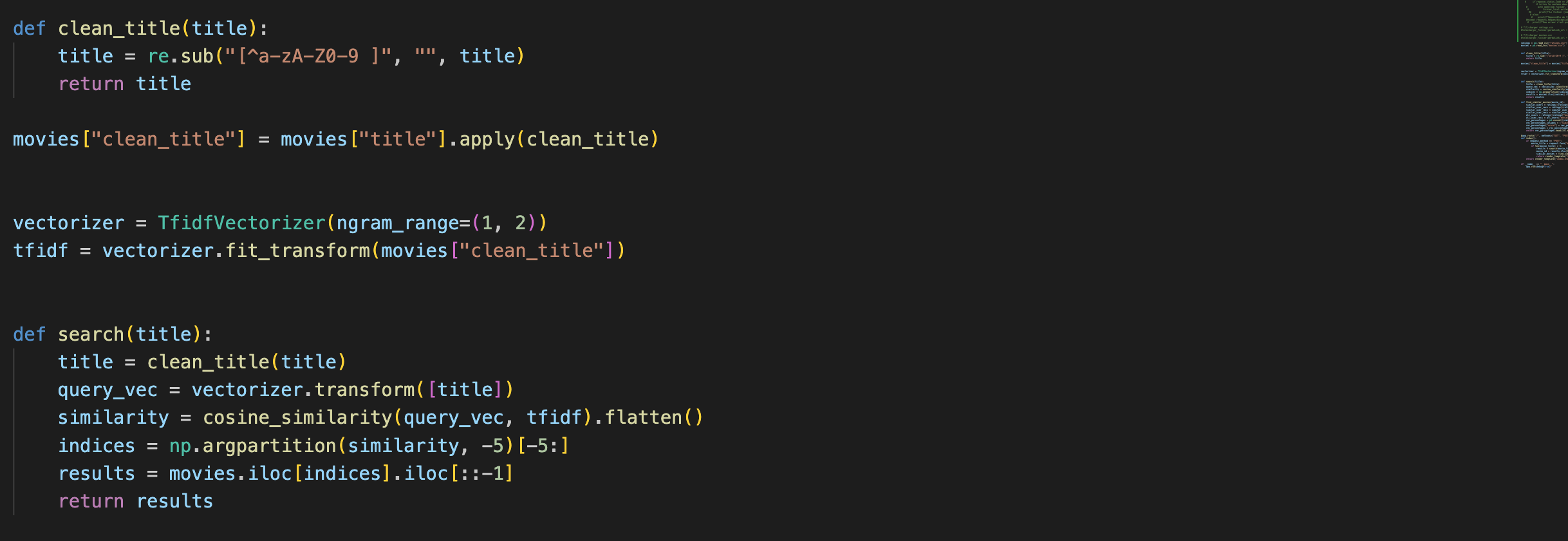


Figure : Fonction de recherche

## 2-La recommandation :

Pour avoir une idée plus claire sur la recommandation je vais définir les variables utilisées dans le process :

* Similar\_users : Les utilisateurs qui ont regardé le film en question et l’ont évalué de plus que 4 sur 5.
* Similar\_user\_recs : Les films que ces « Similar\_users » ont apprécié (On a pris que les films que minimum 10% des utilisateurs similaires on apprécié).
* All\_users : Tous les utilisateurs qui ont regardé des films dans la Similar\_User\_Recs et l’on apprécie.
* All\_user\_recs : Le pourcentage des utilisateurs qui recommande chaque film dans la Similar\_User\_Recs
* rec\_percentages : c’est un tableau qui combine Similar\_User\_Recs et All\_User\_Recs.
* Score : la division de Similar\_User\_recs par All\_User\_recs qui sera ajouté au rableau rec\_percentages.

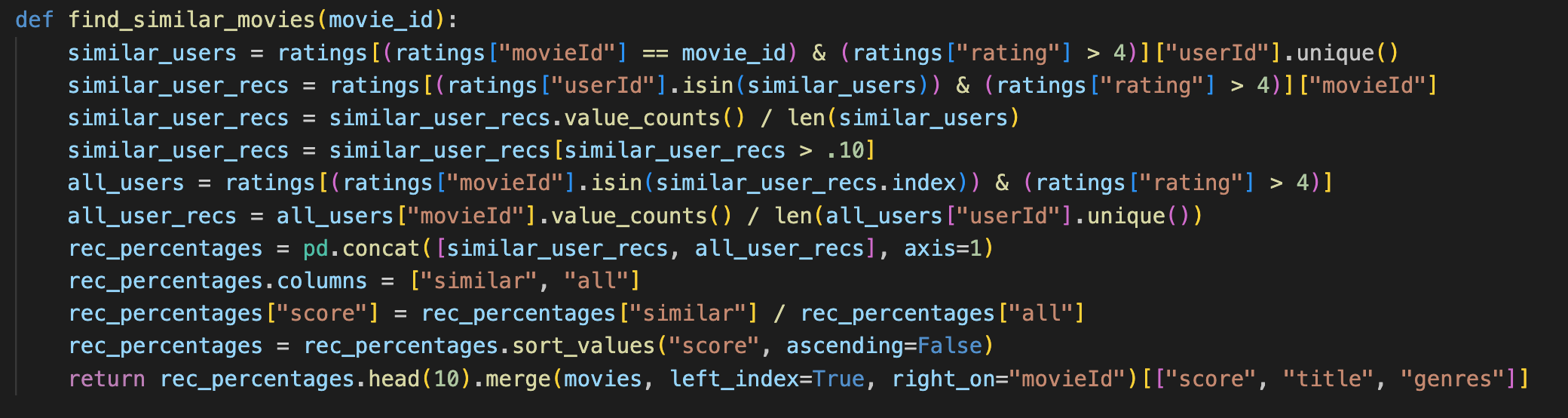


Figure : Fonction de recommandation

En combinant ces deux fonctions on finit par avoir une fonction qui prend un film en entrée et nous retourne 10 recommandations appropriées en sortie en une interface graphique HTML, CSS comme le montre les deux figures 2 et 3.

# 5-Conclusion et Perspectives :

Le système fonctionne correctement sans lenteur et sans ambiguïtés, mais ce qui reste à faire serait de donner la main aux utilisateurs pour cliquer sur les films et les rediriger vers des sites tels que Allo Ciné ou Rotten Tomatoes pour avoir un résumé de film devant eux et pourront ensuite, ensuite mettre en place un login « pas obligatoire » aux utilisateurs pour pouvoir eliminer les films qui ont déjà regardé alors proposer des films avec un score moins haut mais qui est aussi une bonne recommandation.

## Stockage :

Le Stockage ou la recherche de données est faite depuis MovieLens Dataset pour avoir deux fichier qui sont movies.csv et ratings.csv qui seront traité par la bibliothèque Pandas fourni par python ensuite appliquer les fonctions dessus.

## Traitement :

Qui représente la fonctionnalité majeure de cette application est faite comme cité avant dans le rapport, avec la combinaison de deux fonctions de recherche et de recommandation. Où les données stockées seront traitées de leur état brut vers un résultat humanoïde compréhensible par l’homme et qui retourne un résultat satisfaisant.

## Visualisation :

La partie Html/css est la partie de visualisation, la partie UI (User interface) est celle qui fait communiquer l’utilisateur humain avec la machine alors l’objectif derrière c’est qu’elle soit aussi instinctive qu’apaisante pour l’utilisateur.

# Références :

Github : <https://github.com/TalipOrdu/Movie_Recommendation_WebApp>

BlackBox : Une IA pour génère des bouts de codes