Truc Minh Thanh LY

Gia Bao Ngoc HOANG

RÉSUMÉ DU PROJET

INTRODUCTION

Le système de recommandation est une forme spécifique de filtrage de l'information visant à présenter les éléments d'information (films, musique, livres, news, images, pages web, etc.) qui sont susceptibles d'intéresser l'utilisateur. Les systèmes de recommandation ont été étudiés dans de nombreux domaines : la recherche d'informations, le web, le e-commerce, l'exploitation des usages du web et bien d'autres. Au cours de ce projet, nous allons créer des systèmes de recommandation pour suggérer des films aux utilisateurs en utilisant les bases de données de MovieLens-1M.

I. PRÉ-TRAITEMENT DE DONNÉES

1. Importer les bases de données

a. Importer les 3 bases principales : « users.dat », « ratings.dat », « movies.dat ».
b. Créer la base « age_interval » pour déterminer les tranches d'âge des utilisateurs.
   ⇨ La variable « age » dans la base « users » est enregistrée sous la forme : 1, 18, 25, 35, 45, 50, 56. Nous voulons les transformer en tranche d'âge : Under 18, 18-24, 25-34, etc.
c. Créer la base "occupation" pour déterminer les métiers des utilisateurs.
   ⇨ La variable « occupation » dans la base « users » est enregistrée sous la forme : 1, 2, 3, 4, etc. Nous voulons les transformer en nom de métier : Artist, college/student, writer, unemployed, engineer, etc.

2. Vue générale des bases de données

a. Déterminer la forme de base « users » : (6040, 5).
b. Déterminer la forme de base « ratings » : (1000209, 4).
c. Déterminer la forme de base « movies » : (3883, 3).

3. Fusionner les bases de données

a. Fusionner les bases « movies » et « ratings » : base « movies_ratings ».
b. Transformer la variable « genre » en matrice binaire.
c. Fusionner la base « movies_ratings » et la matrice binaire de genre.
d. Changer la variable unix_timestamp en date.

II. ANALYSE DESCRIPTIVE

1. **Description de base**

Nombre de films évalués : 1000209

Nombre d'utilisateurs uniques : 6040

Nombre de films uniques : 3706

Le nombre de lignes de doublons : 0

2. **Quelle est la distribution de ratings ?**

a. Regrouper et compter les ratings dans la base « ratings » avec « group by ».
b. Dessiner la distribution : la majorité de ratings est 4 (34,89%), la minorité de rating est 1 (5,62%).

3. **Quelle est la distribution de l'âge des utilisateurs ?**

a. Fusionner les bases « users » et « age_interval ».
b. Regrouper et compter les tranches d'âge avec « group by ».
c. Dessiner la distribution : la majorité des utilisateurs ont de 25 à 34 ans (34,70%). La minorité des utilisateurs ont moins de 18 ans (3,68%).

4. **Quelle est la distribution des métiers des utilisateurs ?**

a. Fusionner les bases « users » et « occupation_df » (nom des métiers).
b. Regrouper et compter les métiers avec « group by ».
c. Dessiner la distribution : 12,57% des utilisateurs sont des étudiants (1ère place), 11,77% sont d'autres métiers, 11,24% sont des cadres. La minorité des utilisateurs sont des agriculteurs.

5. **Quelle est la distribution des sexes des utilisateurs ?**

a. Regrouper et compter les sexes dans la base « users ».
b. Dessiner la distribution : 71,71% d'hommes, 28,29% de femmes.

6. **Quels sont les genres de films les plus regardés par les utilisateurs ?**

a. Compter le nombre d'occurrences de chaque genre.
b. Visualisation : les utilisateurs regardent beaucoup des comédies, drama, action et thriller. Les documentaires sont en dernière place.

7. **Quels sont le nombre et la moyenne de ratings noté par sexe, par tranche d'âge et par métier ?**

a. Ratings par sexe : la moyenne de ratings des deux sexes est assez similaire.

b. Ratings par tranche d'âge : la moyenne de ratings par tranche d'âge est assez similaire.
c. Ratings par métier : la moyenne de rating par métier est assez similaire.

8. Quel est le nombre de ratings et la moyenne de ratings par film ?

Compter le nombre d'avis par film et calculer la note moyenne de chaque film.

9. Densité du nombre de ratings en fonction de film

a. Tracer la densité du nombre de ratings en fonction des films.
b. La distribution est très biaisée, la plupart des films ont moins de 1000 ratings.

10. Quel est le nombre de ratings par genre de films en fonction du sexe ?

a. Compter le nombre de ratings par genre de films en fonction du sexe.
b. Exprimer le nombre de ratings par genre de films en fonction du sexe en pourcentage.
c. Visualisation et commentaires.

11. Quel est le nombre de ratings par genre de films en fonction de tranche d'âge ?

a. Exprimer le nombre de ratings par genre de films en fonction de tranche d'âge en pourcentage.
b. Commentaires.

12. Quel est le nombre de ratings par genre de films en fonction des métiers ?

a. Compter le nombre de ratings par genre de films en fonction des métiers.
b. Commentaires.

13. Quel est le nombre de ratings par utilisateur ?

a. Donner les statistiques de ratings par utilisateur : dans notre base de données, un utilisateur évalue au moins 20 films.
b. Visualisation par la fonction de densité.

III. MODÉLISATION DU SYSTÈME DE RECOMMANDATION

1. Popularity Model

Objectif : Recommander les films les plus populaires et les mieux notés. Les scores sont calculés grâce à la méthode de calcul de IMDB.

Étapes :

a. Calculer le nombre de votes et la moyenne de ratings pour chaque film.
b. Déterminer ensuite le rating moyen de tous les films.
c. Définir le minimum de votes pour entrer dans la liste recommandée.
   ⇨ 80% quantile = 429 votes.
d. Filtrer les films qualifiés.
e. Créer une fonction pour appliquer la formule de score de IMDB.

f. Calculer les scores des films et afficher la liste des films « les plus populaires » dans notre base de données.
   ⇨ 1ère place : Shawshank Redemption, The (1994)
   ⇨ 2e place : The Godfather (1992)

2. Collaborative Filtering by SVD – Surprise Package

Prérequis : Installer *le package Surprise*

Objectif : Recommander des films aux utilisateurs et chercher à prédire les ratings donnés par les utilisateurs.

Étapes :

a. Commencer par splitter les données : 80% en base de train et 20% en base de test.
b. Appliquer ensuite l'algorithme SVD à la base de train.
c. Prédire la note manquante du film i pour l'utilisateur u en appliquant l'algorithme SVD dans la base de test.
d. Évaluer l'approche SVD en calculant la racine carrée de l'erreur quadratique moyenne RMSE.
e. Comparer la note prédite et la note réelle donnée par l'utilisateur.
f. Prédire ensuite le TOP 10 des films que l'utilisateur u pourrait aimer.
   ⇨ Créer la matrice User-Item.
   ⇨ Écrire la fonction qui nous permet de prédire le TOP 10 des films.
g. Terminer en créant une DataFrame se composant des films vus et du TOP 10 des films suggérés à chaque utilisateur selon l'algorithme SVD.

3. Collaborative Filtering : Method-based by User-user Neighbor (N Nearest Neighbor)

Objectif : Recommander des films aux utilisateurs et chercher à prédire l'avis que donnerait un utilisateur.

Étapes :

a. Créer la matrice User-Item.
   ⇨ Splitter tout d'abord la base de données en train et test : 80% en train et 20% en test.
   ⇨ Normaliser la base de train.
   ⇨ Créer la matrice User-Item.
b. Trouver des utilisateurs similaires.
   ⇨ Par distance euclidienne.
   ⇨ Par similarité cosinus.
   ⇨ Répondre aux questions : quelle est la méthode à prendre ? Est-ce que la méthode choisie marche ? Calculer la note espérée pour l'utilisateur en question.
c. Recommander les films.
d. Évaluer l'approche : comparer les notes espérées et les notes réelles données dans la base de test.
e. Limite de l'approche Filtrage Collaboratif par voisinage User-User.

- ⇨ Les nouveaux films ou les films n'ayant pas été évalués par un nombre important d'utilisateurs ne sont pas dans la liste des films recommandés.
- ⇨ L'approche User-User n'est pas rapide quand les utilisateurs sont nombreux.
- ⇨ Dispersion de la matrice (Les matrices **sparse**).

4. The simplest item-item recommendation system

Objectif : Créer le système de recommandation le plus simple (basé sur l'approche item-item) grâce au coefficient de corrélation de Pearson.

Étapes :

a. Créer la matrice user-movies_name.
b. Calculer les coefficients de corrélation de Pearson entre le film en question et les autres films.
c. Classer par ordre décroissant les corrélations.
- ⇨ Recommander les films dont les corrélations sont proches de 1 et dont le nombre de votes est supérieur à 5.
- ⇨ Le résultat n'est pas très pertinent.

5. Item-based collaborative filtering model (Unsupervised NN)

Objectif : Utiliser la méthode de Machine Learning non supervisée Nearest Neighbor pour trouver les "voisins" du film en question.

Étapes :

a. Créer la matrice movies-users.
b. Transformer la matrice user-ratings en forme de "sparse matrix scipy" pour utiliser la méthode de KNN.
c. Définir le modèle NN.
- ⇨ On choisit 20 films (20 voisins) les plus proches à recommander.

d. Fit le modèle.
e. Afficher les recommandations.
- ⇨ Le résultat n'est pas pertinent.

6. Item-based collaborative filtering (by cosine similarity)

Objectif : Créer un modèle item-item basé sur la similarité cosinus.

Étapes :

a. Créer la matrice users-movies.
b. Calculer la "sparsity" de la matrice.
- ⇨ 95,5% cases de la matrice sont vides.

c. Séparer les données en train et test sets.
- ⇨ Retirer 10 ratings aléatoires de chaque utilisateur et les mettre dans le « test set ».

d. Créer la matrice de similarité entre les films par la similarité cosinus.

⇨ Utiliser train set pour créer la matrice de similarité.

e. Prédire les ratings.

⇨ Le rating du film i est déterminé par le total pondéré de similarité de ce film avec les autres films, en normalisant par la valeur absolue de la somme de similarité en fonction des films.

⇨ Le max de rating estimé est de 4,63.

f. Évaluation du modèle par MSE.

⇨ Le MSE est de 12,92 (assez important).

g. Créer la liste de recommandation.

⇨ Le système de recommandation est amélioré mais reste insatisfaisant.

7. User-Item content based model

Objectif : Les méthodes de collaborative filtering donnent des résultats insatisfaisants. Nous essayons de construire un autre système avec l'approche **« content based ».** Cependant, nous n'avons pas assez d'information pour analyser seulement les détails des films (acteurs/actrices, description, directeurs, studio). C'est la raison pour laquelle il faut prendre en compte aussi les utilisateurs. Nous utilisons « genre » de films comme le facteur principal pour analyser.

Étapes :

a. Créer la matrice movies-genres

⇨ Il faut changer la variable « genre » en variables binaires (1 et 0).

b. Créer la matrice user-genres

c. Utiliser la méthode TF-IFD (Term Frequenct-Invverse Document Frequency).

d. Prédire les scores.

⇨ Pour chaque utilisateur, nous trouvons les poids attribués pour tous les films.

e. Créer la liste de recommandation pour les utilisateurs

⇨ Ce modèle est le plus fiable.

CONCLUSION

Chaque approche a ses propres avantages et inconvénients. Il nous semble que le modèle « user-item content based » donne le système de recommandation le plus performant.

BIBLIOGRAPHIE

https://towardsdatascience.com/introduction-to-two-approaches-of-content-based-recommendation-system-fc797460c18c

https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada

https://medium.com/@bindhubalu/content-based-recommender-system-4db1b3de03e7

https://towardsdatascience.com/how-to-build-a-simple-recommender-system-in-python-375093c3fb7d

https://realpython.com/build-recommendation-engine-collaborative-filtering/

https://medium.com/@cfpinela/recommender-systems-user-based-and-item-based-collaborative-filtering-5d5f375a127f

https://www.analyticsvidhya.com/blog/2019/08/5-applications-singular-value-decomposition-svd-data-science/

https://github.com/youonf/recommendation_system/tree/master/content_based_filtering

https://blog.codecentric.de/en/2019/07/recommender-system-movie-lens-dataset/

https://www.kaggle.com/jieyima/netflix-recommendation-collaborative-filtering

https://blog.cambridgespark.com/nowadays-recommender-systems-are-used-to-personalize-your-experience-on-the-web-telling-you-what-120f39b89c3c

http://cedric.cnam.fr/vertigo/Cours/RCP216/coursSimilariteRecommandation.html