Web Mining – Filtrage collaboratif

Nous travaillons sous Python dans cet exercice. Nous utiliserons notamment le package « scikit-surprise » (https://surpriselib.com/) pour la mise en œuvre de l'algorithme SlopOne (mais aussi d'autres de la même libriaire).

A. Exploration des données

Le fichier « **movies_ratings.xlsx** » recense les notes (**rating**) attribuées par des utilisateurs (**user**) à des films (**item**). Ouvrez le fichier dans Excel dans un premier temps pour en appréhender la structure.

1. Importez le fichier. Combien de lignes disposons nous ? (99392) De colonnes ? (3)
2. Combien y a-t-il d'utilisateurs distincts ? (943)
3. Combien y a-t-il de films distincts ? (1664, c'est pas fait exprès)
4. Quelle est la note minimale attribuée par les utilisateurs ? (1) Maximale ? (5)
5. Calculez le nombre d'évaluations réalisées par chaque utilisateur. Quels sont les 5 utilisateurs qui ont le plus évalué ? (user 405 : 735 ; 655 : 677 ; etc.)
6. Quels sont les 5 films qui ont le plus été évalués ? (Star Wars : 583 ; Contact : 509 ; etc.)
7. Calculez les notes moyennes attribuées par chaque utilisateur. Quels sont les 5 utilisateurs les plus généreux ? (849 : 4.869 ; 688 : 4.833 ; etc.)
8. Quels sont les 20 films les mieux notés en moyenne ? (Aiqing wansui : 5.0 ; etc. ; Wallace & Gromit : 4.447761)
9. Pour avoir une vision plus juste de leur popularité, quels sont les 20 films les mieux notés en moyenne parmi ceux qui ont été évalués au moins 50 fois ? (Close Shave : 4.491 (évalué 112 fois), etc., Manchurian Candidate : 4.2595 (évalué 131 fois))
10. Chaque utilisateur est censé noter une seule fois un film. Pouvez-vous vérifier cela ? (oui, c'est le cas effectivement).

B. Préparation des données

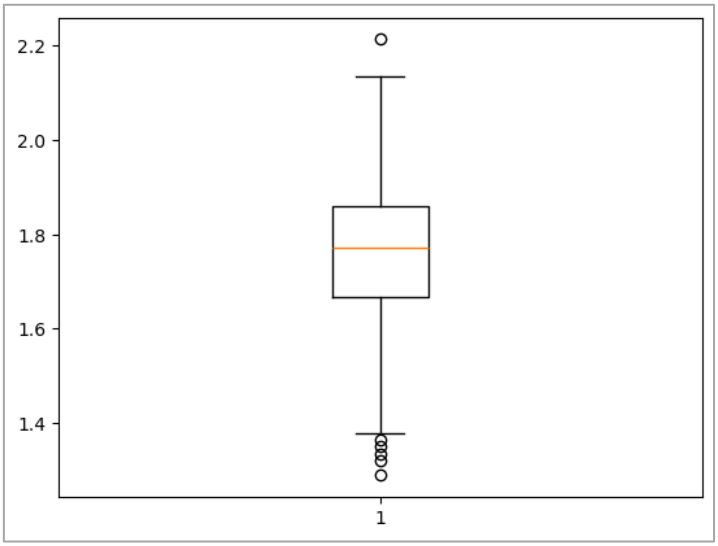
Notre objectif est de prédire au mieux les évaluations du film « **101 Dalmatians (1996)** ».

11. Combien de fois ce film a-t-il été évalué ? (109) Quelle est sa note moyenne ? (2.908257)
12. De manière aléatoire, isolez 50 observations parmi ces 109 dans un data frame à part. Il fera office d'échantillon de TEST.
13. Regroupez les autres individus (tous films confondus) dans un second data frame. Combien d'observations avons-nous ici ? (99392 – 50 = 99342). Ce sera notre échantillon d'apprentissage (TRAIN).

C. Prédiction aléatoire

Si nous effectuons une prédiction aléatoire (une note comprise entre 1 et 5) sur l'échantillon TEST, quel serait notre RMSE ? La valeur obtenue servira de référence. On devra faire mieux (a priori) avec les « vraies » méthodes de recommandation que nous implémenterons par la suite.

14. Générer un vecteur de valeurs aléatoires entières comprises entre 1 et 5. Calculer le RMSE (en utilisant la fonction de « scikit-learn / metrics » par exemple. Quelle valeur obtenez-vous ?
15. Pour disposer d'une vision plus claire de la configuration, répétez 1000 fois l'opération. Calculez la moyenne (**RMSE = 1.76** pour moi, la vôtre devrait être un peu différente, mais dans ces eaux-là). Affichez le boxplot pour vous faire une idée de la dispersion.



D. Prédiction systématique

Une seconde valeur référence du RMSE serait d'effectuer une prédiction systématique à l'aide de la moyenne calculée sur l'échantillon d'apprentissage.

16. Calculez la note moyenne de « 101 Dalmatians » sur l'échantillon TRAIN (2.83 pour moi, votre valeur sera un peu différente j'imagine). Utilisez cette moyenne comme prédiction systématique sur l'échantillon TEST. Toujours en confrontant la note prédite et la note observée, lorsqu'elle est disponible, calculez le RMSE. La méthode est-elle meilleure que la prédiction au hasard ? (**RMSE = 1.07** pour moi, mieux que la prédiction aléatoire déjà).

E. Algorithme Slope-One

Nous souhaitons explorer les algorithmes de la librairie « scikit-surprise » (https://surpriselib.com/), « SlopeOne » (https://surprise.readthedocs.io/en/stable/slope_one.html#surprise.prediction_algorithms.slope_one.SlopeOne) dans un premier temps.

Remarque : La documentation est vraiment nébuleuse à souhait. On s'y noie très facilement. Voici un

des très rares tutos qui tient à peu près la route. Moralité, j'ai beaucoup utilisé COPILOT pour préparer cette partie quitte à, après coup, rechercher chaque étape dans la documentation en ligne.

17. Il nous faut transformer nos données d'apprentissage dans un format compréhensible par la librairie (voir : https://surprise.readthedocs.io/en/stable/getting_started.html#use-a-custom-dataset, voir en particulier la partie qui commence par « To load a dataset from a pandas dataframe… », avec les outils Reader et Dataset). Affichez les premières « .df.head() » et dernières « .df.tail() » du dataset.
18. Transformez le dataset en données d'apprentissage « .build_full_trainset() ».
19. Instanciez l'algorithme « Slope One » (inspirez-vous de l'exemple pour l'algorithme SVD sur cette page : https://surprise.readthedocs.io/en/stable/getting_started.html). Faites alors appel à la méthode « .fit() » pour lancer « l'entraînement » sur le dataset d'apprentissage.
20. Affichez la description du premier individu de l'échantillon test (user, item, rating).
21. Calculée la note escomptée avec la méthode « .predict(…) » (voir l'exemple avec les KNNBasic sur cette page : https://surprise.readthedocs.io/en/stable/getting_started.html, seuls les deux premiers paramètres de .predict() nous concernent).
22. Pour obtenir les notes prédites pour les individus de l'échantillon TEST, appliquez .predict() sur chacun d'entre eux. Quelle est la valeur du RMSE ? (**RMSE = 0.9713** pour ma part, juste un peu mieux que la prédiction systématique).
23. En réalité, il était possible d'obtenir directement l'ensemble des prédictions à l'aide de la méthode « .test() » (cf. https://surprise.readthedocs.io/en/stable/getting_started.html ; sachant que l'échantillon test est dans un data frame à part en ce qui concerne). Obtenons-nous la même valeur de RMSE qu'avec la procédure précédente (oui, sinon très gros problème, effectuer la prédiction pas à pas ou d'une traite ne doit pas produire des résultats différents).

F. Algorithme SVD

24. On observe un benchmark au bas de la page de présentation de l'outil (https://surpriselib.com/). Essayez voir si l'algorithme SVD fait mieux sur notre jeu de données (j'ai mis random_state = 1 et j'ai obtenu **RMSE = 0.969** pour ma part).
(https://surprise.readthedocs.io/en/stable/matrix_factorization.html#surprise.prediction_algorithms.matrix_factorization.SVD).

G. Matrice d'utilité

25. Revenons sur nos données initiales (complètes, sans la partition train / test). Construisez la matrice d'utilité. Quelles en sont les dimensions ? [(943, 1664), forcément].
26. Répondez aux questions n° 5 à 9 à partir de cette matrice. Est-ce que vous retrouvez les résultats ci-dessus (oui, sinon gros problème…).

H. Régression simple

27. Transformez les données TRAIN en matrice d'utilité. Quel est le film le plus lié avec « 101 Dalmatians » au sens du carré de la corrélation ? (bien sûr, il faut exclure les données manquantes des calculs…)
28. A partir de cette variable, construisez un modèle régression simple qui permet de prédire le mieux « 101 Dalmatians ».
29. Appliquez ce modèle sur l'échantillon TEST. Quelle serait la valeur du RMSE alors ?

FIN DU TD…