UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

Big Data et Systèmes de Recommandation

Jordhy Jean Jaurès OTOGONGOUA EMEMAGA

Master 2 Statistique pour l'Évaluation et la Prévision (SEP)

Année 2023 - 2024

Table des matières

1 Introduction

Le big data est devenu un sujet de plus en plus important dans le monde de la technologie. Les entreprises collectent des quantités massives de données sur les utilisateurs et les clients, et cherchent des moyens de les utiliser pour améliorer leurs produits et services. Le Big Data a aussi émergé comme un catalyseur majeur dans l'évolution des systèmes de recommandation, jouant un rôle pivot dans la personnalisation des expériences utilisateur. À l'ère de la surabondance d'informations, la capacité à extraire des connaissances significatives à partir de vastes ensembles de données est devenue un impératif, et c'est là que le Big Data se révèle essentiel.

Cette étude vise à explorer la convergence du Big Data et des systèmes de recommandation, en mettant particulièrement l'accent sur l'utilisation d'outils reconnus tels que Spark pour la manipulation des données massives. L'objectif est de comprendre comment l'exploitation judicieuse du Big Data peut non seulement améliorer la précision des recommandations mais aussi ouvrir de nouvelles perspectives dans divers domaines.

Dans cette perspective, la méthodologie de recherche repose sur la collecte et l'analyse de données pertinentes, suivie de l'application pratique de concepts à travers un exemple concret. Cette approche nous permettra de saisir pleinement l'impact du Big Data sur les systèmes de recommandation et de fournir des insights tangibles pour les chercheurs, les développeurs et les professionnels impliqués dans ce domaine en constante évolution.

Vous pouvez trouver le code source complet de ce projet sur mon profil GitLab à l'adresse suivante : GitLab [1]

2 Big Data

Le concept de big data a évolué pour devenir un moteur fondamental dans la transformation de la manière dont nous percevons, stockons et analysons les données. Il englobe un ensemble de caractéristiques distinctives telles que le **volume** massif de données générées, la **variété** des formats de données, la **vélocité** à laquelle les données sont produites et leur valeur potentielle pour l'analyse.

Le volume impressionnant de données générées quotidiennement, provenant de diverses sources telles que les médias sociaux, les transactions en ligne, et les capteurs connectés, nécessite des solutions adaptées pour les gérer efficacement. Le big data offre ces solutions en utilisant des technologies telles que le stockage distribué et le traitement parallèle, permettant ainsi de manipuler et d'analyser ces vastes ensembles de données de manière efficiente.

Des entreprises renommées, à l'image de Netflix, ont adopté avec succès le Big Data pour optimiser leurs systèmes de recommandation. En exploitant le big data, ces entreprises peuvent analyser les habitudes d'utilisation, les préférences et les comportements des utilisateurs en temps réel, ouvrant ainsi la voie à des recommandations plus précises et personnalisées.

En résumé, le big data représente bien plus qu'une simple quantité massive de données ; c'est un catalyseur qui transforme la façon dont nous abordons l'analyse des informations. Dans le contexte des systèmes de recommandation, il offre la possibilité d'exploiter des données à une échelle sans précédent pour améliorer la précision des recommandations et offrir des expériences utilisateur plus enrichissantes.

Pour notre étude, nous mettrons l'accent sur l'utilisation de la bibliothèque PySpark, spécialement conçue pour le traitement distribué en Python, afin de tirer parti des avantages offerts par le big data.

3 Systèmes de recommandation

Les systèmes de recommandation sont devenus omniprésents dans notre quotidien numérique, influençant les décisions d'achat, les choix de divertissement et bien d'autres aspects de notre expérience en ligne. Ces systèmes sont conçus pour anticiper et répondre aux besoins des utilisateurs en suggérant des éléments pertinents, basés sur leurs préférences et leur comportement passé.

Généralement, les systémes de recommandation se divisent en trois catégories principales : le **filtrage collaboratif**, le **filtrage basé sur le contenu**, et les **approches hybrides**. Le filtrage collaboratif repose sur le comportement d'utilisateurs similaires, tandis que le filtrage basé sur le contenu se concentre sur les caractéristiques des éléments eux-mêmes. Les approches hybrides combinent ces deux méthodes pour tirer parti de leurs avantages respectifs.

Pour notre étude, nous utilisons la technique de filtrage collaboratif pour générer des recommandations de films pour les utilisateurs. L'algorithme spécifique utilisé ici est l'Alternating Least Squares (ALS).

3.1 Fonctionnement et architecture du modèle ALS

ALS est un algorithme d'optimisation itératif qui minimise l'erreur quadratique entre les évaluations observées et les évaluations prédites. Il alterne entre l'optimisation des facteurs utilisateur (représentant les préférences utilisateur) et des facteurs d'élément (représentant les caractéristiques des éléments) jusqu'à convergence.

Formulation mathématique :
Définissons les termes clés et les variables utilisés dans l'algorithme ALS :
— R : La matrice d'évaluation utilisateur-élément, où chaque entrée r_{ui} représente l'évaluation donnée par l'utilisateur u à l'élément i.
— P : La matrice des facteurs utilisateur, où chaque ligne p_u représente les facteurs latents pour l'utilisateur u.
— Q : La matrice des facteurs d'élément, où chaque ligne q_i représente les facteurs latents pour l'élément i.
— $\hat{\mathrm{R}}$: La matrice d'évaluation prédite, obtenue par le produit scalaire de P et de Q^T
La fonction objective d'ALS peut être formulée comme suite :

$$\min_{P,Q} \sum_{(u,i)\in\Omega} (r_{ui} - p_u \cdot q_i)^2 + \lambda \left(\sum_u ||p_u||^2 + \sum_i ||q_i||^2 \right) [2] \tag{1}$$

Où :
— Ω représente l'ensemble des paires utilisateur-élément observées.
— λ est le paramètre de régularisation pour prévenir le surajustement.
— $\|\cdot\|$ la norme L2.
Les règles de mise à jour pour P et Q dans chaque itération peuvent être exprimées comme suit :

$$p_u = (Q^T Q + \lambda I)^{-1} Q^T r_u \tag{2}$$

$$q_i = (P^T P + \lambda I)^{-1} P^T r_i \tag{3}$$

Où r_u et r_i sont des vecteurs contenant les évaluations données par l'utilisateur uu et l'élément i respectivement, I est la matrice identité.

En mettant à jour de manière itérative P et Q, l'algorithme ALS converge vers une solution où les évaluations prédites correspondent étroitement aux évaluations observées dans RR, permettant des recommandations précises de films pour les utilisateurs.

Parmi les plus grands utilisatueurs de systèmes de recommandation, on peut citer **Netflix** et **LinkedIn**.

Netflix [6] utilise l'algorithme SVD pour recommander des films et des émissions de télévision à ses utilisateurs. Leur algorithme utilise la notation des utilisateurs pour prédire les notes que les utilisateurs donneraient à des films qu'ils n'ont pas encore vus.

LinkedIn [3] utilise également l'algorithme SVD pour recommander des connexions et des offres d'emploi à ses utilisateurs. Leur algorithme utilise des informations sur les compétences et les expériences des utilisateurs pour recommander des connexions et des offres d'emploi qui pourraient les intéresser.

Étant donné que les bases de données, notamment la base de données qui sera utilisée dans l'étude de cas, peuvent être volumineuses, l'utilisation d'algorithmes de recommandation sur de telles données peut être complexe en raison des exigences en matière de calcul et de mémoire. Pour faciliter le processus, Spark a directement mis à disposition le modèle ALS (Alternating Least Squares).

L'algorithme ALS est particulièrement adapté aux environnements distribués comme Spark en raison de sa nature itérative et parallélisable. Spark permet de distribuer le processus de calcul sur un cluster, ce qui permet de traiter efficacement de grandes quantités de données.

4 Etude de cas

Pour la mise en pratique, nous allons nous concentrer sur un cas de recommandation de contenu cinématographique. Le jeu de données utilisé provient de **GroupLens** [4]. GroupLens est un laboratoire de recherche du département d'informatique et d'ingénierie de l'Université du Minnesota, Twin Cities spécialisée dans les systèmes de recommandation, les communautés en ligne, les technologies mobiles et omniprésentes, les bibliothèques numériques et les systèmes d'information géographique locaux.

4.1 Jeu de données

Le jeu de données qui sera utilisée à titre d'exemple appplicatif est la base **MovieLens 25M Dataset** [5]. Cette base constitue une ressource substantielle pour notre étude de cas sur les systèmes de recommandation. Cette base de données compile 25 millions d'évaluations concernat 62423 films provenant d'utilisateurs du site MovieLens. Chaque évaluation est associée à des informations détaillées sur le film correspondant, fournissant ainsi une base solide pour analyser les préférences cinématographiques des utilisateurs. Ce nombre important d'observations montre clairement l'aspect **Big** de ces données et nous recadre clairement dans le contexte big data.

Cette base est organisée en plusieurs tables comprenant des informations sur les utilisateurs, les films et les évaluations.

- **movies** : Contient les informations relatives à chaque film. Il contient les colonnes ‘movieId‘ correspondant à l'identifiant du film, ‘title‘ pour le titre du film et ‘genres‘ pour le genre de film associé (Adventure, Comedy, ...) un seul film pouvant être associé plusieurs catégories à la fois ;
- **ratings** : Contient les notes attribuées à chaque film par les utilisateurs. Cette table contient les colonnes ‘userId‘ pour l'utilisateur ayant attribué la note, ‘movieId‘, ‘rating‘ pour la note attribuée (de 0.5 à 5.0) et ‘timestamp‘ pour la date d'attribution de la note (le format timestamp correspondant au nombre de secondes depuis le 1er janvier 1970) ;
- **tags** : Contient les descriptifs attribués par les utilisateurs à chaque film. Elle contient les colonnes ‘userId‘, ‘movieId‘ et ‘tag‘ correspondant au descriptif attribué.

4.2 Collecte des données

Les données ont été téléchargées directement sur le site de MovieLens [5] dans une format compressé (zip) d'une taille de 263 mo et après décompression, l'ensemble des fichiers a une taille de 1.2 go. On retrouve bien un aspect de volumétrie importante dans le cadre d'une gestion en local.

4.3 Traitements et nettoyage de la base

Tout d'abord, les ensembles de données sur les films (**movies**) et les évaluations (**ratings**) ont été joints à l'aide de **PySpark** en raison de problèmes d'allocation de mémoire rencontrés lors de la tentative de jointure avec **Pandas**. Ces bases ont été choisies car elles contiennet l'essentiel des informations nécessaires à la réalisation de la tâche et la clé de jointure utilisée est "movieId".

Une fois les ensembles de données joints avec succès, la première étape du nettoyage des données a consisté à supprimer les films avec un genre inconnu, marqué comme "(no genres listed)". Il est important de noter qu'il n'y avait aucune valeur manquante dans l'ensemble de données, ce qui a simplifié le processus de nettoyage. Aussi, les dates (colonne "timestamp") ont été mises au format standard Y-M-D h-m-s.

Un filtrage a été appliqué sur les données pour retirer les films ayant reçus moins de 5 votes. La raison étant que les notes de ces films se retouvent excellente, comparé à certains ayant reçu plus de 1000 vites et dont la note générale se trouve réduite. Cela permet également de faire des comparaisons des notes moyennes sur une échelle raisonnable.

L'API DataFrame de PySpark a facilité les opérations efficaces de manipulation et de nettoyage des données, permettant une gestion distribuée des ensembles de données.

4.4 Analyse de la base

Les figures 2 et 3 présentent la distribution des notes de films dans notre base de données. Il s'agit d'un outil visuel essentiel pour comprendre la répartition des évaluations attribuées par les spectateurs. Analysons de plus près les caractéristiques de cet histogramme :

- Échelle des notes : Les notes sont échelonnées de 0,5 à 5,0, avec des incréments de 0,5. Plus la note est élevée, meilleure est la réception du film.
- Barres oranges : Elles représentent les films mal notés (entre 0,5 et 2).
- Barres jaunes : Ces barres indiquent les films moyennement notés (autour de 3).
- Barres bleues : Elles représentent les films bien notés (entre 4 et 4,5).

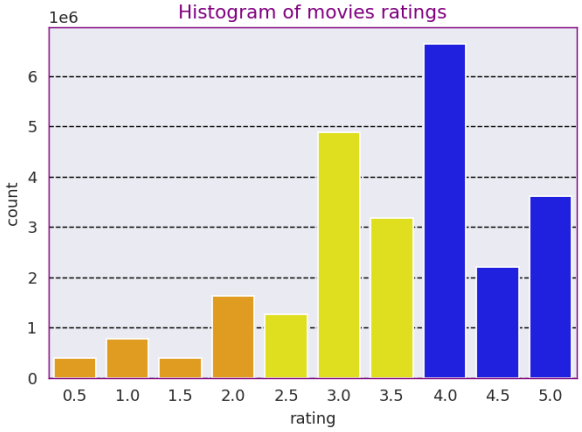

FIGURE 2 – Diagramme à barres de répartition des notes de films

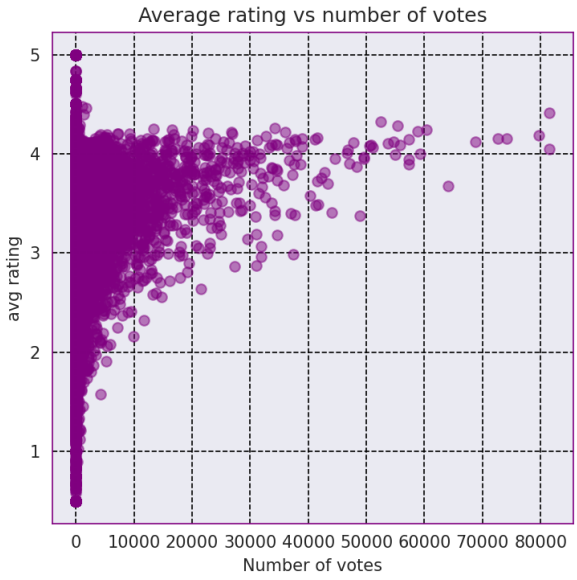

FIGURE 3 – Répartition des notes selon le nombre de votes

En utilisant cet histogramme, nous pouvons identifier les tendances et les préférences des spectateurs en matière de notation. Il s'en dégage que la majorité des films présents est plutôt bien appréciée par les votants. Les notes les plus attribuées sont 4.0, 3.0 et 5.0.

D'autre part, L'histogramme des votes 4 par genre de films nous offre un aperçu intéressant de la préférence du public. Les principales observations montrent que Le genre dramatique (Drama) est en tête avec le plus grand nombre de votes. Cela

suggère que les films dramatiques attirent l'attention, ou du moins que c'est pratiquement le genre de film le plus regardé par les spectateurs ayan voté. Les comédies (Comedy) et films d'action (Action) arrivent en deuxième et troisième position, avec un nombre de votes significatif. On observe également que les documentaires (Documentary) et films noires (Film Noir) sont les moins notés

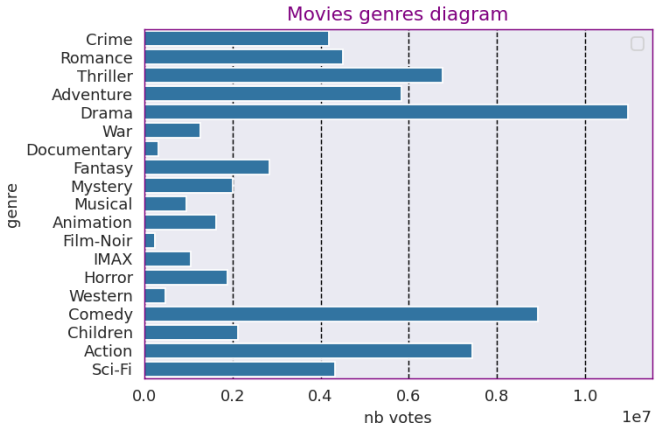

FIGURE 4 – Diagramme à barres de répartition du nombre de notes attribuées par genre de film

En résumé, les spectateurs semblent privilégier les drames et les comédies, tandis que les genres plus spécifiques ont également leur propre public. Cette répartition des votes peut guider les décisions de programmation et de production cinématographique.

On peut également obtenir un classement des films les mieux notés en moyenne comme présenté sur la figure 5. On peut noter qu'un film tel que Planet Earth II (2016), malgré le nombre important de votes, conservent tout de même une note élevée, 4.3 en l'occurrence ? Cela traduit clairement la qualité ou l'appréciation de ce film. Cela permet de se rendre compte de l'importance à accoredre aux nombres de votes reçus par un film pour juger de sa qualité.

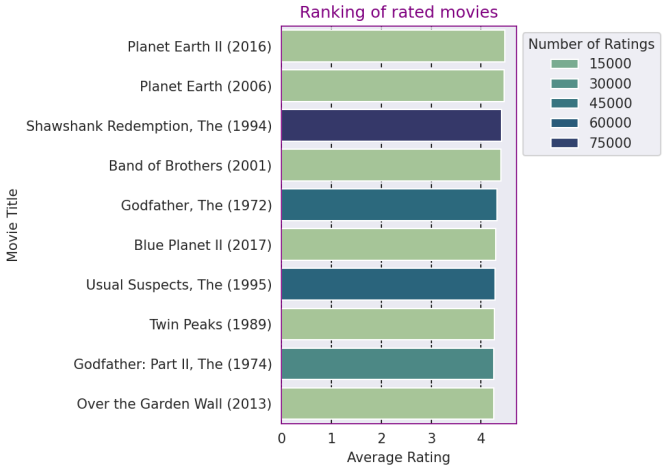

FIGURE 5 – Classement des films selon la note moyenne

ces prmiers résultats sont bien sur influencés par le temps, en ce sent que l'apparition permanente de nouveaux films et leur popularité influence le nombre et la qualité des votes qu'ills reçoivent. ainsi les films plus anciens ne vont certainement plus recevoir plus de votes en raison de l'apparition des nouveaux. Mais on peut tout de même analyser les préférences des spectateurs en termes de genres de films. Pour cela, on peut particulièrement s'intéresser à l'évolution du top 5 des genres de films selon l'année, en fonction de la note moyenne reçue par catégorie. C'est l'intérêt de la visualisation proposée dans la figure 6.

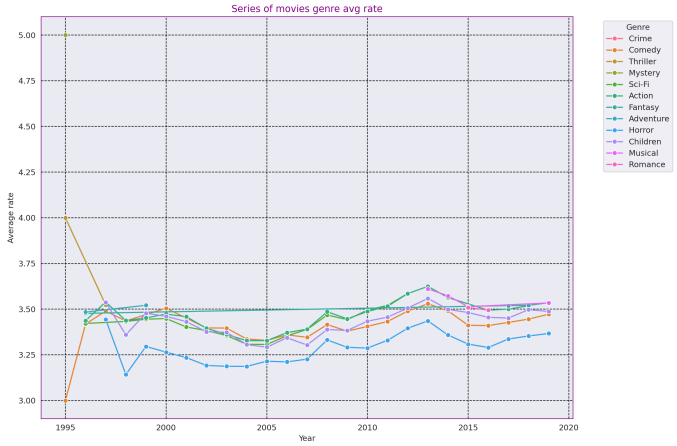

FIGURE 6 – Evolution annuelle du classement des genres de films selon leur note moyenne

4.5 Modèle ALS

Une fois les données traitées, la mise en place du modèle a pu être élaborée. Les étapes en sont les suivantes :

- prétraitement supplémentaire des données : Les données ont été prétraitées pour éliminer les films peu populaires (ayant reçu moins de 100 votes), afin de garantir des recommandations pertinentes.
- Division des données : Les données ont été divisées en ensembles d'entraînement et de test dans un ratio de 80/20. Cela nous a permis d'entraîner le modèle sur une partie des données et de l'évaluer sur une autre partie afin d'approximer sa performance.
- Entraînement du modèle : Pendant l'entraînement, nous avons effectué une recherche sur la grille des hyperparamètres pour trouver les meilleures valeurs des paramètres du modèle, telles que le nombre de facteurs latents et les paramètres de régularisation. L'entraînement aura pris 6.5 minutes.
- Évaluation du modèle : Une fois le modèle entraîné, nous l'avons évalué sur l'ensemble de test en calculant la racine de l'erreur quadratique moyenne (RMSE). Cette mesure nous a donné une indication de la précision du modèle dans la prédiction des évaluations des utilisateurs sur les films.
- Sauvegarde du modèle : Enfin, le meilleur modèle obtenu après la recherche sur la grille des hyperparamètres a été sauvegardé pour une utilisation future.

Résultats :

Le modèle entraîné a donné un RMSE de 0.83 sur l'ensemble de test, ce qui indique une bonne adéquation entre les évaluations prédites et les évaluations réelles des utilisateurs. La sauvegarde réussie du modèle nous a permis de le tester facilement pour effectuer des recommandations pour un autre utilisateur présent dans la base.

Nous avons donc rendu la réutilisation du modèle entraîné en proposant un script exécutable en ligne de commande. Pour ce faire, l'utilisateur doit spécifier l'identifiant de l'utilisateur pour lequel les recommandations doivent être générées ainsi que le nombre de recommandations souhaitées. Le script utilise ensuite le modèle pour générer des recommandations de films pour cet utilisateur. Il en résulte une liste de films recommandés, affichant les titres et les genres des films recommandés pour l'utilisateur spécifié.

5 Conclusion

Ce projet de recommandation de films a été une expérience enrichissante qui m'a permis de mettre en pratique des compétences avancées en Big Data, en particulier l'utilisation de Spark pour manipuler des ensembles de données volumineux. L'un des principaux défis rencontrés lors de ce projet était la volumétrie des données, qui dépassait les capacités des bibliothèques standard de Python telles que Pandas. Cela m'a conduit à adopter des outils plus puissants comme Spark pour gérer et analyser efficacement ces données massives.

Travailler sur un sujet aussi divertissant que la recommandation de films a été un réel plaisir. Non seulement cela a été une expérience gratifiante du point de vue technique, mais cela m'a également permis d'explorer un domaine qui m'intéresse personnellement. La recommandation de films a des implications importantes dans les domaines de l'industrie du divertissement et du commerce électronique, où la personnalisation des recommandations peut influencer les décisions des utilisateurs et améliorer leur expérience.

L'utilisation de Spark s'est avérée être un choix judicieux, car elle a grandement facilité la manipulation et l'analyse des données à grande échelle. Sa capacité à distribuer le traitement sur un cluster de machines a permis de traiter efficacement les données massives tout en offrant des performances élevées. Cela a ouvert de nouvelles possibilités dans la façon dont nous pouvons aborder et résoudre des problèmes liés au Big Data, démontrant ainsi le potentiel et la valeur de cette technologie dans le domaine de l'analyse de données.

Références

[1] `https://gitlab-mi.univ-reims.fr/otog0001/outils-big-data.git`.

[2] `https://nightlies.apache.org/flink/flink-docs-release-1.2/dev/libs/ml/als.html`.

[3] `https://www.emarketinglicious.fr/reseaux-sociaux/algorithme-linkedin/`.

[4] `https://grouplens.org/`.

[5] `https://grouplens.org/datasets/movielens/25m/`.

[6] Robert Bell Yehuda Koren and Chris Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer Society*, 2009.