

Rapport de Projet: Système de Recommandation de Films

Mohamed Reda Nkira

May 19, 2025

1 Introduction

Ce rapport présente un système de recommandation de films développé en Java, intégrant une interface graphique basée sur Swing. L'application utilise des techniques de filtrage collaboratif pour suggérer aux utilisateurs des films susceptibles de les intéresser, en se basant sur les préférences similaires d'autres utilisateurs.

2 Objectifs du Projet

Les objectifs principaux de ce projet sont les suivants :

- Implémenter un algorithme de filtrage collaboratif basé sur la similarité de Pearson.
- Fournir une interface utilisateur intuitive pour visualiser les recommandations.
- Représenter graphiquement les similarités entre utilisateurs.
- Offrir une expérience utilisateur fluide avec des calculs en arrière-plan.

3 Architecture du Système

3.1 Structure de l'Application

L'application est construite autour d'une classe principale **RecommendationApp** qui étend **JFrame** pour créer l'interface graphique. Le système se compose des éléments suivants :

- Une interface utilisateur avec des contrôles pour sélectionner les utilisateurs.
- Un moteur de recommandation basé sur l'algorithme de similarité de Pearson.
- Un module de visualisation pour représenter les similarités entre utilisateurs.
- Une base de données simulée contenant les évaluations des utilisateurs, stockées en mémoire.

3.2 Modèle de Données

Les données sont structurées comme suit :

- Utilisateurs : Alice, Bob, Charlie, David, Eve.
- Évaluations de films sur une échelle de 1 à 5.
- Films : Star Wars, Harry Potter, Lord of the Rings, Frozen, Inception, Avengers.

4 Algorithmes Implémentés

4.1 Similarité de Pearson

L'algorithme central du système est le calcul du coefficient de corrélation de Pearson entre les préférences des utilisateurs. Cette mesure permet de quantifier la similarité des goûts entre deux utilisateurs.

```
public double pearsonSimilarity(String user1, String user2) {
    Map<String, Double> ratings1 = userRatings.get(user1);
    Map<String, Double> ratings2 = userRatings.get(user2);

    // Identification des films évalués par les deux utilisateurs
    Set<String> sharedItems = new HashSet<>(ratings1.keySet());
    sharedItems.retainAll(ratings2.keySet());

    // Calcul du coefficient de corrélation
    int n = sharedItems.size();
    if (n < 2) return 0;

    double sum1 = 0, sum2 = 0;
    double sum1Sq = 0, sum2Sq = 0;
    double pSum = 0;

    for (String item : sharedItems) {
        double r1 = ratings1.get(item);
        double r2 = ratings2.get(item);

        sum1 += r1;
        sum2 += r2;
        sum1Sq += r1 * r1;
        sum2Sq += r2 * r2;
        pSum += r1 * r2;
    }

    double numerator = pSum - (sum1 * sum2 / n);
    double denominator = Math.sqrt((sum1Sq - (sum1 * sum1) / n) * (sum2Sq - (sum2 * sum2) / n));

    if (denominator == 0) return 0;
    return numerator / denominator;
}
```

4.2 Génération de Recommandations

Le système génère des recommandations en suivant les étapes suivantes :

1. Calculer la similarité entre l'utilisateur sélectionné et tous les autres utilisateurs.
2. Pour chaque film non évalué par l'utilisateur cible, calculer un score pondéré basé sur :
 - Les évaluations des autres utilisateurs pour ce film.
 - La similarité entre ces utilisateurs et l'utilisateur cible.
3. Trier les recommandations par score décroissant.

5 Interface Utilisateur

L'interface utilisateur est conçue pour être intuitive et informative.

5.1 Composants Principaux

Les composants principaux de l'interface incluent :

- Un menu déroulant pour sélectionner l'utilisateur.
- Un bouton pour générer les recommandations.
- Une zone de texte pour afficher les recommandations et analyses.
- Un panneau graphique visualisant les similarités entre utilisateurs.

5.2 Aspects Esthétiques

Les aspects esthétiques de l'interface comprennent :

- Palette de couleurs définie (couleurs primaire, secondaire et de fond).
- Utilisation de boutons stylisés.
- Visualisation des scores sous forme d'étoiles.

5.3 Expérience Utilisateur

L'expérience utilisateur est optimisée par :

- Des calculs effectués en arrière-plan pour éviter de bloquer l'interface.
- Une barre d'état informant l'utilisateur de l'avancement des opérations.
- Des messages d'erreur explicites en cas de problème.

6 Visualisation des Données

Le système inclut une visualisation sous forme de graphique à barres qui :

- Affiche la similarité entre l'utilisateur sélectionné et les autres utilisateurs.
- Utilise des couleurs différentes pour les corrélations positives et négatives.
- Affiche les valeurs numériques précises de similarité.

7 Limites et Améliorations Possibles

7.1 Limites Actuelles

Les limites actuelles du système incluent :

- Base de données en mémoire avec un jeu de données limité et préconfiguré.
- Absence de persistance des données.
- Interface utilisateur non adaptative.

7.2 Améliorations Envisageables

Les améliorations possibles comprennent :

- Connecter le système à une base de données externe.
- Permettre aux utilisateurs d'ajouter leurs propres évaluations.
- Implémenter d'autres algorithmes de recommandation (SVD, filtrage basé sur le contenu).
- Améliorer la visualisation avec des graphiques interactifs.
- Ajouter des fonctionnalités d'exportation des recommandations.
- Implémenter un système d'authentification des utilisateurs.

8 Conclusion

Ce projet démontre une implémentation fonctionnelle d'un système de recommandation basé sur le filtrage collaboratif. L'utilisation de l'algorithme de similarité de Pearson permet de générer des recommandations pertinentes en identifiant les utilisateurs aux goûts similaires.

L'interface graphique offre une expérience utilisateur agréable avec une visualisation claire des similarités entre utilisateurs et des recommandations personnalisées. Bien que le système actuel présente certaines limitations, il constitue une base solide pour des développements futurs plus avancés.

9 Annexes

9.1 Structure des Données de Test

Le système est préconfiguré avec les utilisateurs suivants et leurs évaluations :

Film	Alice	Bob	Charlie	David	Eve
Star Wars	5.0	4.0	1.0	2.0	5.0
Harry Potter	3.0	2.0	5.0	5.0	1.0
Lord of Rings	4.0	5.0	1.0	2.0	4.0
Frozen	1.0	1.0	5.0	4.0	-
Inception	5.0	4.0	2.0	3.0	4.0
Avengers	-	-	-	-	5.0

Table 1: Évaluations des utilisateurs

9.2 Exemple de Résultats

Pour l'utilisateur "Alice", le système identifie :

- Une forte similarité avec "Bob" et "Eve" (préférences pour les films d'action/sci-fi).
- Une corrélation négative avec "Charlie" (préférences opposées).
- Des recommandations potentielles pour "Avengers" basées sur l'évaluation d'Eve.